

# Introduction à SAS



*Programmation & Logiciels Statistiques*

Cours 4

# Etape proc

## Généralités

- Une fois les données lues dans un tableau de données SAS, les procédures SAS peuvent être utilisées pour les analyser
- L'étape **proc** est un bloc d'instructions permettant de spécifier quel tableau de données doit être analysé et de quelle façon
- Une étape **proc** commence par une instruction **proc** et se termine par une instruction **run** ou lorsque l'étape **data** ou **proc** suivante commence

# Etape proc

## Généralités

- L'instruction **proc** spécifie la procédure à utiliser et éventuellement des **options** d'analyse
- L'option la plus importante est l'option **data =** qui spécifie le tableau de données à utiliser
- Si cette option est omise, la procédure utilise la table SAS **la plus récemment créée**
- On omet donc souvent cette option, **bien qu'il soit plus prudent de toujours l'explicitier**

# Etape proc

## Généralités

- On doit faire particulièrement attention lorsque plusieurs procédures sont exécutées ou lorsque l'une ou plusieurs d'entre elles produisent un tableau de données
- La plupart des instructions suivant une instruction **proc** sont spécifiques à la procédure correspondante et seront vues au fur et à mesure sur des exemples
- Toutefois, un certain nombre sont générales et s'appliquent à **diverses** procédures

# Etape proc

## Généralités

- L'instruction **var** permet de spécifier les variables utilisées par l'étape **proc**
- Par exemple,

```
proc print data = wgtclub;  
    var nom equipe perte;  
run;
```

Permet de restreindre l'affichage aux trois variables mentionnées, alors que par défaut toutes les variables seraient affichées.

# Etape proc

## Généralités

- L'instruction **where** permet de spécifier les observations à utiliser. Le mot-clef **where** est suivi d'une condition logique et seules les observations pour lesquelles la condition est vérifiée sont prises en compte par la procédure

```
proc print data = wgtclub;  
    where perte > 16;  
run;
```

# Etape proc

## Généralités

- L'instruction **by** permet d'analyser les données par groupes
- Les observations sont groupées selon les valeurs de la variable spécifiée par l'instruction **by** et des analyses distinctes sont effectuées pour chaque groupe
- Pour ce faire, les données doivent **d'abord être triées** selon la variable **by**

# Etape proc

## Généralités

```
proc sort data = wgtclub;  
    by equipe;  
proc means;  
    var perte;  
    by equipe;  
run;
```

# Etape proc

## Généralités

- L'instruction `class` est utilisées par diverses procédures pour spécifier des variables à utiliser comme variables de classification ou facteurs.
- Les variables en question peuvent être de type numérique ou caractère et ne peuvent prendre typiquement qu'un nombre restreint de modalités.

# Etape proc

## Statistique descriptive

- **Print** : permet de lister les données avec des totaux et sous-totaux
- **Means/Summary** : calcul de statistiques descriptives sur un jeu de données, éventuellement par groupe d'observations. Summary est identique à Means sauf qu'elle ne produit, par défaut, aucun résultat dans la fenêtre Output
- **Freq** : fournit la répartition des individus selon une ou deux variables qualitatives

# Etape proc

## Statistique descriptive

- **Tabulate** : Propose les statistiques des procédures Means/Summary et Freq dans un tableau dont la mise en forme est plus facilement contrôlable
- **Report** : combine les possibilités de Print et Tabulate
- **Univariate** : Reprend les sorties de Means ou Summary et ajoute d'autres statistiques, ainsi que des graphiques
- **Kde** : estimation non-paramétrique de la densité d'une variable quantitative

# Etape proc

## Tests statistiques

- **Freq** : tests de liaison entre deux variables qualitatives (Chi-2, Cramer), tests de tendance (Cochran-Armitage)
- **Corr** : Calcul des corrélations entre variables quantitatives et test de la nullité du coefficient de corrélation
- **Univariate** : Propose des tests de normalité et de nullité de la moyenne
- **Ttest/Glm** : Comparaison des moyennes de groupes d'observation à l'aide de tests statistiques. La procédure Ttest se limite à l'étude

# Etape proc

## Tests statistiques

de deux groupes, alors que Glm autorise les études à plusieurs groupes et propose davantage de tests. Les conditions requises par Glm (égalité des variances) sont toutefois plus contraignantes que celles de Ttest. Dans les deux procédures, les tests sont paramétriques.

- **Npar1way** : tests non-paramétriques de comparaison de moyennes de deux groupes ou davantage

# Etape proc

## Analyse de la variance

- **Glm/Anova** : permet l'étude de l'impact de facteurs qualitatifs et quantitatifs *contrôlés* sur une variable quantitative. La procédure Glm reprend et enrichit les possibilités de la procédure Anova, qui est plus ancienne. Cette dernière est plus rapide dans le cas de Plans d'Expériences Equilibrés.

# Etape proc

## Analyse de la variance

- **Glm/Mixed** : permet l'étude de l'impact de facteurs qualitatifs et quantitatifs, *contrôlés et non contrôlés*, sur une variable quantitative. La procédure Glm propose une instruction random dont les fonctionnalités sont plus étendues dans la procédure Mixed, plus récente et conçue explicitement pour ce type d'analyses.

# Etape proc

## Graphiques

- **gchart** : permet de tracer des histogrammes, des diagrammes en bâtons et des diagrammes circulaires
- **(s)gplot** : permet de tracer des nuages de points, des courbes et des nuages où les points sont de tailles variables
- **Gbarline** : permet de superposer un histogramme et une courbe
- **Boxplot** : permet de tracer des « boîtes à moustaches »

# Etape proc

## Echantillonnage

- **Surveyselect** : permet de tirer des échantillons d'observations à partir d'une table SAS en proposant de nombreux modes d'échantillonnage

# Etape proc

## Préparation des données

- **Rank** : remplace les valeurs par leur rang (1 pour le minimum) ce qui autorise des analyses non paramétriques
- **Stdize** : centre et réduit une ou plusieurs variables quantitatives
- **Transreg/glmmod** : construit des indicatrices (variables 0/1) à partir de variables qualitatives. La procédure Transreg construit des macro-variables contenant la liste des indicatrices créées et gère mieux labels et formats que Glmmod.

# Etape proc

## Valeurs manquantes

- **Stdize** : permet de remplacer les données manquantes par une moyenne, éventuellement la moyenne d'un groupe d'observations si on utilise l'instruction **by**
- **Mi/Mianalyze** : permet de remplacer les valeurs manquantes par des valeurs simulées pour l'analyse statistique. Ces deux procédures fonctionnent de pair (Mi en amont et Mianalyze en aval de l'analyse)

# Etape proc

## Régression

- **Reg/Glm** : permet l'estimation d'un modèle de régression linéaire. Etude et prévision des valeurs d'une variables quantitative en fonction de variables quantitatives seulement (proc Reg) ou des variables de toutes natures (proc Glm, plus récente : General Linear Models)
- **Robustreg** : permet d'effectuer une régression robuste, i.e. qui minimise l'impact des valeurs aberrantes, par diverses méthodes

# Etape proc

## Régression

- **Pls** : permet d'effectuer une régression PLS (celle-ci permet de prendre en compte des facteurs corrélés)
- **Genmod/Catmod** : permet d'estimer des modèles de régression pour des variables dont la distribution n'est pas normale (Gamma, Log-normale, Poisson,...etc) (Generalized Linear Models). La procédure Catmod est consacrée aux variables qualitatives
- **Calis** : permet d'estimer des modèles à équations structurelles (variables latentes)

# Etape proc

## Données de Survie

- **Lifereg** : estimation d'un modèle de régression à partir de données censurées à droite (données de survie)
- **Lifetest** : calcul d'indicateurs usuels sur des données de survie, dont la fonction de survie et l'estimateur de Kaplan-Meier
- **Phreg/Tphreg** : Estimation d'un modèle de régression pour des données de survie avec une hypothèse de hasards proportionnels. La procédure tphreg permet de prendre en compte des variables explicatives qualitatives

# Etape proc

## Analyses Factorielles

- **Princomp** : résultats et aides à l'interprétation de l'Analyse en Composantes Principales
- **Corresp** : résultats et aides à l'interprétation d'une analyse des correspondances simple (AFC) ou multiple (ACM)
- **Factor** : résultats et aides à l'interprétation de diverses méthodes d'analyse factorielle dont l'ACP et l'Analyse en Facteurs

# Etape proc

## Typologies

- **Cluster** : résultats et aides à l'interprétation d'une CAH (Classification Ascendante Hiérarchique)
- **Tree** : permet d'obtenir le dendrogramme à l'issue d'une CAH
- **Fastclust** : résultats et aides à l'interprétation d'une classification par centres mobiles (k-moyennes ou nuées dynamiques)

# Etape proc

## Modèles décisionnels

- **Logistic/Catmod** : permet d'estimer un modèle de régression logistique, où une variable qualitative est expliquée par des variables quantitatives ou qualitatives. La proc Logistic reprend les fonctionnalités de la proc Catmod (plus ancienne) et les enrichit de sorties plus orientées vers le Data Mining
- **Discrim** : résultats et aides à l'interprétation d'une analyse discriminante
- **Stepdisc** : sélection de modèle pour l'analyse discriminante

# Etape proc

## Plans d'expériences

- **Plan** : Construction de plans d'expériences factoriels complets, plans en blocs, carrés latins
- **Factex** : construction de plans d'expérience factoriels, complets ou fractionnaires
- **Optex** : construction de plans d'expériences optimaux, pour divers critères d'optimalité

# Etape proc

## Séries temporelles

- **Arima** : estimation et prévision de modèles arima
- **Hpf/Forecast** : permet la prévision de données quantitatives suite à l'analyse d'une série temporelle par diverses méthodes. La proc Hpf reprend de nombreuses fonctionnalités de la proc Forecast, plus ancienne.
- **Spectra** : analyse spectrale de séries temporelles

# Etape proc

## Autres méthodes

- Si une méthode statistique n'est disponible dans aucun des modules de SAS (STAT, OR, QC, INSIGHT, ETS, Enterprise Miner), il est possible de la programmer. La procédure **IML** (rattachée au module de même nom) propose un langage matriciel complet permettant de traiter les tables SAS et de produire des graphiques et des listings.

# Instructions globales

- Les instructions globales peuvent apparaître à n'importe quel point d'un programme SAS et **restent actives jusqu'à réinitialisation**
- L'instruction `title` est globale et spécifie un titre à afficher sur chaque page de sorties **et** sur chaque graphe jusqu'à réinitialisation

```
title "Analyse des données  
d'amaigrissement";
```

# Instructions globales

- Le texte à afficher doit être compris entre guillemets (simples ou doubles).
- Un titre sur plusieurs lignes peut être spécifié à l'aide de l'instruction `title2` pour la deuxième ligne, `title3` pour la troisième, etc jusqu'à 10.
- L'instruction `title` est synonyme de `title1`
- Les titres sont réinitialisés par une instruction du type `title2;`

# Instructions globales

- Cette instruction a pour effet de réinitialiser la deuxième ligne du titre **et toutes les lignes suivantes** i.e. `title3` etc.
- `title1`; aurait pour effet de réinitialiser tous les lignes de titres.
- Les **instructions de commentaire** sont des instructions globales en ce sens qu'elle peuvent apparaître à n'importe quelle point du programme

# Instructions globales

- Il y a deux forme d'instructions de commentaire
- La première forme, correspondant à un commentaire d'une seule ligne, commence par un **astérisque** et se termine par un **point-virgule**. Par exemple,  

```
*Ceci est un commentaire d'une seule ligne;
```

# Instructions globales

- La deuxième forme commence par /\* et se termine par \*/ et le commentaire peut s'étendre sur plusieurs lignes

```
/* Ceci est un  
   commentaire  
   sur plusieurs lignes  
*/
```

- Des commentaires de cette forme peuvent également apparaître sur la même ligne qu'une instruction SAS; par exemple,

# Instructions globales

```
perte = poids_deb-poids_fin; /* perte de  
poids */
```

- Les instructions globales **options** et **goptions** sont utilisées respectivement pour spécifier les options du système SAS et les options graphiques
- La plupart des options système sont en général laissées à leurs valeurs par défaut. Certaines des plus utiles sont les suivantes :

# Instructions globales

- **nocenter** : aligne les sorties à gauche au lieu de les centrer; particulièrement utile lorsque la ligne de sortie est plus large que l'écran
- **nodate** : supprime l'affichage de la date et de l'heure sur les sorties
- **ps = n** : permet de fixer la longueur des pages de sorties à n lignes
- **ls = n** : permet de fixer la longueur d'une ligne à n caractères

# Instructions globales

- `pageno = n` : permet de spécifier le numéro de page pour la page suivante de sorties, par exemple `pageno = 1` au début d'un programme exécuté de façon répétitive. Pour supprimer la numérotation, on peut utiliser `nonumber`.
- Plusieurs options peuvent être spécifiées à l'aide d'une seule instruction `options`, par exemple

```
options nodate nocenter nonumber;
```

# Instructions globales

- L'instruction `goptions` est analogue mais permet de spécifier les options graphiques.
- Quelques options graphiques utiles seront décrites dans la suite.

# Graphiques SAS

- Bien que des diagrammes et graphiques puissent être produits par SAS/BASE, il est nécessaire, pour obtenir des **graphiques à haute résolution** d'avoir recours aux procédures du module SAS/GRAPH.
- L'une des plus importantes est la procédure **gplot**.

# Graphiques SAS

## Procédure gplot

- L'usage le plus simple de la procédure gplot permet d'obtenir un nuage de point correspondant à deux variables, par exemple

```
proc gplot;  
    plot y*x;  
run;
```

- Diverses variations sur cette forme basique peuvent être obtenues en utilisant une ou plusieurs instructions `symbol` et en modifiant l'instruction `plot`

# Graphiques SAS

## Procédure gplot

```
symbol1 i = join;  
proc gplot;  
    plot y * x = 1;  
run;
```

- En modifiant l'instruction `plot` en `plot y*x=1;` on spécifie que les points doivent être représentés de la façon spécifiée par l'instruction `symbol1`
- L'instruction `symbol1` utilise l'option `i` (interpolation) pour spécifier que les points doivent être reliés

# Graphiques SAS

## Procédure gplot

- Chaque point est représenté par le symbole par défaut (+). L'option `v = (value =)` de l'instruction `symbol` peut être utilisée pour modifier ou supprimer ce symbole.
- Dans l'exemple précédent, si on veut faire en sorte que les points soient reliés par une ligne sans que ceux-ci soient représentés, on peut utiliser l'expression

```
symbol1 v = none i = join;
```

# Graphiques SAS

## Procédure gplot

- Une autre version de l'instruction plot permet d'utiliser une troisième variable pour représenter graphiquement des sous-ensembles des données
- En supposant que la variable sexe a pour valeurs 1 ou 2, alors

```
symbol1 v = square i = join;  
symbol2 v = triangle i = join;  
proc gplot;  
    plot y * x = sexe;  
run;
```

a pour effet de produire deux lignes avec différents symboles de points.

# Graphiques SAS

## Procédure gplot

- Une alternative consiste à supprimer les symboles de points et utiliser des **types de lignes différents** pour les deux sous-ensembles.
- L'option `l = (linetype = )` peut être utilisée pour ce faire, ce qui donne par exemple

```
symbol1 v = none i = join l = 1;
```

```
symbol2 v = none i = join l = 2;
```

```
proc gplot;
```

```
    plot y * x = sexe;
```

```
run;
```

- D'autres exemples seront donnés dans la suite.

# Graphiques SAS

## Procédure gplot

- Notons que les exemples précédents ne fonctionneront pas tels quels lorsque SAS est paramétré pour générer des **graphiques couleur**
- La raison en est que SAS utilisera le symbole spécifié par `symbol1` pour chacune des couleurs du graphique avant d'utiliser `symbol2`
- Si on souhaite obtenir un graphique en noir et blanc, la solution la plus simple consiste à spécifier `colours = (black)` dans `goptions`

# Graphiques SAS

## Procédure gplot

- Si on souhaite obtenir un graphique couleur, alors il est plus simple d'utiliser l'option **c = (colour = )** dans l'instruction **symbol**; par exemple

```
symbol1 v = none i = join c = blue;
```

```
symbol2 v = none i = join c = red;
```

```
proc gplot;
```

```
    plot y * x = sexe;
```

```
run;
```

# Graphiques SAS

## Procédure gplot

- Les instructions `symbol` sont globales et restent donc actives jusqu'à réinitialisation
- Pour réinitialiser une sinstruction `symbol1`, il suffit de faire `symbol1;`
- On peut réinitialiser l'ensemble des instructions `symbol` à l'aide de l'option `reset = symbol` de l'instruction `goptions`
- Dans la pratique, il est souhaitable d'inclure au début de chaque programme, l'instruction  
`goptions reset = all;`

# Graphiques SAS

## Procédure sgplot

- Depuis la version 9.2, SAS propose de nouvelles procédures graphiques plus sophistiquées : `sgplot`, `sgpanel`, `sgmatrix` et `sgrender` (les procédures traditionnelles sont `gplot`, `gchart`,...etc)
- Ces nouvelles procédures, en particulier `sgplot`, sont d'usage relativement simple et peuvent produire une grande variété de graphiques
- Elles peuvent suffire à la plupart des utilisateurs

# Graphiques SAS

## Procédure sgplot

- Pour représenter un nuage de points, on peut utiliser sgplot de la façon suivante

```
proc sgplot data=bodyfat;  
    scatter y=pctfat x=age;  
run;
```

- Pour d'autres types de graphiques, d'autres instructions que **scatter** sont utilisées; par exemple,

# Graphiques SAS

## Procédure sgplot

- **Series** : les points sont reliés par des lignes
- **Step** : les points sont reliés par des lignes en escaliers
- **Needle** : des traits verticaux sont tracés entre les points et l'axes des abscisses
- **Reg** : nuage de points avec une droite de régression
- **Loess** : régression par lissage loess
- **Pbspline** : régression par Beta splines pénalisés

# Graphiques SAS

## Procédure sgplot

- Pour les graphiques de type **Series** et **Step**, les points sont représentés dans l'ordre dans lequel ils apparaissent dans le tableau de données
- Il est donc en général nécessaire de trier les données selon la variable **x** avant de les représenter graphiquement

# Graphiques SAS

## Procédure sgplot

- On peut souhaiter distinguer différents sous-ensembles de données dans un nuage de points en utilisant des symboles de points différents et/ou différents types de traits
- Ceci peut être fait à l'aide de l'option **group =**
- Par exemple,

```
proc sgplot data=bodyfat;  
    scatter y=pctfat x=age / group=sex;  
run;
```

# Graphiques SAS

## Procédure sgplot

- Il est souvent utile de combiner l'information fournie par plusieurs graphiques en les superposant. `sgplot` permet de le faire automatiquement lorsque plus d'une instruction de tracé est spécifiée.
- Par exemple, pour comparer une droite de régression linéaire à un ajustement loess, on peut procéder de la façon suivante :

# Graphiques SAS

## Procédure sgplot

```
proc sgplot data=bodyfat;  
    reg y=pctfat x=age;  
    loess y=pctfat x=age / nomarkers;  
run;
```

- L'option **nomarkers** empêche que les points soient représentés deux fois, **sgplot** utilisant des symboles de points différents à chaque fois

# Graphiques SAS

## Procédure sgplot

- Des représentations graphiques sont souvent utilisées pour comparer des statistiques associées à des sous-ensembles de données.
- `sgplot` permet de représenter graphiquement des moyennes, des fréquences ou des sommes par des diagrammes en barres, en lignes ou en points.
- Les instructions correspondantes sont `vbar/hbar` et `vline/hline`, selon l'orientation souhaité, et `dot`

# Graphiques SAS

## Procédure sgplot

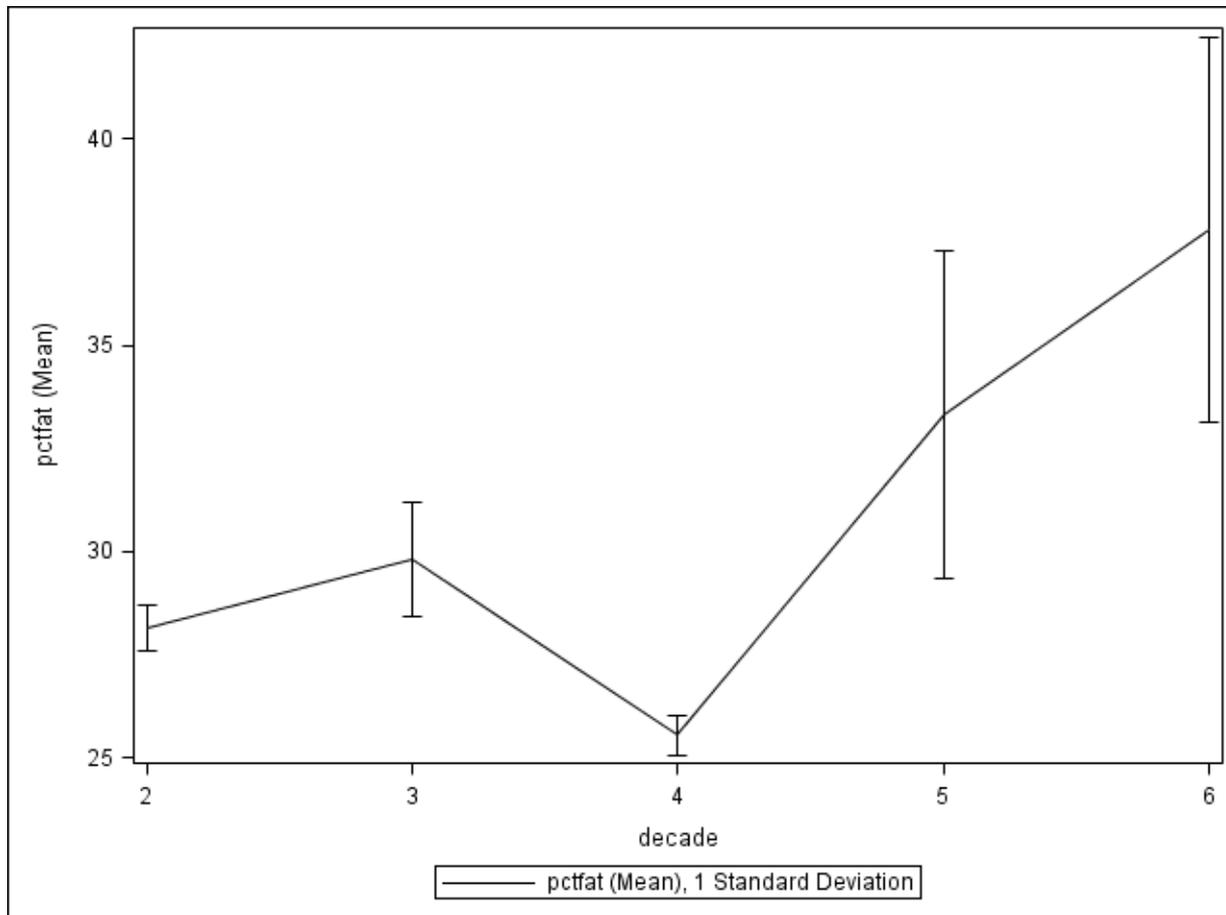
- Par exemple,

```
data bodyfat;  
    set bodyfat;  
    decade = int(age/10);  
  
run;  
  
proc sgplot data=bodyfat;  
    vline decade/response=pctfat stat=mean  
    limitstat=stdev;  
  
run;
```

produit le graphique suivant :

# Graphiques SAS

## Procédure sgplot



# Graphiques SAS

## Procédure sgplot

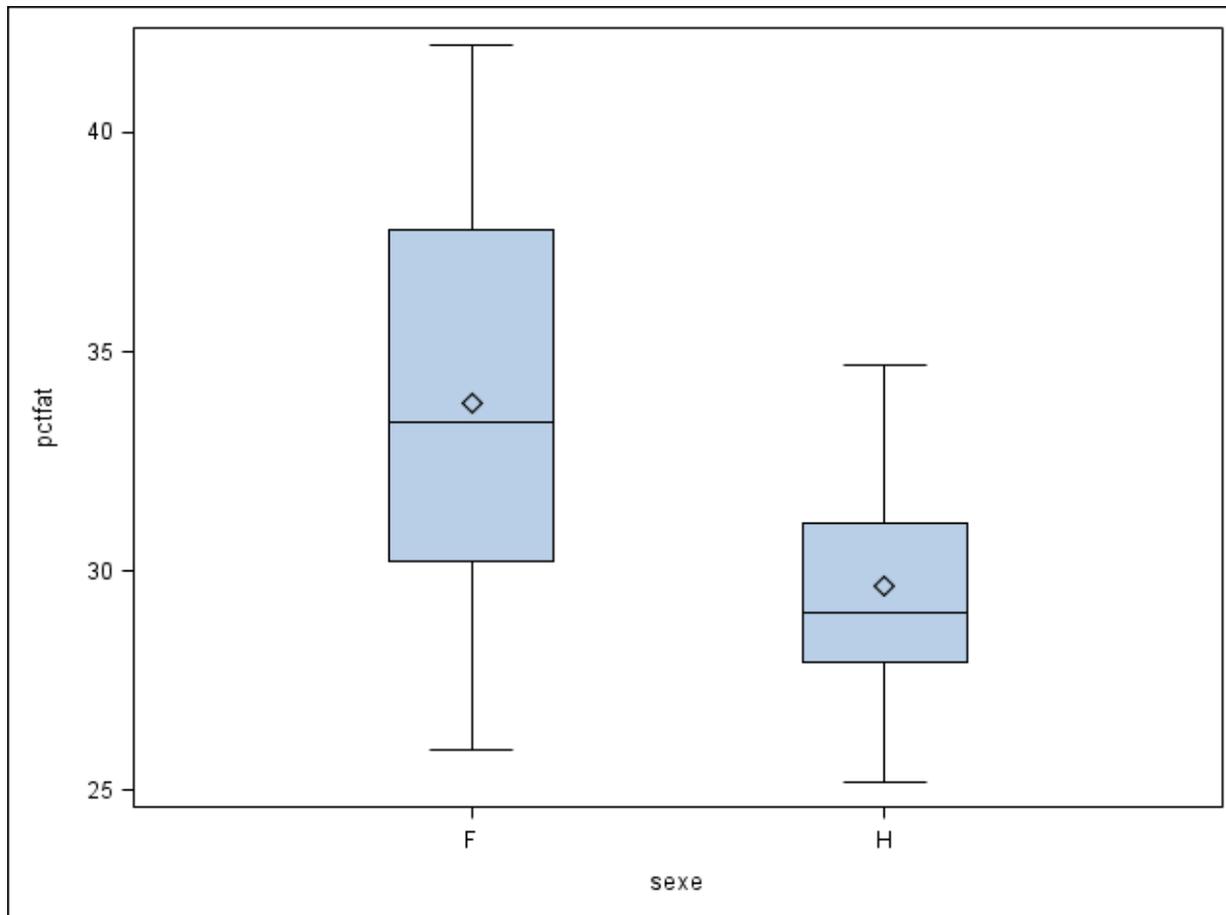
- Un autre type de graphique utile est le diagramme en boîte (boxplot)
- Celui-ci peut être obtenu avec **sgplot** à l'aide de l'instruction **vbox/hbox** (selon l'orientation souhaitée)
- Par exemple,

```
proc sgplot data=bodyfat;  
    vbox pctfat/category=sexe;  
run;
```

produit le graphique suivant :

# Graphiques SAS

## Procédure sgplot



# Graphiques SAS

## Procédure sgplot

- Notons qu'il existe également une procédure graphique spécifique, **boxplot**, permettant d'obtenir des diagrammes en boîte
- Celle-ci autorise davantage d'options

# Graphiques SAS

## Procédures `sgpanel`, `sgscatter`

- Les procédures `sgpanel` et `sgscatter` produisent toutes deux des graphiques multiples positionnés sur une grille
- Pour `sgpanel`, la grille est spécifiée par les valeurs de certaines variables de la table SAS. Ainsi, chaque graphique représente un sous-ensemble des données
- Pour `sgscatter`, chaque graphique représente la totalité des données et la grille consiste en un arrangement deux à deux des variables
- Des exemples seront vus en TD

# Graphiques SAS

## Affichage et sorties (SAS < 9.2)

- SAS permet d'utiliser un grand nombre de périphériques graphiques : imprimantes, traceurs et caméras par exemple
- L'option `device =` de l'instruction `goptions` est utilisée pour spécifier le type de périphérique utilisé pour visualiser les graphiques et les sauvegarder
- Afin de prévisualiser les graphiques sur un périphérique, par exemple l'écran, et les sauvegarder sur un autre périphérique, on peut spécifier le périphérique de prévisualisation à l'aide de l'option `device =` et le périphérique de sauvegarde à l'aide de l'option `targetdevice =`

# Graphiques SAS

## Affichage et sorties

- L'instruction suivante permet par exemple de visualiser les graphiques à l'écran et de les sauvegarder à l'aide d'une imprimante postscript au format paysage

```
goptions device = win target = ps
```

```
    colours = (black) rotate = landscape;
```

- Sous Windows **device = win** permet d'afficher les graphiques à l'écran.

# Graphiques SAS

## Affichage et sorties

- L'option `target = ps` spécifie que la sauvegarde doit être effectuée sur une imprimante postscript. `ps` est le nom du driver générique postscript et `target` une abbréviation de `targetdevice`
- Comme l'écran est en général en couleur, l'option `colours = (black)` permet d'éviter que SAS répète les définitions de symboles pour chaque couleur
- L'option `rotate` détermine l'orientation du graphique. L'alternative est `rotate = portrait`

# Graphiques SAS

## Affichage et sorties

- Lorsqu'une instruction `goptions` telle que celle vue précédemment est utilisée, les graphiques sont affichés **un par un** dans la fenêtre graphique et le programme pause entre deux graphiques en affichant le message « **Press forward to see next graph** » dans la ligne de commande
- Les touches PageUp et PageDown sont utilisées respectivement pour Forward et Backward

# Graphiques SAS

## Affichage et sorties

- Une sauvegarde peut être effectuée sur le périphérique de sauvegarde (target) à l'aide de la commande **Print** du menu **File** de la fenêtre graphique. Cela peut être fait alors que le programme est en pause ou après l'exécution de celui-ci
- Lorsqu'un programme génère un grand nombre de graphiques, **il peut être plus pratique de l'exécuter deux fois** : une première fois pour prévisualiser les graphiques et une deuxième fois pour les envoyer directement vers le périphérique de sauvegarde

# Graphiques SAS

## Affichage et sorties

- La première fois que le programme est exécuté, il peut par exemple commencer par

```
goptions reset = all;
```

```
goptions device = win colours = (black)  
rotate = landscape;
```

- La deuxième fois, il pourra commencer par

```
goptions reset = all;
```

```
goptions device = ps colours = (black)  
rotate = landscape;
```

# Graphiques SAS

## Affichage et sorties

- Afin d'obtenir la liste des périphériques graphiques autorisés par SAS et les noms des drivers correspondants, il suffit d'exécuter `proc gdevice;`
- Sous Windows, il existe quatre drivers supplémentaires (génériques) qui fournissent une alternative à l'utilisation de drivers spécifiques :

# Graphiques SAS

## Affichage et sorties

- `winprtm` : imprimantes monochromes
- `winprtg` : imprimantes à niveaux de gris
- `winprtc` : imprimantes couleur
- `winplot` : traceurs

# ODS

- L'ODS (Output Delivery System) est aujourd'hui une partie essentielle de SAS et cela pour trois raisons
  - La qualité des sorties obtenues à l'aide d'ODS
  - La sauvegarde des sorties dans des tables SAS
  - Les graphiques ODS

# ODS

- Un premier avantage d'ODS est qu'il permet de gagner du temps
- Quelque soit le format dans lequel les résultats d'une analyse doivent être publiés, ODS simplifie le processus en sauvegardant les sorties directement au format approprié
- Les formats les plus communs sont [html](#) et [rtf](#), mais d'autres possibilités incluent [xml](#), [latex](#) et [pdf](#)

# ODS

- Le format **rtf** est particulièrement adapté aux traitements de texte
- Chacun des formats précédents est appelé dans la jargon SAS une « **destination ODS** »
- Les sorties d'une ou plusieurs procédures peuvent être sauvegardées au format souhaité en commençant par **ouvrir** la destination ODS correspondante puis en la **refermant**

# ODS

- La destination `rtf` est ouverte par l'instruction `ods rtf;` et refermée par l'instruction `ods rtf close;` comme dans l'exemple suivant :

```
ods rtf;  
proc print data=bodyfat;  
proc corr data=bodyfat;  
run;  
ods rtf close;
```

# ODS

- Les sorties apparaissent normalement dans la fenêtre Output mais la version formatée est également sauvegardée dans un fichier `sasrtf.rtf` dans le répertoire courant
- Ce fichier étant écrasé à chaque nouvelle ouverture de la destination `rtf`, il est en général préférable de sauvegarder les sorties dans un fichier explicitement spécifié à l'aide de l'option `file="nomfichier"` de l'instruction `ods rtf`

# ODS

## Styles ODS

- Les sorties ODS peuvent être formatées selon divers styles pré-existants
- Chaque destination possède un style par défaut optimisé pour cette destination
- Le style par défaut de la destination `html` est `default` et celui de la destination `rtf` est `rtf`
- Les noms des autres styles disponibles peuvent être obtenus en exécutant

# ODS

## Styles ODS

```
proc template;  
    list styles;  
run;
```

- Des styles intéressants sont **theme** et **journal**, ce dernier étant en particulier recommandé pour des sorties en noir et blanc avec ombres ou remplissages en niveaux de gris

# ODS

## Sorties ODS

- Un deuxième avantage d'ODS est qu'il permet de sauvegarder les sorties sous la forme de tables SAS
- Avant la mise au point d'ODS, les sorties des procédures SAS (estimations, valeurs ajustées, résidus, etc) pouvaient être sauvegardées dans des tables SAS à l'aide de l'instruction **output** ou d'options spécifiques aux diverses procédures

# ODS

## Sorties ODS

- Dans le cadre du développement d'ODS, les sorties des différentes procédures ont été décomposées en différentes tables, chacune de ces tables pouvant être sauvegardée comme table SAS en insérant l'instruction

```
ods output table = tableSAS;
```

dans l'étape proc qui génère les sorties

- Les noms des tables générées par une procédure sont donnés dans la section « Details » de la documentation de la procédure
- Pour identifier les noms des variables, on peut utiliser l'explorateur ou

```
proc contents data = tableSAS;
```

# ODS

## Graphiques ODS

- La prise en charge des graphiques est un développement récent d'ODS
- De même que pour les tables ODS, l'information sur les graphiques ODS disponibles pour chaque procédure est fournie dans la section « Details » de la documentation correspondante
- Les graphiques ODS sont activés et désactivés par les instructions `ods graphics on;` et `ods graphics off;` respectivement

# ODS

## Graphiques ODS

- La production de graphiques ODS requiert également qu'une destination soit ouverte
- Par exemple,

```
ods html gpath="d:\data";  
ods graphics on;  
proc sgplot data=bodyfat;  
    vbox pctfat/category=sexe;  
run;  
ods graphics off;  
ods html close;
```

# ODS

## Graphiques ODS

- On aurait également pu utiliser `ods rtf`;
- Il y a toutefois une différence : avec `rtf`, les graphiques sont inclus dans le document rtf avec les tables éventuelles
- Avec `html`, chaque graphique est sauvegardé dans un fichier différent
- Par défaut, les sorties html et les graphiques sont sauvegardés dans le répertoire courant mais des répertoires différents peuvent être spécifiés à l'aide des options `path=` et `gpath=` de l'instruction `ods html`

# ODS

## Graphiques ODS

- Le format par défaut des graphiques est différent selon la destination ODS. Des possibilités offertes sont **gif**, **jpeg** et **png**
- Le format souhaité peut être sélectionné à l'aide de l'option **imagefmt=** de l'instruction **ods graphics**. Par exemple,

```
ods html gpath="d:\data";  
ods graphics on/imagefmt=jpeg;
```

# ODS

## Graphiques ODS

- Les graphiques ODS étant dirigés vers la destination ODS courante, ils sont formatés dans le même style que les sorties tabulaires
- De plus, la production de graphique ODS requiert un temps de traitement plus important, ce qui pose problème pour de grands tableaux de données
- Enfin, il est important de noter que les sorties produites par ODS pour la destination `rtf` utilise le format de pagination courant

# ODS

## Graphiques ODS

- Par conséquent, celui-ci doit correspondre à celui du document qu'on souhaite produire
- On peut spécifier ce format à l'aide du menu **File/Page Setup**, mais également à l'aide de l'instruction **options**; par exemple,

```
options papersize=a4 orientation=portrait;  
    bottommargin=2.5cm topmargin=2.5cm;  
    leftmargin=2.5cm rightmargin=2.5cm;
```

# ODS

## Graphiques ODS

- La valeur par défaut de papersize est letter
- Lorsque les sorties sont destinées à être incluses dans un document word, les options et instructions suivantes peuvent être utiles :

```
options nodate nonumber;  
ods noproctitle;  
ods rtf bodytitle;  
title;
```

# Labels de variables

- Bien que les noms des variables soient limités à 32 caractères sans espaces, les **labels** de variables peuvent contenir des espaces et être beaucoup plus longs : jusqu'à 256 caractères
- Si on a affecté un label à une variable, alors celui-ci peut être utilisé dans les sorties
- L'utilisation ou non des labels est spécifiée par les instructions `options label;` et `options nolabel;` respectivement. Par défaut, ceux-ci sont activés

# Labels de variables

- Pour affecter un label à une variable, on utilise l'instruction **label**. Par exemple,

```
label pctfat="Taux de graisse en % de la  
masse corporelle";
```

- Si une instruction **label** apparaît dans une étape **data**, alors le label est associé à la variable de façon permanente
- Si elle apparaît dans une étape **proc**, le label est utilisé uniquement pour les sorties de cette procédure

# Labels de variables

- Pour supprimer le label d'une variable, on utilise une instruction de la forme

```
label sexe='';
```

dans l'étape **data** ou **proc** correspondante.

# Labels de variables

## Formats SAS

- On peut utiliser les **formats** SAS pour affecter aux valeurs d'une variables des labels plus parlants.
- On utilise la `proc format` pour créer le format et l'instruction `format` pour associer le format à une variable. Par exemple,

# Labels de valeurs

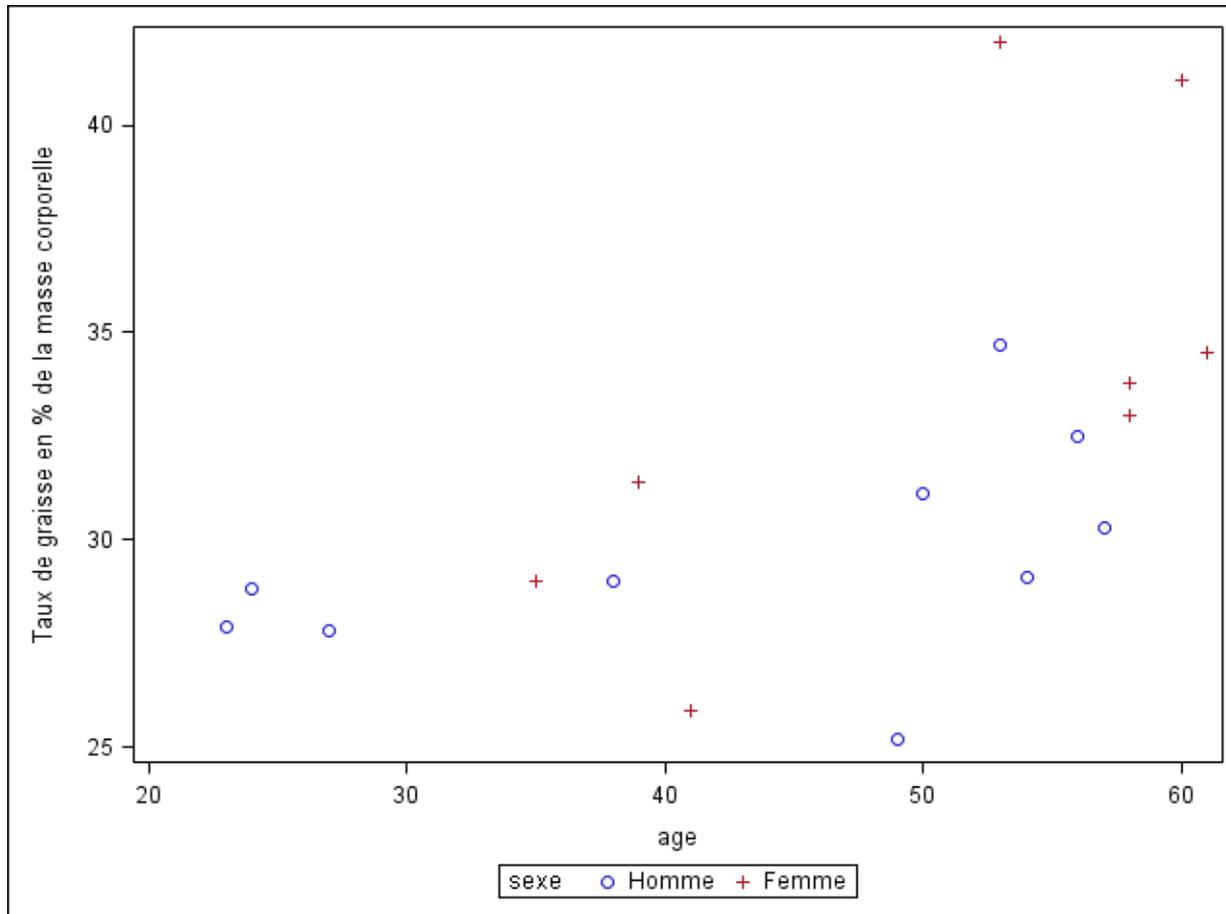
## Formats SAS

```
proc format;  
    value $sex 'H'='Homme' 'F'='Femme';  
run;  
proc sgplot data=bodyfat;  
    scatter y=pctfat x=age/group=sexe;  
    format sexe $sex.;  
    label pctfat='Taux de graisse en % de  
    la masse corporelle';  
run;
```

produit le graphique suivant :

# Labels de valeurs

## Formats SAS



# Labels de valeurs

## Formats SAS

- Il est possible d'associer le même format à **plusieurs** variables
- Par exemple, si diverses variables sont codées en 0/1 avec la signification Oui/Non, les instructions `value` et `format` correspondantes peuvent être

```
value on10f 0='Non' 1='Oui';
```

...

```
format q1-q20 on10f.;
```

# Astuces de programmation

## Prévenir et corriger les erreurs

- Utiliser une instruction par ligne
- Terminer chaque étape par une instruction `run`
- Indenter chaque instruction au sein d'une étape
- Spécifier le chemin complet pour les fichiers de données brutes dans l'instruction `infile`
- Commencer tout programme qui produit des graphiques par l'instruction `goptions reset = all;` puis spécifier les options requises (SAS < 9.2)

# Astuces de programmation

## Prévenir et corriger les erreurs

- Vérifier que chaque instruction se termine par un point-virgule
- Vérifier que tous les guillemets sont refermés
- Utiliser les couleurs de l'éditeur pour vérifier toute instruction qui ne commence pas par un mot-clef (bleu) ou un nom de variable (noir)
- Des blocs en violets peuvent indiquer que des guillemets n'ont pas été refermés
- Des blocs en vert peuvent indiquer qu'un `*/` est manquant

# Astuces de programmation

## Prévenir et corriger les erreurs

- Faire un « collapse » du programme pour vérifier sa structure globale. Pour ce faire, utiliser la combinaison de touches Ctrl/Alt/-
- Pour faire un « expand » du programme, utiliser la combinaison de touches Ctrl/Alt/+

# Astuces de programmation

## Prévenir et corriger les erreurs

- Après exécution, lire le log de façon à prendre connaissance des messages de warning et d'erreur
- Repérer le message « SAS went to a new line when INPUT statement reached past the end of line » lors de l'utilisation d'un **input** de mode liste

# Astuces de programmation

## Prévenir et corriger les erreurs

- Vérifier que le nombre d'observations et de variables lues est correct
- Vérifier le nombre de lignes lues et les longueurs minimales et maximales de lignes
- Afficher des petits sous-ensembles de données pour s'assurer que celles-ci ont été lues correctement
- Pour corriger des guillemets manquants, exécuter `";run;` ou `';run;` puis corriger le programme

# Références

- *A Handbook of Statistical Analyses Using SAS*, 3rd ed., G. Derr & B. S. Everitt, CRC Press, Chapitre 1
- *Introduction à SAS*, E. Duguet, Economica, Chapitres 1 à 5

# Pour aller plus loin...

- H. Kontchou-Kouomegni, O. Decourt. *SAS : Maîtriser SAS Base et SAS Macro*, Dunod, Chapitres 1 à 5
- S. Ringuedé. *SAS : Introduction au décisionnel Méthode et maîtrise du langage*, Pearson Education, Chapitres 1 à 8