Subject index

A

algorithms
- BFGS 19–20, 71
- BHHH 19, 71–75
- DFP 19–20, 71
- Newton's method 14–16
- Newton–Raphson 16–18, 71–75

ancillary parameters 241–243

arguments
- `b` 97–101, 131
- `g` 102–103, 136–138, 152–156, 271–272
- `H` 103–106, 138–139, 156–164, 272–273
- `lnf` 80, 132–134, 148–152, 270–271
- `lnfj` 81, 101–102
- `todo` 97, 131

`aweights` . . . . . *see* weights, `aweights`

B

`_b` 60

`b` argument . . . . . *see* arguments, `b`

backed-up message . . . . . *see* messages, backed-up

BFGS algorithm . . *see* algorithms, BFGS

BHHH algorithm . . . . . *see* algorithms, BHHH

bivariate Poisson regression . . . . . *see* models, bivariate Poisson regression

Break key 202–204

`by` prefix command 218

byable 218

C

censoring 124, 316

coefficient vector argument . . . . . *see* arguments, `b`

concentrated log likelihood 330

`constraints()` option . . . . *see* options, `constraints()`

`continue` option . . . . . *see* options, `continue`

convergence 26–28, 207–208

Cox regression . . . . . *see* models, Cox regression

D

`d0` . . . . . *see* methods, `d0`

`d1` . . . . . *see* methods, `d1`

`d1debug` . . . . . *see* methods, `d1debug`

`d2` . . . . . *see* methods, `d2`

`d2debug` . . . . . *see* methods, `d2debug`

derivatives . . . *see* numerical derivatives

DFP algorithm . . . . *see* algorithms, DFP

`difficult` option . . . . . *see* options, `difficult`

`_diparm` 243

`diparm()` option . . . . . *see* options, `diparm()`

do-files 213–215

donuts 354–356

E

`e(cmd)` 241

`e(k_aux)` 241

`e(k_eq)` 241

`e(predict)` 254

empirical variance estimator . . . . . *see* variance estimators, robust

equation names 58, 60, 193–195

equation notation 54–62

estimation commands 217–249, 309–357
evaluators . . . . . *see* methods

F
`fweights` . . . . . *see* weights, `fweights`

G
`g` argument . . . . . *see* arguments, `g`
`_get_diopts` 222–223
`gf` . . . . . *see* methods, `gf`
gradient vector 6, 267–268
gradient vector argument . . . . . *see* arguments, `g`

H
`H` argument . . . . . *see* arguments, `H`
Hessian argument . . . . *see* arguments, `H`
Hessian matrix 6, 268–269
heteroskedasticity 161

I
initial values 31–33, 191–199, 238–240, 290–291, 296–299
`iweights` . . . . . *see* weights, `iweights`

L
`lf` . . . . . *see* methods, `lf`
`lf0` . . . . . *see* methods, `lf0`
`lf1` . . . . . *see* methods, `lf1`
`lf1debug` . . . . . *see* methods, `lf1debug`
`lf2` . . . . . *see* methods, `lf2`
`lf2debug` . . . . . *see* methods, `lf2debug`
likelihood evaluators . . . . . *see* methods
likelihood evaluators, Mata-based 257–278
likelihood theory 5–13
likelihood-ratio test 10–11
linear regression . . . . . *see* models, linear regression
linear-form restrictions 54, 80
linearization method . . . . . *see* variance estimators, robust
`lnf` argument . . . . . *see* arguments, `lnf`
`lnfj` argument . . . *see* arguments, `lnfj`
logit model . . . . . *see* models, logit

M
manipulable 11
`markout` 226–228
`marksample` 226
Mata 257–307
maximization . . . . . *see* numerical, maximization
messages
  backed-up 21
  not concave 20–21, 87–88
methods 62–65
  `d0` 129–142, 264–265, 420–421
  `d1` 129–138, 142–144, 264–265, 421–423
  `d1debug` 173–185
  `d2` 129–138, 144–146, 264–265, 423–426
  `d2debug` 173–174, 185–188
  `gf` 63
  `lf` . . 79–93, 257–259, 263, 280–283, 419–420
  `lf0` 110–111, 263–264, 426–427
  `lf1` 111–113, 263–264, 427–429
  `lf1debug` 173–185
  `lf2` . . . 113–114, 260–264, 283–286, 429–431
  `lf2debug` 173–174, 185–188
`missing` option . . *see* options, `missing`
`ml check` 171–173, 366
`ml clear` 366
`ml count` 366
`ml display` 209–211, 218, 241–243, 366
`ml footnote` 366
`ml graph` 207–208, 366, 387
`ml init` 197–199, 333–335, 366
`ml maximize` 201–205, 366, 386
`ml model` 55–78, 365
`ml plot` 195–197, 366
`ml query` 193–195, 203, 366
`ml report` 204, 366
`ml score` 254, 366, 389
`ml search` 192–195, 366

`ml trace` 188–189, 366
`mleval` 99–101, 131, 371, 379, 388
`mlexp` 29–42
`mlexp` command
  syntax 359–363
`mlmatbysum` 156–158, 161–164, 268–269, 371, 379, 389
`mlmatsum` . . 105–106, 138–139, 371, 379, 389
`mlopts` 230–233
`ML_samp` 89, 420, 422, 424, 426, 428, 430
`mlsum` 134–135, 371, 379, 389
`mlvecsum` 136–138, 371, 379, 389
`ML_w` . . . 89, 132, 150, 420, 422, 425, 426, 428, 430
`ML_y1` . . 79, 419, 420, 422, 424, 426, 428, 430
models
  bivariate Poisson regression . . 338–357, 455–462
  Cox regression . . 164–170, 319–321, 442–444
  linear regression . . 83–85, 117–123, 218, 302, 313–316, 437–439
  logit 309–311, 433–435
  normal . . . . . *see* models, linear regression
  panel-data 146–164, 269–275
  probit 36–37, 81–82, 101–102, 106, 115–117, 135, 138–139, 257–259, 280–283, 311–313, 435–437
  random-effects regression 146–164, 269–275, 322–325, 444–447
  seemingly unrelated regression 325–337, 448–454
  Weibull . . 85–86, 124–127, 260–262, 283–286, 316–319, 440–442
`moptimize` 279–307
`moptimize()` function
  syntax 391–417
`moptimize_init_by` 268–269
`moptimize_util_depvar` 258, 266
`moptimize_util_eq_indices` . . 266–267
`moptimize_util_matbysum` 268–269
`moptimize_util_matsum` 261, 268
`moptimize_util_sum` 267
`moptimize_util_vecsum` 267–268
`moptimize_util_xb` 258, 266
`mybipoisson` 455–462
`mycox` 320, 442–444
`mylogit` 310–311, 433–435
`mynormal` 315–316, 437–439
`myprobit` 312–313, 435–437
`myrereg` 324–325, 444–447
`mysureg` 335, 448–454
`myweibull` 318, 440–442

N

Newton–Raphson algorithm . . . . . *see* algorithms, Newton–Raphson
Newton's method . . . . . *see* algorithms, Newton's method
`noconstant` option . . . . . *see* options, `noconstant`
nonconcavity 16–18
noninteractive mode 221–222, 365, 379–384
nonlinear specifications 90–91
not concave message . . . . . *see* messages, not concave
numerical
  derivatives 21–26
  maximization 14–26
  root finding 14–18

O

OPG . . . . . *see* variance estimators, outer product of gradients
options
  `constraints()` 71, 368, 380
  `continue` 234, 382
  `difficult` 18, 204–205
  `diparm()` 241–243
  `eform()` 243–245, 370, 388
  `exposure()` 245–247
  `missing` 226–227
  `noconstant` 58, 384

options, *continued*
`offset()` 245–247
`subpop()` 71
`svy` 70–71, 251–256, 290, 383
`technique()` 71–75, 368, 383
`vce()` . . 35–36, 48, 68–70, 292–293, 380
outer product . . *see* variance estimators, outer product of gradients

P
panel-data models . . . . . *see* models, panel-data
Poisson regression . . . . . *see* models, bivariate Poisson regression
`predict` 252
probit model . . . . . *see* models, probit
program properties . . 247–249, 251–254
`svyb` 252
`svyj` 252
`svyr` 252
`swml` 247–248
`properties()` option . . . . . *see* program properties
proportional hazards model . . . . . *see* models, Cox regression
`pweights` . . . . . *see* weights, `pweights`

Q
qualifiers, subsample 66, 286–288

R
random-effects regression . . *see* models, random-effects regression
restricted parameters 33–34, 84
`robust` option . . . . . *see* options, `vce()`
root finding . . . . . *see* numerical root finding

S
sandwich estimator . . . . . *see* variance estimators, robust
scalars 107–109
scores . . *see* variance estimators, robust
seemingly unrelated regression . . . . . *see* models, seemingly unrelated regression
speed 91–93
starting values . . . . . *see* initial values
`subpop()` option . . . . . *see* options, `subpop()`
subsample 226–230
survey data . . 70–71, 253, 256, 290, 313, 315
`svy` option . . . . . *see* options, `svy`
`svy` prefix 251–256
`svyb` . . . . . *see* program properties, `svyb`
`svyj` . . . . . *see* program properties, `svyj`
`svyr` . . . . . *see* program properties, `svyr`
`svyset` 70–71, 290
`swml` . . . . . *see* program properties, `swml`

T
Taylor-series linearization method . . *see* variance estimators, robust
`technique()` option . . . . . *see* options, `technique()`
`todo` argument . . . *see* arguments, `todo`

V
variance estimators,
outer product of gradients . . 11–12, 68, 136–138
robust 12–13, 35–36, 48, 69–70, 136–138, 292–293, 380
`vce()` option . . . . . *see* options, `vce()`

W
Wald test 10–11
Weibull model . . . . . *see* models, Weibull
weights 49, 67, 289
`aweights` 49, 67, 142, 289, 421, 427
`fweights` 49, 67, 142, 289, 421, 427
`iweights` 49, 67, 142, 252, 289, 421, 427
`pweights` 49, 67, 70, 233–234, 289–290