

Cómo construir y personalizar tablas con Stata

Isabel Cañette

Principal Mathematician & Statistician
StataCorp LLC

Conferencia Stata México 2021

Introducción

La presentación efectiva de los resultados es una parte esencial (y muchas veces descuidada) del análisis estadístico.

La presentación de tablas adecuadas es un elemento importante dentro de esta comunicación.

Stata 17 incluye una nueva versión del comando **table** y el nuevo conjunto de comandos **collect**, herramientas que nos permiten crear tablas con flexibilidad, incluyendo los resultados que consideremos relevantes, en el orden que querramos, con la apariencia que querramos.

Con las nuevas funcionalidades de Stata para tablas, podemos crear tablas que desplieguen:

- Estadísticos de resumen
- Resultados de pruebas de hipótesis clásicas
- Resultados de regresiones
- Tests de post-estimación, medias marginales, efectos marginales
- Combinaciones de los anteriores
- Resultados retornados por cualquier comando (o combinaciones de comandos) de Stata.

Podemos personalizar nuestra tabla

- Disposición de la tabla
- Formatos numéricos
- Apariencia de las etiquetas en filas y columnas
- Asteriscos y otros textos adicionales
- Fuente, tamaño y color
- Sombreado, bordes, márgenes, alineamiento, y más

Podemos exportar nuestras tablas a

- Word
- Excel
- \LaTeX
- PDF
- Markdown
- HTML
- SMCL
- Plain text

Podemos incluir la tabla en reportes creados por **putdocx**, **putexcel**, or **putpdf**.

Descripción general

- Introducción a la nueva versión del comando **table**
- Introducción al nuevo conjunto de comandos **collect**
- Ejemplo: Tablas de resultados de regresión

Introducción al comando `table`

Usaremos algunas variables del conjunto de datos NHANES II

```
. use nhanesmod_sp, clear
```

```
(Subconjunto de NHANES 2)
```

```
. describe
```

```
Contains data from nhanesmod_sp.dta
```

```
Observations:      10.351
```

```
Subconjunto de NHANES 2
```

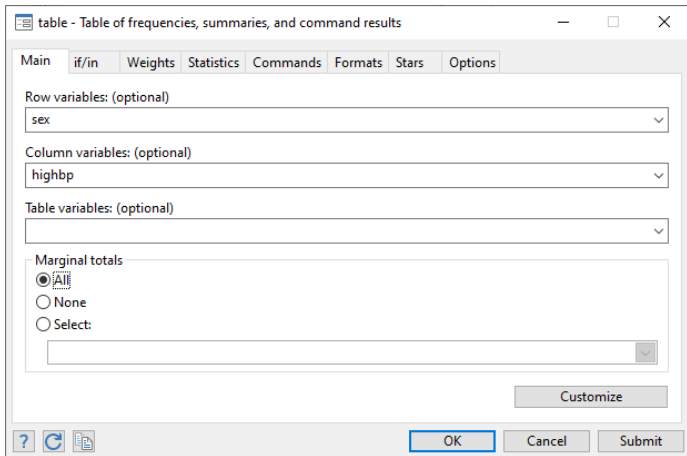
```
Variables:         4
```

```
15 Oct 2021 08:43
```

Variable name	Storage type	Display format	Value label	Variable label
edad	byte	%9.0g		Edad
rural	byte	%11.0g	labrural	Rural
grupo_edad	byte	%8.0g	labgrupoedad	Grupo etario
pa_alta	byte	%8.0g		Pres. arterial alta

```
Sorted by:
```


Las interfaz de usuario para **table** :



Elementos básicos

Sintaxis simplificada para **table**:

```
. table ( variables para filas ) ( variables para columnas )
```

Tablas de una vía

Por defecto, **table** construye tablas de frecuencia. Si especificamos sólo las filas (categorías determinadas por la variable **rural**), los resultados se despliegan en una sola columna.

```
. table (rural) ()
```

	Frequency
Rural	
Zona Urbana	6.548
Zona Rural	3.803
Total	10.351

Nota: en general, en Stata, para desplegar los números en formato en castellano, escribir `set dp comma`. (revertir con `set dp period`)

Tablas de una vía

Si especificamos sólo las columnas, los resultados se despliegan en una sola fila

```
. table () (rural)
```

	Zona Urbana	Rural Zona Rural	Total
Frequency	6.548	3.803	10.351

Tablas de dos vías

Tabulamos variable **rural** versus **grupo_edad**

```
. table (rural) (grupo_edad)
```

	Grupo etario						Total
	20-29	30-39	40-49	50-59	60-69	70+	
Rural							
Zona Urbana	1.631	981	777	810	1.743	606	6.548
Zona Rural	689	641	495	481	1.117	380	3.803
Total	2.320	1.622	1.272	1.291	2.860	986	10.351

Por defecto, la tabla muestra frecuencias (número de observaciones) por celda y totales por fila y columna

Tablas de dos vías

Podemos suprimir los totales con opción **nototals**

```
. table (rural) (grupo_edad), nototals
```

	Grupo etario					
	20-29	30-39	40-49	50-59	60-69	70+
Rural						
Zona Urbana	1.631	981	777	810	1.743	606
Zona Rural	689	641	495	481	1.117	380

Tablas de dos vías

Podemos desplegar sólo totales por fila o por columna.

```
. table (rural) (grupo_edad), totals(rural)
```

	Grupo etario						Total
	20-29	30-39	40-49	50-59	60-69	70+	
Rural							
Zona Urbana	1.631	981	777	810	1.743	606	6.548
Zona Rural	689	641	495	481	1.117	380	3.803

```
. table (rural) (grupo_edad), totals(grupo_edad)
```

	Grupo etario					
	20-29	30-39	40-49	50-59	60-69	70+
Rural						
Zona Urbana	1.631	981	777	810	1.743	606
Zona Rural	689	641	495	481	1.117	380
Total	2.320	1.622	1.272	1.291	2.860	986

Estadísticos de resumen

La opción **statistic()** nos permite tabular, además de frecuencias, otros estadísticos como medias, medianas, proporciones, etc.

```
. table ( variables filas ) ( variables columnas ),  
        statistic( statspec )
```

Tipos de argumentos para statistic()	
nombre genérico	uso
<i>freqstat</i>	estadísticos de frecuencia
<i>sumstat</i>	estadísticos de resumen
<i>ratiostat</i>	estadísticos de proporción

Estadísticos de resumen

statistic(*freqstat*) produce estadísticos de frecuencia.

<i>freqstat</i>	Estadístico
frequency	frecuencia (Número de observaciones)
sumw	suma de los pesos

Estadísticos de resumen

statistic(*sumstat varlist*) despliega estadísticos de resumen para las variables en *varlist*.

<i>sumstat</i>	Estadístico
<code>mean</code>	media
<code>semmean</code>	error estándar de la media
<code>sebinomial</code>	error estándar de la media, binomial
<code>sepoisson</code>	error estándar de la media, Poisson
<code>variance</code>	variancia
<code>sd</code>	desviación estándar
<code>skewness</code>	coeficiente de simetría
<code>kurtosis</code>	coeficiente de apuntamiento (kurtosis)
<code>cv</code>	coeficiente de variación

Estadísticos de resumen

<i>sumstat</i>	Estadístico
<code>count</code>	Número de valores no faltantes
<code>median</code>	mediana
<code>p#</code>	percentil #
<code>q1</code>	primer cuartil
<code>q2</code>	segundo cuartil
<code>q3</code>	tercer cuartil
<code>iqr</code>	rango intercuartil
<code>min</code>	valor mínimo
<code>max</code>	valor máximo
<code>range</code>	rango
<code>...</code>	primer valor, último, total, y otros

Estadísticos de resumen

`statistic(ratiostat [varlist])` despliega estadísticos de proporción.

<code>proportion</code>	proporción
<code>percent</code>	porcentaje
<code>rawproportion</code>	proporción ignorando pesos
<code>rawpercent</code>	porcentaje ignorando pesos

Estadísticos de resumen

Ejemplo con dos estadísticos en una tabla de una vía:

```
. table () (rural),
> statistic(frequency)
> statistic(percent)
```

	Zona Urbana	Rural Zona Rural	Total
Frequency	6.548	3.803	10.351
Percent	63,26	36,74	100,00

Estadísticos de resumen

Ejemplo con dos estadísticos en una tabla de dos vías:

```
. table (rural) (pa_alta),
> statistic(frequency)
> statistic(percent)
```

	Pres. arterial alta		
	0	1	Total
Rural			
Zona Urbana			
Frequency	3.798	2.750	6.548
Percent	36,69	26,57	63,26
Zona Rural			
Frequency	2.177	1.626	3.803
Percent	21,03	15,71	36,74
Total			
Frequency	5.975	4.376	10.351
Percent	57,72	42,28	100,00

Estadísticos de resumen

Podemos calcular proporciones a través de filas o columnas con subopción **across**(*varnames*)

```
. table (rural) (pa_alta),
> statistic(frequency)
> statistic(percent, across(pa_alta))
> totals(rural)
```

	Pres. arterial alta		
	0	1	Total
Rural			
Zona Urbana			
Frequency	3.798	2.750	6.548
Percent	58,00	42,00	100,00
Zona Rural			
Frequency	2.177	1.626	3.803
Percent	57,24	42,76	100,00

```
. table (rural) (pa_alta),
> statistic(frequency)
> statistic(percent, across(rural))
> totals(pa_alta)
```

	Pres. arterial alta	
	0	1
Rural		
Zona Urbana		
Frequency	3.798	2.750
Percent	63,56	62,84
Zona Rural		
Frequency	2.177	1.626
Percent	36,44	37,16
Total		
Frequency	5.975	4.376
Percent	100,00	100,00

Estadísticos de resumen

Agregamos medias y desviaciones estándar a la tabla

```
. table (rural) (pa_alta),
>     statistic(frequency)
>     statistic(percent)
>     statistic(mean edad)
>     statistic(sd edad)
>     nototals
```

	Pres. arterial alta	
	0	1
Rural		
Zona Urbana		
Frequency	3.798	2.750
Percent	36,69	26,57
Mean	41,14613	54,69127
Standard deviation	16,77735	15,2568
Zona Rural		
Frequency	2.177	1.626
Percent	21,03	15,71
Mean	43,94258	55,44895
Standard deviation	16,61634	14,29346

Cambio de formato del estadístico

Podemos cambiar los formatos de los estadísticos con la sintaxis:

```
. table ( variables filas ) ( variables columnas ),  
        nformat() sformat()
```

- `nformat()` - Formato numérico; escribir `help format` para ver las especificaciones disponibles.
- `sformat()` - Formato alfanumérico, donde `%s` se sustituye por el valor del estadístico dentro de un texto entre comillas. (por ejemplo para un precio, podemos escribir `sformat("$%s")`)

```

. collect clear
. table (rural) (pa_alta),
>     statistic(frequency)
>     statistic(percent)
>     statistic(mean edad)
>     statistic(sd edad)
>     nototals
>     nformat(%9.0fc frequency)
>     sformat("%s%" percent)
>     nformat(%6.2f mean sd)
>     sformat("%s" sd)

```

	Pres. arterial alta	
	0	1
Rural		
Zona Urbana		
Frequency	3.798	2.750
Percent	36,69%	26,57%
Mean	41,15	54,69
Standard deviation	(16,78)	(15,26)
Zona Rural		
Frequency	2.177	1.626
Percent	21,03%	15,71%
Mean	43,94	55,45
Standard deviation	(16,62)	(14,29)

Resultados de otros comandos

También podemos usar **table** para tabular resultados de otros comandos.

La sintaxis es:

```
. table ( rowspec ) ( colspec ),  
        command( cmdspec )
```

Resultados de otros comandos

Por ejemplo:

```
. table (...) (...),  
      command(correlate edad rural pa_alta)
```

correlate guarda las correlaciones en la matriz **r(C)**.

```
. correlate edad rural pa_alta
(obs=10.351)
```

	edad	rural	pa_alta
edad	1,0000		
rural	0,0568	1,0000	
pa_alta	0,3676	0,0074	1,0000

```
. return list
scalars:
           r(N) = 10351
           r(rho) = ,0567812234319929

matrices:
           r(C) : 3 x 3

. matrix list r(C)
symmetric r(C) [3,3]
           edad      rural      pa_alta
   edad           1
   rural    .05678122           1
   pa_alta  .36755099    .00739899           1
```

Construimos una tabla con la matriz de correlación

```
. table (rowname) (colname),
>       command(r(C): correlate edad rural pa_alta) nformat(%5.4f)
```

	Edad	Rural	Pres. arterial alta
Edad	1,0000	0,0568	0,3676
Rural	0,0568	1,0000	0,0074
Pres. arterial alta	0,3676	0,0074	1,0000

Introducción al uso de `collect`

Elementos básicos: Construir tablas con `collect`

- 1 Recolectar resultados de otros comandos: comando `—collect get`, prefijo `collect:`, o `table`
- 2 Explorar la colección—`collect dims`, `collect levelsof`, y `collect label list`
- 3 Definir las filas y las and columnas de la tabla—`collect layout` o `table`
- 4 Personalizar la tabla, especificar formatos, etiquetas, fuente, sombreado, y más—`collect label`, `collect style`, `collect stars`, . . .
- 5 Exportar la tabla a Word, Excel, \LaTeX , PDF, Markdown, HTML, SMCL, o simple texto—`collect export`
- 6 Grabar el estilo, etiquetas, y la colección utilizada para seguir modificando más tarde o para utilizar el estilo con otra colección—`collect label save`, `collect style save`, `collect save`

Interfaz para construir tablas (Tables Builder)

Tables Builder

Collection: Table

Dimensions

- High blood pressure (highbp)
- Sex (sex)
- sex#result
- Statistic: option variable (var)
- Across (across)
- Covariate names and column n...
- Command results index (cmds...
- Statistic/command option inde...
- Command option index (com...
- Table cell type (cell_type)
- Table border block (border_blo...

Levels

- Mean (mean)
- Standard deviation (sd)
- Frequency (frequency)
- Percent (percent)

Label and style dialogs

- Edit dimension labels
- Edit level labels
- Construct significance stars
- Compose row headers
- Compose column headers
- Compose table headers
- Show/hide header content
- Cell appearance styles
- Intercept position
- Show/hide coefficient styles
- Automatic dimension levels
- Recode dimension levels
- Remap tags

Rows

- sex#result

Columns

- highbp

Tables

Preview

	High blood pressure	
	0	1
Sex		
Male		
Frequency	2,611	2,304
Percent	25.22%	22.26%
Mean	42.86	52.59
Standard deviation	(16.97)	(15.88)
Female		
Frequency	3,364	2,072
Percent	32.50%	20.02%
Mean	41.62	57.62
Standard deviation	(16.60)	(13.26)

Paso 1: Recolectar resultados

El primer paso es poner los resultados de interés en una colección

- Prefijo **collect**—Recolecta los resultados de *command*
`. collect: command`
- **collect get**—Recolecta resultados específicos del comando previo
`. command`
`. collect get results`
- **table**—Pone los resultados en una colección automáticamente.

Comenzamos usando la colección que creamos con **table**.

```

. table (rural) (pa_alta),
>     statistic(frequency)
>     statistic(percent)
>     statistic(mean edad)
>     statistic(sd edad)
>     nototals
>     nformat(%9.0fc frequency)
>     sformat("%s%" percent)
>     nformat(%6.2f mean sd)
>     sformat("(%s)" sd)

```

	Pres. arterial alta	
	0	1
Rural		
Zona Urbana		
Frequency	3.798	2.750
Percent	36,69%	26,57%
Mean	41,15	54,69
Standard deviation	(16,78)	(15,26)
Zona Rural		
Frequency	2.177	1.626
Percent	21,03%	15,71%
Mean	43,94	55,45
Standard deviation	(16,62)	(14,29)

Paso 2: Explorar la colección

Los valores en una colección están organizados en lo que llamamos *tags*.

Una tag consiste en una dimensión (*dimension*) y un nivel (*level*) dentro de la dimensión. Una tag se especifica indicando dimensión y nivel (**dimension[level]**).

Valor	Tag 1	Tag 2	Tag 3
3.789	rural [1]	pa_alta [0]	result [frequency]
1,625	rural [1]	pa_alta [1]	result [frequency]
21,03	rural [1]	pa_alta [0]	result [percent]

Las dimensiones involucradas en la tabla son **rural**, **pa_alta**, y **result**, por lo que son parte de la colección, junto con otras dimensiones.

Paso 2: Explorar la colección

`collect dims` lista todas las dimensiones en una colección.

```
. collect dims
```

```
Collection dimensions
```

```
Collection: Table
```

	Dimension	No. levels
Layout, style, header, label		
	across	2
	cmdset	1
	colname	1
	command	1
	pa_alta	2
	result	4
	rural	2
	statcmd	4
	var	2
Style only		
	border_block	4
	cell_type	4

Paso 2: Explorar la colección

collect levelsof lista los niveles de la dimensión especificada.

```
. collect levelsof pa_alta
Collection: Table
Dimension: pa_alta
Levels: 0 1
```

collect label list lista los niveles y etiquetas asociadas a una dimensión.

```
. collect label list pa_alta, all
Collection: Table
Dimension: pa_alta
Label: Pres. arterial alta
Level labels:
    .m Total
    0
    1
```

Por defecto, la dimensión asociada a una variable y sus niveles heredan las etiquetas de la variable, pero las podemos cambiar.

Paso 3: Especificar filas y columnas

Hemos especificado filas y columnas con **table**. Desplegamos dicha disposición con **collect layout**.

```
. collect layout
Collection: Table
      Rows: rural#result
      Columns: pa_alta
      Table 1: 11 x 2
```

	Pres. arterial alta	
	0	1
Rural		
Zona Urbana		
Frequency	3.798	2.750
Percent	36,69%	26,57%
Mean	41,15	54,69
Standard deviation	(16,78)	(15,26)
Zona Rural		
Frequency	2.177	1.626
Percent	21,03%	15,71%
Mean	43,94	55,45
Standard deviation	(16,62)	(14,29)

Como estamos representando tres dimensiones (**pa_alta**, **rural** y **result** en filas y columnas, (en este caso) las filas contienen la combinación de dos dimensiones.

Paso 3: Especificar filas y columnas

collect layout también nos permite establecer la disposición

```
. collect layout (result) (rural#pa_alta)
```

```
Collection: Table
```

```
Rows: result
```

```
Columns: rural#pa_alta
```

```
Table 1: 4 x 4
```

	Rural			
	Zona Urbana		Zona Rural	
	Pres. arterial alta 0	1	Pres. arterial alta 0	1
Frequency	3.798	2.750	2.177	1.626
Percent	36,69%	26,57%	21,03%	15,71%
Mean	41,15	54,69	43,94	55,45
Standard deviation	(16,78)	(15,26)	(16,62)	(14,29)

Paso 3: Especificar filas y columnas

Podemos especificar los niveles que queremos tabular para las distintas dimensiones

```
. collect layout (result[mean sd]) (rural#pa_alta)
```

```
Collection: Table
```

```
  Rows: result[mean sd]
```

```
 Columns: rural#pa_alta
```

```
Table 1: 2 x 4
```

	Zona Urbana		Rural	
	Pres. arterial alta 0	1	Pres. arterial alta 0	1
Mean	41,15	54,69	43,94	55,45
Standard deviation	(16,78)	(15,26)	(16,62)	(14,29)

Volvemos a la tabla original:

```
. collect clear
. table (rural) (pa_alta),
>     statistic(frequency)
>     statistic(percent)
>     statistic(mean edad)
>     statistic(sd edad)
>     nototals
>     nformat(%9.0fc frequency)
>     sformat("%s%" percent)
>     nformat(%6.2f mean sd)
>     sformat("(%s)" sd)
```

	Pres. arterial alta	
	0	1
Rural		
Zona Urbana		
Frequency	3.798	2.750
Percent	36,69%	26,57%
Mean	41,15	54,69
Standard deviation	(16,78)	(15,26)
Zona Rural		
Frequency	2.177	1.626
Percent	21,03%	15,71%
Mean	43,94	55,45
Standard deviation	(16,62)	(14,29)

Paso 4: Personalizar la tabla

Con **`collect label dim`** especificamos la etiqueta para una dimensión.

Con **`collect label levels`** especificamos etiquetas para los niveles de una dimensión. Especificamos etiquetas de dimensión y niveles para **`pa_alta`**

```
. collect label dim pa_alta "Hipertensión", modify
. collect label levels pa_alta 0 "No" 1 "Sí"
. collect label list pa_alta, all

Collection: Table
Dimension: pa_alta
Label: Hipertensión
Level labels:
    .m Total
    0 No
    1 Sí
```

Paso 4: Personalizar la tabla

Especificamos etiquetas de dimensión y niveles para **rural**

```
. collect label dim rural "Zona", modify
. collect label levels rural 0 "Urbana" 1 "Rural", modify
. collect label list rural, all

Collection: Table
Dimension: rural
Label: Zona
Level labels:
    .m Total
    0 Urbana
    1 Rural
```

Paso 4: Personalizar la tabla

collect preview nos muestra la tabla con las nuevas etiquetas.

```
. collect preview
```

	Hipertensión	
	No	Sí
Zona		
Urbana		
Frequency	3.798	2.750
Percent	36,69%	26,57%
Mean	41,15	54,69
Standard deviation	(16,78)	(15,26)
Rural		
Frequency	2.177	1.626
Percent	21,03%	15,71%
Mean	43,94	55,45
Standard deviation	(16,62)	(14,29)

Paso 4: Personalizar la tabla

También podemos modificar las etiquetas de valores para los resultados; las desplegamos con **collect label list**

```
. collect label list result
Collection: Table
Dimension: result
Label: Result
Level labels:
frequency Frequency
mean Mean
percent Percent
sd Standard deviation
```

Paso 4: Personalizar la tabla

... y las modificamos con **collect label levels**

```
. collect label levels result frequency "Frecuencia"
>                               mean      "Media (edad)"
>                               percent   "Porcentaje"
>                               sd        "Desviación estándar (edad)",
>                               replace
. collect preview
```

	Hipertensión	
	No	Sí
Zona		
Urbana		
Frecuencia	3.798	2.750
Porcentaje	36,69%	26,57%
Media (edad)	41,15	54,69
Desviación estándar (edad)	(16,78)	(15,26)
Rural		
Frecuencia	2.177	1.626
Porcentaje	21,03%	15,71%
Media (edad)	43,94	55,45
Desviación estándar (edad)	(16,62)	(14,29)

Paso 4: Personalizar la tabla

Los subcomandos **collect style** nos permiten hacer muchos tipos de modificaciones en la apariencia de la tabla. Aquí removemos el borde a la derecha de los encabezados de fila.

```
. collect style cell border_block, border(right, pattern(nil))
. collect preview
```

	Hipertensión	
	No	Sí
Zona		
Urbana		
Frecuencia	3.798	2.750
Porcentaje	36,69%	26,57%
Media (edad)	41,15	54,69
Desviación estándar (edad)	(16,78)	(15,26)
Rural		
Frecuencia	2.177	1.626
Porcentaje	21,03%	15,71%
Media (edad)	43,94	55,45
Desviación estándar (edad)	(16,62)	(14,29)

Paso 5: Exportar la tabla

collect export exporta la tabla personalizada al formato que especificamos. `chosen format`.

Cuando exportamos a Word, **collect style putdocx** nos permite hacer cambios específicos para este formato.

```
. collect style putdocx, layout(autofitcontents)
. collect export mitabla.docx, replace
(collection Table exported to file mitabla.docx)
```

Paso 5: Exportar la tabla

The screenshot shows a Microsoft Word document with a table of hypertension data. The table is titled "Hipertensión" and is organized into two main sections: "Urbana" and "Rural". Each section lists four statistics: "Frecuencia", "Porcentaje", "Media (edad)", and "Desviación estándar (edad)". The columns are labeled "No" and "Sí".

Hipertensión		
	No	Sí
Zona Urbana		
Frecuencia	3.798	2.750
Porcentaje	36.69%	26.57%
Media (edad)	41,1461295	54,6912727
Desviación estándar (edad)	(16.78)	(15.26)
Zona Rural		
Frecuencia	2.177	1.626
Porcentaje	21.03%	15.71%
Media (edad)	43,9425815	55,4489545
Desviación estándar (edad)	(16.62)	(14.29)

Page 1 of 1 36 words Accessibility: Unavailable 120%

Paso 5: Exportar la tabla

Exportamos la tabla a \LaTeX ; la línea siguiente genera un archivo `.tex` que se puede compilar directamente para generar un archivo con la tabla.

```
. collect export mitabla0.tex
```

La opción **tableonly** nos permite exportar solamente el código que genera la tabla,

```
. collect export mitabla.tex, tableonly
```

así podemos incluir la tabla en otro archivo en el que estamos trabajando, por ejemplo:

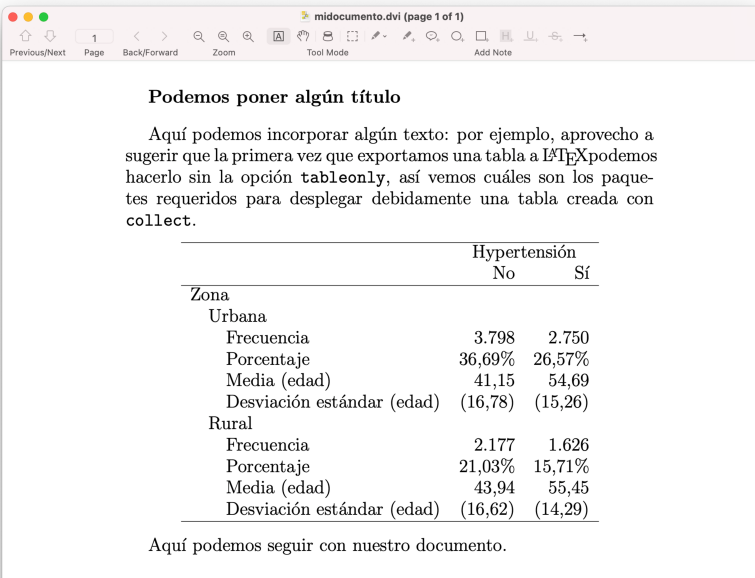
```
\documentclass{article}
% los siguientes paquetes se requieren para tablas
\usepackage{multirow}
\usepackage{amsmath}
\usepackage{ulem}
\usepackage[table]{xcolor}
% El siguiente paquete es para que LaTeX entienda Unicode
% así podemos usar el teclado en castellano.
\usepackage[utf8]{inputenc}

\begin{document}

\large{\bf Podemos poner algún título}
Aquí podemos incorporar algún texto:
(... texto omitido)
\vspace{2ex}

\begin{center}
\input{mitabla}
\end{center}

Aquí podemos seguir con nuestro documento.
\end{document}
```



midocumento.dvi (page 1 of 1)

Previous/Next Page Back/Forward Zoom Tool Mode Add Note

Podemos poner algún título

Aquí podemos incorporar algún texto: por ejemplo, aprovecho a sugerir que la primera vez que exportamos una tabla a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ podemos hacerlo sin la opción `tableonly`, así vemos cuáles son los paquetes requeridos para desplegar debidamente una tabla creada con `collect`.

Zona	Hipertensión	
	No	Sí
Urbana		
Frecuencia	3.798	2.750
Porcentaje	36,69%	26,57%
Media (edad)	41,15	54,69
Desviación estándar (edad)	(16,78)	(15,26)
Rural		
Frecuencia	2.177	1.626
Porcentaje	21,03%	15,71%
Media (edad)	43,94	55,45
Desviación estándar (edad)	(16,62)	(14,29)

Aquí podemos seguir con nuestro documento.

Ejemplo: Tabla de resultados de regresión

Queremos crear una tabla que incluya los resultados de una regresión logística como razón de momios (odds ratios) y sus errores estándar.

```
. logistic pa_alta c.edad i.rural
```

```
Logistic regression
```

```
Number of obs = 10.351
```

```
LR chi2(2) = 1464,14
```

```
Prob > chi2 = 0,0000
```

```
Pseudo R2 = 0,1038
```

```
Log likelihood = -6318,6948
```

pa_alta	Odds ratio	Std. err.	z	P> z	[95% conf. interval]	
edad	1,048494	,0013846	35,86	0,000	1,045784	1,051212
rural						
Zona Rural	,9426129	,0418033	-1,33	0,183	,8641398	1,028212
_cons	,0744018	,0052447	-36,86	0,000	,0648009	,0854253

Note: _cons estimates baseline odds.

También queremos incluir los índices AIC y BIC.

```
. estat ic
Akaike's information criterion and Bayesian information criterion
```

Model	N	ll(null)	ll(model)	df	AIC	BIC
.	10.351	-7050,765	-6318,695	3	12643,39	12665,12

Note: BIC uses N = number of observations. See [R] BIC note.

```
. return list
```

```
matrices:
```

```
      r(S) : 1 x 6
```

```
. mat list r(S)
```

```
r(S) [1,6]
```

```
      N      ll0      ll      df      AIC      BIC
.    10351 -7050.7655 -6318.6948      3    12643.39    12665.124
```

```
. display r(S)[1,"AIC"]
```

```
12643,39
```

Los comandos de Stata graban resultados en **r()** (como vimos para **correlate** y **estatic**) y/o en **e()** (comandos de estimación). La lista y la descripción de estos resultados están en la ayuda del comando correspondiente. (“Stored results”). Luego de correr un comando, los vemos con

```
. return list  
. return list
```



```
. logistic pa_alta c.edad i.rural
(salida omitida)

. ereturn list
scalars:
           e(rank) = 3
           e(N) = 10351
(...)
           e(l1) = -6318,6948199982
(...)
macros:
           e(cmdline) : "logistic pa_alta c.edad i.rural"
           e(cmd) : "logistic"
(...)
matrices:
           e(b) : 1 x 4
           e(V) : 4 x 4
           e(Cns) : 1 x 5
(...)
functions:
           e(sample)
```

Por defecto, el prefijo **collect:** recolecta todos los resultados en **e()** o **r()**. La excepción son las matrices **e(b)** y **e(V)** (aunque las podemos recolectar si queremos). Los resultados de la tabla de estimación se recolectan bajo los nombres **_r_b**, **_r_se**, **_r_z**, **_r_p**, etc. (escribir **. help collect get**)

Creamos la colección **Mismodelos** y allí ponemos nuestros resultados de regresión y le asignamos la tag **modelo[(1)]**. Aquí estamos creando las dos cosas, una dimensión (**modelo**) y un nivel (**1**). (aquí estamos recolectando sólo coeficientes y errores estándar)

```
. collect clear
. collect create Mismodelos
(current collection is Mismodelos)
. collect _r_b _r_se, tag(modelo[(1)]) : logistic pa_alta c.edad i.rural

Logistic regression                                     Number of obs = 10.351
                                                         LR chi2(2)      = 1464,14
                                                         Prob > chi2     = 0,0000
                                                         Pseudo R2      = 0,1038

Log likelihood = -6318,6948
```

pa_alta	Odds ratio	Std. err.	z	P> z	[95% conf. interval]	
edad	1,048494	,0013846	35,86	0,000	1,045784	1,051212
rural						
Zona Rural	,9426129	,0418033	-1,33	0,183	,8641398	1,028212
_cons	,0744018	,0052447	-36,86	0,000	,0648009	,0854253

Note: _cons estimates baseline odds

Agregamos los resultados de **estat ic** a la colección, también asignándole la dimensión **modelo** y el nivel **(1)**.

```
. collect AIC=r(S)[1,"AIC"] BIC=r(S)[1,"BIC"], tag(modelo[[1]]) : estat ic
Akaike's information criterion and Bayesian information criterion
```

Model	N	ll(null)	ll(model)	df	AIC	BIC
.	10.351	-7050,765	-6318,695	3	12643,39	12665,12

Note: BIC uses N = number of observations. See [R] BIC note.

En total, recolectamos resultados de dos modelos

```
. collect clear
. collect create Mismodelos

. collect _r_b _r_se, tag(modelo[(1)]): logistic pa_alta c.edad i.rural
. collect AIC=r(S)[1,5] BIC=r(S)[1,6], tag(modelo[(1)]): estat ic

. collect _r_b _r_se, tag(modelo[(2)]): logistic pa_alta c.edad##i.rural
. collect AIC=r(S)[1,5] BIC=r(S)[1,6], tag(modelo[(2)]): estat ic
```

A continuación especificamos la disposición de nuestra tabla: ponemos variables independientes (**colnames**) y estadísticos (**result**) en las filas y la dimensión **modelo** en las columnas.

```
. collect layout (colname#result) (modelo)
```

```
. collect layout (colname#result) (modelo)
```

```
Collection: Mismodelos
```

```
Rows: colname#result
```

```
Columns: modelo
```

```
Table 1: 18 x 2
```

	(1)	(2)
Edad		
Coefficient	1,048494	1,049813
Std. error	,0013846	,0017259
Zona Urbana		
Coefficient	1	1
Std. error	0	0
Zona Rural		
Coefficient	,9426129	1,129659
Std. error	,0418033	,1650808
Zona Urbana # Edad		
Coefficient		1
Std. error		0
Zona Rural # Edad		
Coefficient		,9964202
Std. error		,0027514
Intercept		
Coefficient	,0744018	,069929
Std. error	,0052447	,0059844

Omitimos los niveles de base en la tabla.

```
. collect style showbase off
. collect preview
```

	(1)	(2)
Edad		
Coefficient	1,048494	1,049813
Std. error	,0013846	,0017259
Zona Rural		
Coefficient	,9426129	1,129659
Std. error	,0418033	,1650808
Zona Rural # Edad		
Coefficient		,9964202
Std. error		,0027514
Intercept		
Coefficient	,0744018	,069929
Std. error	,0052447	,0059844

Especificamos el formato numérico de los resultados. (Hemos las opciones **nformat()** y **sformat()** directamente con el comando **table**; cuando constrimos la tabla con **collect**, esas son opciones para **collect style cell**.)

```
. collect style cell, nformat(%5,2f)
. collect style cell result[_r_se], sformat("(%)s")
. collect preview
```

	(1)	(2)
Edad		
Coefficient	1,05	1,05
Std. error	(0,00)	(0,00)
Zona Rural		
Coefficient	0,94	1,13
Std. error	(0,04)	(0,17)
Zona Rural # Edad		
Coefficient		1,00
Std. error		(0,00)
Intercept		
Coefficient	0,07	0,07
Std. error	(0,01)	(0,01)

Removemos el borde a la derecha de las etiquetas de fila.

```
. collect style cell border_block, border(right, pattern(nil))
. collect preview
```

	(1)	(2)
Edad		
Coefficient	1,05	1,05
Std. error	(0,00)	(0,00)
Zona Rural		
Coefficient	0,94	1,13
Std. error	(0,04)	(0,17)
Zona Rural # Edad		
Coefficient		1,00
Std. error		(0,00)
Intercept		
Coefficient	0,07	0,07
Std. error	(0,01)	(0,01)

Centramos los encabezados de columna (**column-header**) y los resultados (**item**)

```
. collect style cell cell_type[item column-header], halign(center)
. collect preview
```

	(1)	(2)
Edad		
Coefficient	1,05	1,05
Std. error	(0,00)	(0,00)
Zona Rural		
Coefficient	0,94	1,13
Std. error	(0,04)	(0,17)
Zona Rural # Edad		
Coefficient		1,00
Std. error		(0,00)
Intercept		
Coefficient	0,07	0,07
Std. error	(0,01)	(0,01)

Ocultamos las etiquetas para los estadísticos.

```
. collect style header result, level(hide)
. collect preview
```

	(1)	(2)
Edad	1,05 (0,00)	1,05 (0,00)
Zona Rural	0,94 (0,04)	1,13 (0,17)
Zona Rural # Edad		1,00 (0,00)
Intercept	0,07 (0,01)	0,07 (0,01)

Agregamos un espacio entre columnas.

```
. collect style column, extraspace(1)
. collect preview
```

	(1)	(2)
Edad	1,05 (0,00)	1,05 (0,00)
Zona Rural	0,94 (0,04)	1,13 (0,17)
Zona Rural # Edad		1,00 (0,00)
Intercept	0,07 (0,01)	0,07 (0,01)

Agregamos un espacio entre filas y usamos una **x** para indicar interacciones.

```
. collect style row stack, spacer delimiter(" x ")
. collect preview
```

	(1)	(2)
Edad	1,05 (0,00)	1,05 (0,00)
Zona Rural	0,94 (0,04)	1,13 (0,17)
Zona Rural x Edad		1,00 (0,00)
Intercept	0,07 (0,01)	0,07 (0,01)

Agregamos los AIC y BIC al final de cada columna.

```
. collect layout (colname#result result[AIC BIC]) (modelo)
```

```
Collection: Mismodelos
```

```
Rows: colname#result result[AIC BIC]
```

```
Columns: modelo
```

```
Table 1: 13 x 2
```

	(1)	(2)
Edad	1,05 (0,00)	1,05 (0,00)
Zona Rural	0,94 (0,04)	1,13 (0,17)
Zona Rural x Edad		1,00 (0,00)
Intercept	0,07 (0,01)	0,07 (0,01)
	12643,39	12643,71
	12665,12	12672,69

Agregamos etiquetas para AIC y BIC.

```
. collect style header result[AIC BIC], level(label)
. collect preview
```

	(1)	(2)
Edad	1,05 (0,00)	1,05 (0,00)
Zona Rural	0,94 (0,04)	1,13 (0,17)
Zona Rural x Edad		1,00 (0,00)
Intercept	0,07 (0,01)	0,07 (0,01)
AIC	12643,39	12643,71
BIC	12665,12	12672,69

Especificamos otro formato para AIC y BIC.

```
. collect style cell result[AIC BIC], nformat(%8,0f)
. collect preview
```

	(1)	(2)
Edad	1,05 (0,00)	1,05 (0,00)
Zona Rural	0,94 (0,04)	1,13 (0,17)
Zona Rural x Edad		1,00 (0,00)
Intercept	0,07 (0,01)	0,07 (0,01)
AIC	12643	12644
BIC	12665	12673

Grabar los estilos y etiquetas (Les debía el Paso 6)

Después de personalizar una tabla, podemos grabar los estilos y etiquetas para usarlos luego en otras tablas.

```
. collect style save style_reg1, replace  
(style from Mismodelos saved to file style_reg1.stjson)  
. collect label save label_reg1, replace  
(labels from Mismodelos saved to file label_reg1.stjson)
```

Ahora recolectamos resultados de modelos logísticos aplicados a otro conjunto de datos. Especificamos las filas y columnas, y aplicamos los estilos que hemos grabado.

```
. collect clear
. use lbw_sp, clear

. collect _r_b _r_se, tag(modelo[(1))]: logistic bajo_peso c.edad i.fuma
. collect AIC=r(S)[1,5] BIC=r(S)[1,6], tag(modelo[(1))]: estat ic

. collect _r_b _r_se, tag(modelo[(2))]: logistic bajo_peso c.edad##i.fuma
. collect AIC=r(S)[1,5] BIC=r(S)[1,6], tag(modelo[(2))]: estat ic
. collect layout (colname#result result[AIC BIC]) (modelo)

. collect style use style_reg1, override
. collect label use label_reg1
```

```
. collect preview
```

	(1)	(2)
Edad	0,95 (0,03)	0,92 (0,04)
Fuma	2,00 (0,64)	0,38 (0,58)
Fuma x Edad		1,08 (0,07)
Intercept	1,06 (0,80)	2,24 (2,29)
AIC	233	234
BIC	243	247

- El comando **table** nos permite crear fácilmente nuestras tablas con estadísticos de resumen, y resultados de otros comandos de Stata.
- El conjunto de comandos **collect** también nos permite construir tablas simples o complejas, personalizarlas, y exportarlas a varios formatos para incorporarlas directamente en nuestros reportes.
- Podemos guardar estilos y etiquetas en disco, para luego usarlos en nuevas tablas en el futuro.

Recursos adicionales:

- Por preguntas adicionales, pueden escribirnos a `tech-support@stata.com`
- Documentación
 - **table**
<https://www.stata.com/manuals/rtableintro.pdf>
 - **collect**
<https://www.stata.com/manuals/tables.pdf>
- Videos de Stata (sobre éste y otros temas)
<https://www.stata.com/links/video-tutorials/>
- Otros recursos oficiales de Stata:
<https://www.stata.com/support/>
- Foro de Stata (moderado por usuarios)
<https://www.statalist.org/>