Title

stata.com

Example 52g — Latent profile model

Description Remarks and examples References Also see

Description

To demonstrate latent profile models, we use the following data:

```
. use https://www.stata-press.com/data/r17/gsem_lca2
(Latent profile analysis)
. describe
Contains data from https://www.stata-press.com/data/r17/gsem_lca2.dta
 Observations:           145                  Latent profile analysis
    Variables:             7                  18 Jan 2021 12:39
                                              (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
patient	int	%9.0g		Patient ID
relwgt	float	%9.0g		Relative weight
fglucose	int	%9.0g		Fasting plasma glucose
glucose	float	%9.0g		Glucose area (mg/10mL/hr)
insulin	float	%9.0g		Insulin area (mIU/10mL/hr)
sspg	float	%9.0g		Steady-state plasma glucose
cclass	byte	%17.0g	class	Clinical classification

```
Sorted by:
. notes
_dta:
  1.  Source: Data originally analyzed in Reaven, G. M., and R. G. Miller.
      1979. An attempt to define the nature of chemical diabetes using a
      multidimensional analysis. Diabetologia 16: 17-24.
      https://doi.org/10.1007/BF00423145.
  2.  Data made publicly available in Andrews, D. F., and A. M. Herzberg. 1985.
      Data: A Collection of Problems from Many Fields for the Student and
      Research Worker. New York: Springer.
  3.  Data includes variables related to diabetes for 145 nonobese adults.
```

See *Latent class models* in [SEM] **Intro 5** for background.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Fitting the two-class model

In this manual, when we talk about latent class analysis, we are referring to an analysis that involves fitting models with categorical latent variables. Sometimes, these models are given more specific names. In [SEM] **Example 50g**, we fit a latent class model with a categorical latent variable and categorical observed variables. This is a typical latent class model. However, models with categorical latent variables are not limited to having categorical observed variables. A latent class model that instead has continuous observed variables is often referred to as a latent profile model.

Masyn (2013) uses the data described above to fit a series of latent profile models, each having one categorical latent variable and three observed variables, `glucose`, `insulin`, and `sspg`. The goal is to determine categories of diabetes based on these three variables. We begin by fitting a model in which the latent variable, `C`, has two classes. We fit a linear regression model for each observed variable where the intercept, α_{jc}, is allowed to vary across the classes of the latent variable. Because we are using linear regression, we also estimate the variances of the error terms `e.glucose`, `e.insulin`, and `e.sspg`.

More specifically, for class 1 we fit

$$\texttt{glucose} = \alpha_{11} + \texttt{e.glucose}$$
$$\texttt{insulin} = \alpha_{21} + \texttt{e.insulin}$$
$$\texttt{sspg} = \alpha_{31} + \texttt{e.sspg}$$

and for class 2 we fit

$$\texttt{glucose} = \alpha_{12} + \texttt{e.glucose}$$
$$\texttt{insulin} = \alpha_{22} + \texttt{e.insulin}$$
$$\texttt{sspg} = \alpha_{32} + \texttt{e.sspg}$$

We also estimate the probability of being in each class using multinomial logistic regression,

$$\Pr(C = 1) = \frac{e^{\gamma_1}}{e^{\gamma_1} + e^{\gamma_2}}$$
$$\Pr(C = 2) = \frac{e^{\gamma_2}}{e^{\gamma_1} + e^{\gamma_2}}$$

where γ_1 and γ_2 are intercepts in the multinomial logit model. By default, the first class will be treated as the base, so $\gamma_1 = 0$.

We will assume that the errors are uncorrelated, which is the default, and that the variances do not differ across classes, also the default.

```
. gsem (glucose insulin sspg <- _cons), lclass(C 2)
```

(*iteration log omitted*)

```
Generalized structural equation model                     Number of obs = 145
Log likelihood = -1702.5542

 ( 1)  [/]var(e.glucose)#1bn.C - [/]var(e.glucose)#2.C = 0
 ( 2)  [/]var(e.insulin)#1bn.C - [/]var(e.insulin)#2.C = 0
 ( 3)  [/]var(e.sspg)#1bn.C - [/]var(e.sspg)#2.C = 0
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
1.C	(base outcome)					
2.C						
_cons	-1.541025	.2205682	-6.99	0.000	-1.973331	-1.10872

```
Class:    1

Response: glucose
Family:   Gaussian
Link:     Identity

Response: insulin
Family:   Gaussian
Link:     Identity

Response: sspg
Family:   Gaussian
Link:     Identity
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
glucose						
_cons	41.22237	1.298051	31.76	0.000	38.67824	43.7665
insulin						
_cons	20.98005	1.000974	20.96	0.000	19.01817	22.94192
sspg						
_cons	14.96579	.6868081	21.79	0.000	13.61967	16.31191
var(e.gluc~e)	191.5596	23.83815			150.0992	244.4723
var(e.insu~n)	119.0542	14.00336			94.54204	149.9217
var(e.sspg)	55.91283	6.713667			44.18801	70.7487

```
Class:    2
Response: glucose
Family:   Gaussian
Link:     Identity
Response: insulin
Family:   Gaussian
Link:     Identity
Response: sspg
Family:   Gaussian
Link:     Identity
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
glucose						
_cons	115.7123	2.849914	40.60	0.000	110.1266	121.2981
insulin						
_cons	7.553144	2.160949	3.50	0.000	3.317761	11.78853
sspg						
_cons	34.5529	1.53117	22.57	0.000	31.55187	37.55394
var(e.gluc~e)	191.5596	23.83815			150.0992	244.4723
var(e.insu~n)	119.0542	14.00336			94.54204	149.9217
var(e.sspg)	55.91283	6.713667			44.18801	70.7487

```
. estimates store c2inv
```

Notes:

1. The first table in the output provides the estimated coefficients in the multinomial logit model for `C`.
2. The next two tables are the results for the linear regression models for the first and second classes.

Comparing models

Before we interpret any results, we will fit and compare other models. We modify our command above to specify that `C` has three, four, and then five latent classes, and we store the results of those models by typing

```
. gsem (glucose insulin sspg <- _cons), lclass(C 3)
. estimates store c3inv
. gsem (glucose insulin sspg <- _cons), lclass(C 4) ///
  startvalues(randomid, draws(5) seed(15)) emopts(iter(20))
. estimates store c4inv
. gsem (glucose insulin sspg <- _cons), lclass(C 5) ///
  startvalues(randomid, draws(5) seed(15)) emopts(iter(20))
. estimates store c5inv
```

For the models with four and five latent classes, we added the `startvalues(randomid), draws(5) seed(15))` option to request that starting values be computed using random class assignments. In this option, `draws(5)` specifies that five random draws be taken and that the one with the best log likelihood after the EM iterations be selected. The `emopts(iter(20))` option says that 20 EM iterations are used for each random draw. We also set the seed for reproducible results. We could have used the same options in the models with two classes and three classes. Difficulty finding good starting values is fairly common when fitting latent class models, so `gsem` provides a variety

of options for obtaining starting values. See [SEM] **Intro 12** and [SEM] **gsem estimation options** for more information on starting values.

We can compare the four models fit above using Akaike's information criterion (AIC) and Schwarz's Bayesian information criterion (BIC).

```
. estimates stats c2inv c3inv c4inv c5inv
Akaike's information criterion and Bayesian information criterion
```

Model	N	ll(null)	ll(model)	df	AIC	BIC
c2inv	145	.	-1702.554	10	3425.108	3454.876
c3inv	145	.	-1653.238	14	3334.476	3376.15
c4inv	145	.	-1626.828	18	3289.656	3343.237
c5inv	145	.	-1578.207	22	3200.414	3265.902

```
Note: BIC uses N = number of observations. See [R] BIC note.
```

The model with five latent classes has the smallest values of both AIC and BIC and would be considered the best based on these information criteria.

Fitting the three-class model with covariances

Masyn's final model was a three-class model that allowed for covariances among the error terms and that estimated all parameters separately across classes. To estimate the covariances, we add the `covstructure(e._OEn, unstructured)` option. See [SEM] **sem and gsem option covstructure()** for details on this option. To allow all parameters to vary across classes, we add the `lcinvariant(none)` option. Here `none` specifies that no parameters are constrained to be equal across classes.

```
. gsem (glucose insulin sspg <- _cons), lclass(C 3) lcinvariant(none)
> covstructure(e._OEn, unstructured)
```

(*iteration log omitted*)

```
Generalized structural equation model                    Number of obs = 145
Log likelihood = -1536.6409
```

		Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
1.C		(base outcome)					
2.C	_cons	-.8853513	.2386536	-3.71	0.000	-1.353104	-.4175988
3.C	_cons	-.612664	.2260018	-2.71	0.007	-1.055619	-.1697085

```
Class:    1

Response: glucose
Family:   Gaussian
Link:     Identity

Response: insulin
Family:   Gaussian
Link:     Identity

Response: sspg
Family:   Gaussian
Link:     Identity
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
glucose						
_cons	35.68584	.5741752	62.15	0.000	34.56048	36.81121
insulin						
_cons	16.58066	.6204724	26.72	0.000	15.36456	17.79677
sspg						
_cons	10.49755	.5833606	17.99	0.000	9.354183	11.64091
var(e.gluc~e)	19.30952	3.932547			12.9544	28.78233
var(e.insu~n)	26.7354	4.494093			19.23108	37.16804
var(e.sspg)	18.71079	3.970509			12.34422	28.36094
cov(e.gluc~e, e.insulin)	3.456027	2.942391	1.17	0.240	-2.310954	9.223008
cov(e.gluc~e, e.sspg)	5.474303	2.811729	1.95	0.052	-.0365846	10.98519
cov(e.insu~n, e.sspg)	7.995803	3.020304	2.65	0.008	2.076115	13.91549

```
Class:    2

Response: glucose
Family:   Gaussian
Link:     Identity

Response: insulin
Family:   Gaussian
Link:     Identity

Response: sspg
Family:   Gaussian
Link:     Identity
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
glucose						
_cons	47.66176	1.492718	31.93	0.000	44.73609	50.58744
insulin						
_cons	34.35203	3.00337	11.44	0.000	28.46554	40.23853
sspg						
_cons	24.414	.7395383	33.01	0.000	22.96453	25.86347
var(e.gluc~e)	53.21326	15.56547			29.99396	94.40735
var(e.insu~n)	228.6332	59.03553			137.832	379.2526
var(e.sspg)	13.75515	3.838523			7.960284	23.76853
cov(e.gluc~e, e.insulin)	40.02875	23.12762	1.73	0.083	-5.300552	85.35805
cov(e.gluc~e, e.sspg)	.7294854	5.48065	0.13	0.894	-10.01239	11.47136
cov(e.insu~n, e.sspg)	-5.743169	11.4943	-0.50	0.617	-28.27158	16.78524

```
Class:    3

Response: glucose
Family:   Gaussian
Link:     Identity

Response: insulin
Family:   Gaussian
Link:     Identity

Response: sspg
Family:   Gaussian
Link:     Identity
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
glucose						
_cons	93.92473	6.985336	13.45	0.000	80.23372	107.6157
insulin						
_cons	10.37614	1.123135	9.24	0.000	8.174836	12.57744
sspg						
_cons	28.4787	1.94975	14.61	0.000	24.65726	32.30013
var(e.gluc~e)	1279.011	312.6774			792.1048	2065.218
var(e.insu~n)	36.38521	9.26287			22.09163	59.92692
var(e.sspg)	113.3239	27.67628			70.21642	182.8961
cov(e.gluc~e, e.insulin)	-163.4383	47.637	-3.43	0.001	-256.8051	-70.07153
cov(e.gluc~e, e.sspg)	276.9206	81.60543	3.39	0.001	116.9769	436.8643
cov(e.insu~n, e.sspg)	-25.4313	11.66564	-2.18	0.029	-48.29554	-2.567057

Because we do not have any predictors in our regression models, the intercepts can be interpreted as the predicted class-specific means of the corresponding variables. In class 1, `glucose` has an estimated mean of 35.69, `insulin` has an estimated mean of 16.58, and `sspg` has an estimated mean of 10.50. Also because we have no predictors, the estimated variances and covariances of the error terms are simply class-specific estimates of the variances and covariances of the variables. In class 1, the estimated variance of `glucose` is 19.31, the estimated covariance of `glucose` and `insulin` is 3.46. The remaining coefficients can be interpreted in a similar manner.

We can determine expected classification for each individual in the dataset based on the predicted posterior class probabilities.

```
. predict cpost*, classposteriorpr

. egen max = rowmax(cpost*)

. generate predclass = 1 if cpost1==max
(69 missing values generated)

. replace predclass = 2 if cpost2==max
(32 real changes made)

. replace predclass = 3 if cpost3==max
(37 real changes made)
```

```
. tabulate cclass predclass, col
```

Key
frequency *column percentage*

Clinical classification	predclass 1	2	3	Total
Overt diabetic	0 0.00	2 6.25	31 83.78	33 22.76
Chemical diabetic	7 9.21	23 71.88	6 16.22	36 24.83
Normal	69 90.79	7 21.88	0 0.00	76 52.41
Total	76 100.00	32 100.00	37 100.00	145 100.00

When we compare the predicted classes (`predclass`) with the assigned clinical classifications (`cclass`) given to these individuals, we see that 91% of the individuals predicted to be in class 1 were given a clinical classification of normal. Of those predicted to be in class 2, 72% were assigned a clinical classification of chemical diabetic. Finally, 84% of those predicted to be in class 3 had a clinical classification of overt diabetic.

Masyn went on to examine the individuals who were classified differently when using the clinical definition and when using the results from the model. She found that the predictions from the latent profile model could be explained medically and may be an improvement over the clinical definitions.

References

Andrews, D. F., and A. M. Herzberg, ed. 1985. *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. New York: Springer.

Masyn, K. E. 2013. Latent class analysis and finite mixture modeling. In *The Oxford Handbook of Quantitative Methods*, ed. T. D. Little, vol. 2, 551–610. New York: Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199934898.013.0025.

Reaven, G. M., and R. G. Miller. 1979. An attempt to define the nature of chemical diabetes using a multidimensional analysis. *Diabetologia* 16: 17–24. https://doi.org/10.1007/BF00423145.

Also see

[SEM] **Example 50g** — Latent class model

[SEM] **Example 51g** — Latent class goodness-of-fit statistics

[SEM] **Intro 5** — Tour of models

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