GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Advanced Programming

Stata Self-Learning Course

Introduction to the syntax

- Last lecture, you were familiarized with arguments and program classes
- But there is more information we can pass to programs (if, in, weights, options…), take for example

  reg income age sex if country == "France"
- How can we interpret the user‘s input in a meaningful way?
- How can we assure that the user only specifies reasonable input?
- For this, we can tell Stata which elements should be accepted by the program, using the syntax command

Introduction to the syntax

- In this example, regress needs at least one variable, and we want to allow the user to specify if/in specifications
- Thus, the syntax command would read as follows*:

```
syntax varlist(min=1) [if] [in]
```

- This specification ***demands*** at least one variable and ***allows*** if and in specifications optionally
- If the user now types

```
reg income age sex if country == "France"
```

syntax creates the following locals:

`varlist'	income age sex
`if'	if country == "France"
`in'	*(empty)*

* The real regress command also allows for weights and options, this is a simplified version

Introduction to the syntax

```
syntax varlist(min=1) [if] [in]
```

- In this example, varlist is the input/argument
- There are three types of arguments which can be used for the syntax command: varlist, namelist, and anything
- Both varlist and namelist have subtypes, but the respective local will always be `varlist‘/`namelist‘

Type	Subtype	Comment	Local
varlist	varlist	List of existing variables	varlist
	varname	Abbrev. for varlist(max=1)	
	newvarlist	List of names for new variables	
	newvarname	Abbrev. for newvarlist(max=1)	
namelist	namelist	List of names for matrices/locals/variables	namelist
	name	Abbrev. for name(max=1)	
anything		Any input (commas need to be in quotes)	anything

Parsing

- The syntax command is part of the ***parsing*** process
- This process describes the break-down of the user‘s entry into meaningful elements and conversion into a meaningful structure
- For example, syntax stores the elements passed to the command before if/in or other special elements in `varlist‘
- You can define your own parsing rules using gettoken and tokenize

Helpful options and commands

- The marksample command can be used after syntax to generate a temporary indicator variable marking the observations which should be used (e.g. according to if)
- Remember that quietly suppresses the Stata output but still stores the results in r() etc. if applicable