

Example 38g — Random-intercept and random-slope models (multilevel)

[Description](#) [Remarks and examples](#) [Reference](#) [Also see](#)

Description

Below we discuss random-intercept and random-slope models in the context of multilevel models, and specifically, 2-level models, although we could just as well use higher-level models (see [SEM] [Example 39g](#)). Some people refer to these models as random-effects models and as mixed-effects models.

To demonstrate random-intercept and random-slope models, we will use the following data:

```
. use https://www.stata-press.com/data/r18/gsem_nlsy
(NLSY 1968)
. describe
Contains data from https://www.stata-press.com/data/r18/gsem_nlsy.dta
Observations:      2,763      NLSY 1968
Variables:         21        29 Mar 2022 11:30
                    (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
idcode	int	%8.0g		NLS ID
year	int	%8.0g		Interview year
birth_yr	byte	%8.0g		Birth year
age	byte	%8.0g		Age in current year
race	byte	%8.0g	racelbl	Race
msp	byte	%8.0g		1 if married, spouse present
nev_mar	byte	%8.0g		1 if never married
grade	byte	%8.0g		Current grade completed
collgrad	byte	%8.0g		1 if college graduate
not_smsa	byte	%8.0g		1 if not SMSA
c_city	byte	%8.0g		1 if central city
south	byte	%8.0g		1 if south
ind_code	byte	%8.0g		Industry of employment
occ_code	byte	%8.0g		Occupation
union	byte	%8.0g		1 if union
wks_ue	byte	%8.0g		Weeks unemployed last year
ttl_exp	float	%9.0g		Total work experience
tenure	float	%9.0g		Job tenure, in years
hours	byte	%8.0g		Usual hours worked
wks_work	int	%8.0g		Weeks worked last year
ln_wage	float	%9.0g		ln(wage/GNP deflator)

Sorted by: idcode year

```
. notes
```

```
_dta:
```

1. Data from National Longitudinal Survey of Young Women 14-27 years of age in 1968 (NLSY), Center for Human Resource Research, Ohio State University, first released in 1989.
2. These data were subsetted for purposes of demonstration.

2 Example 38g — Random-intercept and random-slope models (multilevel)

These 2-level data are recorded in long form, that is, each observation corresponds to a year within a subject and the full set of data is spread across repeated observations.

```
. list id year ln_wage union grade in 1/20, sepby(idcode)
```

	idcode	year	ln_wage	union	grade
1.	1	1970	1.451214	.	12
2.	1	1971	1.02862	.	12
3.	1	1972	1.589977	1	12
4.	1	1973	1.780273	.	12
5.	1	1975	1.777012	.	12
6.	1	1977	1.778681	0	12
7.	1	1978	2.493976	.	12
8.	1	1980	2.551715	1	12
9.	1	1983	2.420261	1	12
10.	1	1985	2.614172	1	12
11.	1	1987	2.536374	1	12
12.	1	1988	2.462927	1	12
13.	2	1971	1.360348	0	12
14.	2	1972	1.206198	.	12
15.	2	1973	1.549883	.	12
16.	2	1975	1.832581	.	12
17.	2	1977	1.726721	1	12
18.	2	1978	1.68991	1	12
19.	2	1980	1.726964	1	12
20.	2	1982	1.808289	1	12

In the repeated observations for a subject, some variables vary (they are at the observation level, such as `ln_wage`) and other variables do not vary (they are at the subject level, such as `grade`).

When using `gsem`, multilevel data must be recorded in the long form except in one case. The exception is latent growth curve models, which can be fit in the long or wide form. In the wide form, there is one physical observation for each subject and multiple variables within subject, such as `ln_wage_1970`, `ln_wage_1971`, and so on. Researchers from a structural equation modeling background think about latent growth models in the wide form; see [SEM] [Example 18](#).

In all other cases, if your data are in the wide form, use Stata's `reshape` command (see [D] [reshape](#)) to convert the data to long form.

See *Structural models 1: Linear regression* and *Multilevel mixed-effects models* in [SEM] [Intro 5](#) for background.

Remarks and examples

[stata.com](http://www.stata.com)

Remarks are presented under the following headings:

- [Random-intercept model, single-equation formulation](#)
- [Random-intercept model, within-and-between formulation](#)
- [Random-slope model, single-equation formulation](#)
- [Random-slope model, within-and-between formulation](#)
- [Fitting the random-intercept model with the Builder](#)
- [Fitting the random-slope model with the Builder](#)

Random-intercept model, single-equation formulation

There are two formulations of the random-intercept model, which we call the single-equation formulation and the within-and-between formulation. Results from both formulations are identical unless you have observations with missing values, in which case the within-and-between formulation will sometimes use more of the data.

We will show you both formulations, but the single-equation formulation makes a good starting point. The model we wish to fit is

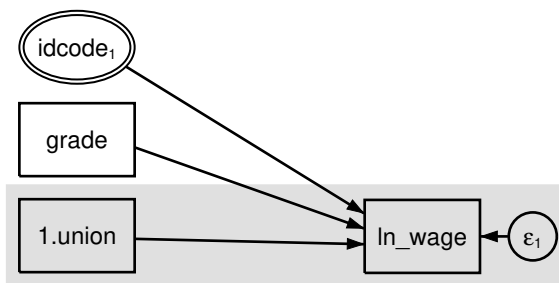


Figure 1

We use factor-variable notation in the diagram above; see [SEM] Example 37g.

We are using multilevel data. `ln_wage` and `union` (union membership) vary at the observation level, while `grade` (school completion) varies at the subject level. We have used shading to emphasize that.

In this model, we are including a random intercept (a random effect) at the subject level. Double-ringed `idcode` is saying, “I am a latent variable at the `idcode` level—meaning I am constant within identification codes and vary across identification codes—and I correspond to a latent variable named M1.” The M1 part of the statement came from the subscript 1; the M part is fixed.

Double-ringed `idcode` indicates a latent variable constant within `idcode`—a random effect. And the fact that the path from the latent variable is pointing to a box and not to another path means that the latent variable is used as a random intercept rather than a random slope. By the way, variable `idcode` in the data contains each subject’s identification number.

Using command syntax, we can fit this model by typing

```
. gsem (ln_wage <- i.union grade M1[idcode])
Fitting fixed-effects model:
Iteration 0: Log likelihood = -925.06629
Iteration 1: Log likelihood = -925.06629
Refining starting values:
Grid node 0: Log likelihood = -763.3769
Fitting full model:
Iteration 0: Log likelihood = -763.3769
Iteration 1: Log likelihood = -622.04625 (backed up)
Iteration 2: Log likelihood = -613.54948
Iteration 3: Log likelihood = -607.56242
Iteration 4: Log likelihood = -607.49246
Iteration 5: Log likelihood = -607.49233
Iteration 6: Log likelihood = -607.49233
```

4 Example 38g — Random-intercept and random-slope models (multilevel)

```

Generalized structural equation model                                Number of obs = 1,904
Response: ln_wage
Family: Gaussian
Link: Identity
Log likelihood = -607.49233
( 1) [ln_wage]M1[idcode] = 1

```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
ln_wage						
1.union	.1637408	.0227254	7.21	0.000	.1191998	.2082818
grade	.0767919	.0067923	11.31	0.000	.0634791	.0901046
M1[idcode]	1 (constrained)					
_cons	.7774129	.0906282	8.58	0.000	.5997848	.955041
var(M1[idcode])	.080247	.0073188			.0671113	.0959537
var(e.ln_w~e)	.078449	.0028627			.0730342	.0842653

Notes:

1. The `ln_wage <- M1[idcode]` coefficient is constrained to be 1. Such constraints are automatically supplied by `gsem` to identify the latent variable. Our model is

$$\begin{aligned}
 \text{ln_wage} &= \dots + \beta_3 \times \text{M1[idcode]} + \text{_cons} \\
 &= \dots + 1 \times \text{M1[idcode]} + 0.7774
 \end{aligned}$$

Thus `M1[idcode]` is being used as a random intercept.

Remember that the square bracketed `[idcode]` means that `M1` is constant within `idcode` and varies only across `idcode`.

2. The variance of our random intercept is estimated to be 0.0802, which is greater than the estimated error variance of 0.0784.
3. Although it is obvious in this case that the latent variable (random intercept) has sufficient variance that it cannot be ignored, we can test whether the variance is large enough that we could not ignore it. The test will be up against a boundary (variances cannot be less than 0), and so the test will be conservative. To perform the test, we would type

```

. gsem (ln_wage <- i.union grade) // fit model w/o random intercept
. estimates store without
. gsem (ln_wage <- i.union grade M1[idcode]) // fit model w/ random intercept
. estimates store with
. lrtest without with // order of arguments unimportant

```

Random-intercept model, within-and-between formulation

The other way to write the random-intercept model is

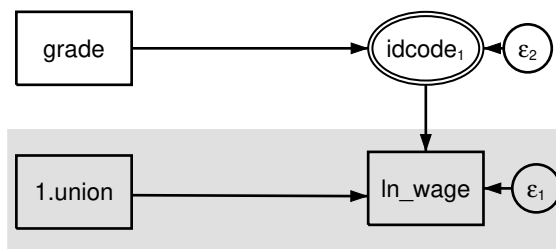


Figure 2

Do not read `grade` pointing to double-ringed `idcode` as `grade` being a predictor of `idcode`. That would make no sense. Double rings indicate a latent variable, and `grade` is a predictor of a latent variable. In particular, the subscript 1 on `idcode` indicates that the latent variable is named `M1`. Thus `grade` is a predictor of `M1`. The `idcode` inside the double rings says that `M1` is constant within `idcode`. Thus `grade`, which itself does not vary within `idcode`, is a predictor of `M1`, which does not vary within `idcode`; said more elegantly, `grade` is a predictor of `M1` at the subject level.

It is logically required that `grade` vary at the same or higher level as `M1[idcode]`, and `gsem` will check that requirement for you.

In this model, `M1[idcode]` contains both the random intercept and the `grade` effect. There is now an equation for `M1[idcode]`, with an error associated with it, and it will be the variance of the error term that will reflect the variance of the random intercept.

To fit this within-and-between formulation of our model, we type

```
. gsem (ln_wage <- i.union M1[idcode]) (M1[idcode] <- grade)
Fitting fixed-effects model:
Iteration 0:  Log likelihood = -1091.655
Iteration 1:  Log likelihood = -1091.655
Refining starting values:
Grid node 0:  Log likelihood = -886.50236
Fitting full model:
Iteration 0:  Log likelihood = -886.50236 (not concave)
Iteration 1:  Log likelihood = -683.7337
Iteration 2:  Log likelihood = -630.91327
Iteration 3:  Log likelihood = -608.14536
Iteration 4:  Log likelihood = -607.49376
Iteration 5:  Log likelihood = -607.49233
Iteration 6:  Log likelihood = -607.49233
Generalized structural equation model                               Number of obs = 1,904
Response: ln_wage
Family:  Gaussian
Link:    Identity
Log likelihood = -607.49233
( 1) [ln_wage]M1[idcode] = 1
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
ln_wage						
1.union	.1637408	.0227254	7.21	0.000	.1191998	.2082818
M1[idcode]	1 (constrained)					
_cons	.7774129	.0906282	8.58	0.000	.5997848	.955041
M1[idcode]						
grade	.0767919	.0067923	11.31	0.000	.0634791	.0901046
var(e.M1[idcode])	.080247	.0073188			.0671113	.0959537
var(e.ln_w~e)	.078449	.0028627			.0730342	.0842653

Notes:

1. Results are identical to what we [previously](#) obtained.
2. The within-and-between formulation is equivalent to the single-equation formulation if there are no missing values in the data.
3. In this simple model, the two formulations are also equivalent even in the presence of missing values.
4. If M1[idcode] were also being used to predict another endogenous variable, then missing values in grade would only cause the equation for the other endogenous variable to have to omit those observations in the within-and-between formulation.

Random-slope model, single-equation formulation

Let us now turn to random slopes. Because it is generally—not always—a good idea to include random intercepts with random slopes, we are going to include both. In addition, we are going to complicate our model by adding an interaction term between union membership and grade so we can make another point that we will not explain until the next section.

The model we wish to fit is

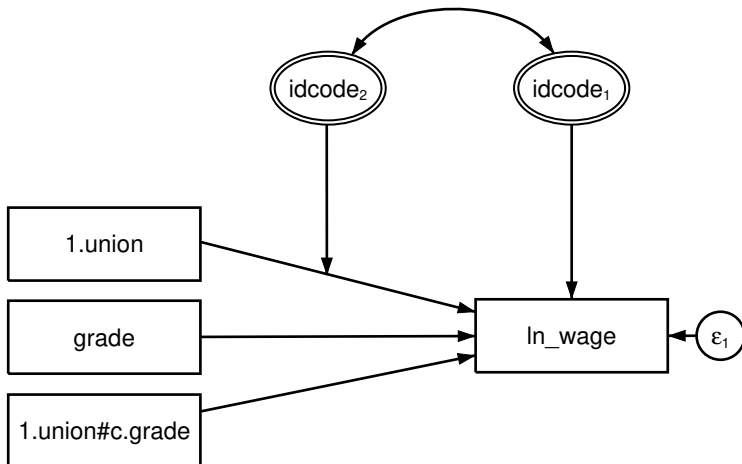


Figure 3. Demonstrated formulation 1

Ignore the interaction between union membership and grade. We could have omitted it from this example, but we want to show later the strikingly different way that interaction can be written in the within-and-between formulation.

In this model, we have double-ringed `idcode`, just as we did in a [previous section](#) introducing a random intercept into the model. This time, however, we also have a second double-ringed `idcode` adding a random slope. When a path from a latent variable points to another path, it is specifying a random slope. When specifying a random slope for a variable, you still include the variable in the usual, fixed-slope way, and then you add the extra component.

The result of the extra component will be to add an interaction into the model. `1.union` will affect `ln_wage`, just as we have drawn, and the extra component will add `1.union × M2[idcode]`. That is, the effect of `1.union` in our model will be

$$\begin{aligned}
 \text{ln_wage} &= \dots + \beta_3 \times 1.\text{union} + \beta_4 \times 1.\text{union} \times \text{M2}[\text{idcode}] + \dots \\
 &= \dots + \beta_3 \times 1.\text{union} + \beta_4 \times \text{M2}[\text{idcode}] \times 1.\text{union} + \dots \\
 &= \dots + (\beta_3 + \beta_4 \times \text{M2}[\text{idcode}]) \times 1.\text{union} + \dots
 \end{aligned}$$

The fixed-plus-random, total slope is $\beta_3 + \beta_4 \times \text{M2}[\text{idcode}]$. Latent variable `M2` has an arbitrary scale, and therefore coefficient β_4 cannot be identified. So constraining $\beta_4 = 1$ results in the total slope being $\beta_3 + \text{M2}[\text{idcode}]$, which is exactly how you would expect a fixed-plus-random slope to look. In other words, drawing the path from `idcode2` to the path with coefficient β_3 indicates that `M2[idcode]` is added to β_3 .

To fit this model in the command language, we type

```
. gsem (ln_wage <- i.union grade i.union#c.grade M1[idcode] 1.union#M2[idcode])
Fitting fixed-effects model:
Iteration 0:  Log likelihood = -925.06629
Iteration 1:  Log likelihood = -925.06629
Refining starting values:
Grid node 0:  Log likelihood = -869.92256
Fitting full model:
Iteration 0:  Log likelihood = -869.92256 (not concave)
Iteration 1:  Log likelihood = -727.15806 (not concave)
Iteration 2:  Log likelihood = -711.74718 (not concave)
Iteration 3:  Log likelihood = -684.33867 (not concave)
Iteration 4:  Log likelihood = -665.94123 (not concave)
Iteration 5:  Log likelihood = -610.14526
Iteration 6:  Log likelihood = -589.89989
Iteration 7:  Log likelihood = -582.24119
Iteration 8:  Log likelihood = -581.298
Iteration 9:  Log likelihood = -581.29004
Iteration 10: Log likelihood = -581.29003
Generalized structural equation model                               Number of obs = 1,904
Response: ln_wage
Family: Gaussian
Link: Identity
Log likelihood = -581.29003
( 1) [ln_wage]M1[idcode] = 1
( 2) [ln_wage]1.union#M2[idcode] = 1
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
ln_wage						
1.union	.1199049	.1508189	0.80	0.427	-.1756946	.4155045
grade	.0757883	.0081803	9.26	0.000	.0597552	.0918215
union#						
c.grade						
1	.0019983	.0113534	0.18	0.860	-.020254	.0242506
M1[idcode]	1 (constrained)					
union#						
M2[idcode]						
1	1 (constrained)					
_cons	.7873884	.1086476	7.25	0.000	.574443	1.000334
var(
M1[idcode])	.0927931	.0088245			.0770136	.1118056
var(
M2[idcode])	.0823065	.018622			.052826	.1282392
cov(
M1[idcode],						
M2[idcode])	-.0549821	.0116103	-4.74	0.000	-.077738	-.0322263
var(e.ln_w~e)	.0720873	.0027135			.0669603	.0776068

Notes:

1. `M1[idcode]` is the random intercept. The coefficient on it is constrained to be 1, just as [previously](#) and just as we would expect.
2. The coefficient on `1.union` is the fixed part of the slope of `union`.
3. `M2[idcode]` is the random part of the slope of `union`.

The coefficient on `1.union#M2[idcode]` is constrained to be 1, just as we would expect. `1.union#M2[idcode]` is the way Stata writes $1.union \times M2[idcode]$.

4. There is an unexpected term in the output, `0.union#M2[idcode]`, shown with coefficient 0. The first thing to remember about unexpected terms is that they are irrelevant if their coefficients are 0. `gsem` reports the coefficient as being 0 (omitted), which is `gsem`'s way of saying, "Here is a line that I did not even include in the model." There are a lot of terms `gsem` could tell us about that were not included in the model, so why did `gsem` feel obligated to tell us about this term? The term has to do with how Stata tracks base levels of factor variables.

There is a setting—`set showbaselevels off`—that will prevent lines like that from being displayed. There is also a setting—`set showbaselevels all`—that will show even more of them! The default is `set showbaselevels on`.

5. We specified the interaction as `1.union#M2[idcode]` rather than `i.union#M2[idcode]`. Even so, using `#.union#M2[idcode]` or `i.union#M2[idcode]` makes no difference because `union` takes on two values. If `union` took on three values, however, think about how we would diagram the model. We would have two latent variables, and we would want `1.union#M2[idcode]` and `2.union#M3[idcode]`. If `union` took on three or more values, typing `i.union#M2[idcode]` simply would not produce the desired result.

Random-slope model, within-and-between formulation

To properly demonstrate the within-and-between formulation, we need a more complicated model; we made one in the [previous section](#) when we included `1.union#c.grade` and then told you to ignore it.

Pretend that we had omitted `1.union#c.grade` from the random-slope, single-equation formulation model. It would have looked like this:

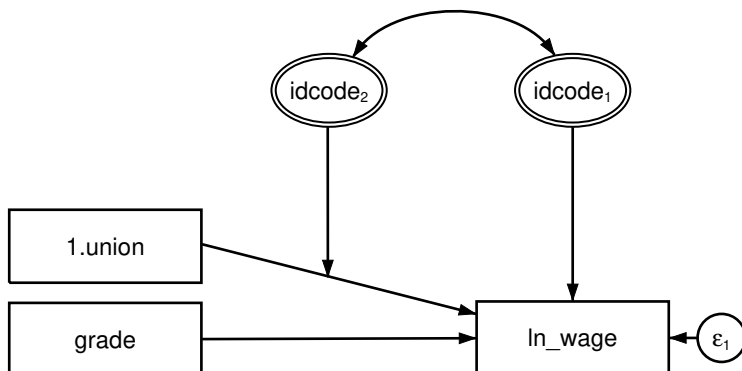


Figure 4. Simple formulation 1

Let's call this model Simple formulation 1 as compared with the model that we fit in the [previous section](#), which we will call [Demonstrated formulation 1](#). We could have fit Simple formulation 1 by typing

```
. gsem (ln_wage <- i.union grade M1[idcode] 1.union#M2[idcode])
```

The corresponding within-and-between model, which we will call Simple formulation 2, would look something like this:

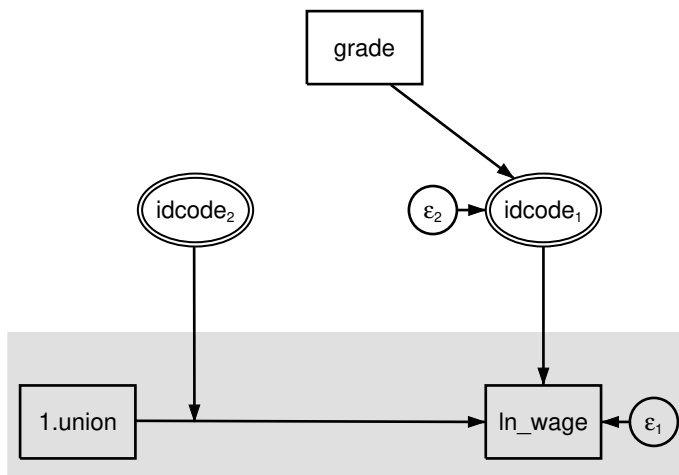


Figure 5. Simple formulation 2

We say that it would look something like the above because Simple formulations 1 and 2 are not identical models. In particular, there is nothing in Simple formulation 2 that corresponds to the correlation between `M1[idcode]` and `M2[idcode]` in Simple formulation 1. Other than that, the models are identical. In Simple formulation 1, `grade` directly affects `ln_wage`. In Simple formulation 2, `grade` affects `M1[idcode]` and `M1[idcode]` affects `ln_wage`. The results are the same either way. Many researchers prefer Simple formulation 2 because it visually separates the within (subject) and between (subject) levels.

But as we said, the two formulations are not identical. They would be identical if we added a curved covariance path between ϵ_2 and `idcode2`, but Stata will not allow us to do that.

On the other hand, if you wanted to fit the simple model without the correlation, you could do it in either formulation. For Simple formulation 1, you would type

```
. gsem (ln_wage <- i.union grade M1[idcode] 1.union#M2[idcode]), ///
      cov(M1[idcode]*M2[idcode]@0)
```

For Simple formulation 2, you would type

```
. gsem (ln_wage <- i.union M1[idcode] 1.union#M2[idcode]) (M1[idcode] <- grade)
```

Either way, results would be identical.

In more complicated models, you can correlate the random intercepts and slopes even in a second formulation. Consider Demonstrated formulation 2:

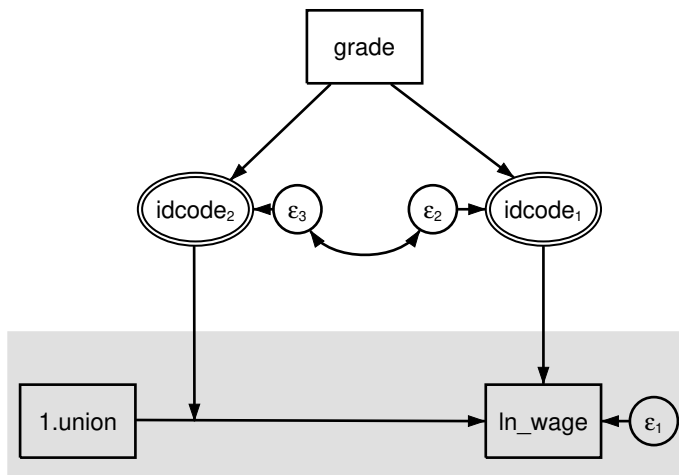


Figure 6. Demonstrated formulation 2

In this model, we allow correlation between ϵ_3 and ϵ_2 —which Stata does allow—and that is the same as allowing correlation between $M1[idcode]$ and $M2[idcode]$ in the formulation-1 model. The general rule is that formulation 2 can be used to correlate the random intercepts and random slopes when both $M1$ and $M2$ are (latent) exogenous or (latent) endogenous.

Demonstrated formulation 2 corresponds to [Demonstrated formulation 1](#). In Demonstrated formulation 1, we included `ln_wage <- i.union#c.grade`. Do you see how that term is implied by the path diagram above? Variable `grade` affects $M2[idcode]$, which in turn affects the slope of `1.union`. Affecting the slope is the same as an interaction.

And now you know why we complicated Demonstrated formulation 1. We did that so that we can show you Demonstrated formulation 2:

```
. gsem (ln_wage <- i.union M1[idcode] 1.union#M2[idcode])
>      (M1[idcode] M2[idcode] <- grade), cov(e.M1[idcode]*e.M2[idcode])

Fitting fixed-effects model:
Iteration 0:  Log likelihood = -1091.655
Iteration 1:  Log likelihood = -1091.655

Refining starting values:
Grid node 0:  Log likelihood = -982.18261

Fitting full model:
Iteration 0:  Log likelihood = -982.18261 (not concave)
Iteration 1:  Log likelihood = -690.60714 (not concave)
Iteration 2:  Log likelihood = -677.13819 (not concave)
Iteration 3:  Log likelihood = -661.26658
Iteration 4:  Log likelihood = -615.66049
Iteration 5:  Log likelihood = -585.69402
Iteration 6:  Log likelihood = -581.34158
Iteration 7:  Log likelihood = -581.29005
Iteration 8:  Log likelihood = -581.29003
```

```

Generalized structural equation model                Number of obs = 1,904
Response: ln_wage
Family: Gaussian
Link: Identity
Log likelihood = -581.29003
( 1) [ln_wage]M1[idcode] = 1
( 2) [ln_wage]1.union#M2[idcode] = 1

```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
ln_wage						
1.union	.119905	.1508188	0.80	0.427	-.1756945	.4155045
M1[idcode]	1 (constrained)					
union#						
M2[idcode]	1 (constrained)					
1						
_cons	.7873884	.1086476	7.25	0.000	.5744431	1.000334
M1[idcode]						
grade	.0757883	.0081803	9.26	0.000	.0597552	.0918215
M2[idcode]						
grade	.0019983	.0113534	0.18	0.860	-.020254	.0242506
var(
e.M1[idcode])	.092793	.0088245			.0770136	.1118055
var(
e.M2[idcode])	.0823065	.018622			.052826	.1282392
cov(
e.M1[idcode],						
e.M2[idcode])	-.0549822	.0116103	-4.74	0.000	-.077738	-.0322263
var(e.ln_w~e)	.0720873	.0027135			.0669603	.0776068

Note:

1. Results of the above are identical to the results of [Demonstrated formulation 1](#).

Fitting the random-intercept model with the Builder

Use the diagram in [Random-intercept model, single-equation formulation](#) above for reference.

1. Open the dataset.


In the Command window, type


```
. use https://www.stata-press.com/data/r18/gsem_nlsy
```

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 endogenous variable.
 - a. Select the Add observed variable tool, , and then click in the diagram about one-third of the way in from the right and one-third of the way up from the bottom. After adding it, you can click inside the rectangle and move the variable if you wish.
 - b. In the Contextual Toolbar, select `ln_wage` with the *Variable* control.
5. Create the observed exogenous variables.


Select the Add observed variables set tool, , and then click in the diagram about one-third of the way in from the left and one-third of the way up from the bottom.

In the resulting dialog box,

- a. select the *Select variables* radio button (it may already be selected);
 - b. use the *Variables* control and select `grade`;
 - c. type `1.union` in the *Variables* control after `grade` (typing `1.union` rather than using the  button to create `i.union` prevents the rectangle corresponding to the base category for this binary variable from being created);
 - d. select *Vertical* in the *Orientation* control;
 - e. click on **OK**.
- If you wish, move the set of variables by clicking on any variable and dragging it.
6. Create the multilevel latent variable corresponding to the random intercept.
 - a. Select the Add multilevel latent variable tool, , and click above the rectangle for `grade`.
 - b. In the Contextual toolbar, click on the  button.
 - c. Select the nesting level and nesting variable by selecting 2 from the *Nesting depth* control and selecting `idcode > Observations` in the next control.
 - d. Specify `M1` as the *Base name*.
 - e. Click on **OK**.
 7. Create the paths from the exogenous variables to `ln_wage`.
 - a. Select the Add path tool, .
 - b. Click in the right side of the `1.union` rectangle (it will highlight when you hover over it), and drag a path to the left side of the `ln_wage` rectangle (it will highlight when you can release to connect the path).
 - c. Continuing with the  tool, draw paths from the right side of the `grade` rectangle to the left side of the `ln_wage` rectangle and from the right side of the `idcode1` double oval to the left side of the `ln_wage` rectangle.
 8. Clean up the location of the paths.

If you do not like where the paths have been connected to the rectangles or oval, use the Select tool, , to click on the path, and then simply click on where it connects to a rectangle or oval and drag the endpoint.

9. 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_rint
```

Fitting the random-slope model with the Builder

Use the diagram in *Random-slope model, single-equation formulation* above for reference.

1. Open the dataset.

In the Command window, type

```
. use https://www.stata-press.com/data/r18/gsem_nlsy
```

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. Increase the width of the observed variable rectangles to accommodate the length of the name of the interaction term.

From the SEM Builder menu, select **Settings > Variables > All observed...**


In the resulting dialog box, change the first size to 1 and click on **OK**.

5. Create the endogenous variable.

a. Select the Add observed variable tool, , and then click in the diagram about one-third of the way in from the right and one-third of the way up from the bottom. After adding it, you can click inside the rectangle and move the variable if you wish.


b. In the Contextual Toolbar, select `ln_wage` with the *Variable* control.

6. Create the observed exogenous variables.

Select the Add observed variables Set tool, , and then click in the diagram about one-third of the way in from the left and one-third of the way up from the bottom.

In the resulting dialog box,

a. select the *Select variables* radio button (it may already be selected);

b. type `1.union` in the *Variables* control (typing `1.union` rather than using the  button to create `i.union` prevents the rectangle corresponding to the base category for this binary variable from being created);









c. use the *Variables* control and select `grade`;


d. type `1.union#c.grade` in the *Variables* control after `grade`;

e. select `Vertical` in the *Orientation* control;


f. click on **OK**.

If you wish, move the set of variables by clicking on any variable and dragging it.

6. Create the multilevel latent variable corresponding to the random intercept.
 - a. Select the Add multilevel latent variable tool, , and click above the rectangle for `ln_wage`.
 - b. In the Contextual Toolbar, click on the  button.
 - c. Select the nesting level and nesting variable by selecting 2 from the *Nesting depth* control and selecting `idcode > Observations` in the next control.
 - d. Specify M1 as the *Base name*.
 - e. Click on **OK**.
7. Create the paths from the exogenous variables to `ln_wage`.
 - a. Select the Add path tool, .
 - b. Click in the right side of the `1.union` rectangle (it will highlight when you hover over it), and drag a path to the left side of the `ln_wage` rectangle (it will highlight when you can release to connect the path).
 - c. Continuing with the  tool, draw paths from the right sides of the `grade` and `1.union#c.grade` rectangles to the left side of the `ln_wage` rectangle and from the bottom of the `idcode1` double oval to the top of the `ln_wage` rectangle.
8. Create the random slope.
 - a. Select the Add multilevel latent variable tool, , and click above the path from `1.union` to `ln_wage`.
 - b. In the Contextual Toolbar, click on the  button.
 - c. Select the nesting level and nesting variable by selecting 2 from the *Nesting depth* control and selecting `idcode > Observations` in the next control.
 - d. Specify M2 as the *Base name*.
 - e. Click on **OK**.
 - f. Select the Add path tool, .
 - g. Click in the bottom of the `idcode2` double oval, and drag a path to the path between `1.union` and `ln_wage`.
9. Create the covariance between the random slope and random intercept.
 - a. Select the Add covariance tool, .
 - b. Click in the top-right quadrant of the `idcode2` double oval, and drag a covariance to the top left of the `idcode1` double oval.
10. Clean up paths and covariance.

If you do not like where a path has been connected to its variables, use the Select tool, , to click on the path, and then simply click on where it connects to a rectangle and drag the endpoint. Similarly, you can change where the covariance connects to the latent variables by clicking on the covariance and dragging the endpoint. You can also change the bow of the covariance by clicking on the covariance and dragging the control point that extends from one end of the selected covariance.

11. 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_rslope
```

Reference

Center for Human Resource Research. 1989. *National Longitudinal Survey of Labor Market Experience, Young Women 14–24 years of age in 1968*. Columbus, OH: Ohio State University Press.

Also see

[SEM] **Example 39g** — Three-level model (multilevel, generalized response)

[SEM] **Example 42g** — One- and two-level mediation models (multilevel)

[SEM] **Intro 5** — Tour of models

[SEM] **gsem** — Generalized structural equation model estimation command

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 © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.



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