

[Description](#)

[Options](#)

[Acknowledgment](#)

[Quick start](#)

[Remarks and examples](#)

[References](#)

[Menu](#)

[Stored results](#)

[Also see](#)

[Syntax](#)

[Methods and formulas](#)

## Description

`tpoisson` fits a truncated Poisson regression model when the number of occurrences of an event is restricted to be above a truncation point, below a truncation point, or between two truncation points. Truncated Poisson models are appropriate when neither the dependent variable nor the covariates are observed in the truncated part of the distribution. By default, `tpoisson` assumes left-truncation occurs at zero, but truncation may be specified at other fixed points or at values that vary across observations.

## Quick start

Truncated Poisson regression of `y` on `x1` and `x2` with left-truncation at 0

```
tpoisson y x1 x2
```

Add categorical variable `a` using [factor-variable](#) syntax

```
tpoisson y x1 x2 i.a
```

Same as above, but report incidence-rate ratios and use a constant truncation point of 4

```
tpoisson y x1 x2 i.a, irr ll(4)
```

With offset variable `lnexp`

```
tpoisson y x1 x2 i.a, offset(lnexp)
```

Same as above, but with a variable truncation point stored in variable `min`

```
tpoisson y x1 x2 i.a, offset(lnexp) ll(min)
```

With variable left- and right-truncation points

```
tpoisson y x1 x2, ll(min) ul(max)
```

With variable right-truncation points

```
tpoisson y x1 x2, ul(max)
```

Constrain the coefficients for `2.a` and `3.a` to equality

```
constraint define 1 2.a = 3.a
tpoisson y x1 x2 i.a, constraints(1)
```

## Menu

Statistics > Count outcomes > Truncated Poisson regression

# Syntax

```
tpoisson depvar [indepvars] [if] [in] [weight] [, options]
```

<i>options</i>	Description
Model	
<code>noconstant</code>	suppress constant term
<code>ll(# <i>varname</i>)</code>	lower limit for truncation; default is <code>ll(0)</code> when neither <code>ll()</code> nor <code>ul()</code> is specified
<code>ul(# <i>varname</i>)</code>	upper limit for truncation
<code>exposure(<i>varname</i><sub>e</sub>)</code>	include <code>ln(<i>varname</i><sub>e</sub>)</code> in model with coefficient constrained to 1
<code>offset(<i>varname</i><sub>o</sub>)</code>	include <i>varname</i> <sub>o</sub> in model with coefficient constrained to 1
<code>constraints(<i>constraints</i>)</code>	apply specified linear constraints
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>oim</code> , <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>opg</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>irr</code>	report incidence-rate ratios
<code>nocnsreport</code>	do not display constraints
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
<code>collinear</code>	keep collinear variables
<code>coeflegend</code>	display legend instead of statistics

*indepvars* may contain factor variables; see [U] 11.4.3 Factor variables.

*depvar* and *indepvars* may contain time-series operators; see [U] 11.4.4 Time-series varlists.

`bayes`, `bayesboot`, `bootstrap`, `by`, `collect`, `fmm`, `fp`, `jackknife`, `rolling`, `statsby`, and `svy` are allowed; see [U] 11.1.10 Prefix commands. For more details, see [BAYES] `bayes: tpoisson` and [FMM] `fmm: tpoisson`.

Weights are not allowed with the `bootstrap` prefix; see [R] `bootstrap`.

`vce()` and weights are not allowed with the `svy` prefix; see [SVY] `svy`.

`fweights`, `iweights`, and `pweights` are allowed; see [U] 11.1.6 `weight`.

`collinear` and `coeflegend` do not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

## Options

### Model

`noconstant`; see [R] [Estimation options](#).

`ll(#|varname)` and `ul(#|varname)` specify the lower and upper limits for truncation, respectively. You may specify nonnegative integer values for one or both.

When neither `ll()` nor `ul()` is specified, the default is zero truncation, `ll(0)`, equivalent to left-truncation at zero.

`exposure(varnamee)`, `offset(varnameo)`, `constraints(constraints)`; see [R] [Estimation options](#).

### SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`, `opg`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce\\_option](#).

### Reporting

`level(#)`; see [R] [Estimation options](#).

`irr` reports estimated coefficients transformed to incidence-rate ratios, that is,  $e^{\beta_i}$  rather than  $\beta_i$ . Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. `irr` may be specified at estimation or when replaying previously estimated results.

`nocnsreport`; see [R] [Estimation options](#).

*display\_options*: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

### Maximization

*maximize\_options*: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

Setting the optimization type to `technique(bhhh)` resets the default `vcetype` to `vce(opg)`.

The following options are available with `tpoisson` but are not shown in the dialog box:

`collinear`, `coeflegend`; see [R] [Estimation options](#).

## Remarks and examples

`tpoisson` fits a truncated Poisson regression model by maximum likelihood estimation when the number of occurrences of an event is restricted to be above a truncation point, below a truncation point, or between two truncation points. If the dependent variable is not truncated, standard Poisson regression may be more appropriate; see [R] [poisson](#).

When the data are truncated, we do not observe either the dependent variable or the covariates. For example, consider a study about the number of days that individuals with hyperglycemia are hospitalized after presenting to the hospital. If we select our sample only from admission records, then the sample is truncated at zero because we have data only on individuals who stayed at least one day. Now assume that we are relying on billing data and that hospitals may submit either a final bill when a patient is discharged or an interim bill every 30 days. In this case, we have no information about patients who are hospitalized fewer than 1 day or more than 30 days. Our data are left-truncated at 0 and right-truncated at 30.

A related phenomenon is censoring. For censored observations, we observe complete covariate information but only a censored value of the dependent variable. Different research designs can give rise to censored data or truncated data. See [R] [cpoisson](#) for information about censored Poisson regression.

Truncated Poisson regression was first proposed by [Grogger and Carson \(1991\)](#). For an introduction to Poisson regression, see [Cameron and Trivedi \(2005, 2022\)](#) and [Long and Freese \(2014\)](#). For an introduction to truncated Poisson models, see [Cameron and Trivedi \(2013\)](#) and [Long \(1997, chap. 8\)](#).

▷ Example 1: Left-truncation at zero

Consider the [Simonoff \(2003\)](#) dataset of running shoes for a sample of runners who registered an online running log. A running-shoe marketing executive is interested in knowing how the number of pairs of running shoes purchased relates to other factors such as gender, marital status, age, education, income, typical number of runs per week, average miles run per week, and the preferred type of running. These data are naturally truncated at zero. A truncated Poisson model is fit to the number of pairs of shoes owned on runs per week, miles run per week, gender, age, and marital status.

No options are needed because zero truncation is the default for `tpoisson`.

```
. use https://www.stata-press.com/data/r19/runshoes
(Running shoes)

. toipson shoes rpweek mpweek male age married

Iteration 0:  Log likelihood = -88.328151
Iteration 1:  Log likelihood = -86.272639
Iteration 2:  Log likelihood = -86.257999
Iteration 3:  Log likelihood = -86.257994

Truncated Poisson regression               Number of obs   =           60
Limits: lower =           0               LR chi2(5)      =          22.75
        upper =          +inf             Prob > chi2     =          0.0004
Log likelihood = -86.257994               Pseudo R2      =          0.1165
```

shoes	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
rpweek	.1575811	.1097893	1.44	0.151	-.057602	.3727641
mpweek	.0210673	.0091113	2.31	0.021	.0032094	.0389252
male	.0446134	.2444626	0.18	0.855	-.4345246	.5237513
age	.0185565	.0137786	1.35	0.178	-.008449	.045562
married	-.1283912	.2785044	-0.46	0.645	-.6742498	.4174674
_cons	-1.205844	.6619774	-1.82	0.069	-2.503296	.0916078

Using the zero-truncated Poisson regression with these data, only the coefficient on average miles per week is statistically significant at the 5% level.



► Example 2: Left-truncation with a fixed-truncation point

Semiconductor manufacturing requires that silicon wafers be coated with a layer of metal oxide. The depth of this layer is strictly controlled. In this example, a critical oxide layer is designed for  $300 \pm 20$  angstroms (Å).

After the oxide layer is coated onto a wafer, the wafer enters a photolithography step in which the lines representing the electrical connections are printed on the oxide and later etched and filled with metal. The widths of these lines are measured. In this example, they are controlled to  $90 \pm 5$  micrometers ( $\mu\text{m}$ ).

After these and other steps, each wafer is electrically tested at probe. If too many failures are discovered, the wafer is rejected and sent for engineering analysis. In this example, the maximum number of probe failures tolerated for this product is 10.

A major failure at probe has been encountered—88 wafers had more than 10 failures each. The 88 wafers that failed were tested using 4 probe machines. The engineer suspects that the failures were a result of faulty probe machines, poor depth control, or poor line widths. The line widths and depths in these data are the actual measurement minus its specification target, 300 Å for the oxide depths and 90  $\mu\text{m}$  for the line widths.

The following table tabulates the average failure rate for each probe using Stata’s mean command; see [R] [mean](#).

```
. use https://www.stata-press.com/data/r19/probe
(Silicon wafers)
. mean failures, over(probe)
```

Mean estimation		Number of obs = 88		
	Mean	Std. err.	[95% conf. interval]	
c.failures@probe				
1	15.875	1.186293	13.51711	18.23289
2	14.95833	.5912379	13.78318	16.13348
3	16.47059	.9279866	14.62611	18.31506
4	23.09677	.9451117	21.21826	24.97529

The 95% confidence intervals in this table suggest that there are about 5–11 additional failures per wafer on probe 4. These are unadjusted for varying line widths and oxide depths. Possibly, probe 4 received the wafers with larger line widths or extreme oxide depths.

Truncated Poisson regression more clearly identifies the root causes for the increased failures by estimating the differences between probes adjusted for the line widths and oxide depths. It also allows us to determine whether the deviations from specifications in line widths or oxide depths might be contributing to the problem.

```
. tpoisson failures i.probe depth width, ll(10) nolog
Truncated Poisson regression      Number of obs      =          88
Limits: lower =          10      LR chi2(5)            =          73.70
      upper =        +inf      Prob > chi2            =          0.0000
Log likelihood = -239.35746      Pseudo R2           =          0.1334
```

failures	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
probe						
2	-.1113037	.1019786	-1.09	0.275	-.3111781	.0885707
3	.0114339	.1036032	0.11	0.912	-.1916245	.2144924
4	.4254115	.0841277	5.06	0.000	.2605242	.5902989
depth	-.0005034	.0033375	-0.15	0.880	-.0070447	.006038
width	.0330225	.015573	2.12	0.034	.0025001	.063545
_cons	2.714025	.0752617	36.06	0.000	2.566515	2.861536

The coefficients listed for the probes are testing the null hypothesis:  $H_0 : \text{probe}_i = \text{probe}_1$ , where  $i$  equals 2, 3, and 4. Because the only coefficient that is statistically significant is the one for testing for  $H_0 : \text{probe}_4 = \text{probe}_1$ ,  $p < 0.001$ , and because the  $p$ -values for the other probes are not statistically significant, that is,  $p \geq 0.275$ , the implication is that there is a difference between probe 4 and the other machines. Because the coefficient for this test is positive, 0.425, the conclusion is that the average failure rate for probe 4, after adjusting for line widths and oxide depths, is higher than the other probes. Possibly, probe 4 needs calibration or the head used with this machine is defective.

Line-width control is statistically significant,  $p = 0.034$ , but variation in oxide depths is not causing the increased failure rate. The engineer concluded that the sudden increase in failures is the result of two problems. First, probe 4 is malfunctioning, and second, there is a possible lithography or etching problem.



## Stored results

tpoisson stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(r2_p)</code>	pseudo- $R^2$
<code>e(ll)</code>	log likelihood
<code>e(ll_0)</code>	log likelihood, constant-only model
<code>e(N_clust)</code>	number of clusters
<code>e(chi2)</code>	$\chi^2$
<code>e(p)</code>	$p$ -value for model test
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	tpoisson
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(llopt)</code>	contents of <code>ll()</code> , or 0 if neither <code>ll()</code> nor <code>ul()</code> is specified
<code>e(ulopt)</code>	contents of <code>ul()</code> , if specified
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(offset)</code>	linear offset variable
<code>e(chi2type)</code>	Wald or LR; type of model $\chi^2$ test
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	max or min; whether optimizer is to perform maximization or minimization
<code>e(ml_method)</code>	type of <code>ml</code> method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	b V
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(Cns)</code>	constraints matrix
<code>e(ilog)</code>	iteration log (up to 20 iterations)
<code>e(gradient)</code>	gradient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
<code>e(V_modelbased)</code>	model-based variance

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`

matrix containing the coefficients with their standard errors, test statistics,  $p$ -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

## Methods and formulas

For a nonnegative count outcome  $Y$  with left-truncation point  $ll_j$  and right-truncation point  $ul_j$ , we can write the truncated Poisson model as

$$f(y_j) = \frac{\exp(-\lambda_j)\lambda_j^{y_j}}{y_j! \Pr(ll_j < Y < ul_j | \xi_j)}$$

where

$$\begin{aligned}\xi_j &= \mathbf{x}_j\beta + \text{offset}_j \\ \lambda_j &= \exp(\xi_j)\end{aligned}$$

and  $\mathbf{x}_j$  is a vector of observed covariates. The conditional probability of observing  $y_j$  events, therefore given by  $ll_j < y_j < ul_j$ , is

$$\Pr(Y = y_j | ll_j < y_j < ul_j, \mathbf{x}_j) = \frac{\exp(-\lambda_j)\lambda_j^{y_j}}{y_j! \Pr(ll_j < Y < ul_j | \mathbf{x}_j)}$$

The log likelihood is given by

$$\ln L = \sum_{j=1}^n w_j [-\lambda_j + \xi_j y_j - \ln(y_j!) - \ln \{ \Pr(ll_j < Y < ul_j | \xi_j) \}]$$

If no weights are specified,  $w_j = 1$ .

This command supports the Huber/White/sandwich estimator of the variance and its clustered version using `vce(robust)` and `vce(cluster clustvar)`, respectively. See [P] [\\_robust](#), particularly [Maximum likelihood estimators](#) and [Methods and formulas](#).

`tpoisson` also supports estimation with survey data. For details on VCEs with survey data, see [SVY] [Variance estimation](#).

## Acknowledgment

We gratefully acknowledge the previous work by Joseph Hilbe (1944–2017) (1999), coauthor of the Stata Press book [Generalized Linear Models and Extensions](#).



## References

- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- . 2013. *Regression Analysis of Count Data*. 2nd ed. New York: Cambridge University Press.
- . 2022. *Microeconometrics Using Stata*. 2nd ed. College Station, TX: Stata Press.
- Farbmacher, H. 2011. Estimation of hurdle models for overdispersed count data. *Stata Journal* 11: 82–94.
- Grogger, J. T., and R. T. Carson. 1991. Models for truncated counts. *Journal of Applied Econometrics* 6: 225–238. <https://doi.org/10.1002/jae.3950060302>.
- Hardin, J. W., and J. M. Hilbe. 2015. Regression models for count data from truncated distributions. *Stata Journal* 15: 226–246.
- Hilbe, J. M. 1999. sg102: Zero-truncated Poisson and negative binomial regression. *Stata Technical Bulletin* 47: 37–40. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 233–236. College Station, TX: Stata Press.
- Long, J. S. 1997. *Regression Models for Categorical and Limited Dependent Variables*. Thousand Oaks, CA: Sage.
- Long, J. S., and J. Freese. 2014. *Regression Models for Categorical Dependent Variables Using Stata*. 3rd ed. College Station, TX: Stata Press.
- Simonoff, J. S. 2003. *Analyzing Categorical Data*. New York: Springer. <https://doi.org/10.1007/978-0-387-21727-7>.

## Also see

- [R] **tpoisson postestimation** — Postestimation tools for **tpoisson**
- [R] **poisson** — Poisson regression
- [R] **nbreg** — Negative binomial regression
- [R] **tnbreg** — Truncated negative binomial regression
- [R] **zinb** — Zero-inflated negative binomial regression
- [R] **zip** — Zero-inflated Poisson regression
- [BAYES] **bayes: tpoisson** — Bayesian truncated Poisson regression
- [FMM] **fmm: tpoisson** — Finite mixtures of truncated Poisson regression models
- [SVY] **svy estimation** — Estimation commands for survey data
- [XT] **xtpoisson** — Fixed-effects, random-effects, and population-averaged Poisson models
- [U] **20 Estimation and postestimation commands**

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on [citing Stata documentation](#).

