## stata

# Maximum likelihood estimation

	🔄 regress - Linear regression — 🗌 🗙
<ul> <li>Over a thousand built-in estimators</li> </ul>	Model by/if/in Weights SE/Robust Reporting
<ul> <li>Intuitive and consistent syntax</li> </ul>	Dependent variable: Independent variables: y v x1 x2 v
<ul> <li>Survey data support</li> </ul>	Treatment of constant
<ul> <li>Program your own estimators</li> </ul>	Suppress constant term Has user-supplied constant
<ul> <li>Numerical or analytical derivatives</li> </ul>	Total SS with constant (advanced)
<ul> <li>Standard errors: OIM, OPG, Robust–Huber/White/sandwich, cluster–robust, bootstrap, jackknife, and more</li> </ul>	
<ul> <li>Powerful postestimation features</li> </ul>	
<ul> <li>Use point and click or type commands</li> </ul>	
	? C C K Cancel Submit

## Stata offers over a thousand built-in ML estimators

All follow elegant and intuitive syntax and have consistent output. Learn one command, know how to use them all.

Linear regression

```
. regress y x1 x2
```

Logistic regression

. logistic y x1 x2

Poisson regression

. poisson y x1 x2

Poisson regression with identity link (GLM) . glm y x1 x2, family(poisson) link(identity)

#### ARIMA/ARMAX

. arima y x1 x2, arima(2,1,3)

Logistic regression with survey data

. svy: logistic y x1 x2

## After estimation, easily access powerful postestimation features.

Postestimation Selector	-	
Postestimation commands:		
Harginal effects, estimated means, interactions,		Launch
Fests, contrasts, and comparisons of parameter estimates		
- Linear tests of parameter estimates		
Nonlinear tests of parameter estimates		
Contrasts		
Contrasts of margins		
- Pairwise comparisons		
-Pairwise comparisons of margins		
- Linear expressions of parameter estimates		
-Nonlinear expressions of parameter estimates		
Likelihood-ratio test comparing models		
Seemingly unrelated regression by combining models		
Specification, diagnostic, and goodness-of-fit analysis		
Predictions		
Other reports		
He Manage estimation results		
		Cancel

## Write your own ML estimators

Stata offers a powerful environment for you to add your own ML estimators. For log likelihoods that can be written as simple expressions, just type the expression in the **mlexp** command. For more complicated expressions, you can write a program in Stata's scripting or matrix language and use the **ml** suite to do the rest for you. You can even turn your ML evaluator into a command.

#### Type a simple expression

Use **mlexp** when your log likelihood can be expressed simply. For example, for normal linear regression, type

							U	>
view ml1.smcl	×							
F					Dialog 🔻	Also	see 💌	Jump
mlexp (ln(n	ormalden(y, {	o1}*x + {b0]	, {sigma	<b>})))</b>				
aximum likel:	ihood estimat:	Lon						
aximum likel:	ihood estimat:	ion						
aximum likel: og likelihood	ihood estimat: d = -195.3886	ion			Numbe	r o	f obs :	= 74
aximum likel: og likelihoo	ihood estimat: d = -195.3886 Coefficient	ion 9 Std. err.	z	P> z	Numbe	r o <sup>.</sup> nf.	f obs : inter	= <b>74</b>
aximum likel: og likelihood /b1	ihood estimat: d = -195.38869 Coefficient 6008687	ion Std. err. .0510832	z -11.76	P> z  0.000	Numbe [95% co - <b>.70098</b> 9	r o <sup>.</sup> nf. 9	f obs : inter 500	= 74 val]
laximum likel: og likelihood /b1 /b0	ihood estimat: d = -195.38869 Coefficient 6008687 39.44028	Std. err. .0510832 1.592043	z -11.76 24.77	P> z  0.000 0.000	Numbe [95% co 700989 36.3199	r o <sup>.</sup> nf. 9	f obs : inter 500	= 74 val] 7475 6063

#### Type a simple expression

With another small program, you can turn your likelihood-evaluation program into a full-fledged Stata command.

📔 C	o-file Editor - mynormal –	-		×
my	normal ×			-
1 2 3 4 5 6 7 8 9 10	<pre>program mynormal version 19.5 // parse command syntax syntax varlist [, *] gettoken depvar xvars : varlist // maximize the likelihood ml model 1f mynormal_1f (`depvar' = `xvars') (sigma:), ///</pre>			
	Line: 12, Col: 1	CAP	NUM	OVR
٥ ۷	iewer - view ml3.smcl —	-		Х
viev	v ml3.smcl ×			
1.1	Distance Al		- 1.	

Τ.						Dialog + Also	Jump to
. mynorm	al y x	C					
Log like	lihood	i = -195.38869	,			Number of ob Wald chi2(1) Prob > chi2	os = 74 = 138.36 = 0.0000
	У	Coefficient	Std. err.	z	P> z	[95% conf.	interval]
eq1							
	x	6008687	.0510832	-11.76	0.000	7009899	5007475
-	cons	39.44028	1.592043	24.77	0.000	36.31994	42.56063
sigma							
	cons	3.392099	.2788288	12.17	0.000	2.845605	3.938594
						(	AD NUM INS

#### Write a program

Write a program to evaluate more complicated likelihood functions.

📔 Do-file Editor - mynormal_lf	– 🗆 X
mynormal_lf ×	~
<pre>1 program mynormal_lf 2 version 19.5 3 args lnfj xb sigma 4 quietly replace 'lnfj' = lnnormalden(\$ML_y1, ' 5 end</pre>	`xb', `sigma')
	Line: 7, Col: 1 CAP NUM OVR
Viewer - view ml2.smcl	- D X
view ml2.smcl ×	•
+	Dialog • Also see • Jump to •
<pre>. ml model 1f mynormal_lf (y = x) (sigma:), maximize . ml display Log likelihood = -195.38869</pre>	Number of obs = 74 Wald chi2(1) = 138.36 Prob > chi2 = 0.0000
y Coefficient Std. err. z P> z	[95% conf. interval]
eq1 x6008687 .0510832 -11.76 0.000 _cons 39.44028 1.592043 24.77 0.000	70098995007475 36.31994 42.56063
sigma _cons 3.392099 .2788288 12.17 0.000	2.845605 3.938594
	CAP NUM INS

Your new command automatically has many nice features such as options for robust and cluster–robust standard errors without any extra programming effort.

- . mynormal y x, vce(robust)
- . mynormal y x, vce(cluster id)

With a few more lines of code, your command can even support survey data,

. svy: mynormal y x

Your command will also automatically work with postestimation features such as Wald tests, likelihood-ratio tests, contrasts, and much more.