

An Introduction to Stata Programming

Christopher F. Baum
Boston College

TECHNISCHE
INFORMATIONSBIBLIOTHEK
UNIVERSITÄTSBIBLIOTHEK
HANNOVER



A Stata Press Publication
StataCorp LP
College Station, Texas

Contents

List of tables	xv
List of figures	xvii
Preface	xix
Acknowledgments	xxi
Notation and typography	xxiii
1 Why should you become a Stata programmer?	1
Do-file programming	1
Ado-file programming	2
Mata programming for ado-files	2
1.1 Plan of the book	3
1.2 Installing the necessary software	3
2 Some elementary concepts and tools	5
2.1 Introduction	5
2.1.1 What you should learn from this chapter	5
2.2 Navigational and organizational issues	5
2.2.1 The current working directory and profile.do	6
2.2.2 Locating important directories: sysdir and adopath	6
2.2.3 Organization of do-files, ado-files, and data files	7
2.3 Editing Stata do- and ado-files	8
2.4 Data types	9
2.4.1 Storing data efficiently: The compress command	11
2.4.2 Date and time handling	11
2.4.3 Time-series operators	12
2.5 Handling errors: The capture command	14

2.6	Protecting the data in memory: The preserve and restore commands	14
2.7	Getting your data into Stata	15
2.7.1	Inputting data from ASCII text files and spreadsheets . . .	15
	Handling text files	16
	Free format versus fixed format	17
	The insheet command	18
	Accessing data stored in spreadsheets	20
	Fixed-format data files	20
2.7.2	Importing data from other package formats	25
2.8	Guidelines for Stata do-file programming style	26
2.8.1	Basic guidelines for do-file writers	27
2.8.2	Enhancing speed and efficiency	29
2.9	How to seek help for Stata programming	29
3	Do-file programming: Functions, macros, scalars, and matrices	33
3.1	Introduction	33
3.1.1	What you should learn from this chapter	33
3.2	Some general programming details	34
3.2.1	The varlist	35
3.2.2	The numlist	35
3.2.3	The if exp and in range qualifiers	35
3.2.4	Missing data handling	36
	Recoding missing values: The mvdecode and mvencode commands	37
3.2.5	String-to-numeric conversion and vice versa	37
	Numeric-to-string conversion	38
	Working with quoted strings	39
3.3	Functions for the generate command	40
3.3.1	Using if exp with indicator variables	42
3.3.2	The cond() function	44
3.3.3	Recoding discrete and continuous variables	45

3.4	Functions for the <code>egen</code> command	47
	Official <code>egen</code> functions	47
	<code>egen</code> functions from the user community	49
3.5	Computation for by-groups	50
3.5.1	Observation numbering: <code>_n</code> and <code>_N</code>	50
3.6	Local macros	53
3.7	Global macros	56
3.8	Extended macro functions and macro list functions	56
3.8.1	System parameters, settings, and constants: <code>creturn</code>	57
3.9	Scalars	58
3.10	Matrices	60
4	Cookbook: Do-file programming I	63
4.1	Tabulating a logical condition across a set of variables	63
4.2	Computing summary statistics over groups	65
4.3	Computing the extreme values of a sequence	66
4.4	Computing the length of spells	67
4.5	Summarizing group characteristics over observations	71
4.6	Using global macros to set up your environment	73
4.7	List manipulation with extended macro functions	74
4.8	Using <code>creturn</code> values to document your work	76
5	Do-file programming: Validation, results, and data management	79
5.1	Introduction	79
5.1.1	What you should learn from this chapter	79
5.2	Data validation: The <code>assert</code> , <code>count</code> , and <code>duplicates</code> commands	79
5.3	Reusing computed results: The <code>return</code> and <code>ereturn</code> commands	86
5.3.1	The <code>ereturn list</code> command	90
5.4	Storing, saving, and using estimated results	93
5.4.1	Generating publication-quality tables from stored estimates	98
5.5	Reorganizing datasets with the <code>reshape</code> command	99
5.6	Combining datasets	105

5.7	Combining datasets with the append command	107
5.8	Combining datasets with the merge command	108
5.8.1	The dangers of many-to-many merges	110
5.9	Other data-management commands	111
5.9.1	The fillin command	112
5.9.2	The cross command	112
5.9.3	The stack command	112
5.9.4	The separate command	114
5.9.5	The joinby command	115
5.9.6	The xpose command	115
6	Cookbook: Do-file programming II	117
6.1	Efficiently defining group characteristics and subsets	117
6.1.1	Using a complicated criterion to select a subset of observations	118
6.2	Applying reshape repeatedly	119
6.3	Handling time-series data effectively	123
6.4	reshape to perform rowwise computation	126
6.5	Adding computed statistics to presentation-quality tables	128
6.5.1	Presenting marginal effects rather than coefficients	130
6.6	Generating time-series data at a lower frequency	132
7	Do-file programming: Prefixes, loops, and lists	139
7.1	Introduction	139
7.1.1	What you should learn from this chapter	139
7.2	Prefix commands	139
7.2.1	The by prefix	140
7.2.2	The xi prefix	142
7.2.3	The statsby prefix	145
7.2.4	The rolling prefix	146
7.2.5	The simulate and permute prefix	148
7.2.6	The bootstrap and jackknife prefixes	151
7.2.7	Other prefix commands	153

7.3	The forvalues and foreach commands	154
8	Cookbook: Do-file programming III	161
8.1	Handling parallel lists	161
8.2	Calculating moving-window summary statistics	162
8.2.1	Producing summary statistics with rolling and merge	164
8.2.2	Calculating moving-window correlations	165
8.3	Computing monthly statistics from daily data	166
8.4	Requiring at least n observations per panel unit	167
8.5	Counting the number of distinct values per individual	169
9	Do-file programming: Other topics	171
9.1	Introduction	171
9.1.1	What you should learn from this chapter	171
9.2	Storing results in Stata matrices	171
9.3	The post and postfile commands	175
9.4	Output: The outsheet, outfile, and file commands	177
9.5	Automating estimation output	181
9.6	Automating graphics	184
9.7	Characteristics	188
10	Cookbook: Do-file programming IV	191
10.1	Computing firm-level correlations with multiple indices	191
10.2	Computing marginal effects for graphical presentation	194
10.3	Automating the production of L ^A T _E X tables	197
10.4	Tabulating downloads from the Statistical Software Components archive	202
10.5	Extracting data from graph files' sersets	204
10.6	Constructing continuous price and returns series	209
11	Ado-file programming	215
11.1	Introduction	215
11.1.1	What you should learn from this chapter	216
11.2	The structure of a Stata program	216

11.3	The program statement	217
11.4	The syntax and return statements	218
11.5	Implementing program options	221
11.6	Including a subset of observations	222
11.7	Generalizing the command to handle multiple variables	224
11.8	Making commands byable	226
	Program properties	228
11.9	Documenting your program	228
11.10	egen function programs	231
11.11	Writing an e-class program	232
	11.11.1 Defining subprograms	234
11.12	Certifying your program	234
11.13	Programs for ml, nl, nlsur, simulate, bootstrap, and jackknife	236
	Writing an ml-based command	237
	11.13.1 Programs for the nl and nlsur commands	240
	11.13.2 Programs for the simulate, bootstrap, and jackknife prefixes	242
11.14	Guidelines for Stata ado-file programming style	244
	11.14.1 Presentation	244
	11.14.2 Helpful Stata features	245
	11.14.3 Respect for datasets	246
	11.14.4 Speed and efficiency	246
	11.14.5 Reminders	247
	11.14.6 Style in the large	247
	11.14.7 Use the best tools	248
12	Cookbook: Ado-file programming	249
12.1	Retrieving results from rolling:	249
12.2	Generalization of egen function pct9010() to support all pairs of quantiles	252
12.3	Constructing a certification script	254

12.4	Using the <code>ml</code> command to estimate means and variances	259
12.4.1	Applying equality constraints in <code>ml</code> estimation	261
12.5	Applying inequality constraints in <code>ml</code> estimation	262
12.6	Generating a dataset containing the single longest spell	267
13	Mata functions for ado-file programming	271
13.1	Mata: First principles	271
13.1.1	What you should learn from this chapter	272
13.2	Mata fundamentals	272
13.2.1	Operators	272
13.2.2	Relational and logical operators	274
13.2.3	Subscripts	274
13.2.4	Populating matrix elements	275
13.2.5	Mata loop commands	276
13.2.6	Conditional statements	278
13.3	Function components	279
13.3.1	Arguments	279
13.3.2	Variables	280
13.3.3	Saved results	280
13.4	Calling Mata functions	281
13.5	Mata's <code>st_</code> interface functions	283
13.5.1	Data access	283
13.5.2	Access to locals, globals, scalars, and matrices	285
13.5.3	Access to Stata variables' attributes	286
13.6	Example: <code>st_</code> interface function usage	286
13.7	Example: Matrix operations	288
13.7.1	Extending the command	293
13.8	Creating arrays of temporary objects with pointers	295
13.9	Structures	299
13.10	Additional Mata features	302
13.10.1	Macros in Mata functions	302

13.10.2	Compiling Mata functions	303
13.10.3	Building and maintaining an object library	304
13.10.4	A useful collection of Mata routines	305
14	Cookbook: Mata function programming	307
14.1	Reversing the rows or columns of a Stata matrix	307
14.2	Shuffling the elements of a string variable	311
14.3	Firm-level correlations with multiple indices with Mata	312
14.4	Passing a function to a Mata function	316
14.5	Using subviews in Mata	319
14.6	Storing and retrieving country-level data with Mata structures . . .	321
14.7	Locating nearest neighbors with Mata	327
14.8	Computing the seemingly unrelated regression estimator	331
14.9	A GMM-CUE estimator using Mata's optimize() functions	337
	References	349
	Author index	353
	Subject index	355