Environnement

Table of contents

**Commandes et fonction introduites**  
*En italique, commandes externes*

| Sections | Commandes |
| --- | --- |
| **Menus et raccourcis** | **db** |
| **L’éditeur de programme** | **run** **do** **#delimit** **log** ***log2do2*** |

# 1. Les (principaux) types de fichier

* Les bases de données: **.dta**
* L’éditeur de programme : **.do** (**.ado** pour la programmation de commandes)
* Le log de session : **.smcl** ou **.log** (conversion possible en **.txt**, **.doc** ou **.pdf**)
* Les graphiques : **.gph** (+ enregistrement en format image: png, jpg, svf…)
* Les fichiers d’aides: **.sthlp**

# 2. Où passer ses instructions ?

* Dans l’éditeur de programme (.do ou .ado ou écritue d’un fichier d’aide)
* Dans la fenêtre “Command” de l’environnement principal
* Par les menus (ouverture d’une boîte de dialogue). On récupéré la syntaxe de l’instruction directement dans l’interface principale, et par copier-coller dans l’éditeur de programme. Plutôt pratique.

# 3. L’interface principale

|  |
| --- |
|  |

**Boite output**

Affiche les commandes exécutées et les résultats

**Boite review**

* C’est un log de toutes les instructions passées par la fenêtre command
* En cliquant de dessus, l’instruction est de nouveau affichée dans la fenêtre command
* Avec un clic-droit on peut copier, supprimer les instructions. Les instructions en rouge déclarent celles avec une erreur
* Les instructions passée par l’éditeur de programme ne sont reportée

**Boite variables**

* Liste les variables présentent dans la base chargée avec éventuellement leur label
* En cliquant dessus, là ou les variables sont affichées dans la fenêtre command
* Avec un cliqu-droit, une ou plusieurs variables peuvent être copiées, conservées ou supprimées

**Boite properties**

* Pour les variables affiche leurs propriétés: nom, label, format, type….
* Si le cadenat est délocké, on peut modifier ces propriétés (voir variable manager). Le code qui exécute la modification est affichée dans la fenêtre command et la boîte review

# 4. Menus et raccourcis

|  |
| --- |
|  |

**Menus (boîtes de dialogue)**

* File: gestion des fichiers => création, ouverture, import/export…
* Edit: utile pour la modification des préférences (couleurs interface, thèmes graphique)
* Data: manipulation des données
* Graphics: création de graphiques
* Statistics: instructions stat: descriptifs, modèles
* User: accès à vos boîtes de dialogue programmée en java
* Windows: activation des boîtes de l’interface principale
* help: infos sur le logiciel et accès à des ressources (internes et liens)  
   On peut accéder directement à une boîte de dialogue avec la commande db nom\_commande

db logit

**Raccourcis (dans l’ordre de gauche à droite)**

* Ouverture d’un fichier
* Enregistrement de la base de données active
* Impression de la fenêtre output
* Ouverture d’un log
* Ouverture de la fenêtre d’aide
* Ouverture de l’éditeur de graphique (modification d’un graphique créé)
* Création ou ouverture de l’éditeur de programme (.do .ado . sthlp)
* Ouvre la base de données active en mode édition
* Ouvre la base de données active en mode lecture
* Ouvre le variable manager
* Déblocage du défilement de l’output pendant l’exécution
* Arrête l’exécution d’un programme

# 5. L’éditeur de programme

|  |
| --- |
|  |

## 5.1 Soumettre un programme ou un bloc de programme

* On ne sélectionne aucune ligne: tout le programme est exécuté
* On sélectionne une partie du programme: seul le bloc est exécuté

|  |
| --- |
|  |

* **execute (do)**: les commandes et les résultats sont affichés dans l’output de l’interface principale
* **execute quietly (run)**: les commandes et les résultats ne sont pas affichés dans l’output de l’interface principale

|  |
| --- |
| do et run dans un programme |
| On peut exécuter un ou plusieurs programme enregistré dans un programme  do "path/nom\_prog.do" // avec affichage de l'output run "path/nom\_prog.do" // sans affichage de l'output |

## 5.2 delimiter

Par défaut une ligne = une instruction (**#delimit cr**).

Stata ne comprend pas la deuxième ligne de:

keep X1 X2   
 X3 X4

On peut contourner cela dans un programme pour écrire des instructions longue sur plusieurs lignes avec l’instruction **#delimit ;**

#delimit ;  
keep X1 X2   
 X3 X4;  
   
tabulate X1;

Le délimiteur s’appliquera tant qu’on ne repasse pas à #delimit cr

#delimit ;  
keep X1 X2   
 X3 X4;  
   
#delimit cr   
tabulate X1

Le délimiteur ; est sourtout utilisé pour les graphiques qui peuvent âtre assez gourmants en options (je le conseille).

La solution la plus utilisée est /// à la fin d’une ligne, à l’exception de la dernière de la commande

keep X1 X2 ///   
 X3 ///   
 X4  
  
tabulate X1

## 5.3 Les commentaires

Si le commentaire est sur une seule ligne

\* Commentaire

Si le commentaire est sur une ligne et suit une instruction

instruction // Commentaire

Si le commentaire est sur plusieurs lignes

/\* commentaire 1  
 commentaire 2 \*/

|  |
| --- |
| Pas d’autosave T\_T |
| Il n’y a pas d’autosave avec Stata contrairement à d’autres applications comme R. C’est très regrettable, car un plantage n’arrive pas toujours aux autres, et oublier de sauvegarder un programme n’est pas un évènement d’une extrême rareté. On peut sauver les meubles en générant un log de session en tête de programme ou mieux en générant un log à chaque ouverture de session dans un fichier profile [[marche à suivre](https://mthevenin.github.io/stata_programmation/profile/profile1.html#exemples)].   * Pour générer un log de session qui enregistrera en continu l’output de l’interface, et donc les lignes de commandes exécutée (sauf si exécution en mode run):   + **file** => **log** => **begin**, puis choisir un emplacement pour l’enregistrement du fichier *log*.   + **log using "path/nom\_log.smcl"** * commande externe **log2do2**: à partir d’un fichier log, permet de conserver seulement les lignes de programmes en supprimant les éléments de l’output. Attention les lignes comportant des erreurs seront également conservée. **ssc install log2do2** |

### 5.3.1 Interactions entre l’interface principale, les boîtes de dialogue et l’éditeur de programme

**DEMONSTRATION PENDANT LA FORMATION**