

# **Sesión 5**

# **Introducción a Stata IV**

**Juan D. Barón**  
juandbaron@gmail.com

**Métodos Cuantitativos de Economía Regional y Urbana**  
**Universidad Autónoma de Occidente**

**23 de mayo de 2011**

**Versión : 1.1**

**Copyright © 2011:** La reproducción total o parcial de este material está prohibida  
Material provisional y sujeto a cambios

# Organización de datos

- Organización de datos
- Comando **append**
- Comando **merge**
- Comando **collapse**

En esta sesión aprenderemos a combinar bases de datos y a generar bases de datos agregadas a partir de datos individuales

# 1. Combinando datos: **append**

## El Comando **append**

- El comando **append** combina bases de datos “**verticalmente**”
- Se carga una base de datos a la memoria y la otra se “pega” al final de la que está en memoria
- La base de datos en la memoria se llama “**master**” y la que se va a pegar se llama “**using**” o datos a usar

one.dta		
a	b	c
1	2	3
4	5	6

+

two.dta		
a	b	c
7	8	9
10	11	12
13	14	15

```
. use one, clear  
. append using two
```

Datos en memoria		
a	b	c
1	2	3
4	5	6
7	8	9
10	11	12
13	14	15

## Comando **append** con bases de datos con las **mismas variables**

```
use one, clear
```

```
. list
```

```
+-----+
|  a    b    c |
|-----|
1. |  1    2    3 |
2. |  4    5    6 |
+-----+
```

```
. use two, clear
```

```
. list
```

```
+-----+
|  a    b    c |
|-----|
1. |  7    8    9 |
2. | 10   11   12 |
3. | 13   14   15 |
+-----+
```

```
. use one, clear
```

```
. append using two
```

```
. list
```

```
+-----+
|  a    b    c |
|-----|
1. |  1    2    3 |
2. |  4    5    6 |
3. |  7    8    9 |
4. | 10   11   12 |
5. | 13   14   15 |
+-----+
```

**CONTINUA SIGUIENTE COLUMNA...**

## Comando **append** con bases de datos con **diferentes variables**

- Cuando las bases no contienen exactamente las mismas variables, el comando **append** pega las observaciones adecuadamente y crea **valores inexistentes** (**missing values**) para las que no están en ambas bases de datos

one.dta		
a	b	c
1	2	3
4	5	6

+

two.dta		
a	b	d
7	8	9
10	11	12
13	14	15

```
. use one, clear  
. append using two
```

Datos en memoria			
a	b	c	d
1	2	3	.
4	5	6	.
7	8	.	9
10	11	.	12
13	14	.	15

## 2. Base de datos **master** y **using**

En el ejemplo anterior (abajo también):

- La base de datos **one.dta** es la “**master**”  
(es la que **está en memoria** cuando se hace el **append**)
- La base de datos **two.dta** es la “**using**”  
(es la que **está guardada en el disco duro** cuando se hace el **append**)

one.dta		
a	b	c
1	2	3
4	5	6

+

two.dta		
a	b	d
7	8	9
10	11	12
13	14	15

```
. use one, clear  
. append using two
```

Datos en memoria			
a	b	c	d
1	2	3	.
4	5	6	.
7	8	.	9
10	11	.	12
13	14	.	15



### 3. Combinando datos: **merge**

## El comando **merge**

- “**merge**” combina dos bases de datos “**horizontalmente**”, basándose en una variable(s) **identificadora** de las observaciones
- Varias formas de combinar dos bases de datos de esta manera: **1:1** , **1:m** , **m:1** y **m:m** (la m es de múltiple)
- Útiles solo dos: **1:1** y **m:1**
- En esta notación el **primer término** se refiere a la base de datos **master** y la **segunda** a **using**

**1:1** el identificador de la observación en ambas bases de datos es **único** (ej. en una base tenemos los nombres de las personas con su edad y en la otra tenemos el nombre con la estatura. Solo tenemos un nombre por persona en cada base de datos)

**m:1** el identificador es **único** para las observaciones en la base de datos **using**, pero **no necesariamente** único en la **master** (ej. en una tenemos varias observaciones por persona (incluyendo región) y en la otra tenemos variables a nivel de región, PIB)

- Cada vez que se utiliza el comando **merge**, en la base de datos resultante se crea una **nueva variable**
- La variable creada por el comando merge se llama “**\_merge**” y puede tomar tres códigos (números)
- Los números nos indican si el **id** se encontró en una sola de las bases de datos (y si es así en cuál), o en ambas

Código	Equivalencia	Descripción
1	master	Solo aparece en <b>master</b>
2	using	Solo aparece en <b>using</b>
3	ambos	Aparece en <b>ambas</b>
Significado de los valores que toma la variable <b>_merge</b>		

### **3.1. Combinando de datos 1 a 1 (con variable identificadora)**

## Merge 1:1

- el identificador de la observación en ambas bases de datos es **único** (aparece una sola vez)
- Queremos combinar la información de la misma persona que existe en las bases de datos **one.dta** y **two.dta**

```
. use one, clear  
. merge 1:1 id using two
```

One.dta			Two.dta		Base de datos en memoria				
id	edad	mujer	id	peso	id	edad	mujer	peso	_merge
1	22	0	1	130	1	22	0	130	3
2	56	1	2	180	2	56	1	180	3
5	17	0	4	110	5	17	0	.	1
					4	.	.	110	2

Base “**master**”  
(piense **\_merge==1**)

Base “**using**”  
(piense **\_merge==2**)

**\_merge == 3** (estaba en “**ambas**”)  
**\_merge == 1** (solo en “**master**”)  
**\_merge == 2** (solo en “**using**”)

## En Stata: mirando que hay en las bases de datos

```
. use one.dta, clear
```

```
. list
```

	id	edad	mujer
1.	1	22	0
2.	2	56	1
3.	5	17	0

```
.  
. use two.dta, clear
```

```
. list
```

	id	peso
1.	1	130
2.	2	180
3.	4	110

## Haciendo el merge en Stata por la variable identificadora id

```
. clear all

. use one.dta, clear

. merge 1:1 id using two.dta
```

Result	# of obs.
not matched	2
from master	1   (_merge==1)
from using	1   (_merge==2)
matched	2   (_merge==3)

```
. list
```

	id	edad	mujer	peso	_merge
1.	1	22	0	130	matched (3)
2.	2	56	1	180	matched (3)
3.	5	17	0	.	master only (1)
4.	4	.	.	110	using only (2)

## Si queremos quedarnos solo con las personas para quienes tenemos información completa haríamos:

```
. keep if _merge == 3  
(2 observations deleted)
```

```
. list
```

```
  +-----+  
  | id   edad   mujer   peso   _merge |  
  +-----+  
1. |   1     22       0    130  matched (3) |  
2. |   2     56       1    180  matched (3) |  
  +-----+
```

```
. drop _merge
```



## 3.2. Combinando datos 1:m o m:1 (con variable identificadora)

## Merge m:1

- Combinar **una** observación en una base de datos a **múltiples** observaciones en la otra base de datos
- Ej.** Tengo los resultados del ICFES para los estudiantes de cada colegio, y quiero incorporar información de colegio (mixto, ciudad)

```
. use estudiantes.dta, clear  
. merge m:1 idc using colegios.dta
```

Estudiantes.dta		
idc	idp	ICFES
01	1	46
01	2	69
01	3	25
02	1	41
02	2	38

+

Colegios.dta		
idc	mixto	ciudad
01	0	2
02	1	2
03	1	1

=

Base de datos en memoria					
idc	idp	ICFES	mixto	ciudad	_merge
01	1	46	0	2	3
01	2	69	0	2	3
01	3	25	0	2	3
02	1	41	1	2	3
02	2	38	1	2	3
03	.	.	1	1	2

Base “**master**”  
(piense **\_merge==1**)

Base “**using**”  
(piense **\_merge==2**)

**\_merge == 3** (estaba en “**ambas**”)  
**\_merge == 1** (solo en “**master**”)  
**\_merge == 2** (solo en “**using**”)

## **4. Asegurándonos de que los identificadores son únicos**

- Aunque el comando **merge** les informará si los identificadores son únicos, en muchas ocasiones es indispensable saber si realmente lo son antes de realizar el merge
- En Stata hay varias formas de **verificar** si los **identificadores** son **únicos** (o en general si cualquier variable toma valores únicos)

1. Usando el comando **assert** con **quietly** y **bysort**:

```
. quietly by id: assert _N == 1
```

2. Usando **codebook**

```
. codebook id, compact
```

3. Usando el comando **duplicates**

```
. duplicates report id
```

4. Usando el comando **isid**

```
. isid id
```

- En general, **una sola variable identifica las observaciones** (i.e. la cédula, el nombre, o variables creadas a partir de éstas)
- En algunos casos, sin embargo, **toma más de una variable para identificar las observaciones únicas** (e.j. Encuestas de hogares y otras de las que veremos)
- Ejemplo: En la **Encuesta Continua de Hogares** algunos de los archivos tienen un **identificador de hogar** y un **número de orden** en ese hogar
- El de persona no es único a través de los hogares
- Si queremos usar la **información de persona** debemos **usar** los respectivos **comandos con las dos variables**.
- Si queremos ver que los identificadores de persona son únicos en este caso, usamos:

```
. quietly bysort famid orden: assert == _N
```

## 5. Usando **append** y **merge**

- **Append** and **merge** son muy útiles para combinar bases de datos
- Son **útiles** aún si solo tenemos **una base de datos**
- Suponga que tengo la base de datos de la **izquierda** (**master.dta**) y quiero tener la base de datos organizada como la de la **derecha**

Variable	Descripción
fid	Identificador de familia
casa	1 si tiene casa
h_edad	Edad del esposo
h_ing	Ingreso esposo
m_edad	Edad de la esposa
m_ing	Ingreso esposa

Variable	Descripción
fid	Identificador de familia
per	0 si esposa, 1 si esposo
casa	1 si tiene casa
edad	Edad
ing	Ingreso

**5 variables y dos obs. por pareja**


**6 variables y una obs. por pareja**

## Master.dta


fid	casa	h_edad	h_ing	m_edad	m_ing
1	0	34	1739	32	182
...	...	...	...	...	...
107	1	52	215	54	368

## maridos.dta

fid	per	casa	edad	ing
1	1	0	32	182
...	...	...	...	...
107	1	1	54	368



```
. use master, clear  
. rename h_edad edad  
. rename h_ing ing  
. drop m_edad m_ing  
. gen per = 1  
. save maridos
```



```
. use master, clear  
. rename m_edad edad  
. rename m_ing ing  
. drop h_edad h_ing  
. gen per = 0  
. save esposas
```

fid	per	casa	edad	ing
1	0	0	34	1739
...	...	...	...	...
107	0	1	52	215

## esposas.dta



### maridos.dta

fid	per	casa	edad	ing
1	1	0	32	182
...	...	...	...	...
107	1	1	54	368

### esposas.dta

fid	per	casa	edad	ing
1	0	0	34	1739
...	...	...	...	...
107	0	1	52	215

- . use maridos, clear
- . append using esposas

### Combinada.dta

fid	per	casa	edad	ing
1	1	0	32	182
...	...	...	...	...
107	1	1	54	368
1	0	0	34	1739
...	...	...	...	...
107	0	1	52	215

## 6. Generando bases de datos a nivel agregado (**collapse**)

- En múltiple ocasiones tenemos **información bastante desagregada (individuos, empresas)** y queremos **generar agregados** por alguna otra variable
- **E.j.:** tenemos información individual (y conocemos la ciudad donde viven) y queremos sacar alguna estadística agregada para cada ciudad
- **E.j:** tenemos información de ingresos de personas (pertenecientes a hogares), y queremos una base de datos de los ingresos promedio (o totales) por hogar
- Estas tareas se logran con el comando **collapse**

- La base de datos **SABER11\_2009.dta** contiene información de los **resultados** del componente de **matemáticas** para todos los estudiantes que tomaron la prueba SABER 11 en el 2009
- También contiene información de **género**, **estrato** y **departamento** (para Atlántico, Bogotá y Santander), además de un identificador único de persona
- Queremos una base de datos que tenga el valor **promedio** de los **resultados** de **matemáticas** para cada departamento

```
. collapse (mean) res_mate, by(dep_col)
```
- Queremos una base de datos que tenga el valor **promedio** de los **resultados** de **matemáticas** por género:

```
. collapse (mean) res_mate, by(genero)
```

## La base contiene 112.074 observaciones

Contains data from D:\Datos\My  
Dropbox\TallerUTB\Slides\ArchivosAdicionales\SABER11\_2009.dta

obs: 112,074  
vars: 5 27 Apr 2011 15:51  
size: 2,577,702 (99.2% of memory free) (\_dta has notes)

---

	storage	display	value	
variable name	type	format	label	variable label
<hr/>				
id	long	%9.0g		
genero	str1	%9s		
estrato	str1	%9s		
dep_col	str9	%9s		
res_mate	float	%9.0g		

---

Queremos una base de datos que tenga el valor **promedio** de los **resultados de matemáticas** para cada departamento

```
. collapse (mean) res_mate, by(dep_col)
. list
```

```
+-----+
| dep_col  res_mate |
+-----+
1. | ATLANTICO    43.90939 |
2. |      BOGOTA    45.81212 |
3. | SANTANDER     46.46029 |
+-----+
```

- Queremos una base de datos que tenga el **promedio** y el **valor máximo** del puntaje en **matemáticas** para cada género en cada depto

```
. collapse (mean) media = res_mate ///  
          (max) vlrmx = res_mate, by(dep_col genero)
```

### Notas:

- Se pueden sacar **varios estadísticos a la vez** (help collapse)
- Se pueden definir los **agregados** con **más de una variable** (género y departamento)
- **“///”**: Tres *slash* sirven para partir la línea de comando, **en un archivo do**, para que no quede tan larga (Stata la lee como una sola)
- Si no se usa la opción **by()**, collapse calcula el agregado para los tres departamentos (promedio de los resultados de matemáticas de todos los estudiantes en estos departamentos)

Queremos una base de datos que tenga el **promedio** y el **valor máximo** del puntaje en **matemáticas** para cada género en cada depto

```
. collapse (mean) media=res_mate (max) vlrmx=res_mate, by(dep_col genero)

. list
```

	genero	dep_col	media	vlrmx
1.	F	ATLANTICO	43.09215	115.59
2.	M	ATLANTICO	44.82618	115.59
3.	F	BOGOTA	44.48866	115.34
4.	M	BOGOTA	47.31806	115.59
5.	F	SANTANDER	45.05326	115.59
6.	M	SANTANDER	48.13642	115.59





# Resumen

Hasta aquí usted debería ser capaz de:

- Combinar múltiples bases de datos. Identificar cual es la base de datos **master** y cual la **using**
- Evaluar si los identificadores son únicos
- Generar bases de datos a nivel agregado