

## gsem family-and-link options — Family-and-link options

[Syntax](#)    [Description](#)    [Options](#)    [Remarks and examples](#)    [Also see](#)

## Syntax

`gsem paths ... , ... family_and_link_options`

<i>family_and_link_options</i>	Description
<code><u>family</u>(<i>family</i>)</code>	distribution family; default is <code>family(gaussian)</code>
<code><u>link</u>(<i>link</i>)</code>	link function; default varies per family
<code>cloglog</code>	synonym for <code>family(bernoulli) link(cloglog)</code>
<code>gamma</code>	synonym for <code>family(gamma) link(log)</code>
<code>logit</code>	synonym for <code>family(bernoulli) link(logit)</code>
<code>nbreg</code>	synonym for <code>family(nbreg mean) link(log)</code>
<code>mlogit</code>	synonym for <code>family(multinomial) link(logit)</code>
<code>ocloglog</code>	synonym for <code>family(ordinal) link(cloglog)</code>
<code>oprobit</code>	synonym for <code>family(ordinal) link(logit)</code>
<code>poisson</code>	synonym for <code>family(poisson) link(log)</code>
<code>probit</code>	synonym for <code>family(bernoulli) link(probit)</code>
<code>regress</code>	synonym for <code>family(gaussian) link(identity)</code>
<code><u>exposure</u>(<i>varname<sub>e</sub></i>)</code>	include $\ln(\text{varname}_e)$ with coefficient constrained to 1
<code><u>offset</u>(<i>varname<sub>o</sub></i>)</code>	include <i>varname<sub>o</sub></i> with coefficient constrained to 1

<i>family</i>	Description
<code><u>gaussian</u> [ , <i>options</i> ]</code>	Gaussian (normal); the default
<code><u>bernoulli</u></code>	Bernoulli
<code><u>binomial</u> [#   <i>varname</i> ]</code>	binomial; default number of binomial trials is 1
<code><u>gamma</u></code>	gamma
<code><u>multinomial</u></code>	multinomial
<code><u>nbinomial</u> [ <i>mean</i>   <u>constant</u> ]</code>	negative binomial; default dispersion is <i>mean</i>
<code><u>ordinal</u></code>	ordinal
<code><u>poisson</u></code>	Poisson

<i>link</i>	Description
<code><u>identity</u></code>	identity
<code><u>log</u></code>	log
<code><u>logit</u></code>	logit
<code><u>probit</u></code>	probit
<code><u>cloglog</u></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)`.

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
binomial			D	x	x
family multinomial			D		
gamma		D			
negative binomial		D			
ordinal			D	x	x
Poisson		D			

D denotes the default.

## Description

`gsem` not only allows models of the form  $y_i = \mathbf{x}_i\beta + u_i$ , it 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.

## 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.

`cloglog`, `gamma`, `logit`, `nbreg`, `mlogit`, `ocloglog`, `ologit`, `oprobit`, `poisson`, `probit`, and `regress` are shorthands for specifying popular models.

`exposure(varnamee)` and `offset(varnameo)` are used only with families `poisson` and `nbreg`, that is, they 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  $\mathbf{x}_i\boldsymbol{\beta}$  with coefficient constrained to be 1.

`offset()` enters `varnameo` into  $\mathbf{x}_i\boldsymbol{\beta}$  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, udepvar(depvar2)
depvar2 <- ... , family(gauss, ldepvar(depvar1)
```

Thus only one of `ldepvar()` or `udepvar()` is allowed. In either case, `depvar1` and `depvar2` should have the following form:

Type of data		<i>var</i> <sub>1</sub>	<i>var</i> <sub>2</sub>
point data	$a = [a, a]$	<i>a</i>	<i>a</i>
interval data	$[a, b]$	<i>a</i>	<i>b</i>
left-censored data	$(-\infty, b]$	.	<i>b</i>
right-censored data	$[a, +\infty)$	<i>a</i>	.

`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()`.

## 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:_cons] = 0
. gsem (y <- x1 x2, gamma), constraints(1)
```

The name `_b[y_logs:_cons]` changes according to the name of the dependent variable. Had `y` instead been named `waitingtime`, the parameter would have been named `_b[waitingtime_logs:_cons]`. 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

## 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