Subject index

Symbols

`/* */` comments . . . 412
`// NotReached` . . . 358
`!!` (sic) . . . 357

A

abort with error . . . 43–44
    equivalent to C crashing . . . 416
accuracy . . . 115–120, 131, *also see* numerical accuracy
`.ado` file . . . 12
ado-files, writing Mata code for . . . 145–147, 373–399
    accessing Stata's data . . . 379–384, 393–394
    changing Stata's dataset . . . 396
    displaying results . . . 387–388
    executing Stata commands . . . 397–399
    handling errors . . . 384–386
    modifying Stata's data . . . 394
    obtaining all variable names . . . 396
    parsing . . . 375–378
    placement of Mata function . . 145–153, 373–374
    private Mata function . . . 145–153, 373–374
    returning results . . . 389–393
    Stata interface functions . . 393–399
    structuring of code . . . 375
    working with Stata macros, scalars, matrices . . . 396–397
    working with Stata's metadata . . . 394–396
    working with views . . . 393–394
approved source directory . . . 154–156, 218–219, 261
`args()` function . . . 125–127
argument . . . *see* function argument
arithmetic operators . . . 75–76
ASD . . . *see* approved source directory
`assert()` function . . . 136–137
`assert` Stata command . . . 150–152
assigment operator . . . 73–74
associative array . . . 281–284, 341–344, 348
    `firstval()` . . . 343–344, 348
    `key()` . . . 343–344, 348
    `keys()` . . . 341–344, 348
    `nextval()` . . . 343–344, 348

B

boolean type . . . *see* macroed type, boolean

C

C and C++, difference from Mata . . . 411–416
`C()` function . . . 72, 104–105, 408
certification file . . . 136–139, 141–144, 213–218, 260–261, 321, 360, 365–366
    comparison with validation . . . 144
    for ado . . . 150–152
class . . . 36–38, 112–113, 221–247, 249–261, 296–297, 312–320
    `::` prefix to call external function . . . 224–225, 233
    calling external function . . 224–225
    constructor function . . . 231–233
    creation . . . 231–233
    deletion . . . 231–233
    `destroy()` user-written function . . . 231–233

class, *continued*
  do not document (DND) member 309–311, 364
  freeing of memory 232–233
  function shadowing 224–225
  inheritance 227–231, 238–247
    final member 243–245
    function shadowing 229
    polymorphism 245–246
    virtual function 240–244
    when to use . . . 238–240, 246–247
  instance definition . . *see* structure, instance definition
  macroed type 273–276
  `new()` user-written function . . 231–233
  no member variable 236–238
  pointer 247
  private and public members . . 225–226, 234–235
  protected members 228–229
  `setup()` user-written function 230–231
  `super` prefix 230–231
  `this` prefix 233
`class` eltype 103–105
class `->` operator 94–96
`classname()` function 296–297
Code numeric type . . *see* macroed type, Code
code, self-threading . . *see* self-threading code
colon operators 82–83
`cols()` function 25
column-join operator 26–29, 80–82
`colvector` orgtype 103–105
comment, nested 412
`complex` eltype 103–105, 401–409
complex value . . 22, 71–72, 78, 401–409
compound quote character 72–73
`conj()` function 401–402, 405
constant 277–280
constructor function . . . . . . *see* class, *see* structure, *see* `J()` function

`cross()` function 202–204, 209–211
`crossdev()` function 209–211

D
decrement operator 76–77
derived type . . . . *see* macroed type
design document
  formula sheet 192–193, 207
  notes 308–309
  shortcomings 311–312
  summary 309–311
development process 285–321, 324–367
  idea 338–340
DND . . . . *see* class, do not document (DND) member
do-file
  containing Mata code 134–144
  for scientific research 314
double-bang comment 357

E
element types . . . . *see* variable type
eltype 103–105
`eltype()` function . . . 296–297, 408–409
`_error()` function 43–44
expr . . . . *see* expression
expression . . . . 47–49, 62–97, *see also* constant
  assignment 73–74
  E notation 64–65
  literal 64–71
    base 10 . . . . 64–65, 71
    base 2 . . . . 66–71, 217
    complex 71–72, 403–404
    missing value 65
    missing values with complex 404
    numeric 64–71, 217
    string 72–73
  operator precedence 74–75
  operators 18–21
    `(void)` cast 97
    `=` assignment 18–21, 73–74
    `-` negation 75–76

expression operators, *continued*
- + addition 75–76
- - subtraction 75–76
- * multiplication 75–76
- / division 75–76
- ^ power 75–76
- ++ increment 76–77
- -- decrement 76–77
- == equality 77–79, 167
- != inequality 77–79, 167
- < less than 77–79
- <= less than or equal to . . . 77–79
- > greater than 77–79
- >= greater than or equal to . . 77–79
- & logical and 77–79
- | logical or 77–79
- ! logical negation 77–79
- ? : conditional (ternary) . . . 79–80
- [] element-and-list subscripts 83–92
- [| |] range (submatrix) subscripts 83–84, 92–93
- ' matrix transpose . . . 75–76, 405
- \ row join 26–29, 80–81
- , column join 26–29, 80–82
- :+ elementwise addition . . 82–83
- :- elementwise subtraction . . 82–83
- :* elementwise multiplication 82–83
- :/ elementwise division . . . 82–83
- :^ elementwise power 82–83
- :== elementwise equality . . 82–83
- :!= elementwise inequality . . 82–83
- :< elementwise less than . . 82–83
- :<= elementwise less than or equal to 82–83
- :> elementwise greater than 82–83
- :>= elementwise greater than or equal to 82–83

expression operators, *continued*
- :& elementwise logical and . . 82–83
- :| elementwise logical or . . 82–83
- .. row count vector 80, 82
- :: column count vector . . 80, 82
- & address (pointer) of 38–40, 94–96, 172–175
- * pointer dereference 38–40, 94–96, 172–175
- -> structure dereference . . 94–96
- -> class dereference 94–96

F

factorial() function 130
Filehandle type . . . . . *see* macroed type, Filehandle
final member . . . . . *see* class, inheritance
formula sheet 192–193, 207, 254
Fortran 47, 56, 58
free() C and C++ function 415
function
- argument 81–82, 121–128, 412
  - difference in treatment between Mata C and C++ 412
  - output 110, 123–125
  - overloaded . . . 109–110, 127–128, 296–297, 315–318
  - passthru 111–112, 170–172
  - varying number 125–128
- *arguments* 46
- *body* 47
- break statement 57
- continue statement 57
- do statement 50
- do while statement 50, 56–57
- expression . . . . . *see* expression
- for statement 53–56
- function statement 100–101
- goto statement 58
- if statement 49–50
- looping construct . . . . . *see* function, do while statement, *see* function, do statement, *see* function, while statement

function, *continued*
- *program body* 46
- `return` statement 59–60
- *returnedtype* 46
- returning `void` 107–109
- structure of 46
- syntax 127–128
- virtual . . . . . *see* class, inheritance
- `while` statement 50–53

H

hello, world 9

I

`I()` function 25
`if` statement . . . . . *see* function, `if` statement
`Im()` function 401–402, 408
increment operator 76–77
inheritance . . . . . *see* class, inheritance
initialization of variables, default . . 105
instance, definition 165
integer value 130–133
interactive mode 40–42
`invorder()` function 88–91
`invsym()` function 40–42, 259
`issymmetric()` function 406

J

`J()` function 26

K

key . . . . . *see* associative array

L

`length()` function 25
library . . . . . *see* Mata library
linear-regression example 177–219, 249–261
list subscript 83–92
`lmbuild` command 14–16
`lnfactorial()` function 130
logical operators 77–79
`lr*()` function 187–211
`luinv()` function 401–402

M

macroed type 263–276
- boolean 268–269
- Code 269–271
- Filehandle 271
- for classes 273–276
- for structures 272–276
- idiosyncratic 272
- Ocode 271
- Ordinal 271

`malloc()` C and C++ function . . . 413–414
Mata
- comparison with C and C++ programming languages 2–3
- description 2
- difference from C and C++ . . 411–416
- library 14–16
  - building . . 153–159, 218–219, 261

`.mata` file . . . 10, 137–141, 153–154, 158
Mata Reference Manual 1
`matastrict` 115–120, 149, 167–168
`matrix` orgtype 103–105
matrix, view 85
`mean()` function 209–211
member function . . . . . *see* class
member variable . . . . . *see* class, *see* structure
missing value 65
`.mlib` file . . . . . 14–16, *also see* Mata library
- erasing 159

`mreldif()` function 258–259, 366

N

name conflict . . . 236–237, 273–276, 313
`n_choose_k()` function 130–133
- example 129–159
- packaged as ado-file 145–153
- packaged as do-file 134–144
- packaged as Mata library routine 153–159

not-reached comment 358
`NotReached` 358

null vector and matrix 29–31
`numeric` eltype 103–105, 406–408
numerical accuracy 66–71, 130–133, 205–206, 211–212, 256–259, 365–366
  matrix balancing 259

O
operators . . . . . *see* expression operators
`order()` function 88–91
Ordinal numeric code type . . . . *see* macroed type, Ordinal
organizational types . . *see* variable type
orgtype 103–105
overloaded function . . . . *see* function argument, overloaded

P
passthru variable 111–112, 170–172
permutation matrix 91–92
permutation vector 88–92
`pointer` eltype 103–105
pointers 38–40, 112–113, 247, 344–348, 417
  advanced use 344–348, 417
  arithmetic is not allowed 416
  conserving memory 174–175, 186–187
  declaration 112–113
  differences from C and C++ . . 413–416
  element of vector or matrix . . . 415
  to create 3-dimensional arrays 417
  to function 94–96
  value 94–96, 172–175
polymorphism . . . . *see* class, inheritance
pragma 115–120
program . . . . *see* function
programmer, serious 1–2
programming, semicolon line end character 411–412
project to-do list . . . . . . *see* double-bang comment

Q
`quadcross()` function 205, 209–211
`quadcrossdev()` function 209–211
quote character 72–73

R
range subscript 83–84, 92–93
`rcof` Stata command 150–152
`Re()` function 401–402, 408
`real` eltype 103–105, 406–408
real value 22, 64–71
  closest to zero 22
  missing values 22
  range 22
  range of precise integer . . . 131, 133
  range of precise integers 22
`round()` function 130–131
row-join operator 26–29, 80–81
`rows()` function 25
`rowvector` orgtype 103–105

S
`scalar` orgtype 103–105
self-threading code . . . 181–187, 252–253
semicolon, use to indicate end of line 411–412
`set matastrict` . . . . . . *see* `matastrict`
`set rmsg` 325–330, 360
`sort()` function 88–91
sorting data matrices 88–91
source directory . . . *see* approved source directory
sparse-matrix example 285–321, 324–367
  use with views 382–383
Stata . . *see* ado-files, writing Mata code for
storage type . . . . *see* variable type
strict setting . . . . *see* `matastrict`
`string` eltype 103–105
string value 22, 24, 72–73, 78–79
string, differences from C and C++ 412–413
`struct` . . . . *see* structure
`struct` eltype 103–105

structure 34–36, 112–113, 161–175, 177–219
  * operator 172–175
  & operator 172–175
  accessing member 162
  adding member variable 166
  assignment 167
  common error 167–168
  constructor function 168–170
  containing other structure 166
  defining 164
  instance definition 165
  macroed type 272–276
  pointer 172–175
  returning 163–164
  scalar 167–168
  subscripting 170
  testing equality 167
  use of transmorphic 170–172
  vector and matrix 168–170
structure `->` operator 94–96
submatrix subscript 83–84, 92–93
subscript 83–93
`switch` and `case` C functions 416
symmetric matrix 256–259
  how to code 258–259

T
ternary conditional operator 79–80
`test_*.do` file 137–139, 141–144, 158–159
`timer()` function 330–338
timing code
  detailed 330–338
  overall 325–330, 360
to-do list for project . . *see* double-bang comment
tolerance 66–71, 217
traceback log 42–44, 106
transmorphic 109–112, 170–172
  use with classes 296–297
`transmorphic` eltype 103–105
`_transpose()` function 300–301
`trunc()` function 133
type . . . . . *see* variable type

U
uninitialized
  value table 105
  variable 105

V
validation 144
variable
  initialization, default 105
  instance, definition 165
  passthru 111–112, 170–172
  type . . 32–34, 99–113, 401–409, *see also* eltype, *see also* macroed type, *see also* orgtype
    complex 401–409
    omitted 100–101
    partial 106–107
    real 22
    string 22–24
`vector` orgtype 103–105
vector, permutation . . . *see* permutation vector
`version` statement and number . . 11–13
view matrices . . . . . *see* matrix, view
virtual function . . *see* class, inheritance
`void` and `void` casting 97, 107–109

X
X notation 66–71, 217