

Description

gsem not only allows models of the form $y_i = \mathbf{x}_i\beta + u_i$ but also allows

$$g\{E(y_i)\} = \mathbf{x}_i\beta$$

$$y_i \sim F$$

where you can choose F and $g(\cdot)$ from a menu. F is called the family, and $g(\cdot)$ is called the link. One set of choices is the Gaussian distribution for F and the identity function for $g(\cdot)$. In that case, gsem reproduces linear regression. Other combinations of $g(\cdot)$ and F produce other popular models, including logit (also known as logistic regression), probit, multinomial logit, Poisson regression, and more.

Syntax

gsem *paths* ..., ... *family_and_link_options*

<i>family_and_link_options</i>	Description
<u>family</u> (<i>family</i>)	distribution family; default is family(gaussian)
<u>link</u> (<i>link</i>)	link function; default varies per family
cloglog	synonym for family(bernoulli) link(cloglog)
exponential	synonym for family(exponential) link(log)
gamma	synonym for family(gamma) link(log)
logit	synonym for family(bernoulli) link(logit)
loglogistic	synonym for family(loglogistic) link(log)
lognormal	synonym for family(lognormal) link(log)
llogistic	synonym for family(llogistic) link(log)
lnormal	synonym for family(lnormal) link(log)
mlogit	synonym for family(multinomial) link(logit)
nbreg	synonym for family(nbinomial mean) link(log)
ocloglog	synonym for family(ordinal) link(cloglog)
ologit	synonym for family(ordinal) link(logit)
oprobit	synonym for family(ordinal) link(probit)
poisson	synonym for family(poisson) link(log)
probit	synonym for family(bernoulli) link(probit)
regress	synonym for family(gaussian) link(identity)
weibull	synonym for family(weibull) link(log)
<u>exposure</u> (<i>varname</i> _e)	include ln(<i>varname</i> _e) with coefficient constrained to 1
<u>offset</u> (<i>varname</i> _o)	include <i>varname</i> _o with coefficient constrained to 1

<i>family</i>	Description
<code>gaussian [, <i>options</i>]</code>	Gaussian (normal); the default
<code>bernoulli</code>	Bernoulli
<code>beta</code>	beta
<code>binomial [# <i>varname</i>]</code>	binomial; default number of binomial trials is 1
<code>ordinal</code>	ordinal
<code>multinomial</code>	multinomial
<code>poisson [, <i>poisson</i>]</code>	Poisson
<code>nbinomial [mean <i>constant</i>]</code>	negative binomial; default dispersion is mean
<code>exponential [, <i>survival</i>]</code>	exponential
<code>gamma [, <i>survival</i>]</code>	gamma
<code>loglogistic [, <i>survival</i>]</code>	loglogistic
<code>lognormal [, <i>survival</i>]</code>	lognormal
<code>weibull [, <i>survival</i>]</code>	Weibull
<code>pointmass #</code>	point-mass density at #

<i>link</i>	Description
<code>identity</code>	identity
<code>log</code>	log
<code>logit</code>	logit
<code>probit</code>	probit
<code>cloglog</code>	complementary log–log

<i>options</i>	Description
<code>ldepvar(<i>varname</i>)</code>	lower depvar for interval-response data
<code>udepvar(<i>varname</i>)</code>	upper depvar for interval-response data
<code>lcensored(<i>varname</i> #)</code>	lower limit for left-censoring
<code>rcensored(<i>varname</i> #)</code>	upper limit for right-censoring

Only allowed with `family(gaussian)` with `link(identity)`.

<i>poisson</i>	Description
<code>ltruncated(<i>varname</i> #)</code>	lower limit for left-truncation

<i>survival</i>	Description
<code>ltruncated(<i>varname</i> #)</code>	lower limit for left-truncation
<code>failure(<i>varname</i>)</code>	indicator for failure event
<code>ph</code>	proportional hazards parameterization
<code>aft</code>	accelerated failure-time parameterization

`ph` is allowed only with families `exponential` and `weibull`.

`aft` is allowed only with families `exponential`, `gamma`, `loglogistic`, `lognormal`, and `weibull`.

If you specify both `family()` and `link()`, not all combinations make sense. You may choose from the following combinations:

	identity	log	logit	probit	cloglog
Gaussian	D	x			
Bernoulli			D	x	x
beta			D	x	x
binomial			D	x	x
ordinal			D	x	x
multinomial			D		
Poisson		D			
negative binomial		D			
exponential		D			
Weibull		D			
gamma		D			
loglogistic		D			
lognormal		D			
pointmass	D				

D denotes the default.

Options

`family(family)` and `link(linkname)` specify F and $g(\cdot)$. If neither is specified, linear regression is assumed.

Two of the families allow optional arguments:

`family(binomial [# | varname])` specifies that data are in binomial form, that is, that the response variable records the number of successes from a series of Bernoulli trials. The number of trials is given either as a constant number or as a varname that allows the number of trials to vary over observations, or it is not given at all. In the last case, the number of trials is thus equivalent to specifying `family(bernoulli)`.

`family(nbinomial [mean | constant])` specifies a negative binomial model, a Poisson model with overdispersion. Be aware, however, that even Poisson models can have overdispersion if latent variables are included in the model. Let's use the term "conditional overdispersion" to refer to dispersion above and beyond that implied by latent variables, if any.

That conditional overdispersion can take one of two forms. In mean overdispersion, the conditional overdispersion is a linear function of the conditional (predicted) mean. Constant overdispersion refers to the conditional overdispersion being, of course, constant.

If you do not specify mean or constant, then mean is assumed.

`family(pointmass #)` is a special family allowed when `lclass()` is also specified.

`cloglog`, `exponential`, `gamma`, `logit`, `loglogistic`, `lognormal`, `llogistic`, `lnormal`, `mlogit`, `nbreg`, `ocloglog`, `ologit`, `oprobit`, `poisson`, `probit`, `regress`, and `weibull` are shorthands for specifying popular models.

`exposure(varnamee)` and `offset(varnameo)` are most commonly used with families `poisson` and `nbreg`, that is, they typically concern count models.

`exposure()` specifies a variable that reflects the amount of exposure—usually measured in time units—for each observation over which the responses were counted. If one observation was exposed for twice the time of another, and the observations were otherwise identical, one would expect twice as many events to be counted. To assume that, $\ln(\text{varname}_e)$ is entered into $\mathbf{x}_i\beta$ with coefficient constrained to be 1.

`offset()` enters varname_o into $\mathbf{x}_i\beta$ with coefficient constrained to be 1. `offset()` is just another way of specifying `exposure()` where the offset variable is the log of amount of exposure.

If neither `exposure()` nor `offset()` is specified, observations are assumed to have equal amounts of exposure.

`ldepvar(varname)` and `udepvar(varname)` specify that each observation can be point data, interval data, left-censored data, or right-censored data. The type of data for a given observation is determined by the values in y_i and *varname*. The following specifications are equivalent:

```
depvar1 <- ... , family(gauss, depvar(depvar2))
```

```
depvar2 <- ... , family(gauss, ldepvar(depvar1))
```

Thus only one of `ldepvar()` or `udepvar()` is allowed. In either case, depvar_1 and depvar_2 should have the following form:

Type of data		depvar_1	depvar_2
point data	$a = [a, a]$	a	a
interval data	$[a, b]$	a	b
left-censored data	$(-\infty, b]$.	b
right-censored data	$[a, +\infty)$	a	.

`lcensored(varname|#)` and `rcensored(varname|#)` indicate the lower and upper limits for censoring, respectively. You may specify only one.

`lcensored(arg)` specifies that observations with $y_i \leq \text{arg}$ are left-censored and the remaining observations are not.

`rcensored(arg)` specifies that observations with $y_i \geq \text{arg}$ are right-censored and the remaining observations are not.

Neither `lcensored()` nor `rcensored()` may not be combined with `ldepvar()` or `udepvar()`.

`ltruncated(varname|#)` indicates the lower limits for truncation.

`ltruncated(arg)` specifies that the distribution is truncated on the left at *arg*, meaning that $y_i \leq \text{arg}$ is not within the support for the corresponding distribution family. This option rescales the underlying density function to accommodate the truncated support for y_i . This means that values of y_i that are less than or equal to *arg* do not contribute to the likelihood. For survival families, this means that time (time at risk) starts at *arg* instead of at 0.

`failure(varname)` specifies the failure event.

If `failure()` is not specified, all observations are assumed to indicate a failure.

If `failure(varname)` is specified, *varname* is interpreted as an indicator variable; 0 and missing mean censored, and all other values are interpreted as representing failure.

`ph` specifies the proportional hazards parameterization and is allowed with families `exponential` and `weibull`. This is the default parameterization for these families.

`aft` specifies the accelerated failure-time parameterization and is allowed with families `exponential`, `gamma`, `loglogistic`, `lognormal`, and `weibull`. This is an optional parameterization for `exponential` and `weibull` but the only parameterization for the others.

Remarks and examples

In the command language, the family-and-link options may be specified at the end of a command among the global options:

```
. gsem ..., ... family(...) link(...) ...
. gsem ..., ... poisson exposure(time) ...
```

Specified that way, the options apply to all the response variables. Alternatively, they may be specified inside paths to affect single equations:

```
. gsem (y1 <- x1 x2, family(...) poisson(...)) (y2 <- x2 L) ...
. gsem (y1 <- x1 x2, family(...) link(...))          ///
      (y2 <- x2 L, family(...) link(...)) ...
. gsem (y1 <- x1 x2, poisson exposure(time)) (y2 <- x2 L) ...
. gsem (y1 <- x1 x2, poisson exposure(time))          ///
      (y2 <- x2 L, family(...) link(...)) ...
```

On a different topic, it is worth noting that you can fit exponential-regression models with `family(gamma) link(log)` if you constrain the log of the scale parameter to be 0 with `gsem`'s `constraints()` option. For instance, you might type

```
. constraint 1 _b[/y:logs] = 0
. gsem (y <- x1 x2, gamma), constraints(1)
```

The name `_b[/y:logs]` changes according to the name of the dependent variable. Had `y` instead been named `waitingtime`, the parameter would have been named `_b[/waitingtime:logs]`. Rather than remembering that, remember instead that the best way to discover the names of parameters is to type

```
. gsem (waitingtime <- x1 x2, gamma), noestimate
```

and then look at the output to discover the names. See [\[SEM\] sem and gsem option constraints\(\)](#).

For examples of generalized response variables, see the following:

- [SEM] **Example 27g**: Single-factor measurement model (generalized response)
- [SEM] **Example 28g**: One-parameter logistic IRT (Rasch) model
- [SEM] **Example 29g**: Two-parameter logistic IRT model
- [SEM] **Example 30g**: Two-level measurement model (multilevel, generalized response)
- [SEM] **Example 31g**: Two-factor measurement model (generalized response)
- [SEM] **Example 32g**: Full structural equation model (generalized response)
- [SEM] **Example 33g**: Logistic regression
- [SEM] **Example 34g**: Combined models (generalized responses)
- [SEM] **Example 35g**: Ordered probit and ordered logit
- [SEM] **Example 36g**: MIMIC model (generalized response)
- [SEM] **Example 37g**: Multinomial logistic regression
- [SEM] **Example 39g**: Three-level model (multilevel, generalized response)
- [SEM] **Example 41g**: Two-level multinomial logistic regression (multilevel)
- [SEM] **Example 43g**: Tobit regression
- [SEM] **Example 44g**: Interval regression
- [SEM] **Example 45g**: Heckman selection model
- [SEM] **Example 46g**: Endogenous treatment-effects model
- [SEM] **Example 47g**: Exponential survival model
- [SEM] **Example 48g**: Loglogistic survival model with censored and truncated data
- [SEM] **Example 49g**: Multiple-group Weibull survival model
- [SEM] **Example 50g**: Latent class model
- [SEM] **Example 52g**: Latent profile model

Also see

- [SEM] **gsem** — Generalized structural equation model estimation command
- [SEM] **Intro 2** — Learning the language: Path diagrams and command language
- [SEM] **sem and gsem path notation** — Command syntax for path diagrams
- [SEM] **gsem path notation extensions** — Command syntax for path diagrams

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark 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](#).