Stata Recitation - Week 6 - Strings and Labels

McCourt School of Public Policy, Georgetown University

Key Ideas:

- Create and modify variable and value labels
- Search variable labels
- Use string functions
- Destring variables

```
help label
```

Labels

- Three types of labels, data set, variable, and values

Variable Labels

- Show up in variable window
- Show command syntax in help file
- Use example from previous recitation ago: “‘ clear sysuse nlsw88.dta

gen weekwage = wage*hours label variable weekwage “Ave. Weekly Pay”

```
* Changes can also be made in the Variables Manager `Data > Variables Manager`
* Remember to put the resulting "label" command into your do-file.
```

generate agesq = age^2

```
* Use the varibles manager to label, then add label to do-file.
```

label variable agesq “Age Squared”

```
* A very useful function when you start working with large data sets
* `lookfor`: searches variable names and labels
```

lookfor age

```
### Data Set Labels

* Data set label is similar to variable label, but applies to entire data set.
* Show syntax in help file
* Data labels show up when a data set is opened and in the describe command.
* Useful when you have to save a modified version of your data.

* Label and save data in do-file
```

label data "Modified data set from recitation 6" save "...\nlsw88 - recitation 6.dta"

```
* Clear and re-open saved data to see data set label.
* Describe data to show data set label.
* Add these commands to the end of the do-file, and comment

* Reopen data to demonstrate data label
```

clear use "...\nlsw88 - recitation 6.dta"

```
* Data label can also be seen with describe
```

describe, short

```
### Value Labels
* Value labels are more complicated than data or variable labels
* Value labels are defined and exist independently of variables
* Show value labels using describe and labelbook
* Individual labels can be listed:
```

label list occlbl

```
#### Labeling values is a two-step process
 1. define label
 2. apply label to variable


#### Example from last week's problem set:

* Create an indicator called tenure20 for people with 20 or more years tenure.
```

gen tenure20=0 replace tenure20=1 if tenure>=20 replace tenure20=. if tenure==.

```
* Label variable
```

label variable tenure20 “Tenure of 20 or more years”

```
* Create value label
```

label define tenure20lbl 0 “Less than 20 years” 1 “20 or more years”

```
* Apply value label
```

label values tenure20 tenure20lbl

tab tenure20

```
* Value label management can be done with the "Manage Value Labels" dialogue box:
* `Data > Data utilities > Label utilities > Manage value labels`
* Applying value labels to variables can be done in the Variable Manager
* As always, commands should be recorded in do-file

* Another example from last weeks problem set:
* Create an indicator variable called once_married, for people who were once married, but ar
```

gen once_married=0 replace once_married=1 if married==0 & never_married==0 replace once_married=. if married==. | never_married==.

```
* Label variable and values
```

label variable once_married “Once married, but not currently married” label define once_marriedlbl 1 “Once married” 0 “Never or currently married” label values once_married once_marriedlbl

```
## Strings
 - We have seen strings, but we haven't really worked with them.



### String values always go in quotes
#### Example: Look at key variables for a single vehicle
```

list make mpg weight length if make==“Buick Century”

```
* Strings are case sensitive

#### Example 2: Create an indicator for all Buick vehicles
```

gen buick=0 replace buick=. if make==“”

```
* Missing value for string variables is an empty string, `""`
```

replace buick=1 if inlist(make , “Buick Century” , “Buick Electra” , “Buick LeSabre” , “Buick Opel” , “Buick Regal” , “Buick Riviera” , “Buick Skylark”)
list make mpg weight length if buick==1

```
### String Functions
```

help string functions

```
* String functions request string inputs (s, s1, s2, etc.)
* These can be actual strings or the names of string variables.
* Strings should be in quotes, string variables should not be in quotes.

#### Example: length(s)
```

clear sysuse auto

- actual string, use quotes gen len_1 = length(“test”) browse make len_*
- variable, no quotes gen len_2 = length(make) browse make len_*
- actual string, use quotes gen len_3 = length(“make”) browse make len_*
- variable that doesn’t exist -> error gen len_4 = length(test) “‘

Example 2: Create an indicator for all Buick vehicles

```
gen make1 = word(make,1)
browse make make1

gen buick=0
replace buick=. if make1==""
replace buick=1 if make1=="Buick"
```

Converting between strings and numbers

- You probably won't need these commands until thesis, so we won't cover them in depth at this time. Just know that they exist, and how to find help on them.

'Destring/tostring'

- Sometimes a variable is stored as a string when it should be a number.
- This is a frequent problem after importing data from Excel.
- In this case, you can convert the string to a number using destring.

```
help destring
```

- See examples in the help file and manual.

`Encode/decode`

- A related but different problem:
- A categorical variable exists as a string and needs to be changed to a number.
- Or the other way around.

```
help encode
```

- See examples in help file and manual.