

Lab Worksheet III

Preparing data

In this exercise you will use some real data. Download from LEARN the dataset named **essuk12.dta**, which corresponds to the sixth round of the European Social Survey. This dataset has been simplified for today's tutorial's sake and it comprises information on the UK in 2012. Open the dataset with Stata by clicking on **File→Open** and select **essuk12.dta**. You should now see some variables listed down in the *variables window* on the right side of your screen.

Today's task is to do some light data management, which entails:

- Exploring your dataset,
- renaming variables, and
- recoding variables.

Data exploration

The command describe, as its name indicates, describes data in memory or in file. Many commands in Stata can be abbreviated, thus `describe` can also be used as `de` (notice how I underlined the shortest abbreviation possible above). You can use this command to describe the entire dataset, or just a selection of variables. For example, if you typed

```
describe
```

you would be describing all variables in the dataset. If you wanted to describe only some variables you could type

```
de Tvtot Vote Happy
```

and describe only those three variables of interest. You can get more information on `describe` by using another command:

```
help describe
```

The `help` command opens Stata's manual and shows detailed information on the command you need help with. Remember to consult Stata's manual whenever you are not sure how to use a command. Now try to use `describe` yourself and answer these questions:

1. How many variables are there in the dataset?
2. How many observations does this dataset have?
3. What is the difference between "variable label" and "value label"?

Another useful command is `codebook` (if you type `help codebook` you will see that it cannot be abbreviated). This one also describes the contents of a dataset, however it provides you with a more detailed description of the data. Just like `describe`, you can use `codebook` on the whole dataset or on a selection of variables of interest. Try the following command and answer the questions:

```
codebook Tvtot Gndr Sclmeet
```

1. How many response categories does “Sclmeet” have?
2. How is “Gndr” coded?
3. What does “Tvtot” measure?

Renaming variables

When you download a dataset, variables’ names not always make sense. Consider the variable “Sclmeet”; if you use `describe` and `codebook` on this variable you can find what it measures. Let’s rename the variable to something else, for example “Socmeet” (mind the capital -S). To achieve this type the following:

```
rename Sclmeet Socmeet
```

You can use `help rename` to find out more about the command `rename` (for instance, it can be abbreviated to `ren`). Now try to rename the variables in your dataset to the following new names (mind all the capital letters):

- Gndr to Gender.
- Agea to Age.
- Eiscd to Education.
- Hinctnta to Houseincome.

Although not mandatory, it is common to specify variable names with lower cases. Right now all your variables start with a capital letter, so if you wanted to change their names you could use the command `rename` on each variable and change the first letter only. However, this would take quite a while as you would have to repeat the command twelve times. Stata has a way to deal with problems like this one. Instead of typing the same command twelve times try:

```
re _all , lower
```

Remember that `re` stands for `rename`. Above you used `_all` to tell Stata that you would like to modify all variables in the dataset. Notice that now there is a comma in the command. That comma tells Stata that you are going to use a *command option* (pretty much all commands in Stata allow for options, you can find more about these options in the help page of each command). The option `lower` tells Stata that you want only lower cases.

Recoding variables

Recoding variables is a fundamental part of analysing data. The command `recode`, as its name indicates, recodes variables in your dataset. Use `help` to find out more about its syntax. Let's now try to recode the variable "gender", remember how it is coded (you can find out by using `codebook`). You want to get the following:

- 0 = Male
- 1 = Female

To achieve this you need to type:

```
recode gender (1=0) (2=1), generate(gender2)
```

The command above is telling Stata to recode the variable "gender" so the value 1 is now 0 (1=0), and the value 2 is now 1 (2=1). In short, the "old value" comes first and then you tell Stata the "new value." However, there is one more thing you need to consider. Recoding variables modifies the data, so it is always a good idea to keep intact the original variable (in this case "gender"). The command `recode` allows as a command option another command, in this case the command `generate` (which can be abbreviated to `g`, but people usually use it as `gen`). As you can imagine, `generate` creates a new variable in your dataset. Above, you recoded the variable "gender" and, after recoding it, you created a new variable named "gender2" (mind the round brackets).

Now try yourself to recode the variable "vote" to (do not forget to generate a new variable):

- 1 = Yes
- 0 = No
- -9 = Everything else

Remember that you can use `codebook` to find out how a variable is currently coded. After recoding the variable you can run `codebook` again to see how your new codes look like. You will notice that the new coding does not have appropriate value labels, but luckily you know now how to create and assign value labels!