Overview of temporary objects

- Sometimes you need intermediate variables, matrices or estimations for calculations or other purposes
- In programs, two issues might emerge:
  - You have to make sure that the name you give to the object does not already exist
  - Once you are done with the process, you don't need the object anymore, it is in your way
- Sometimes, preserve and restore can be helpful, but if the program also has permanent outputs, this might not be what you need
- Locals have traits which would solve these issues, but they can only store a certain kind of information

Overview of temporary objects

- For programming, you can use temporary objects which work similar to locals
- There are different types of temporary objects
  - tempvar
  - tempname
  - tempfile
- Used as command, all create temporary names, which then can be used to create objects which will be deleted after the program ends

What exactly is Stata doing?

- To find out how your (or any other) program is working, you can use set trace on and run the program
- Stata will then display you every single working step
- This is very time- and space-consuming, so remember to turn it off using set trace off
- The command is very useful to detect where in the routine an error occurred

Creating your own (e)return lists

- Like the standard Stata programs, results from self-written programs can be stored in r(), e(), or s()
- For this, you can specify the class of the programm as rclass, eclass, or sclass
  - rclass: return list for most commands
  - eclass: return list for estimation commands (for a recommended convention, see [P] eclass)
  - sclass: special return list for locals in subroutines (see next topic)