EPIB 613: Labels and recoding 25 Oct. 2012

Labels and notes for datasets

Stata lets you label your dataset using the **label data** command followed by a label of up to 80 characters.

```
label data "1978 Automobile Data"
```

The label will be associated with the dataset each time you use it in the future

```
. use autolab, clear
(1978 Automobile Data)
```

You can also add notes to the dataset

```
note: data from Consumer Reports
note: Datsuns look cool but Toyotas get better mileage
```

To display all notes in a dataset, type **notes**

```
. notes

_dta:
  1.  data from Consumer Reports
  2.  Datsuns look cool but Toyotas get better mileage
```

Variable Labels

To check current labels on variables you can use **nmlab** (or **codebook** or **describe** also provide similar information)

```
. nmlab

make
price
rep78
trunk    Trunk space (cubic feet)
foreign  Type of car, domestic or foreign
```

The syntax to label a variable is **label variable varname "variable label"**

```
label var make "Make of car"
label var price "Price of car"
label var rep78 "Repair Record 1978"
```

To change an existing variable label you use the same syntax and it just replaces the previous label

```
label var rep78 "Automobile Repair Record for the year 1978"
```

To remove a variable label use **label var varname**

```
label var trunk
```

Just as you can attach notes to the dataset you can also attach notes to specific variables

```
notes foreign: cars not manufactured in the US or Canada
```

Value Labels

Value labels assign text labels to the numeric values of a categorical variable. This is done in two steps:

1. Define the value label

```
label define origin 0 Domestic 1 Foreign
```

2. Associate the value label with a variable

```
label values foreign origin
```

```
→ label (the) values (of variable) foreign (according to labels in) origin
```

Before adding the value label:

```
. tab foreign

Type of car,|
domestic or |
    foreign |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         52       70.27       70.27
          1 |         22       29.73      100.00
------------+-----------------------------------
      Total |         74      100.00
```

After adding the value label:

```
. tab foreign

Type of car,|
domestic or |
    foreign |      Freq.     Percent        Cum.
------------+-----------------------------------
   Domestic |         52       70.27       70.27
    Foreign |         22       29.73      100.00
------------+-----------------------------------
      Total |         74      100.00
```

To see all value labels in a dataset use **label list**

**Remember that although there is a text label the variable is still considered numeric
For example, if you want to list the foreign cars you will get an error if you refer to the label

```
. list make if foreign=="Foreign"
type mismatch
```

```
. list make if foreign==1

     +-----------------+
     | make            |
     |-----------------|
 53. | Audi 5000       |
 54. | Audi Fox        |
 55. | BMW 320i        |
 56. | Datsun 200      |
```

However, there is syntax if you want to refer to the label value rather than the numeric value

```
list make if foreign=="Foreign":origin
```

A useful command to avoid this confusion is **numlabel,** which adds the numeric value to the label

```
numlabel origin, add
```

```
. tab foreign

    Type of |
       car, |
domestic or |
    foreign |      Freq.     Percent        Cum.
------------+-----------------------------------
0. Domestic |         52       70.27       70.27
 1. Foreign |         22       29.73      100.00
------------+-----------------------------------
      Total |         74      100.00
```

To remove the numlabel use **numlabel labelname, remove**

Recoding variables

The **recode** command is useful for recoding the values of a variable and/or combining values into new categories.

For example, we want to create a new variable that categorizes trunk size into 3 categories: Small (<10 cubic ft), Medium (10-19 cubic ft), Large (20+ cubic ft)

One way to do this is to use **generate** followed by **replace**

```
gen trunkcat=.
replace trunkcat=1 if trunk<10
replace trunkcat=2 if trunk>=10 & trunk<20
replace trunkcat=3 if trunk>=20 & trunk!=.
```

```
. tab trunkcat, miss

  trunkcat |      Freq.     Percent        Cum.
------------+-----------------------------------
         1 |         14       18.92       18.92
```

```
          2 |         50       67.57       86.49
          3 |         10       13.51      100.00
------------+-----------------------------------
      Total |         74      100.00
```

And then we still need to create value labels

```
label define trunkcatlab 1 "Small: <10 cubic ft" 2 "Medium: 10-19 cubic ft" 3
"Large: 20+ cubic ft"
label values trunkcat trunkcatlab
```

```
. tab trunkcat

             trunkcat |      Freq.     Percent        Cum.
-----------------------+-----------------------------------
   Small: <10 cubic ft |         14       18.92       18.92
Medium: 10-19 cubic ft |         50       67.57       86.49
   Large: 20+ cubic ft |         10       13.51      100.00
-----------------------+-----------------------------------
                 Total |         74      100.00
```

Or we can do this all in one step using **recode**

```
recode trunk (min/9=1 "Small: <10 cubic ft") (10/19=2 "Medium: 10-19 cubic
ft")(20/max=3 "Large: 20+ cubic ft"), gen(trunkcat2) label(trunkcat2lab)
```

```
RECODE of trunk (Trunk |
   space (cubic feet)) |      Freq.     Percent        Cum.
-----------------------+-----------------------------------
   Small: <10 cubic ft |         14       18.92       18.92
Medium: 10-19 cubic ft |         50       67.57       86.49
   Large: 20+ cubic ft |         10       13.51      100.00
-----------------------+-----------------------------------
                 Total |         74      100.00
```

Note the new variable label generated by **recode**

```
. nmlab trunkcat2

trunkcat2  RECODE of trunk (Trunk space (cubic feet))
```

Creating indicator variables from categorical variables

To generate indicator (1/0) variables for each level of a categorical variable we can use **tab varname, gen(newvarname)**

```
tab trunkcat, gen(trunkcat)
```

This creates three new indicator variables named trunkcat1, trunkcat2, and trunkcat3 with the following variable labels:

```
. nmlab trunkcat*

trunkcat1  trunkcat==Small: <10 cubic ft
trunkcat2  trunkcat==Medium: 10-19 cubic ft
trunkcat3  trunkcat==Large: 20+ cubic ft
```

We can generate a generic yes/no value label and attach to all indicator variables

```
label define yesno 1 yes 0 no
label values trunkcat1 trunkcat2 trunkcat3 yesno
```

```
. tab trunkcat1

trunkcat==S |
  mall: <10 |
   cubic ft |      Freq.     Percent        Cum.
------------+-----------------------------------
         no |         60       81.08       81.08
        yes |         14       18.92      100.00
------------+-----------------------------------
      Total |         74      100.00
```

Creating a variable containing quantile categories

Use the command **xtile** to create a new variable that categorizes into regularly spaced intervals (tertiles, quartiles, quintiles, etc)

```
xtile priceq=price, nq(4)
```

```
. tab priceq

4 quantiles |
   of price |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |         19       25.68       25.68
          2 |         18       24.32       50.00
          3 |         19       25.68       75.68
          4 |         18       24.32      100.00
------------+-----------------------------------
      Total |         74      100.00
```