

17.801

Primavera de 2002

Utilización de los comandos **infile** e **infix** de STATA

STATA es un programa de gran flexibilidad que permite la lectura y manipulación de datos en diversas formas, lo que supone una ventaja, ya que los datos de ciencias sociales se presentan en distintos formatos y ello exige gran flexibilidad entre los paquetes estadísticos. Por desgracia, el manual de *STATA* que empleamos sólo explica la introducción de conjuntos de datos muy simples; incluso los ejemplos que muestra rayan en lo trivial. El objetivo de estas notas, por tanto, es introducirle en el manejo de los comandos **infile** e **infix** de *STATA* con un tratamiento algo más profundo que el del libro de Hamilton.

Un supuesto sencillo

Supongamos que tenemos los datos de cuatro alumnos que han realizado un test estándar. Tenemos sus nombres y edades, y sus puntuaciones en ambas pruebas (Test 1 y Test 2). Los datos presentados en forma de tabla son los siguientes:

Nombre	Edad	Test 1	Test 2
Bob	18	95	18
Carol	21	43	27
Ted	14	67	9
Alice	12	23	31

El modo más sencillo de pasar estos datos a *STATA* es activar el STATA Data Editor e introducir los datos en la interfaz de la hoja de cálculo.

Otro sistema un poco menos sencillo consiste en copiar los datos en un archivo y dejar que *STATA* los lea. Supongamos que su nombre de usuario en Athena es *janedoe*. Puede crear un archivo de datos con un editor de texto como *EMACS* y guardar los datos en un archivo de su directorio al que llame, por ejemplo, *scores.dat*. Este archivo tendrá la siguiente apariencia:¹

Ejemplo1

```
Bob 18 95 18
Carol 21 43 27
Ted 14 67 9
Alice 12 23 31
```

A continuación, y ya desde *STATA*, escriba:

¹Tenga en cuenta que todos los archivos que lee *STATA* **deben** terminar con un retorno de carro.

```
infile str5 nombre edad test1 test2 using /mit/janedoe/scores.dat
```

La palabra **infile** es el nombre del comando. Las palabras **nombre**, **edad**, **test1**, y **test2** son los nombres variables. **Los nombres variables en STATA deben tener un máximo de 32² caracteres y comenzar con una letra o un guión bajo (_).** STATA normalmente deduce que las variables contienen números.

Si los datos son no numéricos, es necesario decirle al programa que la variable es no numérica (una "cadena" de texto) e indicarle la longitud de ésta. Para eso está la palabra **str5** antes de la palabra **nombre**: para indicar que **nombre** es una cadena de texto que tiene un máximo de 5 caracteres.

Una vez haya escrito el comando **infile**, deberá introducir el comando **compress**. STATA tiene ciertos problemas de sensibilidad de la memoria de gestión, y de ahí el introducir este comando, para convertir todas las variables a sus representaciones internas más eficientes.

Un ejemplo con datos de campos fijos

El ejemplo anterior muestra el caso más simple de lectura de datos para su uso en STATA. Aparte de ser un conjunto de datos corto, podemos expresarlo "sin formato"; es decir, introduciendo libremente los datos en el ordenador sin más restricción que un espacio que separe los valores de las variables. Pero los conjuntos de datos rara vez son tan simples. Si, por ejemplo, uno de los nombres fuera "Mary Jane", tendría que prescindir del espacio y escribir MaryJane o Mary_Jane). Y si tuviera, no cuatro, sino cientos de observaciones y variables, los espacios necesarios para delimitar cada observación harían que el conjunto de datos creciera sin cesar, mucho más de lo necesario para contener la información original de los datos. Por esta y otras razones los conjuntos de datos se suelen organizar mediante "formato fijo". En este sistema de formato fijo, cada línea es el comienzo de una nueva observación³ y cada variable ocupa la misma columna (o las mismas columnas en cada línea).

La versión de los datos en formato fijo tendría una apariencia parecida a ésta:

Ejemplo 2

```
Bob 189518
Carol214327
Ted 1467 9
Alice122331
```

Para leer estos datos usaríamos el comando **infix** de STATA. Así, para leer los datos del Ejemplo 2, deberá escribir lo siguiente:

²En otras versiones de STATA (versiones 6 y anteriores) los nombres de variables tenían un límite máximo de 8 caracteres. Hemos mantenido esta restricción por razones de compatibilidad.

³Hay importantes excepciones que veremos más adelante.

```
infix str5 nombre 1-5 edad 6-7 test1 8-9 test2 10-11 using scores.dat
```

Observación sobre datos faltantes

En ocasiones ocurre que faltan datos en un conjunto. *STATA* puede indicar la falta de datos de tres formas: (1) periodos (2) ausencia de códigos de valores, y (3) espacios en blanco.

Periodos

STATA suele representar la falta de valores introduciendo un periodo en blanco en el lugar que correspondería al valor de la variable. Si, por ejemplo, Ted no nos hubiera revelado su edad, lo representaríamos situando un periodo en el lugar en el que debería ir el dato de su edad, bien sin formato:

```
Bob 18 95 18
Carol 21 43 27
Ted . 67 9
Alice 12 23 31
```

o bien en formato fijo:

```
Bob 189518
Carol214327
Ted .67 9
Alice122331
```

El programa excluirá a Ted de los cálculos o procedimientos para los que sea necesario el uso de la variable "edad".

Códigos de valor faltante

En la mayoría de los conjuntos de datos de ciencias sociales, éste es el sistema empleado para indicar la falta de un valor. Lo más habitual es asignar un valor absurdo a la variable cuando el verdadero valor de ésta falta, y a continuación dar datos al programa sobre esa variable. La edad, por ejemplo, debe ser un valor positivo, luego podemos hacer que el valor -1 represente que es un valor faltante, asignando a Ted una edad -1, y diciéndole a *STATA* lo que hemos hecho. Los datos aparecerían así:

```
Bob 189518
Carol214327
Ted -167 9
Alice122331
```

Existen dos formas en *STATA* de convertir un código de valor faltante en una representación real de ese valor. La más general es mediante el comando **replace**:

```
replace age=. if age== -1
```

Este comando indica a *STATA* que sustituya los valores de **edad** con la representación del valor faltante en aquellos casos en los que **edad** sea igual a -1.

Esta técnica resulta tediosa cuando se tiene un gran número de variables en las que falta el mismo código de valor. *STATA* dispone de un comando, **mvdecode**, que convierte los valores faltantes en su representación adecuada. Por ejemplo, si se ha usado el valor -1 para los datos faltantes de **edad**, **test1**, y **test2**, se podría introducir un único comando para acomodar los valores que faltan:

```
mvdecode age test1 test2, mv(-1)
```

Una lamentable limitación del comando **mvdecode** es que sólo funciona cuando las variables en cuestión usan un mismo código de valor faltante. Sobre todo en encuestas, los investigadores suelen emplear distintos códigos para indicar los distintos motivos por los que no existe el valor. En estos casos, es preciso ejecutar un comando **mvdecode** por separado para cada código de valor faltante, por lo que resulta más conveniente utilizar el comando **replace**.

Espacios en blanco

Cuando *STATA* encuentra, en un conjunto de datos de formato fijo, un espacio en blanco donde debería haber una variable, interpreta el valor de esa variable como faltante. (Pregunta: ¿por qué no ocurre esto con datos sin formato?) En el ejemplo, este conjunto de datos de formato fijo indicaría que falta el valor de la edad de Ted:

```
Bob 189518
Carol214327
Ted 67 9
Alice122331
```

¿Cuál es la mejor forma de indicar que faltan valores?

Conviene acostumbrarse a indicar los valores que faltan mediante códigos de valores faltantes, en vez de usar los otros dos métodos. El empleo de espacios en blanco es una invitación al error y al descuido, mientras que usar periodos funciona bien con *STATA*, pero no todos los paquetes estadísticos (ni todos los lenguajes de programación de alto nivel) emplean los mismos símbolos para los valores faltantes. Además, se corre el riesgo de confundir el periodo por un punto decimal y producir nuevos errores.

Conjuntos de datos multirregistro

Por lo general, los archivos de datos contienen una sola línea de datos para cada observación (o caso). Pero en ocasiones, hay tantos datos sobre cada observación que no entran (de modo legible) en una única línea. Cuando esto ocurre, es muy importante indicarle claramente a *STATA* cuántas líneas de datos forman un caso. De lo contrario, nos encontraremos con problemas muy graves.

Supongamos que en el conjunto de datos que venimos usando tenemos nombre, edad y las puntuaciones de los tests en la primera línea, y la nota media (GPA) en la segunda, del siguiente modo:

Ejemplo 3

```
Bob 189518
3.35
Carol214327
2.97
Ted -167 9
0.75
Alice122331
4.00
```

(La edad de Ted sigue faltando). Vemos que las variables siguen ocupando cada vez las mismas columnas, sólo cambia el que los datos de cada individuo ocupen también dos líneas. Se trata de leer "múltiples registros por caso" modificando el comando **infix** de dos formas. Primero, hay que decir a **infix** cuántas líneas de datos forman cada observación. Y segundo, hay que especificar la línea en la que se puede encontrar cada variable.

Para leer los datos del Ejemplo 3 introduciremos el siguiente comando:

```
infix 2 lines 1: str5 nombre 1-5 edad 6-7 test1 8-9 test2 10-11 2: gpa 1-4
```

La frase "**2 lines**" indica que cada observación comprende dos líneas de datos. "**1: str5 ... test2 10-11**" indica las variables que se encuentran en la primera línea. Del mismo modo, "**2: gpa 1-4**" indica que la variable gpa (nota media) se halla en la segunda línea, en las columnas 1-4.

Obsérvese que *no* es necesario leer las variables de cada línea. Si, por ejemplo, no necesitamos la variable gpa para el análisis que estamos haciendo, escribiremos:

```
infix 2 lines 1: str5 nombre 1-5 edad 6-7 test1 8-9 test2 10-11
```

y pasaremos por alto la segunda línea.