

Notas de clase – Taller de Stata¹
Clase 4 - Variables III - Fechas y tiempo

Contenido

1. ¿Cómo entiende *Stata* el tiempo?
2. ¿Cómo incluir fechas en *Stata*?
 - 2.1. A partir de un texto
 - 2.2. A partir de un número
3. ¿Cómo asignarle el formato a la variable de fecha?
4. ¿Cómo alternar entre tipos de fechas en *Stata*?
5. ¿Cómo trabajar con fechas en *Stata*?

[Estas notas se diseñaron con base en la versión 16 de *Stata*, conforme el software cambie las notas deben ser actualizadas].

¹ Estas notas están basadas en la guía desarrollada previamente para este curso por Rodrigo Azuero Melo, Nicolás de Roux, Luis Roberto Martínez, Román Andrés Zárate y Santiago Gómez Echeverry
Taller de Stata
Clase 4 – Variables III – Fechas y tiempo
Miguel Garzón Ramírez, Cristhian Acosta Pardo
Facultad de Economía, Universidad de los Andes

1. ¿Cómo entiende el tiempo *Stata*?

En *Stata* es posible definir variables de tiempo que van desde segundos hasta años. Estas variables son variables numéricas, donde el número registra el tiempo que ha transcurrido desde un periodo inicial. *Stata* toma como base para definir el tiempo el inicio del año 1960, ajustando la definición de periodo según la unidad de medida, como se puede ver en la [Tabla 1](#).

Tabla 1: Formatos de fecha e interpretación de una unidad de tiempo

Formato	Significado	Numérico e interpretación		
		Valor = -1	Valor = 0	Valor = 1
%tc	clock	31dec1959	01jan1960	01jan1960
		23:59:59.999	00:00:00.000	00:00:00.001
%td	days	31dec1959	01jan1960	02jan1960
%tw	weeks	1959w52	1960w1	1960w2
%tq	quarters	1959q4	1960q1	1960q2
%th	half-year	1959h2	1960h1	1960h2
%tg	generic	-1	0	1

Para poder leer los periodos de tiempo que trabaja *Stata* necesario emplear el comando `format`. La primera columna de la [Tabla 1](#) muestra que “%tx” es la sintaxis general de formatos de tiempo, donde *x* indica la unidad de tiempo.

2

Según lo anterior, si se asigna a la variable *Var1* el formato %td, un valor de 1 será interpretado por *Stata* como Enero 2 de 1960. Un valor de 4569 está asociado con Julio 5 de 1972, mientras que -4569 representa Junio 29 de 1947. Pero si el formato es %tw, un valor de 1 representa la segunda semana de 1960 y un valor de -725 corresponde a la cuarta semana de 1946.

El formato %tg es un formato genérico de tiempo que, al igual que los demás, registra el tiempo que ha pasado desde un periodo base pero en el que dicho tiempo no corresponde a una fecha calendario. Este formato se puede usar, por ejemplo, si se cuenta con datos del número de meses que una firma lleva operando o del número de días a partir de la fecha en que un paciente empieza un tratamiento. Se puede tratar también de divisiones de tiempo que no cuentan con un formato propio, como bimestres o quinquenios.

Además de los formatos %tx presentados en la [Tabla 1](#), existe el formato %ty. A diferencia de los formatos anteriores %ty toma como punto de partida el año 0 A.D., de tal forma que un valor de 1999 será interpretado bajo este formato como el año 1999.

2. ¿Cómo incluir fechas en *Stata*?

Para convertir una variable a fecha es necesario llevar a cabo un proceso en dos etapas:

- Crear una variable numérica cuyos valores representen fechas para *Stata*.
- Asignar a dicha variable el formato de fecha correspondiente.

2.1 ¿Cómo crear una variable numérica de fecha a partir de una variable de texto?

En muchos casos, las fechas están originalmente guardadas como variables de texto. Por ejemplo: “14/06/1984” (texto debido al backslash) o “Junio 14 de 1984”. *Stata* cuenta con funciones específicas a cada formato de fecha para llevar a cabo esta tarea. La [Tabla 2](#) muestra las funciones apropiadas para cada formato.

Tabla 2: Funciones para convertir Texto en Fecha

Formato	Función de conversión de texto a fecha
%tc	clock(string, mask [, toyear])
%tC	clock(string, mask [, toyear])
%td	date(string, mask [, toyear])
%tw	weekly(string, mask [, toyear])
%tm	monthly(string, mask [, toyear])
%tq	quarterly(string, mask [, toyear])
%th	halfyearly(string, mask [, toyear])
%ty	yearly(string, mask [, toyear])

La palabra *string* en la sintaxis hace alusión a la variable de texto relevante y *mask* hace referencia al orden de los componentes. La [Tabla 3](#) muestra las opciones disponibles para conformar el *mask* de la fecha. *mask* puede ser, por ejemplo, "DM20Yhm", en cuyo caso se le estaría indicando a *Stata* que la fecha viene en el orden: día, mes, año (en formato 20xx), hora y minuto. El símbolo # hace referencia a palabras al interior del string que se deban obviar, pero no se debe usar para referirse a separadores de la fecha (como / o -). De hecho, estas funciones de conversión de texto a fecha funcionan incluso si la fecha no cuenta con un separador al interior del string.

[, toyear] es una opción que se usa poco, pero es útil cuando los años están en dos dígitos y la variable incluye fechas de varios siglos (ejemplo: 98 y 06). [, toyear] le indica a *Stata* que convierta el año según corresponda. Por ejemplo, en el caso de cambio de siglo [, toyear] haría referencia al año 1999.

Tabla 3: Códigos disponibles para la opción *mask*

Código	Significado
M	month
D	day within month
Y	4-digit year
19Y	2-digit year to be interpreted as 19xx
20Y	2-digit year to be interpreted as 20xx
h	hour of day
m	minutes within hour
s	seconds within minute
#	ignore one element

Para interpretar el *mask*, *Stata* sigue 3 pasos:

1. Remover toda la puntuación en la fecha, excepto los puntos que separan segundos de centésimas y unidades más pequeñas. La puntuación removida es remplazada por espacios.
2. Insertar espacios cada vez que haya una letra seguida de un número o viceversa.
3. Aplicar la interpretación especificada.

Ej. 1 → “01dec2006 14:22EST” con mask “DMYhm#”

- i. “01dec2006 14 22EST”
- ii. “01 dec 2006 14 22 EST”
- iii. D=01, M=dec, Y=2006, h=14, m=22, omitir EST

Una vez se ha creado la variable numérica de fecha es necesario asignarle el formato correspondiente.

2.2 ¿Cómo crear una variable numérica de fecha a partir de un conjunto de variables numéricas?

Otra posibilidad es contar con varias variables que en conjunto indican la fecha. Por ejemplo, la variable *Y* es el año, *M* es el mes y *D* es el día. La [Tabla 4](#) muestra las funciones disponibles según la información disponible y el tipo de fecha deseada. Para que *Stata* no genere un error, las variables incluidas como argumento de la función deben incluir valores válidos. Por ejemplo, la variable de *día* debe tomar números enteros entre 1 y 31, mientras que la variable de *mes* debe tomar números enteros entre 1 y 12.

Tabla 4: Funciones para construcción de fechas a partir de variables numéricas

Formato	Función
%tc	mdyhms(<i>M,D,Y,h,m,s</i>)
%tc	dhms(<i>td,h,m,s</i>)
%tc	hms(<i>h,m,s</i>)
%tC	Cmdyhms(<i>M,D,Y,h,m,s</i>)
%tC	Cdhms(<i>td,h,m,s</i>)
%tC	Chms(<i>h,m,s</i>)
%td	mdy(<i>M,D,Y</i>)
%tw	yw(<i>Y,W</i>)
%tm	ym(<i>Y,M</i>)
%tq	yq(<i>Y,Q</i>)
%th	yh(<i>Y,H</i>)
%ty	<i>Y</i>

Stata permite que estas funciones se usen para agregar información adicional a una variable que ya está en formato de fecha, como la hora. Es a esta posibilidad a la que alude ‘td’ como posible input en la [Tabla 4](#).

Una vez se ha creado la variable numérica de fecha es necesario asignarle el formato correspondiente.

3. ¿Cómo asignar el formato a la variable de fecha?

5

En el caso de las fechas el comando format cumple dos funciones: Por una parte le indica a *Stata* las unidades de tiempo en que está la variable (%td, %tm, %tc, etc.); por otra parte le indica a *Stata* cómo se desea visualizar esa fecha. Esta segunda función es opcional y cada uno de los formatos de tiempo tiene una visualización predeterminada, como se muestra en la [Tabla 5](#).

Tabla 5: Visualización predeterminada de los distintos formatos de tiempo

Formato	Ejemplo
%tC	05jul1972 21:38:01
%tc	05jul1972 21:38:02
%td	05jul1972
%tw	1972w27
%tm	1972m7
%tq	1972q3
%th	1972h2
%ty	1972
%tg	(actual integer shown)

La sintaxis de formato sería entonces:

format varname %tc | %td | %tm...[código visualización]

Para el ejemplo de “05jul1972” en el caso de %td en la [Tabla 5](#), podemos especificarle a *Stata* mediante el código de visualización que muestre esa fecha como “July 5, 1972” o “7-5-72” o “5/7/1972”. Para introducir estas modificaciones debemos construir la combinación necesaria de códigos a partir de la lista de la [Tabla 6](#).

Ej. 2 →

Visualización: “July 5, 1972” Comando: format var %tdMonth_dd, CCYY

Visualización: “7-5-72” Comando: format var %tdnn-dd-YY

Visualización: “5/7/72” Comando: format var %tddd/mm/YY

Tabla 6: Listado de códigos de visualización de fechas en *Stata*

Código	Significado	Resultado	Código	Significado	Resultado
CC	century-1	36161	HH	hour	00-23
cc	century-1	36161	Hh	hour	00-12
YY	2-digit-year	00-99	hH	hour	0-23
yy	2-digit year	0-99	hh	hour	0-12
JJJ	day within year	001-366	MM	minute	00-59
jji	day within year	1-366	mm	minute	0-59
Mon	month	Jan, Feb,... Dec	SS	second	00-60 (sic, due to leap seconds)
Month	month	January, February,...December	ss	second	0-60(sic, due to leap seconds)
mon	month	jan, feb, ... dec	.s	tenths	.0-.9
month	month	january, february,...december	.ss	hundredths	.00-.99
NN	month	01-12	.sss	thousandths	.000-.999
nn	month	1-12	am	show am or pm	am or pm
DD	day within month	01-31	a.m.	show a.m. or p.m.	a.m. or p.m.
dd	day within month	1-31	AM	show AM or PM	AM or PM
DAYNAM	day of week	Sunday, Monday,(aligned)	A.M.	show A.M. or P.M.	A.M. or P.M.
E	day of week	Sunday, Monday,(unaligned)	.	display period	.
Dayname	day of week	Sunday, Monday,(unaligned)	,	display comma	,
Day	day of week	Sun, Mon, ...	:	display colon	:
Da	day of week	Su, Mo,...	-	display hyphen	-
day	day of week	sun, mon,...	—	display space	
da	day of week	su, mo,...	/	display slash	/
h	half	1-2	\	display backslash	\
q	quarter	1-4	!c	display character	c
WW	week	01-52	*	separator (see note)	
ww	week	1-52			

4. ¿Cómo alternar entre tipos de fechas en *Stata*?

Una vez se cuenta con una variable de fecha en formato *Stata*, es posible moverse entre tipos de fecha usando los comandos de la [Tabla 7](#). Por ejemplo, para pasar de día a mes se escribe:

Ej. 3 →

```
generate newvarname = mofd(variable con formato %td)
```

La razón por la que la tabla no está completa es porque %td es el formato base, de tal forma que toda transformación que no involucre a %td debe usar este formato como paso intermedio. Sin embargo, se puede hacer todo en un único comando. Por ejemplo, para pasar de mes a año se usaría la siguiente sintaxis:

Ej. 4 →

```
generate newvarname = yofd(dofm(variable con formato %tm))
```

Tabla 7: Funciones para conversión entre tipos de fecha

Desde	A...	%tc	%tC	%td	%tw	%tm	%tq	%th	%ty
%tc			Cofc()	dofc()					
%tC		cofC()		dofC()					
%td		cofd()	Cofd()		wofd()	mofd()	qofd()	hofd()	yofd()
%tw				dofw()					
%tm				dofm()					
%tq				dofq()					
%th				dofh()					
%ty				dofy()					

También es posible extraer parte de la información contenida en una fecha. Para el caso de variables en formato %td, la [Tabla 8](#) muestra las funciones disponibles. De nuevo, si la variable no está en formato %td se debe incluir una conversión a %td al interior del comando.

Tabla 8: Funciones para extraer información contenida en una fecha

Función	Resultado	Resultado if $d=td(05jul1972)$ (i.e. $d=4,569$)
$year(d)$	calendar year	1972
$month(d)$	calendar month	7
$day(d)$	day within month	5
$doy(d)$	day of year	187
$halfyear(d)$	half of year	2
$quarter(d)$	quarter	3
$week(d)$	week within year	27
$dow(d)$	day of week (o=Sunday)	3 (significa Wednesday)

5. ¿Cómo trabajar con fechas en *Stata*?

Siempre y cuando dos variables de tiempo estén en las mismas unidades estas se pueden sumar o restar. Por ejemplo, si las variables *fecha_despido* y *fecha_contrato* están en formato %td, se puede calcular el tiempo desempleado (en días) usando el comando:

Ej. 5 →

`generate t_desempleo = fecha_contrato - fecha_despido.`

También es posible aprovechar el formato de fecha al interior de expresiones. Por ejemplo, si se quiere borrar todas las observaciones antes de Julio 5 de 1972 de una base de datos no es necesario averiguar cuántos días hubo entre esa fecha y el primero de enero de 1960. Se puede usar el conjunto de funciones de la [Tabla 9](#) y emplear un comando como:

Ej. 6 →

`drop if var_fecha_en_formato_%td < td(5jul1972)`

El comando anterior muestra que el interior de las funciones de la [Tabla 9](#) no hace referencia al formato de la fecha de interés sino al orden de la misma.

Tabla 9: Funciones para convertir fechas a los números enteros asociados

Formato	Pseudofunción
%tc	$tc([day-month-year] hh:mm[:ss[:sss]])$
%tC	$tC([day-month-year] hh:mm[:ss[:sss]])$
%td	$td(day-month-year)$
%tw	$tw(year-week)$
%tm	$tm(year-month)$
%tq	$tq(year-quarter)$
%th	$th(year-half)$
%ty	none necessary: just type year
%tg	none necessary