

## Description

This is the first example in the *g* series. The *g* means that the example focuses exclusively on the `gsem` command. If you are interested primarily in standard linear SEMs, you may want to skip the remaining examples. If you are especially interested in generalized SEMs, we suggest you read the remaining examples in order.

`gsem` provides three features not provided by `sem`: the ability to fit SEMs containing generalized linear response variables, the ability to fit multilevel mixed SEMs, and the ability to fit models with categorical latent variables.

Generalized response variables means that the response variables can be specifications from the generalized linear model (GLM). These include probit, logistic regression, ordered probit and logistic regression, multinomial logistic regression, and more. We use generalized linear response variables in this example.

Multilevel mixed models refer to the simultaneous handling of group-level effects, which can be nested or crossed. Thus you can include unobserved and observed effects for subjects, subjects within group, group within subgroup, ..., or for subjects, group, subgroup, .... See [SEM] Example 30g for an example of a multilevel model.

Categorical latent variables are latent variables with categories that correspond to groups in the population. The categories are called classes, and we do not observe which individuals belong to which class. Instead, we estimate the probability of being in each class. `gsem` does not allow categorical latent variables and continuous latent variables (observation level or multilevel) to be present in the same model. See [SEM] Example 50g for an example with a categorical latent variable.

Below we demonstrate a single-factor measure model with pass/fail (binary outcome) responses rather than continuous responses. We use the following data:

```
. use https://www.stata-press.com/data/r19/gsem_1fmm
(Single-factor pass/fail measurement model)
. summarize
```

Variable	Obs	Mean	Std. dev.	Min	Max
x1	123	.4065041	.4931897	0	1
x2	123	.4065041	.4931897	0	1
x3	123	.4227642	.4960191	0	1
x4	123	.3495935	.4787919	0	1
s4	123	690.9837	77.50737	481	885

```
. notes
_dta:
1. Fictional data.
2. The variables x1, x2, x3, and x4 record 1=pass, 0=fail.
3. Pass/fail for x1, x2, x3: score > 100
4. Pass/fail for x4: score > 725
5. Variable s4 contains actual score for test 4.
```

See *Single-factor measurement models* in [SEM] Intro 5 for background.

## Remarks and examples

Remarks are presented under the following headings:

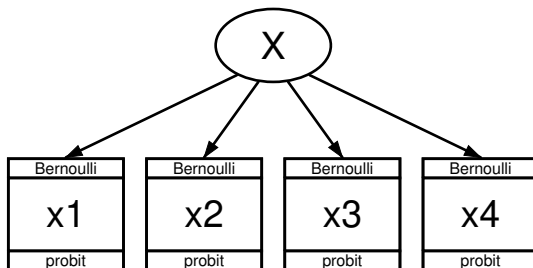
*Single-factor pass/fail measurement model*

*Single-factor pass/fail + continuous measurement model*

*Fitting the model with the Builder*

### Single-factor pass/fail measurement model

Below we fit the following model:



The measurement variables we have ( $x_1, \dots, x_4$ ) are not continuous. They are pass/fail, coded as 1 (pass) and 0 (fail). To account for that, we use probit (also known as family Bernoulli, link probit). The equations for this model are

$$\Pr(x_1 = 1) = \Phi(\alpha_1 + X\beta_1)$$

$$\Pr(x_2 = 1) = \Phi(\alpha_2 + X\beta_2)$$

$$\Pr(x_3 = 1) = \Phi(\alpha_3 + X\beta_3)$$

$$\Pr(x_4 = 1) = \Phi(\alpha_4 + X\beta_4)$$

where  $\Phi(\cdot)$  is the  $N(0, 1)$  cumulative distribution.

One way to think about this is to imagine a test that is scored on a continuous scale. Let's imagine the scores were  $s_1, s_2, s_3$ , and  $s_4$  and distributed  $N(\mu_i, \sigma_i^2)$ , as test scores often are. Let's further imagine that for each test, a cutoff  $c_i$  is chosen and the student passes the test if  $s_i \geq c_i$ .

If we had the test scores, we would fit this as a linear model. We would posit

$$s_i = \gamma_i + X\delta_i + \epsilon_i$$

where  $\epsilon_i \sim N(0, \sigma_i^2)$ . However, we do not have test scores in our data; we have only the pass/fail results  $x_i = 1$  if  $s_i > c_i$ .

So let's consider the pass/fail problem. The probability that a student passes test  $i$  is determined by the probability that the student scores above the cutoff:

$$\begin{aligned}
 \Pr(s_i > c_i) &= \Pr(\gamma_i + X\delta_i + \epsilon_i > c_i) \\
 &= \Pr\{\epsilon_i > c_i - (\gamma_i + X\delta_i)\} \\
 &= \Pr\{-\epsilon_i \leq -c_i + (\gamma_i + X\delta_i)\} \\
 &= \Pr\{-\epsilon_i \leq (\gamma_i - c_i) + X\delta_i\} \\
 &= \Pr\{-\epsilon_i/\sigma_i \leq (\gamma_i - c_i)/\sigma_i + X\delta_i/\sigma_i\} \\
 &= \Phi\{(\gamma_i - c_i)/\sigma_i + X\delta_i/\sigma_i\}
 \end{aligned}$$

The last equation is the probit model. In fact, we just derived the probit model, and now we know the relationship between the parameters we will be able to estimate with our pass/fail data:  $\alpha_i$  and  $\beta_i$ . We also now know the parameters we could have estimated if we had the continuous test scores:  $\gamma_i$  and  $\delta_i$ . The relationship is

$$\begin{aligned}
 \alpha_i &= (\gamma_i - c_i)/\sigma_i \\
 \beta_i &= \delta_i/\sigma_i
 \end{aligned}$$

Notice that the right-hand sides of both equations are divided by  $\sigma_i$ , the standard deviation of the error term from the linear model for the  $i$ th test score. In pass/fail data, we lose the original scale of the score, and the slope coefficient we will be able to estimate is the slope coefficient from the linear model divided by the standard deviation of the error term. Meanwhile, the intercept we will be able to estimate is just the difference of the continuous model's intercept and the cutoff for passing the test, sans scale.

The command to fit the model and the results are

```

. gsem (x1 x2 x3 x4 <-X), probit
Fitting fixed-effects model:
Iteration 0:  Log likelihood = -329.82091
Iteration 1:  Log likelihood = -329.57665
Iteration 2:  Log likelihood = -329.57664
Refining starting values:
Grid node 0:  Log likelihood = -273.75437
Fitting full model:
Iteration 0:  Log likelihood = -273.75437
Iteration 1:  Log likelihood = -264.3035
Iteration 2:  Log likelihood = -263.37815
Iteration 3:  Log likelihood = -262.305
Iteration 4:  Log likelihood = -261.69025
Iteration 5:  Log likelihood = -261.42132
Iteration 6:  Log likelihood = -261.35508
Iteration 7:  Log likelihood = -261.3224
Iteration 8:  Log likelihood = -261.3133
Iteration 9:  Log likelihood = -261.30783
Iteration 10: Log likelihood = -261.30535
Iteration 11: Log likelihood = -261.30405
Iteration 12: Log likelihood = -261.30337
Iteration 13: Log likelihood = -261.30302
Iteration 14: Log likelihood = -261.30283
Iteration 15: Log likelihood = -261.30272
Iteration 16: Log likelihood = -261.30267
Iteration 17: Log likelihood = -261.30264
Iteration 18: Log likelihood = -261.30263

```

```

Generalized structural equation model                                Number of obs = 123
Response: x1
Family:   Bernoulli
Link:     Probit
Response: x2
Family:   Bernoulli
Link:     Probit
Response: x3
Family:   Bernoulli
Link:     Probit
Response: x4
Family:   Bernoulli
Link:     Probit
Log likelihood = -261.30263
( 1)  [x1]X = 1

```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
x1						
X	1 (constrained)					
_cons	-.3666763	.1896773	-1.93	0.053	-.738437	.0050844
x2						
X	1.33293	.4686743	2.84	0.004	.4143455	2.251515
_cons	-.4470271	.2372344	-1.88	0.060	-.911998	.0179438
x3						
X	.6040478	.1908343	3.17	0.002	.2300195	.9780761
_cons	-.2276709	.1439342	-1.58	0.114	-.5097767	.0544349
x4						
X	9.453342	5.151819	1.83	0.067	-.6440375	19.55072
_cons	-4.801027	2.518038	-1.91	0.057	-9.736291	.1342372
var(X)	2.173451	1.044885			.847101	5.576536

#### Notes:

1. In the path diagrams, x1, ..., x4 are shown as being family Bernoulli, link probit. On the command line, we just typed probit although we could have typed family(bernoulli) link(probit). In the command language, probit is a synonym for family(bernoulli) link(probit).
2. Variable X is latent exogenous and thus needs a normalizing constraint. The variable is anchored to the first observed variable, x1, and thus the path coefficient is constrained to be 1. See [Identification 2: Normalization constraints \(anchoring\)](#) in [SEM] [Intro 4](#).
3. The path coefficients for X->x1, X->x2, and X->x3 are 1, 1.33, and 0.60. Meanwhile, the path coefficient for X->x4 is 9.45. This is not unexpected; we at StataCorp generated these fictional data, and we made the x4 effect large and less precisely estimable.

## Single-factor pass/fail + continuous measurement model

In the above example, all equations were probit. Different equations within a single SEM can have a different family and link.

Below we refit our model with the continuous measure for test 4 (variable s4) in place of the pass/fail measure (variable x4). We continue to use the pass/fail measures for tests 1, 2, and 3.

```
. gsem (x1 x2 x3 <-X, probit) (s4<-X)
Fitting fixed-effects model:
Iteration 0:  Log likelihood = -959.23492
Iteration 1:  Log likelihood = -959.09499
Iteration 2:  Log likelihood = -959.09499
Refining starting values:
Grid node 0:  Log likelihood = -905.14944
Fitting full model:
Iteration 0:  Log likelihood = -905.14944 (not concave)
Iteration 1:  Log likelihood = -872.33773
Iteration 2:  Log likelihood = -869.83144
Iteration 3:  Log likelihood = -869.69578
Iteration 4:  Log likelihood = -869.68928
Iteration 5:  Log likelihood = -869.6892
Generalized structural equation model                                Number of obs = 123
Response: x1
Family:    Bernoulli
Link:      Probit
Response: x2
Family:    Bernoulli
Link:      Probit
Response: x3
Family:    Bernoulli
Link:      Probit
Response: s4
Family:    Gaussian
Link:      Identity
Log likelihood = -869.6892
( 1) [x1]X = 1
```

		Coefficient	Std. err.	z	P> z	[95% conf. interval]	
x1	X	1 (constrained)					
	_cons	-.4171085	.1964736	-2.12	0.034	-.8021896	-.0320274
x2	X	1.298311	.3280144	3.96	0.000	.6554142	1.941207
	_cons	-.4926357	.2387179	-2.06	0.039	-.9605142	-.0247573
x3	X	.682969	.1747328	3.91	0.000	.3404989	1.025439
	_cons	-.2942021	.1575014	-1.87	0.062	-.6028992	.0144949
s4	X	55.24829	12.19904	4.53	0.000	31.3386	79.15798
	_cons	690.9837	6.960106	99.28	0.000	677.3422	704.6253
	var(X)	1.854506	.7804393			.812856	4.230998
	var(e.s4)	297.8565	408.64			20.24012	4383.299

Notes:

1. We obtain similar coefficients for  $x_1, \dots, x_3$ .
2. We removed  $x_4$  (a pass/fail variable) and substituted  $s_4$  (the actual test score).  $s_4$  turns out to be more significant than  $x_4$ . This suggests a poor cutoff was set for “passing” test 4.
3. The log-likelihood values for the two models we have fit are strikingly different:  $-261$  in the previous model and  $-870$  in the current model. The difference has no meaning. Log-likelihood values are dependent on the model specified. We changed the fourth equation from a probit specification to a continuous (linear-regression) specification, and just doing that changes the metric of the log-likelihood function. Comparisons of log-likelihood values are only meaningful when they share the same metric.

## Fitting the model with the Builder

Use the diagram in [Single-factor pass/fail measurement model](#) above for reference.


1. Open the dataset.


In the Command window, type

```
. use https://www.stata-press.com/data/r19/gsem_1fmm
```

2. Open a new Builder diagram.

Select menu item **Statistics > SEM (structural equation modeling) > Model building and estimation**.

3. Put the Builder in gsem mode by clicking on the  button.
4. Create the measurement component for X.


Select the Add measurement component tool, , and then click in the diagram about one-third of the way down from the top and slightly left of the center.



In the resulting dialog box,

- a. change the *Latent variable name* to X;
- b. select  $x_1, x_2, x_3$ , and  $x_4$  by using the *Measurement variables* control;
- c. check *Make measurements generalized*;
- d. select `Bernoulli`, `Probit` in the *Family/Link* control;
- e. select `Down` in the *Measurement direction* control;
- f. click on **OK**.

If you wish, move the component by clicking on any variable and dragging it.

5. Estimate.

Click on the **Estimate** button, , in the Standard Toolbar, and then click on **OK** in the resulting *GSEM estimation options* dialog box.

6. To fit the model in *Single-factor pass/fail + continuous measurement model*, modify the diagram created in the previous steps.
  - a. Use the Select tool, , to click on the x4 rectangle.
  - b. In the Contextual Toolbar, select s4 in the *Variable* control.
  - c. In the Contextual Toolbar, select Gaussian, Identity in the *Family/Link* control.
7. Estimate.  
Click on the **Estimate** button, , in the Standard Toolbar, and then click on **OK** in the resulting *GSEM estimation options* dialog box.

You can open a completed diagram in the Builder by typing

```
. webgetsem gsem_1fmm
```

## Also see

[SEM] [Example 1](#) — Single-factor measurement model

[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 50g](#) — Latent class model

[SEM] [gsem](#) — Generalized structural equation model estimation command

[SEM] [Intro 5](#) — Tour of models

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](#).

