Automatic generation of documents

Rosa Gini Jacopo Pasquini
rosa.gini@arsanita.toscana.it
jacopo.pasquini@arsanita.toscana.it

Agenzia Regionale di Sanità della Toscana

September 13, 2006

Stata cannot hold a mouse

- While a common way for generating documents is *via* visual programs, such as MS Office or OpenOffice, it is completely impossible for *Stata* to produce documents this way, since it lacks eyes to format a table and hands to hold a mouse in order to cut–and–paste graphs.
- Nevertheless such documents as paper reports, web pages, screen presentations, . . . can also be obtained *via* the use of a *markup language*: HTML, LaTeX. . .

Stata can write text

- A markup language is a programming language for composing documents.
- The code that generates a document is simply a text file, which contains *both* the contents of the document *and* the instructions for the markup language to compose them in a beautiful way.
- *Stata* is able to write text, basically via the `file` suite of commands.
- The main topic of this communication is to summarize some experiences on how to make *Stata* produce documents this way.

What is an automated document?

An automated document is a piece of *Stata* code whose aim is

- to analyze data
- to write a piece of markup language code which presents the data in a fashionable form, i.e. to generate an *actual* document (in pdf, or HTML...)

Warning

It is useful to distinguish between

- the automated document, which is a text file (containing *Stata* code) for *Stata* to execute,
- the code of the actual document, which is a text file (containing the markup language code) for the markup language to interpret (possibily via compilation)
- the actual generated document, which is a pdf or HTML file for a user to read or print or browse or show

When to write an automated document?

It is worthwile investing time in producing an automated document when:

- the actual document that is needed is based on figures that can change (e.g. periodically): an automated document in fact easly generates an updated actual document when data change
- and/or the document is long but is structured: an automated document is a piece of *Stata* code, hence it writes the code of the actual document by means of cycles, conditional statements. . . that can repeat hundreds of times the same operations in very little time and without mistakes

Recall we're talking about documents for data to be analyzed and presented, so we think of it as a sequence of basic elements, such as tables, graphics,. . . , for data analysis.

Structure of a markup language code

Normally we produce several files for the markup language to interpret

- basic elements of the document, such as tables, graphics,...
- the *control code* of the document, which assembles the basic elements and provides the general structure, possibly adding tables of contents or similar features that allow navigating through the document

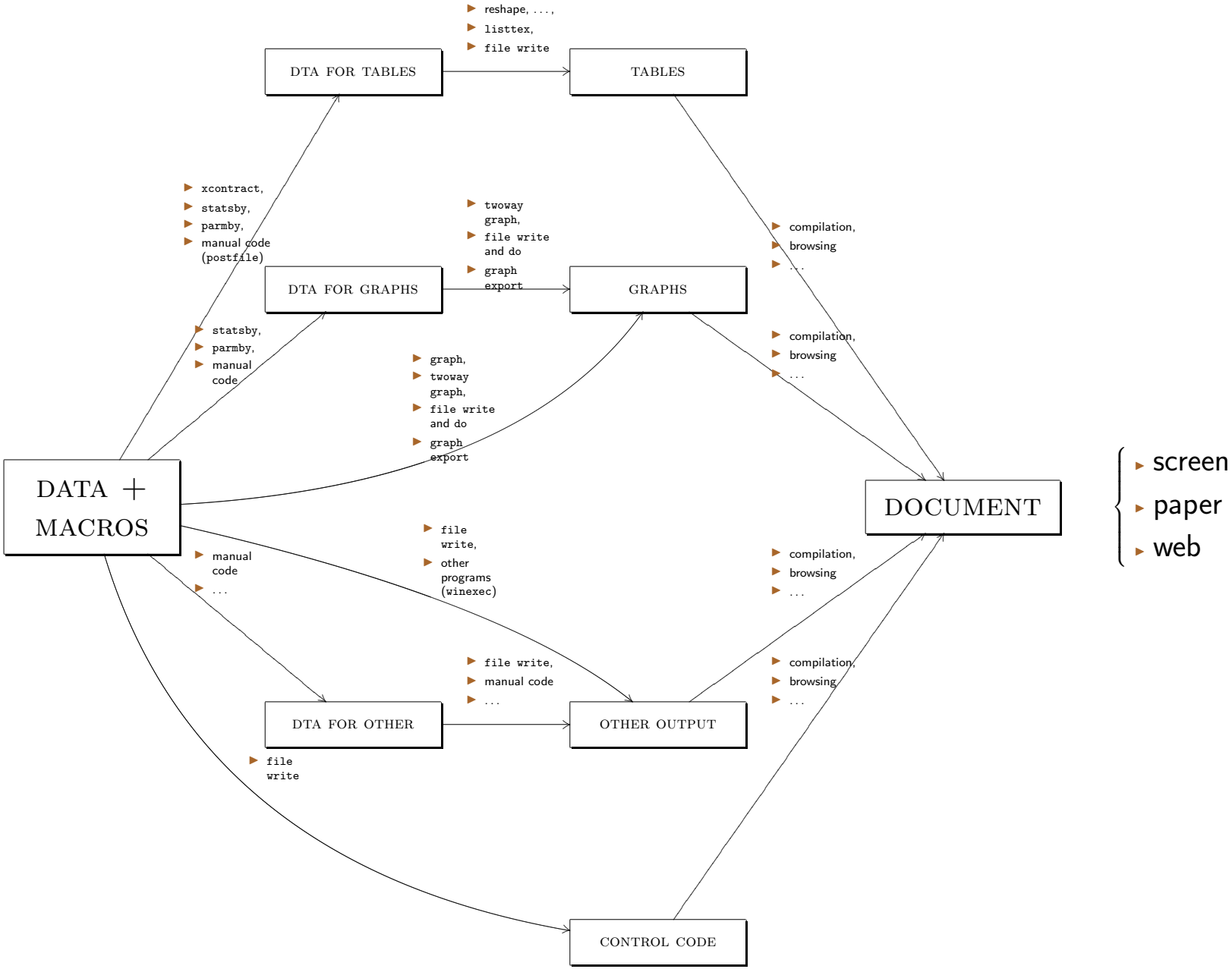
Traditional basic elements

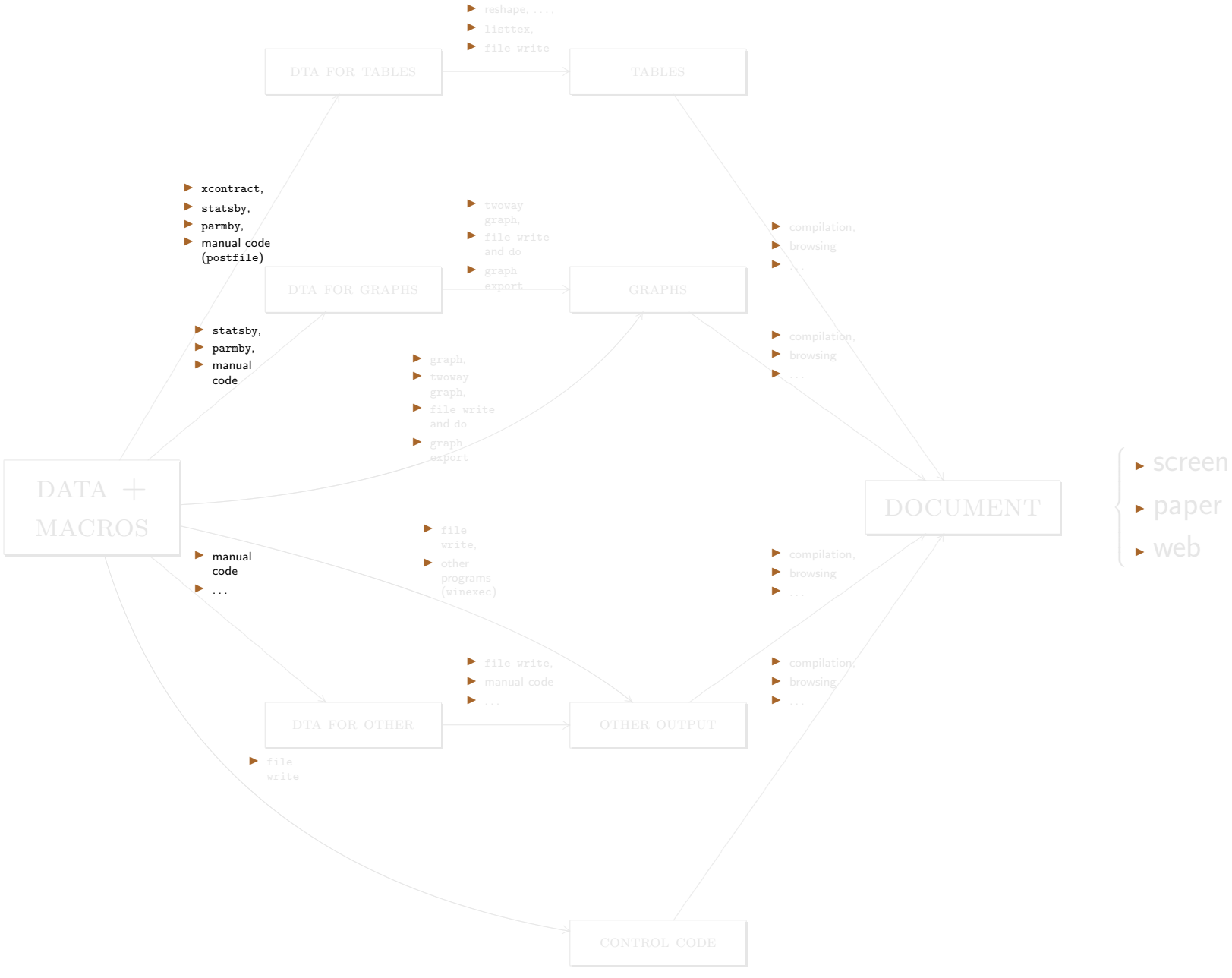
- Tables
- Graphics

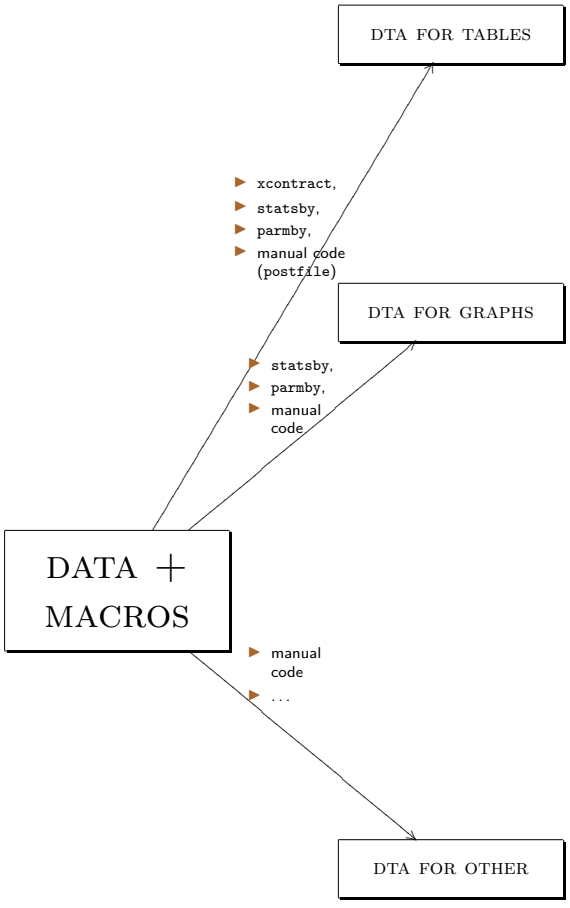
Other basic elements

- Lists
- Trees
- ...


Agenzia
Regionale
Sanità
Toscana

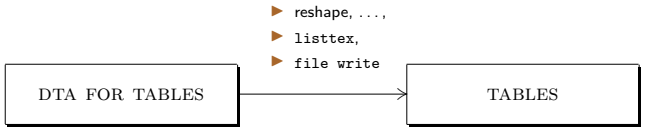

reshape, ...,
listtex,
file write
DTA FOR TABLES
TABLES
xcontract,
statsby,
parmby,
manual code (postfile)
twoway graph,
file write and do
graph export
DTA FOR GRAPHS
GRAPHS
compilation,
browsing
...
statsby,
parmby,
manual code
graph,
twoway graph,
file write and do
graph export
compilation,
browsing
...
DATA + MACROS
DOCUMENT
screen
paper
web
manual code
...
file write,
other programs (winexec)
compilation,
browsing
...
file write,
manual code
...
compilation,
browsing
...
DTA FOR OTHER
OTHER OUTPUT
file write
CONTROL CODE


reshape, ...,
listtex,
file write
DTA FOR TABLES
TABLES
xcontract,
statsby,
parmby,
manual code (postfile)
twoway graph,
file write and do
graph export
DTA FOR GRAPHS
GRAPHS
compilation,
browsing
...
statsby,
parmby,
manual code
graph,
twoway graph,
file write and do
graph export
compilation,
browsing
...
DATA + MACROS
DOCUMENT
screen
paper
web
manual code
...
file write,
other programs (winexec)
compilation,
browsing
...
file write,
manual code
...
DTA FOR OTHER
OTHER OUTPUT
compilation,
browsing
...
file write
CONTROL CODE



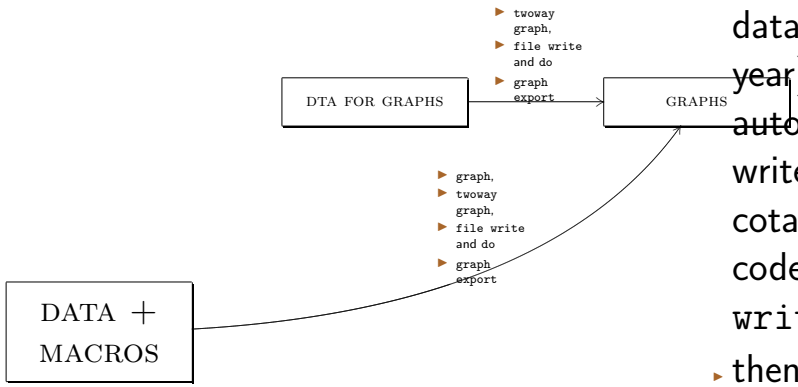
- this is the actual data analysis
- the results of analysis are stored (generally) as `dta` files, more conveniently stored in a separate directory
- the process of storing results of data analysis in `dta` is *not* homogeneously guaranteed by *Stata* commands, some of which do *not* support such a feature as `statsby`
- in case this support is not granted it necessary to use `postfile`

Fortunately Roger Newson's `parmby` provides such a support altogether for all estimation commands



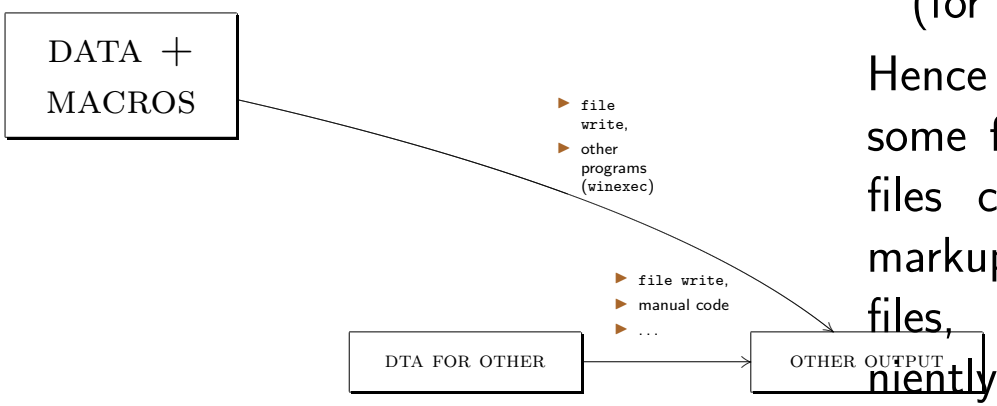
- this passage is granted by several commands by Roger Newson's, mainly `listtex`
- some feature of LaTeX's or HTML's tables only might be available through directly writing markup language's code (grouping columns,...)
- the more one knows the markup language the more one can make fine tuning of tables' layout

However since it's a simple matter of writing *text* one can generate code for dozens of tables altogether, according to some style parameters one can store in *Stata* macrosRemark that this passage produces some text files containing pieces of markup language code (hence with extensions `tex`, `htm`,...), more conveniently stored in a separate directory (`capture mkdir`) named something like `tables`



- *Stata*'s `graph` commands often do analysis and graph composition at the same time
- sometimes when producing graphs conditioned on the data (i.e. a line for each year) one can make the automated document write temporary files cotaining pieces of *Stata* code *and* execute it (`file write` and `do`)
- then graphs must be exported in suitable formats (`pdf`, `png`...) for the markup language to read them

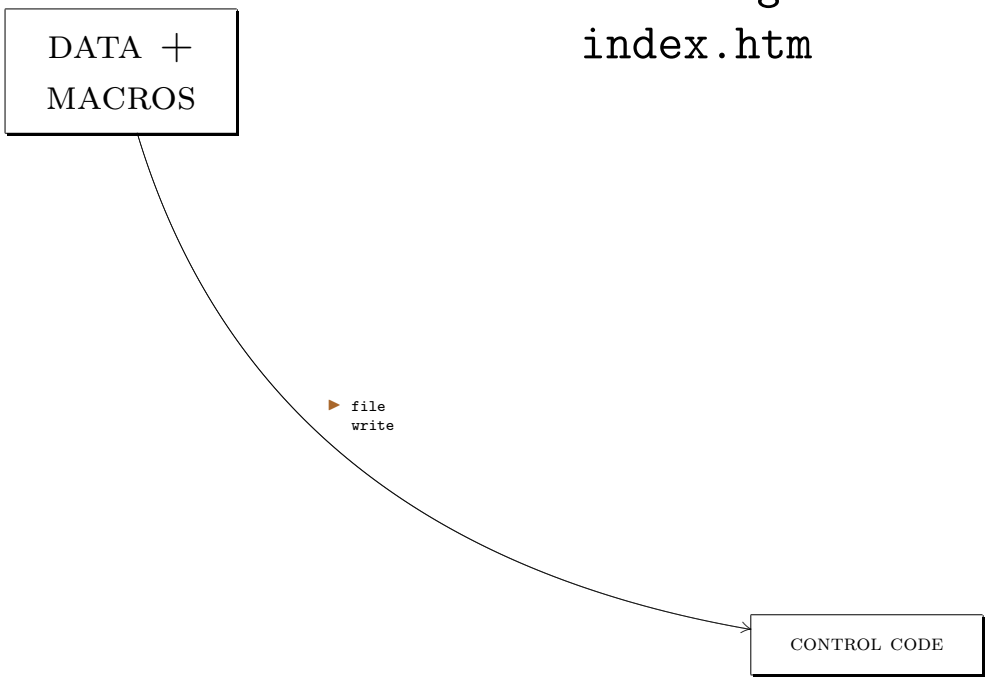
Hence this passage produces some image files in the formats that the particular markup language accepts, more conveniently stored in a separate directory (`capture mkdir`) named something like `figures`



- Many other elements accepted by the markup language can be generated
- tipically lists, trees,...
- but also archives of data (for php to read them)

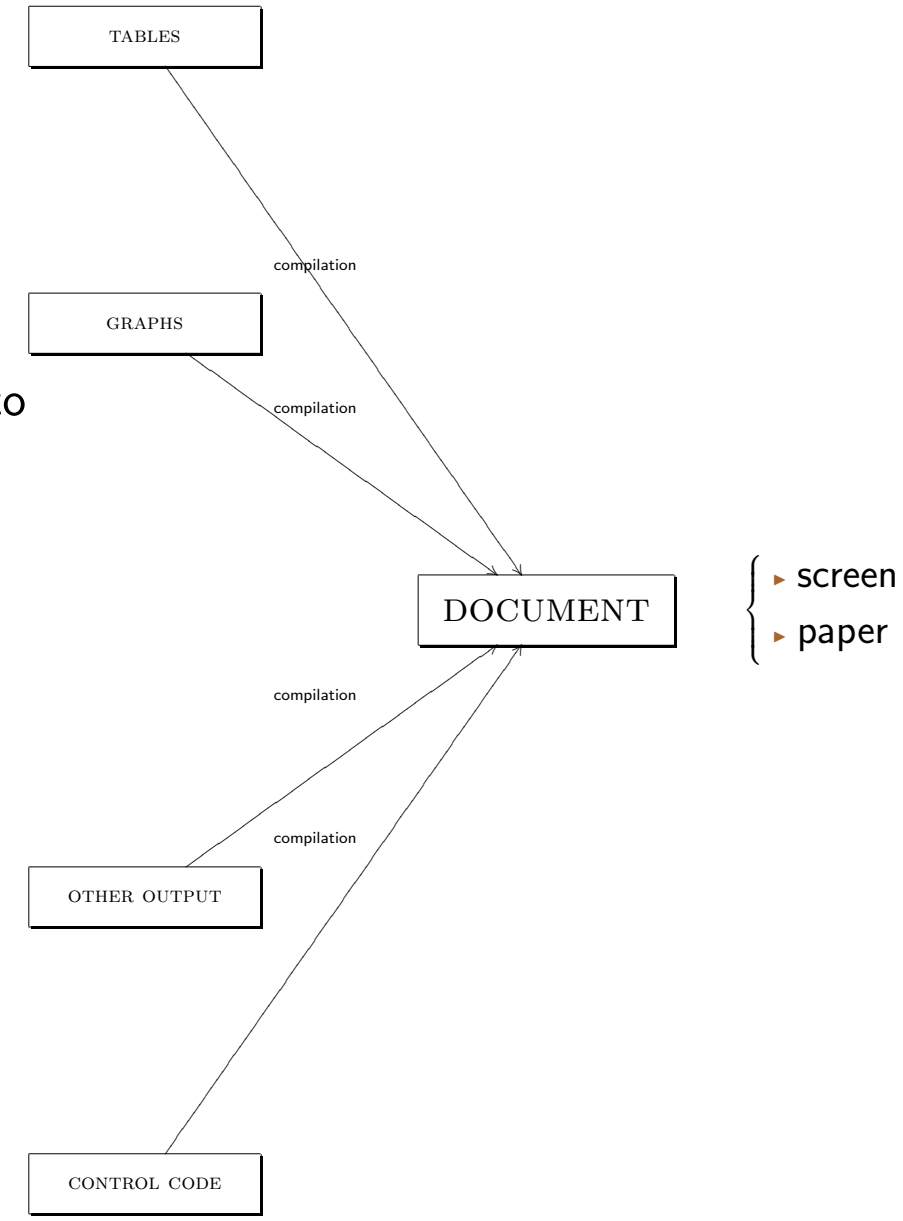
Hence this passage produces some files, that can be text files containg code of the markup language or image files, ..., all more conveniently stored in a separate directory (capture mkdir) named something like other

Data and macros must be used to produce the control code, which gathers together all the piecesHence this passage produces a single text file containg code of the markup language named something like `main.tex` or `index.htm`



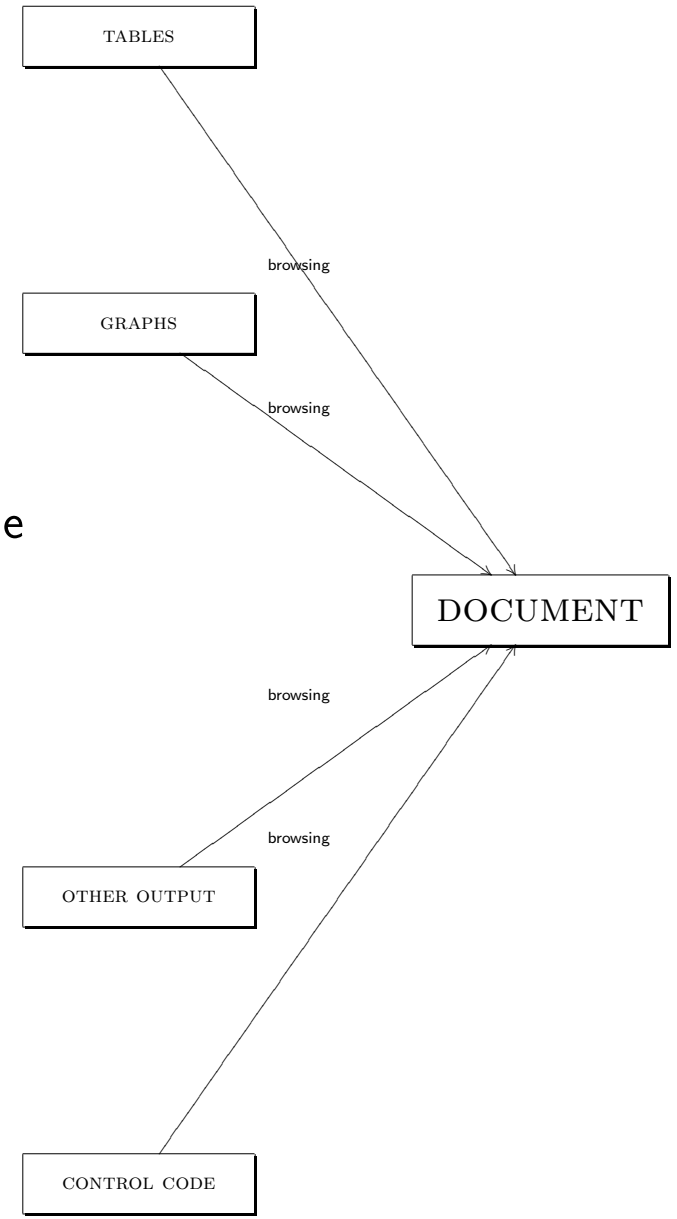
LaTeX

- The file `main.tex` must be compiled (`winexec`) to produce a single `pdf` file named `main.pdf`
- that then can be seen on the screen or printed



HTML

- There is no need of compiling
- the file `index.htm` can be browsed via any browser



- web

Can some parts of the process be standardised?

- This process is very flexible
- This way one can obtain virtually any kind of automatic document
- Is it possible to standardize some passages by writing ado files?

It would be very useful if the possibility of storing results of basic analysis on dta files became more systematic

Just like graph's schemes:

- Creating *Stata* "schemes" of LaTeX tables?
- Creating *Stata* "schemes" of HTML tables (css)?

Implement an ado file for trees?


Agenzia Regionale di Sanità della Toscana.
Isa 65+. Indicatori sulla salute e l'assistenza agli anziani.
www.arsanita.toscana.it, 2005.


Rosa Gini, Jacopo Pasquini.
2006. pr0020:
Automatic generation of documents.
The Stata Journal 6(1) 245-269.


Ghostscript, Ghostview and GSview.
http://www.cs.wisc.edu/~ghost/.


Leslie Lamport.
LaTeX – A Document Preparation System.
Addison-Wesley Publishing Co., 2nd edition, 1994.


Roger Newson.
2003. st0043:
Confidence intervals and p-values for delivery to the end user.