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 11, 2006
1 Introduction

2 Automated documents
   - Automated and actual documents
   - Motivation
   - Elements of an automatically generated document

3 Examples

4 Workflow for an automated document to complete its task
   - The general scheme
   - Generating dta files
   - Basic elements
   - Control code
   - Document production

5 Writing ado commands?
   - Analysis
   - Elements
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.
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.
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 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.
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
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 \textit{Stata} code) for \textit{Stata} to execute,
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 (possibly via compilation)
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 (possibly 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 worthwhile 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 easily generates an updated actual document when data change
When to write an automated document?

It is worthwhile 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 easily 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, ...
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
Traditional basic elements

- Tables
- Graphics
Other basic elements

- Lists
Other basic elements

- Lists
- Trees
Other basic elements

- Lists
- Trees
- ...

Introduction
Automated documents
Examples
Workflow for an automated document to complete its task
Writing ado commands?

Automated and actual documents
Motivation
Elements of an automatically generated document
Introduction
Automated documents

Examples
Workflow for an automated document to complete its task
Writing ado commands?
Introduction

Automated documents

Examples

Workflow for an automated document to complete its task

Writing ado commands?

The general scheme

Generating dta files

Basic elements

Control code

Document production

data + macros

dta for tables

dta for graphs

dta for other output

control code

DOCUMENT

screen

paper

web

DOCUMENT

DATA + MACROS

DTA FOR OTHER

DTA FOR GRAPHS

DTA FOR TABLES

OTHER OUTPUT

\[ \text{graph, \ twowaygraph, \ file write \ and \ do} \]

\[ \text{file write} \]

\[ \text{manual code} \]

\[ \text{reshape, \ listtex} \]

\[ \text{graphexport} \]

\[ \text{file write} \]

\[ \text{other programs (winexec)} \]

\[ \text{file write} \]
Introduction
Automated documents
Examples
Workflow for an automated document to complete its task
Writing ado commands?

The general scheme
Generating dta files
Basic elements
Control code
Document production

DATA + MACROS

DTA FOR TABLES

xcontract,
statby,
parmby,
manual code

DTA FOR GRAPHS

statby,
parmby,
manual code

TABLES

reshape,
list
file write

twoway graph,
file write

twoway graph,
file write and do

twoway graph,
file write

GRAPHS

graph,
twoway graph,
file write

DOCUMENT

▷ screen
▷ paper
▷ web

▷ compilation
▷ browsing

DATA + MACROS

DTA FOR OTHER

file write

OTHER OUTPUT

▷ file write

CONTROL CODE

▷ manual code

▷ other programs (windows)

▷ file write

▷ file write and do

▷ twoway graph

▷ manual code
Workflow for an automated document to complete its task

- Introduction
- Automated documents
- Examples

Writing ado commands?

The general scheme

Generating dta files

Basic elements

Control code

Document production

- Introduction
- Automated documents
- Examples

The general scheme

Generating dta files

Basic elements

Control code

Document production

**This is the actual data analysis**
this is the actual data analysis

the results of analysis are stored (generally) as dta files, more conveniently stored in a separate directory
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
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 \texttt{parmby} provides such a support altogether for all estimation commands.
this passage is granted by several commands by Roger Newson's, mainly listtex
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,...)
Introduction
Automated documents
Examples
Workflow for an automated document to complete its task
Writing ado commands?

The general scheme
Generating dta files
Basic elements
Control code
Document production

Introduction
Automated documents
Examples
Workflow for an automated document to complete its task
Writing ado commands?

The general scheme
Generating dta files
Basic elements
Control code
Document production

- 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* macros.
Remark 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 (`capturemkdir`) named something like `tables`
Workflow for an automated document to complete its task

**Examples**

**Introduction**

Automated documents

- Writing ado commands?

**Basic elements**

- Generating dta files

**The general scheme**

- Control code

**Document production**

- Data + macros

- DTA for graphs

- Graphs

- Graph, twoway graph, file write and do

- Graph export

- Stata’s graph commands often do analysis and graph composition at the same time

- Then graphs must be exported in suitable formats (pdf, png... for the markup language to read them)
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 containing pieces of Stata code and execute it (file write and do).
- 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 containing 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
Many other elements accepted by the markup language can be generated:
- typically lists, trees, ...
Hence this passage produces some files, that can be text files containing 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 pieces.
Hence this passage produces a single text file containing 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
\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
Introduction
Automated documents
Examples
Workflow for an automated document to complete its task
Writing ado commands?

The general scheme
Generating dta files
Basic elements
Control code
Document production

HTML

- There is no need of compiling
HTML

- There is no need of compiling
- The file `index.htm` can be browsed via any browser
Can some parts of the process be standardised?

- This process is very flexible
Can some parts of the process be standardised?

- This process is very flexible
- This way one can obtain virtually any kind of automatic document
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?
Introduction
Automated documents
Examples
Workflow for an automated document to complete its task
Writing ado commands?

Analysis
Elements

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.*

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.*

Roger Newson.
2003. st0043:
*Confidence intervals and p-values for delivery to the end user.*
*The Stata Journal* 6(1) 245-269.