Initiation à SAS® PC

Loïc Desquilbet, PhD

Département des **P**roductions **A**nimales et de **S**anté **P**ublique

Ecole **N**ationale **V**étérinaire d'**A**lfort

ldesquilbet@vet-alfort.fr

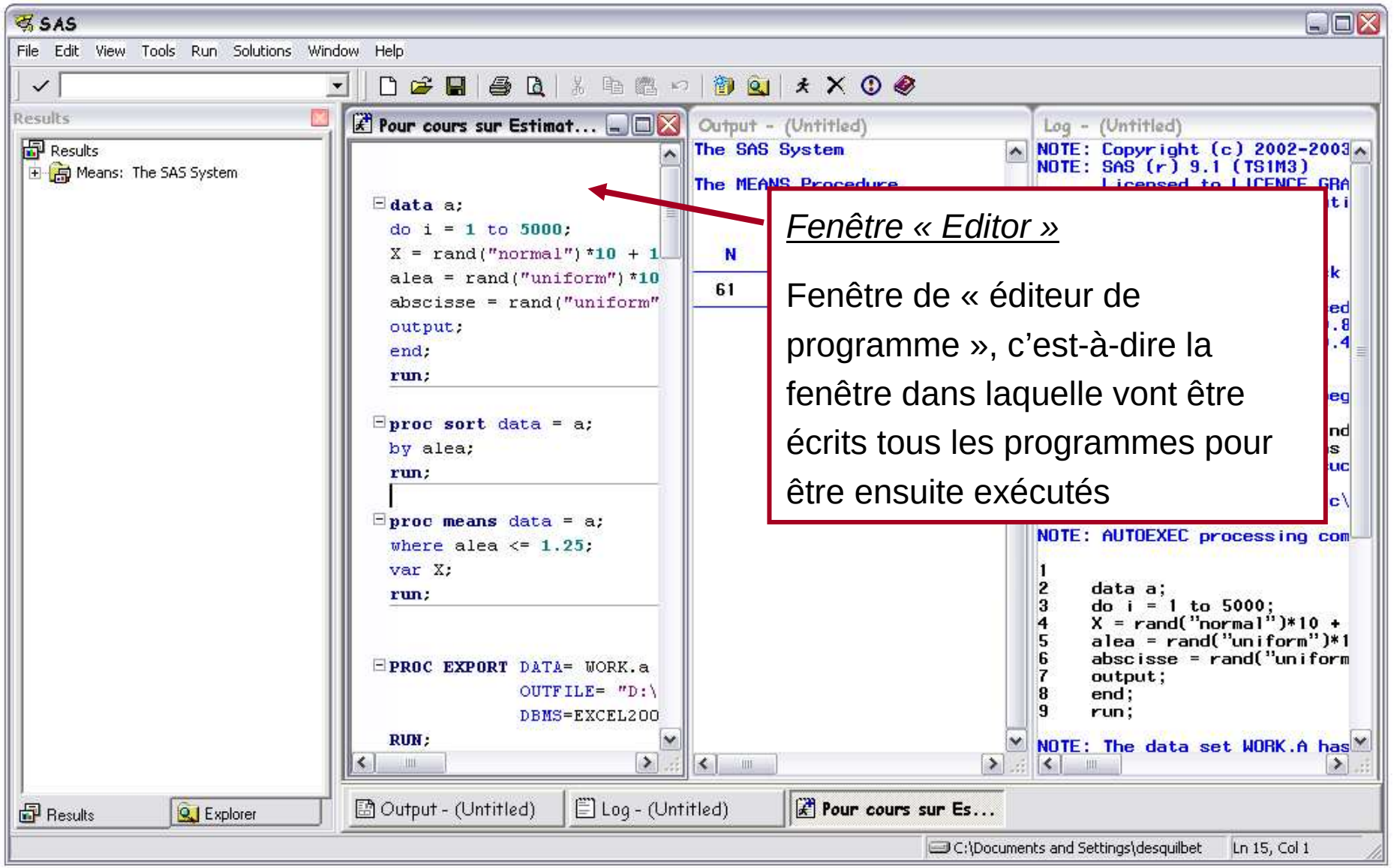
Version v1

Présentation générale de SAS

Présentation de SAS

- SAS est un des principaux logiciels de statistiques, et possède son propre langage pour gérer les données
- Pas idéal pour la saisie des données !
- SAS comporte aussi des modules utilisés dans des domaines comme la finance, la géographie ou l'analyse décisionnelle
- Fonctionne sur PC ou sur gros système (Unix) avec une licence annuelle
- Version présentée ici : 9.1.3, en anglais…

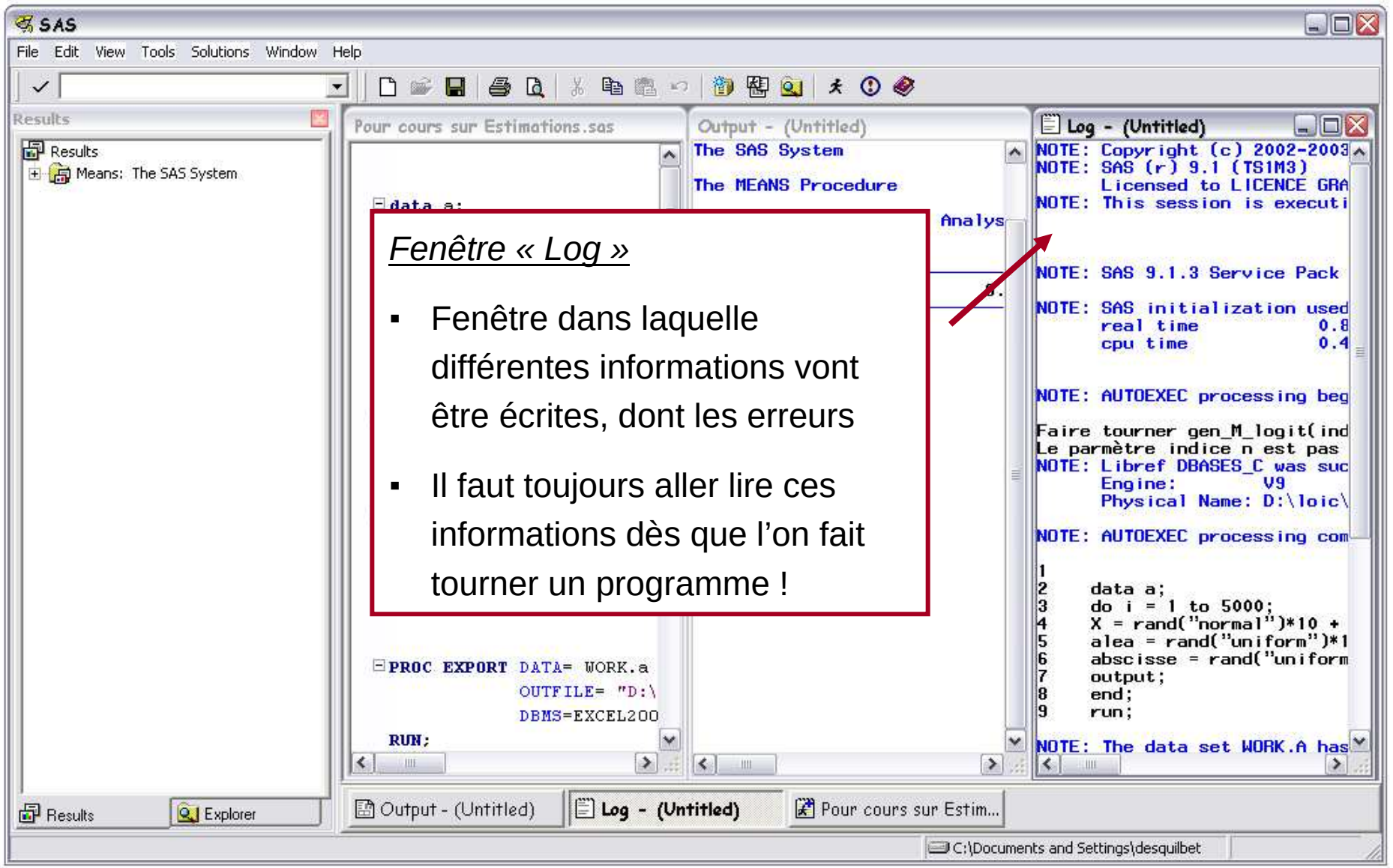
Les fenêtres de SAS



Fenêtre « Editor »

Fenêtre de « éditeur de programme », c'est-à-dire la fenêtre dans laquelle vont être écrits tous les programmes pour être ensuite exécutés

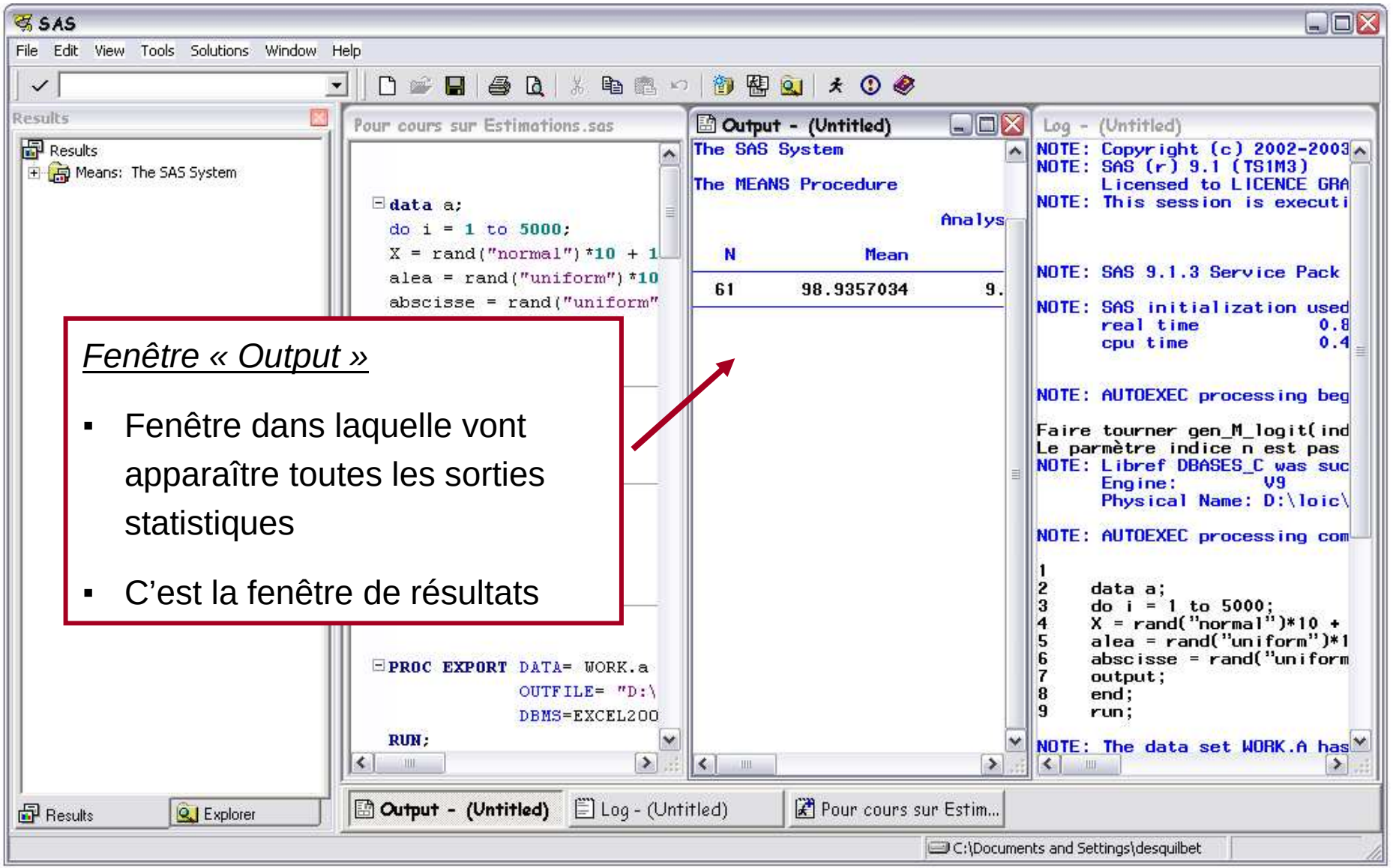
Les fenêtres de SAS



Fenêtre « Log »

- Fenêtre dans laquelle différentes informations vont être écrites, dont les erreurs
- Il faut toujours aller lire ces informations dès que l'on fait tourner un programme !

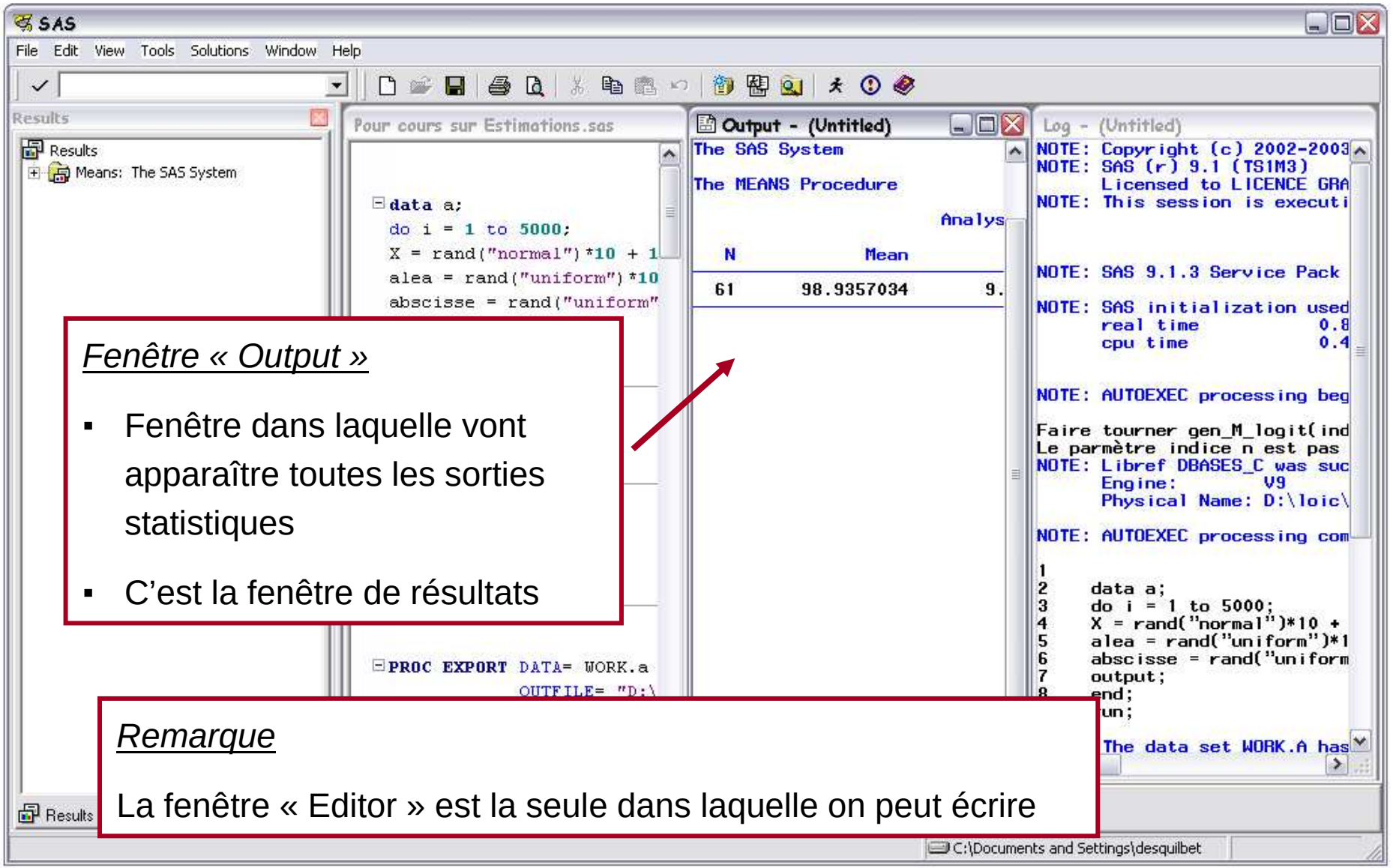
Les fenêtres de SAS



Fenêtre « Output »

- Fenêtre dans laquelle vont apparaître toutes les sorties statistiques
- C'est la fenêtre de résultats

Les fenêtres de SAS



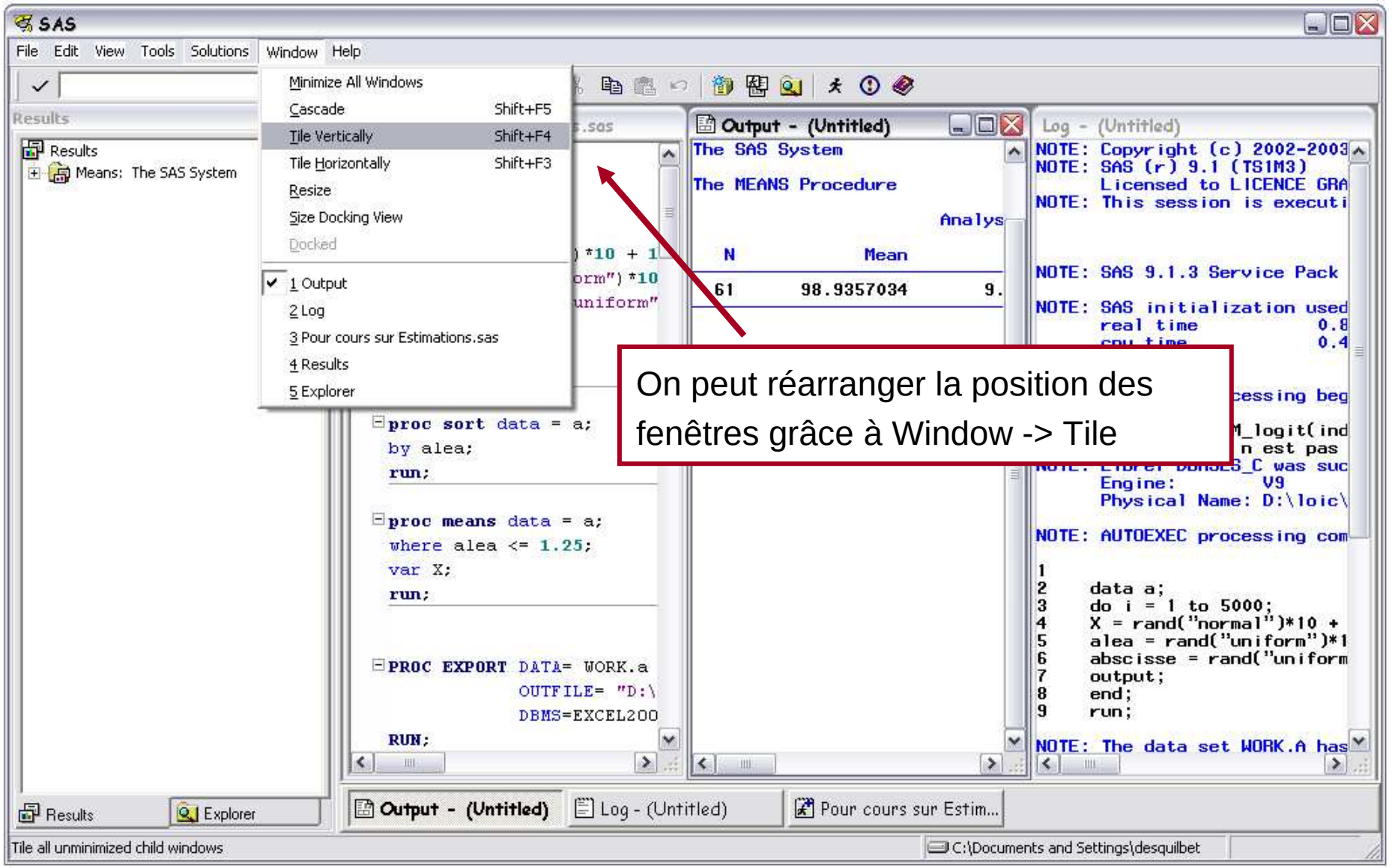
Fenêtre « Output »

- Fenêtre dans laquelle vont apparaître toutes les sorties statistiques
- C'est la fenêtre de résultats

Remarque

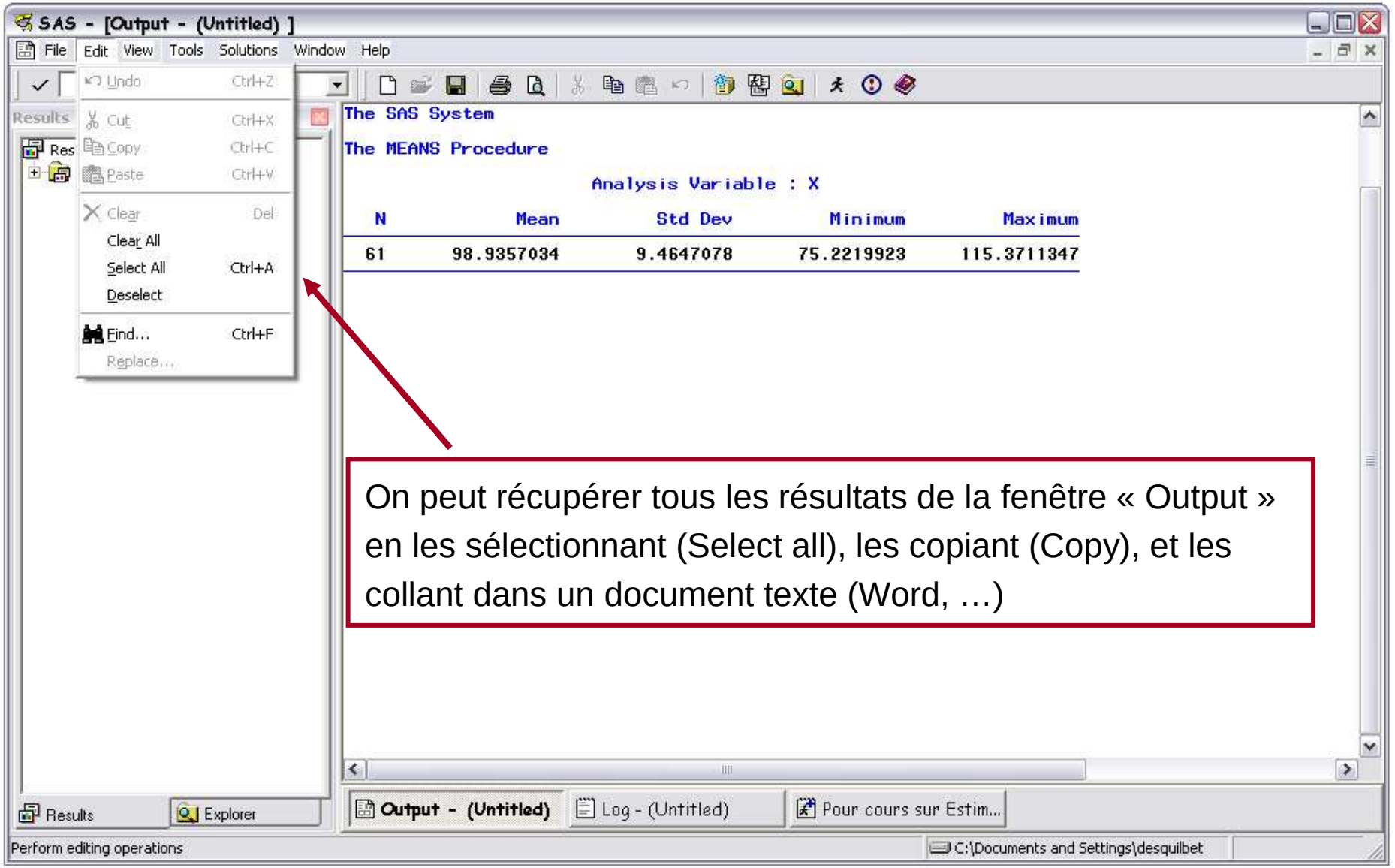
La fenêtre « Editor » est la seule dans laquelle on peut écrire

Les fenêtres de SAS



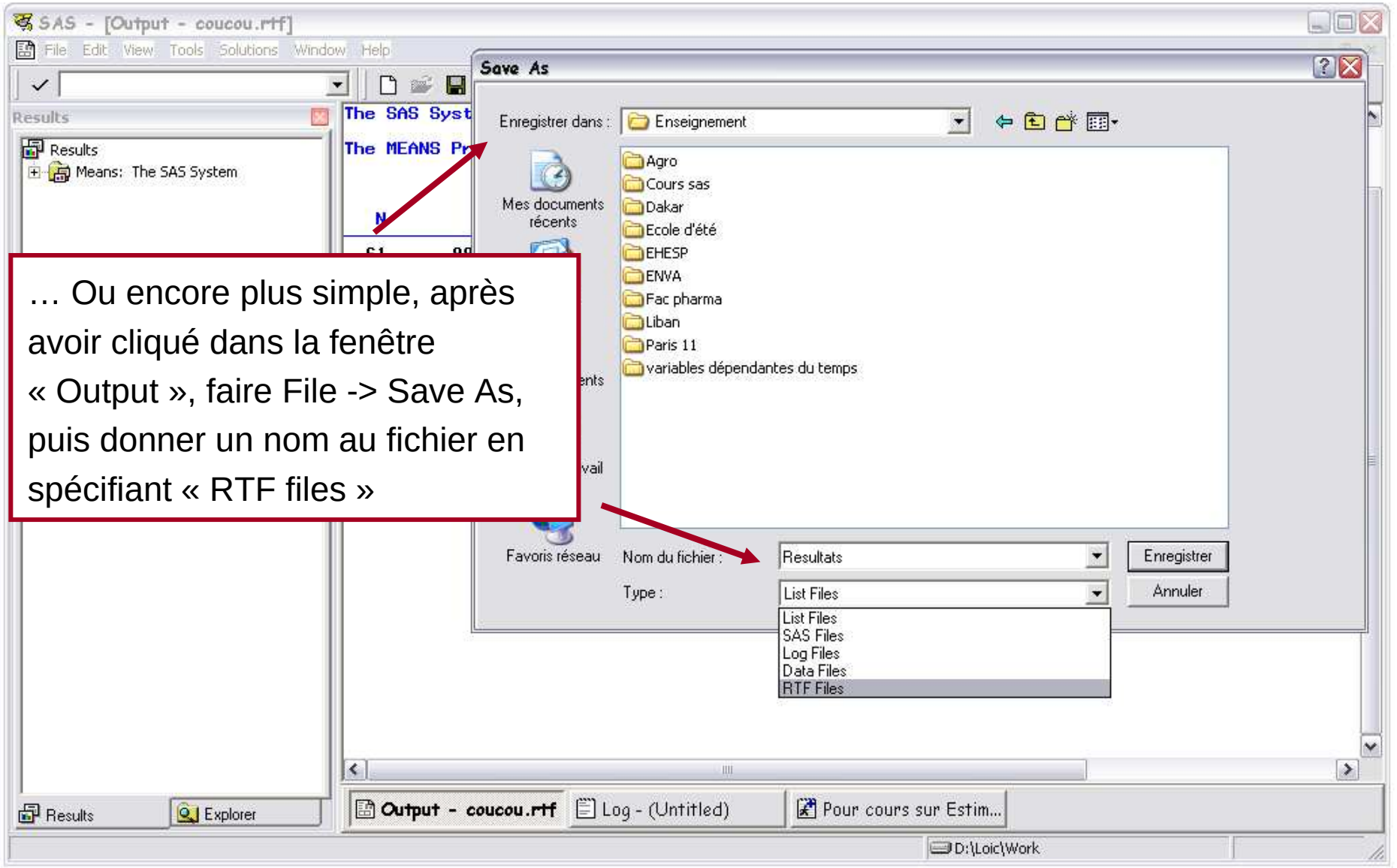
On peut réarranger la position des fenêtres grâce à Window -> Tile

Les fenêtres de SAS



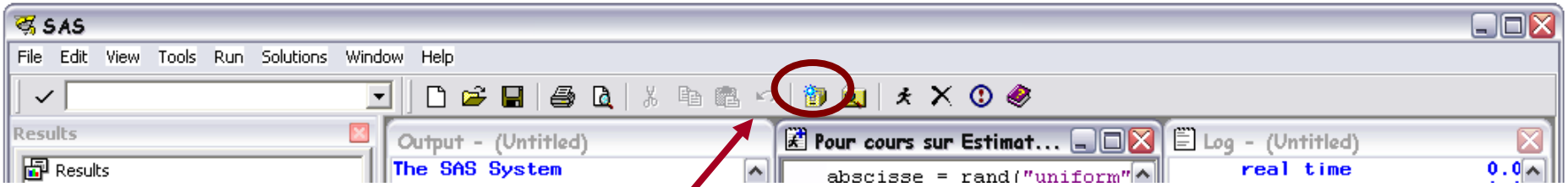
On peut récupérer tous les résultats de la fenêtre « Output » en les sélectionnant (Select all), les copiant (Copy), et les collant dans un document texte (Word, …)

Les fenêtres de SAS



La barre d'icônes

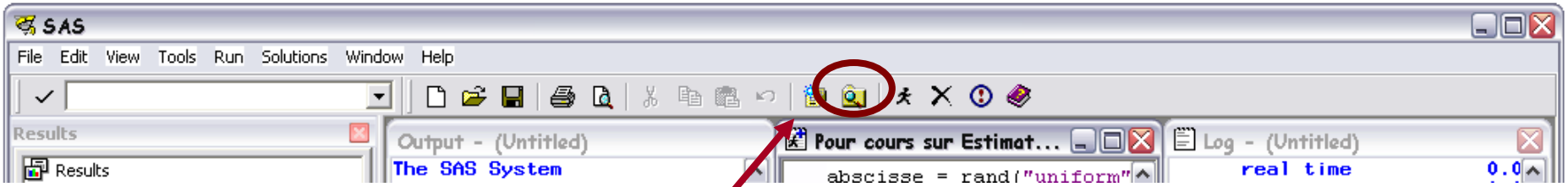
Quelle que soit la fenêtre sélectionnée



Création d'une « bibliothèque » (library)

La barre d'icônes

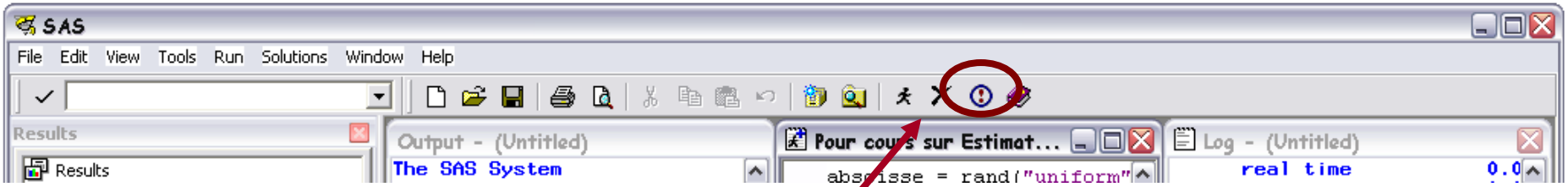
Quelle que soit la fenêtre sélectionnée



Ouvre une fenêtre de type « explorateur Windows »

La barre d'icônes

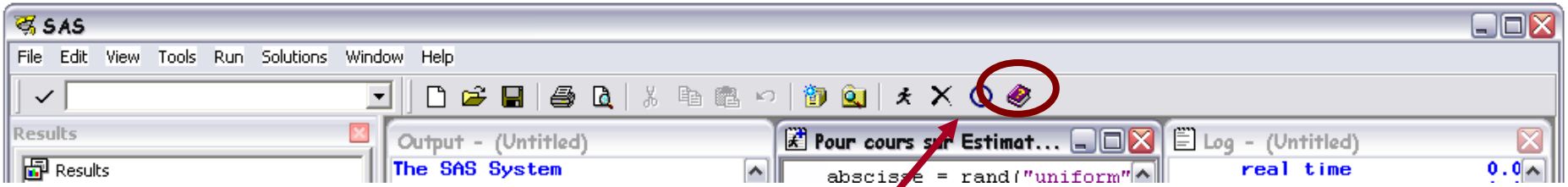
Quelle que soit la fenêtre sélectionnée



Interrompt une exécution en cours (par exemple, si une boucle n'en finit pas de « boucler »)

La barre d'icônes

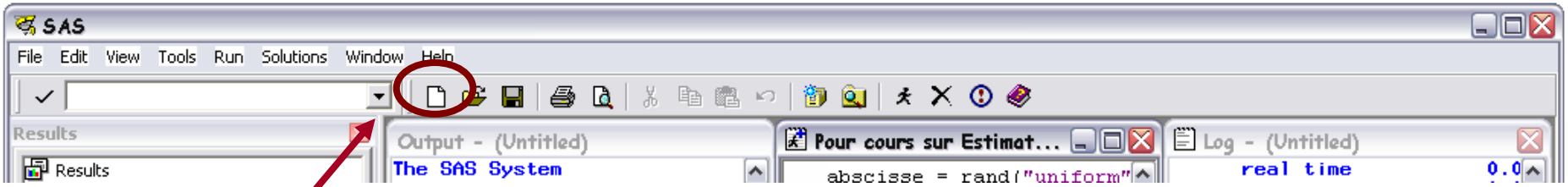
Quelle que soit la fenêtre sélectionnée



Aide de SAS

La barre d'icônes

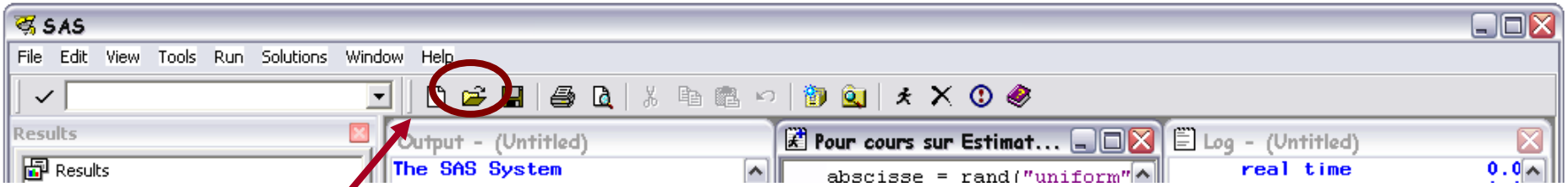
Fenêtre « Editor » sélectionnée



Nouvelle fenêtre « Editor »

La barre d'icônes

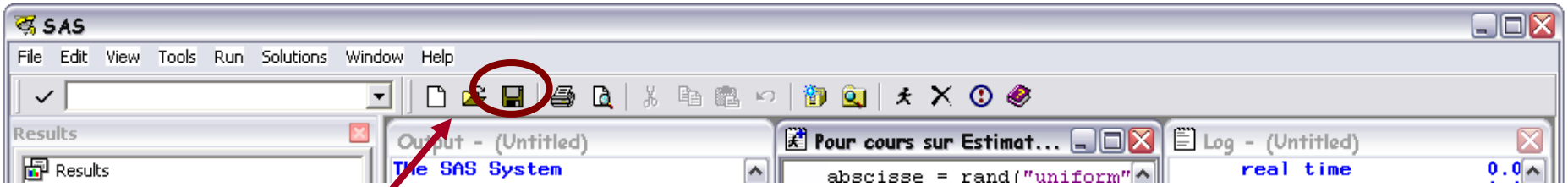
Fenêtre « Editor » sélectionnée



Ouverture d'une fenêtre « Editor »,c'est-à-dire, ouverture d'un programme SAS

La barre d'icônes

Fenêtre « Editor » sélectionnée



Enregistre le programme SAS en cours (fenêtre « Editor » sélectionnée, dans le cas où il y a plusieurs fenêtres « Editor » ouvertes)

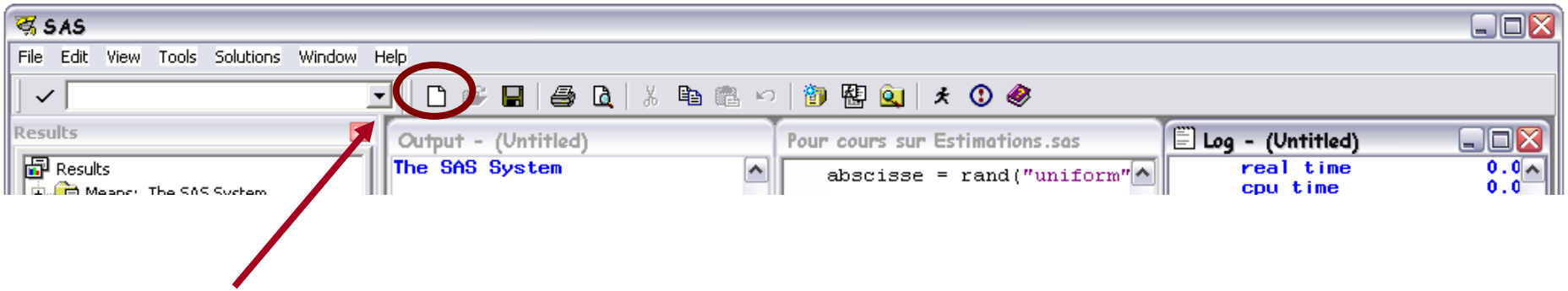
La barre d'icônes

Fenêtre « Editor » sélectionnée

Exécute tout le contenu de la fenêtre « Editor » (non recommandé) ou bien seulement les lignes de programmes sélectionnées avec la souris (recommandé)

La barre d'icônes

Fenêtre « Log » sélectionnée

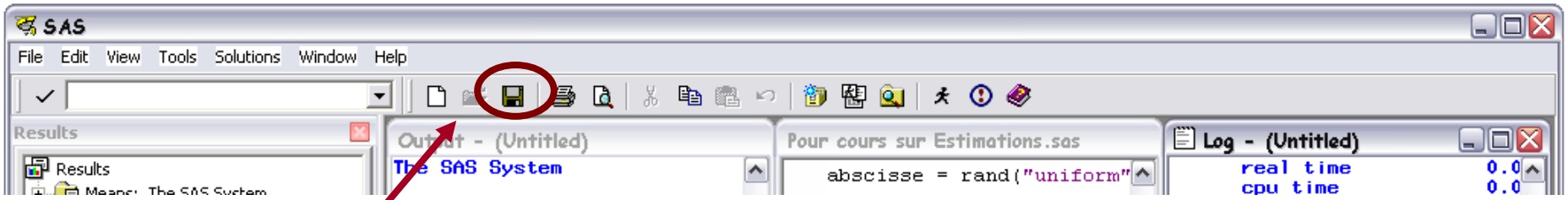


Efface le contenu de la fenêtre « Log » (pour faire du nettoyage, quand il commence à y avoir trop de notes et d'erreurs !)

Cette commande possède un raccourci clavier : « Ctrl + E »

La barre d'icônes

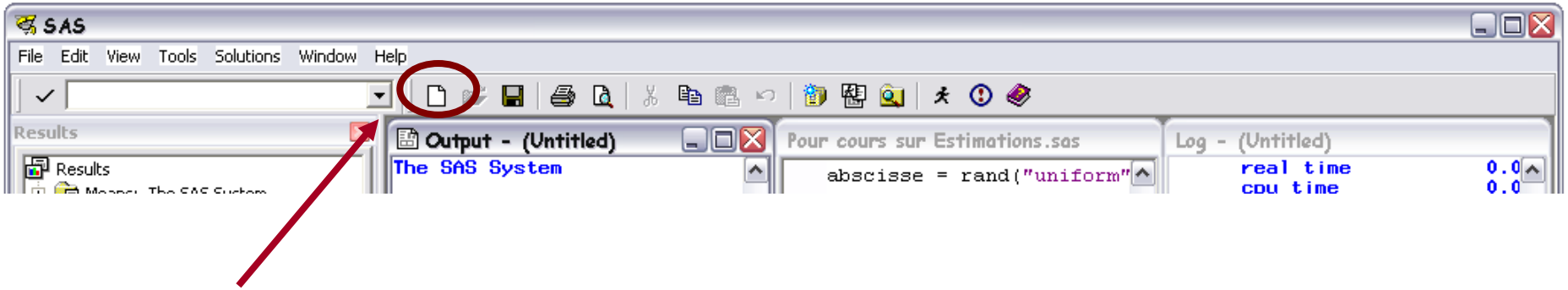
Fenêtre « Log » sélectionnée



Enregistre le contenu de la fenêtre « Log »

La barre d'icônes

Fenêtre « Output » sélectionnée

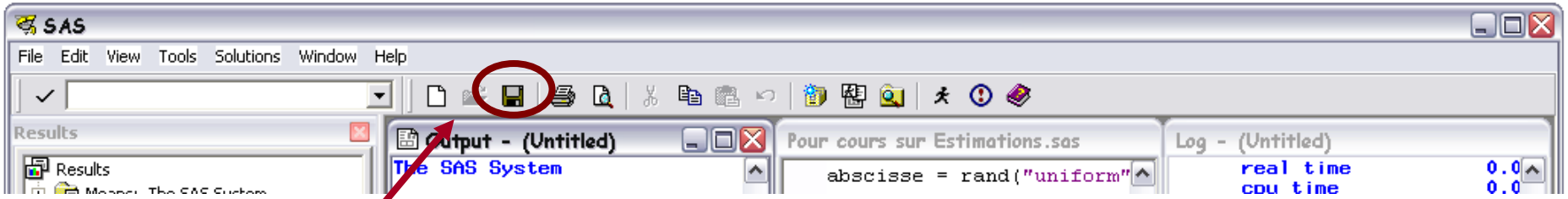


Efface le contenu de la fenêtre « Output » (pour faire du nettoyage, quand il commence à y avoir trop de résultats !)

Cette commande possède un raccourci clavier : « Ctrl + E »

La barre d'icônes

Fenêtre « Output » sélectionnée

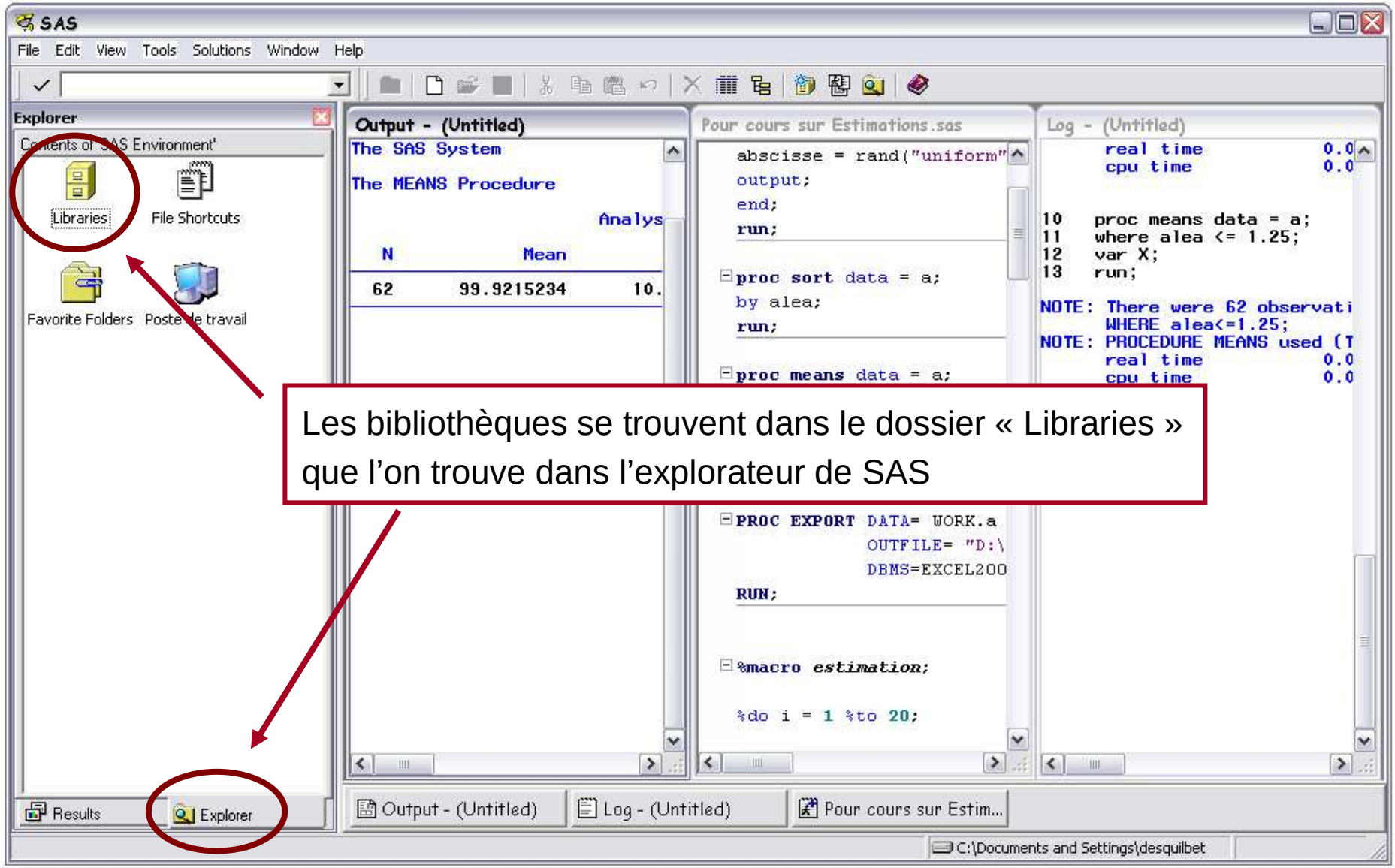


Enregistre le contenu de la fenêtre « Output »

Les bibliothèques (libraries)

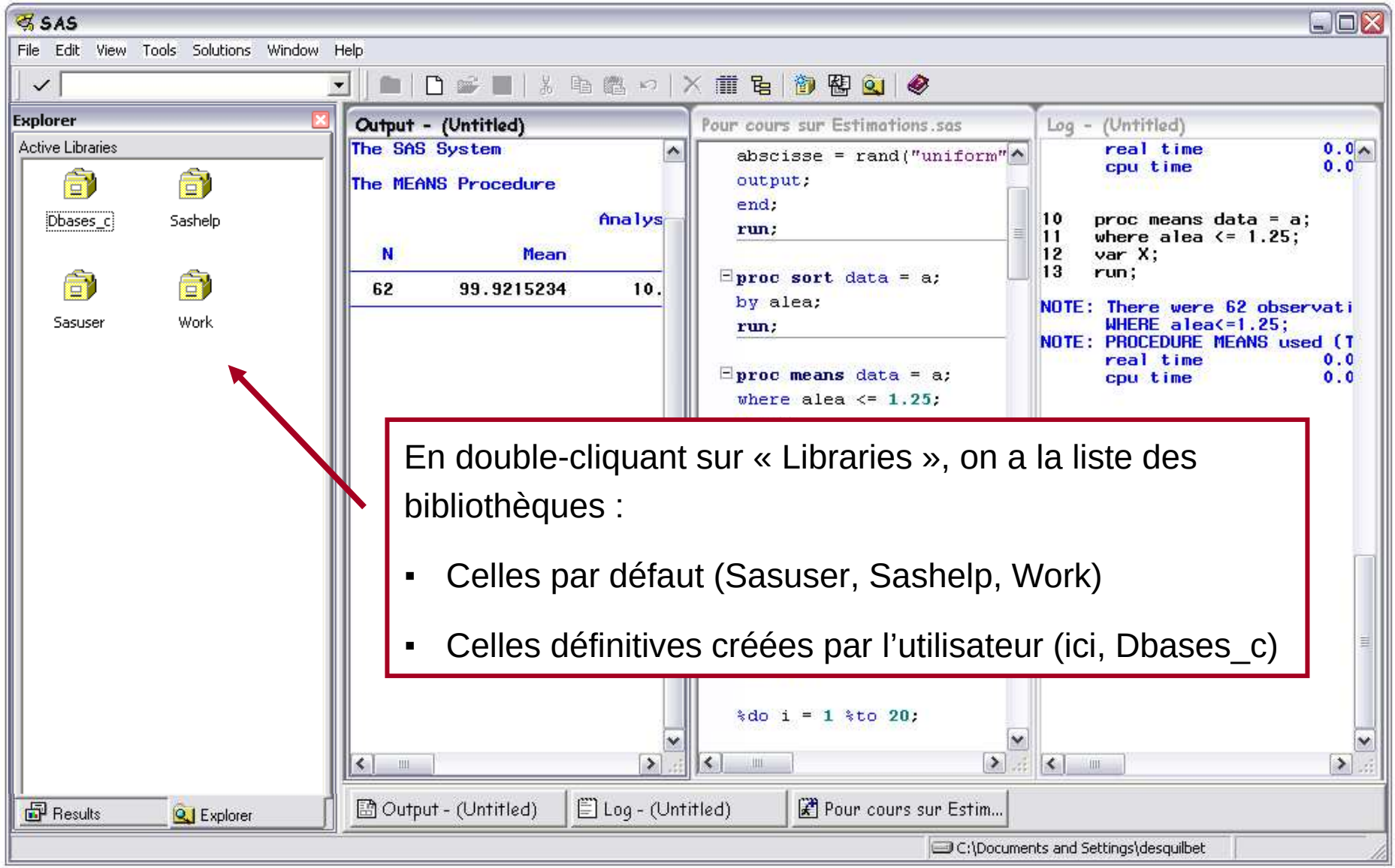
- Une bibliothèque fait référence à un dossier qui contient des fichiers de données SAS
- Un fichier de données SAS est un fichier repéré par l'extension « sas7bdat »
- Une bibliothèque est concrètement un « raccourci » (jargon Windows) ou un « alias » (jargon Macintosh), pointant vers le dossier contenant les fichiers de données
- Si une bibliothèque est supprimée, les fichiers de données ne sont pas supprimés du disque dur

Les bibliothèques (libraries)



Les bibliothèques se trouvent dans le dossier « Libraries » que l'on trouve dans l'explorateur de SAS

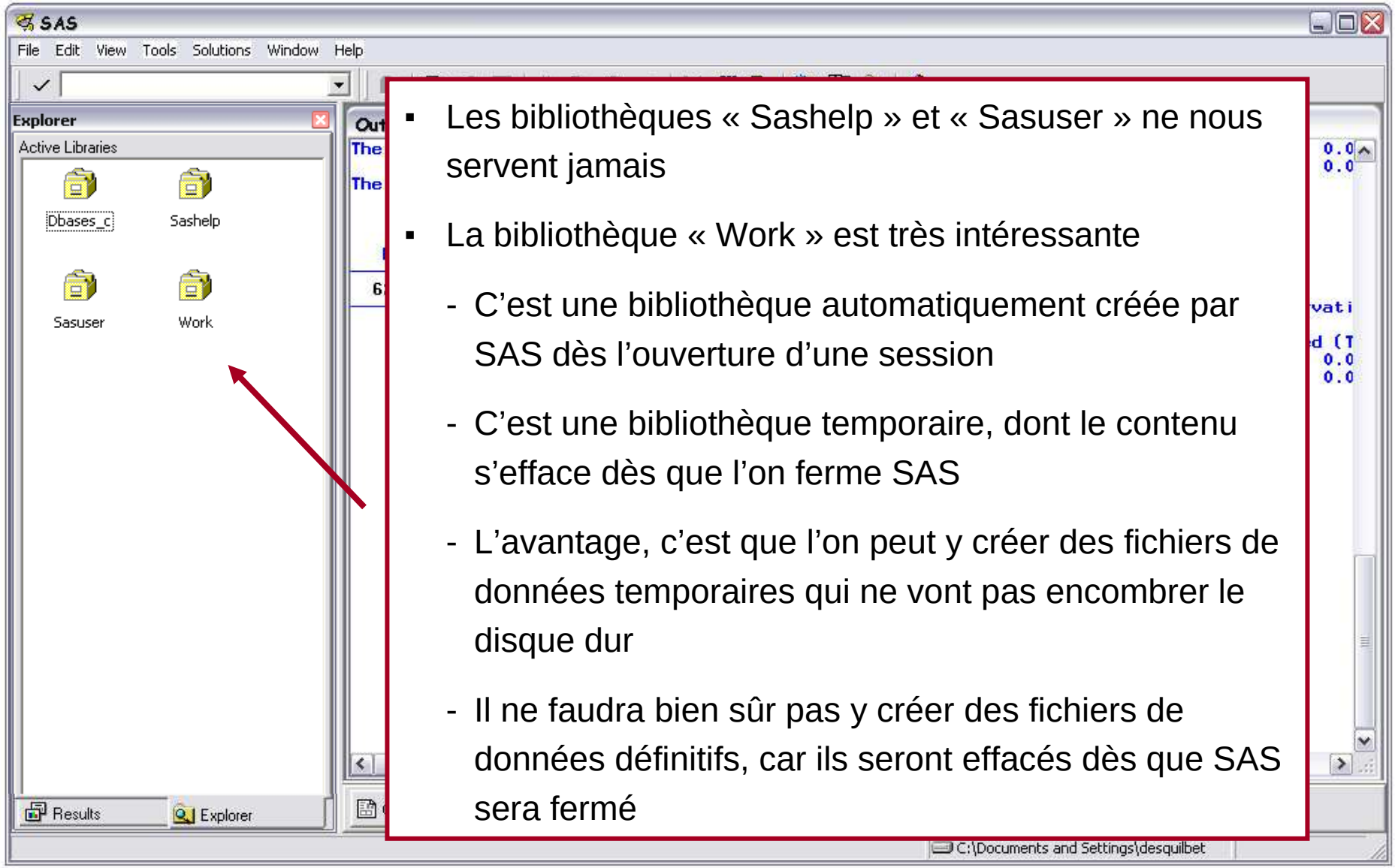
Les bibliothèques (libraries)



En double-cliquant sur « Libraries », on a la liste des bibliothèques :

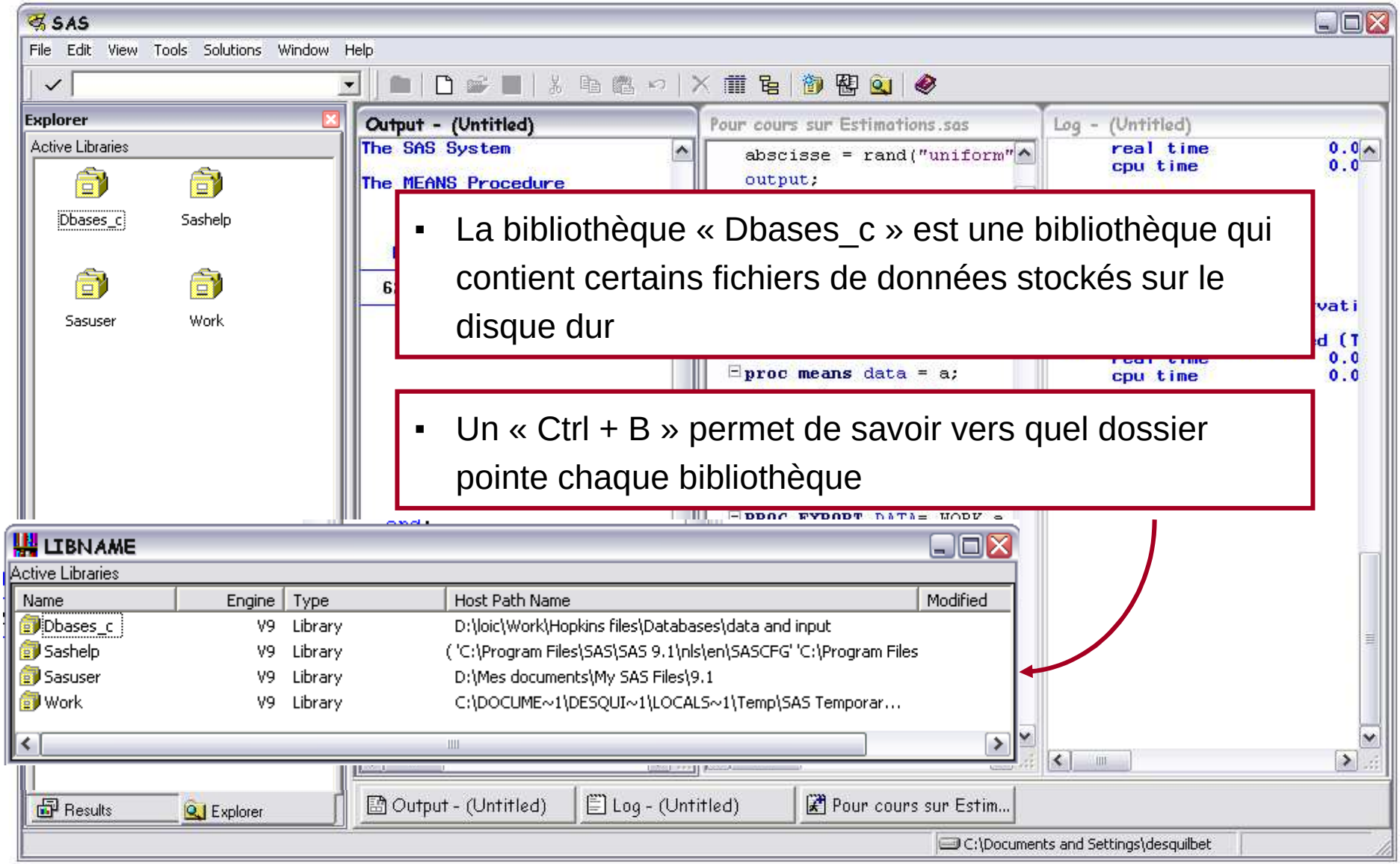
- Celles par défaut (Sasuser, Sashelp, Work)
- Celles définitives créées par l'utilisateur (ici, Dbases_c)

Les bibliothèques (libraries)



- Les bibliothèques « Sashelp » et « Sasuser » ne nous servent jamais
- La bibliothèque « Work » est très intéressante
  - C'est une bibliothèque automatiquement créée par SAS dès l'ouverture d'une session
  - C'est une bibliothèque temporaire, dont le contenu s'efface dès que l'on ferme SAS
  - L'avantage, c'est que l'on peut y créer des fichiers de données temporaires qui ne vont pas encombrer le disque dur
  - Il ne faudra bien sûr pas y créer des fichiers de données définitifs, car ils seront effacés dès que SAS sera fermé

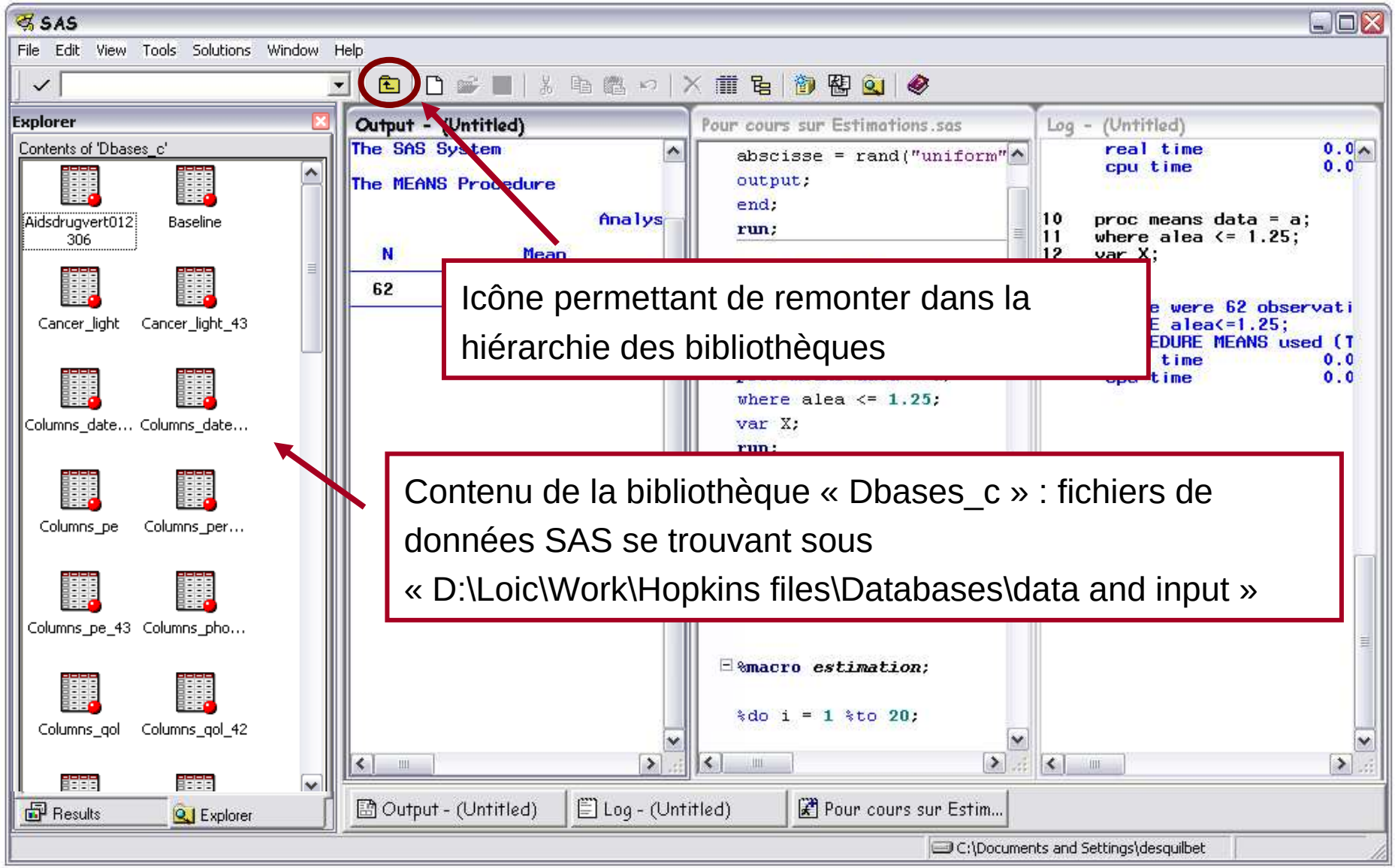
Les bibliothèques (libraries)



- La bibliothèque « Dbases_c » est une bibliothèque qui contient certains fichiers de données stockés sur le disque dur

- Un « Ctrl + B » permet de savoir vers quel dossier pointe chaque bibliothèque

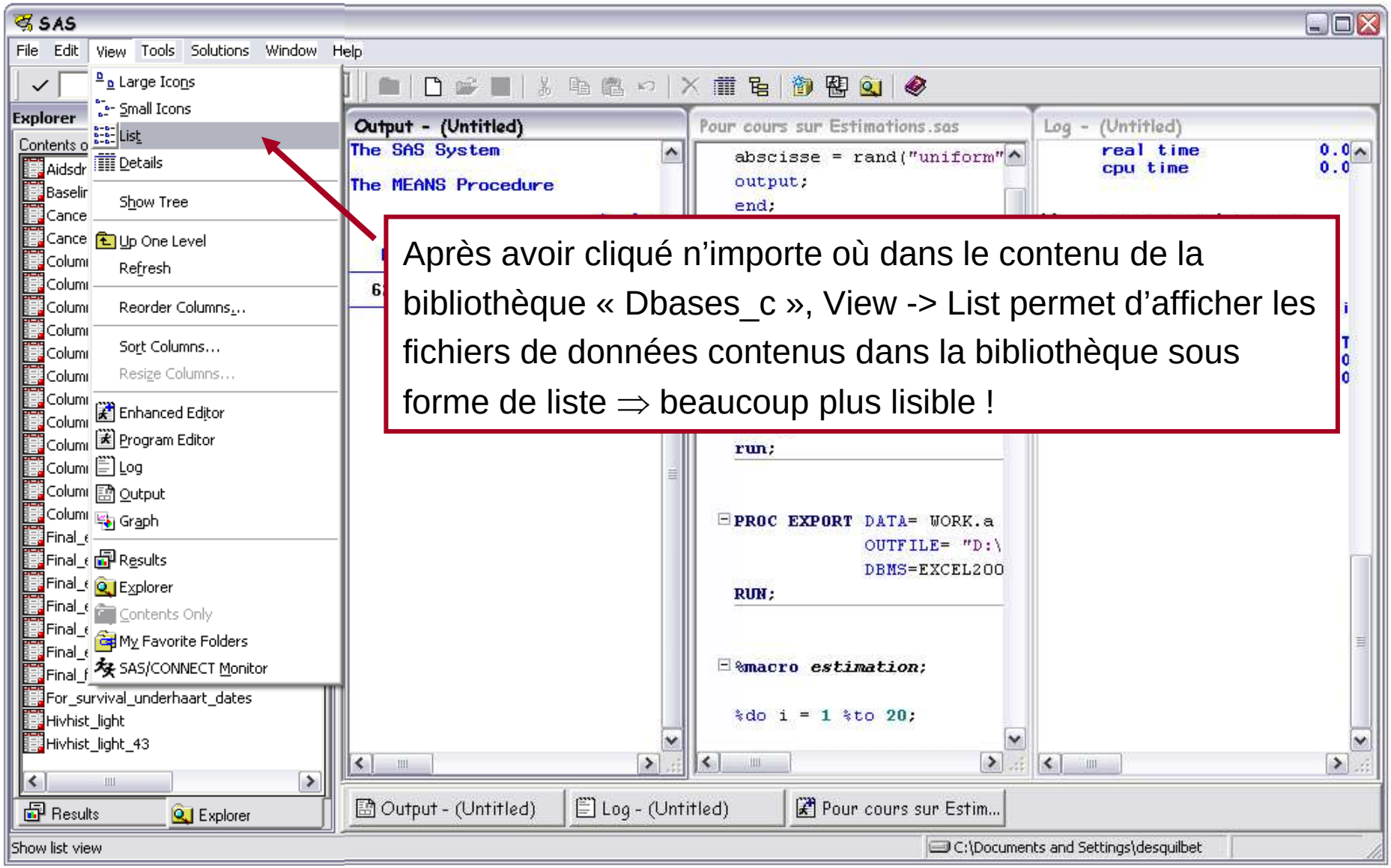
Les bibliothèques (libraries)



Icône permettant de remonter dans la hiérarchie des bibliothèques

Contenu de la bibliothèque « Dbases_c » : fichiers de données SAS se trouvant sous « D:\Loic\Work\Hopkins files\Databases\data and input »

Les bibliothèques (libraries)



Après avoir cliqué n'importe où dans le contenu de la bibliothèque « Dbases_c », View -> List permet d'afficher les fichiers de données contenus dans la bibliothèque sous forme de liste ⇒ beaucoup plus lisible !

Les bibliothèques (libraries)

Pourquoi devoir créer une bibliothèque ?

- Pour pouvoir travailler sur des fichiers de données (création ou manipulation de fichiers de données, ou analyses statistiques), il faut dire à SAS où ils se trouvent !

- La bibliothèque « Work », seule bibliothèque présente à l'ouverture de SAS, ne contient initialement aucun fichier de données

- Il faut donc créer ≥ 1 bibliothèque(s) qui pointera / pointeront vers les fichiers de données initiaux (fichiers de données SAS, extension « sas7bdat »)

Les bibliothèques (libraries)

Créer une bibliothèque temporaire ou définitive ?

- Bibliothèque temporaire : bibliothèque qui disparaitra après fermeture de SAS (sans que les fichiers de données vers lesquels elle pointe soient effacés)

- Bibliothèque définitive : bibliothèque qui sera toujours présente parmi la liste de bibliothèques, même après fermeture puis ouverture d'une session SAS

Les bibliothèques (libraries)

Avantage / inconvénient de créer des bibliothèques définitives

- Avantage

  Evite de créer systématiquement la bibliothèque sur laquelle on sait que l'on travaillera tout le temps

- Inconvénient

  - Dès que cette bibliothèque ne sert plus, elle devient « encombrante »

  - « Solution » : il est très simple de supprimer une bibliothèque de l'explorateur (clic droit sur la bibliothèque à supprimer -> Delete)

Les bibliothèques (libraries)

Avantage / inconvénient de créer des bibliothèques temporaires

- Avantage

  Evite d'encombrer l'explorateur de bibliothèques qui ne servent plus

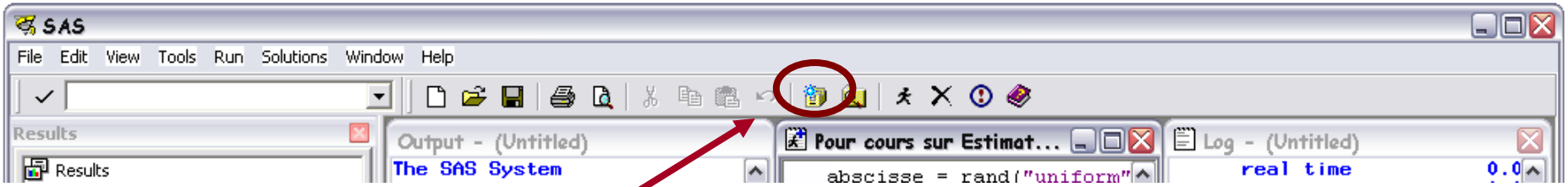
- Inconvénient

  - Il faut créer la bibliothèque à chaque ouverture de session SAS

  - « Solution » : il est très simple de créer une bibliothèque (dans un programme) !

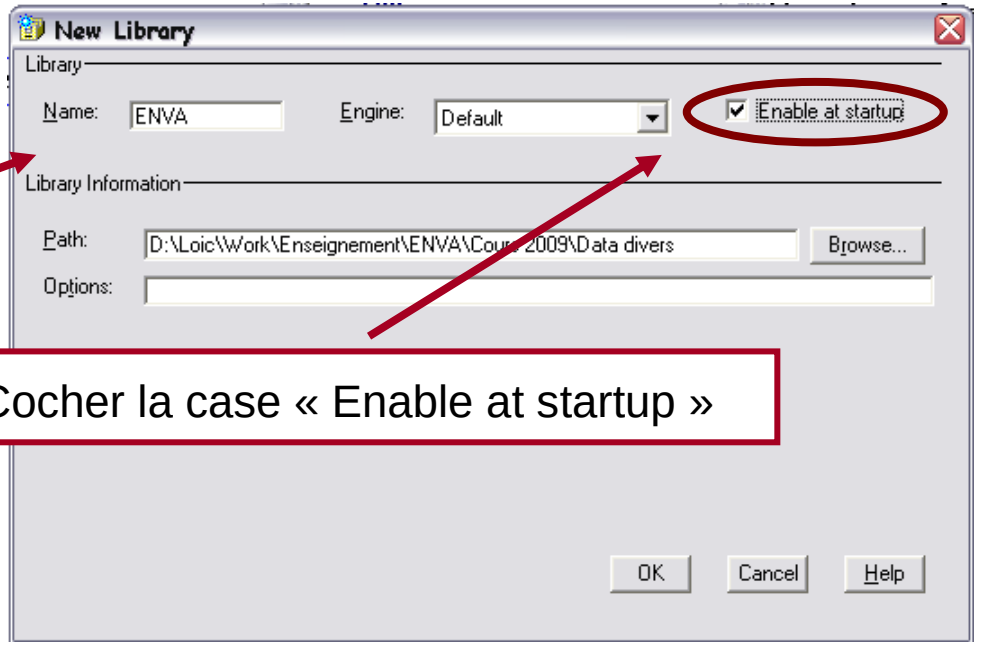
Les bibliothèques (libraries)

Créer une bibliothèque définitive



1) Clic sur « New Library »

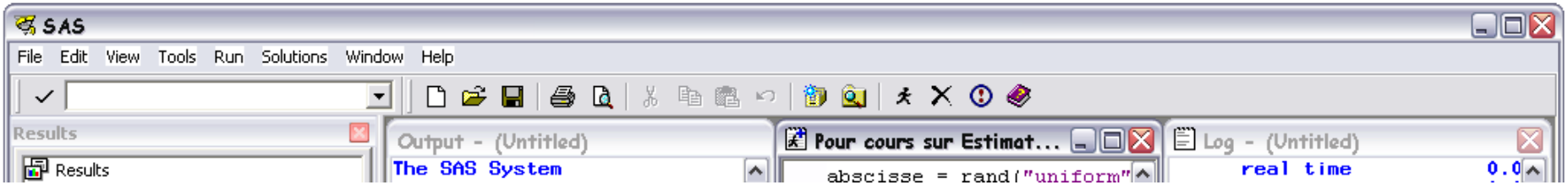
2) Remplir les différents champs de la boite de dialogue

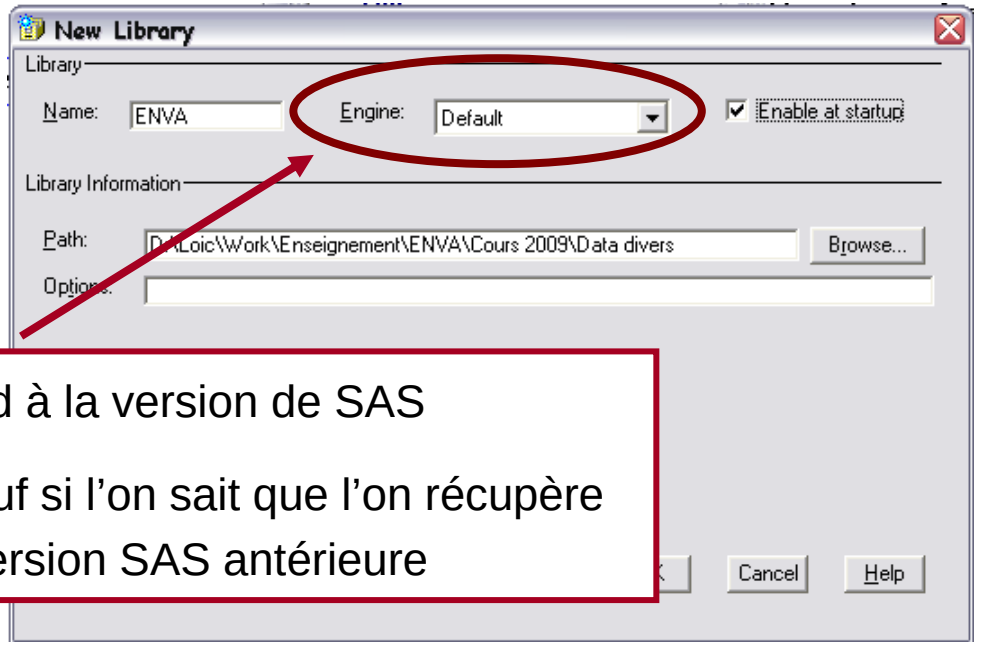


3) Cocher la case « Enable at startup »

Les bibliothèques (libraries)

Créer une bibliothèque définitive

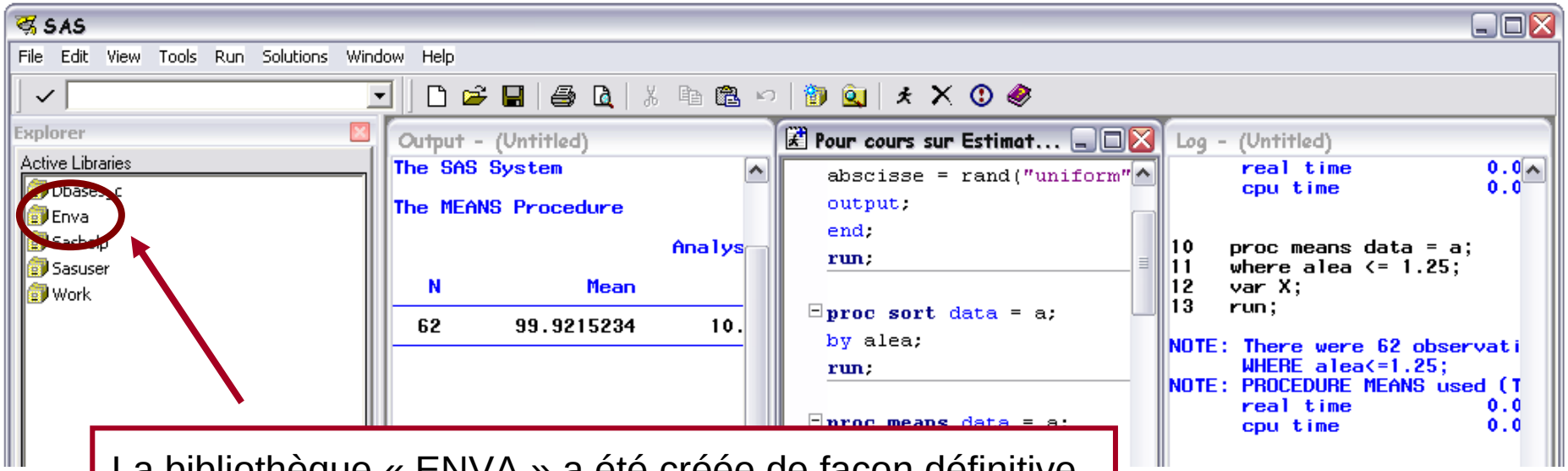




- Le champ « Engine » correspond à la version de SAS
- Il faut le laisser à « Default », sauf si l'on sait que l'on récupère des fichiers de données d'une version SAS antérieure

Les bibliothèques (libraries)

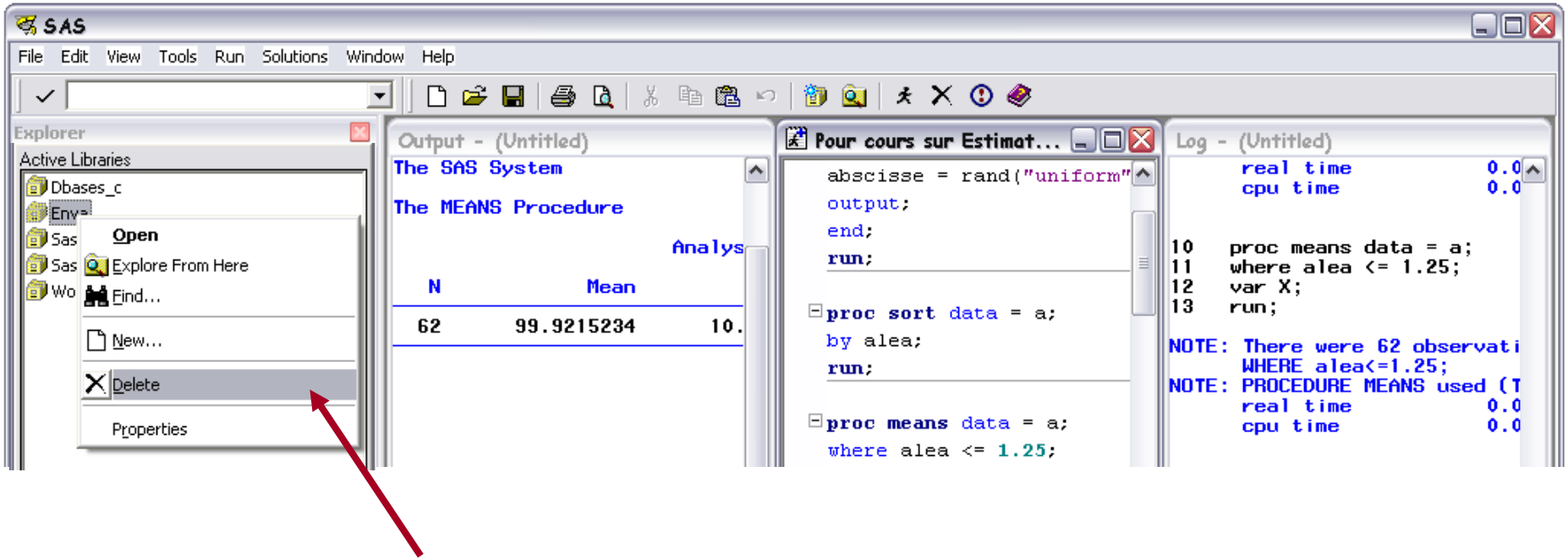
Créer une bibliothèque définitive



La bibliothèque « ENVA » a été créée de façon définitive

Les bibliothèques (libraries)

Créer une bibliothèque définitive



Après un clic-droit sur « ENVA », un clic sur « Delete » supprimera de façon définitive la bibliothèque

Les bibliothèques (libraries)

Créer une bibliothèque temporaire

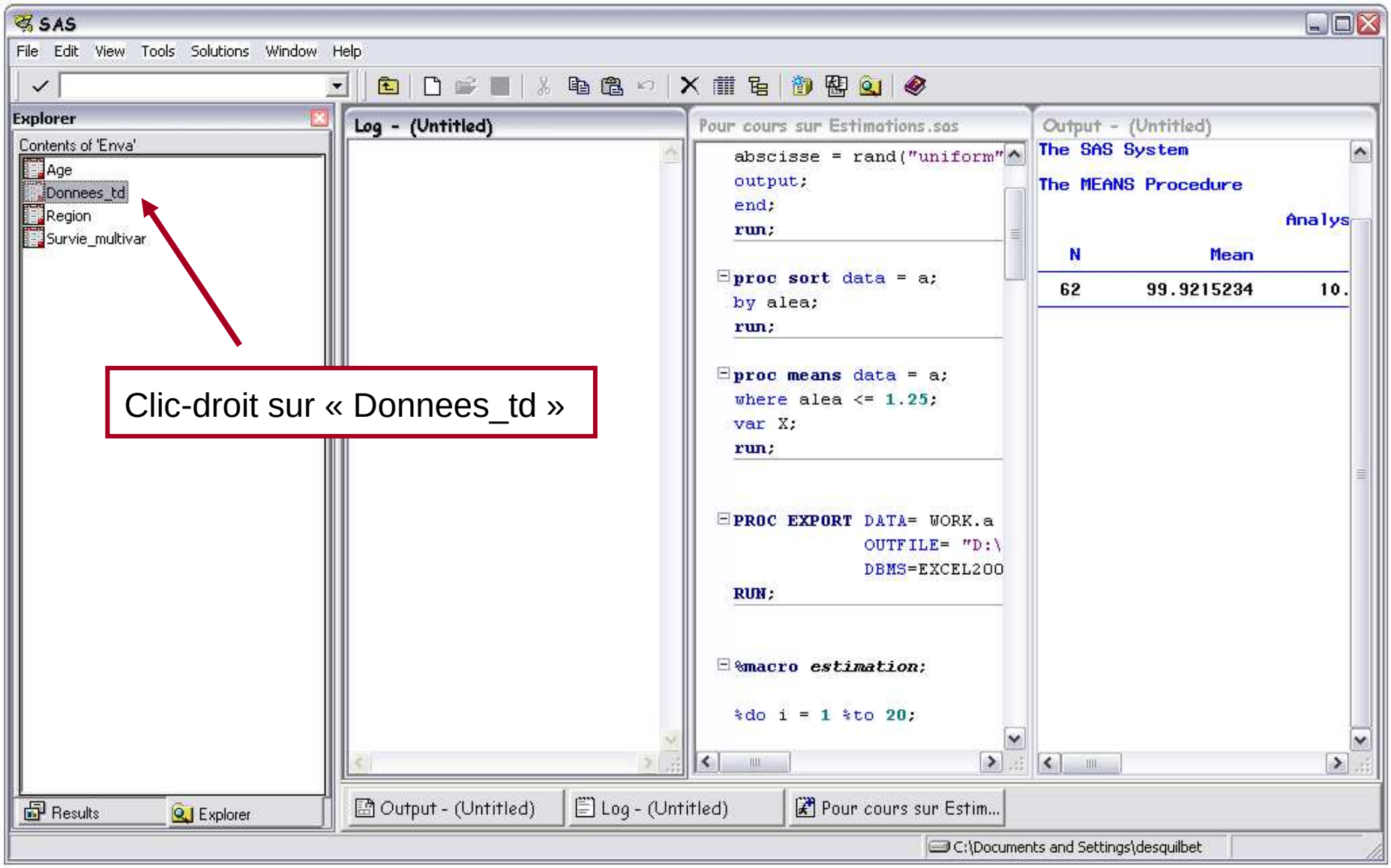
- Pour créer une bibliothèque temporaire, il faut taper la ligne de programme dans l'éditeur de programme (fenêtre « Editor ») :

  libname *nom_bibliothèque* **"***chemin_bibliothèque***" ;**

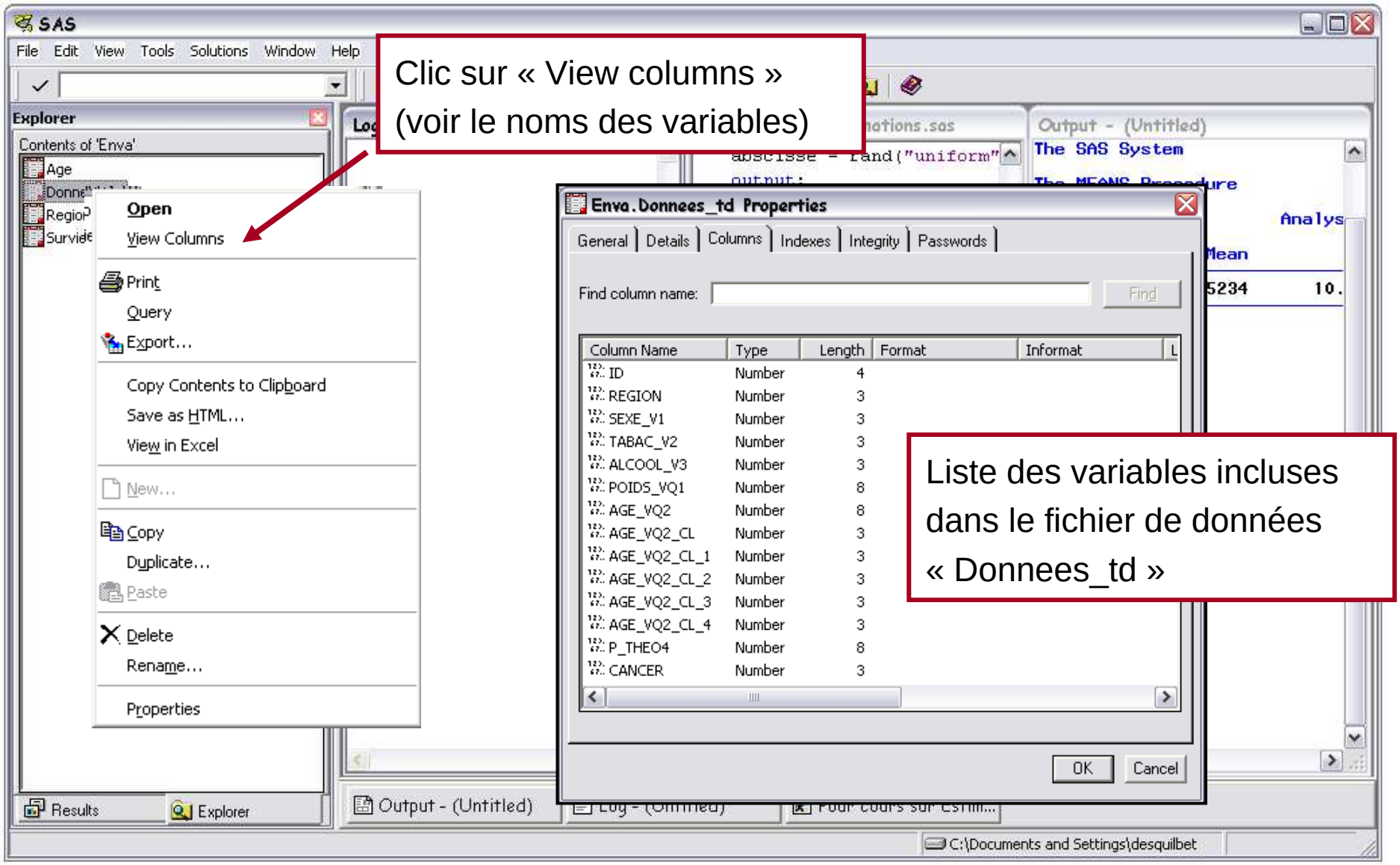
  - Le nom de la bibliothèque ne doit pas dépasser 8 caractères
  - Le chemin de la bibliothèque peut être copié de l'explorateur Windows, et être collé entre les guillemets

  - La ligne de programme n'est à exécuter qu'une seule fois par session SAS

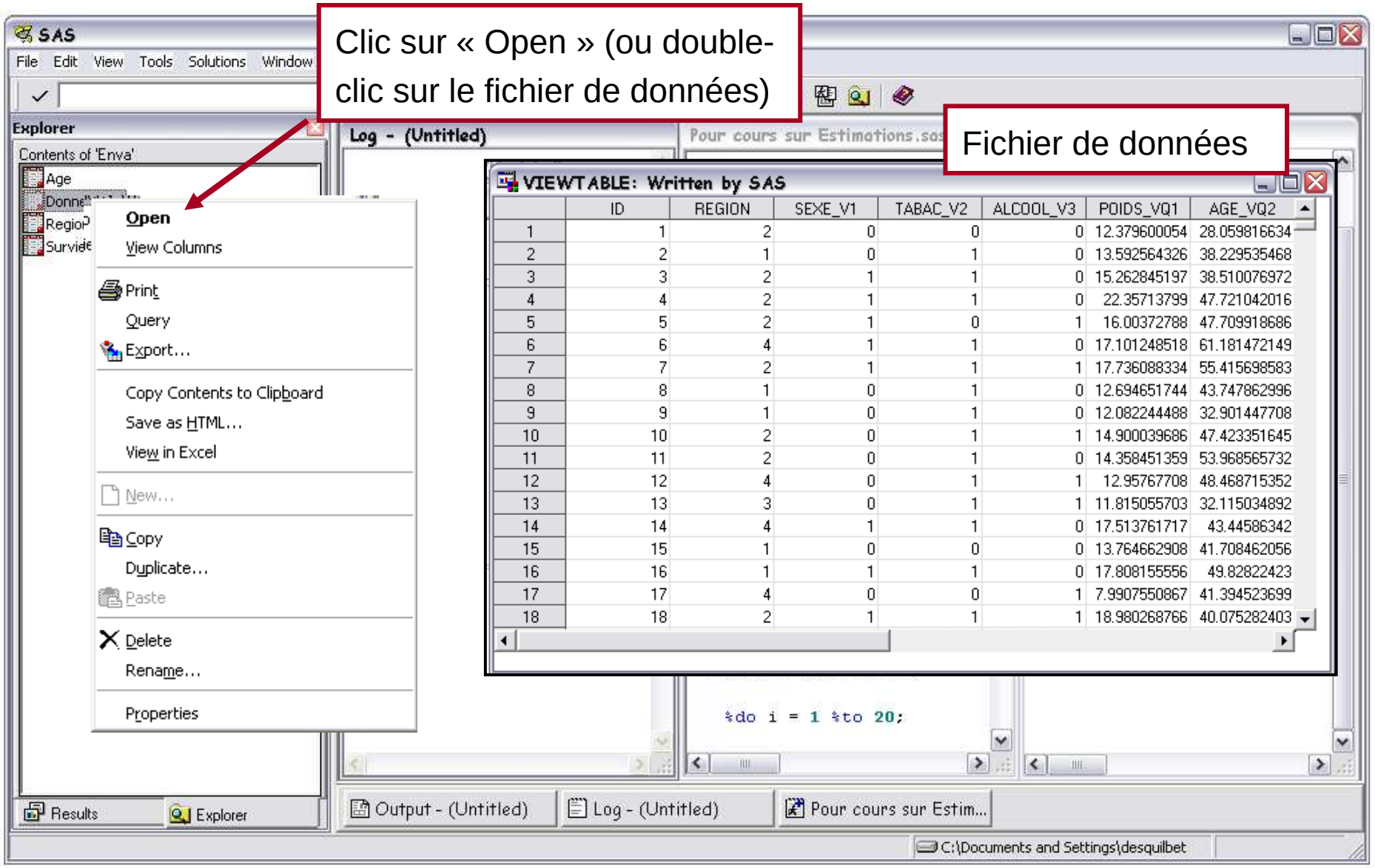
Décrire (très) rapidement un fichier de données



Décrire (très) rapidement un fichier de données



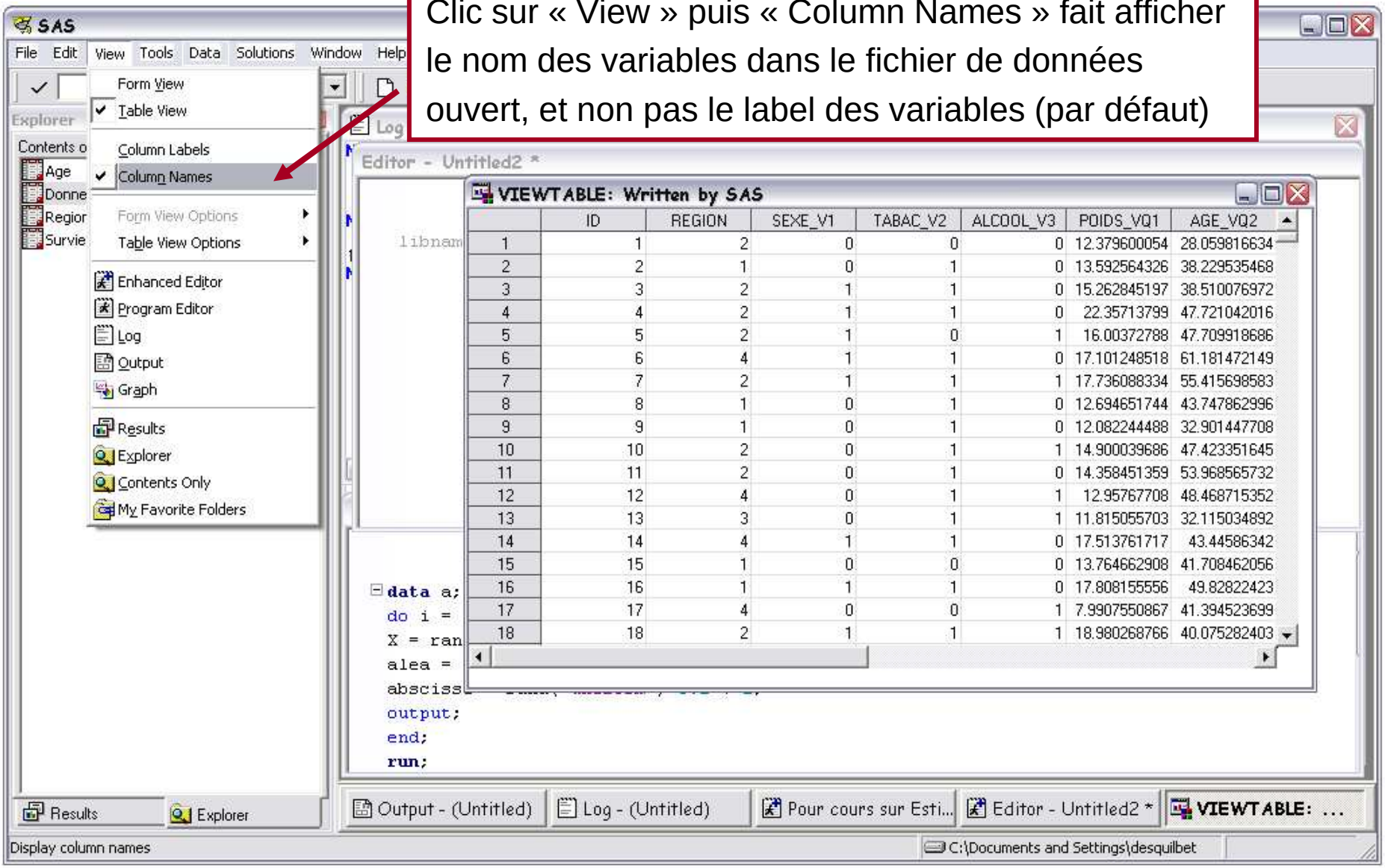
Décrire (très) rapidement un fichier de données



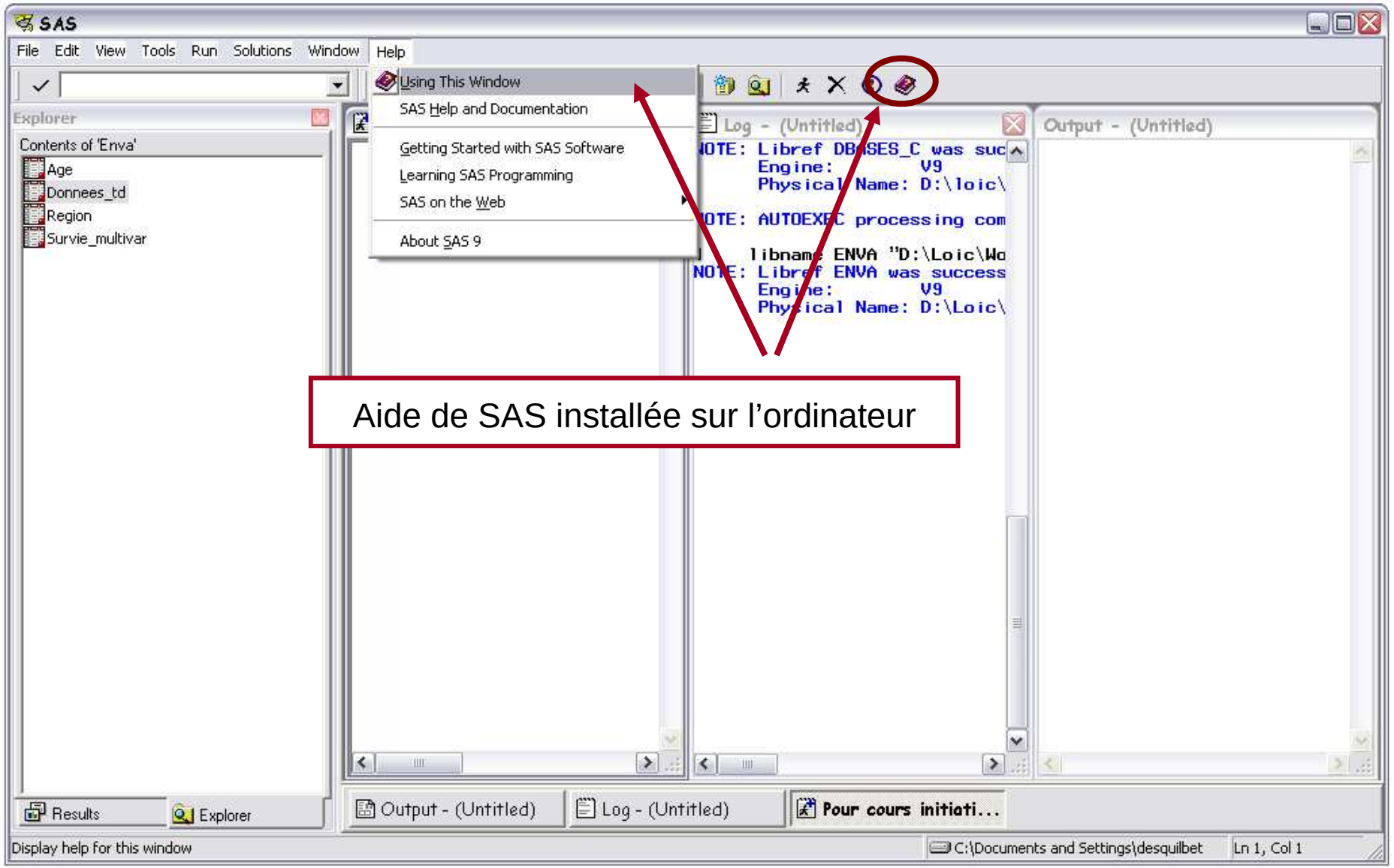
	ID	REGION	SEXE_V1	TABAC_V2	ALCOOL_V3	POIDS_VQ1	AGE_VQ2
1	1	2	0	0	0	12.379600054	28.059816634
2	2	1	0	1	0	13.592564326	38.229535468
3	3	2	1	1	0	15.262845197	38.510076972
4	4	2	1	1	0	22.35713799	47.721042016
5	5	2	1	0	1	16.00372788	47.709918686
6	6	4	1	1	0	17.101248518	61.181472149
7	7	2	1	1	1	17.736088334	55.415698583
8	8	1	0	1	0	12.694651744	43.747862996
9	9	1	0	1	0	12.082244488	32.901447708
10	10	2	0	1	1	14.900039686	47.423351645
11	11	2	0	1	0	14.358451359	53.968565732
12	12	4	0	1	1	12.95767708	48.468715352
13	13	3	0	1	1	11.815055703	32.115034892
14	14	4	1	1	0	17.513761717	43.44586342
15	15	1	0	0	0	13.764662908	41.708462056
16	16	1	1	1	0	17.808155556	49.82822423
17	17	4	0	0	1	7.9907550867	41.394523699
18	18	2	1	1	1	18.980268766	40.075282403

Décrire (très) rapidement un fichier de données

Clic sur « View » puis « Column Names » fait afficher le nom des variables dans le fichier de données ouvert, et non pas le label des variables (par défaut)

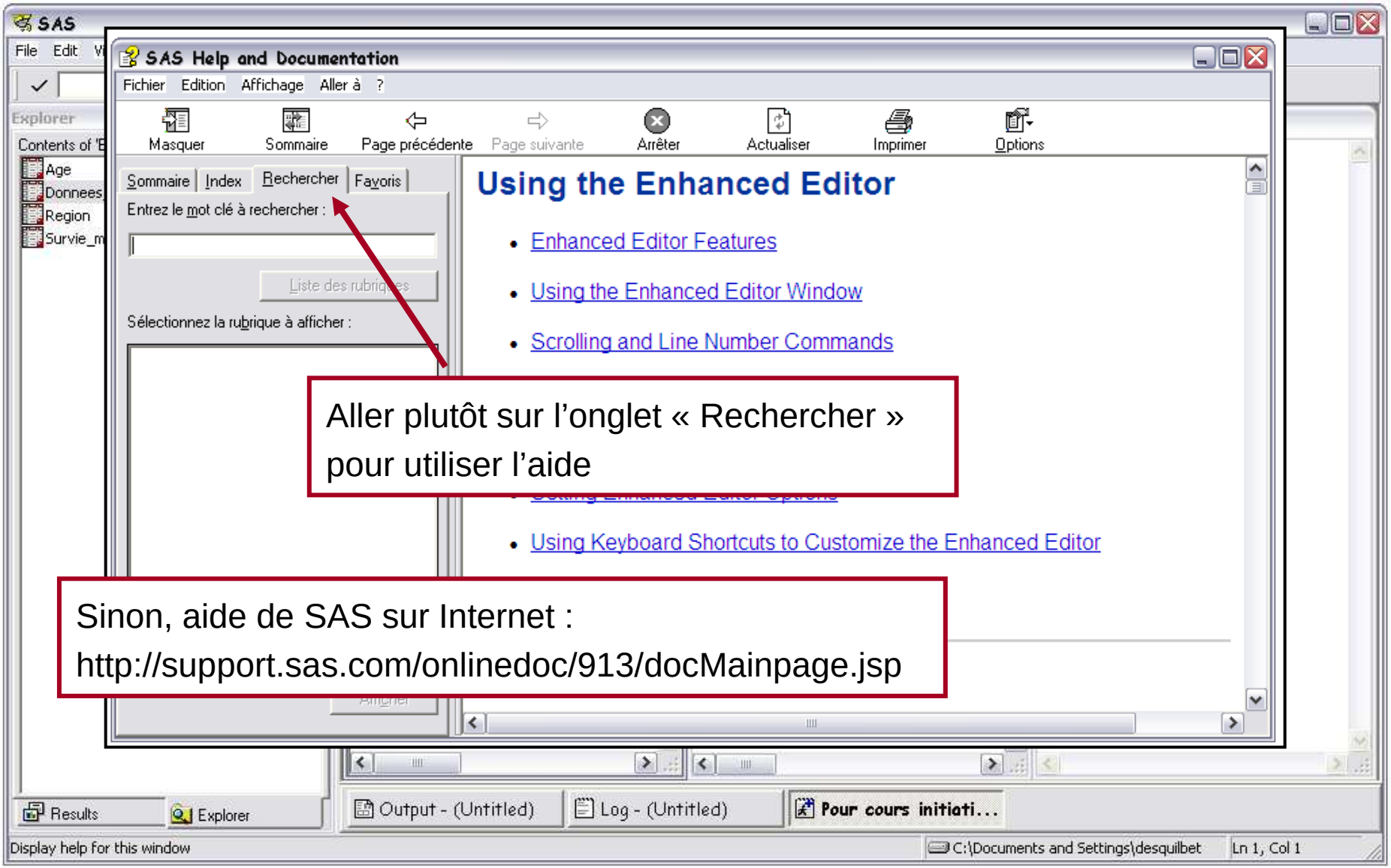


Aide de SAS



Aide de SAS installée sur l'ordinateur

Aide de SAS


SAS
File Edit
Explorer
Contents of 'E
Age
Donnees
Region
Survie_m
SAS Help and Documentation
Fichier Edition Affichage Aller à ?
Masquer
Sommaire
Page précédente
Page suivante
Arrêter
Actualiser
Imprimer
Options
Sommaire Index Rechercher Favoris
Entrez le mot clé à rechercher :
Liste des rubriques
Sélectionnez la rubrique à afficher :
Using the Enhanced Editor
Enhanced Editor Features
Using the Enhanced Editor Window
Scrolling and Line Number Commands
Using Keyboard Shortcuts to Customize the Enhanced Editor
Aller plutôt sur l'onglet « Rechercher » pour utiliser l'aide
Sinon, aide de SAS sur Internet :
http://support.sas.com/onlinedoc/913/docMainpage.jsp
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Pour cours initiati...
Display help for this window
C:\Documents and Settings\desquilbet
Ln 1, Col 1

Les commandes SAS de base

Les programmes SAS

- Notations utilisées pour l'écriture de programmes SAS

  - Ecriture droite : ce qui doit être écrit en fonction des objectifs des analyses

  - *Ecriture en italique* : ce qui doit être remplacé en fonction des objectifs des analyses

  - **Ecriture en gras (rouge)** : commandes ou mots clés SAS (« RUN », « ; », …)

  - Ecriture normale (noire) : ce que doit définir l'utilisateur (souvent des noms de variables ou de fichiers de données)

  - [Ecriture entre crochets] : commandes optionnelles (les crochets ne feront jamais partie du programme, sauf à de très rares exceptions…)

Les programmes SAS

- Les programmes SAS sont écrits dans la fenêtre « Editor »

- Les programmes SAS sont composés d'étapes **DATA**, de **procédures**, d'options générales, de macro programmes, …

- Les étapes DATA manipulent les fichiers de données

  - Création (créer un fichier de données à partir d'un précédent fichier de données)

  - Modification (suppression d'observations, création de variables, …)

  - Notation par la suite : « table » fera référence à un fichier de données SAS

Les programmes SAS

- Des commentaires peuvent être insérés n'importe où dans le programme, et doivent être situés entre « /* » et « */ » :

  /* ceci est un commentaire */

- Introduction rapide sur les procédures SAS

  - Les procédures utilisent ou exploitent (statistiquement) les tables

  - Il est fortement recommandé de spécifier le nom de la table sur laquelle on veut travailler (par défaut, la procédure travaillera sur la dernière table créée $\Rightarrow$ risque d'erreurs d'interprétation élevé !)

  - Les procédures ne peuvent travailler que sur des tables SAS !

Structure générale des programmes SAS

Les programmes SAS sont une succession d'étape DATA et de procédure

Data *fichier2* **;**
Set *fichier1* **;**
[*instructions…* **;**]
Run ;

Etape DATA qui crée *fichier2* à partir de *fichier1*

PROC *NOM_PROCEDURE* **data =** *fichier2* **;**
instructions… **;**
Run ;

Procédure qui travaille sur *fichier2*

Data *fichier3* **;**
Set *fichier2* **;**
[*instructions…* **;**]
Run ;

Etape DATA qui crée *fichier3* à partir de *2*

PROC *NOM_PROCEDURE* **data =** *fichier3* **;**
instructions… **;**
Run ;

Procédure qui travaille sur *fichier3*

Etc…

Remarques

- **Chaque ligne de programme se termine par « ; »**
- **Chaque étape DATA ou procédure se termine par « Run ; »**

L'étape DATA

Créer une table *newtable* à partir d'un table existante *oldtable*

- 1er cas de figure : *oldtable* est situé physiquement sur le disque dur, à l'adresse « C:\donnees\etude1 », et on veut créer *newtable* sous « C:\donnees\etude1 »

  Soit « lib1 » la bibliothèque (temporaire) qui va pointer vers « C:\donnees\etude1 »

```
libname lib1 "C:\donnees\etude1" ;

Data lib1.newtable ;
Set lib1.oldtable ;
Run ;
```

L'étape DATA

Créer une table *newtable* à partir d'un table existante *oldtable*

- 2ème cas de figure : *oldtable* est situé physiquement sur le disque dur, à l'adresse « C:\donnees\etude1 », et on veut créer *newtable* sous « D:\etude_contaminants »

  Soit « lib2 » la bibliothèque (temporaire) qui va pointer vers « D:\etude_contaminants »

```
libname lib1 "C:\donnees\etude1" ;
libname lib2 "D:\etude_contaminants" ;
Data lib2.newtable ;
Set lib1.oldtable ;
Run ;
```

L’étape DATA

Créer une table *newtable* à partir d’un table existante *oldtable*

- 3ème cas de figure : *oldtable* est situé physiquement sur le disque dur, à l’adresse « C:\donnees\etude1 », et on veut créer *newtable* dans la bibliothèque temporaire de SAS (bibliothèque « Work »)

```
libname lib1 "C:\donnees\etude1" ;

Data newtable ;
Set lib1.oldtable ;
Run ;
```

L'étape DATA

Créer une table *newtable* à partir d'un table existante *oldtable*

- 4ème cas de figure : *oldtable* est situé dans la bibliothèque « Work », et on veut créer *newtable* dans la bibliothèque « Work »

```
Data newtable ;
Set oldtable ;
Run ;
```

- Remarque

```
Data lib1.oldtable ;
Set lib1.oldtable ;
[instructions… ;]
Run ;
```

Très dangereux !! On « écrase » la table *oldtable* située sur le disque dur par elle-même

⇒ A ne ***jamais*** faire

L’étape DATA

Créer des variables dans *newtable* à partir de *oldtable*

- Le nom des variables ne doit pas excéder 32 caractères
- Un nom de variable peut commencer par une lettre ou « _ »
- Dans le nom des variables, SAS ne fait pas de distinction entre les majuscules et les minuscules

  « REGION » et « region » seront considérées comme une même variable
- Les variables peuvent être numériques, ou alphanumériques (les variables numériques sont **très** fortement recommandées pour les analyses statistiques)
- Le « . » symbolise la donnée manquante pour les variables numériques

  La valeur d’une donnée manquante est -∞

L’étape DATA

Créer des variables dans *newtable* à partir de *oldtable*

- Supposons le cas de figure suivant
  - *oldtable* est situé dans la bibliothèque « Work »
  - On veut créer 7 variables *var3*, *var4*, *…*, et *var9*, à partir des variables existantes *var1* et *var2* contenues dans *oldtable*
  - Toutes les variables sont numériques
  - Soit *newtable* la table qui va contenir les variables *var1* à *var9*, que l’on va créer dans la bibliothèque « Work »

L'étape DATA

Créer des variables dans *newtable* à partir de *oldtable*

- Exemple de programmation avec ce cas de figure

```
Data newtable ;
Set oldtable ;
var3 = var1 + 3*var2 ;
var4 = log10(var1) – 4*var2 ;
var5 = (-2)*var3 + var2**2 ;
If var4 > 8 Then var6 = 1 ; Else var6 = 0 ;
If var4 ne . And var1 > 0 Then var7 = 1 / var1 ;
   If var5 >= 0 Then Do ;
   var8 = sqrt(var5) ;
   var9 = log(var5 +1) ;
   End ;
Run ;
```

Opérateurs classiques : +, -, *, /

Fonctions mathématiques

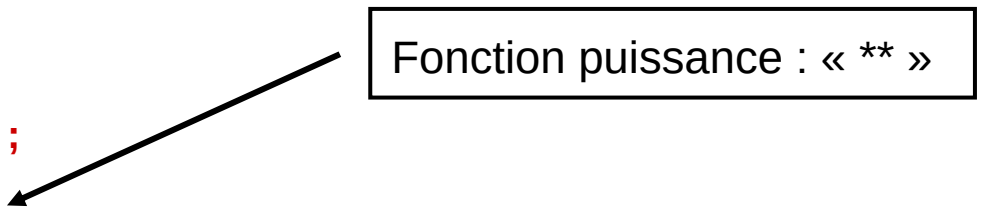
- log() : logarithme népérien
- log10() : logarithme décimal
- sqrt() : racine carrée
- abs() : valeur absolue
- exp() : exponentielle
- int() : partie entière
- round*(var, 0.01)* : arrondi *au centième*

L'étape DATA

Créer des variables dans *newtable* à partir de *oldtable*

- Exemple de programmation avec ce cas de figure

```
Data newtable ;
Set oldtable ;
var3 = var1 + 3*var2 ;
var4 = log10(var1) – 4*var2 ;
var5 = (-2)*var3 + var2**2 ;
If var4 > 8 Then var6 = 1 ; Else var6 = 0 ;
If var4 ne . And var1 > 0 Then var7 = 1 / var1 ;
   If var5 >= 0 Then Do ;
   var8 = sqrt(var5) ;
   var9 = log(var5 +1) ;
   End ;
Run ;
```



L'étape DATA

Créer des variables dans *newtable* à partir de *oldtable*

- Exemple de programmation avec ce cas de figure

```
Data newtable ;
Set oldtable ;
var3 = var1 + 3*var2 ;
var4 = log10(var1) – 4*var2 ;
var5 = (-2)*var3 + var2**2 ;
If var4 > 8 Then var6 = 1 ; Else var6 = 0 ;
If var4 ne . And var1 > 0 Then var7 = 1 / var1 ;
   If var5 >= 0 Then Do ;
   var8 = sqrt(var5) ;
   var9 = log(var5 +1) ;
   End ;
Run ;
```

- Création d'une seule variable sous condition :

  If … **Then** … **; Else** … **;**

- Attention à l'utilisation de « **Else** » : on oublie souvent certaines alternatives à la condition de « **If** »
- Si *var4* est manquante pour un sujet, sa valeur est $-\infty \Rightarrow$ *var6* = 0 pour ce sujet !

L'étape DATA

Créer des variables dans *newtable* à partir de *oldtable*

- Exemple de programmation avec ce cas de figure

```
Data newtable ;
Set oldtable ;
var3 = var1 + 3*var2 ;
var4 = log10(var1) – 4*var2 ;
var5 = (-2)*var3 + var2**2 ;
If var4 > 8 Then var6 = 1 ; Else var6 = 0 ;
If var4 ne . And var1 > 0 Then var7 = 1 / var1 ;
   If var5 >= 0 Then Do ;
   var8 = sqrt(var5) ;
   var9 = log(var5 +1) ;
   End ;
Run ;
```

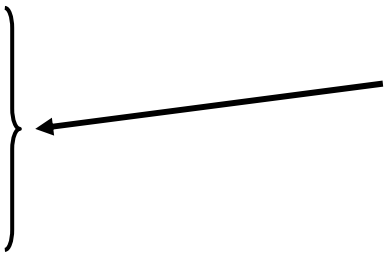
- Le mot-clé « ne » signifie « non équivalent à » (donc, « différent de »)
- Les mots-clés « And » et « Or » permettent de cumuler des conditions
- Si *var4* = . ou *var1* ≤ 0 (y compris *var1* manquante), alors *var7* = .

L'étape DATA

Créer des variables dans *newtable* à partir de *oldtable*

- Exemple de programmation avec ce cas de figure

```
Data newtable ;
Set oldtable ;
var3 = var1 + 3*var2 ;
var4 = log10(var1) – 4*var2 ;
var5 = (-2)*var3 + var2**2 ;
If var4 > 8 Then var6 = 1 ; Else var6 = 0 ;
If var4 ne . And var1 > 0 Then var7 = 1 / var1 ;
   If var5 >= 0 Then Do ;
   var8 = sqrt(var5) ;
   var9 = log(var5 +1) ;
   End ;
Run ;
```



- Dès qu'il y a plusieurs variables créées sous une (ou plusieurs) condition(s), il faut un « **Then Do ;** », qui doit se terminer par un « **End ;** »
- Si *var5* < 0 (y compris *var5* manquante), alors *var8* = *var9* = .

Création d'un label de variable

- Un label de variable est un ensemble de mots qui décrivent en détails la variable
- Le label de variable apparaît dans les sorties SAS (tableaux, …)
- Syntaxe

  Label *var* **= "**Voici le label de la variable *var***" ;**

- Exemple

```
Data newtable ;
Set oldtable ;
var3 = log10(var1) ;
Label var3 = "Logarithme décimal de var3" ;
Run ;
```

Format des variables numériques

- Un format d'une variable numérique permet la modification d'**affichage** des valeurs de la variable
- On formate une variable pour rendre la lecture de ses valeurs plus lisibles
- Un formatage ne modifie jamais la valeur intrinsèque de la variable
- Syntaxe

Format *var type_format* **;** **/*** affectation unique ***/**

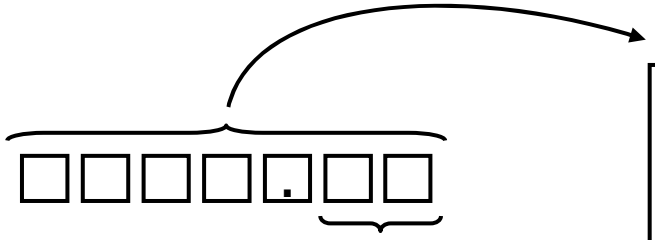
Format *var1 type_format1 var2 var3 type_format2* **;** **/*** affectation multiple ***/**

Affectation du format *type_format1* à la variable *var1* et du format *type_format2* aux variables *var2* et *var3*

Format des variables numériques

- Liste des formats

  - **Format** *var* X.Y **;**



X : nombre maxi de chiffres qui apparaîtront, y compris le point (virgule)

Y : nombre de chiffres après la virgule

Exemples à partir de la valeur non formatée : 12.46788

Format *var* 1.0 ⇒ * (impossible pour SAS d'afficher la valeur)

Format *var* 2.0 ⇒ 12

Format *var* 3.0 ⇒ 12

Format *var* 2.1 ⇒ 12

Format *var* 3.1 ⇒ 12

Format *var* 4.1 ⇒ 12.5

Format *var* 4.2 ⇒ 12.5

Format *var* 5.2 ⇒ 12.47

Format des variables numériques

- Liste des formats

  - **Format** *var* **z**X**.**Y **;**

    Identique au format X.Y sauf que les « cases » non remplies à gauches sont remplacées par des « 0 »

    Exemples à partir de la valeur non formatée : 12.46788

    Format *var* z1.0 ⇒ * (impossible pour SAS d'afficher la valeur)

    Format *var* z2.0 ⇒ 12

    Format *var* z3.0 ⇒ 012

    Format *var* z2.1 ⇒ 12

    Format *var* z3.1 ⇒ 012

    Format *var* z4.1 ⇒ 12.5

    Format *var* z4.2 ⇒ 12.5

    Format *var* z5.2 ⇒ 12.47

    Format *var* z6.2 ⇒ 012.47

Format des variables numériques

- Liste des formats

  - **Format** *var* **best**X**. ;**

    Affichage de maxi X chiffres (dont éventuellement le point)

    Exemples à partir de la valeur non formatée : 12.46788

    Format *var* best1. ⇒ * (impossible pour SAS d'afficher la valeur)

    Format *var* best2. ⇒ 12

    Format *var* best3. ⇒ 12

    Format *var* best4. ⇒ 12.5

    Format *var* best5. ⇒ 12.47

Les dates

Description générale

- Les variables relatives aux dates sont des variables numériques, dont la valeur est le nombre de jours écoulés depuis le 01/01/1960

- Une variable de date manquante pour un sujet vaudra « . »

- Si la valeur de *date_naiss* (variable de la date de naissance) d'un sujet vaut -31, cela signifie qu'il est né le 01/12/1959

Les dates

Description générale

- Les variables de dates sont nécessaires pour calculer des délais

  Attention, l'unité de tels délais sera en jours

- Exemple : calcul de l'âge à l'inclusion dans une étude, en années

```
Data newtable ;
Set oldtable ;
age = (date_inclusion – date_naissance) / 365.25 ;
Run ;
```

Les dates

Fonction mdy() pour la création de variables date

- La fonction mdy() permet de créer une variable de date à partir de 3 variables : variable de jours, de mois, et d'année

- Syntaxe

  var_date **= mdy(***var_mois*, *var_jours*, *var_annee***) ;**

- Exemple de programmation SAS

```
Data newtable ;
Set oldtable ;
date = mdy(mois, jours, annee) ;
Run ;
```

oldtable

id	jours	mois	annee
1	22	3	1987
2	13	11	2002

newtable

id	jours	mois	annee	date
1	22	3	1987	9942
2	13	11	2002	15657

Les dates

Fonction mdy() pour la modification de variables date

- La fonction mdy() permet aussi de modifier « localement » une date pour un sujet particulier

- Syntaxe

If *condition* **Then** *var_date* **= mdy(***valeur_mois*, *valeur_jours*, *valeur_annee***) ;**

- Exemple de programmation SAS

```
Data newtable_corrigee ;
Set newtable ;
If id = 2 Then date = mdy(1, 13, 2002) ;
Run ;
```

newtable

id	jours	mois	annee	date
1	22	3	1987	9942
2	13	11	2002	15657

Newtable_corrigee

id	jours	mois	annee	date
1	22	3	1987	9942
2	13	11	2002	15353

Les dates

Formatage de dates

- Le formatage d'une date permet de voir apparaître la date sous forme d'une date, et non pas sous forme d'un nombre de jours écoulés depuis 1960

- Rappel

  Le formatage d'une date ne modifie pas sa valeur intrinsèque, la variable est toujours le nombre écoulés de jours depuis 1960

- Syntaxe

  Format *var_date* **ddmmyy10. ;**

Les dates

Formatage de dates

- Exemple de programmation sous SAS

```
Data newtable_formatee ;
Set oldtable ;
date = mdy(mois, jours, annee) ;
date_bis = date ;
Format date_bis ddmmyy10. ;
Run ;
```

oldtable

id	jours	mois	annee
1	22	3	1987
2	13	11	2002

Newtable_formatee

id	jours	mois	annee	date	date_bis
1	22	3	1987	9942	22/03/1987
2	13	11	2002	15657	13/11/2002

Manipulation de tables SAS

Modification de la structure d'une table

Sélection / suppression de variables

- Pour « alléger » une table, on peut vouloir supprimer ou ne conserver que certaines variables

- Commandes SAS : **keep=** ou **drop=**

- Syntaxe pour sélectionner des variables

  Data newtable **(keep =** *var1 var2 var3* … **) ;**

  Set oldtable **;**

  [*instructions…* **;**]

  Run ;

  - Commande qui sélectionne les variables *var1*, *var2*, *var3*, …
  - Les variables sélectionnées appartiennent soit à *oldtable* soit ont été créées dans l'étape DATA

- La syntaxe pour supprimer les variables d'une table utilise de la même façon « **(drop =** *var1 var2 var3 …* **)** »

Modification de la structure d'une table

Sélection / suppression de variables

- Exemples d'écritures « raccourcies » en utilisant « **keep =** »
  (écritures identiques avec « **drop =** »)

  - **keep =** var1-var10 : sélectionne les variables *var1*, *var2*, *var3*, …, *var9*, et *var10*

  - **keep =** var1--var10 : sélectionne les variables de la table qui sont physiquement comprises entre *var1* et *var10* (par exemple : *var1*, *sexe*, *var3*, …, *var9*, et *var10*)

Modification de la structure d'une table

Sélection / suppression d'observations

- On peut vouloir supprimer de la table certaines observations (par exemple, les sujets dont le poids à l'inclusion est manquant) ou ne conserver que certaines observations (par exemple, les sujets qui ont eu un cancer)

- Commandes SAS :

  If *condition* **then output ;** ← Conserve les observations qui répondent à *condition*

  If *condition* **then delete ;** ← Supprime les observations qui répondent à *condition*

- Remarque : il est préférable de taper les commandes **output** ou **delete** à la fin de l'étape DATA

Modification de la structure d'une table

Sélection / suppression d'observations

- Exemple : sélection des sujets selon qu'il y a ou non une erreur sur l'âge (âge négatif, nul, manquant, ou âge > 97 ans)

  Etape 1 : création de la variable *erreur*

```
Data a ;
Set oldtable ;
erreur = 0 ;
If age <= 0 or age > 97 then erreur = 1 ;
Run ;
```

Modification de la structure d'une table

Sélection / suppression d'observations

- Exemple : sélection des sujets selon qu'il y a ou non une erreur sur l'âge (âge négatif, nul, manquant, ou âge > 97 ans)

  Etape 2 : sélection des sujets dans deux fichiers distincts

```
Data pb_age ;
Set a ;
If erreur = 1 then output ;
Run ;

Data sans_pb_age ;
Set a ;
If erreur = 0 then output ;
Run ;
```

Modification de la structure d'une table

Sélection / suppression d'observations

- Exemple : sélection des sujets selon qu'il y a ou non une erreur sur l'âge (âge négatif, nul, manquant, ou âge > 97 ans)

  Ecriture plus condensée

```
Data pb_age  ;
Set a ;
If age <= 0 or age > 97 then output ;
Run ;

Data sans_pb_age  ;
Set a ;
If age <= 0 or age > 97 then delete ;
Run ;
```

Création de tables

Création d'une table à partir d'une table

- Cas le plus simple, déjà vu

Data [*bibliothèque1***.**]*newtable* **;**
Set [*bibliothèque2***.**]*oldtable* **;**
[*Instruction*… **;**]
Run ;

S'il n'y a pas d'instruction, cette étape DATA est équivalent à copier *oldtable* (située éventuellement dans le dossier pointé par *bibliothèque2*) et à la coller éventuellement dans un autre dossier pointé par *bibliothèque1*] sous le nom de *newtable*

Création de tables

Création d'une table à partir de plusieurs tables : fusion de tables

- Fusion verticale : les sujets sont différents, mais en général le nom des variables dans les tables sources sont identiques

  - Illustration

table1

Sujet	Age	Sexe	Poids
1	28	1	63.5
2	32	1	56.7
4	45	0	81.6

+

table2

Sujet	Taille	Sexe	Poids
3	162.3	1	59.7
5	180.1	0	83.8
6	180.7	0	93.2

newtable

Sujet	Age	Sexe	Poids	Taille
1	28	1	63.5	.
2	32	1	56.7	.
4	45	0	81.6	.
3	.	1	59.7	162.3
5	.	0	83.8	180.1
6	.	0	93.2	180.7

Création de tables

Création d'une table à partir de plusieurs tables : fusion de tables

- Fusion verticale : les sujets sont différents, mais en général le nom des variables dans les tables sources sont identiques

  - Programme SAS

    Data [*bibliothèque3***.**]*newtable* **;**
    Set [*bibliothèque1***.**]*table1* [*bibliothèque2***.**]*table2* **;**
    [*Instruction*… **;**]
    Run ;

    Les bibliothèques *bibliothèque1*, *bibliothèque2*, et *bibliothèque3* peuvent bien sûr être identiques

Création de tables

Création d'une table à partir de plusieurs tables : fusion de tables

- Fusion verticale : les sujets sont différents, mais en général le nom des variables dans les tables sources sont identiques

  - Il est préférable que les tables *table1* et *table2* aient les même variables pour éviter les données manquantes

  - Exemple : recueil de données d'une enquête multicentrique

    On recueille les mêmes expositions ou caractéristiques des individus dans des sites (cliniques) différents

    $\Rightarrow$ On aura autant de tables que de sites, avec des individus différents, mais avec les mêmes variables

Création de tables

Création d'une table à partir de plusieurs tables : fusion de tables

- Fusion horizontale : les sujets sont *a priori* identiques, et les variables doivent être différentes (à part la variable d'identifiant)

  - Illustration n°1 : sujets identiques (1 ligne / sujet)

table1

Sujet	Age	Poids
1	28	63.5
2	32	76.7
3	45	81.6

+

Sujet	Taille	Sexe
1	162.3	1
2	180.1	0
3	180.7	0

table2

newtable

Sujet	Age	Poids	Taille	Sexe
1	28	63.5	162.3	1
2	32	76.7	180.1	0
3	45	81.6	180.7	0

Création de tables

Création d'une table à partir de plusieurs tables : fusion de tables

- Fusion horizontale : les sujets sont *a priori* identiques, et les variables doivent être différentes (à part la variable d'identifiant)

  - Illustration n°2 : sujets différents (1 ligne / sujet)

table1

Sujet	Age	Poids
1	28	63.5
2	32	76.7
3	45	81.6

+

table2

Sujet	Taille	Sexe
2	180.1	0
3	180.7	0
4	168.5	1

newtable

Sujet	Age	Poids	Taille	Sexe
1	28	63.5	.	.
2	32	76.7	180.1	0
3	45	81.6	180.7	0
4	.	.	168.5	1

Création de tables

Création d'une table à partir de plusieurs tables : fusion de tables

- Fusion horizontale : les sujets sont *a priori* identiques, et les variables doivent être différentes (à part la variable d'identifiant)

  - Illustration n°3 : sujets différents (plusieurs lignes / sujet)

table1

Sujet	Age
1	28
2	32
3	45

+

table2

Sujet	Visite	CD4
1	1	221
1	2	356
1	3	441
2	1	741
2	2	733
4	1	521
4	2	301

newtable

Sujet	Visite	CD4	Age
1	1	221	28
1	2	356	28
1	3	441	28
2	1	741	32
2	2	733	32
3	.	.	45
4	1	521	.
4	2	301	.

Création de tables

Création d'une table à partir de plusieurs tables : fusion de tables

- Fusion horizontale : les sujets sont *a priori* identiques, et les variables doivent être différentes (à part la variable d'identifiant)

  - Programme SAS

```
PROC SORT Data = [bibliothèque1.]table1 ;
By var_identifiant ;
Run ;

PROC SORT Data = [bibliothèque2.]table2 ;
By var_identifiant ;
Run ;
```

Etape indispensable de tri selon la variable identifiant *var_identifiant* grâce à la procédure PROC SORT

Création de tables

Création d'une table à partir de plusieurs tables : fusion de tables

- Fusion horizontale : les sujets sont *a priori* identiques, et les variables doivent être différentes (à part la variable d'identifiant)

  - Programme SAS (suite)

    Data [*bibliothèque3***.**]*newtable* **;**
    Merge [*bibliothèque1***.**]*table1* [*bibliothèque2***.**]*table2* **;**
    By *var_identifiant* **;**
    [*Instruction*… **;**]
    Run ;

  - La commande « **By** *var_identifiant* **;** »…

    … est indispensable !

    … nécessite d'avoir trié les tables sources par la variable *var_identifiant*

Création de tables

Création de plusieurs tables à partir d'une table

- Ventilation : en fonction de certaines conditions, on peut ventiler les observations à destination de tables différentes

  - Illustration : création de 2 tables *table_H* et *table_F* à partir de *oldtable* en fonction du sexe des individus (« 1 » pour les femmes, « 0 » pour les hommes)

oldtable

Sujet	Age	Sexe	Poids	Taille	
1	28	1	63.5	.	→ Table_F
2	32	1	56.7	.	→ Table_F
4	45	0	81.6	.	→ Table_H
3	.	1	59.7	162.3	→ Table_F
5	.	0	83.8	180.1	→ Table_H
6	.	0	93.2	180.7	→ Table_H

Création de tables

Création de plusieurs tables à partir d'une table

- Ventilation : en fonction de certaines conditions, on peut ventiler les observations à destination de tables différentes

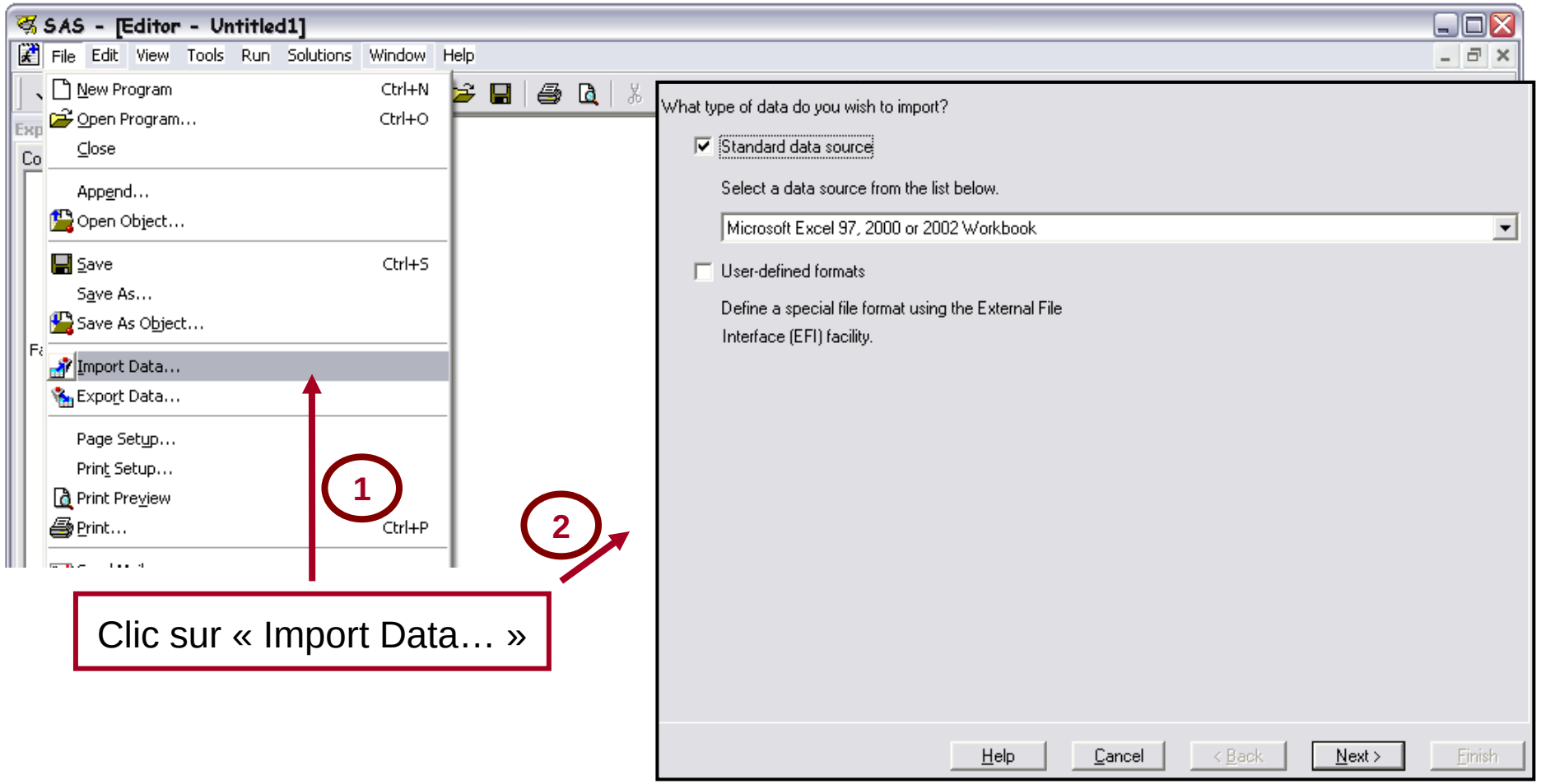
  - Commandes SAS : **If** *condition* **Then Output** *nom_table* **;**

  - Programme SAS avec l'illustration précédente

```
Data [bibliothèqueH.]table_H [bibliothèqueF.]table_F ;
Set [bibliothèque1.]oldtable ;
[Instruction… ;]
If sexe = 0 Then Output table_H ;
If sexe = 1 Then Output table_F ;
Run ;
```

Création de tables

Création d'une table à partir d'un fichier Excel (import)



Clic sur « Import Data… »

Création de tables

Création d'une table à partir d'un fichier Excel (import)

- Attention !

  Si votre séparateur de décimale est une virgule dans Excel, la virgule sera conservée dans l'import

  ⇒ Toutes les variables avec virgules seront considérées dans la table SAS comme des variables alphanumériques !!

- Solution

  Dans Excel, Outils -> Options -> International, puis modifier le séparateur de décimale

Message d'erreurs dans la fenêtre « Log »

Exemple d'erreurs en rouge

- Illustration n°1

```
1
2     data data_exo2;
3     set TD_M2bio.exo2_m2bio;
ERROR: Libname TD_M2BIO is not assigned.
4
5     if date_cancer = .  then cancer = 0;
6     if date_cancer ne . then cancer = 1;
7
8     if sexe = 1 and tabac = 1 then homme_fumeur = 1;
9     if sexe = 2 or  tabac = 0 then homme_fumeur = 0;
10
11    age_inc =     INT( (date_inc - date_naiss)/365.25 );
12    age_cancer = INT( (date_cancer - date_naiss)/365.25 );  /* manquant si date_cancer = . */
13
14    if .  < age_cancer <= 60 then age_K_cl = 0;
15    if 60 < age_cancer <= 70 then age_K_cl = 1;
16    if 70 < age_cancer <= 80 then age_K_cl = 2;
17    if 80 < age_cancer       then age_K_cl = 3;
18
19    diff_poids_relative = ROUND( ( poids_cancer - poids_inc ) / poids_inc * 100, 0.1);
20
21    label   age_K_cl = "Age au cancer en 4 classes"
22            diff_poids_relative = "Différence relative de poids au cancer (%)"
23            ;
24
25    format date_naiss date_inc date_cancer ddmmyy10.;
26
27    run;

NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.DATA_EXO2 may be incomplete.  When this step was stopped there were 0 observations and 13
         variables.
NOTE: DATA statement used (Total process time):
      real time           0.28 seconds
      cpu time            0.03 seconds
```

Message d'erreurs dans la fenêtre « Log »

Exemple d'erreurs en rouge

- Illustration n°2

```
55   data data_exo2;
56   set TD_M2bio.exo2_m2bio;
57
58   if date_cancer = .  then cancer = 0;
59   if date_cancer ne . then cancer = 1;
60
61   if sexe = 1 and tabac = 1 then homme_fumeur = 1;
62   if sexe = 2 or  tabac = 0 then homme_fumeur = 0;
63
64   age_inc =     INT( (date_inc - date_naiss)/365.25 );
65   age_cancer = INT( (date_cancer - date_naiss)/365.25 );  /* manquant si date_cancer = . */
66
67   if .  < age_cancer <= 60 then age_K_cl = 0;
68   if 60 < age_cancer <= 70 then age_K_cl = 1;
69   if 70 < age_cancer <= 80 then age_K_cl = 2;
70   if 80 < age_cancer       then age_K_cl = 3;
71
72   diff_poids_relative = ROND( ( poids_cancer - poids_inc ) / poids_inc * 100, 0.1);
                           ----
                           68
ERROR 68-185: The function ROND is unknown, or cannot be accessed.

73
74   label   age_K_cl = "Age au cancer en 4 classes"
75           diff_poids_relative = "Différence relative de poids au cancer (%)"
76           ;
77
78   format date_naiss date_inc date_cancer ddmmyy10.;
79
80   run;

NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.DATA_EXO2 may be incomplete.  When this step was stopped there were 0 observations and 14
         variables.
WARNING: Data set WORK.DATA_EXO2 was not replaced because this step was stopped.
NOTE: DATA statement used (Total process time):
      real time           0.09 seconds
      cpu time            0.04 seconds
```

Message d'erreurs dans la fenêtre « Log »

Exemple d'erreurs en rouge

- Illustration n°3

```
29    data data_exo2;
30    set TD_M2bio.exo2_m2bio;
31
32    if date_cancer = .  then cancer = 0;
33    if date_cancer ne . then cancer = 1;
34
35    if sexe = 1 and tabac = 1 then homme_fumeur = 1;
36    if sexe = 2 or  tabac = 0 then homme_fumeur = 0;
37
38    age_inc =     INT( (date_inc - date_naiss)/365.25 );
39    age_cancer = INT( (date_cancer - date_naiss)/365.25 );  /* manquant si date_cancer = . */
40
41    if .  < age_cancer <= 60 then age_K_cl = 0;
42    if 60 < age_cancer <= 70 then age_K_cl = 1;
43    if 70 < age_cancer <= 80 then age_K_cl = 2;
44    if 80 < age_cancer       then age_K_cl = 3;
45
46    diff_poids_relative = ROUND( ( poids_cancer - poids_inc ) / poids_inc * 100, 0.1;
                                                                                    -
                                                                                    79
ERROR 79-322: Expecting a ).

47
48    label   age_K_cl = "Age au cancer en 4 classes"
49            diff_poids_relative = "Différence relative de poids au cancer (%)"
50            ;
51
52    format date_naiss date_inc date_cancer ddmmyy10.;
53
54    run;

NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.DATA_EXO2 may be incomplete.  When this step was stopped there were 0 observations and 14
         variables.
WARNING: Data set WORK.DATA_EXO2 was not replaced because this step was stopped.
NOTE: DATA statement used (Total process time):
      real time           0.09 seconds
      cpu time            0.01 seconds
```

Message d'erreurs dans la fenêtre « Log »

Exemple d'erreurs en bleu (« variable non initialisée »)

- Illustration

```
185  data data_exo2;
186  set TD_M2bio.exo2_m2bio;
187
188  if date_cancer = .  then cancer = 0;
189  if date_cancer ne . then cancer = 1;
190
191  if sexe = 1 and tabac = 1 then homme_fumeur = 1;
192  if sexe = 2 or  tabac = 0 then homme_fumeur = 0;
193
194  age_inc =     INT( (date_inc      - date_naiss)/365.25 );
195  age_cancer = INT( (date_cance - date_naiss)/365.25 );   /* manquant si date_cancer = . */
196
197  if .  < age_cancer <= 60 then age_K_cl = 0;
198  if 60 < age_cancer <= 70 then age_K_cl = 1;
199  if 70 < age_cancer <= 80 then age_K_cl = 2;
200  if 80 < age_cancer       then age_K_cl = 3;
201
202  diff_poids_relative = ROUND( ( poids_cancer - poids_inc
203
204  label   age_K_cl = "Age au cancer en 4 classes"
205          diff_poids_relative = "Différence relative de po
206          ;
207
208  format date_naiss date_inc date_cancer ddmmyy10.;
209
210  run;

NOTE: Variable date_cance is uninitialized.
NOTE: Missing values were generated as a result of performing
      Each place is given by: (Number of times) at (Line):(Column).
      786 at 195:14   786 at 195:31   786 at 195:44   578 at 202:23   578 at 202:45   578 at 202:59   578 at 202:71
NOTE: There were 786 observations read from the data set TD_M2BIO.EXO2_M2BIO.
NOTE: The data set WORK.DATA_EXO2 has 786 observations and 15 variables.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.01 seconds
```

Lorsque SAS dit qu'une variable est non initialisée (« uninitialized »), cela signifie qu'il y a eu une faute de frappe !

Ici, dans le programme, on a tapé « date_cance » au lieu de « date-cancer »

Message d’erreurs dans la fenêtre « Log »

Commentaires

- Il faut toujours vérifier la fenêtre « Log » dès que l’on exécute une ligne de programme

- Il n’y a pas que les messages en rouge qui signalent des erreurs, mais aussi les messages en bleu !

- Ne pas hésiter à vider (très) fréquemment le contenu de la fenêtre « Log » pour n’avoir que les dernières lignes de programmes exécutées dans toute la fenêtre

  (Rappel : « Ctrl + E »)

Procédure d’analyses statistiques univariées et bivariées

Nature des variables dans une table

Nature des variables numériques : notations

- **binaire** (très souvent variable en Oui/Non)

  Consommation de tabac, antécédents de cancer, présence d'un traitement, être malade, …

- **qualitative** (variable en plusieurs classes)

  ordinale : niveau d'études, dose de radiations reçue (faible, moyenne, forte), indice de satisfaction dans un questionnaire de qualité de vie, ...

  nominale : zone d'habitation, catégories socio-professionnelles, état matrimonial (célibataire, marié(e), divorcé(e), …), ...

- **quantitative** : âge, poids, durée de symptômes (en jours), …

Nature des variables dans une table

Quelques remarques préliminaires à propos du codage des variables

A de très rares exceptions près, les modalités des variables binaires et qualitatives doivent être codées de façon numérique pour être traitées statistiquement

- Variables binaires

  Dans le fichier de données, ces variables sont très souvent codées « 0 » pour « non », et « 1 » pour « oui »

  Une polémique a toujours existé avec la variable *sexe*, généralement codée « 1 » pour les hommes, et « 2 » pour les femmes

Nature des variables dans une table

Quelques remarques préliminaires à propos du codage des variables

- Variables qualitatives

  - Ces variables sont codées « 0 », « 1 », « 2 », …ou bien « 1 », « 2 », « 3 », ...

  - Dans le cas des variables **ordinales**, l'ordre a un sens : dire que « 2 » est plus grand que « 1 », qui lui-même est plus grand que « 0 », doit avoir un sens

  - Ce qui n'est pas le cas des variables **nominales** : pour le statut marital, on codera par exemple « 0 » pour le statut célibataire, « 1 » pour le statut marié, et « 2 » pour le statut divorcé, sans que cet ordre ait un sens

Liste des procédures de statistiques

Procédures de statistiques descriptives

- PROC CONTENTS
- PROC PRINT
- PROC FREQ
- PROC UNIVARIATE
- PROC MEANS
- PROC GPLOT
- PROC BOXPLOT

Liste des procédures de statistiques

Procédures de statistiques comparatives (tests statistiques)

- PROC FREQ avec Chi-deux
- PROC TTEST
- PROC ANOVA
- PROC CORR
- PROC NPAR1WAY

Syntaxe générale des procédures

- Syntaxe

PROC ***NOM_PROCEDURE*** **Data =** [*bibliothèque*.]*nom_table* [***options***] **;**

[*Instructions spécifiques à chaque procédure* **;**]

[**Title "***Titre à afficher dans la fenêtre Output ou dans la fenêtre graphique***" ;**]

[**Where** *condition* **;**]

Run ;

- Commentaires

  - La commande « **Data =** [*bibliothèque*.]*nom_table* » n'est pas indispensable, mais fortement recommandée pour spécifier sur quelle table SAS on veut exécuter la procédure

  - Pour supprimer le titre, il faut taper en dehors d'une procédure ou d'une étape DATA la commande « **Title ;** »

Syntaxe générale des procédures

- Syntaxe

  PROC ***NOM_PROCEDURE*** **Data =** [*bibliothèque*.]*nom_table* [***options***] **;**

  [*Instructions spécifiques à chaque procédure* **;**]

  [**Title "***Titre à afficher dans la fenêtre Output ou dans la fenêtre graphique***" ;**]

  [**Where** *condition* **;**]

  Run ;

- Commentaires (suite)

  - La commande « **Where** *condition* **;** » permet de n'exécuter la procédure que sur une sélection d'individus

    Exemple : « **Where** sexe **=** 2 **and** age **>** 35 **;** »

  - Les résultats issus de l'exécution d'une procédure sont affichés dans la fenêtre Output

PROC CONTENTS

- Descriptif de la procédure

  La procédure PROC CONTENTS fournit un grand nombre d'informations sur la table

- Syntaxe

  PROC CONTENTS Data = [*bibliothèque***.**]*nom_table* **;**

  Run ;

PROC CONTENTS

- Illustration

Programme dans la fenêtre « Editor »

```
proc contents data = TD_M2bio.data_exo2;
run;
```

Sortie dans la fenêtre « Output »

The CONTENTS Procedure

Data Set Name	TD_M2BIO.DATA_EXO2	Observations	786
Member Type	DATA	Variables	14
Engine	V9	Indexes	0
Created	mardi 11 août 2009 17 h 44	Observation Length	112
Last Modified	mardi 11 août 2009 17 h 44	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Information de la table *data_exo2*

Engine/Host Dependent Information

Information sur les variables contenues dans *data_exo2*

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
13	age_K_cl	Num	8		Age au cancer en 4 classes
12	age_cancer	Num	8		
11	age_inc	Num	8		
9	cancer	Num	8		
8	date_cancer	Num	8	DDMMYY10.	
7	date_inc	Num	8	DDMMYY10.	
6	date_naiss	Num	8	DDMMYY10.	
14	diff_poids_relative	Num	8		Différence relative de poids au cancer (%)
10	homme_fumeur	Num	8		
1	id	Num	8		
5	poids_cancer	Num	8		
3	poids_inc	Num	8		
2	sexe	Num	8		
4	tabac	Num	8		

PROC PRINT

- Descriptif de la procédure

  La procédure PROC PRINT affiche tout le contenu de la table, ou bien seulement certaines variables

- Syntaxe

  PROC PRINT Data = [*bibliothèque***.**]*nom_table* [***options***] **;**

  [**Var** *variable1 variable2 … variablek* **;**]

  Run ;

- La commande (optionnelle) « **Var** *variable1 variable2* … *variablek* **;** » permet de n'afficher que les valeurs des variables listées dans la commande

PROC PRINT

- Options de PROC PRINT

  - noobs : permet de ne pas afficher le n° de la ligne

  - n : permet d'afficher combien la table contient de lignes

  - heading=V : permet d'afficher le nom des variables verticalement

  - heading=H : permet d'afficher le nom des variables horinzontalement

PROC PRINT

- Illustration n°1

```
proc print data = TD_M2bio.data_exo2;
run;
```

Obs	id	sexe	poids_ inc	tabac	poids_ cancer	date_naiss	date_inc	date_ cancer	cancer	homme_ fumeur	age_inc	age_ cancer	age_K_cl	diff_poids_ relative
1	1	1	68.8	0	.	23/08/1923	22/08/1960	.	0	0	36	.	.	.
2	2	2	65.0	1	.	11/11/1929	14/05/1960	.	0	0	30	.	.	.
3	3	1	74.5	.	.	02/01/1932	11/10/1959	.	0	.	27	.	.	.
4	4	2	.	0	.	22/09/1921	18/11/1959	.	0	0	38	.	.	.
5	5	2	72.1	.	.	26/01/1925	15/02/1958	.	0	0	33	.	.	.
6	6	1	73.7	0	.	24/04/1923	16/07/1958	.	0	0	35	.	.	.
7	7	2	66.4	1	.	11/12/1920	26/09/1958	.	0	0	37	.	.	.
8	8	2	58.0	0	.	30/01/1922	19/09/1958	.	0	0	36	.	.	.
9	9	1	84.5	1	.	24/01/1929	23/02/1959	.	0	1	30	.	.	.
10	10	2	64.9	.	.	02/05/1921	19/07/1959	.	0	0	38	.	.	.
11	11	1	71.5	0	57.9	08/04/1935	27/03/1958	13/01/2002	1	0	22	66	1	-19.0
777	777	1	88.4	0	73.1	30/12/1923	08/01/1960	06/10/1993	1	0	36	69	1	-17.3
778	778	1	86.6	.	.	10/08/1936	20/10/1960	.	0	.	24	.	.	.
779	779	1	65.6	1	56.3	02/05/1922	11/11/1958	13/02/1989	1	1	36	66	1	-14.2
780	780	1	92.0	0	.	14/10/1928	05/08/1958	.	0	0	29	.	.	.
781	781	1	.	1	.	01/12/1929	02/12/1959	.	0	1	30	.	.	.
782	782	2	58.0	1	50.8	27/07/1933	23/06/1958	19/10/2006	1	0	24	73	2	-12.4
783	783	2	60.8	1	.	26/06/1926	03/01/1960	.	0	0	33	.	.	.
784	784	2	66.1	0	.	17/05/1922	06/03/1960	.	0	0	37	.	.	.
785	785	1	84.1	0	.	31/05/1924	03/07/1958	.	0	0	34	.	.	.
786	786	2	74.6	.	.	23/06/1923	04/09/1959	.	0	0	36	.	.	.

Colonne « obs » qui donne le n° de la ligne dans la tab le

PROC PRINT

- Illustration n°2

```
proc print data = TD_M2bio.data_exo2 noobs n ;
var id poids_inc poids_cancer diff_poids_relative;
where sexe = 2 and tabac = 0;
title "Table pour les femmes non fumeuses";
run;
```

Table pour les femmes non fumeuses

id	poids_ inc	poids_ cancer	diff_poids_ relative
4	.	.	.
8	58.0	.	.
16	69.3	.	.
19	75.0	67.9	-9.5
20	58.2	75.5	29.7
26	61.1	58.6	-4.1
31	63.6	.	.
760	71.2	63.0	-11.5
765	61.3	55.2	-10.0
768	73.4	70.7	-3.7
776	63.4	58.9	-7.1
784	66.1	.	.

N = 150

L'option « noobs » a enlevé la colonne « Obs » qui ne sert pas à grand-chose…

La table contient 150 femmes non fumeuses

PROC FREQ

- Descriptif de la procédure

  - La procédure PROC FREQ affiche des tableaux de fréquences simples ou croisés

  - Elle fournit aussi les % en ligne, en colonne, et par cellule (dans les tableaux croisés)

  - De façon optionnelle, elle fournit les tests du Chi-deux, de tendance, et le test de Fisher

- Remarque

  Bien évidemment, il n'est pas question d'utiliser la procédure PROC FREQ sur des variables quantitatives… !

PROC FREQ

- Syntaxe pour des tableaux simples

  PROC FREQ Data = [*bibliothèque***.**]*nom_table* **;**
  Table *variable1 variable2 … variablek* [**/ options**] **;**
  Run ;

- Options de PROC FREQ pour des tableaux simples

  - nocum : ne fournit pas les % cumulés
  - nopercent : ne fournit pas les %
  - missing : affiche les données manquantes comme une modalité

PROC FREQ

- Illustration n°1

```
proc freq data = TD_M2bio.data_exo2;
  table sexe tabac cancer age_K_cl  ;
  run;
```

The FREQ Procedure

sexe	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	400	50.89	400	50.89
2	386	49.11	786	100.00

tabac	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	380	54.13	380	54.13
1	322	45.87	702	100.00

Frequency Missing = 84

cancer	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	557	70.87	557	70.87
1	229	29.13	786	100.00

Age au cancer en 4 classes

age_K_cl	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	13	5.68	13	5.68
1	105	45.85	118	51.53
2	86	37.55	204	89.08
3	25	10.92	229	100.00

Frequency Missing = 557

Parmi les 786 sujets, il y a 51% d'hommes, 49% de femmes

Il y a 84 données manquantes sur la variable *tabac*

Il y a 229 sujets ayant eu un cancer (29% des sujets de l'étude)

Il y a 13 sujets ≤ 60 ans au moment d'un cancer (6% des 229 sujets avec cancer)

PROC FREQ

- Illustration n°2

```
proc freq data = TD_M2bio.data_exo2;
table age_K_cl / missing nocum nopercent ;
run;
```

The FREQ Procedure

Age au cancer en 4 classes

age_K_cl	Frequency
.	557
0	13
1	105
2	86
3	25

Les données manquantes de *age_K_cl* apparaissent comme une modalité

PROC FREQ

- Syntaxe pour des tableaux croisés

PROC FREQ Data = [*bibliothèque***.**]*nom_table* **;**
Table *variable1 * variable2 variable3 * (variable 4 variable5) …* [**/ options**] **;**
Run ;

La commande ci-dessus fournit les tableaux croisés suivants :

- *variable1* x *variable2*
- *variable3* x *variable4*
- *variable3* x *variable5*

- On peut « cumuler » la demande de tableaux simples et croisés

PROC FREQ Data = [*bibliothèque***.**]*nom_table* **;**
Table *variable1 variable2 variable3 * (variable 4 variable5) …* [**/ options**] **;**
Run ;

PROC FREQ

- Options de PROC FREQ pour des tableaux croisés
  - nocum : ne fournit pas les % cumulés
  - nopercent : ne fournit pas les % de la cellule par rapport au total
  - missing : affiche les données manquantes comme une modalité
  - nocol : ne fournit pas les % en colonne
  - norow : ne fournit pas les % en ligne
  - chisq : affiche les tests du Chi-deux et de tendance (+ le test de Fisher s'il s'agit d'un tableau croisé à 4 cases)
  - fisher : affiche le test de Fisher (à spécifier seulement si une des 2 variables a plus de 2 modalités)
  - expected : affiche le nombre attendu de sujets dans chaque case sous l'hypothèse nulle selon laquelle les variables croisées sont indépendantes (pas d'association entre ces deux variables)

PROC FREQ

- Illustration n°3

```
proc freq data = TD_M2bio.data_exo2;
table tabac * (sexe age_K_cl) / norow ;
run;
```

The FREQ Procedure

Table of tabac by sexe

tabac sexe

Frequency Percent Col Pct	1	2	Total
0	230 32.76 65.16	150 21.37 42.98	380 54.13
1	123 17.52 34.84	199 28.35 57.02	322 45.87
Total	353 50.28	349 49.72	702 100.00

Frequency Missing = 84

Table of tabac by age_K_cl

tabac age_K_cl(Age au cancer en 4 classes)

Frequency Percent Col Pct	0	1	2	3	Total
0	7 3.06 53.85	56 24.45 53.33	54 23.58 62.79	14 6.11 56.00	131 57.21
1	6 2.62 46.15	49 21.40 46.67	32 13.97 37.21	11 4.80 44.00	98 42.79
Total	13 5.68	105 45.85	86 37.55	25 10.92	229 100.00

Frequency Missing = 557

- 54% des sujets ne sont pas fumeurs (n=380)
- 230/353 (65%) des hommes sont non fumeurs contre 150/349 (43%) des femmes

  ⇒ Ces % sont-ils significativement différents ?

PROC FREQ

- Illustration n°3

```
proc freq data = TD_M2bio.data_exo2;
table tabac * (sexe age_K_cl) / norow ;
run;
```

The FREQ Procedure

Table of tabac by sexe

tabac sexe

Frequency Percent Col Pct	1	2	Total
0	230 32.76 65.16	150 21.37 42.98	380 54.13
1	123 17.52 34.84	199 28.35 57.02	322 45.87
Total	353 50.28	349 49.72	702 100.00

Frequency Missing = 84

Table of tabac by age_K_cl

tabac age_K_cl(Age au cancer en 4 classes)

Frequency Percent Col Pct	0	1	2	3	Total
0	7 3.06 53.85	56 24.45 53.33	54 23.58 62.79	14 6.11 56.00	131 57.21
1	6 2.62 46.15	49 21.40 46.67	32 13.97 37.21	11 4.80 44.00	98 42.79
Total	13 5.68	105 45.85	86 37.55	25 10.92	229 100.00

Frequency Missing = 557

- Parmi les 229 sujets qui ont eu un cancer…
  - 46% des sujets ≤ 60 ans sont fumeurs
  - 47% des sujets 60-70 ans sont fumeurs
  - 37% des sujets 70-80 ans sont fumeurs
  - 44% des sujets > 80 ans sont fumeurs

PROC FREQ

- Illustration n°4

```
proc freq data = TD_M2bio.data_exo2;
table tabac * sexe / missing norow;
run;
```

The FREQ Procedure

Table of tabac by sexe

tabac sexe

Frequency Percent Col Pct	1	2	Total
.	47 5.98 11.75	37 4.71 9.59	84 10.69
0	230 29.26 57.50	150 19.08 38.86	380 48.35
1	123 15.65 30.75	199 25.32 51.55	322 40.97
Total	400 50.89	386 49.11	786 100.00

12% des hommes ont des données manquantes sur la consommation de tabac, contre 10% des femmes

PROC FREQ

- Illustration n°5

```
proc freq data = TD_M2bio.data_exo2;
table sexe * cancer / chisq norow expected ;
run;
```

Table of sexe by cancer

sexe cancer

Frequency Expected Percent Col Pct	0	1	Total
1	298 283.46 37.91 53.50	102 116.54 12.98 44.54	400 50.89
2	259 273.54 32.95 46.50	127 112.46 16.16 55.46	386 49.11
Total	557 70.87	229 29.13	786 100.00

Statistics for Table of sexe by cancer

Statistic	DF	Value	Prob
Chi-Square	1	5.2122	0.0224
Likelihood Ratio Chi-Square	1	5.2183	0.0224
Continuity Adj. Chi-Square	1	4.8599	0.0275
Mantel-Haenszel Chi-Square	1	5.2056	0.0225
Phi Coefficient		0.0814	
Contingency Coefficient		0.0812	
Cramer's V		0.0814	

Fisher's Exact Test

Cell (1,1) Frequency (F)	298
Left-sided Pr <= F	0.9909
Right-sided Pr >= F	0.0137
Table Probability (P)	0.0046
Two-sided Pr <= P	0.0230

Sample Size = 786

Parmi les sujets sans cancer, il y a 47% de femmes, contre 55% des sujets avec cancer

⇒ Le % de femmes est donc plus élevé chez les sujets avec cancer que sans cancer

PROC FREQ

- Illustration n°5

```
proc freq data = TD_M2bio.data_exo2;
table sexe * cancer / chisq norow expected ;
run;
```

Table of sexe by cancer

sexe cancer

Frequency Expected Percent Col Pct	0	1	Total
1	298 283.46 37.91 53.50	102 116.54 12.98 44.54	400 50.89
2	259 273.54 32.95 46.50	127 112.46 16.16 55.46	386 49.11
Total	557 70.87	229 29.13	786 100.00

Statistics for Table of sexe by cancer

Statistic	DF	Value	Prob
Chi-Square	1	5.2122	0.0224
Likelihood Ratio Chi-Square	1	5.2183	0.0224
Continuity Adj. Chi-Square	1	4.8599	0.0275
Mantel-Haenszel Chi-Square	1	5.2056	0.0225
Phi Coefficient		0.0814	
Contingency Coefficient		0.0812	
Cramer's V		0.0814	

Fisher's Exact Test

Cell (1,1) Frequency (F)	298
Left-sided Pr <= F	0.9909
Right-sided Pr >= F	0.0137
Table Probability (P)	0.0046
Two-sided Pr <= P	0.0230

Sample Size = 786

Le test du Chi-deux teste si 47% est significativement différent de 55%

$\Rightarrow p = 0{,}02 < 0{,}05$ (5%)

$\Rightarrow$ Le test est significatif, et on peut dire que le sexe est significativement associé à la présence d'un cancer

PROC FREQ

- Illustration n°5

```
proc freq data = TD_M2bio.data_exo2;
table sexe * cancer / chisq norow expected ;
run;
```

Table of sexe by cancer

sexe cancer

Frequency Expected Percent Col Pct	0	1	Total
1	298 283.46 37.91 53.50	102 116.54 12.98 44.54	400 50.89
2	259 273.54 32.95 46.50	127 112.46 16.16 55.46	386 49.11
Total	557 70.87	229 29.13	786 100.00

Statistics for Table of sexe by cancer

Statistic	DF	Value	Prob
Chi-Square	1	5.2122	0.0224
Likelihood Ratio Chi-Square	1	5.2183	0.0224
Continuity Adj. Chi-Square	1	4.8599	0.0275
Mantel-Haenszel Chi-Square	1	5.2056	0.0225
Phi Coefficient		0.0814	
Contingency Coefficient		0.0812	
Cramer's V		0.0814	

Fisher's Exact Test

Cell (1,1) Frequency (F)	298
Left-sided Pr <= F	0.9909
Right-sided Pr >= F	0.0137
Table Probability (P)	0.0046
Two-sided Pr <= P	0.0230

Sample Size = 786

S'il n'y avait pas du tout d'association entre le sexe et la présence de cancer (hypthèse nulle H0), on aura eu 49% de femmes chez les sujets sans cancer, et 49% de femmes chez les sujets avec cancer

PROC FREQ

- Illustration n°5

```
proc freq data = TD_M2bio.data_exo2;
table sexe * cancer / chisq norow expected ;
run;
```

Table of sexe by cancer

sexe cancer

Frequency
Expected
Percent
Col Pct

	0	1	Total
1	298 283.46 37.91 53.50	102 116.54 12.98 44.54	400 50.89
2	259 273.54 32.95 46.50	127 112.46 16.16 55.46	386 49.11
Total	557 70.87	229 29.13	786 100.00

Statistics for Table of sexe by cancer

Statistic	DF	Value	Pr
Chi-Square	1	5.2122	0.0224
Likelihood Ratio Chi-Square	1	5.2183	0.0224
Continuity Adj. Chi-Square	1	4.8599	0.0275
Mantel-Haenszel Chi-Square	1	5.2056	0.0225
Phi Coefficient		0.0814	
Contingency Coefficient		0.0812	
Cramer's V		0.0814	

Fisher's Exact Test

Cell (1,1) Frequency (F)	298
Left-sided Pr <= F	0.9909
Right-sided Pr >= F	0.0137
Table Probability (P)	0.0046
Two-sided Pr <= P	0.0230

Sample Size = 786

Ce qui donne les effectifs attendus sous l'hypothèse nulle de :

- 274 femmes sans cancer (au lieu des 259 observées)
- 112 femmes avec cancer (au lieu des 127 observées)

PROC FREQ

- Illustration n°5

```
proc freq data = TD_M2bio.data_exo2;
table sexe * cancer / chisq norow expected ;
run;
```

Table of sexe by cancer

sexe cancer

Frequency Expected Percent Col Pct	0	1	Total
1	298 283.46 37.91 53.50	102 116.54 12.98 44.54	400 50.89
2	259 273.54 32.95 46.50	127 112.46 16.16 55.46	386 49.11
Total	557 70.87	229 29.13	786 100.00

Statistics for Table of sexe by cancer

Statistic	DF	Value	Prob
Chi-Square	1	5.2122	0.0224
Likelihood Ratio Chi-Square	1	5.2183	0.0224
Continuity Adj. Chi-Square	1	4.8599	0.0275
Mantel-Haenszel Chi-Square	1	5.2056	0.0225
Phi Coefficient		0.0814	
Contingency Coefficient		0.0812	
Cramer's V		0.0814	

Fisher's Exact Test

Cell (1,1) Frequency (F)	298
Left-sided Pr <= F	0.9909
Right-sided Pr >= F	0.0137
Table Probability (P)	0.0046
Two-sided Pr <= P	0.0230

Sample Size = 786

Remarque

Le test de Fisher donne la même statistique de test (p = 0,023) que le test du Chi-deux (p = 0,022)

PROC FREQ

- Illustration n°6

```
proc freq data = TD_M2bio.data_exo2;
 table sexe * age_K_cl / chisq norow expected ;
 run;
```

Table of sexe by age_K_cl

sexe age_K_cl(Age au cancer en 4 classes)

Frequency Expected Percent Col Pct	0	1	2	3	Total
1	5 5.7904 2.18 38.46	51 46.769 22.27 48.57	37 38.306 16.16 43.02	9 11.135 3.93 36.00	102 44.54
2	8 7.2096 3.49 61.54	54 58.231 23.58 51.43	49 47.694 21.40 56.98	16 13.865 6.99 64.00	127 55.46
Total	13 5.68	105 45.85	86 37.55	25 10.92	229 100.00

Frequency Missing = 557

Statistics for Table of sexe by age_K_cl

Statistic	DF	Value	Prob
Chi-Square	3	1.7035	0.6362
Likelihood Ratio Chi-Square	3	1.7160	0.6334
Mantel-Haenszel Chi-Square	1	0.6945	0.4046
Phi Coefficient		0.0862	
Contingency Coefficient		0.0859	
Cramer's V		0.0862	

Effective Sample Size = 229
Frequency Missing = 557

WARNING: 71% of the data are missing.

Condition d'application d'un test du Chi-deux : les effectifs attendus sous H_0 doivent tous être > 5 (c'est le cas ici)

PROC FREQ

- Illustration n°6

```
proc freq data = TD_M2bio.data_exo2;
 table sexe * age_K_cl / chisq norow expected ;
 run;
```

Table of sexe by age_K_cl

sexe age_K_cl(Age au cancer en 4 classes)

Frequency Expected Percent Col Pct	0	1	2	3	Total
1	5 5.7904 2.18 38.46	51 46.769 22.27 48.57	37 38.306 16.16 43.02	9 11.135 3.93 36.00	102 44.54
2	8 7.2096 3.49 61.54	54 58.231 23.58 51.43	49 47.694 21.40 56.98	16 13.865 6.99 64.00	127 55.46
Total	13 5.68	105 45.85	86 37.55	25 10.92	229 100.00

Frequency Missing = 557

Statistics for Table of sexe by age_K_cl

Statistic	DF	Value	Prob
Chi-Square	3	1.7035	0.6362
Likelihood Ratio Chi-Square	3	1.7160	0.6334
Mantel-Haenszel Chi-Square	1	0.6945	0.4046
Phi Coefficient		0.0862	
Contingency Coefficient		0.0859	
Cramer's V		0.0862	

Effective Sample Size = 229
Frequency Missing = 557

WARNING: 71% of the data are missing.

Le test du Chi-deux teste si les % de femmes dans chacune des classes de *age_K_cl* sont significativement différents (H_0 : les 4 % sont égaux ; H1 : ≥ 1 % est différent des autres)

Les 4 % ne sont pas significativement différents les uns des autres (p = 0,64)

PROC FREQ

- Illustration n°6

```
proc freq data = TD_M2bio.data_exo2;
 table sexe * age_K_cl / chisq norow expected ;
 run;
```

Table of sexe by age_K_cl

sexe age_K_cl(Age au cancer en 4 classes)

Frequency Expected Percent Col Pct	0	1	2	3	Total
1	5 5.7904 2.18 38.46	51 46.769 22.27 48.57	37 38.306 16.16 43.02	9 11.135 3.93 36.00	102 44.54
2	8 7.2096 3.49 61.54	54 58.231 23.58 51.43	49 47.694 21.40 56.98	16 13.865 6.99 64.00	127 55.46
Total	13 5.68	105 45.85	86 37.55	25 10.92	229 100.00

Frequency Missing = 557

Statistics for Table of sexe by age_K_cl

Statistic	DF	Value	Prob
Chi-Square	3	1.7035	0.6362
Likelihood Ratio Chi-Square	3	1.7160	0.6334
Mantel-Haenszel Chi-Square	1	0.6945	0.4046
Phi Coefficient		0.0862	
Contingency Coefficient		0.0859	
Cramer's V		0.0862	

Effective Sample Size = 229
Frequency Missing = 557

WARNING: 71% of the data are missing.

Le test de tendance (1 ddl) teste s'il existe une évolution du % de femmes avec l'augmentation *age_K_cl*

Il n'existe pas de tendance significative à l'augmentation ou à la diminution du % de femmes lorsque l'âge au cancer augmente (p = 0,41)

PROC UNIVARIATE

- Descriptif de la procédure

  - La procédure PROC UNIVARIATE affiche de nombreuses informations pour les variables quantitatives (moyenne, écart-type, médiane, …)

  - Le fait qu'elle fournisse beaucoup d'information la rend moins pratique que PROC MEANS

  - De façon optionnelle, elle fournit l'histogramme de la distribution de la variable quantitative et le graphique QQ-Plot

- Remarque

  Bien évidemment, il n'est pas question d'utiliser la procédure PROC UNIVARIATE sur des variables qualitatives binaires ou à plusieurs classes… !

PROC UNIVARIATE

- Syntaxe pour des descriptions simples

PROC UNIVARIATE Data = [*bibliothèque***.**]*nom_table* **;**

[**Var** *variable1 variable2 … variablek* **;**]

Run ;

Si la commande « **Var** *variable1* … *variablek* **;** » n'est pas écrite, SAS va exécuter cette procédure sur toutes les variables de la table, y compris sur les variables binaires et qualitatives !

PROC UNIVARIATE

- Syntaxe pour des descriptions croisées entre une variable quantitative et une ou plusieurs variables qualitatives

PROC UNIVARIATE Data = [*bibliothèque*.]*nom_table* **;**

Class *variable_qual1 variable_qual2 … variable_qualk* **;**

[**Var** *variable1 variable2 … variablek* **;**]

Run ;

- La commande « **Class** *variable_qual1* … *variable_qualk* **;** » permet de fournir l'ensemble des résultats de la PROC UNIVARIATE (moyenne, médiane, …) par modalité du k-uplet des variables qualitatives *variable_qual1* … *variable_qualk*

  (Avec 3 variables binaires dans la commande « **Class** », on va avoir 2^3 strates formées par ces 3 variables, donc 2^3 sorties pour les variables quantitatives de la commande « **Var** »)

- Dans la très grande majorité des cas, on préfère ne mettre qu'une seule variable qualitative après « **Class** » pour éviter des résultats ininterprétables !

PROC UNIVARIATE

- Illustration n°1

```
proc univariate data = TD_M2bio.data_exo2;
  var poids_inc ;
  run;
```

Sortie n°1

The UNIVARIATE Procedure
Variable: poids_inc

Moments

N	718	Sum Weights	718
Mean	70.6341226	Sum Observations	50715.3
Std Deviation	9.33849235	Variance	87.2074393
Skewness	0.61223655	Kurtosis	-0.4807254
Uncorrected SS	3644758.45	Corrected SS	62527.734
Coeff Variation	13.2209363	Std Error Mean	0.34850944

Basic Statistical Measures

Location		Variability	
Mean	70.63412	Std Deviation	9.33849
Median	69.60000	Variance	87.20744
Mode	74.60000	Range	36.70000
		Interquartile Range	12.10000

Tests for Location: Mu0=0

Test	-Statistic-		-----p Value------
Student's t	t	202.6749	Pr > |t| <.0001
Sign	M	359	Pr >= |M| <.0001
Signed Rank	S	129060.5	Pr >= |S| <.0001

Informations utiles

- 718 observations non manquantes
- Moyenne : 70,6 kg
- Ecart-type (SD) : 9,3 kg
- Variance : 87,2 kg^2
- Ecart-type de m (SE) : 0,35 kg

PROC UNIVARIATE

- Illustration n°1

```
proc univariate data = TD_M2bio.data_exo2;
  var poids_inc ;
  run;
```

Sortie n°1

The UNIVARIATE Procedure
Variable: poids_inc

Moments

N	718	Sum Weights	718
Mean	70.6341226	Sum Observations	50715.3
Std Deviation	9.33849235	Variance	87.2074393
Skewness	0.61223655	Kurtosis	-0.4807254
Uncorrected SS	3644758.45	Corrected SS	62527.734
Coeff Variation	13.2209363	Std Error Mean	0.34850944

Basic Statistical Measures

Location		Variability	
Mean	70.63412	Std Deviation	9.33849
Median	69.60000	Variance	87.20744
Mode	74.60000	Range	36.70000
		Interquartile Range	12.10000

Tests for Location: Mu0=0

Test		-Statistic-	-----p Value------	
Student's t	t	202.6749	Pr > |t|	<.0001
Sign	M	359	Pr >= |M|	<.0001
Signed Rank	S	129060.5	Pr >= |S|	<.0001

On retrouve les infos principales + mode (valeur la plus fréquemment rencontrée) + range (écart entre mini et maxi) + InterQuartile Range (écart entre le 1er et le 3ème quartile)

- Mode : 74,6 kg
- Range : 36,7 kg
- IQR : 12,1 kg

PROC UNIVARIATE

- Illustration n°1

```
proc univariate data = TD_M2bio.data_exo2;
  var poids_inc ;
  run;
```

Sortie n°1

The UNIVARIATE Procedure
Variable: poids_inc

Moments

N	718	Sum Weights	718
Mean	70.6341226	Sum Observations	50715.3
Std Deviation	9.33849235	Variance	87.2074393
Skewness	0.61223655	Kurtosis	-0.4807254
Uncorrected SS	3644758.45	Corrected SS	62527.734
Coeff Variation	13.2209363	Std Error Mean	0.34850944

Basic Statistical Measures

Location		Variability	
Mean	70.63412	Std Deviation	9.33849
Median	69.60000	Variance	87.20744
Mode	74.60000	Range	36.70000
		Interquartile Range	12.10000

Tests for Location: Mu0=0

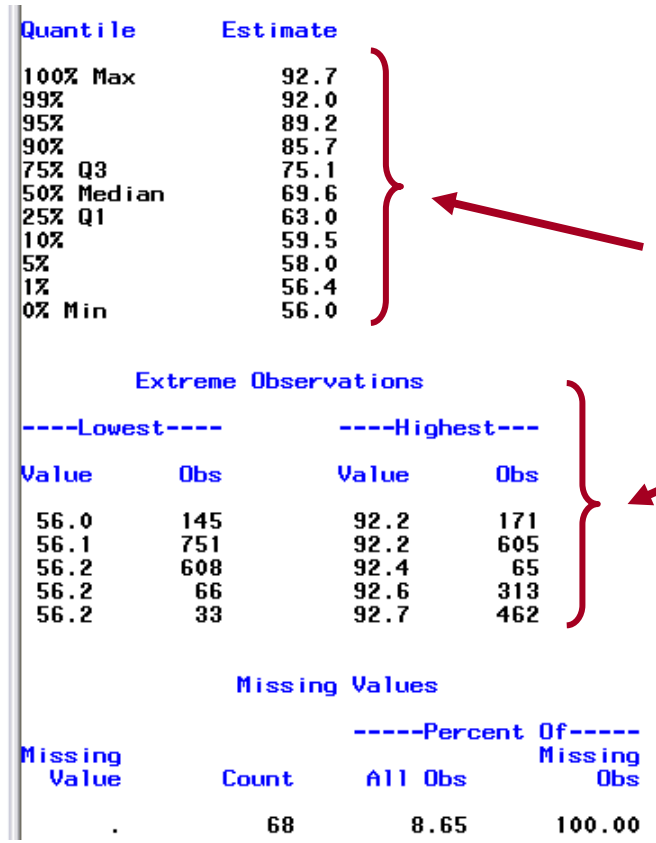
Test		-Statistic-	-----p Value------	
Student's t	t	202.6749	Pr > |t|	<.0001
Sign	M	359	Pr >= |M|	<.0001
Signed Rank	S	129060.5	Pr >= |S|	<.0001

Tests testant si la moyenne de *poids_inc* (ici, 70,6 kg) est significativement différente de 0 (ici, oui car $p_{Student} < 5\%$)

PROC UNIVARIATE

- Illustration n°1

```
proc univariate data = TD_M2bio.data_exo2;
  var poids_inc ;
  run;
```



Quantile	Estimate
100% Max	92.7
99%	92.0
95%	89.2
90%	85.7
75% Q3	75.1
50% Median	69.6
25% Q1	63.0
10%	59.5
5%	58.0
1%	56.4
0% Min	56.0

Extreme Observations

----Lowest----		----Highest---	
Value	Obs	Value	Obs
56.0	145	92.2	171
56.1	751	92.2	605
56.2	608	92.4	65
56.2	66	92.6	313
56.2	33	92.7	462

Missing Values

Missing Value	Count	-----Percent Of----- All Obs	Missing Obs
.	68	8.65	100.00

Sortie n°2

Percentiles de la distribution de *poids_inc*

Valeurs extrêmes de *poids_inc*

Valeurs manquantes de *poids_inc* (nombre, %)

PROC UNIVARIATE

- Illustration n°2

```
proc univariate data = TD_M2bio.data_exo2;
  class tabac ;
  var age_inc ;
  run;
```

Résultats pour les 380 sujets qui ne fument pas

```
The UNIVARIATE Procedure
Variable:  age_inc
tabac =            0

                          Moments

N                         380    Sum Weights              380
Mean                30.5263158    Sum Observations       11600
Std Deviation       4.99834718    Variance          24.9834745
Skewness            -0.0458387    Kurtosis          -1.1727909
Uncorrected SS          363574    Corrected SS      9468.73684
Coeff Variation     16.3738959    Std Error Mean     0.2564098


        Extreme Observations

----Lowest----        ----Highest---

Value      Obs        Value      Obs

   21      731           39      450
   21      693           39      492
   21      650           39      593
   21      273           40      348
   21      229           40      673
```

PROC UNIVARIATE

- Illustration n°2

```
proc univariate data = TD_M2bio.data_exo2;
  class tabac ;
  var age_inc ;
  run;
```

Résultats pour les 322 sujets qui fument

```
The UNIVARIATE Procedure
Variable:  age_inc
tabac =            1

                        Moments

N                         322    Sum Weights              322
Mean                30.2204969    Sum Observations        9731
Std Deviation       5.01752217    Variance          25.1755287
Skewness            0.03295112    Kurtosis          -1.0546314
Uncorrected SS          302157    Corrected SS      8081.34472
Coeff Variation     16.6030433    Std Error Mean    0.27961558


        Extreme Observations

----Lowest----        ----Highest---

Value      Obs        Value      Obs

   21      655           40      168
   21      500           40      267
   21      404           40      509
   21      322           40      613
   21      266           40      705
```

PROC UNIVARIATE

- Illustration n°3

```
proc univariate data = TD_M2bio.data_exo2;
  class tabac sexe ;
  var age_inc ;
  run;
```

Résultats pour les 230 hommes qui ne fument pas

```
The UNIVARIATE Procedure
Variable:  age_inc
tabac =              0
sexe =               1

                      Moments

N                         230    Sum Weights              230
Mean                30.1130435    Sum Observations        6926
Std Deviation       4.96496937    Variance          24.6509208
Skewness            0.02185644    Kurtosis          -1.1738326
Uncorrected SS          214208    Corrected SS      5645.06087
Coeff Variation     16.4877701    Std Error Mean    0.32738039
```

Résultats pour les 150 femmes qui ne fument pas

```
The UNIVARIATE Procedure
Variable:  age_inc
tabac =              0
sexe =               2

                      Moments

N                         150    Sum Weights              150
Mean                31.1[illegible]    Sum Observations        4674
Std Deviation       4.99943621    Variance          24.9943624
Skewness            -0.1561318    Kurtosis          -1.1423662
Uncorrected SS          149366    Corrected SS         3724.16
Coeff Variation     16.0444038    Std Error Mean    0.40820226
```

PROC UNIVARIATE

- Illustration n°3

```
proc univariate data = TD_M2bio.data_exo2;
  class tabac sexe ;
  var age_inc ;
  run;
```

Résultats pour les 123 hommes qui fument

```
The UNIVARIATE Procedure
Variable:  age_inc
tabac =            1
sexe =             1

                        Moments

N                    123    Sum Weights              123
Mean           29.6829268    Sum Observations        3651
Std Deviation  4.98574537    Variance          24.8576569
Skewness       0.14277323    Kurtosis          -0.9669191
Uncorrected SS     111405    Corrected SS      3032.63415
Coeff Variation 16.7966771   Std Error Mean    0.44954952
```

Résultats pour les 199 femmes qui fument

```
The UNIVARIATE Procedure
Variable:  age_inc
tabac =            1
sexe =             2

                        Moments

N                    199    Sum Weights              199
Mean           30.5527638    Sum Observations        6080
Std Deviation  5.02076294    Variance          25.2080605
Skewness       -0.0350079    Kurtosis          -1.0802917
Uncorrected SS     190752    Corrected SS      4991.19598
Coeff Variation 16.4330892   Std Error Mean    0.35591245
```

PROC UNIVARIATE

- Syntaxe pour dresser un histogramme

PROC UNIVARIATE Data = [*bibliothèque***.**]*nom_table* [**noprint**] **;**
[**Class** *variable_qual1 variable_qual2 … variable_qualk* **;**]
[**Var** *variable1 variable2 … variablek* **;**]
Histogram *variable1 variable2 … variablek* [**/ options**] **;**
Run ;

- La commande « **noprint** » évite d'encombrer la fenêtre Output de résultats de la PROC UNIVARIATE

- La commande « **Class** » permet dresser des histogrammes par modalité de variables qualitatives (ou par strate si > 1 variable qualitative)

- La commande « **Var** » n'est pas nécessaire pour dresser des histogrammes

PROC UNIVARIATE

- Syntaxe pour dresser un histogramme

PROC UNIVARIATE Data = [*bibliothèque***.**]*nom_table* [**noprint**] **;**
[**Class** *variable_qual1 variable_qual2 … variable_qualk* **;**]
[**Var** *variable1 variable2 … variablek* **;**]
Histogram *variable1 variable2 … variablek* [**/ options**] **;**
Run ;

- Options de « **Histogram** »

> Cfill = *color* : permet de spécifier la couleur de remplissage des histogrammes

> Cframe = *color* : permet de spécifier la couleur de fond des histogrammes

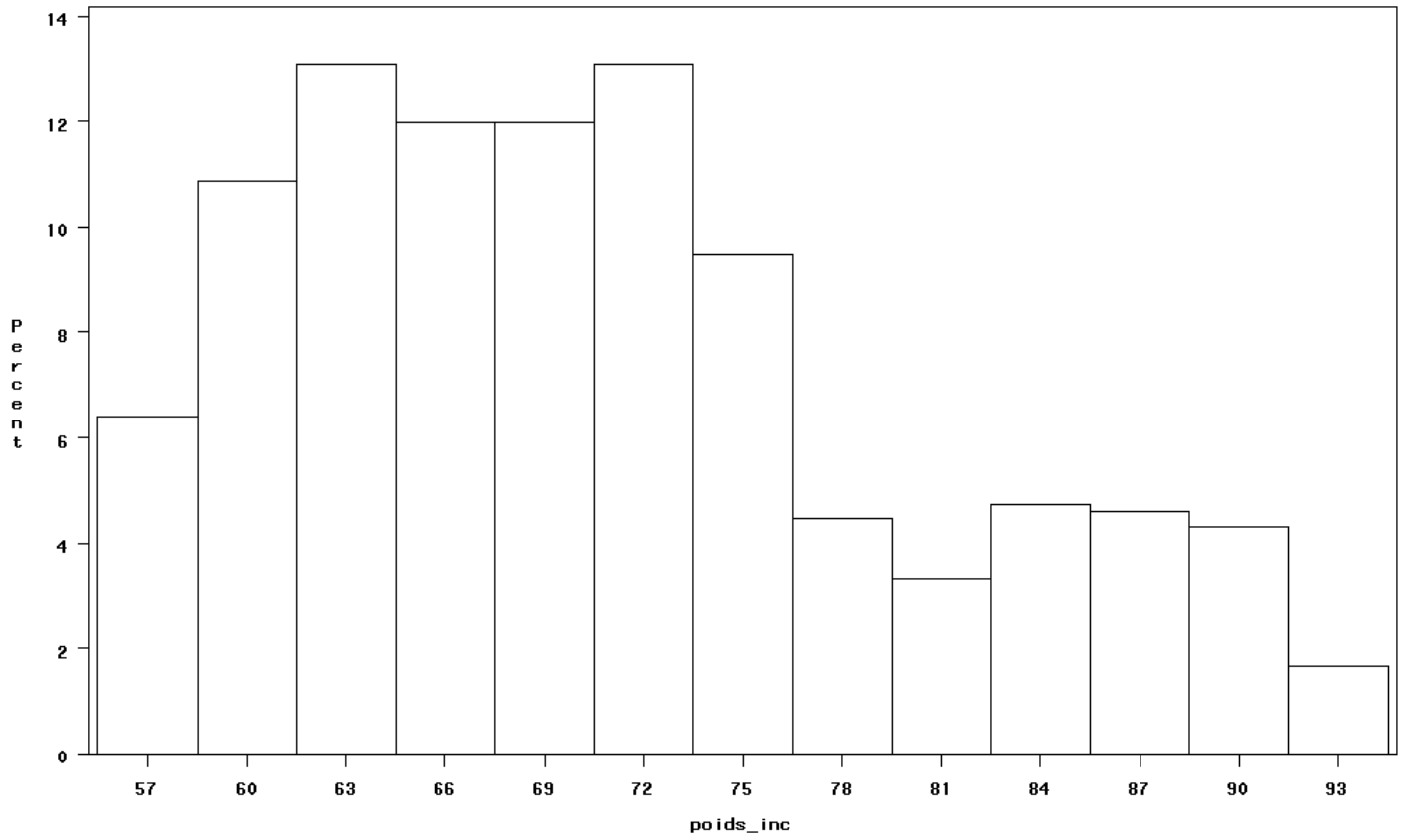
> Normal(*options*) : dresse la courbe de la loi normale de moyenne et d'écart-type ceux de la variable quantitative

Dans les options de « Normal », on peut entre autre spécifier la couleur de la courbe et sa largeur

PROC UNIVARIATE

- Illustration n°1

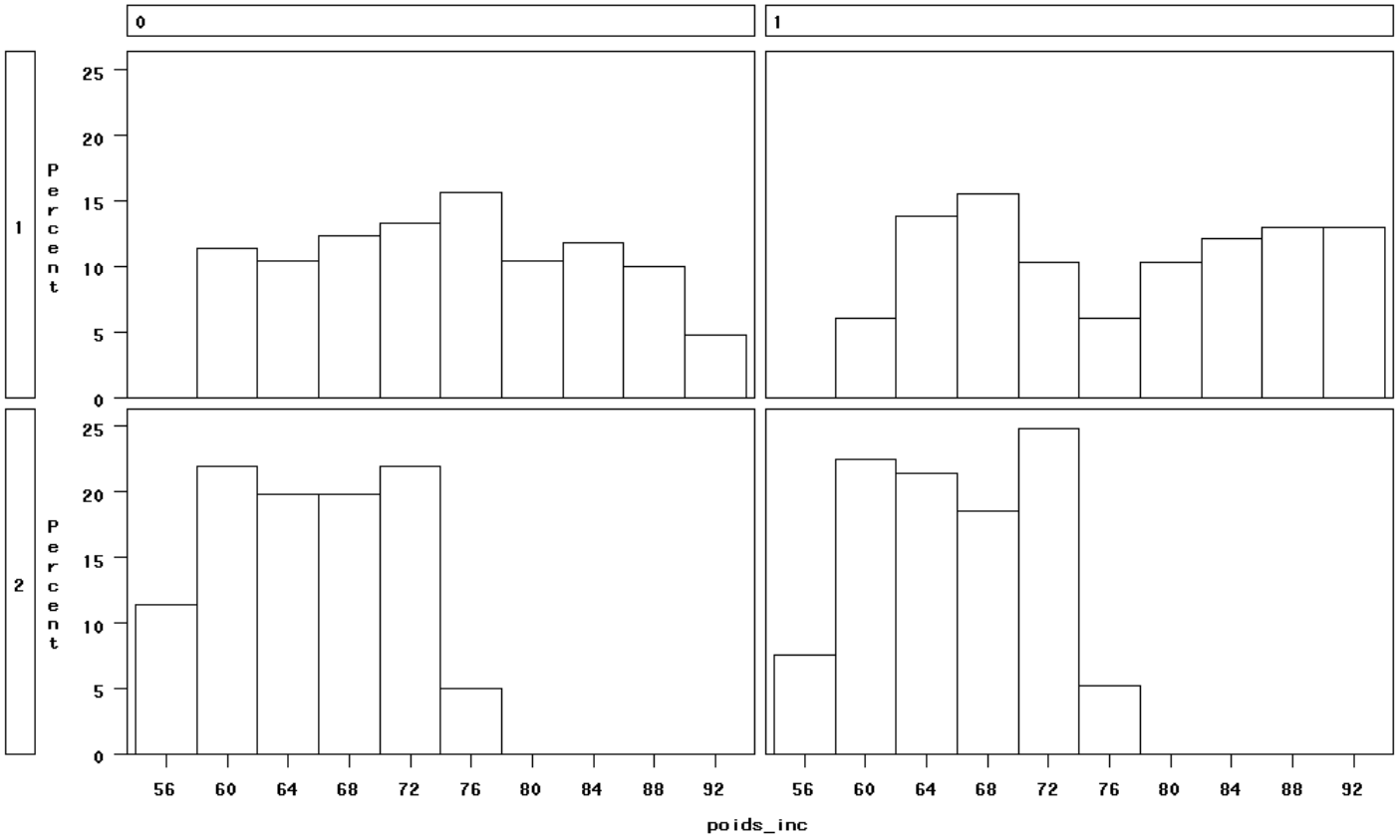
```
proc univariate data = TD_M2bio.data_exo2 noprint;
histogram poids_inc ;
run;
```



PROC UNIVARIATE

- Illustration n°2

```
proc univariate data = TD_M2bio.data_exo2 noprint;
class sexe tabac ;
histogram poids_inc ;
run;
```

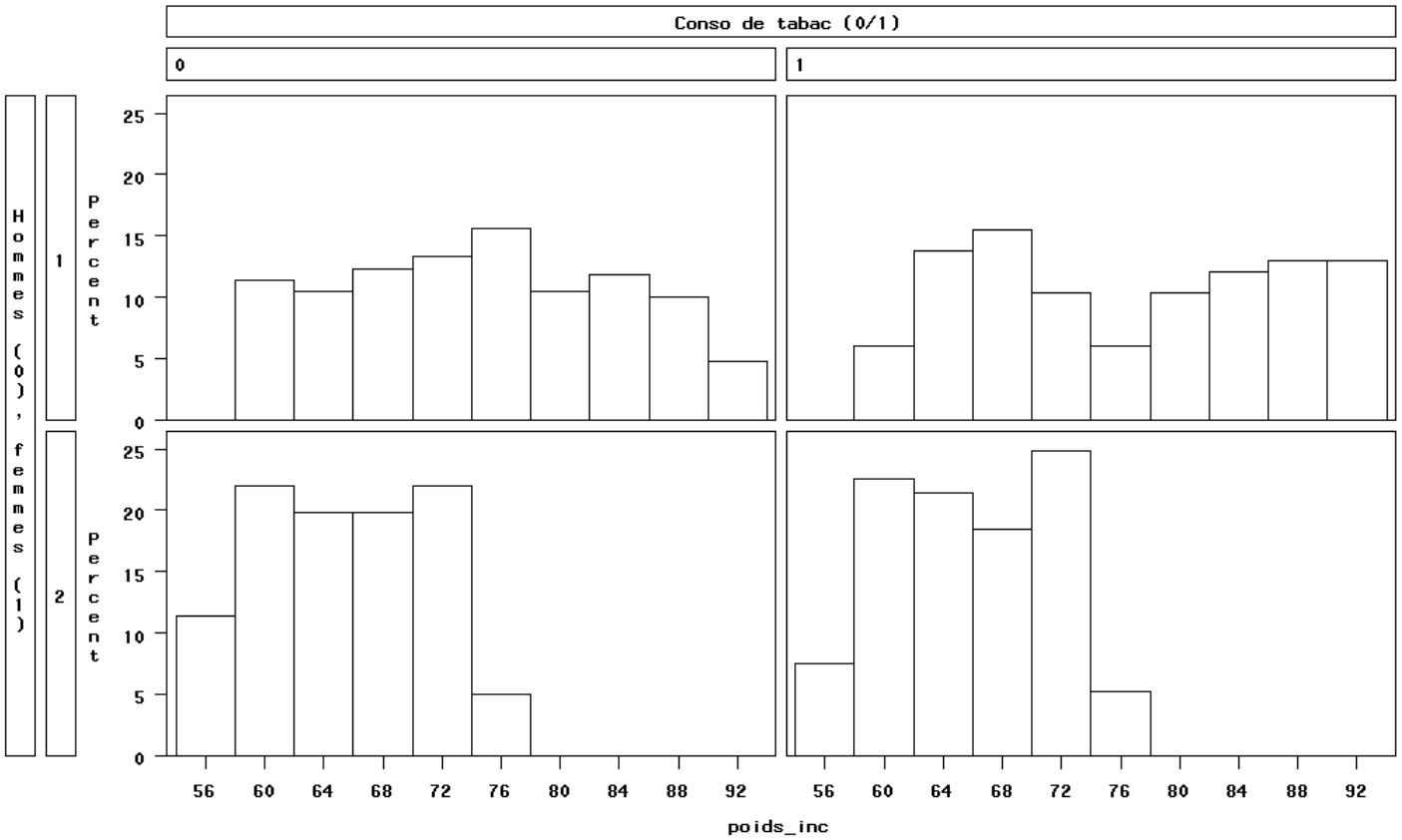


PROC UNIVARIATE

- Illustration n°3

```
data a;
set TD_M2bio.data_exo2 ;
label    tabac = "Conso de tabac (0/1)"
         sexe = "Hommes (0), femmes (1)";
run;

proc univariate data = a noprint;
class sexe tabac ;
histogram poids_inc ;
run;
```



PROC UNIVARIATE

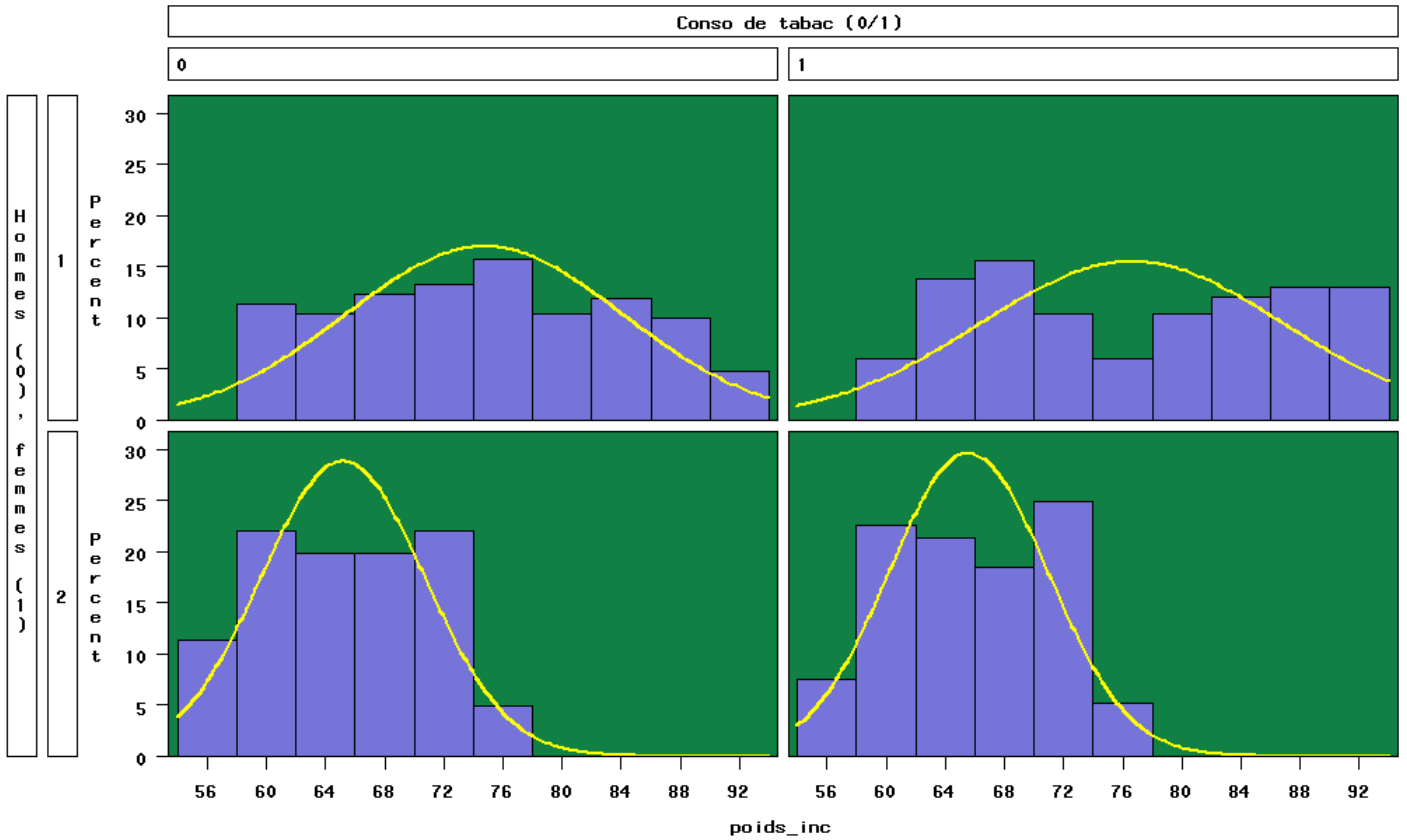
- Liste des couleurs disponible dans SAS

Hue	Abbreviation	Prefix	Abbreviation
red	R	pale	PA
pink	PK	brilliant	BI
olive	OL	light	LI
brown	BR	moderate	MO
orange	O	medium	ME
yellow	Y	strong	ST
yellow-green	LG	dark	DA
yellowish green	YG	deep	DE
green	G	vivid	VI
blue	B	very pale	VPA
purple	P	very light	VLI
violet	V	very dark	VDA
gray	GR	very deep	VDE
black	BL		
white	WH		

PROC UNIVARIATE

- Illustration n°4

```
proc univariate data = a noprint;
class sexe tabac ;
histogram poids_inc / cfill=VLIB cframe=VIG normal(color=Y w=2) ;
run;
```



PROC UNIVARIATE

- Illustration n°4

```
proc univariate data = a noprint;
  class sexe tabac ;
  histogram poids_inc / cfill=VLIB cframe=VIG normal(color=Y w=2) ;
  run;
```

Sortie dans la fenêtre « Output »

```
The UNIVARIATE Procedure
Fitted Distribution for poids_inc
sexe = 1
tabac = 0

Parameters for Normal Distribution

Parameter   Symbol   Estimate

Mean        Mu       74.77773
Std Dev     Sigma     9.41501

      Goodness-of-Fit Tests for Normal Distribution

Test                   ---Statistic----    -----p Value-----

Kolmogorov-Smirnov     D      0.06531681   Pr > D       0.026
Cramer-von Mises       W-Sq   0.20143770   Pr > W-Sq   <0.005
Anderson-Darling       A-Sq   1.75410678   Pr > A-Sq   <0.005
```

- Tests d'écart à la normalité
- Si le test est significatif, la distribution de la variable quantitative n'est significativement pas normale

PROC UNIVARIATE

- Syntaxe pour dresser un graphe QQ Plot

  Le graphe QQ plot (pour Quantile-Quantile plot) permet de visualiser l'écart à une distribution prédéfinie de la variable quantitative

- Syntaxe pour dresser un graphe QQ Plot pour comparer à la loi normale

  PROC UNIVARIATE Data = [*bibliothèque***.**]*nom_table* [**noprint**] **;**

  [**Class** *variable_qual1 variable_qual2 … variable_qualk* **;**]

  [**Var** *variable1 variable2 … variablek* **;**]

  QQplot *variable1 variable2 … variablek* **/ normal(mu=est sigma=est** ***options*****)** [**options**] **;**

  Run ;

PROC UNIVARIATE

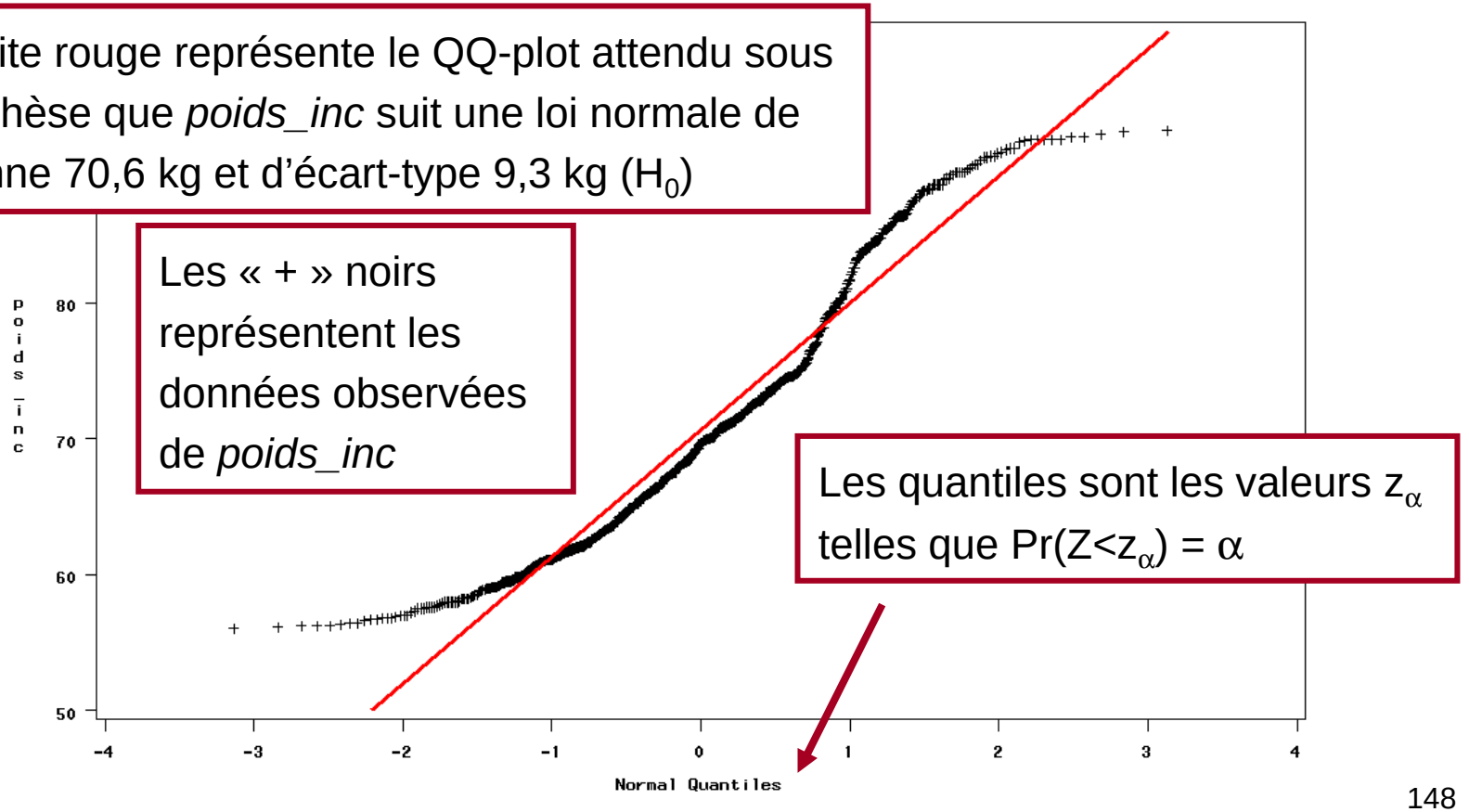
- Illustration n°1 – Interprétation d'un graph QQ-plot

```
proc univariate data = a noprint;
qqplot poids_inc / normal(color=RED mu=est sigma=est w=2) ;
run;
```

La droite rouge représente le QQ-plot attendu sous l'hypothèse que *poids_inc* suit une loi normale de moyenne 70,6 kg et d'écart-type 9,3 kg (H_0)

Les « + » noirs représentent les données observées de *poids_inc*

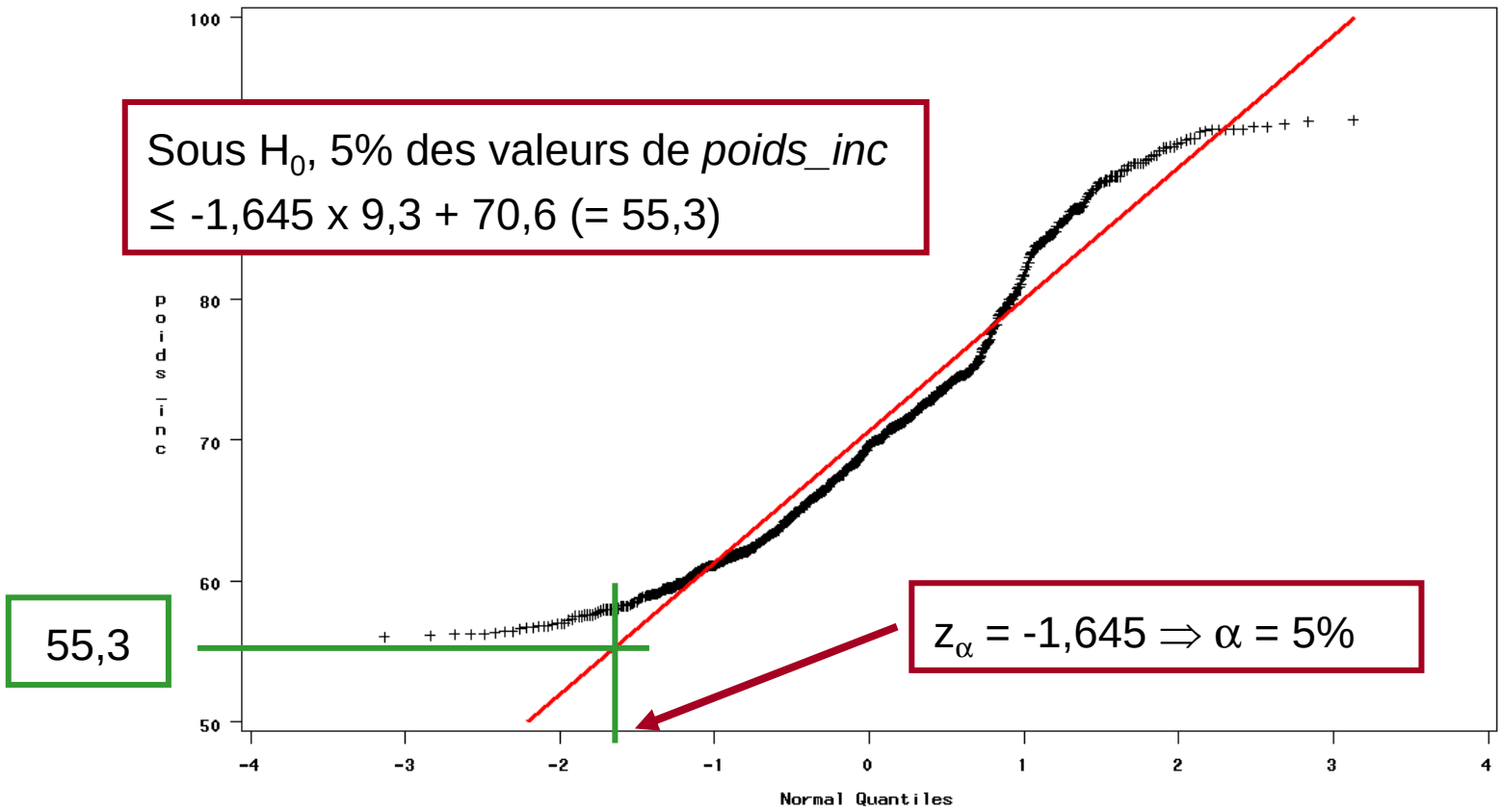
Les quantiles sont les valeurs z_α telles que $Pr(Z<z_\alpha) = \alpha$



PROC UNIVARIATE

- Illustration n°1 – Interprétation d'un graph QQ-plot

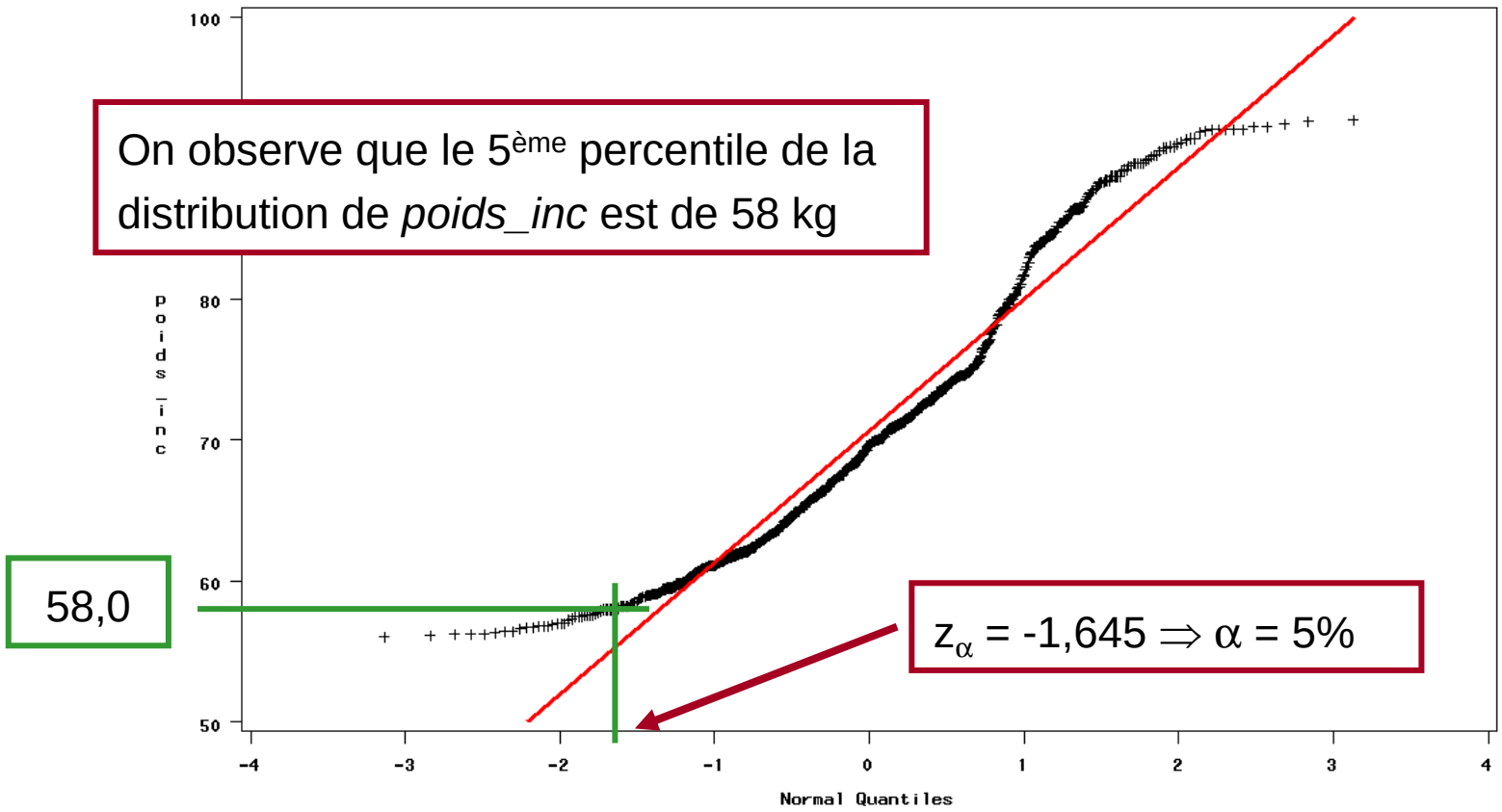
```
proc univariate data = a noprint;
qqplot poids_inc / normal(color=RED mu=est sigma=est w=2) ;
run;
```



PROC UNIVARIATE

- Illustration n°1 – Interprétation d'un graph QQ-plot

```
proc univariate data = a noprint;
qqplot poids_inc / normal(color=RED mu=est sigma=est w=2) ;
run;
```



On observe que le 5ème percentile de la distribution de *poids_inc* est de 58 kg

58,0

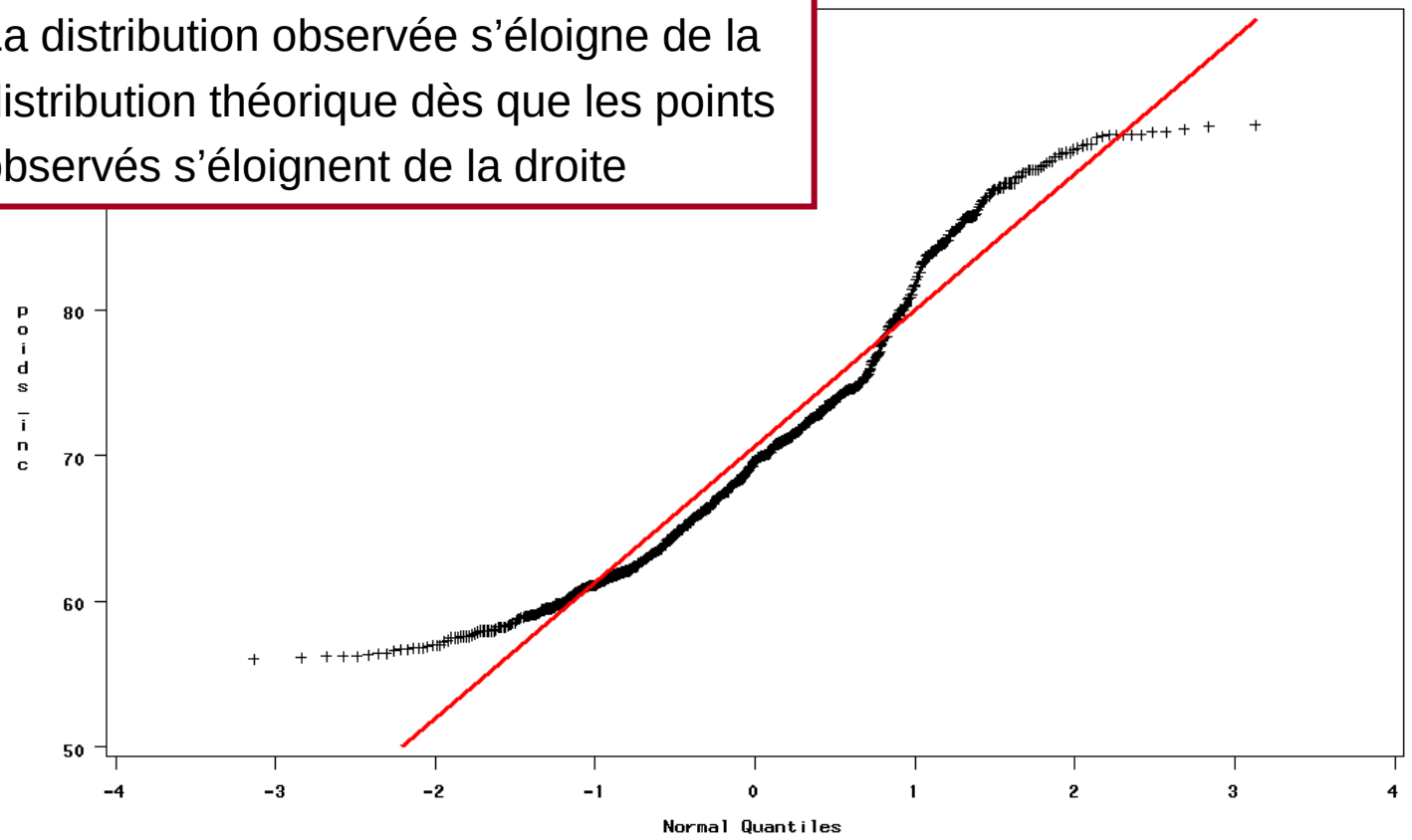
$z_\alpha = -1{,}645 \Rightarrow \alpha = 5\%$

PROC UNIVARIATE

- Illustration n°1 – Interprétation d'un graph QQ-plot

```
proc univariate data = a noprint;
qqplot poids_inc / normal(color=RED mu=est sigma=est w=2) ;
run;
```

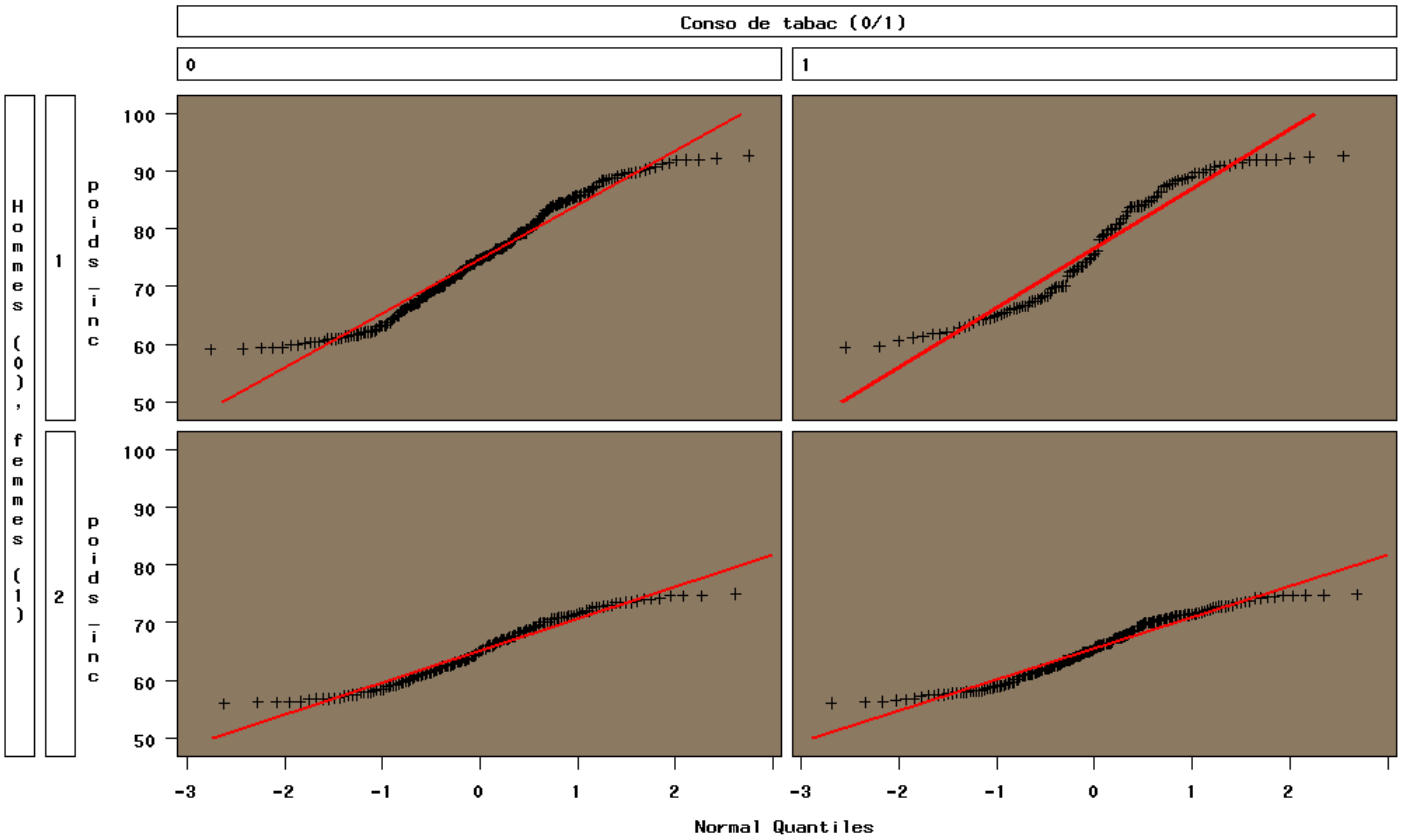
La distribution observée s'éloigne de la distribution théorique dès que les points observés s'éloignent de la droite



PROC UNIVARIATE

- Illustration n°2

```
proc univariate data = a noprint;
class sexe tabac ;
qqplot poids_inc / cframe=LIBR normal(color=RED mu=est sigma=est w=2) ;
run;
```



PROC MEANS

- Descriptif de la procédure

  - La procédure PROC MEANS est une procédure qui donne des informations de bases sur la distribution d'une variable quantitative

  - On peut personnaliser les sorties de PROC MEANS, en spécifiant les mots-clés correspondant aux informations que l'on souhaite

  - Beaucoup plus lisible que PROC UNIVARIATE !

PROC MEANS

- Syntaxe pour des descriptions simples

PROC MEANS Data = [*bibliothèque***.**]*nom_table* [***maxdec=****chiffre* ***mots-clés***] **;**

[**Var** *variable1 variable2 … variablek* **;**]

Run ;

- Si la commande « **Var** *variable1* … *variablek* **;** » n'est pas écrite, SAS va exécuter cette procédure sur toutes les variables de la table, y compris sur les variables binaires et qualitatives !

- La commande « **Maxdec=***chiffre* » permet de ne faire apparaître que quelques chiffres après la virgule

PROC MEANS

- Syntaxe pour des descriptions simples

PROC MEANS Data = [*bibliothèque***.**]*nom_table* [***maxdec=****chiffre* ***mots-clés***]**;**

[**Var** *variable1 variable2 … variablek* **;**]

Run ;

- Si aucun mot-clé n'est spécifié, la procédure fournira (par défaut) :
  - > Le nombre de données non manquantes
  - > La moyenne
  - > L'écart-type (SD)
  - > Les minimum et maximum

PROC MEANS

- Syntaxe pour des descriptions simples

PROC MEANS Data = [*bibliothèque***.**]*nom_table* [***maxdec=****chiffre* ***mots-clés***]**;**

[**Var** *variable1 variable2 … variablek* **;**]

Run ;

- Dès qu'un mot-clé est spécifié, il faut tous les spécifier (les statistiques de ceux par défaut ne seront plus automatiquement fournies)

- Liste des mots-clés les plus intéressants
  - **mean** : moyenne ; **median** : médiane
  - **N** : nombre de données non manquantes
  - **Nmiss** : nombre de données manquantes
  - **STD** : écart-type (SD) ; **STDerr** : écart-type de la moyenne (SE)
  - **min**, **max** : minium et maximum
  - **p25**, **p75** : 1er et 3ème quartiles

PROC MEANS

- Syntaxe pour des descriptions croisées entre une variable quantitative et une ou plusieurs variables qualitatives

PROC MEANS Data = [*bibliothèque***.**]*nom_table* [***maxdec=****chiffre* ***mots-clés***] **;**

Class *variable_qual1 variable_qual2 … variable_qualk* **;**

[**Var** *variable1 variable2 … variablek* **;**]

Run ;

- De façon identique à PROC UNIVARIATE, s'il y a plusieurs variables qualitatives pour la commande « Class », les résultats seront fournis pour toutes les strates formées par le k-uplet des variables qualitatives

PROC MEANS

- Illustration n°1

```
proc means data = TD_M2bio.data_exo2 ;
  class sexe tabac ;
  var poids_inc ;
  run;
```

The MEANS Procedure

Analysis Variable : poids_inc

sexe	tabac	N Obs	N	Mean	Std Dev	Minimum	Maximum
1	0	230	211	74.7777251	9.4150096	59.1000000	92.7000000
	1	123	116	76.6525862	10.3232996	59.5000000	92.6000000
2	0	150	141	65.1921986	5.5402691	56.0000000	75.0000000
	1	199	173	65.5774566	5.3921789	56.1000000	74.9000000

N = nombre d'observations non manquantes

N obs = nombre d'observations total
= nombre de données manquantes + non manquantes

PROC MEANS

- Illustration n°2

```
proc means data = TD_M2bio.data_exo2 nmiss min p25 median p75 max maxdec=2 ;
  class sexe tabac ;
  var poids_inc ;
  run;
```

The MEANS Procedure

Analysis Variable : poids_inc

sexe	tabac	N Obs	N Miss	Minimum	25th Pctl	Median	75th Pctl	Maximum
1	0	230	19	59.10	67.00	74.80	83.10	92.70
	1	123	7	59.50	67.00	75.50	86.40	92.60
2	0	150	9	56.00	60.60	65.20	70.00	75.00
	1	199	26	56.10	61.10	65.40	70.50	74.90

N miss = nombre d'observations manquantes

PROC GPLOT

- Descriptif de la procédure

  - La procédure PROC GPLOT est une procédure qui trace des nuages de points, avec la possibilité de relier ces points

  - Les deux variables en abscisses et en ordonnées doivent être quantitatives

  - La procédure est puissante, et seule sa syntaxe de base sera vue ici (le nombre d'options graphiques est trop important… !)

PROC GPLOT

- Syntaxe pour un nuage de point unique (1 variable X croisée avec 1 variable Y)

PROC GPLOT Data = [*bibliothèque***.**]*nom_table* **;**

Plot *varY* ***** *varX* **;**

Run ;

- Syntaxe pour un nuage de point multiple (≥ 1 variable(s) X croisée avec ≥ 1 variable(s) Y)

PROC GPLOT Data = [*bibliothèque***.**]*nom_table* **;**

Plot (*varY1 varY1 .. varYk***) * (***varX1 varX1 .. varXp***) ;**

Run ;

PROC GPLOT

- Syntaxe pour un nuage de point unique (1 variable X croisée avec 1 variable Y) selon les modalités d'une variable qualitative

PROC GPLOT Data = [*bibliothèque*.]*nom_table* **;**

Plot *varY* ***** *varX* **=** *var_qual* **;**

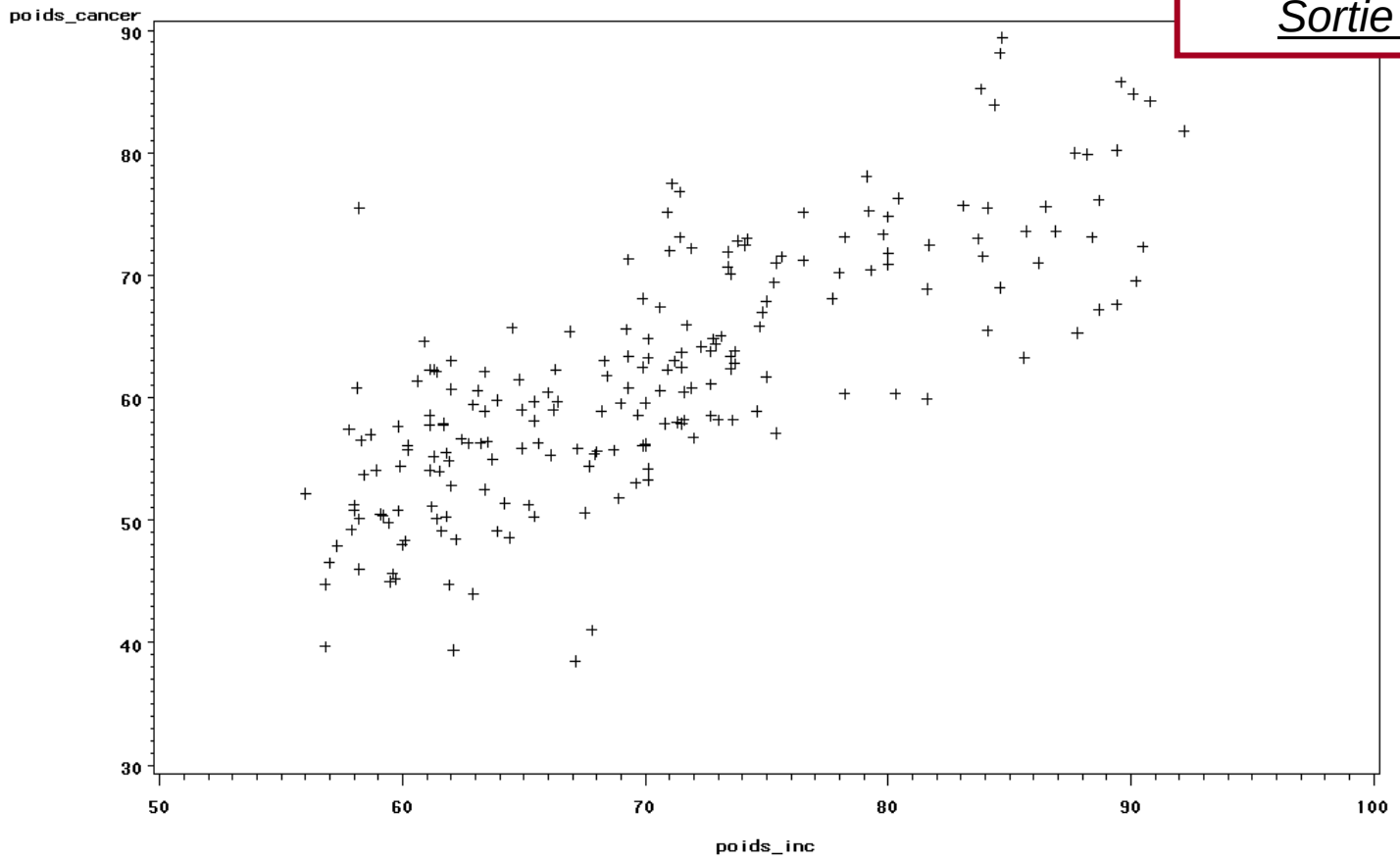
Run ;

PROC GPLOT

- Illustration

```
proc gplot data = TD_M2bio.data_exo2 ;
plot poids_cancer * (poids_inc date_cancer) ;
run ;
```

Sortie n°1

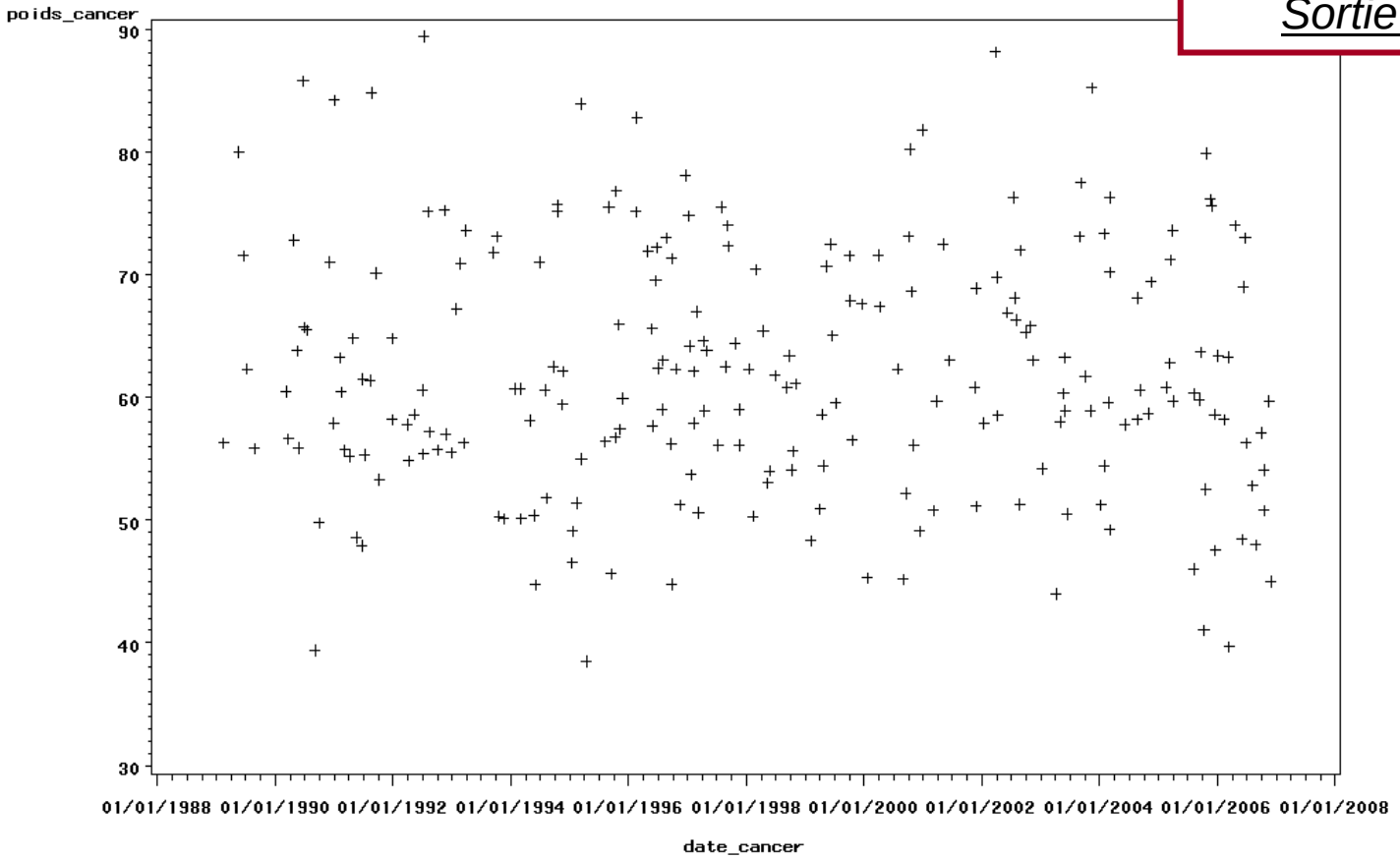


PROC GPLOT

- Illustration

```
proc gplot data = TD_M2bio.data_exo2 ;
plot poids_cancer * (poids_inc date_cancer) ;
run ;
```

Sortie n°2



PROC BOXPLOT

- Descriptif de la procédure

  - La procédure PROC BOXPLOT est une procédure qui trace des « boxplot » ou « boites à moustache »

  - La variable qualitative se trouve en abscisse ; la variable quantitative se trouve en ordonnée

- Syntaxe pour une boite à moustache unique

PROC SORT Data = [*bibliothèque***.**]*nom_table* **;**
By *var_qual* **;**
Run ;

PROC BOXPLOT Data = [*bibliothèque***.**]*nom_table* **;**
Plot *var_quant* ***** *var_qual* **;**
Run ;

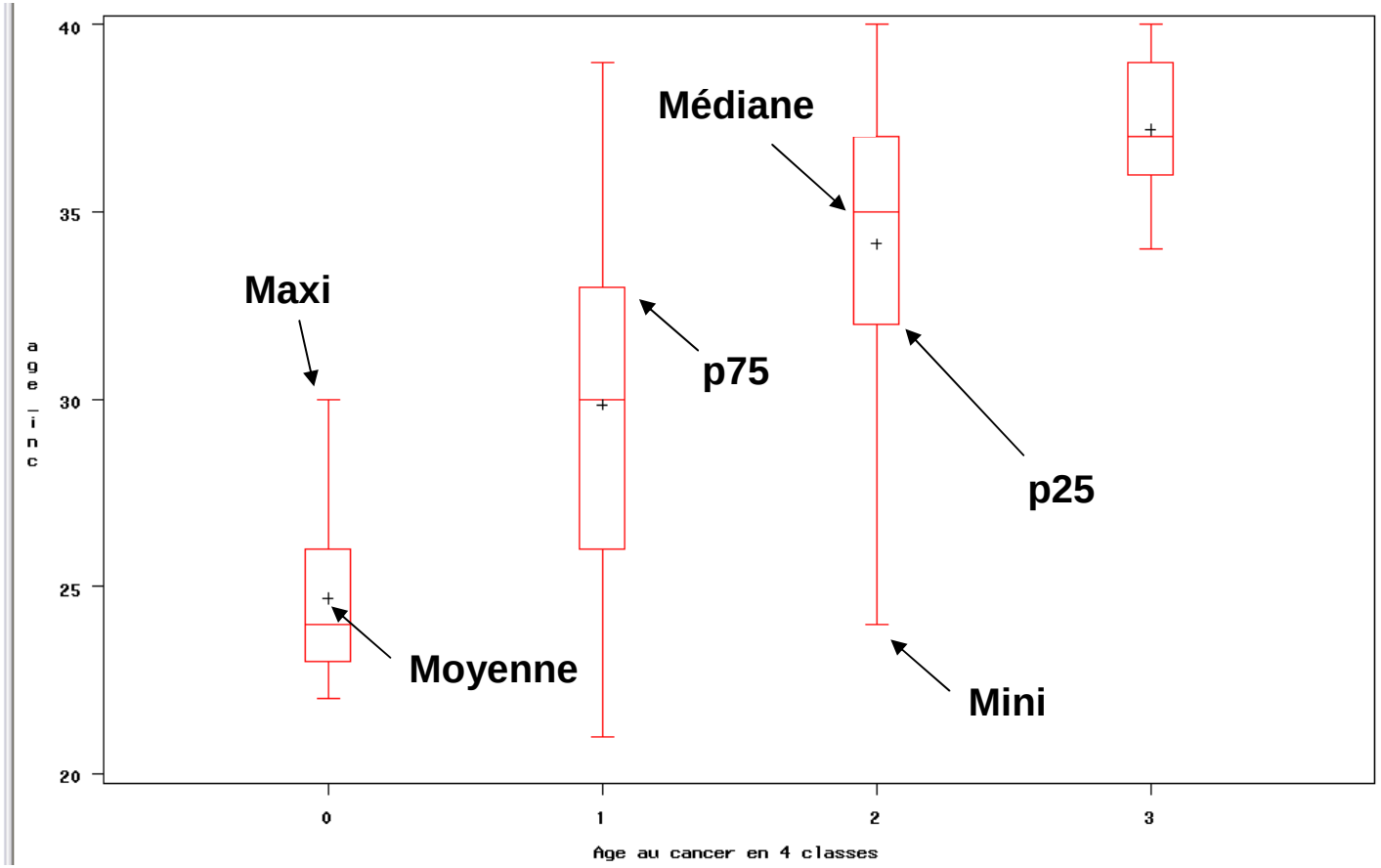
Un tri par la variable qualitative est nécessaire avant d'effectuer un boxplot

PROC BOXPLOT

- Illustration et interprétation

```
proc sort data = TD_M2bio.data_exo2 ;
by age_K_cl;
run;

proc boxplot data = TD_M2bio.data_exo2 ;
plot age_inc * age_K_cl ;
run;
```



PROC TTEST

- Descriptif de la procédure

  La procédure PROC TTEST est une procédure qui teste si deux moyennes sont significativement différentes, en utilisant le test de Student

- Conditions d'application du test

  - La variable quantitative doit suivre une loi normale

  - Si l'effectif dans au moins un des deux groupes est < 30, il faut vérifier que les variances dans les groupes ne sont pas significativement différentes

    Si elles sont significativement différentes, SAS propose un test de Student approché

PROC TTEST

- Syntaxe

PROC TTEST Data = [*bibliothèque***.**]*nom_table* **;**

Class *var_binaire* **;**

Var *var_quantitative* **;**

Run ;

PROC TTEST

- Illustration n°1

```
proc ttest data = TD_M2bio.data_exo2 ;
  class sexe ;
  var age_inc ;
  run;
```

Moyenne d'âge chez les 400 hommes : 29,97 ans

Moyenne d'âge chez les 386 femmes : 30,94 ans

Différence de -0,97 ans

The TTEST Procedure

Statistics

Variable	sexe	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
age_inc	1	400	29.479	29.97	30.461	4.6666	4.9901	5.3622	0.2495	21	40
age_inc	2	386	30.442	30.943	31.444	4.6733	5.0031	5.3833	0.2546	21	40
age_inc	Diff (1-2)		-1.673	-0.973	-0.273	4.761	4.9965	5.2567	0.3565		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
age_inc	Pooled	Equal	784	-2.73	0.0065
age_inc	Satterthwaite	Unequal	783	-2.73	0.0065

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
age_inc	Folded F	385	399	1.01	0.9587

PROC TTEST

- Illustration n°1

```
proc ttest data = TD_M2bio.data_exo2 ;
  class sexe ;
  var age_inc ;
  run;
```

Intervalle de confiance à 95% de la différence d'âge :

Diff = -0,97 ans [-1,67 ; -0,27]

The TTEST Procedure

Statistics

Variable	sexe	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
age_inc	1	400	29.479	29.97	30.461	4.6666	4.9901	5.3622	0.2495	21	40
age_inc	2	386	30.442	30.943	31.444	4.6733	5.0031	5.3833	0.2546	21	40
age_inc	Diff (1-2)		-1.673	-0.973	-0.273	4.761	4.9965	5.2567	0.3565		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
age_inc	Pooled	Equal	784	-2.73	0.0065
age_inc	Satterthwaite	Unequal	783	-2.73	0.0065

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
age_inc	Folded F	385	399	1.01	0.9587

PROC TTEST

- Illustration n°1

```
proc ttest data = TD_M2bio.data_exo2 ;
  class sexe ;
  var age_inc ;
  run;
```

Effectifs dans chaque groupe > 30 ⇒ « Method pooled » ⇒ p = 0,0065 < 5% ⇒ différence d'âge significative entre les hommes et les femmes

The TTEST Procedure

Statistics

Variable	sexe	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
age_inc	1	400	29.479	29.97	30.461	4.6666	4.9901	5.3622	0.2495	21	40
age_inc	2	386	30.442	30.943	31.444	4.6733	5.0031	5.3833	0.2546	21	40
age_inc	Diff (1-2)		-1.673	-0.973	-0.273	4.761	4.9965	5.2567	0.3565		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
age_inc	Pooled	Equal	784	-2.73	0.0065
age_inc	Satterthwaite	Unequal	783	-2.73	0.0065

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
age_inc	Folded F	385	399	1.01	0.9587

PROC TTEST

- Illustration n°2

```
proc ttest data = TD_M2bio.data_exo2 ;
  where age_K_cl = 1 and tabac = 1;
  class sexe ;
  var age_inc ;
  run;
```

Parmi les sujets avec cancer, âgés ≤ 60 ans, et fumeurs, la différence d'âge à l'inclusion entre les 20 hommes et les 29 femmes : +0,28 ans [-2,39 ; 2,94]

The TTEST Procedure

Statistics

Variable	sexe	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
age_inc	1	20	27.653	29.9	32.147	3.6505	4.8002	7.0111	1.0734	21	38
age_inc	2	29	27.952	29.621	31.289	3.4813	4.3868	5.9329	0.8146	22	39
age_inc	Diff (1-2)		-2.386	0.2793	2.9448	3.7948	4.5584	5.7098	1.325		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
age_inc	Pooled	Equal	47	0.21	0.8339
age_inc	Satterthwaite	Unequal	38.5	0.21	0.8369

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
age_inc	Folded F	19	28	1.20	0.6505

PROC TTEST

- Illustration n°2

```
proc ttest data = TD_M2bio.data_exo2 ;
  where age_K_cl = 1 and tabac = 1;
  class sexe ;
  var age_inc ;
  run;
```

Les effectifs étant < 30, il faut vérifier si les variance ne sont pas significativement inégales

Ici, le test donne *p* = 0,65

The TTEST Procedure

Statistics

Variable	sexe	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
age_inc	1	20	27.653	29.9	32.147	3.6505	4.8002	7.0111	1.0734	21	38
age_inc	2	29	27.952	29.621	31.289	3.4813	4.3868	5.9329	0.8146	22	39
age_inc	Diff (1-2)		-2.386	0.2793	2.9448	3.7948	4.5584	5.7098	1.325		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
age_inc	Pooled	Equal	47	0.21	0.8339
age_inc	Satterthwaite	Unequal	38.5	0.21	0.8369

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
age_inc	Folded F	19	28	1.20	0.6505

PROC TTEST

- Illustration n°2

```
proc ttest data = TD_M2bio.data_exo2 ;
  where age_K_cl = 1 and tabac = 1;
  class sexe ;
  var age_inc ;
  run;
```

Donc, on peut encore lire le résultat du test de Student à la ligne « Method Pooled »

$\Rightarrow p = 0{,}83 > 5\%$ (différence non significative)

The TTEST Procedure

Statistics

Variable	sexe	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
age_inc	1	20	27.653	29.9	32.147	3.6505	4.8002	7.0111	1.0734	21	38
age_inc	2	29	27.952	29.621	31.289	3.4813	4.3868	5.9329	0.8146	22	39
age_inc	Diff (1-2)		-2.386	0.2793	2.9448	3.7948	4.5584	5.7098	1.325		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
age_inc	Pooled	Equal	47	0.21	0.8339
age_inc	Satterthwaite	Unequal	38.5	0.21	0.8363

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
age_inc	Folded F	19	28	1.20	0.6505

PROC TTEST

- Illustration n°3

```
proc ttest data = TD_M2bio.data_exo2 ;
  where age_K_cl = 1 and tabac = 1;
  class sexe ;
  var poids_inc ;
  run;
```

Parmi les sujets avec cancer, âgés ≤ 60 ans, et fumeurs, la différence de poids à l'inclusion entre les 20 hommes et les 27 femmes : +9,6 kg [5,1 ; 14,0]

The TTEST Procedure

Statistics

Variable	sexe	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
poids_inc	1	20	70.982	75.755	80.528	7.7558	10.198	14.895	2.2804	61.8	90.8
poids_inc	2	27	64.311	66.193	68.074	3.7452	4.7557	6.5174	0.9152	58.2	73.5
poids_inc	Diff (1-2)		5.077	9.5624	14.048	6.2611	7.5486	9.5076	2.227		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
poids_inc	Pooled	Equal	45	4.29	<.0001
poids_inc	Satterthwaite	Unequal	25.1	3.89	0.0006

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
poids_inc	Folded F	19	26	4.60	0.0004

PROC TTEST

- Illustration n°3

```
proc ttest data = TD_M2bio.data_exo2 ;
  where age_K_cl = 1 and tabac = 1;
  class sexe ;
  var poids_inc ;
  run;
```

Les effectifs étant < 30, il faut vérifier si les variance ne sont pas significativement inégales

Ici, le test donne *p* < 5% (variances significativement inégales)

The TTEST Procedure

Statistics

Variable	sexe	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
poids_inc	1	20	70.982	75.755	80.528	7.7558	10.198	14.895	2.2804	61.8	90.8
poids_inc	2	27	64.311	66.193	68.074	3.7452	4.7557	6.5174	0.9152	58.2	73.5
poids_inc	Diff (1-2)		5.077	9.562	14.048	6.2611	7.5486	9.5076	2.227		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
poids_inc	Pooled	Equal	45	4.29	<.0001
poids_inc	Satterthwaite	Unequal	25.1	3.89	0.0006

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
poids_inc	Folded F	19	26	4.60	0.0004

PROC TTEST

- Illustration n°3

```
proc ttest data = TD_M2bio.data_exo2 ;
  where age_K_cl = 1 and tabac = 1;
  class sexe ;
  var poids_inc ;
  run;
```

Donc, on doit lire le résultat du test de Student à la ligne « Method Satterhwaite »

$\Rightarrow p = 0{,}0006 < 5\%$ (différence significative)

The TTEST Procedure

Statistics

Variable	sexe	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
poids_inc	1	20	70.982	75.755	80.528	7.7558	10.198	14.895	2.2804	61.8	90.8
poids_inc	2	27	64.311	66.193	68.074	3.7452	4.7557	6.5174	0.9152	58.2	73.5
poids_inc	Diff (1-2)		5.077	9.562	14.048	6.2611	7.5486	9.5076	2.227		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
poids_inc	Pooled	Equal	45	4.29	<.0001
poids_inc	Satterthwaite	Unequal	25.1	3.89	0.0006

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
poids_inc	Folded F	19	26	4.60	0.0004

PROC ANOVA

- Descriptif de la procédure

  - La procédure PROC ANOVA est une procédure qui teste si > 2 moyennes sont significativement différentes, par le test d'analyse de variance

  - Principe : on teste si la variance entre les groupes à comparer est supérieure à la variance au sein de chaque groupe

  - On peut comparer plusieurs moyennes après prise en compte d'autres expositions (analyse de variance multivariée)

- Conditions d'application du test

  - La variable quantitative doit suivre une loi normale

  - La variance doit être égale dans chacun des groupes à comparer

PROC ANOVA

- Syntaxe pour une ANOVA univariée

PROC ANOVA Data = [*bibliothèque*.]*nom_table* **;**
Class *var_qual* **;**
Model *var_quantitative* **=** *var_qual* **;**
Run ;

Problème de la procédure : elle ne fournit pas les moyennes que l'on teste

⇒ Il faut au préalable avoir exécuté une PROC MEANS (ou PROC UNIVARIATE) pour afficher les moyennes qui vont être comparées et testées

PROC ANOVA

- Illustration

```
proc means data = TD_M2bio.data_exo2 ;
class age_K_cl ;
var poids_inc ;
run;

proc anova data = TD_M2bio.data_exo2 ;
class age_K_cl ;
model poids_inc = age_K_cl ;
run;
```

Sortie n°1

The MEANS Procedure

Analysis Variable : poids_inc

Age au cancer en 4 classes	N Obs	N	Mean	Std Dev	Minimum	Maximum
0	13	13	65.6615385	6.7895187	58.7000000	83.9000000
1	105	98	70.3000000	9.0794432	56.0000000	90.8000000
2	86	76	71.0500000	9.2109428	56.8000000	92.2000000
3	25	21	70.2714286	10.2184217	56.8000000	86.5000000

Les 4 moyennes à comparer sont :

- 65,7 kg pour les sujets avec cancer ≤ 60 ans
- 70,3 kg pour les sujets avec cancer 60-70 ans
- 71,0 kg pour les sujets avec cancer 70-80 ans
- 70,3 kg pour les sujets avec cancer > 80 ans

PROC ANOVA

- Illustration

```
proc means data = TD_M2bio.data_exo2 ;
class age_K_cl ;
var poids_inc ;
run;

proc anova data = TD_M2bio.data_exo2 ;
class age_K_cl ;
model poids_inc = age_K_cl ;
run;
```

Sortie n°2

The ANOVA Procedure

Class Level Information

Class	Levels	Values
age_K_cl	4	0 1 2 3

Number of Observations Read	786
Number of Observations Used	208

The ANOVA Procedure

Dependent Variable: poids_inc

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	322.39325	107.46442	1.29	0.2791
Error	204	17000.92363	83.33786		
Corrected Total	207	17323.31687			

R-Square	Coeff Var	Root MSE	poids_inc Mean
0.018610	12.98918	9.128957	70.28125

Source	DF	Anova SS	Mean Square	F Value	Pr > F
age_K_cl	3	322.3932486	107.4644162	1.29	0.2791

Test de Fisher non significatif (p = 0,28)

⇒ Les 4 moyennes ne sont donc pas significativement différentes

PROC CORR

- Descriptif de la procédure
  - La procédure PROC CORR est une procédure qui teste si 2 variables quantitatives sont significativement associées
  - La procédure calcule un coefficient de corrélation et elle le teste à 0 (ce coefficient est-il significativement différent de 0 ?)
  - On fait cependant l'hypothèse que si les variables sont associées, elles le sont plus ou moins linéairement : le test du coefficient de corrélation teste la partie linéaire de l'association

- Conditions d'application du test
  - Chacune des deux variables, à niveau égal de l'autre, doit suivre une loi normale
  - Les 2 variables doivent être indépendantes

PROC CORR

- Syntaxe

PROC CORR Data = [*bibliothèque***.**]*nom_table* **;**

Var *var_quantitative1 var_quantitative2 … var_quantitativek* **;**

Run ;

La procédure va calculer les coefficients de corrélation entre toutes les variables listées dans la commande « **Var** »

PROC CORR

- Illustration

```
proc corr data = TD_M2bio.data_exo2 ;
  var age_inc age_cancer poids_inc poids_cancer ;
  run ;
```

The CORR Procedure

4 Variables: age_inc age_cancer poids_inc poid

Simple Statistics

Variable	N	Mean	Std Dev	
age_inc	786	30.44784	5.01696	23
age_cancer	229	70.74236	7.13522	16
poids_inc	718	70.63412	9.33849	50
poids_cancer	229	62.00742	10.05790	14

Pearson Correlation Coefficients
Prob > |r| under H0: Rho=0
Number of Observations

	age_inc	age_cancer	poids_inc	poids_cancer
age_inc	1.00000	0.67640	0.03187	0.09393
		<.0001	0.3938	0.1565
	786	229	718	229
age_cancer	0.67640	1.00000	0.12778	0.04091
	<.0001		0.0659	0.5379
	229	229	208	229
poids_inc	0.03187	0.12778	1.00000	0.77685
	0.3938	0.0659		<.0001
	718	208	718	208
poids_cancer	0.09393	0.04091	0.77685	1.00000
	0.1565	0.5379	<.0001	
	229	229	208	229

- Le coefficient de corrélation entre *age_cancer* et *poids_inc* est de 0,13
- Il n'est pas significativement différent de 0 (p = 0,07)
- Il a été calculé sur 208 données non manquantes

PROC NPAR1WAY

- Descriptif de la procédure

  - La procédure PROC NPAR1WAY est une procédure qui teste si 2 ou > 2 médianes sont significativement différentes

  - Intérêts de tester des médianes plutôt que des moyennes

    > Lorsque la distribution de la variable quantitative n'est pas normale

    > Lorsque les effectifs sont trop petits, on peut préférer le test des médianes

  - La procédure fournit un grand nombre de tests, tous non paramétriques (en plus d'une analyse de variance fournie)

  - Les principaux tests non paramétriques que l'on rencontre dans la littérature :

    > test de Wilcoxon pour la comparaison de 2 médianes

    > test de Kruskal-Wallis pour la comparaison de > 2 médianes

PROC NPAR1WAY

- Conditions d'application des tests de Wilcoxon et Kruskal-Wallis

  Les effectifs à comparer doivent être ≥ 10

- Syntaxe

  PROC NPAR1WAY Data = [*bibliothèque***.**]*nom_table* **;**
  Class *var_binaire* **;**
  Var *var_quantitative* **;**
  Run ;

  Problème de la procédure : elle ne fournit pas les médianes que l'on teste

  ⇒ Il faut au préalable avoir exécuté une PROC MEANS (ou PROC UNIVARIATE) pour afficher les médianes qui vont être comparées et testées

PROC NPAR1WAY

- Illustration avec la comparaison de 2 médianes

```
proc means data = TD_M2bio.data_exo2 median p25 p75 ;
class tabac ;
var poids_inc ;
run;

proc npar1way data = TD_M2bio.data_exo2 ;
class tabac ;
var poids_inc ;
run ;
```

Sortie n°1

The MEANS Procedure

Analysis Variable : poids_inc

tabac	N Obs	Median	25th Pctl	75th Pctl
0	380	70.0	63.1	76.6
1	322	68.1	62.9	73.7

Les 2 médianes [IQR] à comparer sont :

- 70,0 kg [63,1 ; 76,6] pour les non fumeurs
- 68,1 kg [62,9 ; 73,7] pour les fumeurs

PROC NPAR1WAY

- Illustration avec la comparaison de 2 médianes

Sortie n°2

The NPAR1WAY Procedure

Wilcoxon Scores (Rank Sums) for Variable poids_inc
Classified by Variable tabac

tabac	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
0	352	116516.0	112992.0	2332.87019	331.011364
1	289	89245.0	92769.0	2332.87019	308.806228

Average scores were used for ties.

Wilcoxon Two-Sample Test

Statistic	89245.0000
Normal Approximation	
Z	-1.5104
One-Sided Pr < Z	0.0655
Two-Sided Pr > |Z|	0.1309
t Approximation	
One-Sided Pr < Z	0.0657
Two-Sided Pr > |Z|	0.1314

Z includes a continuity correction of 0.5.

Kruskal-Wallis Test

Chi-Square	2.2819
DF	1
Pr > Chi-Square	0.1309

Le résultat du test donne p = 0,13

Les deux médianes ne sont pas significativement différentes

Le test de Kruskal-Wallis donne des résultats quasiment identiques à ceux de Wilcoxon

PROC NPAR1WAY

- Illustration avec la comparaison de > 2 médianes

```
proc means data = TD_M2bio.data_exo2 median p25 p75 maxdec=1;
  class age_K_cl;
  var poids_inc ;
  run;

proc npar1way data = TD_M2bio.data_exo2 ;
  class age_K_cl;
  var poids_inc ;
  run ;
```

Sortie n°1

The MEANS Procedure

Analysis Variable : poids_inc

Age au cancer en 4 classes	N Obs	Median	25th Pctl	75th Pctl
0	13	63.1	61.8	68.7
1	105	69.5	62.9	73.7
2	86	71.0	63.4	75.5
3	25	70.1	61.1	77.7

Les 4 médianes à comparer

PROC NPAR1WAY

- Illustration avec la comparaison de > 2 médianes

The NPAR1WAY Procedure

Wilcoxon Scores (Rank Sums) for Variable poids_inc
Classified by Variable age_K_cl

age_K_cl	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
0	13	955.50	1358.50	210.115544	73.500000
1	98	10245.00	10241.00	433.289914	104.540816
2	76	8390.50	7942.00	417.987179	110.401316
3	21	2145.00	2194.50	261.516834	102.142857

Average scores were used for ties.

Kruskal-Wallis Test

Chi-Square	4.2117
DF	3
Pr > Chi-Square	0.2395

Sortie n°2

Le résultat du test donne $p = 0,24$

Les 4 médianes ne sont pas significativement différentes

PROC NPARWAY ne donne pas le test de Wilcoxon lorsqu'il y a plus de 2 médianes à comparer…