Frame

Table of contents

Les **frames** on été implémentées à Stata à partir de la version 16 (2019). Elles n’améliorent pas directement les durées d’éxécution. C’est indirectement, en donnant la possibiliter de travailler simultanément sur plusieurs en parallèle sans multiplier les opérations d’ouverture ou de sauvegarde qu’elles permettent de “gagner du temps”. Mieux encore, elles ont été pensées pour faciliter les fusions et les manipulations d’informations entre bases de données en implémentant une alternative aux *merge* via une logique de liaison.  
***Les frames ont été pensées pour faciliter le travail de programmation entre bases individuelles et bases contextuelles***.

La présentation des **frames** se fera par l’exemple, avec la base nhanes2:

* créer plusieurs frames à partir d’une base source (*nhanes2*).
* *collapser* des informations dans une frame
* lier deux frames et importer des variables d’une frame à une autre .
* générer des variables dans une frame en utilisant des informations issues d’une ou plusieurs autres *frame*.

Depuis Stata 16, à chaque base active est associée une *frame* nommée *default*. On peut récupérer l’information sur les frames créées lors d’une session avec les instruction **frame dir** ou **frame list**. Le nombre de lignes (observations) et colonnes (variables) sont indiquées.

On ouvre une session avec Stata, la frame \* default\* avec 0 ligne et 0 colonnes est indiquée avec frame dir.

frame dir

\* imc 2 x 2  
\* nhanes 10351 x 60; nhanes2.dta  
\* tension 2 x 2  
  
Note: Frames marked with \* contain unsaved data.

On ouvre une base, et de nouveau avec frame dir:

webuse nhanes2  
  
frame dir   
  
des, s

no; dataset in memory has changed since last saved  
r(4);  
  
end of do-file  
r(4);

On souhaite renommer cette frame *default*, par exemple *nhanes*: *frame rename*

frame rename default nhanes  
  
frame dir

no; dataset in memory has changed since last saved  
r(4);  
  
  
frame default not found  
r(111);  
  
end of do-file  
r(111);

# 1. **Création d’une frame**

On souhaite, par exemple, créer des bases d’indicateurs en collapsant des informations issues de la base/frame *nhanes*, sur la tension artérielle (*bpsystol*) et sur l’IMC (*bmi*), tout en gardant les informations sur le sexe (*sex*). On va générer deux frames avec **frame put**.

On indique simplement la sélection de variables et avec l’option **into()** le nom de la nouvelle frame.

frame put sex bpsystol, into(tension)  
frame put sex bmi, into(imc)  
  
frame dir

no; dataset in memory has changed since last saved  
r(4);  
  
  
frame tension already defined  
r(110);  
  
end of do-file  
r(110);

*Remarques*

* pas de préserve/restore et aucune opération de sauvegarde n’ont été nécessaire.
* frame dir indique les frames qui ne sont pas enregistrées en dur (.dta).
* On peut utiliser les arguments if in pour sélectionner des observations

# 2. **Opérations en parallèle**

On a maintenant 3 frames dans la session: la frame active *nhanes* (chargée dans l’interface), et en arrière plan *tension* et *imc*.  
On peut continuer à travailler sur la base *nhanes*, mais aussi simultanément sur les deux nouvelles frames.

Une seule ligne de commande:

frame nom\_frame: command

Plusieurs lignes de commandes:

frame nom\_frame {  
command1  
command2  
.  
.  
.  
}

\* frame active (nhanes2)  
tab agegrp  
  
frame tension: rename bpsystol tension  
frame tension: mean tension, over(sex)  
  
frame imc: rename bmi imc  
frame imc: mean imc, over(sex)

no; dataset in memory has changed since last saved  
r(4);  
  
  
  
 Age group | Freq. Percent Cum.  
------------+-----------------------------------  
 20–29 | 2,320 22.41 22.41  
 30–39 | 1,622 15.67 38.08  
 40–49 | 1,272 12.29 50.37  
 50–59 | 1,291 12.47 62.84  
 60–69 | 2,860 27.63 90.47  
 70+ | 986 9.53 100.00  
------------+-----------------------------------  
 Total | 10,351 100.00  
  
variable bpsystol not found  
r(111);  
  
end of do-file  
r(111);

Sur le même principe, on va collapser les frames et *tension* et *imcs* (moyennes de *tension* et *imc* par sexe)

frame tension {  
collapse tension, by(sex)  
list  
}  
  
frame imc {  
collapse imc, by(sex)  
list  
}

no; dataset in memory has changed since last saved  
r(4);  
  
 +-------------------+  
 | sex tension |  
 |-------------------|  
 1. | Male 132.8877 |  
 2. | Female 129.0679 |  
 +-------------------+  
  
 +-------------------+  
 | sex imc |  
 |-------------------|  
 1. | Male 25.50999 |  
 2. | Female 25.56256 |  
 +-------------------+

frame dir

no; dataset in memory has changed since last saved  
r(4);  
  
  
\* imc 2 x 2  
\* nhanes 10351 x 60; nhanes2.dta  
\* tension 2 x 2  
  
Note: Frames marked with \* contain unsaved data.

Remarque: la frame/base *nhanes* et toujours celle qui est chargée dans l’interface. On peut changer à tout instant de frame chargée dans l’interface avec **frame change nom\_frame**

# 3. **Liaison entre frames**

L’autre nouveauté est le système de liaison: - Il permet d’apparier des informations entre bases - Il permet de faire des opérations entre frames sans avoir à apparier des bases

Pour lier deux frames on utilise la commande **frlink**:  
- Nécessite une clé d’appariement, comme pour merge une ou plusieurs variables - On doit indiquer le type de liaison: **1:1** ou **m:1** - Stata génère une variable de liaison, on peut lui donner un nom - Petit regret: on ne peut lier que deux frames à la fois. Dans notre exemple, on aurait par exemple aimer lier la frame *nhanes* aux frames *tension* et *imc* en une ligne. On devra le faire deux fois, avec deux variables de liaison identiques.

frlink varlist, frame(nom\_frame) gen(link\_name)

Ici la clé d’appariement est la variable *sex*, la liaison est de type **m:1**

frlink m:1 sex, frame(tension) gen(l1)  
frlink m:1 sex, frame(imc) gen(l2)

no; dataset in memory has changed since last saved  
r(4);  
  
  
variable l1 already exists  
r(110);  
  
end of do-file  
r(110);

on souhaite apparier les variables *tension* et *inc* à la base/frame *nhanes*: **frget**. La variable de liaison joue le rôle de la clé d’identification.

Avantage:

* on merge des variables et non des bases. On sélectionne donc les variables que l’on juge utile
* on peut renommer les variables avec **frget**, ici *mtension* pour *tension* et *mimc* pour *imc*. Attention aux habitudes, ici pour renommer c’est nouveau\_nom=ancien\_nom.

frget mtension=tension, from(l1)  
frget mimc=imc, from(l2)

no; dataset in memory has changed since last saved  
r(4);  
  
  
 (1 variable copied from linked frame)  
  
 (1 variable copied from linked frame)

Les variables *mtension* et *minc* sont maintenant dans la base/frame *nhanes*

mean mtension mimc, over(sex)

no; dataset in memory has changed since last saved  
r(4);  
  
  
variable mtension not found  
r(111);  
  
end of do-file  
r(111);

Imaginons que nous ne souhaitions pas importer les variables *mtension* *minc* qui joueraient ici le rôle de variables contextuelles, mais mesurer simplement un écart entre les observations individuelles et les moyennes, en tenant compte du sexe de la personnes. On peut générer des variables sans appariement préalable avec une nouvelle fonction, ajoutée à **gen** et **replace**, **frval**.

**syntaxe** : **frval(variable\_liaison, nom\_variable)**

En amont je supprime les variables mtension et mimc de la frame *nhanes* (inutiles) et renomme les variables tension et imc dans leur frame respective (pour facilité la lecture de frval)

drop mtension mimc  
  
frame tension: rename tension mean\_tension  
frame imc: rename imc mean\_imc

no; dataset in memory has changed since last saved  
r(4);  
  
  
variable mtension not found  
r(111);  
  
end of do-file  
r(111);

Création des variables *diff\_tens* et *diff\_imc*

gen diff\_tens = bpsystol - frval(l1, mean\_tension)  
gen diff\_imc = bmi - frval(l2, mean\_imc)  
  
list diff\_tens diff\_imc in 1/10

no; dataset in memory has changed since last saved  
r(4);  
  
  
variable mean\_tension not found  
r(111);  
  
end of do-file  
r(111);

Dans une même création de variable, on peut utiliser **frval** pour plusieurs liaisons, et dans des expression conditionnelles. Si on souhaite ici créer une indicatrice qui repère les personnes dont la tension et l’imc sont supérieurs à la moyenne. Les variables *diff\_tens* et *diff\_imc* étant inutiles.

drop diff\_tens diff\_imc  
  
gen sup\_tens\_imc = bpsystol> frval(l1, mean\_tens) & bmi > frval(l2, mean\_imc)  
  
list sup\_tens\_imc in 1/10

no; dataset in memory has changed since last saved  
r(4);  
  
  
variable diff\_tens not found  
r(111);  
  
end of do-file  
r(111);

# 4. **Autres opérations**

**Vérifier la frame chargée dans l’interface**

**frame pwf** ou plus rapidement **pwf**

pwf

no; dataset in memory has changed since last saved  
r(4);  
  
  
 (current frame is nhanes)

**Changer de frame chargée dans l’interface**

Depuis le début, seule la frame *nhanes* était chargée dans l’interface. On peut à tout moment changer la frame chargée avec *frame change nom\_frame*.

frame change tension  
des  
  
frame nhanes: des, s  
  
frame change nhanes

no; dataset in memory has changed since last saved  
r(4);  
  
  
  
  
Contains data  
 Observations: 2   
 Variables: 2   
-------------------------------------------------------------------------------  
Variable Storage Display Value  
 name type format label Variable label  
-------------------------------------------------------------------------------  
sex byte %9.0g sex Sex  
tension float %9.0g (mean) tension  
-------------------------------------------------------------------------------  
Sorted by: sex  
 Note: Dataset has changed since last saved.  
  
  
Contains data from https://www.stata-press.com/data/r17/nhanes2.dta  
 Observations: 10,351   
 Variables: 60 20 Dec 2020 10:07  
Sorted by:   
 Note: Dataset has changed since last saved.

**Supprimer des frames**

* On peut supprimer à tout moment une frame qui n’est pas chargée dans l’interface avec **frame drop nom\_frame**. Si on applique cette commande à la frame chargée, Stata renvoie un message d’erreur.
* On peut supprimer toutes les frames avec **frame reset**. Dans ce cas plus aucune base ne sera chargée dans l’interface.

frame drop tension  
frame dir  
  
frame reset  
frame dir

no; dataset in memory has changed since last saved  
r(4);  
  
  
  
\* imc 2 x 2  
\* nhanes 10351 x 60; nhanes2.dta  
  
Note: Frames marked with \* contain unsaved data.  
  
  
 default 0 x 0

Il reste la frame “vide” *default* générée autotmatiquement par Stata.

**Création d’une frame**

On peut créer une frame *vide* à tout moment avec **frame create nom\_frame**. On peut également ajouter une liste de nom de variables: **frame create nom\_frame varlist**

frame create new  
frame dir

no; dataset in memory has changed since last saved  
r(4);  
  
  
  
\* imc 2 x 2  
 new 0 x 0  
\* nhanes 10351 x 60; nhanes2.dta  
\* tension 2 x 2  
  
Note: Frames marked with \* contain unsaved data.

frame create new2 x  
frame dir

no; dataset in memory has changed since last saved  
r(4);  
  
  
  
\* imc 2 x 2  
\* new2 0 x 1  
\* nhanes 10351 x 60; nhanes2.dta  
\* tension 2 x 2  
  
Note: Frames marked with \* contain unsaved data.

La frame *new2* à une colonne, mais aucune observation

frame new2 {  
set obs 1000  
replace x = rnormal()  
label variable x "x = random normal (0,1)"  
des  
}  
  
frame dir

no; dataset in memory has changed since last saved  
r(4);  
  
  
frame new2 not found   
r(111);  
  
end of do-file  
r(111);

**Ensemble du programme**

frame dir   
  
webuse nhanes2  
  
frame dir   
des, s  
  
frame rename default nhanes  
frame dir   
  
frame put sex bpsystol, into(tension)  
frame put sex bmi, into(imc)  
  
frame dir  
  
\* frame active (nhanes2)  
tab agegrp  
  
  
\* frame tension  
frame tension: rename bpsystol tension  
frame tension: mean tension, over(sex)  
  
\* frame imc  
frame imc: rename bmi imc  
frame imc: mean imc, over(sex)  
  
  
frame tension {  
collapse tension, by(sex)  
list  
}  
  
frame imc {  
collapse imc, by(sex)  
list  
}  
  
frame dir  
  
frlink m:1 sex, frame(tension) gen(l1)  
frlink m:1 sex, frame(imc) gen(l2)  
  
  
frget mtension=tension, from(l1)  
frget mimc=imc, from(l2)  
  
  
  
drop mtension mimc  
frame tension: rename tension mean\_tension  
frame imc: rename imc mean\_imc  
  
  
gen diff\_tens = bpsystol - frval(l1, mean\_tension)  
gen diff\_imc = bmi - frval(l2, mean\_imc)  
  
mean diff\_tens diff\_imc, over(sex)  
  
drop diff\_tens diff\_imc  
gen sup\_tens\_imc = bpsystol> frval(l1, mean\_tension) & bmi > frval(l2, mean\_imc)  
  
  
frame change tension  
des  
frame nhanes: des, s  
frame change nhanes  
  
frame drop tension  
frame dir  
  
frame reset  
frame dir  
  
frame create new  
frame dir  
  
frame create new2 x  
frame dir  
  
frame new2 {  
set obs 1000  
replace x = rnormal()  
label variable x "x = random normal (0,1)"  
des  
}  
  
frame dir