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**
```plaintext
. regress y x1 x2
```

**Logistic regression**
```plaintext
. logistic y x1 x2
```

**Poisson regression**
```plaintext
. poisson y x1 x2
```

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

**ARIMA/ARMAX**
```plaintext
. arima y x1 x2, arima(2,1,3)
```

**Logistic regression with survey data**
```plaintext
. svy: logistic y x1 x2
```

After estimation, easily access powerful postestimation features
```
. postest
```

• Over a thousand built-in estimators
• Intuitive and consistent syntax
• Survey data support
• Program your own estimators
• Numerical or analytical derivatives
• Standard errors: OIM, OPG, Robust–Huber/White/sandwich, cluster–robust, bootstrap, jackknife, and more
• Powerful postestimation features
• Use point-and-click or type commands
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 languages 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,

```
.mlexp (lnormalden(y, (b1 + x0), (b0)))
```

This results in:

| Coef. | Std. Err. | z  | P>|z| | [95% Conf. Interval] |
|-------|-----------|----|------|----------------------|
| b1    | -0.6008687 | 0.501032 | -1.17 | 0.242 | -0.7009999 -0.5007475 |
| b0    | 39.44028    | 1.592043 | 24.77 | 0.000 | 36.31994 42.56063  |
| sigma | 3.3921      | 0.278828 | 12.17 | 0.000 | 2.045600 3.938594 |

**Write a program**

Write a program to evaluate more complicated likelihood functions.

```
1. program mynormal
2. version 14
3. syntax varlist (; )
4. quietly replace `lnf' = ln(normalden($ML_y1', $`x', $sigma'))
5. end
```

Use `mynormal` when your log likelihood can be expressed more complicately:

```
.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.

**Create your own command**

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

```
1. program mynormal
2. version 14
3. // parse command syntax
4. syntax varlist (; )
5. gettoken depvar xvars : varlist
6. // maximize the likelihood
7. ml model if mynormal_if (depvar = xvars) (sigma)
8. ml display
9. end
```

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)
```

Learn more about MLE and other Stata features at [stata.com/features](http://stata.com/features).