

-
-

- [Home](#)
- [About](#)
-



[Home](#) > [Statistics](#) > Use poisson rather than regress; tell a friend

Use poisson rather than regress; tell a friend

22 August 2011 [William Gould, President Emeritus](#) [22 Comments](#)

Share 14

Do you ever fit regressions of the form

$$\ln(y_j) = b_0 + b_1x_{1j} + b_2x_{2j} + \dots + b_kx_{kj} + \varepsilon_j$$

by typing

```
. generate lny = ln(y)
```

```
. regress lny x1 x2 ... xk
```

The above is just an ordinary linear regression except that $\ln(y)$ appears on the left-hand side in place of y .

The next time you need to fit such a model, rather than fitting a regression on $\ln(y)$, consider typing

```
. poisson y x1 x2 ... xk, vce(robust)
```

which is to say, fit instead a model of the form

$$y_j = \exp(b_0 + b_1x_{1j} + b_2x_{2j} + \dots + b_kx_{kj} + \varepsilon_j)$$

Wait, you are probably thinking. Poisson regression assumes the variance is equal to the mean,

$$E(y_j) = \text{Var}(y_j) = \exp(b_0 + b_1x_{1j} + b_2x_{2j} + \dots + b_kx_{kj})$$

whereas linear regression merely assumes $E(\ln(y_j)) = b_0 + b_1x_{1j} + b_2x_{2j} + \dots + b_kx_{kj}$ and places no constraint on the variance. Actually regression does assume the variance is constant but since we are working the logs, that amounts to assuming that $\text{Var}(y_j)$ is proportional to y_j , which is reasonable in many cases and can be relaxed if you specify **vce(robust)**.

In any case, in a Poisson process, the mean is equal to the variance. If your goal is to fit something like a Mincer earnings model,

$$\ln(\text{income}_j) = b_0 + b_1 \cdot \text{education}_j + b_2 \cdot \text{experience}_j + b_3 \cdot \text{experience}_j^2 + \varepsilon_j$$

there is simply no reason to think that the the variance of the log of income is equal to its mean. If a person has an expected income of \$45,000, there is no reason to think that the variance around that mean is 45,000, which is to say, the standard deviation is \$212.13. Indeed, it would be absurd to think one could predict income so accurately based solely on years of schooling and job experience.

Nonetheless, I suggest you fit this model using Poisson regression rather than linear regression. It turns out that the estimated coefficients of the maximum-likelihood Poisson estimator in no way depend on the assumption that $E(y_j) = \text{Var}(y_j)$, so even if the assumption is violated, the estimates of the coefficients b_0, b_1, \dots, b_k are unaffected. In the maximum-likelihood estimator for Poisson, what does depend on the assumption that $E(y_j) = \text{Var}(y_j)$ are the estimated standard errors of the coefficients b_0, b_1, \dots, b_k . If the $E(y_j) = \text{Var}(y_j)$ assumption is violated, the reported standard errors are useless. I did not suggest, however, that you type

```
. poisson y x1 x2 ... xk
```

I suggested that you type

```
. poisson y x1 x2 ... xk, vce(robust)
```

That is, I suggested that you specify that the variance-covariance matrix of the estimates (of which the standard errors are the square root of the diagonal) be estimated using the Huber/White/Sandwich linearized estimator. That estimator of the variance-covariance matrix does not assume $E(y_j) = \text{Var}(y_j)$, nor does it even require that $\text{Var}(y_j)$ be constant across j . Thus, Poisson regression with the Huber/White/Sandwich linearized estimator of variance is a permissible alternative to log linear regression — which I am about to show you — and then I'm going to tell you why it's better.

I have created simulated data in which

$$y_j = \exp(8.5172 + 0.06 \cdot \text{educ}_j + 0.1 \cdot \text{exp}_j - 0.002 \cdot \text{exp}_j^2 + \varepsilon_j)$$

where ε_j is distributed normal with mean 0 and variance 1.083 (standard deviation 1.041). Here's the result of estimation using **regress**:

```
. regress lny educ exp exp2
```

Source	SS	df	MS	Number of obs = 5000		
Model	141.437342	3	47.1457806	F(3, 4996) = 44.72		
Residual	5267.33405	4996	1.05431026	Prob > F = 0.0000		
Total	5408.77139	4999	1.08197067	R-squared = 0.0261		
				Adj R-squared = 0.0256		
				Root MSE = 1.0268		

lny	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
educ	.0716126	.0099511	7.20	0.000	.052104	.0911212
exp	.1091811	.0129334	8.44	0.000	.0838261	.1345362
exp2	-.0022044	.0002893	-7.62	0.000	-.0027716	-.0016373
_cons	8.272475	.1855614	44.58	0.000	7.908693	8.636257

I intentionally created these data to produce a low R-squared.

We obtained the following results:

	truth	est.	S.E.	
educ	0.0600	0.0716	0.0100	
exp	0.1000	0.1092	0.0129	
exp2	-0.0020	-0.0022	0.0003	
_cons	8.5172	8.2725	0.1856	<- unadjusted (1)
	9.0587	8.7959	?	<- adjusted (2)

(1) To be used for predicting $E(\ln(y_j))$

(2) To be used for predicting $E(y_j)$

Note that the estimated coefficients are quite close to the true values. Ordinarily, we would not know the true values, except I created this artificial dataset and those are the values I used.

For the intercept, I list two values, so I need to explain. We estimated a linear regression of the form,

$$\ln(y_j) = b_0 + \mathbf{X}_j \mathbf{b} + \varepsilon_j$$

As with all linear regressions,

$$\begin{aligned} E(\ln(y_j)) &= E(b_0 + \mathbf{X}_j \mathbf{b} + \varepsilon_j) \\ &= b_0 + \mathbf{X}_j \mathbf{b} + E(\varepsilon_j) \\ &= b_0 + \mathbf{X}_j \mathbf{b} \end{aligned}$$

We, however, have no real interest in $E(\ln(y_j))$. We fit this log regression as a way of obtaining estimates of our real model, namely

$$y_j = \exp(b_0 + \mathbf{X}_j \mathbf{b} + \varepsilon_j)$$

So rather than taking the expectation of $\ln(y_j)$, let's take the expectation of y_j :

$$\begin{aligned} E(y_j) &= E(\exp(b_0 + \mathbf{X}_j \mathbf{b} + \varepsilon_j)) \\ &= E(\exp(b_0 + \mathbf{X}_j \mathbf{b}) * \exp(\varepsilon_j)) \\ &= \exp(b_0 + \mathbf{X}_j \mathbf{b}) * E(\exp(\varepsilon_j)) \end{aligned}$$

$E(\exp(\varepsilon_j))$ is not one. $E(\exp(\varepsilon_j))$ for ε_j distributed $N(0, \sigma^2)$ is $\exp(\sigma^2/2)$. We thus obtain

$$E(y_j) = \exp(b_0 + \mathbf{X}_j\mathbf{b}) * \exp(\sigma^2/2)$$

People who fit log regressions know about this — or should — and know that to obtain predicted y_j values, they must

1. Obtain predicted values for $\ln(y_j) = b_0 + \mathbf{X}_j\mathbf{b}$.
2. Exponentiate the predicted log values.
3. Multiply those exponentiated values by $\exp(\sigma^2/2)$, where σ^2 is the square of the root-mean-square-error (RMSE) of the regression.

They do in this in Stata by typing

```
. predict yhat
. replace yhat = exp(yhat).
. replace yhat = yhat*exp(e(rmse)^2/2)
```

In the table I that just showed you,

	truth	est.	S.E.	
educ	0.0600	0.0716	0.0100	
exp	0.1000	0.1092	0.0129	
exp2	-0.0020	-0.0022	0.0003	
_cons	8.5172	8.2725	0.1856	<- unadjusted (1)
	9.0587	8.7959	?	<- adjusted (2)

(1) To be used for predicting $E(\ln(y_j))$
 (2) To be used for predicting $E(y_j)$

I'm setting us up to compare these estimates with those produced by **poisson**. When we estimate using **poisson**, we will not need to take logs because the Poisson model is stated in terms of y_j , not $\ln(y_j)$. In preparation for that, I have included two lines for the intercept — 8.5172, which is the intercept reported by **regress** and is the one appropriate for making predictions of $\ln(y)$ — and 9.0587, an intercept appropriate for making predictions of y and equal to 8.5172 plus $\sigma^2/2$. Poisson regression will estimate the 9.0587 result because Poisson is stated in terms of y rather than $\ln(y)$.

I placed a question mark in the column for the standard error of the adjusted intercept because, to calculate that, I would need to know the standard error of the estimated RMSE, and **regress** does not calculate that.

Let's now look at the results that **poisson** with option **vce(robust)** reports. We must not forget to specify option **vce(robust)** because otherwise, in this model that violates the Poisson assumption that $E(y_j) = \text{Var}(y_j)$, we would obtain incorrect standard errors.

```
. poisson y educ exp exp2, vce(robust)
note: you are responsible for interpretation of noncount dep. variable
```

```
Iteration 0: log pseudolikelihood = -1.484e+08
Iteration 1: log pseudolikelihood = -1.484e+08
Iteration 2: log pseudolikelihood = -1.484e+08
```

```
Poisson regression              Number of obs   =      5000
                                Wald chi2(3)        =       67.52
                                Prob > chi2         =      0.0000
Log pseudolikelihood = -1.484e+08  Pseudo R2       =      0.0183
```

y	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
educ	.0575636	.0127996	4.50	0.000	.0324769	.0826504
exp	.1074603	.0163766	6.56	0.000	.0753628	.1395578
exp2	-.0022204	.0003604	-6.16	0.000	-.0029267	-.0015141
_cons	9.016428	.2359002	38.22	0.000	8.554072	9.478784

So now we can fill in the rest of our table:

	truth	regress est.	S.E.	poisson est.	S.E.
educ	0.0600	0.0716	0.0100	0.0576	0.1280

exp	0.1000	0.1092	0.0129	0.1075	0.0164	
exp2	-0.0020	-0.0022	0.0003	-0.0022	0.0003	
<hr/>						
_cons	8.5172	8.2725	0.1856	?	?	<- (1)
	9.0587	8.7959	?	9.0164	0.2359	<- (2)
<hr/>						
(1) To be used for predicting $E(\ln(y_j))$						
(2) To be used for predicting $E(y_j)$						

I told you that Poisson works, and in this case, it works well. I'll now tell you that in all cases it works well, and it works better than log regression. You want to think about Poisson regression with the **vce(robust)** option as a better alternative to log regression.

How is Poisson better?

First off, Poisson handles outcomes that are zero. Log regression does not because $\ln(0)$ is $-\infty$. You want to be careful about what it means to handle zeros, however. Poisson handles zeros that arise in correspondence to the model. In the Poisson model, everybody participates in the $y_j = \exp(b_0 + \mathbf{X}_j\mathbf{b} + \varepsilon_j)$ process. Poisson regression does not handle cases where some participate and others do not, and among those who do not, had they participated, would likely produce an outcome greater than zero. I would never suggest using Poisson regression to handle zeros in an earned income model because those that earned zero simply didn't participate in the labor force. Had they participated, their earnings might have been low, but certainly they would have been greater than zero. Log linear regression does not handle that problem, either.

Natural zeros do arise in other situations, however, and a popular question on [Statalist](#) is whether one should recode those natural zeros as 0.01, 0.0001, or 0.0000001 to avoid the missing values when using log linear regression. The answer is that you should not recode at all; you should use Poisson regression with **vce(robust)**.

Secondly, small nonzero values, however they arise, can be influential in log-linear regressions. 0.01, 0.0001, 0.0000001, and 0 may be close to each other, but in the logs they are -4.61, -9.21, -16.12, and $-\infty$ and thus not close at all. Pretending that the values are close would be the same as pretending that that $\exp(4.61)=100$, $\exp(9.21)=9,997$, $\exp(16.12)=10,019,062$, and $\exp(\infty)=\infty$ are close to each other. Poisson regression understands that 0.01, 0.0001, 0.0000001, and 0 are indeed nearly equal.

Thirdly, when estimating with Poisson, you do not have to remember to apply the $\exp(\sigma^2/2)$ multiplicative adjustment to transform results from $\ln(y)$ to y . I wrote earlier that people who fit log regressions of course remember to apply the adjustment, but the sad fact is that they do not.

Finally, I would like to tell you that everyone who estimates log models knows about the Poisson-regression alternative and it is only you who have been out to lunch. You, however, are in esteemed company. At the recent [Stata Conference](#) in Chicago, I asked a group of knowledgeable researchers a loaded question, to which the right answer was Poisson regression with option **vce(robust)**, but they mostly got it wrong.

I said to them, "I have a process for which it is perfectly reasonable to assume that the mean of y_j is given by $\exp(b_0 + \mathbf{X}_j\mathbf{b})$, but I have no reason to believe that $E(y_j) = \text{Var}(y_j)$, which is to say, no reason to suspect that the process is Poisson. How would you suggest I estimate the model?" Certainly not using Poisson, they replied. Social scientists suggested I use log regression. Biostatisticians and health researchers suggested I use negative binomial regression even when I objected that the process was not the gamma mixture of Poissons that negative binomial regression assumes. "What else can you do?" they said and shrugged their collective shoulders. And of course, they just assumed over dispersion.

Based on those answers, I was ready to write this blog entry, but it turned out differently than I expected. I was going to slam negative binomial regression. Negative binomial regression makes assumptions about the variance, assumptions different from that made by Poisson, but assumptions nonetheless, and unlike the assumption made in Poisson, those assumptions do appear in the first-order conditions that determine the fitted coefficients that negative binomial regression reports. Not only would negative binomial's standard errors be wrong — which **vce(robust)** could fix — but the coefficients would be biased, too, and **vce(robust)** would not fix that. I planned to run simulations showing this.

When I ran the simulations, I was surprised by the results. The negative binomial estimator (Stata's **nbreg**) was remarkably robust to violations in variance assumptions as long as the data were overdispersed. In fact, negative binomial regression did about as well as Poisson regression. I did not run enough simulations to make generalizations, and theory tells me those generalizations have to favor Poisson, but the simulations suggested that if Poisson does do better, it's not in the first four decimal places. I was impressed. And disappointed. It would have been a dynamite blog entry.

So you'll have to content yourself with this one.

Others have preceded me in the knowledge that Poisson regression with **vce(robust)** is a better alternative to log-linear regression. I direct you to [Jeffery Wooldridge, *Econometric Analysis of Cross Section and Panel Data*, 2nd ed.](#), chapter 18. Or see [A. Colin Cameron and Pravin K. Trivedi, *Microeconomics Using Stata*, revised edition](#), chapter 17.3.2.

I first learned about this from a talk given by [Austin Nichols, *Regression for nonnegative skewed dependent variables*](#), given in 2010 at the [Stata Conference](#) in Boston. That talk goes far beyond what I have presented here, and I heartily recommend it.

Categories: [Statistics](#) Tags: [Huber](#), [log linear regression](#), [nbreg](#), [negative binomial regression](#), [Poisson regression](#), [Sandwich](#), [White](#)

22 Comments

1 Login ▾

G

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

♥ 3

Share

Best Newest Oldest**Rob Shaw**

a year ago

Thanks for a great post. One thing you don't mention is that poisson regression takes a LOT longer to run (10-20x in my experience) than regress. In cases with huge datasets that makes it impractical.

0 0 Reply Share ›

D

Dorothy Chiu

4 years ago

thank you for this! one question though - i applied the poisson regression to an outcome of (discrete) scores that is right skewed. however, when i run postestimation goodness of fit tests (estat gof), significant p-values are returned. it seems i should reject the null hypothesis that my data follows a poisson distribution, but.... is this necessarily the case?

0 0 Reply Share ›

SB

Simon Berrebi

5 years ago

"Actually regression does assume the variance is constant but since we are working the logs, that amounts to assuming that $\text{Var}(y_j)$ is proportional to y_j "

I've been looking for a reference on that point exactly. Is there a source that elaborates/proves that point?

0 0 Reply Share ›

C

conan007

7 years ago

Regarding handling of zero, why not try something like:

```
gen lny=ln(y+1)
```

```
reg lny x
```

You may get negative prediction of y, but you could censored those cases to be zero.

Also, since $\log(0) = -\infty$, one may consider such cases as censored, and thus tobit model:

```
gen lny=ln(y)
```

```
quietly sum lny
```

```
local newmin=r(min)-1
```

```
replace lny=`newmin'
```

```
tobit lny x, ll(`newmin')
```

Above, $\text{newmin} = r(\text{min}) - 1$, is used just to create a value lower than the $r(\text{min})$, and the actual value does not matter since they would be censored.

0 0 Reply Share ›

C

conan007

→ conan007

a year ago edited

Update: two recent papers discussed why something like $\log(y+1)$ is not a good idea:

<https://www.nber.org/papers...>

<https://www.jonathandroth.c...>

0 0 Reply Share ›

R **radek**

9 years ago

Thank you Bill for a great post. Would such solution be applicable in multilevel framework, basically replacing `mixed' used with `log(outcome)` with `mepoisson`?

0 0 Reply Share ›

kinggary

9 years ago

The general point of this post – that we should move to models that could be true at least in principle, such as event count models when analyzing event counts – is excellent. However, the specific advice to use a Poisson model with robust SEs is sensible if you are interested ONLY in the (difficult to interpret) coefficients, ONLY if the model has a very narrow form of misspecification, and ONLY if you have no interest in validating out of sample or cross-validation predictions within their standard errors. If instead you care about quantities of interest such as $\text{Prob}(Y < 10)$, for one of many examples, then Poisson with robust se's can be severely biased; to get all the estimates right, you need to get the specification right. Robust SE's do not fix the specification; if they differ from classical SEs, they reveal problems. The point is more general than event count models: if you are running any model for which Robust and classical SEs differ, then you should go back and fix the specification and try again; failing to do so reveals biases in at least some quantities for your analysis. Here's the longer form of this point: Gary King and Margaret E Roberts. 2014. "How Robust Standard Errors Expose Methodological Problems They Do Not Fix, And What To Do About It." *Political Analysis*, 1-21. Copy at <http://j.mp/1BQDeQT> (Thanks to @BrendanTHalpin for the reference to this interesting discussion.)

47 0 Reply Share ›

nahla betelmal

11 years ago

if the dependent variable has negative values as well, is it possible to add a constant to make it all positive and then run Poisson regression rather than log transformation and regress?

2 0 Reply Share ›

JS **Joao Santos Silva**

12 years ago

Dear Bill,

I was delighted to see your recommendation to use poisson rather than regress, but I have to confess that I was a bit disappointed not to see a mention to the paper Silvana Tenreiro and I wrote on the subject; please see here

www.mitpressjournals.org/do...

As far as I understand, we were the first to put forward the use of Poisson regression as work-horse for the estimation of constant elasticity models. We explain why that should be done (actually, the problem is much more serious than what is stated in your post; estimation of the log linear model generally leads to inconsistent estimates of the elasticities), and we even provide the Stata command to do it (see page 645).

After writing the paper we realized that the poisson command in Stata has some issues and we have written an alternative Stata command to estimate Poisson regression. This is described in our paper:

Santos Silva, J.M.C. and Tenreiro, S. (2011), "poisson: Some Convergence Issues," *STATA Journal*, 11(2), pp. 207-212.

Finally, about the use of nbreg; it is clear that if poisson is consistent then nbreg is also consistent. However, nbreg (like ZIP) is not invariant to the scale of the dependent variable because changing the scale of y changes the amount of overdispersion. Of course this is not an issue when y is a count, but is very unpleasant when y is just a non-negative variable without a natural scale. This is

[see more](#)

2 1 Reply Share ›

BillGould

→ Joao Santos Silva

12 years ago

Joao Santos Silva is correct, my blog post is directly related to Santos Silva and Tenreyro's paper,

Santos Silva, J.M.C., S. Tenreyro 2006. "The log of gravity", *The Review of Economics and Statistics* 88(4):641-658.

I should have cited it. The use of poisson regression is related to a strand of research in econometrics and statistics that is huge and goes under the name quasi-likelihood theory. In addition to Santos Silva and Tenreyro's paper, I recommend that interested readers see

Jeffery Wooldridge, *Econometric Analysis of Cross Section and Panel Data*, 2nd ed.

which I did cite in my blog entry, and two more that I did not,

A. Colin Cameron and Pravin K. Trivedi, *Microeconometrics: Theory and Applications*.

McCullagh P. and J. A. Nelder, *Generalized Linear Models*, 2nd ed.

Joao mentioned that they provided Stata code that checks identification conditions and we are looking into adding those features to Stata's poisson command. These checks are discussed in

Santos Silva, J.M.C., S. Tenreyro 2010. "On the existence of the maximum likelihood estimates in Poisson regression",

[see more](#)

0 0 Reply Share ›

AP Alari Paulus

12 years ago

Thanks, this is very useful. PS There seem to be a couple of typos:

$y_j = \exp(5000 + 0.06 \cdot \text{educ}_j + 0.1 \cdot \text{exp}_j - 0.002 \cdot \text{exp}_j^2 + \varepsilon_j)$ should be

$y_j = \exp(8.5172 + 0.06 \cdot \text{educ}_j + 0.1 \cdot \text{exp}_j - 0.002 \cdot \text{exp}_j^2 + \varepsilon_j)$

and

" $E(\exp(\varepsilon_j))$ is not zero" should be " $E(\exp(\varepsilon_j))$ is not one".

0 0 Reply Share ›

BillGould

→ Alari Paulus

12 years ago

Alari is right on both counts. I fixed them. Thank you.

0 0 Reply Share ›

KD Kevin Denny

13 years ago

I think it is also worth looking at "The log of gravity" by Santos Silva & Tenreyro (REStats) in this context, if you are hoping to recover elasticities from log linear models.

4 1 Reply Share ›

G guest

13 years ago

Is there any way to use poisson in IV as the second step in a 2SLS?

0 0 Reply Share ›

D

dvmaster → guest

12 years ago

There's also an `ivpois` command by Austin Nichols at SSC. Or you can write up your own gmm estimator in a line or two.

0 0 Reply Share ›

G

guest → guest



12 years ago

Yes, go to <http://privatewww.essex.ac...>

1 0 Reply Share ›

Maarten Buis

13 years ago edited

Another nice reference on this issue is: Nicholas J. Cox, Jeff Warburton, Alona Armstrong, Victoria J. Holliday (2007) "Fitting concentration and load rating curves with generalized linear models" Earth Surface Processes and Landforms, 33(1):25-39. DOI: [10.1002/esp.1523](https://doi.org/10.1002/esp.1523)

I can especially recommend it to *non*-geographers as it is well written and it is just refreshing not to see the same old examples all over again.

0 0 Reply Share ›

MP

Marc Pelath

13 years ago

Great post, Bill, I'm going to reread it ASAP with some data in front of me. We've got a recurring problem for which this may be a great solution.

NB The link to Austin's paper isn't working for me, although I can probably figure out the right one.

0 0 Reply Share ›

BillGould

➔ Marc Pelath



13 years ago

Link to Austin's paper is fixed.

0 0 Reply Share ›

A

Anna Manzoni

➔ BillGould



8 years ago

what is the most appropriate way to cite this blog post?

0 0 Reply Share ›

BillGould

➔ Anna Manzoni



8 years ago

Gould, William. 22 August 2011. "Use poisson rather than regress; tell a friend" at _The Stata Blog_, "<http://blog.stata.com/2011/...>".

0 0 Reply Share ›

A

Anna Manzoni

➔ BillGould 8 years ago



Thank you!

0 0 Reply Share ›

[Subscribe](#)[Privacy](#)[Do Not Sell My Data](#)

[Advanced Mata: Pointers](#) [2011 Stata Conference recap](#)

[RSS](#)[Twitter](#)[Facebook](#)

 Follow 10,387

Subscribe to the Stata Blog

Receive email notifications of new blog posts

Name

Email Address*

Subscribe

Recent articles

- [Announcing StataNow](#)
- [New FAQs about customizable tables are here!](#)
- [From datasets to framesets and alias variables: Data management advances in Stata](#)
- [StataCorp's Author Support Program—Publish with confidence](#)
- [A Stata command to run ChatGPT](#)

Archives

- [2024](#)
- [2023](#)
- [2022](#)
- [2021](#)
- [2020](#)
- [2019](#)
- [2018](#)
- [2017](#)
- [2016](#)
- [2015](#)
- [2014](#)
- [2013](#)
- [2012](#)
- [2011](#)
- [2010](#)

Categories

- [Blogs](#)
- [Company](#)
- [Data Management](#)
- [Graphics](#)
- [Mathematics](#)
 - [Linear Algebra](#)
 - [Numerical Analysis](#)

- [Performance](#)
 - [Hardware](#)
 - [Memory](#)
 - [Multiprocessing](#)
- [Programming](#)
 - [Mata](#)
- [Reporting](#)
- [Resources](#)
 - [Documentation](#)
 - [Meetings](#)
 - [Support](#)
- [Stata Products](#)
 - [New Books](#)
 - [New Products](#)
- [Statistics](#)

Tags

[#StataProgramming](#) [ado](#) [ado-file](#) [Bayesian](#) [bayesmh](#) [binary](#) [biostatistics](#) [books](#) [collections](#) [conference](#) [customizable tables](#)
[econometrics](#) [estimation](#) [Excel](#) [forthcoming](#) [gmm](#) [graphics](#) [import](#) [marginal effects](#) [margins](#) [Mata](#) [meeting](#) [mlexp](#) [new release](#) [nonlinear model](#) [numerical](#)
[analysis](#) [OLS](#) [power](#) [precision](#) [probit](#) [programming](#) [Python](#) [random numbers](#) [release](#) [reporting](#) [runiform\(\)](#) [sample size](#) [SEM](#) [simulation](#) [Stata 17](#)
[stata.press](#) [statistics](#) [tables](#) [time series](#) [treatment effects](#)

Links

- [Stata](#)
- [Stata Press](#)
- [The Stata Journal](#)
- [Stata FAQs](#)
- [Statalist](#)
- [Statalist archives](#)
- [Links to others](#)

[Top](#) www.stata.com

Copyright © 2010-2024 StataCorp LLC

[Terms of use](#)