

HEC MONTRÉAL

Analyse et inférence statistique

Laboratoire SAS

INTRODUCTION À SAS

Déroulement du laboratoire

- Récupérer les fichiers sur ZoneCours et les copier dans un répertoire de travail.
- Présentation du logiciel et exemples à l'aide du fichier `MATH60619_SAS_intro.sas`.
- Exercices :
 - Le fichier pdf `MATH60619_SASexercices` contient les exercices à compléter.
 - Les fichiers `elnino.sas7bdat` et `aapl.csv` contiennent les données pour les exercices.
 - Les solutions aux exercices seront mises en ligne sur ZoneCours.

Introduction à SAS

- Partie I : Familiarisation avec SAS et gestion des fichiers de données
- Partie II : Les énoncés DATA et PROC
- Partie III : ODS
- Partie IV : Graphiques avec SAS
- Partie V : Consulter l'aide de SAS

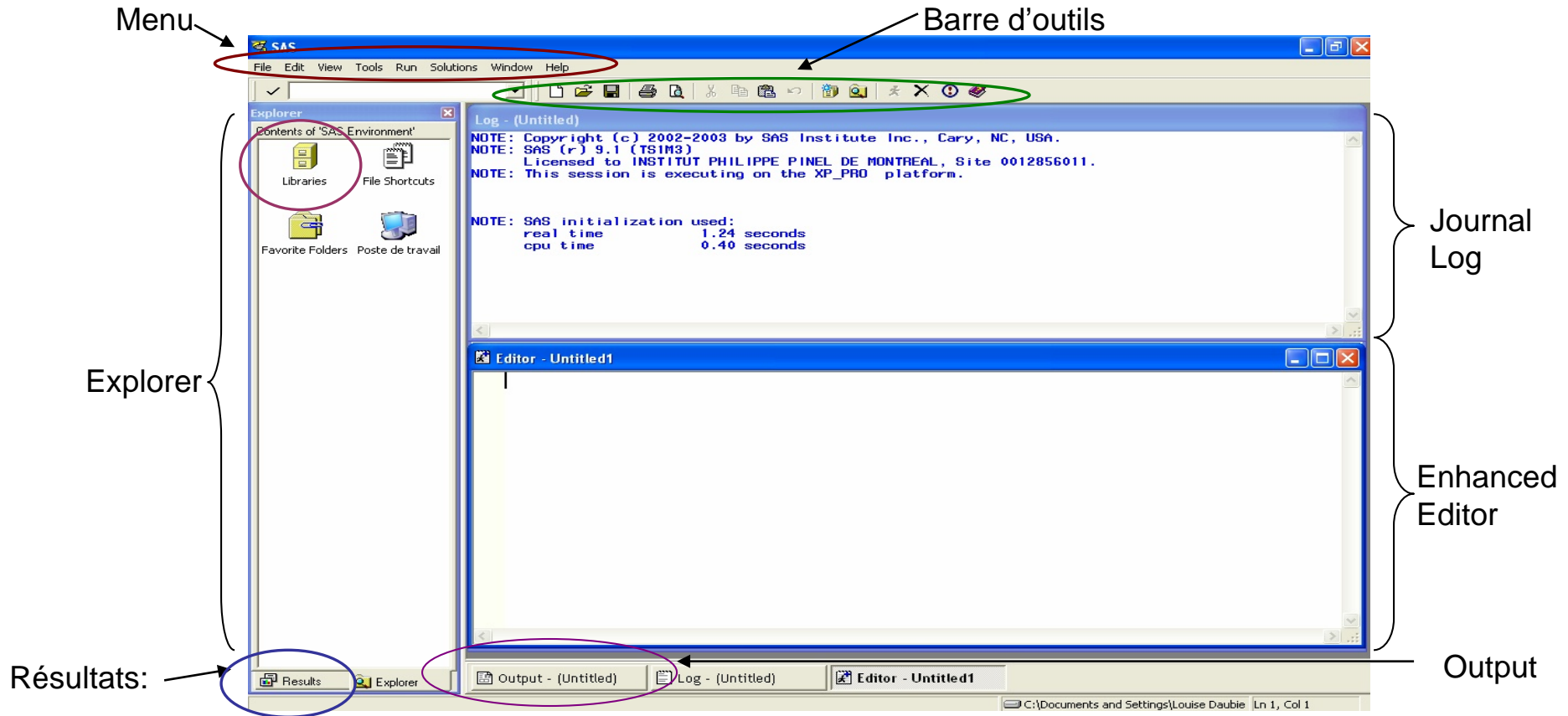
Familiarisation avec SAS

Partie I



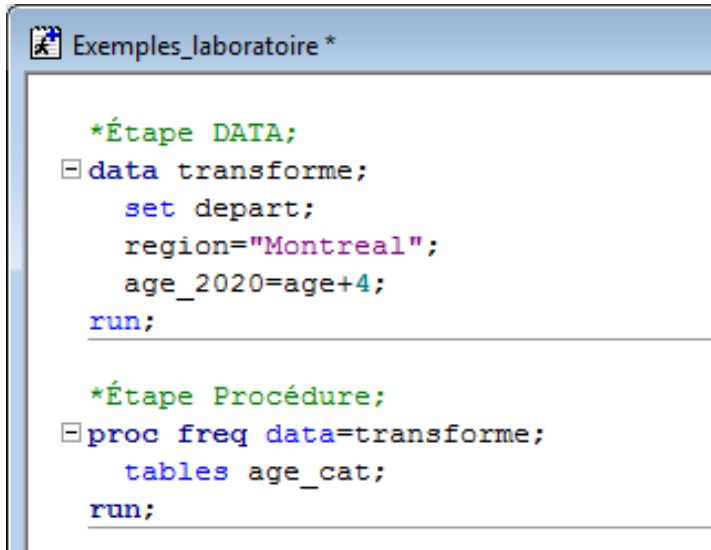
HEC MONTRÉAL

Vue d'ensemble



Fenêtre Editor

Fenêtre où l'on tape les instructions, le programme à exécuter



```
*Étape DATA;
data transforme;
    set depart;
    region="Montreal";
    age_2020=age+4;
run;

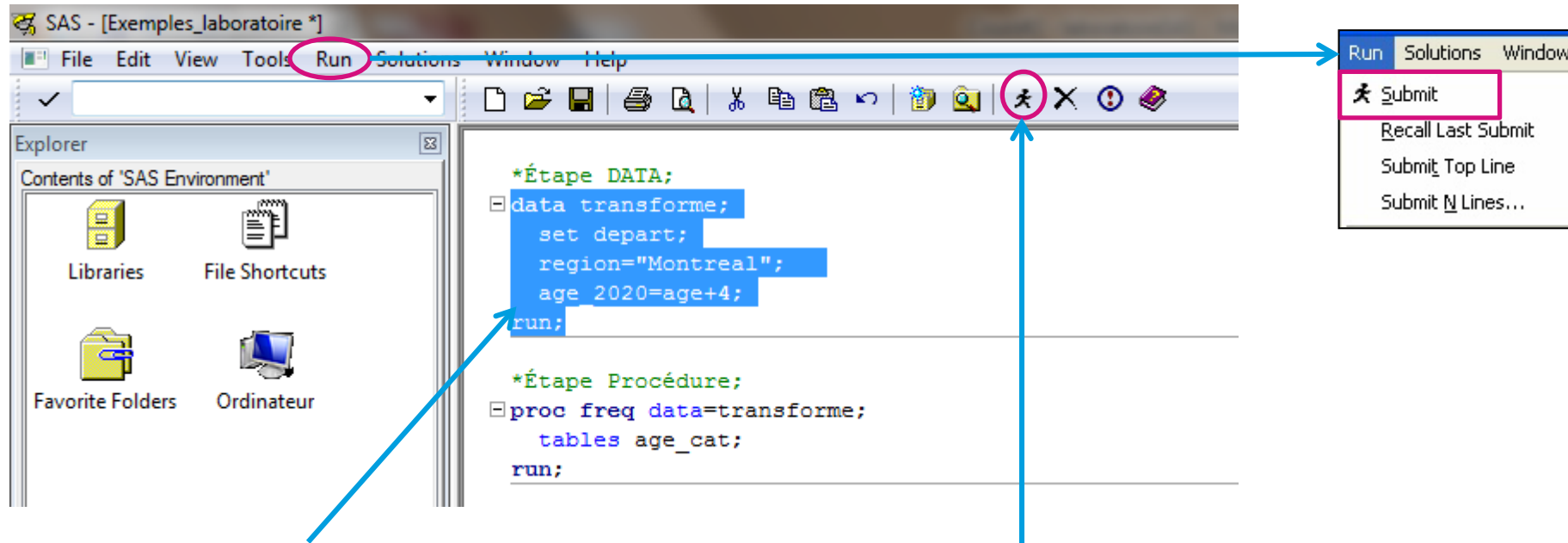
*Étape Procédure;
proc freq data=transforme;
    tables age_cat;
run;
```

Couleurs : donnent des indications utiles

- **Vert clair**: commentaires
- **Rouge**: incorrect
- **Bleu gras**: mot clé début et fin d'instruction
- **Bleu** : mot clé à l'intérieur de l'instruction
- **Violet**: chaîne de caractères (doit être encadrée d'apostrophes ou de guillemets)
- **Vert turquoise**: chiffres
- **Noir**: variables, tables, bibliothèques,...

Fenêtre Editor

Exécution d'un programme



1- Si l'on ne veut pas exécuter tout le programme dans l'éditeur, on surligne la partie à exécuter.

2- cliquer sur le bouton « Submit »

Généralité sur les syntaxes SAS

Dans SAS, il existe deux grands ensembles d'instructions :

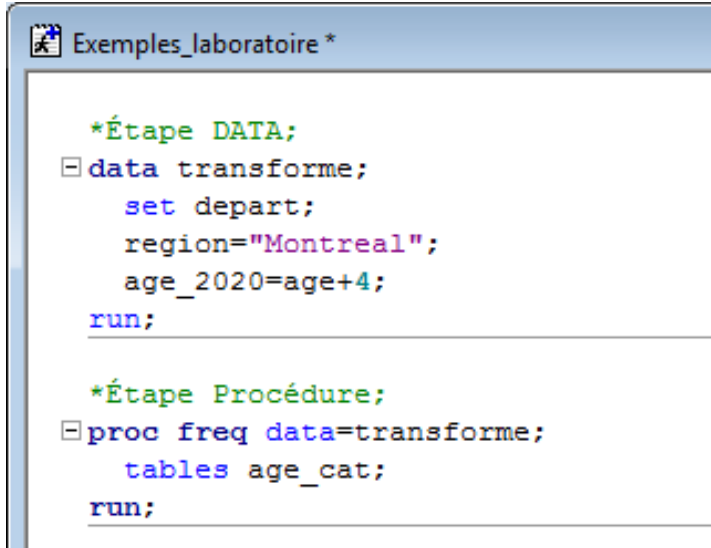
Les étapes DATA

- Permettent de manipuler les données
- Exemples : création de tables, création/ modification/ suppression de variables, fusion de tables...

Les étapes PROC (les procédures)

- Permettent de traiter les données
 - Exemples : produire des statistiques descriptives, imprimer les données, modèles statistiques...
- Il existe aussi des instructions globales qui permettent de définir des options qui resteront actives pour toutes les étapes du programme (exemple : OPTIONS et TITLE).

Généralité sur les syntaxes SAS



```
*Étape DATA;
data transforme;
  set depart;
  region="Montreal";
  age_2020=age+4;
run;

*Étape Procédure;
proc freq data=transforme;
  tables age_cat;
run;
```

- Une étape SAS commence par une instruction DATA ou une instruction PROC.
 - Une étape SAS se termine par une instruction RUN.
 - Une étape est composée de plusieurs énoncés qui débutent par un mot clé (ex: SET, VAR,...) et terminent par un point-virgule ; .
-
- SAS ne fait pas de distinction entre majuscules et minuscules.
 - Les chaînes de caractères doivent être encadrées d'apostrophes ou de guillemets.
 - Les commentaires débutent par /* et se terminent par */ .

Fenêtre Log (Journal)

Contient l'historique de l'exécution

3 types de messages :

- Note

```
NOTE: The data set WORK.TABLE1 has 3 observations and 3 variables.  
NOTE: DATA statement used (Total process time):  
      real time           0.01 seconds  
      cpu time            0.00 seconds
```

- Warning

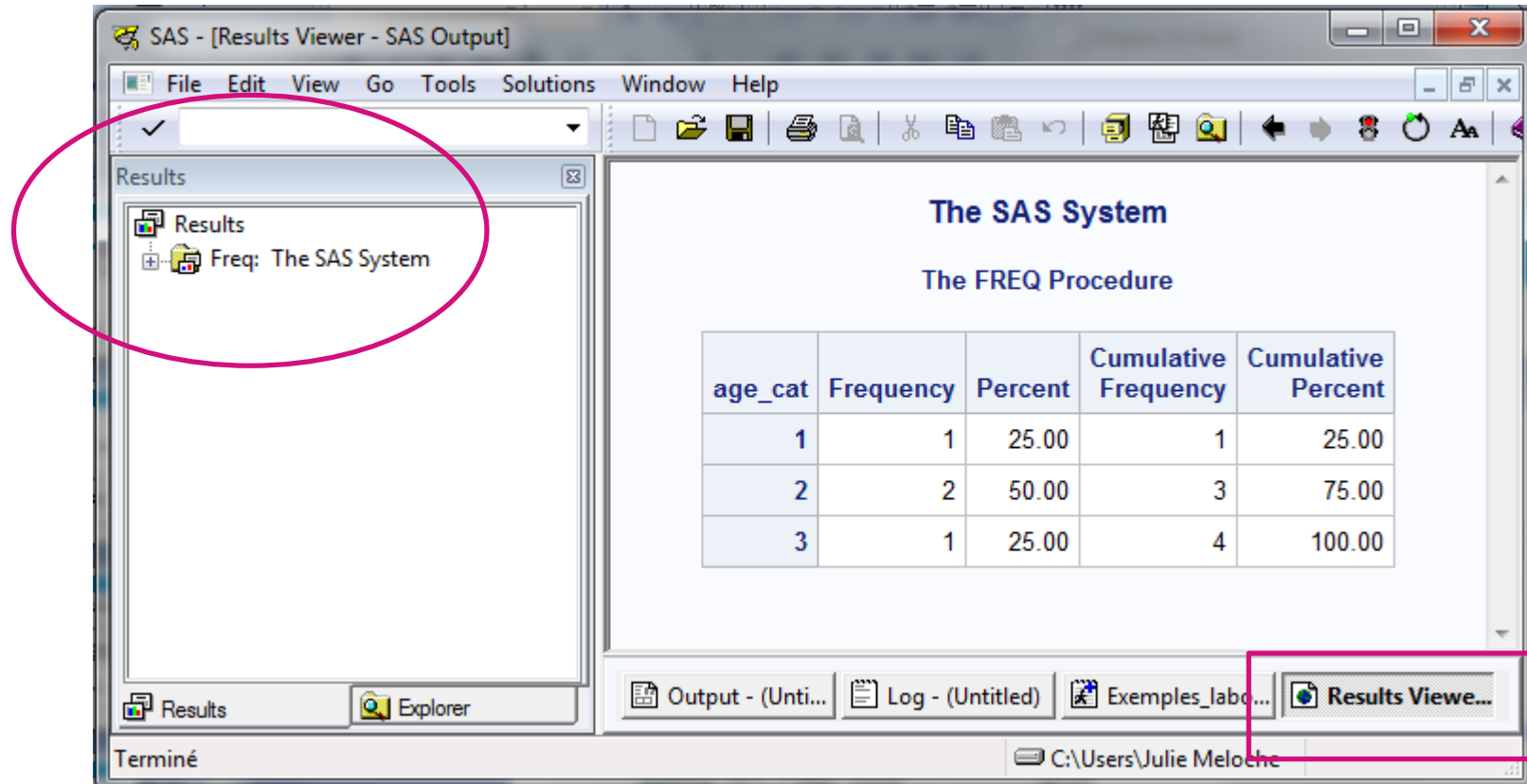
```
WARNING: The data set WORK.TABLE may be incomplete. When this step was stopped there were 0  
         observations and 2 variables.  
WARNING: Data set WORK.TABLE was not replaced because this step was stopped.
```

- Error (le programme est interrompu)

```
ERROR 85-322: Expecting a format name.  
ERROR 76-322: Syntax error, statement will be ignored.
```

Fenêtre Output (Sortie)

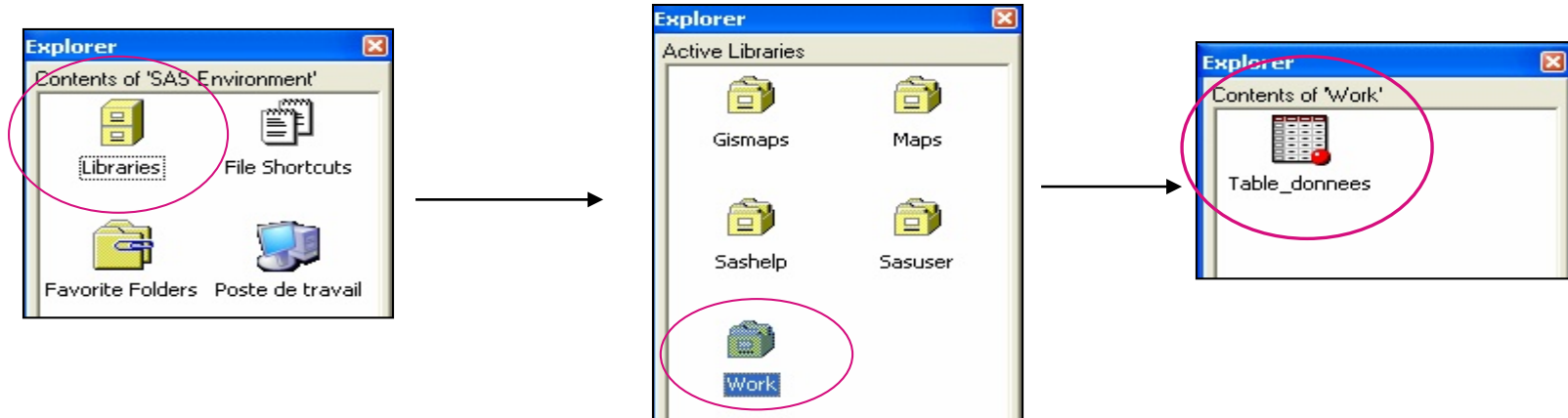
Contient les résultats



HEC MONTRÉAL

Bibliothèques (Librairies)

Dans la fenêtre Explorer



Bibliothèque (library)

- lieu de stockage des tables de données

Bibliothèque WORK

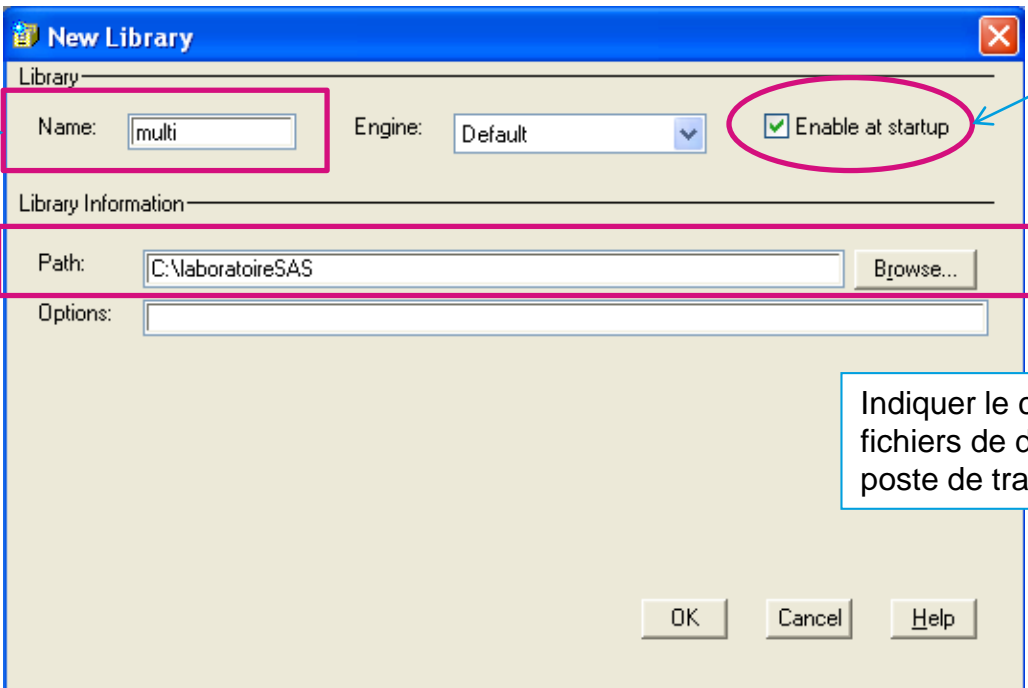
- Bibliothèque de stockage des tables par défaut
- Fichiers temporaires

Bibliothèques (Librairies)

Création d'une bibliothèque

- Cliquer sur le bouton *New Library*  dans la barre d'outils.
- Le nom de la bibliothèque est d'un maximum de 8 caractères

Nom de la bibliothèque:
8 caractères maximum →



Library

Name: multi Engine: Default ☒ Enable at startup

Library Information

Path: C:\laboratoireSAS Browse...

Options:

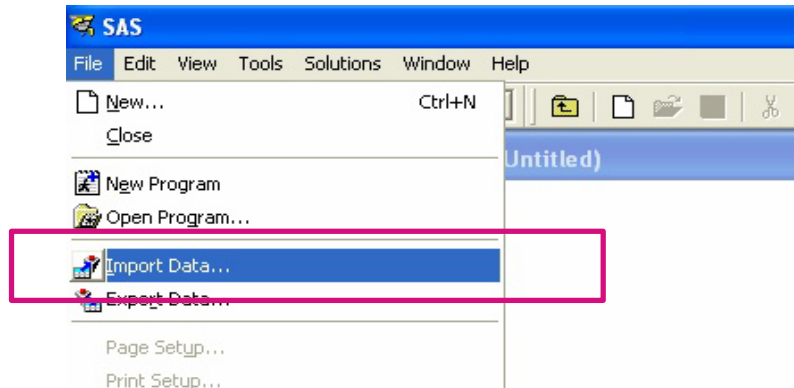
OK Cancel Help

Cocher « Enable at startup » pour charger la bibliothèque à chaque ouverture de SAS.

Indiquer le chemin d'accès aux fichiers de données SAS sur votre poste de travail.

Importer un fichier séparé par des virgules (CSV)

- Un assistant permet d'importer des fichiers CSV ou XLS en SAS



Création d'un jeu de données

- Définir un nouveau fichier de données en inscrivant nous même les données

```
data depart;  
  input age age_cat n_visite n_visite_cat sexe;  
cards;  
32 2 10 2 1  
30 2 6 2 1  
48 3 11 3 0  
17 1 21 3 0  
;  
run;
```



VIEWTABLE: Work.Depart						
	age	age_cat	n_visite	n_visite_cat	sexe	
1	32	2	10	2	1	
2	30	2	6	2	1	
3	48	3	11	3	0	
4	17	1	21	3	0	

Conclusion sur les fichiers de données

- Il y a essentiellement 2 façons de définir un nouveau jeu de données SAS : Nous pouvons inscrire les données nous mêmes ou nous pouvons modifier un fichier qui existe déjà.

```
data transforme;  
  set depart;  
  region="Montreal";  
  age_2020=age+4;  
run;
```



**Modification d'un
fichier de données
qui existe déjà**

Les énoncés DATA et PROC

Partie II



HEC MONTRÉAL

Généralité sur les noms SAS

Règles sur les noms de tables et de variables

- Doivent avoir un maximum de 32 caractères.
- Débutent par une lettre ou le symbole souligné (_).
- Les autres caractères doivent être des lettres, des chiffres ou le symbole souligné (_).
- Il faut éviter les espaces à l'intérieur d'un nom.

Types de variables

- Chaînes de caractères (alphanumériques)
 - Exemples : code postal, nom, prénom, ville,...
- Numériques
 - Exemples : âge, poids, date de naissance, salaire,...
 - Il est possible de choisir des formats d'affichage ou d'impression pour les dates, les montants en dollar et les pourcentages.

Étapes DATA

- Permettent de manipuler les données
- Exemples : création de tables, création/ modification/ suppression de variables, fusion de tables...

```
data infe.table_sortie;  
  set infe.table_entree;  
  age2020 = age + 4;  
  if (age > 18);  
  drop age;  
run;
```

Énoncés pour créer/modifier/supprimer des variables et pour sélectionner des observations.

Création de tables temporaires et permanentes

L'étape DATA :

- L'énoncé principal (**data**) spécifie le nom de la table qui sera créée et la bibliothèque où cette table sera classée.
→ *libref.nom_table*
- L'instruction **set** fait référence à une table déjà existante.

Reprenons le fichier de données *aapl* que vous avez importé dans la bibliothèque multi lors des premiers exercices.

```
data aapl;
  set infe.aapl;
run;
```

```
data work.aapl;
  set infe.aapl;
run;
```

Fichier de données **temporaire**



Création de tables temporaires et permanentes

L'étape DATA :

- Si une table SAS doit être conservée pour d'autres sessions SAS, il est important de la créer dans une bibliothèque permanente (*multi* par exemple).

```
data infe.aapl_final; ← Fichier de données permanent  
    set aapl;  
run;
```

- Le contenu de la bibliothèque WORK est supprimé à la fin de la session SAS.

Transformation de variables

Valeurs manquantes

- Pour les nombres, les valeurs manquantes sont représentées par un point .
- Pour les chaînes de caractères (alphanumériques), les valeurs manquantes sont représentées par un espace entre apostrophes (' ') ou guillemets (" ")

Opérateurs

- + Addition
- Soustraction
- * Multiplication
- / Division
- ** Exponentielle
- || Concaténation de chaînes de caractères

```
data intro_ex1_mod;  
    set intro_ex1;  
    age_2020=age+4;  
run;
```

Transformation de variables

Fonctions mathématiques

- **LOG**(*nombre*) : Logarithme naturel
- **EXP**(*nombre*) : Exponentiel
- **ABS**(*nombre*) : Valeur absolue
- **SQRT**(*nombre*) : Racine carrée
- **INT**(*nombre*) : Tronque à l'entier
- **ROUND**(*nombre*, *unités d'arrondi*) : Arrondi (*unit* = 1 /0.1/ 100/...)

```
*Fonctions mathématiques;  
□ data intro_ex1_mod;  
  set intro_ex1_mod;  
  n_visite_mois=round(n_visite/12,0.1);  
run;
```

Transformation de variables

Fonctions statistiques

- **MIN**(*nombre1,nombre2,...*) : minimum d'une série de variables
- **MAX**(*nombre1,nombre2,...*) : maximum d'une série de variables
- **SUM**(*nombre1,nombre2,...*) : somme d'une série de variables
- **MEAN**(*nombre1,nombre2,...*) : moyenne d'une série de variables
- **MEDIAN**(*nombre1,nombre2,...*) : médiane d'une série de variables
- **NMISS**(*nombre1,nombre2,...*) : nombre de valeurs manquantes
- **N**(*nombre1,nombre2,...*) : nombre de valeurs non manquantes

```
*Fonctions statistiques;  
data intro_ex1_mod;  
    set intro_ex1_mod;  
    n_visite_max=max(n_visite,n_visite_future);  
run;
```


Instructions logiques

Les énoncés IF...THEN et ELSE IF...THEN

- Permet d'appliquer une instruction qu'à un sous-groupe d'observations.
- Utile pour recoder une variable continue en variable catégorielle, par exemple recoder l'âge en tranches d'âge.
- La syntaxe à utiliser est la suivante :

```
IF expression THEN instruction;  
ELSE IF expression THEN instruction;  
:  
ELSE instruction;
```

- **IF** permet d'énoncer une expression logique pour sélectionner un sous-groupe d'observations.
- **ELSE** indique que l'expression ou l'instruction s'appliquera qu'aux observations non sélectionnées précédemment.
- **THEN** donne l'instruction à appliquer aux observations ciblées.

Instructions logiques

Les énoncés IF...THEN et ELSE IF...THEN

```
*Conditions;  
data intro_ex1_mod;  
  set intro_ex1;  
  if age <= 17 then adulte=0;  
  else adulte=1;  
  if (adulte=1 & n_visite_cat = 3) then adulte_fanatigue = 1;  
  else adulte_fanatigue = 0;  
  if (n_visite=1 | n_visite=2) then visiteur_frequent=0;  
  else visiteur_frequent=1;  
run;
```

Traitement des variables

DROP= variables à supprimer dans la table cible

```
*Sélection de variables DROP;  
data intro_ex1_mod2;  
    set intro_ex1_mod;  
    drop age adulte_fanatisme visiteur_frequent;  
run;
```

KEEP= variables à conserver dans la table cible

```
*Sélection de variables KEEP;  
data intro_ex1_mod2;  
    set intro_ex1_mod;  
    keep age adulte_fanatisme visiteur_frequent;  
run;
```

Sélection d'une plage de variables

- Dans une instruction plusieurs variables peuvent être spécifiées. On peut utiliser la syntaxe suivante pour faire référence à plusieurs variables.
 - sat1-sat3 : sat1, sat2, sat3.
 - varA -- varB : Toutes les variables situées entre varA et varB dans la table.

```
data Elnino2;  
  set Multi.Elnino;  
  keep obs year--day x1-x5;  
run;
```

=

```
data Elnino2;  
  set Multi.Elnino;  
  keep obs year month day x1 x2 x3 x4 x5;  
run;
```

Sélection d'observation

Création d'une nouvelle table qui contient un sous-groupe d'observations provenant de la table de référence. L'instruction IF suivi d'une condition logique permettra :

- de sélectionner les observations à conserver
ou
- d'identifier les observations à exclure pour ensuite les supprimer.

Conservation des observations si l'indicateur de visiteur fréquent est égal à 1.

```
*Sélection d'observations;  
data intro_ex1_mod3;  
  set intro_ex1_mod;  
  if (visiteur_frequent=1);  
run;
```

Suppression des observations si l'indicateur de visiteur fréquent est différent de 1.

```
data intro_ex1_mod3;  
  set intro_ex1_mod;  
  if (visiteur_frequent NE 1) then delete;  
run;
```

Les procédures

- PROC **CONTENTS** : Permet d'obtenir de l'information sur le fichier de données.
- PROC **PRINT** : impression dans la surface SAS de rapports
- PROC **FREQ** : Tableaux de fréquences et tableaux croisés.
- PROC **MEANS** : Statistiques sommaires (moyenne, écart-type,...).
- PROC **UNIVARIATE** : graphique et statistiques descriptives
- PROC **SORT** : Trier un fichier de données.
 - Options NODUP et NODUPKEY
 - Option MERGE
- PROC **TRANSPOSE** (format long à format court et vice-versa)

PROC CONTENTS

Permet d'obtenir des informations sur la table : nombre d'observations, nombre de variables, liste des variables, date de création,...

```
proc contents data=intro_ex1_mod;  
run;
```

Exemple
de sortie



The CONTENTS Procedure			
Data Set Name	WORK.INTRO_EX1_MOD	Observations	100
Member Type	DATA	Variables	12
Engine	V9	Indexes	0
Created	Tue, Aug 23, 2016 10:05:10 o'clock AM	Observation Length	96
Last Modified	Tue, Aug 23, 2016 10:05:10 o'clock AM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

PROC CONTENTS

Exemple de sortie (suite)...

```
proc contents data=intro_ex1_mod;  
run;
```

Alphabetic List of Variables and Attributes			
#	Variable	Type	Len
10	adulte	Num	8
11	adulte_fanatique	Num	8
1	age	Num	8
7	age_2020	Num	8
2	age_cat	Num	8
3	n_visite	Num	8
4	n_visite_cat	Num	8
5	n_visite_future	Num	8
9	n_visite_max	Num	8
8	n_visite_mois	Num	8
6	sexe	Num	8
12	visiteur_frequent	Num	8

PROC PRINT

Cette procédure affiche un rapport dans la fenêtre **results** de SAS à partir d'un jeu de données. Par exemple, pour imprimer uniquement les 5 premières observations

```
proc print data=multi.elnino (obs=5);  
run;
```

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

Comme pour d'autres procédures, il est également possible de masquer certaines caractéristiques de la procédure présentes par défaut, ici l'identifiant de l'observation déjà donné dans une colonne.

```
proc print data=multi.elnino (obs=10) noobs;  
run;
```

Le Système SAS												
obs	year	month	day	date	latitude	longitude	zon_winds	mer_winds	humidity	air_temp	s_s_temp	
1	80	3	7	800307	-0.02	-109.46	-6.8	0.7	.	26.14	26.24	
2	80	3	8	800308	-0.02	-109.46	-4.9	1.1	.	25.66	25.97	
3	80	3	9	800309	-0.02	-109.46	-4.5	2.2	.	25.69	25.28	
4	80	3	10	800310	-0.02	-109.46	-3.8	1.9	.	25.57	24.31	
5	80	3	11	800311	-0.02	-109.46	-4.2	1.5	.	25.30	23.19	
6	80	3	12	800312	-0.02	-109.46	-4.4	0.3	.	24.72	23.64	
7	80	3	13	800313	-0.02	-109.46	-3.2	0.1	.	24.66	24.34	
8	80	3	14	800314	-0.02	-109.46	-3.1	0.6	.	25.17	24.14	
9	80	3	15	800315	-0.02	-109.46	-3.0	1.0	.	25.59	24.24	
10	80	3	16	800316	-0.02	-109.46	-1.2	1.0	.	26.71	25.94	

PROC FREQ

La procédure FREQ

- Produit des tableaux de fréquences pour des variables numériques et alphanumériques.
- Produit des tableaux de une, deux ou même plusieurs dimensions.
- Offre la possibilité de calculer différentes statistiques comme le khi-carré, le rapport de cotes et le risque relatif.

```
PROC FREQ Data=Table-SAS <options>;  
    BY variables;  
    TABLES variable(s) / <options>;  
RUN;
```

PROC FREQ

Tableau à une dimension

- L'instruction TABLES spécifie la ou les variable(s) sélectionnée(s) pour le calcul des fréquences.

```
proc freq data=intro_ex1;  
    tables n_visite_cat;  
run;
```

The FREQ Procedure

n_visite_cat	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	14	14.00	14	14.00
2	63	63.00	77	77.00
3	23	23.00	100	100.00

PROC FREQ

Tableau à deux dimensions

- Produire un tableau croisé entre deux variables.
- Les deux variables sont spécifiées dans l'instruction TABLES. Le symbole (*) indique que les deux variables seront croisées.
- Il est possible de créer plusieurs tableaux dans une même instruction TABLES.

```
❏ proc freq data=intro_ex1;  
    tables age_cat*n_visite_cat;  
run;
```

PROC FREQ

Sortie de résultats :

```
proc freq data=intro_ex1;  
    tables age_cat*n_visite_cat;  
run;
```

Pourcentage ligne →

Pourcentage colonne →

Frequency Percent Row Pct Col Pct	Table of age_cat by n_visite_cat				
	age_cat	n_visite_cat			
		1	2	3	Total
1		11	6	14	31
		11.00	6.00	14.00	31.00
		35.48	19.35	45.16	
		78.57	9.52	60.87	
2		2	32	4	38
		2.00	32.00	4.00	38.00
		5.26	84.21	10.53	
		14.29	50.79	17.39	
3		1	25	5	31
		1.00	25.00	5.00	31.00
		3.23	80.65	16.13	
		7.14	39.68	21.74	
Total		14	63	23	100
		14.00	63.00	23.00	100.00

PROC MEANS

La procédure MEANS

- Permet d'obtenir des statistiques descriptives pour une variable numérique : nombre d'effectifs, somme, moyenne, écart-type, minimum, maximum, médiane, quartiles,...

```
PROC MEANS Data=Table-SAS <statistiques> <options>;  
    BY variables;  
    CLASS variable(s) / <options>;  
    VAR variable(s);  
RUN;
```

PROC MEANS

L'instruction **VAR** spécifie les variables numériques (si non mentionné, toutes les variables numériques seront traitées).

Les statistiques descriptives par défaut sont :

- Le nombre d'observations non manquantes (N);
- La moyenne (Mean);
- L'écart-type (Std Dev);
- Le minimum (Minimum)
- Le maximum (Maximum).

```
proc means data=intro_ex1_mod;  
    var age;  
run;
```

The MEANS Procedure

Analysis Variable : age				
N	Mean	Std Dev	Minimum	Maximum
100	30.2800000	9.9392905	14.0000000	61.0000000

PROC MEANS

Liste partielle des statistiques disponibles

Option PROC MEANS	Statistique
N	Nombre d'observations complètes
NMISS	Nombre d'observations manquantes
MEAN	Moyenne
SUM	Somme
MIN	Minimum
MAX	Maximum
MEDIAN	Médiane
STD	Écart-type
VAR	Variance
CLM	Intervalle de confiance à 95% pour la moyenne
Q1	1e quartile (25 ^e percentile)
Q3	3e quartile (75 ^e percentile)
QRANGE	Étendue interquartile (Q3-Q1)

Note : l'option `maxdec=valeur` est utile pour contrôler le nombre de décimales affichées dans le tableau de résultats. Par exemple,

PROC MEANS `data=table mean std maxdec=2;`

Les résultats de la moyenne et de l'écart-type seront présentés avec 2 décimales. Cette option s'applique à toutes les statistiques.

```
proc means data=intro_ex1_mod n mean std maxdec=2;  
    var age;  
run;
```


PROC SORT : trier les données

```
proc sort data=intro_ex1_mod;  
  by age_cat descending n_visite;  
run;
```

Options:

- **BY** : identifie la ou les variables selon lesquelles les données seront triées.
- **DESCENDING**: tri descendant. Lorsque cette option est spécifiée devant une variable, les données seront triées de sorte que la plus grande valeur sera au début et la plus petite valeur à la fin.

VIEWTABLE: Work.Intro_ex1_mod


	age	age_cat	n_visite
24	18	1	3
25	22	1	2
26	18	1	2
27	18	1	2
28	21	1	1
29	24	1	1
30	24	1	1
31	17	1	1
32	30	2	12
33	29	2	11
34	33	2	11
35	32	2	11
36	32	2	10
37	25	2	10
38	34	2	10
39	28	2	10
40	34	2	10
41	26	2	10
42	29	2	10
43	26	2	9
44	34	2	9
45	31	2	9

PROC MEANS

L'instruction **BY** :

- sortir les statistiques descriptives par groupes spécifiés par une ou plusieurs variables catégorielles;
- La table doit, au préalable, être triée selon les variables spécifiées dans l'instruction **BY**.

```
❏ proc sort data=intro_ex1_mod;  
    by sexe;  
run;  
  
❏ proc means data=intro_ex1_mod n nmiss mean std median min max;  
    var age;  
    by sexe;  
run;
```



PROC MEANS

The MEANS Procedure

sexe=0

Analysis Variable : age						
N	N Miss	Mean	Std Dev	Median	Minimum	Maximum
59	0	29.1864407	10.2460667	28.0000000	14.0000000	56.0000000

sexe=1

Analysis Variable : age						
N	N Miss	Mean	Std Dev	Median	Minimum	Maximum
41	0	31.8536585	9.3796614	31.0000000	18.0000000	61.0000000

BY : en sortie,
une table par
entité

PROC SORT : nodup et nodupkey

- Deux options connues : **NODUP** et **NODUPKEY**.
- Pour supprimer les doublons dans une base de données.
- La différence principale entre NODUP et NODUPKEY est que:
- **NODUP** : va supprimer les observations qui possèdent des doublons en tout point.

HTR753	Caroline	9	A2019
--------	----------	---	-------

HTR753	Caroline	9	A2019
--------	----------	---	-------

- **NODUPKEY** : va supprimer les valeurs doublons de la variable de tri.

HTR753	Sylvie	12	H2019
--------	--------	----	-------

HTR753	Caroline	9	A2019
--------	----------	---	-------

```
proc sort data=multi.elnino nodupkey;  
by obs;  
run;
```

```
NOTE: There were 10000 observations read from the data set MULTI.ELNINO.  
NOTE: 0 observations with duplicate key values were deleted.  
NOTE: The data set MULTI.ELNINO has 10000 observations and 12 variables.
```

Option MERGE (et proc sort)

Pour juxtaposer des tables dans SAS on utilise la clause MERGE

Il faut toujours trier les jeux de données selon la même variable avant d'utiliser l'option **merge** dans une étape data

Exemple : nous voulons juxtaposer le jeu elnino avec les valeurs de degrés Celsius

```
proc sort data=multi.elnino;  
by obs;  
run;  
proc sort data=multi.farenheit;  
by obs;  
run;  
  
data elninomerge;  
merge multi.elnino multi.farenheit;  
by obs ;  
run;
```



PROC TRANSPOSE

La procédure **TRANSPOSE** crée un jeu de données restructuré, en transposant les variables sélectionnées en observations. Donc les lignes deviennent les colonnes et les colonnes deviennent les lignes. Les variables deviennent les observations et vice-versa.

La forme générale de la procédure pour transposer le jeu de données en entier est

```
proc transpose data=multi.elnino out=multi.transpose;  
run;
```

	NOM DE LA VARIABLE PRECEDENTE	COL1	COL2	COL3	COL4	COL5	COL6
1	obs	1	2	3	4	5	
2	year	80	80	80	80	80	
3	month	3	3	3	3	3	
4	day	7	8	9	10	11	
5	date	800307	800308	800309	800310	800311	800312
6	latitude	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02
7	longitude	-109.46	-109.46	-109.46	-109.46	-109.46	-109.46
8	zon_winds	-6.8	-4.9	-4.5	-3.8	-4.2	-4.2
9	mer_winds	0.7	1.1	2.2	1.9	1.5	0.7
10	humidity
11	air_temp	26.14	25.66	25.69	25.57	25.3	24.7
12	s_s_temp	26.24	25.97	25.28	24.31	23.19	23.6

PROC TRANSPOSE

1. TRANSPOSER UN GROUPE DE VARIABLES

On transpose les données d'un tableau par rapport aux valeurs d'une variable choisie à l'aide d'une instruction **BY**, tout en gardant la variable spécifiée par **by** comme colonne. C'est ce qu'on appelle la transposition par groupe.

```
proc sort data=multi.elnino;
by year;
run;

proc transpose data=multi.elnino
out=multi.transposel;
by year;
run;
```

	year	NOM DE LA VARIABLE PRECEDENTE	COL1	COL2	COL3
1	80	obs	1	2	3
2	80	month	3	3	3
3	80	day	7	8	9
4	80	date	800307	800308	800309
5	80	latitude	-0.02	-0.02	-0.02
6	80	longitude	-109.46	-109.46	-109.46
7	80	zon_winds	-6.8	-4.9	-4.5
8	80	mer_winds	0.7	1.1	2.2
9	80	humidity	.	.	.
10	80	air_temp	26.14	25.66	25.69
11	80	s_s_temp	26.24	25.97	25.28
12	81	obs	167	168	169
13	81	month	1	1	1
14	81	day	1	2	3
15	81	date	810101	810102	810103
16	81	latitude	0	0	0
17	81	longitude	-109.56	-109.56	-109.56
18	81	zon_winds	-6.3	-7.9	-6.1
19	81	mer_winds	3.5	2.7	2.9
20	81	humidity	.	.	.
21	81	air_temp	22.63	22.55	22.35
22	81	s_s_temp	22.58	22.47	22.39
23	82	obs	532	533	534
24	82	month	1	1	1
25	82	day	1	2	3
26	82	date	820101	820102	820103
27	82	latitude	0.03	0.03	0.03

PROC TRANSPOSE

2. TRANSPOSER UNIQUEMENT LES DONNÉES DE CERTAINES VARIABLES

Les données des autres variables seront éliminées. Il faut spécifier dans une instruction **VAR** la ou les variables à garder et à transposer.

```
proc transpose data=multi.elnino out=multi.transpose1;  
var year obs date air_temp mer_winds;  
run;
```

	NOM DE LA VARIABLE PRECEDENTE	COL1	COL2	COL3	COL4	COL5	COL6	COL7
1	obs	1	2	3	4	5	6	7
2	date	800307	800308	800309	800310	800311	800312	800313
3	air_temp	26.14	25.66	25.69	25.57	25.3	24.72	24.72
4	mer_winds	0.7	1.1	2.2	1.9	1.5	0.3	0.3

PROC TRANSPOSE

3. TRANSPOSER CERTAINES DONNÉES SELON UN GROUPE DE VARIABLES

Transposer les valeurs de certaines variables par groupe d'années :

```
proc transpose data=multi.elnino  
  out=multi.transpose3;  
var obs date air_temp mer_winds;  
by year;  
run;
```

	year	NOM DE LA VARIABLE PRECEDENTE	COL1	COL2	COL3	COL4	COL5
1	80	obs	1	2	3	4	5
2	80	date	800307	800308	800309	800310	800311
3	80	air_temp	26.14	25.66	25.69	25.57	25.3
4	80	mer_winds	0.7	1.1	2.2	1.9	1.5
5	81	obs	167	168	169	170	171
6	81	date	810101	810102	810103	810104	810105
7	81	air_temp	22.63	22.55	22.35	22.62	22.77
8	81	mer_winds	3.5	2.7	2.9	2.4	2.4
9	82	obs	532	533	534	535	536
10	82	date	820101	820102	820103	820104	820105
11	82	air_temp	23.7	23.51	23.55	23.44	23.43
12	82	mer_winds	3.9	3.9	3.1	3.2	1.8
13	83	obs	672	673	674	675	676
14	83	date	830101	830102	830103	830104	830105
15	83	air_temp	27.51	27.42	27.74	26.79	27.33
16	83	mer_winds
17	84	obs	843	844	845	846	847
18	84	date	840101	840102	840103	840104	840105
19	84	air_temp	22.91	22.28	22.55	22.4	22.43
20	84	mer_winds	1.2	2.7	0.6	1.1	0.3
21	85	obs	1176	1177	1178	1179	1180
22	85	date	850510	850511	850512	850513	850514
23	85	air_temp	24.07	23.86	23.76	24.02	24.35
24	85	mer_winds	-1	0.6	1.7	1	2.8
25	86	obs	1412	1413	1414	1415	1416
26	86	date	860101	860102	860103	860104	860105
27	86	air_temp

ODS

Partie III



HEC MONTRÉAL

Exportation vers un document (ODS)

ODS permet d'organiser l'exportation de données vers un document PDF, **RTF** ou HTML

```
ods rtf;
```

```
proc freq data=intro_ex1;  
  tables age_cat*n_visite_cat;  
run;
```

```
ods rtf close;
```

11:12 Tuesday, August 23, 2016 1

The SAS System

The FREQ Procedure

Table of age_cat by n_visite_cat

age_cat	n_visite_cat			
Frequency Percent Row Pct Col Pct	1	2	3	Total
1	11 11.00 35.48 78.57	6 6.00 19.35 9.52	14 14.00 45.16 60.87	31 31.00
2	2 2.00 5.26 14.29	32 32.00 84.21 50.79	4 4.00 10.53 17.39	38 38.00
3	1 1.00 3.23 7.14	25 25.00 80.65 39.68	5 5.00 16.13 21.74	31 31.00
Total	14 14.00	63 63.00	23 23.00	100 100.00

Page : 1 sur 1 Mots : 71 Anglais (Canada) 100 %

Conserver un tableau de résultats dans une table (ODS)

- Le module ODS permet aussi de sélectionner une partie de la sortie SAS et d'en faire un jeu de données

```
ods trace on;
```

```
proc freq data=intro_ex1;  
  tables age;  
run;
```

```
ods trace off;
```

```
proc freq data=intro_ex1;  
  tables age;  
  ods output Freq.Table1.OneWayFreqs=dist_age;  
run;
```

Description de chaque partie du output dans le journal (Log)

Log - (Untitled)

Output Added:

Name: OneWayFreqs
Label: One-Way Frequencies
Template: Base.Freq.OneWayFreqs
Path: Freq.Table1.OneWayFreqs

NOTE: There were 100 observations read from the data set WORK.INTRO_EX1.
NOTE: PROCEDURE FREQ used (Total process time):

Conserver un tableau de résultats dans une table (ODS)

- Le tableau de résultats se nomme Freq.Table1.OneWayFreqs.
- Il sera conservé dans le tableau **dist_age**.

```
proc freq data=intro_ex1;  
    tables age;  
    ods output Freq.Table1.OneWayFreqs=dist_age;  
run;
```

VIEWTABLE: Work.Dist_age (One-Way Frequencies)							
	Table	age	age	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	Table age	14	14	1	1.00	1	1.00
2	Table age	15	15	2	2.00	3	3.00
3	Table age	16	16	2	2.00	5	5.00
4	Table age	17	17	3	3.00	8	8.00
5	Table age	18	18	7	7.00	15	15.00
6	Table age	19	19	2	2.00	17	17.00
7	Table age	20	20	1	1.00	18	18.00
8	Table age	21	21	1	1.00	19	19.00
9	Table age	22	22	7	7.00	26	26.00
10	Table age	23	23	1	1.00	27	27.00
11	Table age	24	24	4	4.00	31	31.00
12	Table age	25	25	6	6.00	37	37.00
13	Table age	26	26	4	4.00	41	41.00
14	Table age	27	27	2	2.00	43	43.00
15	Table age	28	28	4	4.00	47	47.00
16	Table age	29	29	4	4.00	51	51.00
17	Table age	30	30	4	4.00	55	55.00

Graphiques

Partie IV



HEC MONTRÉAL

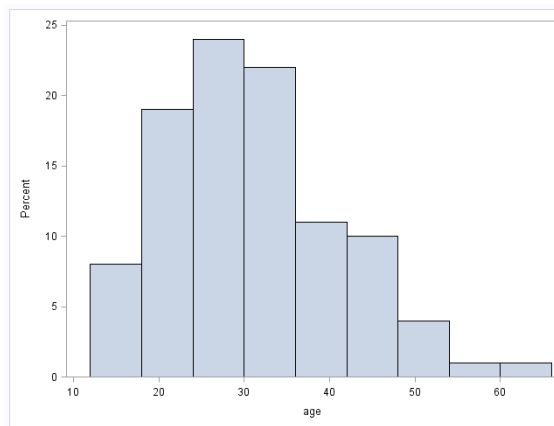
Procédure SGPLOT

PROC SGPLOT

- Graphiques qui illustrent la distribution d'une variable: graphique à ligne, histogramme,...
- Graphiques qui illustrent la relation entre deux variables : nuage de points, box-plots,...
- Possibilité de superposer des graphiques
- Des options sont disponibles pour contrôler l'apparence du graphique et ajouter des éléments comme une légende.

```
proc sgplot data=intro_ex1;  
  histogram age;  
run;
```

→ Spécifie le type de graphique



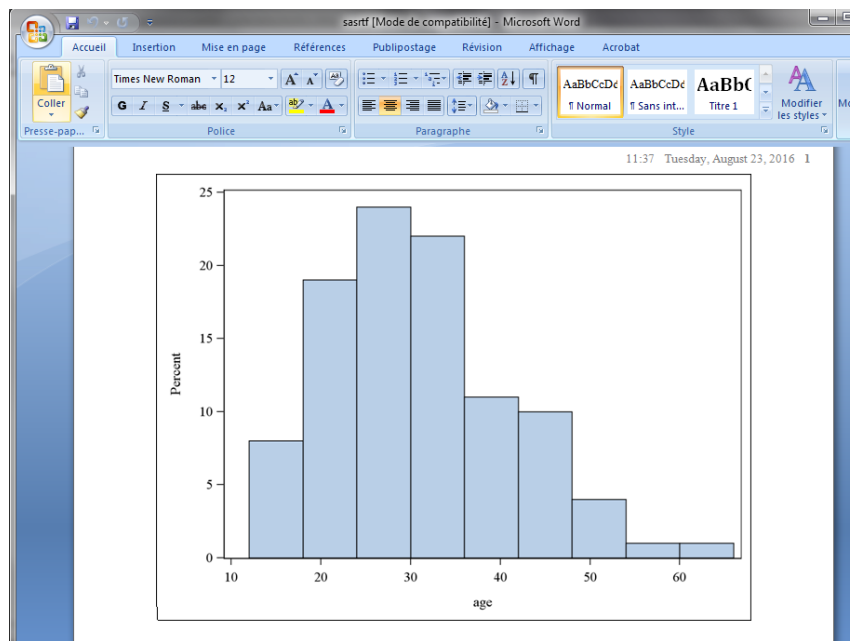
ODS et graphiques

- Les graphiques et sorties peuvent être automatiquement envoyés dans un fichier Word (ou pdf) via une instruction **ods**.

```
ods graphics on;  
ods rtf;
```

```
proc sgplot data=intro_ex1;  
  histogram age;  
run;
```

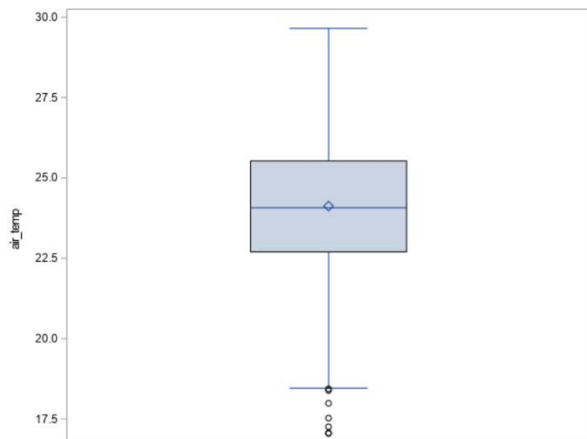
```
ods rtf close;  
ods graphics off;
```



PROC SGPLOT (boîte à moustaches)

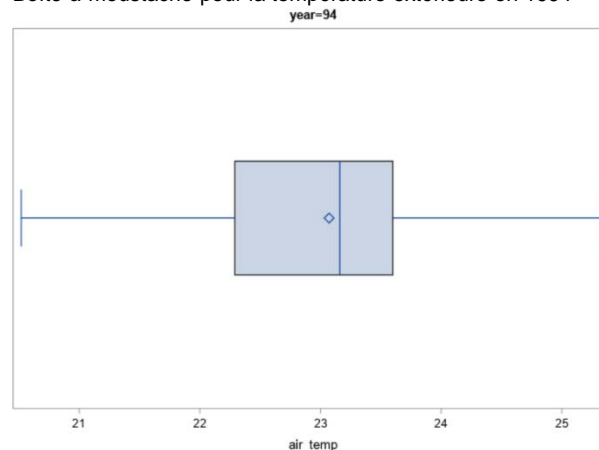
La procédure **sgplot** permet de créer des boîtes à moustaches horizontales (**hbox**) ou verticales (**vbox**), mais également une boîte par catégorie (il faut toujours trier avec **proc sort** au préalable avant d'utiliser **by**).

```
proc sgplot data=multi.elnino;  
vbox air_temp;  
run;
```



```
proc sgplot data=multi.elnino;  
hbox air_temp;  
by year;  
run;
```

Boîte-à-moustache pour la température extérieure en 1994

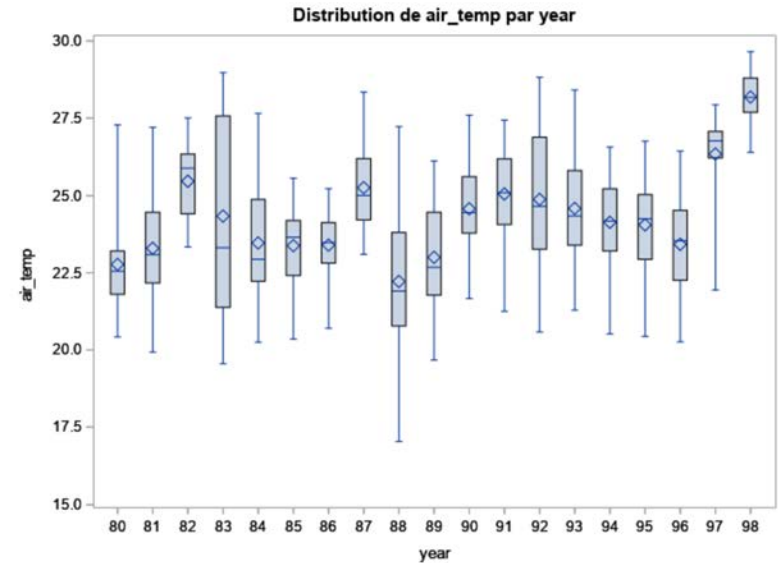


PROC BOXPLOT

Il existe une procédure dédiée **boxplot** spécifiquement avec laquelle nous avons d'autres options pour les graphiques.

Ici nous pouvons montrer une boîte par année. Afin que la procédure fonctionne il faut s'assurer que les données sont triées dans l'ordre croissant des données catégorielles.

```
proc sort data=multi.elnino;  
by year;  
run;  
  
proc boxplot data=multi.elnino;  
plot air_temp*year;  
run;
```



Consulter l'aide SAS

Partie V



HEC MONTRÉAL

Références utiles

- Ron Cody. *Learning SAS by Example : A Programmer's Guide*. Sas Institute Inc.
- Lora D. Delwiche, Susan J. Slaughter. 1998. *The Little SAS Book*, Fourth Edition. Sas Institute Inc.
- SAS Help and Documentation (par les menus)
- Support technique de SAS sur le web : <http://support.sas.com/idex.html> et <http://support.sas.com/techsup/>
- Site internet de SAS : <http://www.sas.com>
- Site intéressant pour des exemples de graphiques : <http://support.sas.com/sassamples/graphgallery/index.html>
- Document : Guide d'introduction à SAS - 21-04-2006.pdf