Introduction to Geographic Information Science

Week 04 Stata Commands

Christopher G. Prener, Ph.D.

Spring, 2016

Contents

1 Labeling Dataset

1.1 Description

Dataset labels are a way add metadata to a dataset's name. The can be used to provide additional documentation about a file.

1.2 Syntax

`label data` ["*label*"]

1.3 Examples

When you open the `census.dta` file that comes with Stata, you see its dataset label:

```
. sysuse census.dta
(1980 Census data by state)
```

The text that appears in the line below the `sysuse` command - `(1980 Census data by state)` - is the dataset label.

To apply a new dataset label, or edit an existing one, use the `label data` command once the dataset is open:

```
. save census80.dta, replace
file census80.dta saved

. label data "Edited 1980 Census Data"
```

Next time you open that data, you will see the new label reported:

```
. clear

. use census80.dta
(Edited 1980 Census Data)
```

Note that labels are restricted to a maximum of 80 characters. If you try to apply a label that is too long, you will receive the following error:

```
. label data "Edited 1980 Census Data - Lorem ipsum dolor sit amet, semper cursus ac,
vel praesent non sit"
note:  label truncated to 80 characters

. save census80.dta, replace
file census80.dta saved

. clear

. use census80.dta
(Edited 1980 Census Data - Lorem ipsum dolor sit amet, semper cursus ac, vel prae)
```

You can drop a label without applying a new one by simply not specifying a label:

```
. label data

. save census80.dta, replace
file census80.dta saved

. clear

. use census80.dta
```

The file `census80.dta` now opens with no label.

1.4 Notes

Additional details for the `label` suite of commands can be found in the Stata documentation for those commands (link).

2 Labeling Variables

2.1 Description

Variable labels are a way to add metadata to a single variable. Variable labels are most often used to add detail about a variable's contents. A dataset may have variables named `gender` and `gender2`, and the variable labels for each provide a way to differentiate between these two variables.

Stata also uses variable labels when it lists variables using the `describe` command, when it generates tables using the `tabulate` command, and when it create output using other commands.

2.2 Syntax

`label variable` *varname* ["*label*"]

2.3 Examples

The variable labels on a pre-existing dataset can be viewed quickly in the Variables manager located on the right side of Stata'a main window, though you may have to resize this portion of the screen to fully read the labels.

You can also see variable names when you list a dataset's contents using the `describe` command:

```
. sysuse census.dta
(1980 Census data by state)

. describe state state2 region pop

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
state           str14   %-14s                 State
state2          str2    %-2s                  Two-letter state abbreviation
region          int     %-8.0g     cenreg     Census region
pop             long    %12.0gc               Population
```

Variable labels can be easily modified using the `label variable` command:

```
. label variable state "Full state name"

. describe state

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
state           str14   %-14s                 Full state name
```

Note how the variable label has now changed from the first instance of the describe command above to the second instance.

Variable labels can also be added to new variables, or variables without a pre-existing label, using the same syntax:

```
. generate newVar = .

. label variable newVar "new empty variable"
```

Finally, variable labels can be removed simply by omitting the text of the label from `label variable` command:

```
. label variable state

. describe state

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
state           str14   %-14s
```

2.4 Notes

Additional details for the `label` suite of commands can be found in the Stata documentation for those commands (link).

3 Labeling Values

3.1 Description

Value labels are the words we associate with numeric values that give them meaning. We often given so-called 'binary' variables (that have only the values `0` and `1`) labels so that we recall that `0` means `no` and `1` means `yes`.

3.2 Syntax

```
label dir
label list [lbllist]
label define lblname # "label" [# "label"...]
label values varlist [lblname | .  ]
```

3.3 Examples: Value Label Utilities

There are a number of key tasks that we do around labeling values. We must define labels before we can apply they to a variable. Once defined, value labels are stored in Stata's memory along with our data. We can see pre-existing value labels in the `census.dta` dataset:

```
. sysuse census.dta
(1980 Census data by state)

. label dir
cenreg
```

We see from the output that there only one label defined in the `census.dta` dataset. The `label dir` command will always list all labels that are defined in the current dataset. If you want to see the defined values, you can use the `label list` command:

```
. label list

cenreg:
           1 NE
           2 N Cntrl
           3 South
           4 West
```

The `label list` command has an optional label list. If a dataset has multiple defined labels, you can choose to list on some of the labels by specifying their names:

```
. label list cenreg

cenreg:
           1 NE
           2 N Cntrl
           3 South
           4 West
```

Finally, you can see which variables have labels applied to them by using the `describe` command:

```
. describe

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
state           str14   %-14s                 State
state2          str2    %-2s                  Two-letter state abbreviation
region          int     %-8.0g     cenreg     Census region
pop             long    %12.0gc               Population
```

Note how the `region` variable has a label name specified under 'value label' while none of the other variables do.

3.4 Examples: Defining Value Labels

To define new labels, use the `label define` command. This command must combine unique numerical values with text strings. These text strings are best kept short and clear. Ambiguous or long text can make later analysis more difficult. The following example defines a label named `binary`:

```
. label define binary 0 "no" 1 "yes"
```

The double quotes around the text strings are optional unless the text string are more than one word. The following example defines multi-word labels for a label named `region`:

```
. label define region 1 "new england" 2 midwest 3 south 4 west
```

Note that value labels and variable labels can share the same names. The previous example could be used in the `census.dta` dataset that comes installed with Stata even though that dataset contains a variable named `region`. Students who are just beginning to use Stata, however, may find it useful to differentiate label names and variable names. For example, in a dataset like `census.dta` that has a variable named `region`, it may be helpful to name a variable `regionlbl`, where `lbl` refers to the standard Stata abbreviation for 'label':

```
. label define regionlbl 1 "new england" 2 midwest 3 south 4 west
```

3.5 Examples: Applying Value Labels

Once defined, labels must then be applied to the relevant variables. To apply labels to specific variables, use the `label values` command. The following example defines the value label `regionlbl` and then applies it to the variable `region`:

```
. sysuse census.dta
(1980 Census data by state)

. label define regionlbl 1 "new england" 2 midwest 3 south 4 west

. label values region regionlbl
```

You need to be careful when spelling value label names. Misspelling these will **not** generate an error in Stata even if no label has been defined with that name. For example, the following does

not produce an error in the dataset `census.dta` even though no such label name has been defined. It will, however, remove the existing value labels from the variable:

```
. tabulate region

    Census |
    region |      Freq.     Percent        Cum.
-----------+-----------------------------------
        NE |          9       18.00       18.00
  N Cntrl |         12       24.00       42.00
     South |         16       32.00       74.00
      West |         13       26.00      100.00
-----------+-----------------------------------
     Total |         50      100.00

. label values region region2

. tabulate region

    Census |
    region |      Freq.     Percent        Cum.
-----------+-----------------------------------
         1 |          9       18.00       18.00
         2 |         12       24.00       42.00
         3 |         16       32.00       74.00
         4 |         13       26.00      100.00
-----------+-----------------------------------
     Total |         50      100.00
```

Finally, the `label values` command allows you to specify a full *varlist*. The following example uses hypothetical data:

```
. label define binary 0 "no" 1 "yes"

. label values var1 var2 var3 binary
```

When a *varlist* is specified, Stata assumes that the final item in the `label values` is the label name.

3.6 Example: Removing Value Labels

To intentially remove value labels, specify a period (`.`) instead of a label name when using the `label values` command:

```
. sysuse census.dta
(1980 Census data by state)
```

(*Continued on next page*)

```
. tabulate region

     Census |
     region |      Freq.     Percent        Cum.
------------+-----------------------------------
         NE |          9       18.00       18.00
    N Cntrl |         12       24.00       42.00
      South |         16       32.00       74.00
       West |         13       26.00      100.00
------------+-----------------------------------
      Total |         50      100.00

. label values region .

. tabulate region

     Census |
     region |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |          9       18.00       18.00
          2 |         12       24.00       42.00
          3 |         16       32.00       74.00
          4 |         13       26.00      100.00
------------+-----------------------------------
      Total |         50      100.00
```

3.7 Notes

Additional details for the `label` suite of commands can be found in the Stata documentation for those commands (link).

4 Notes

4.1 Description

Notes can be used to provide documentation that is saved with a Stata dataset as metadata. Many researchers use notes to store key pieces of information about the providence, structure, and changes made to a variable. Notes can be exported in a variety of ways, and can therefore be used to construct study documentation.

4.2 Syntax

notes [*evarname*]: *note text*
notes
notes *evarlist*
notes drop *evarlist*

4.3 Examples: Creating Dataset Notes

Dataset notes are not associated with any particular variable. Instead, they can be used to provide notation about the dataset as a whole:

```
. sysuse census.dta
(1980 Census data by state)

. notes _dta:  Census data is from 1980

. notes _dta:  Data comes pre-installed with Stata
```

Note that the text `_dta` is included in the syntax of the note to specify that the note should be attached to the dataset itself.

4.4 Examples: Creating Variable Notes

Notes that are associated with specific variables are applied by including the variable name after the notes command:

```
. sysuse census.dta
(1980 Census data by state)

. notes region:  Region names updated to reflect more recent changes to designations
by the Census Bureau by Chris on 03 Feb 2016

. notes region:  Changes to variable label made by Chris on 03 Feb 2016

. notes pop:  Changes to variable label made by Chris on 03 Feb 2016
```

4.5 Examples: Displaying Notes

The simplist way to view notes is by specifying the notes command without any additional syntax. The following examples assume that the notes "created" in the previous two sections:

```
. notes

_dta:
  1.  Census data is from 1980
  2.  Data comes pre-installed with Stata

region:
  1.  Region names updated to reflect more recent changes to designations by the
      Census Bureau by Chris on 03 Feb 2016
  2.  Changes to variable label made by Chris on 03 Feb 2016

pop:
  1.  Changes to variable label made by Chris on 03 Feb 2016
```

Notes for specific items, such as a single variable, can be viewed by specifying an *evarlist* (meaning it can also contain `_dta`):

```
. notes _dta pop

pop:
  1.  Changes to variable label made by Chris on 03 Feb 2016

_dta:
  1.  Census data is from 1980
  2.  Data comes pre-installed with Stata
```

4.6 Examples: Removing Notes

Notes can be removed in one or more variables an *evarlist* (meaning it can also contain `_dta`):

```
. notes drop _dta pop
(3 notes dropped)
```

All notes currently included in a dataset can be dropped by including `_all` as the *evarlist* instead of a list of variables:

```
. notes drop _dta pop
(6 notes dropped)
```

4.7 Notes

Additional details for the notes suite of commands can be found in the Stata documentation for those commands (link).

5 Removing Variables

5.1 Description

The `drop` and `keep` commands allow you to alter the order the dataset by removing variables. Depending on the number of variables you are removing, `keep` may be a more efficient approach to removing variables from a dataset.

5.2 Syntax

`drop` *varlist*
`keep` *varlist*

5.3 Examples: Dropping Variables

The following example removes each variable after `pop` in the `census.dta` dataset that comes preinstalled with Stata:

```
. sysuse census.dta
(1980 Census data by state)

. drop poplt5 pop5_17 pop18p pop65p popurban medage death marriage divorce
```

This leaves only the variables `state, state2, region,` and `pop` in the dataset.

5.4 Examples: Keeping Variables

The following example results in the same final dataset as the previous section does, but with a different approach:

```
. sysuse census.dta
(1980 Census data by state)

. keep state state2 region pop
```

This creates the same final dataset as the previous section, but without specifying as many variables in the *varlist*.

5.5 Notes

Additional details for the `drop` and `keep` commands can be found in the Stata documentation for those commands (link).

6 Removing Observations

6.1 Description

The `drop` and `keep` commands allow you to alter the order the dataset by removing

6.2 Syntax

`drop if` *expression*
`keep if` *expression*

6.3 Examples: Dropping Observations

The following example removes each observation (i.e. state) that is not in New England in the `census.dta` dataset that comes pre-installed with Stata:

```
. sysuse census.dta
(1980 Census data by state)

. drop if region != 1
```

Note that this expression uses the "does not equal" relational operator.

6.4 Examples: Keeping Observations

The following example results in the same final dataset as the previous section does, but with a different approach:

```
. sysuse census.dta
(1980 Census data by state)

. keep if region == 1
```

Note that this expression uses the "equals" relational operator, which uses two equal signs (==). This creates the same final dataset as the previous section.

6.5 Notes

Additional details for the `drop` and `keep` commands can be found in the Stata documentation for those commands (link).

7 Common Errors

7.1 Error 111: Variable Not Found

When referencing a variable using any command that includes a *varlist*, you may receive this error:

```
. sysuse census.dta
(1980 Census data by state)

. list state3 in 1/5
variable state3 not found
r(111);
```

This error indicates one of two conditions: (1) the variable does not exist in the dataset or (2) there is a misspelling in the variable name.

7.2 Error 198: Invalid Syntax

When defining a new value label, you may receive this error if you have not used double quotes to surround multi-word labels:

```
. label define region 1 new england 2 midwest 3 south 4 west
invalid syntax
r(198);
```

The label `new england` should be specified with double quotes like this:

```
. label define region 1 "new england" 2 midwest 3 south 4 west
```

7.3 Error 198: Option Not Allowed

If an option is misspelled or is not actually a valid option for a command, you will see the following error. In this case, the option for the `summarize` command has been misspelled - it is `detail`, not `detailed`:

```
. summarize pop, detailed
option detailed not allowed
r(198);
```

7.4 Error 199: Command Not Recognized

If a command is misspelled, it will generate the following error. In this case, the command `describe` has been misspelled:

```
. dscribe type
command dscribe is unrecognized
r(199);
```

If the command was written as part of a user-installed package, this error could also indicate that package has not been installed locally.