Tiempo Económico
Núm. 31, vol. X
Tercer cuatrimestre de 2015

Microdatos del Censo de Población y Vivienda 2010 con Stata

Miguel Heras Villanueva*
Juan Francisco Islas**

(Recibido: 15 - mayo - 2015 – Aceptado: 21 - septiembre - 2015)

Resumen

Hoy en día el manejo de bancos de datos es una exigencia de los estudiantes y egresados de la carrera de economía. Atendiendo a dicha necesidad, se presenta una serie de herramientas de enseñanza y aprendizaje que incluye notas con definiciones teóricas, ejemplos y sintaxis replicable y modificable por el usuario a partir de los microdatos del Censo de Población y Vivienda 2010 del INEGI.

La pretensión es fortalecer las habilidades en el análisis de datos bajo un esquema de crucigrama temático, así como el diseño, estimación e inferencia sobre modelos de aplicación en las ciencias sociales, aprovechando las facilidades que brinda Stata en el manejo estadístico de los datos y la integración de cajas de herramientas con algoritmos de uso relevante en la práctica.

Abstract

Modern world demands Economics students and graduated a good knowledge of database handle. That is why we show a series of teaching and learning tools which includes theoretical definitions, examples as well as modifiable and replicable syntax for users through Stata. On one hand, the goal is to strengthen the abilities of data analysis by a scheme of thematic crossword, the design, estimation, inference and models applied to social sciences. On the other hand, to provide the user with some tools to make data statistical analysis and the elaboration of tool boxes with algorithms of relevant practical usage. Microdata are taken by the 2010 National Census of Population and Housingof INEGI.

* Profesor de la Universidad Autónoma Metropolitana-Xochimilco, <mikyheras@yahoo.com.mx>
** Consultor-investigador de la ONU FAO-MX, <juanfrancisco.islas@fao.org>
Los autores agradecen las observaciones y sugerencias realizadas por dos dictaminadores anónimos.

Palabras clave: microdatos, bases de datos, Stata, estadística descriptiva
Clasificación JEL: C10, C80, C87

Introducción

El análisis económico conlleva una mirada a datos de todo tipo: población, tipos de población, edad de los habitantes, nivel educativo, número de hogares, ingreso de los hogares, entre muchos otros. De esta manera, dicho estudio rebasa en muchos casos consideraciones meramente económicas para abarcar temas propios de la sociología, psicología, ciencias políticas, urbanismo y demás. Por ello, la manipulación de información estadística es hoy en día una herramienta indispensable del científico social.

Stata[1] es un paquete estadístico diseñado precisamente para cubrir las necesidades del manejo de grandes bancos de datos. Permite trabajarlos de una forma versátil, de tal manera que el usuario puede seguir rutinas ya establecidas previamente por otros investigadores o bien, puede llegar a elaborar otras propias. Sin embargo, es preciso como tarea previa al análisis conocer los fundamentos de Stata, para que a partir de ello, el trabajo estadístico y matemático-estadístico sea lo más ameno y sencillo posible. Precisamente ese es el objetivo del presente documento, afianzar la práctica básica del Paquete por medio de los microdatos del Censo de Población y Vivienda 2010, levantado por el INEGI.[2]

Estructurar una base de datos

Para estructurar una base de datos recurriremos a los microdatos del Censo de Población y Vivienda 2010 del INEGI, ingresamos al siguiente vínculo que forma parte del sitio del instituto:

<http://www.inegi.org.mx/est/contenidos/proyectos/accesomicrodatos/cpv2010/default.aspx>

Nos dirigimos a la parte inferior de dicho sitio para dar clic al vínculo *Tabulados básicos-cuestionario básico.*[3] Posteriormente elegimos el tema de *Población* y el formato asociado correspondiente al ícono de Excel, además de dar clic en *Continuar*.

1 Stata® es una marca registrada de StataCorp LP.

2 Se anticipa que el avance de este documento pretende servir de soporte didáctico para el trabajo con la información de la Encuesta Intercensal 2015, levantada por el organismo mencionado y cuya difusión de microdatos se espera para el año 2016. Véasehttp://www.inegi.org.mx/est/contenidos/proyectos/encuestas/hogares/especiales/ei2015/presentacion.aspx

3 En letras cursivas se identifican vínculos, temas, tabulados y archivos para la conformación de las bases de datos.

Descargamos el tabulado titulado *Población total, edad mediana, relación hombres-mujeres e índice de envejecimiento por entidad federativa según sexo*, que es facilitado por el INEGI en formato Excel bajo el nombre *01_04B_ESTATAL.XLS*, el cual lo guardamos en nuestra computadora para después importarlo desde Stata.[4]

Comandos para importar o exportar una base de datos

Stata permite importar y exportar bases de datos en diferentes formatos. Para realizar tales procedimientos es posible trabajar desde la barra de menús seleccionando *File>Import/Export*.[5] Con el fin de ejemplificar dicha tarea seleccionemos *File>Import>Excel spreadsheet(*.xls;*.xlsx)*. En seguida aparecerá una ventana a partir de la cual debemos rastrear la ubicación del archivo que deseamos importar, el nombre de la hoja en el archivo, así como el rango de las celdas donde se encuentran los datos. Líneas arriba se mencionó el nombre del archivo con el cual el Instituto proporciona los datos de interés, y dentro de éste se localiza la hoja *01.04B ESTATAL FILTROS.* Debido a que dicha hoja contiene filtros, hacemos explícito que el rango de los datos a importar se localiza a partir de la celda *A11*. Posteriormente, el comando aparecerá en la ventana de resultados.


```
. importexcelusing "C:\Documents\01_04B_ESTATAL.xlsx",sheet("01.04B
ESTATAL FILTROS") cellrange(A11) clear
```

Comandos *drop*, *keep*, *rename* y *edit*

Con el fin de eliminar variables se utiliza el comando drop. En el caso de que a partir de una lista de variables solamente se desee mantener algunas de ellas, el comando keep entra en acción. La sintaxis para el uso de los comandos mencionados es la siguiente:

```
. drop lista de variables
. keep lista de variables
```

Para eliminar o mantener solamente ciertas observaciones se hace uso de la condición if seguida de las instrucciones que den cabida a dicha condición. Por otro lado, si lo que deseamos es que el paquete elimine o mantenga ciertos datos en un rango determinado, utilizamos la preposición in seguida del rango en cuestión.

[4] Los ejercicios considerados en el documento pueden ser replicados a partir de la versión 12 del Paquete Stata.

[5] Con fuente *Courier New* tamaños 8 y 12 se plasmarán tanto los resultados que arroja el Paquete en la ventana de resultados del mismo, así como los comandos explicados y nombres de variables. Los comandos y nombres de variables dentro del texto estarán además subrayados para una mejor identificación de ellos por parte del lector, con excepción de la variable _merge y las expresiones _n y _N, pues contienen guiones bajos.

```
. drop if
. drop in rango
. keep if
. keep in rango
```

A fin de renombrar una variable utilice el siguiente comando:

```
. renamevariable_anteriorvariable_nueva
```

Los comandos anteriores serán útiles para comenzar a modificar nuestra base de datos. Primeramente, eliminemos los renglones 33, 34 y 35, renombremos las variables respetando los nombres originales del Instituto y observemos algunos tabulados.


```
. drop in 33/35
(3 observations deleted)
. rename A entidad
. rename B pob
. rename C pob_masc
. rename D pob_fem
. rename E edadmed
. rename F edadmed_masc
. rename G edadmed_fem
. rename H rel_hom_muj
. rename I indenvej
. rename J indenvej_masc
. rename K indenvej_fem
```

Con el uso del comando `edit` es posible visualizar los datos a partir del editor de los mismos. En primera instancia se notará que la columna de la variable entidad es muy ancha, lo cual impide visualizar los datos de las restantes variables. Para disminuir el ancho de dicha columna, debemos asignar un formato a la misma de tipo `%25s`. El signo de porcentaje es utilizado por el paquete para asignar el formato de la presentación de los datos, `25` se refiere a los dígitos o caracteres que deben caber en la columna, mientras que la letra `s` delimita a la variable como tipo `string`, es decir, que posee una llave la cual impide manipularla.

```
. format %25s entidad
. edit
```

Comando `table`

El comando `table` nos permite elaborar tablas de datos a partir de atributos, los cuales permiten al usuario presentarlas en función de una mejor comprensión de las mismas, de visualizarlas más fácilmente o de cierto interés para el usuario. La tabla que se presenta a continuación se elabora a partir de la variable "entidad", conformada por una matriz cuyos

encabezados son el total de la población y por género, con un formato fijo de 9 dígitos de amplitud. La última palabra se refiere a la suma total por columnas, la cual se observa en la parte inferior.[6]

```
. table entidad, c(sum pob sum pob_masc sum pob_fem) f(%9.0f) row
```

entidad	sum(pob)	sum(pob_masc)	sum(pob_fem)
01 Aguascalientes	1184996	576638	608358
02 Baja California	3155070	1591610	1563460
03 Baja California Sur	637026	325433	311593
04 Campeche	822441	407721	414720
...	...	...	...
31 Yucatán	1955577	963333	992244
32 Zacatecas	1490668	726897	763771
Total	112336538	54855231	57481307

El comando table también permite elaborar tablas de estadísticos descriptivos seleccionados, como la media.

```
. table entidad, c(mean edadmed mean edadmed_masc mean edadmed_fem)
f(%9.0f)row
```

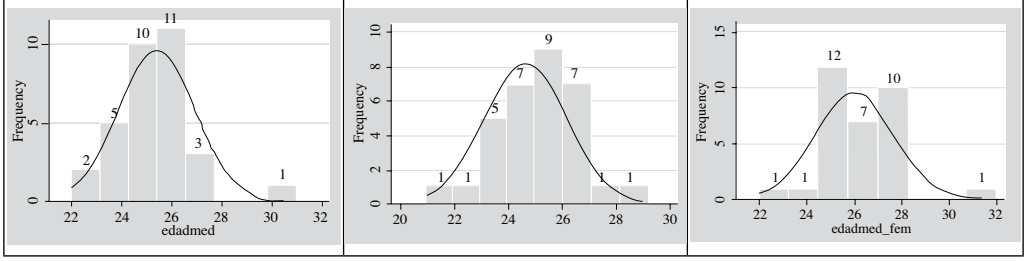
entidad	mean(edadmed)	mean(edadme~c)	mean(edadme~m)
01 Aguascalientes	24	23	25
02 Baja California	26	26	26
03 Baja California Sur	26	26	26
04 Campeche	25	25	26
...	...	...	...
31 Yucatán	26	26	27
32 Zacatecas	25	24	25
Total	25	25	26

Con los comandos histogram y summarize es posible observar que, para la distribución general y para la población masculina, el promedio de la edad mediana es de 25 años y de 26 para la población femenina. Es importante recalcar que el segundo comando va procedido de una coma para incorporar el atributo detail, el cual es utilizado para obtener la estadista descriptiva a detalle de las variables en cuestión.

[6] Por razones de espacio las tablas se recortan.

```
summarize edadmed, detail
histogram edadmed, bin(8) freqaddlabels normal
summarize edadmed_masc, detail
histogram edadmed_masc, bin(8) freqaddlabels normal
summarize edadmed_fem
histogram edadmed_fem, bin(8) freqaddlabels normal
```

edadmed	**edadmed_masc**	**edadmed_fem**
Mínimo 22	Mínimo 21	Mínimo 22
Máximo 31	Máximo 29	Máximo 32
Media 25.4	Media 24.6	Media 26.03
Mediana25	Mediana 25	Mediana 26
Desv. est. 1.52	Desv. est. 1.56	Desv. est. 1.67
Sesgo 1.1	Sesgo 0.17	Sesgo 1.1
Curtosis7.4	Curtosis 3.8	Curtosis 6.8



Se desprenden dos conclusiones sobre la tendencia central de la distribución de la edad mediana en las entidades:

i) La población femenina es al menos un año mayor que la población masculina, tanto en promedio como en mediana.
ii) Estadísticamente, no hay diferencia entre la media y la mediana de estas tres distribuciones, lo que significa que se trata de distribuciones simétricas.

La simetría de las distribuciones se puede validar mediante una prueba de normalidad, como la de Shapiro-Wilk a partir del comando `swilkedadmed*`. El asterisco al final de edadmed obliga al paquete a elaborar la prueba sobre todas las variable que comiencen nombradas de esa forma.

```
. swilkedadmed*
                    Shapiro-Wilk W test for normal data

    Variable          Obs        W          V          z        Prob>z
-------------+-------------------------------------------------------
     edadmed |     32       0.91383     2.874       2.192     0.01419

edadmed_masc |     32       0.98986     0.338      -2.250     0.98777
edadmed_fem |     32       0.93220     2.262       1.694     0.04512
```

Al 99 por ciento de confianza, no se rechaza la hipótesis de normalidad de las tres distribuciones de la edad mediana. De esta forma, se justifica estadísticamente por qué coinciden la media aritmética y la mediana de la edad mediana. Asimismo, se confirma lo establecido por el teorema del límite central: el promedio o valor esperado de cualquier parámetro poblacional coincide con la media aritmética del estadístico correspondiente, bajo normalidad. En este caso la mediana poblacional coincide con el promedio de las medianas muestrales, dado que estas últimas pertenecen a una distribución normal. Ahora bien, debido a que solamente nos interesa contar con las variables `pob` y `edadmed`, utilizamos el comando `keep` para mantenerlas y eliminamos las demás.

```
. keep entidad pob edadmed
```


Comando `save`

Para guardar una base de datos es posible hacer uso de la barra de menús seleccionando *`File>Save`* o bien, *`File>Save As`*. La primera opción guarda los cambios realizados a una base de datos previamente abierta, por lo que no será posible restaurar los datos originales. En caso de no querer hacer cambios a la base original y guardar aquellos hechos sobre dicha base, se debe hacer uso de la segunda opción, registrando una segunda base con un nombre diferente en la ubicación de la computadora elegida por el usuario. Si el usuario hace uso de la barra de herramientas, debe asegurarse de dar clic en el ícono `Save`, el cual grabará los cambios realizados a la base de datos original.

```
. save "C:\Documents\censo_a.dta"
```

Estructurar otra base de datos

Ahora descargamos el tabulado titulado*Localidades y su población por entidad federativa según tamaño de localidad*, también facilitado por el INEGI en formato Excel pero ahora bajo el nombre *01_01B_ESTATAL.XLS*, el cual lo guardamos en nuestra computadora para importarlo posteriormente desde Stata.

```
. import excel using "C:\Documents\01_01B_ESTATAL.
xlsx",sheet("01_01B_ESTATAL") cellrange(A12) clear
```

Como se puede visualizar, estamos usando la hoja con nombre *01_01B_ESTATAL*, y el rango a partir de la celda `A12`. A continuación eliminamos las últimas observaciones y renombramos a las variables con excepción de aquellas que poseen valores para los tamaños de

localidad. Sin embargo, para recordar qué valores tienen, podemos etiquetar a tales variables haciendo uso del comando label que se verá con mayor detalle más adelante.

```
. drop in 65/69
. rename A entidad
. rename B desglose
. rename C total
. label variable D "1-249"
. label variable E "250-499"
. label variable F "500-999"
. label variable G "1000-2499"
. label variable H "2500-4999"
. label variable I "5000-9999"
. label variable J "10000-14999"
. label variable K "15000-29999"
. label variable L "30000-49999"
. label variable M "50000-99999"
. label variable N "100000-249999"
. label variable O "250000-499999"
. label variable P "500000-999999"
. label variable Q ">1000000"
```


Se aprecia en que forma quedan etiquetadas las variables que hacen referencia a los tamaños de población por habitantes en la ventana de resultados. Con la finalidad de eliminar todos los renglones que poseen datos de Localidades, retomamos el comando drop ampliando la sintaxis del mismo por medio de la imposición de la condición if. Así mismo, generamos una nueva variablepor medio del comando gen (el cual también será visto con mayor detalle más adelante). De acuerdo con el INEGI, una población urbana es aquella donde viven más de 2,500 personas, es decir, debemos agregar las cifras de las columnas H a la Q en concordancia con las etiquetas que asignamos a las variables importadas.

```
. drop if total<30000
. gen poburb=H+I+J+K+L+M+N+O+P+Q
```

Con la finalidad de visualizar los valores numéricos de la nueva variable poburb sin el formato científico (ampliando el ancho de su columna a 10 caracteres), así como para observar los nombres de las entidades con un ancho de 25 caracteres, se siguen los siguientes pasos.

```
. format %25s entidad
. format %10.0g poburb
. edit entidad poburb
```

En este punto es importante mencionar que para distinguir y utilizar los distintos formatos que el paquete posee, usamos el comando help seguido del comando del cual requerimos ayuda, en este caso format.

```
. helpformat
```

A partir del trabajo realizado con esta segunda base, solamente requerimos los datos de las variables entidad y poburb. Para eliminar las variables restantes ahora tomamos ventaja del comandokeep y guardamos la base resultante.

```
. keep entidad poburb
. save "C:\Users\Samuel Caso\Documents\KINGSTON\trabajos_constan-
cias\censo_b.dta"
```

El comando merge

El comando merge une observaciones de una base de datos en memoria (denominada base master) con aquellas de un archivo .dta (base en uso), lo cual permite vincular una o más variables. La unión de bases puede ser del tipo one-to-one, one-to-many, many-to-one y many-to-many. A partir del comando se crea una nueva variable llamada _merge, la cual contiene códigos numéricos del origen y características de las observaciones de la base resultante. Para llevar a cabo la unión de dos bases de datos a partir de la barra de menús seleccionamos Data>Combine datasets>Mergetwodatasets.

Hasta este momento hemos elaborado dos bases de datos denominadas *censo_a.dta* y *censo_b.dta*, las cuales poseen una variable en común: entidad. Para unir dichas bases de datos debemos tener en uso la primera base y llamar a la segunda, aunque previamente hacemos uso del comando clear con el fin de eliminar cualquier información en uso por parte del paquete. El tipo que utilizaremos será one-to-one, ya que la pretensión es unir cada uno de los datos de las bases por nombre de entidad federativa para con ello, obtener dos bases en una.

```
. clear
. use "C:\Documents\censo_a.dta", clear
. merge 1:1 _n using "C:\Documents\censo_b.dta"
entidad was str144 now str192

Result                      # of obs.
-----------------------------------------
not matched                 0
matched                     32  (_merge==3)
    -----------------------------------------
. format %25s entidad
. edit
```

Por la tabla desplegada en la ventana de resultados es posible visualizar que 32 observaciones se vincularon con la base en uso, es decir, aquéllas de la segunda base a partir de la variable poburb. Obsérvese cómo nuevamente se dio formato a la columna de la variable entidad para que todas las demás puedan verse en el editor de datos. La nueva base contiene las variables entidad, pob, edadmed, poburb y _merge.

Tipos de variables

Stata permite el trabajo con diversos tipos de variables y definirlas como numéricas enteras (`byte`, `integer`, `long`) y reales(`real` y `float`),además de tipo cadena (`string`). A continuación se lleva a cabo una clasificación de las mismas:

- Integers:
  - Byte: comprende datos de una variable de uno a dos dígitos.
  - Int: datos con valor entero de hasta ±32,740.
  - Long: datos de una variable con valor enterohasta ±2.14 miles de millones.
- Real:
  - Float: Los datos de estas variables poseen precisión sencilla de siete dígitos. Por omisión, las variables numéricas reales se registran de este modo, a menos que se especifique de otra forma.
  - Double: Tales datos, denominados de doble precisión, cuentan con un mínimo de 15 dígitos de precisión.
- String: Las variables de este tipo conllevan una longitud específica que va desde `str1` (un caracter) hasta `str244` (244 caracteres).A partir de la versión 13 de Stata, se admite una longitud de hasta 1,204 caracteres.

El comando `describe` y etiquetas de valor

Por medio del comando `describe` Stata muestra los contenidos de la base de datos, incluyendo los tipos de datos de cada variable.

```
- describe
Contains data from C:\Documents\censo_a.dta
obs:                        32
vars:                        5                          14 Aug 2014 19:24
size:                    6,464
-------------------------------------------------------------
                       storage   display    value
variable     name      type      format     label    variable label
-------------------------------------------------------------
entidad                str192     %25s
pob                     long    %10.0gc
edadmed                 byte     %10.0g
poburb                  float    %10.0g              "poburb"
_merge                  byte     %23.0g     _merge
-------------------------------------------------------------
```

La tabla indica que la base de datos contiene 5 variables y 32 observaciones. La variable `entidad` es tipo `string`, lo cual quiere decir que ningún nombre de las entidades excede los 192 caracteres. Cabe mencionar que el valor de la cadena más larga no necesariamente ocupará los 192 caracteres, por lo que se tiene la posibilidad de optimizar espacio en memoria que ocupa este tipo de variables usando el comando `compress` o bien, redefiniendo el tamaño de la variable específica, en este caso mediante `format%25sentidad`.

Por lo que se refiere a las variables numéricas, `pob`, `edadmed` y `poburb` están registradas como variables tipo `long`, `byte` y `float` respectivamente. Finalmente, la variable `_merge` es tipo `byte` debido al número de caracteres que contiene, sólo un dígito.

Etiquetas de valor

Antes de registrar nuestra última base de datos a partir de la unión de las que elaboramos con anterioridad, agreguemos una nueva variable denominada `region`. La regionalización que tomaremos para el país será aquella elaborada por el Instituto Nacional para el Federalismo y el Desarrollo Municipal (INAFED) según consta en el sitio:

<http://www.inafed.gob.mx/work/enciclopedia/index.html>

El primer paso consiste en generar tal variable por medio del comando `generate`, y después asignar valores numéricos a cada una de las entidades del país de la siguiente manera: 1 para las entidades del centro (`Centro`), 2 para los de la costa del Pacífico (`C Pacif`), 3 para el Distrito Federal (`DF`), 4 para las del Golfo y sur (`Golfo y Sur`) y 5 para aquéllas del norte (`Norte`).

```
. generate region = .
. replace region = 1 in 1
. replace region = 1 in 29
. replace region = 1 in 13
. replace region = 1 in 22
. replace region = 1 in 24
. replace region = 1 in 17
. replace region = 1 in 21
. replace region = 1 in 15
. replace region = 1 in 11
. replace region = 2 in 12
. replace region = 2 in 18
. replace region = 2 in 20
. replace region = 2 in 6
. replace region = 2 in 25
. replace region = 2 in 16
. replace region = 2 in 14
. replace region = 3 in 9
. replace region = 4 in 27
```

```
. replace region = 4 in 31
. replace region = 4 in 4
. replace region = 4 in 30
. replace region = 4 in 23
. replace region = 4 in 7
. replace region = 5 in 2
. replace region = 5 in 3
. replace region = 5 in 5
. replace region = 5 in 8
. replace region = 5 in 10
. replace region = 5 in 19
. replace region = 5 in 26
. replace region = 5 in 28
. replace region = 5 in 32
. label variable region "INAFED"
. label define region 1 "Centro" 2 "C Pacif" 3 "DF" 4 "Golfo y Sur"
5 "Norte", replace
. labelvaluesregionregion
```


Los últimos tres comandos se refieren a etiquetar la variable `region` con el nombre de INAFED, asignar las etiquetas de las regiones a los valores numéricos y definir la etiqueta de valor. Los datos se visualizan en el editor con el comando `edit`.

Con la nueva descripción de la base de datos conformada hasta este momento, se percibe que la variable `region` es tipo `float`, pero claramente se nota que posee dos etiquetas, una que describe a la variable y otra a sus valores, mientras que la variable `_merge` posee una etiqueta de valor que delimita a los valores que toma.

```
. describe
Contains data from C:\Documents\censo_a.dta
obs:                    32
vars:                    6                          15 Aug 2014 12:11
size:                6,592
-------------------------------------------------------------------------------
                  storage  display     value
variable   name       type   format    label  variable label
-------------------------------------------------------------------------------
entidad             str192    %25s
pob                   long  %10.0gc
edadmed               byte   %10.0g
poburb                float  %10.0g
_merge                byte   %23.0g    _merge
region                float  %11.0g    region  INAFED
-------------------------------------------------------------------------------
```

Finalmente, registremos la tercera base en nuestra computadora con la cual trabajaremos posteriormente, no sin antes eliminar la variable `_merge` que no tendrá utilidad para los ejercicios posteriores.

```
. drop _merge
. save "C:\Documents\censo2010.dta"
```

El comando use

Para abrir un archivo de datos en formato Stata, es decir, con extensión de nombre de archivo *.dta, se puede recurrir a diversos procedimientos. Como primera instancia a través del comando use como se muestra a continuación:

```
. use "C:\Documents\censo2010.dta", clear
```

Cabe hacer notar que debe plasmarse la ruta completa de la ubicación del archivo, la cual debe ir delimitada por comillas, seguida de una coma y el comando clear, a través del cual el paquete borra cualquier información previa para que el mismo comience a trabajar con datos nuevos. También es posible abrir un archivo nuevo por medio de la barra de menús seleccionando File>Open, hasta encontrar el archivo que previamente se hubo guardado en la computadora. Un tercer procedimiento se lleva a cabo por medio de la barra de herramientas, para lo cual solamente es necesario dar clic en el ícono Open y buscar el archivo en cuestión.

A través del comando edit el investigador puede echar una mirada a los datos del archivo. Para listar los contenidos del archivo en la ventana de resultados, el requisito es utilizar el comando list.

```
. list

     +-------------------------------------------------------------------------+
  1. |                              entidad |      pob | edadmed |   poburb |
     |                    01 Aguascalientes |  1184996 |      24 |   957589 |
     |-------------------------------------------------------------------------|
     |                                 region                                  |
     |                                 Centro                                  |
     +-------------------------------------------------------------------------+

     +-------------------------------------------------------------------------+
  2. |                              entidad |      pob | edadmed |   poburb |
     |                   02 Baja California |  3155070 |      26 |  2911874 |
     |-------------------------------------------------------------------------|
     |                                 region                                  |
     |                                  Norte                                  |
     +-------------------------------------------------------------------------+

     +-------------------------------------------------------------------------+
  3. |                              entidad |      pob | edadmed |   poburb |
     |               03 Baja California Sur |  637,026 |      26 |   548718 |
     |-------------------------------------------------------------------------|
     |                                 region                                  |
     |                                  Norte                                  |
     +-------------------------------------------------------------------------+
. edit
```

El contenido del archivo es mostrado en forma de matriz o marco de datos (*data frame*). Las observaciones están ubicadas en las filas de la matriz, mientras que las variables en las columnas. La tabla contiene 32 filas que corresponden a las entidades del país y 5 columnas o variables: entidad, region, pob, poburb, edadmed. Los nombres de las variables no deben contener guiones (-) o caracteres no numéricos o no alfabéticos. Puesto que el paquete es sensitivo, ENTIDAD, Entidad y entidad fungen como tres variables. La recomendación es utilizar letras minúsculas para nombrar a las variables.

Puede caber la posibilidad de que la base de datos a utilizar se encuentre en un formato delimitado por comas. Ante dicha situación, el comando insheet debe ser usado para importar los datos correspondientes.

```
. insheetusing "ruta de la base de datos", clear
```


Comandos generate y replace

Los comandos básicos para generar y transformar datos en Stata se denominan generate y replace. El primer comando es útil para crear una nueva variable, mientras que el segundo para modificar una ya existente. A estas alturas vale la pena mencionar que la mayoría de los comandos en Stata pueden abreviarse, por ejemplo, el comando generate puede abreviarse como gen, pero el comando replace no es sujeto de abreviaciones.

A continuación vamos a crear una nueva variable intitulada porcurb, la cual reflejará la proporción de la población urbana en cada una de las entidades de la República Mexicana. Para lograrlo, solo basta ser cuidadosos en plasmar de manera adecuada la razón entre población urbana (popurb) y población total (pop), y después pedimos al paquete que despliegue la estadística descriptiva de la nueva variable para obtener la proporción a nivel nacional. Si queremos observar los datos por entidad, basta con aplicar el comando list o edit.

```
. genporcurb = poburb/pob
. summarizeporcurb

Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
porcurb |        32    .7545359    .1411445   .4732307   .9954032
```

Lo cual nos indica que en promedio, el 75% de la población del país está asentada en zonas urbanas, con un rango que va del 47% a prácticamente el 100%. En caso de que la variable porcurb ya hubiera existido, pero deseamos interpretarla como porcentaje en lugar de términos decimales, utilizamos el comandoreplace.

```
. replaceporcurb = porcurb * 100
(32 real changes made)
. sumporcurb
```

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
porcurb |        32    75.45359    14.11445   47.32307   99.54031
```

Nótese que el comando `replace` reporta el cambio realizado sobre las 32 observaciones, además de que el comando `summarize` se puede abreviar como `sum`.

Comandos `sort` y `gsort`

El comando `sort` ordena una variable tipo numérica o `string` en orden ascendente. De indicar una lista de dos o más variables, el comando ordena las observaciones en función de la primera variable, mientras que para aquéllas con valores iguales lo hace en función de la segunda variable. Después de aplicar el comando, la base de datos quedará ordenada a partir de las variables asignadas, por lo que debería salvarse en caso de querer guardar los cambios.

La ventaja del comando `gsort` sobre `sort` es que ordena datos de forma ascendente o descendente. Para ordenar en forma descendente se requiere utilizar el signo (-) antes de cierta variable y el signo (+) para llevar a cabo el procedimiento contrario. Para ejemplificar, ordenemos a las entidades por región y dentro de cada región, por tamaño de población en forma descendente. Antes de realizar el procedimiento citado, agreguemos un nuevo formato a las variables `pob` y `poburb`, de tal manera que sus cifras estén separadas por comas en cuanto a cientos se refiere.

```
. format %12.0gc pob
. format %10.0gc poburb
. gsort region -pob
. list region entidad pob, sepby(region)
    +-------------------------------------------------------------------+
|           region                                entidad         pob |
     |-------------------------------------------------------------------|
  1. |      Centro                                  15 México   15,175,862 |
  2. |      Centro                                  21 Puebla    5,779,829 |
  3. |      Centro                               11 Guanajuato    5,486,372 |
  4. |      Centro                                  13 Hidalgo    2,665,018 |
  5. |      Centro                        24 San Luis Potosí    2,585,518 |
  6. |      Centro                              22 Querétaro    1,827,937 |
  7. |      Centro                                17 Morelos    1,777,227 |
  8. |      Centro                         01 Aguascalientes    1,184,996 |
  9. |      Centro                               29 Tlaxcala    1,169,936 |
     |-------------------------------------------------------------------|
 10. |     C Pacif                                  14 Jalisco    7,350,682 |
 11. |     C Pacif                   16 Michoacán de Ocampo    4,351,037 |
```

Para listar a las cuatro entidades con el menor y mayor número de habitantes es necesario asignar un rango a través de la palabra `in`. En tal sentido, el rango 1/4 se referirá a las primeras cuatro observaciones en términos de número de habitantes y el rango -4/32 a las

últimas cuatro. Recordando que el comando `sort` ordena de forma ascendente, debemos sustituirlo por `gsort` para que realice el procedimiento de forma decreciente respecto a las entidades con la mayor población.

```
. sort pob
. list entidad region pob in 1/4
```

```
+-------------------------------------------------+
     |                  entidad        region       pob   |
     |-------------------------------------------------|
  1. |   03 Baja California Sur         Norte     637,026 |
  2. |                06 Colima       C Pacif     650,555 |
  3. |              04 Campeche   Golfo y Sur     822,441 |
4. |               18 Nayarit       C Pacif   1,084,979 |
     +-------------------------------------------------+
```



```
. listentidad region pob in -4/32
```

```
+-----------------------------------------------------------------+
     |                             entidad         region       pob    |
     |-----------------------------------------------------------------|
 29. |                          14 Jalisco       C Pacif    7,350,682 |
 30. |   30 Veracruz de Ignacio de la Llave   Golfo y Sur    7,643,194 |
31. |                  09 Distrito Federal            DF8,851,080 |
 32. |                          15 México        Centro15,175,862 |
     +-----------------------------------------------------------------+
```

```
. gsort -pob
. listentidad region pob in 1/4
```

```
+-----------------------------------------------------------------+
     |                             entidadregionpob |                 |
     |-----------------------------------------------------------------|
  1. |                          15 México        Centro   15,175,862 |
  2. |                  09 Distrito Federal            DF    8,851,080 |
  3. |   30 Veracruz de Ignacio de la Llave   Golfo y Sur    7,643,194 |
  4. |                          14 Jalisco       C Pacif    7,350,682 |
     +-----------------------------------------------------------------+
```

Vimos que para restringir el uso de comandos a ciertas condiciones se hace uso de la palabra `if`. Para ejemplificarlo, generemos una nueva variable a partir de aquella denominada `edadmed`, definida como `edadmed1` para las entidades con una población mayor a los 4 millones de habitantes. Cabe señalar que las observaciones que no cumplan con la condición serán reportadas como no existentes por medio de un punto (.).

```
. generate edadmed1 = edadmed if pob> 4000000
(23 missing values generated)
. sort entidad
. list entidad region pob edadmed1
```

```
     +-------------------------------------------------------------------------+
     |                             entidad       region       pob   edadmed1 |
     |-------------------------------------------------------------------------|
  1. |                   01 Aguascalientes        Centro   1,184,996       . |
  2. |                  02 Baja California        Norte    3,155,070       . |
  3. |              03 Baja California Sur        Norte      637,026       . |
  4. |                         04 Campeche   Golfo y Sur     822,441       . |
  5. |             05 Coahuila de Zaragoza        Norte    2,748,391       . |
     |-------------------------------------------------------------------------|
  6. |                           06 Colima       C Pacif      650,555       . |
  7. |                           07 Chiapas   Golfo y Sur    4,796,580      22 |
  8. |                        08 Chihuahua         Norte    3,406,465       . |
9. |                 09 Distrito Federal            DF    8,851,080       31 |
 10. |                          10 Durango         Norte    1,632,934       . |
     |-------------------------------------------------------------------------|
```


Variables categóricas (*dummy o booleano*)

Las variables categóricas, también conocidas como *dummy*o *booleanas*, son aquellas que toman dos valores para indicar que una condición particular se cumple. Generalmente se expresan como {0,1}, frente a lo cual la condición booleana establece que 1 es verdadero y 0 falso. Para ejemplificar lo anterior, generemos dos variables para las entidades con menor (pobpeq) y mayor (pobgde) número de habitantes, por lo cual es necesario el uso de los comandos generate y replace para definir los valores que cumplirán con las condiciones verdadera y falsa.

```
. generate pobpeq = 0
. replace pobpeq = 1 if pob<= 4000000
(23 real changes made)
. generate pobgde = 0
. replace pobgde = 1 if pob> 4000000
(9 real changes made)
. list entidad pob pobpeq pobgde
    +-------------------------------------------------------------------------+
|                             entidad           pob       pob peq  pob gde |
     |-------------------------------------------------------------------------|
  1. |                   01 Aguascalientes    1,184,996       1        0    |
  2. |                  02 Baja California    3,155,070       1        0    |
  3. |              03 Baja California Sur      637,026       1        0    |
  4. |                         04 Campeche      822,441       1        0    |
  5. |             05 Coahuila de Zaragoza    2,748,391       1        0    |
     |-------------------------------------------------------------------------|
  6. |                           06 Colima      650,555       1        0   |
7. |                           07 Chiapas      4,796,580       0        1   |
  8. |                        08 Chihuahua    3,406,465       1        0   |
  9. |                 09 Distrito Federal    8,851,080       0        1   |
 10. |                          10 Durango    1,632,934       1        0   |
|-------------------------------------------------------------------------|
 11. |                        11 Guanajuato    5,486,372        0        1 |
 12. |                          12 Guerrero    3,388,768        1        0 |
```

Estadística descriptiva con `if`, `by` variables(s) y `by()`

Para realizar análisis de estadística descriptiva acerca de un subconjunto de datos es requisito indispensable utilizar la palabra `if`. El siguiente ejemplo es a partir de las regiones uno (Centro) y dos (Costa del Pacífico) con el fin de visualizar lo anterior.

```
. summarize edadmed pob if region==1

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
    edadmed  |         9       24.88889    .781736          24         26
      pob    |         9        4183633    4463022       1169936   1.52e+07
```



```
. summarizeedadmedpob if region==2

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
    edadmed  |         7          25    1.154701         23         26
     pob     |         7     3342249     2234920     650555    7350682
```

Puesto que los datos del Censo son discretos, es posible hacer uso del prefijo `by` seguido de una o más variables. Sin embargo, cabe señalar que el prefijo solo admite la ejecución de un comando sobre un subconjunto de datos.

```
. by region, sort: summarize edadmedpob
-------------------------------------------------------------------------------
->region = Centro
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
    edadmed  |         9    24.88889     .781736         24         26
      pob    |         9     4183633     4463022    1169936   1.52e+07
-------------------------------------------------------------------------------
->region = C Pacif
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         7          25    1.154701         23         26
      pob    |         7     3342249     2234920     650555    7350682
-------------------------------------------------------------------------------
->region = DF
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         1          31           .         31         31
       pob   |         1     8851080           .    8851080    8851080
-------------------------------------------------------------------------------
->region = Golfo y Sur
Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
 edadmed |         6          25     1.67332         22         27
   pob   |         6     3130329     2603883     822441    7643194
```

```
-------------------------------------------------------------------------------
->region = Norte
Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
 edadmed |         9    25.88889    .9279607         24         27
   pob   |         9     2628338     1207267     637026    4653458
```

Es muy importante no confundir el prefijo by seguido de una o más variables, con la opción by() para algunos comandos de Stata. Con el fin de precisar la distinción, realicemos el análisis de estadística descriptiva para edadmed utilizando el comando tabstat, el cual genera la estadística para todas las regiones.

```
. tabstatedadmed, by(region) statistics(N mean sd min max)
Summary for variables: edadmed
by categories of: region (INAFED)
```



```
     region |         N      mean        sd       min       max
------------+--------------------------------------------------
     Centro |         9  24.88889   .781736        24        26
    C Pacif |         7        25  1.154701        23        26
         DF |         1        31         .        31        31
Golfo y Sur |         6        25   1.67332        22        27
      Norte |         9  25.88889  .9279607        24        27
------------+--------------------------------------------------
      Total |        32  25.40625  1.521022        22        31
-------------------------------------------------------------
```

Por medio de la opción by() se modifica el comando tabstat, a partir de lo cual Stata despliega las estadísticas descriptivas por región. Por su parte, el prefijo by repite el comando summarize para cada región. Cabe resaltar que el prefijo by puede incluir más de una variable, cada una de las cuales es evaluada o ejecutada según el comando utilizado. Para dar un ejemplo, combinemos pobpeq y pobgde en una variable categórica, tampob, con un valor de 1 para las entidades pequeñas y de 2 para las grandes en términos de número de habitantes, para después computar la estadística descriptiva para cada subconjunto de entidad en cada región.

```
. generatetampob = pobpeq + 2*pobgde
. by region tampob, sort: summarize edadmedpob
-------------------------------------------------------------------------------
->region = Centro, tampob = 1
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         6          25    .6324555         24         26
       pob   |         6     1868439    650184.8    1169936    2665018
-------------------------------------------------------------------------------
->region = Centro, tampob = 2
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         3    24.66667    1.154701         24         26
       pob   |         3     8814021     5511469    5486372   1.52e+07
```

```
-------------------------------------------------------------------------------
->region = C Pacif, tampob = 1
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         5          25    1.414214         23         26
      pob    |         5     2338805     1400850     650555    3801962
-------------------------------------------------------------------------------
->region = C Pacif, tampob = 2
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         2          25           0         25         25
      pob    |         2     5850860     2121069    4351037    7350682
-------------------------------------------------------------------------------
->region = DF, tampob = 2
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         1          31           .         31         31
      pob    |         1     8851080           .    8851080    8851080
-------------------------------------------------------------------------------
->region = Golfo y Sur, tampob = 1
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         4       25.25          .5         25         26
      pob    |         4     1585550    635955.7     822441    2238603
-------------------------------------------------------------------------------
->region = Golfo y Sur, tampob = 2
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         2        24.5    3.535534         22         27
      pob    |         2     6219887     2012860    4796580    7643194
-------------------------------------------------------------------------------
->region = Norte, tampob = 1
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         8       25.75    .8864053         24         27
      pob    |         8     2375199     1003294     637026    3406465
-------------------------------------------------------------------------------
->region = Norte, tampob = 2
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     edadmed |         1          27           .         27         27
      pob    |         1     4653458           .    4653458    4653458
-------------------------------------------------------------------------------
```


Obsérvese como la población más joven se localiza en la región Golfo y Sur, y aquella con mayor edad en la del Distrito Federal.

Convertir variables tipo `string` a numéricas y viceversa

Con anterioridad se mencionó que Stata identifica dos tipos de variables: numéricas y tipo `string`. Al momento de importar datos, el paquete puede malinterpretar una variable numérica como tipo `string`, por lo que es necesario realizar los ajustes necesarios para

trabajar adecuadamente con el primer tipo de variable. Como ejemplo vamos a generar una nueva variable denominada region2 con los datos de la variable region y convertirla en tipo string.

```
. generate region2=region
. tostring region2, replace
region2 was float now str1

. describe region2
storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
region2         str1   %9s

. summarize region2

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     region2 |         0
```


Como es posible observar, una variable tipo string impide al paquete realizar cálculos estadísticos sobre la misma, por lo que se torna necesario realizar los procedimientos necesarios para transformarla en numérica. El proceso consiste en utilizar el comando destring seguido del nombre de la variable, además de agregar el atributo replace después de una coma. Obsérvese como el proceso para transformar una variable en tipo string el comando requerido fue tostring, seguido igualmente del atributo replace procedido de una coma.

```
. destring region2, replace
region2 has all characters numeric; replaced as byte

. describe region2

storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
region2         byte   %10.0g

. summarize region2

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
     region2 |        32     2.96875    1.655575          1          5
```

Cálculos por grupos

Una de las virtudes del paquete Stata es que permite transformar variables o generar estadísticas por grupos de datos de variables. Para visualizar lo anterior cambiaremos datos

por grupos con los comandos generate, replace y egen. Para cierto grupo, _n y _N poseen diferentes usos y significados, ya que _n se refiere a la observación en cuestión, mientras que _N a la última del grupo. Para comenzar vamos a ordenar los datos de población por región en orden descendente con el comando gsort, para después generar la función sum() por región,

```
. gsort region -pob
. by region: generate totpob = sum(pob)

. format %12.0gc totpob
. list region entidadpobtotpob, sepby(region)

     +-------------------------------------------------------------------+
  1. |      region |                                 entidad |        pob |
     |      Centro |                               15 México | 15,175,862 |
     |-------------------------------------------------------------------|
     |                                totpob                             |
     |                            15,175,862                             |
     +-------------------------------------------------------------------+

     +-------------------------------------------------------------------+
  2. |      region |                                 entidad |        pob |
     |      Centro |                              21 Puebla |  5,779,829 |
     |-------------------------------------------------------------------|
     |                                totpob                             |
     |                            20,955,692                             |
     +-------------------------------------------------------------------+

     +-------------------------------------------------------------------+
  3. |      region |                                 entidad |        pob |
     |      Centro |                           11 Guanajuato |  5,486,372 |
|-------------------------------------------------------------------|
|                              totpob                               |
     |                            26,442,064                             |
     +-------------------------------------------------------------------+
     +-------------------------------------------------------------------+
```

En este punto podemos visualizar de manera independiente el uso de _n y _N, aunque ambos emitirán los mismos valores para la última observación.

```
. by region: list region totpob if _n == _N
-------------------------------------------------------------------------------
->region = Centro
     +----------------------+
     | region       totpob |
     |----------------------|
  9. | Centro   37,652,696 |
     +----------------------+
-------------------------------------------------------------------------------
->region = C Pacif
     +----------------------+
```

```
|  region           tot pob |
     |----------------------|
  7. | C Pacif   23,395,744 |
     +----------------------+
-------------------------------------------------------------------------------
->region = DF
     +--------------------+
     | regiontotpob |
     |--------------------|
1. |     DF   8,851,080 |
+--------------------+
-------------------------------------------------------------------------------
->region = Golfo y Sur
     +---------------------------+
     |      region      tot pob |
     |---------------------------|
  6. | Golfo y Sur   18,781,972 |
     +---------------------------+
-------------------------------------------------------------------------------
->region = Norte
     +----------------------+
     | region      tot pob |
     |----------------------|
  9. |  Norte   23,655,046 |
     +----------------------+
```

Ahora procedamos a obtener el promedio de población de las entidades por región.

```
. by region: egen media pob = mean(pob)
. format %12.0gc media pob
. list region entidad pob media pob, sepby(region)

     +-------------------------------------------------------------------+
  1. |      region |                                 entidad |        pob |
     |      Centro |                              15 México | 15,175,862 |
     |-------------------------------------------------------------------|
     |                               totpob                              |
     |                           15,175,862                              |
     +-------------------------------------------------------------------+

     +-------------------------------------------------------------------+
  2. |      region |                                 entidad |        pob |
     |      Centro |                              21 Puebla |  5,779,829 |
     |-------------------------------------------------------------------|
     |                               totpob                              |
     |                           20,955,692                              |
     +-------------------------------------------------------------------+

     +-------------------------------------------------------------------+
  3. |      region |                                 entidad |        pob |
     |      Centro |                           11 Guanajuato |  5,486,372 |
     |-------------------------------------------------------------------|
     |                               totpob                              |
     |                           26,442,064                              |
     +-------------------------------------------------------------------+
```

```
     +-------------------------------------------------------------+
  4. |      region |                              entidad |      pob |
     |      Centro |                         13 Hidalgo |  2,665,018 |
     |-------------------------------------------------------------|
     |                              totpob                         |
     |                         29,107,080                          |
     +-------------------------------------------------------------+

     +-------------------------------------------------------------+
  5. |      region |                              entidad |      pob |
     |      Centro |                 24 San Luis Potosí |  2,585,518 |
     |-------------------------------------------------------------|
     |                              totpob                         |
     |                         31,692,600                          |
     +-------------------------------------------------------------+
```


Conclusiones

El trabajo sienta las bases para que, a partir de la réplica, los usuarios posean los fundamentos para construir rutinas más elaboradas a partir de la generación de bases de datos. Por otro lado, muestra rutinas básicas en términos de estadística descriptiva de una variable. De la misma manera, describe cómo elaborar tablas de variables, que bien pueden entenderse como tablas de frecuencia o con atributos de ciertas mediciones estadísticas (media aritmética, rango, entre otras).

De llevar a cabo las rutinas presentadas en el documento, el usuario puede ser capaz de adentrarse en el uso de otros paquetes estadísticos, en el entendido de que los algoritmos, aunque diferentes por razones de patentes y derechos de autor, son completamente iguales en términos teóricos. Sin embargo, una presentación en paquetes alternos, ya sean de mercado o libres, es tarea de otro documento, es decir, lo proyectamos como una investigación futura.

Bibliografía

Adkins, Lee and Carter Hill (2008), *Using Stata for principles of econometrics,* USA, Stata Press.

Baum, Christopher (2006), *An introduction to modern econometrics using Stata*, USA, Stata Press.

Instituto Nacional de Estadística y Geografía (2015), "Censo de Población y Vivienda 2010" en <http://www.inegi.org.mx/est/contenidos/proyectos/accesomicrodatos/cpv2010/default.aspx>, consultada en septiembre de 2015.

Instituto Nacional para el Federalismo y el Desarrollo Municipal (2015), "Enciclopedia de los municipios y delegaciones de México" en <http://www.inafed.gob.mx/work/enciclopedia/index.html>, consultada en septiembre de 2015.

Spiegel, Murray and Larry Stephens (2009), *Estadística,* México, Mc Graw Hill.