# Modeling with observed and latent groups

Jeff Pitblado
StataCorp LLC

2025 Swiss Stata Conference
Bern, Switzerland

# Outline

# Motivation

## Observed groups

What can you do with a variable that identifies groups in your data?

## Latent groups (classes)

What can you do when the groups are not deterministically identified by variables in your data?

# Example dataset

## Observed variables

- **`y`** is the dependent variable of interest.
  Suppose it is a count outcome.
  We will be using the Poisson model.
- **`x1`** and **`x2`** are continuous independent variables.
  We are interested in how they are associated with **`y`**.
- **`grp`** identifies group membership.
  We have observed two groups, say 1 and 2.

# `by:` prefix

## Description

- Repeat model fit on subsets of the data.

## Features

- Syntax is easy to learn and use.

## Limitations

- Testing parameters between groups is not easy.
- Constraints on parameters between groups is not possible.

# by: example

```
. use data
(Simulated data--Modeling with observed and latent groups)

. sort grp

. by grp: poisson y x1 x2, nolog

--------------------------------------------------------------------------------
-> grp = 1

Poisson regression                                      Number of obs =    122
                                                        LR chi2(2)    = 131.20
                                                        Prob > chi2   = 0.0000
Log likelihood = -212.8328                              Pseudo R2     = 0.2356

------------------------------------------------------------------------------
           y | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |   .0962749   .0127086     7.58   0.000     .0713666    .1211832
          x2 |  -.1814847   .0206201    -8.80   0.000    -.2218993   -.1410701
       _cons |   2.956803   .2116169    13.97   0.000     2.542041    3.371564
------------------------------------------------------------------------------

--------------------------------------------------------------------------------
-> grp = 2

Poisson regression                                      Number of obs =    178
                                                        LR chi2(2)    = 619.25
                                                        Prob > chi2   = 0.0000
Log likelihood = -410.976                               Pseudo R2     = 0.4297

------------------------------------------------------------------------------
           y | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |  -.0974296   .0062866   -15.50   0.000    -.1097511   -.0851081
          x2 |  -.1929588   .0099521   -19.39   0.000    -.2124646    -.173453
       _cons |   4.968026    .095185    52.19   0.000     4.781467    5.154585
------------------------------------------------------------------------------
```

# `if` clause

## Description

- Fit model to each group separately.

## Features

- Syntax is easy to learn and use.
- Group-specific outcome models.
- Use `etable` to report fitted parameters side-by-side.

## Limitations

- Testing parameters between groups is not easy.
- Constraints on parameters between groups is not possible.

# if example

```
. poisson y x1 x2 if grp==1, nolog
Poisson regression                                  Number of obs =      122
                                                    LR chi2(2)    =   131.20
                                                    Prob > chi2   =   0.0000
Log likelihood = -212.8328                          Pseudo R2     =   0.2356

------------------------------------------------------------------------------
           y | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |   .0962749   .0127086     7.58   0.000     .0713666    .1211832
          x2 |  -.1814847   .0206201    -8.80   0.000    -.2218993   -.1410701
       _cons |   2.956803   .2116169    13.97   0.000     2.542041    3.371564
------------------------------------------------------------------------------

. estimates store g1
. poisson y x1 x2 if grp==2, nolog
Poisson regression                                  Number of obs =      178
                                                    LR chi2(2)    =   619.25
                                                    Prob > chi2   =   0.0000
Log likelihood = -410.976                           Pseudo R2     =   0.4297

------------------------------------------------------------------------------
           y | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
          x1 |  -.0974296   .0062866   -15.50   0.000    -.1097511   -.0851081
          x2 |  -.1929588   .0099521   -19.39   0.000    -.2124646    -.173453
       _cons |   4.968026    .095185    52.19   0.000     4.781467    5.154585
------------------------------------------------------------------------------

. estimates store g2
```

# `if` example

```
. etable, estimates(g1 g2) column(estimates) ///
>         mstat(ll) mstat(N)
```

| | g1 | g2 |
|---|---|---|
| x1 | 0.096 | -0.097 |
| | (0.013) | (0.006) |
| x2 | -0.181 | -0.193 |
| | (0.021) | (0.010) |
| Intercept | 2.957 | 4.968 |
| | (0.212) | (0.095) |
| Log likelihood | -212.83 | -410.98 |
| Number of observations | 122 | 178 |

# `suest` command

## Description

- Combine estimation results into a seemingly unified estimation result, using linearized variance estimation.

## Features

- `test` equality of parameters between groups.
- Support for group-specific outcome models.

## Limitations

- Constraints on parameters between groups is not possible.
- No support for `predict` or `margins`.
- No support for random effects, mixed-effects, or multilevel models.

# suest example

```
. suest g1 g2
Simultaneous results for g1, g2                          Number of obs = 300
```

| | Coefficient | Robust std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| **g1_y** | | | | | | |
| x1 | .0962749 | .0036486 | 26.39 | 0.000 | .0891238 | .103426 |
| x2 | -.1814847 | .0060325 | -30.08 | 0.000 | -.1933083 | -.1696611 |
| _cons | 2.956803 | .0564931 | 52.34 | 0.000 | 2.846078 | 3.067527 |
| **g2_y** | | | | | | |
| x1 | -.0974296 | .0021771 | -44.75 | 0.000 | -.1016967 | -.0931625 |
| x2 | -.1929588 | .0034863 | -55.35 | 0.000 | -.1997919 | -.1861258 |
| _cons | 4.968026 | .0357811 | 138.84 | 0.000 | 4.897896 | 5.038155 |

# suest example

```
. suest, coeflegend
Simultaneous results for g1, g2                     Number of obs = 300

------------------------------------------------------------------------------
             |  Coefficient  Legend
-------------+----------------------------------------------------------------
g1_y         |
          x1 |    .0962749  _b[g1_y:x1]
          x2 |   -.1814847  _b[g1_y:x2]
       _cons |    2.956803  _b[g1_y:_cons]
-------------+----------------------------------------------------------------
g2_y         |
          x1 |   -.0974296  _b[g2_y:x1]
          x2 |   -.1929588  _b[g2_y:x2]
       _cons |    4.968026  _b[g2_y:_cons]
------------------------------------------------------------------------------

. test _b[g1_y:x1] = _b[g2_y:x1]
 ( 1)  [g1_y]x1 - [g2_y]x1 = 0

           chi2(  1) = 2078.51
         Prob > chi2 =    0.0000

. test _b[g1_y:x2] = _b[g2_y:x2]
 ( 1)  [g1_y]x2 - [g2_y]x2 = 0

           chi2(  1) =    2.71
         Prob > chi2 =    0.0996
```

# Factor variables

## Description

- Use factor variables notation to fit group-specific slopes and intercepts.

## Features

- **`test`** equality of parameters between groups.
- Impose equality constraints between groups.
- Use **`lrtest`** to compare model fits with different group constraint patterns.
- Supported by models with random effects, mixed-effects, or multilevel models.
- **`margins`** and **`contrast`** were designed for this.

# Factor variables

## Limitations

- No support for group-specific outcome models.
- Support for group-specific auxiliary parameters is limited to models that allow predictors in the auxiliary parameter equations.
- Random effects, mixed-effects, and multilevel parameters are group invariant.

# Factor variables example

```
. poisson y grp#c.(x1 x2) bn.grp, noconstant nolog
Poisson regression                                     Number of obs =       300
                                                       Wald chi2(6)  =  25499.84
Log likelihood = -623.8088                             Prob > chi2   =    0.0000

------------------------------------------------------------------------------
           y | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
    grp#c.x1 |
           1 |   .0962749   .0127086     7.58   0.000     .0713666    .1211832
           2 |  -.0974296   .0062866   -15.50   0.000    -.1097511   -.0851081
             |
    grp#c.x2 |
           1 |  -.1814847   .0206201    -8.80   0.000    -.2218993   -.1410701
           2 |  -.1929588   .0099521   -19.39   0.000    -.2124646    -.173453
             |
         grp |
           1 |   2.956803   .2116169    13.97   0.000     2.542041    3.371564
           2 |   4.968026    .095185    52.19   0.000     4.781467    5.154585
------------------------------------------------------------------------------

. estimates store free
```

# Factor variables example

```
. poisson, coeflegend

Poisson regression                                  Number of obs =        300
                                                    Wald chi2(6)  =   25499.84
Log likelihood = -623.8088                          Prob > chi2   =     0.0000

------------------------------------------------------------------------------
           y | Coefficient  Legend
-------------+----------------------------------------------------------------
    grp#c.x1 |
           1 |    .0962749  _b[1bn.grp#c.x1]
           2 |   -.0974296  _b[2.grp#c.x1]
             |
    grp#c.x2 |
           1 |   -.1814847  _b[1bn.grp#c.x2]
           2 |   -.1929588  _b[2.grp#c.x2]
             |
         grp |
           1 |    2.956803  _b[1bn.grp]
           2 |    4.968026  _b[2.grp]
------------------------------------------------------------------------------

. test _b[1.grp#x1] = _b[2.grp#x1]

 ( 1)  [y]1bn.grp#c.x1 - [y]2.grp#c.x1 = 0

           chi2(  1) =  186.65
         Prob > chi2 =    0.0000

. test _b[1.grp#x2] = _b[2.grp#x2]

 ( 1)  [y]1bn.grp#c.x2 - [y]2.grp#c.x2 = 0

           chi2(  1) =    0.25
         Prob > chi2 =    0.6163

. test _b[1.grp] = _b[2.grp]

 ( 1)  [y]1bn.grp - [y]2.grp = 0

           chi2(  1) =   75.13
         Prob > chi2 =    0.0000
```

# Factor variables example

```
. constraint 1 _b[1.grp#x2] = _b[2.grp#x2]
. poisson y grp#c.(x1 x2) bn.grp, noconstant nolog constr(1)
Poisson regression                                  Number of obs =       300
                                                    Wald chi2(5)  = 25490.18
Log likelihood = -623.93426                         Prob > chi2   =    0.0000
 ( 1)  [y]1bn.grp#c.x2 - [y]2.grp#c.x2 = 0
------------------------------------------------------------------------------
           y | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
    grp#c.x1 |
          1  |   .0966158   .0126846     7.62   0.000     .0717545    .1214771
          2  |  -.0973955   .0062872   -15.49   0.000    -.1097183   -.0850728
             |
    grp#c.x2 |
          1  |  -.1907964   .0089568   -21.30   0.000    -.2083513   -.1732414
          2  |  -.1907964   .0089568   -21.30   0.000    -.2083513   -.1732414
             |
         grp |
          1  |    3.04447   .1183256    25.73   0.000     2.812556    3.276384
          2  |   4.948426   .0867631    57.03   0.000     4.778373    5.118478
------------------------------------------------------------------------------
. lrtest free .
Likelihood-ratio test
Assumption: . nested within free
 LR chi2(1) =   0.25
Prob > chi2 = 0.6164
. estimates restore free
(results free are active now)
```

# Factor variables example

```
. margins grp, nopv

Predictive margins                                      Number of obs = 300
Model VCE: OIM

Expression: Predicted number of events, predict()
```

| | Margin | Delta-method std. err. | [95% conf. | interval] |
|---|---|---|---|---|
| grp | | | | |
| 1 | 5.572267 | .2247045 | 5.131854 | 6.01268 |
| 2 | 14.91818 | .2842416 | 14.36108 | 15.47528 |

# Factor variables example

```
. contrast r.grp#c.(x1 x2) r.grp
Contrasts of marginal linear predictions
Margins: asbalanced
```

| | df | chi2 | P>chi2 |
|---|---|---|---|
| grp#c.x1 | 1 | 186.65 | 0.0000 |
| grp#c.x2 | 1 | 0.25 | 0.6163 |
| grp | 1 | 75.13 | 0.0000 |

| | Contrast | Std. err. | [95% conf. | interval] |
|---|---|---|---|---|
| grp#c.x1 (2 vs 1) | -.1937045 | .0141785 | -.2214938 | -.1659152 |
| grp#c.x2 (2 vs 1) | -.0114741 | .0228961 | -.0563497 | .0334014 |
| grp (2 vs 1) | 2.011223 | .2320386 | 1.556436 | 2.46601 |

# `sem` command

## Description

- Fit multiple linear outcome models across subgroups of the data while allowing some parameters to vary and constraining others to be equal across subgroups.

## Features

- Easy syntax for constraints, and option `ginvariant()`.
- Test group invariance with postestimation command `estat ginvariant`.
- Use `lrtest` to compare model fits with different group constraint patterns.
- Fit multiple outcomes simultaneously.
- Support for CFA and SEM.

# `sem` command

## Limitations

- Not all outcomes are usefully fit using a linear model.
- No support for random effects, mixed-effects, or multilevel models.

# sem example

```
. generate logy = log(y)
. sem (logy <- x1 x2), group(grp) nolog nodescribe noheader nofootnote
Group: 1                                              Number of obs = 122

------------------------------------------------------------------------------
             |                 OIM
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Structural   |
  logy       |
          x1 |   .0967704   .0033914    28.53   0.000     .0901234    .1034174
          x2 |  -.1797791   .0054579   -32.94   0.000    -.1904764   -.1690819
       _cons |   2.930972   .0590996    49.59   0.000     2.815139    3.046805
-------------+----------------------------------------------------------------
  var(e.logy)|   .0138026   .0017672                      .0107393    .0177397
------------------------------------------------------------------------------

Group: 2                                              Number of obs = 178

------------------------------------------------------------------------------
             |                 OIM
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Structural   |
  logy       |
          x1 |  -.0958377   .0022986   -41.69   0.000    -.1003428   -.0913325
          x2 |  -.1911029   .0034952   -54.68   0.000    -.1979535   -.1842524
       _cons |   4.939567   .0369362   133.73   0.000     4.867173    5.011961
-------------+----------------------------------------------------------------
  var(e.logy)|   .0088759   .0009408                      .0072108    .0109255
------------------------------------------------------------------------------

. estimates store free
```

# sem example

```
. sem (logy <- x1 x2@a), group(grp) nolog nodescribe noheader nofootnote
Group: 1                                               Number of obs = 122

------------------------------------------------------------------------------
             |                 OIM
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Structural   |
  logy       |
          x1 |   .0969217   .0034205    28.34   0.000     .0902177    .1036257
          x2 |  -.1878394   .0029816   -63.00   0.000    -.1936833   -.1819955
       _cons |   3.013281   .0363377    82.92   0.000      2.94206    3.084502
-------------+----------------------------------------------------------------
  var(e.logy)|   .0140493   .0018081                      .0109172    .0180801
------------------------------------------------------------------------------

Group: 2                                               Number of obs = 178

------------------------------------------------------------------------------
             |                 OIM
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Structural   |
  logy       |
          x1 |  -.0957115   .0023031   -41.56   0.000    -.1002255   -.0911975
          x2 |  -.1878394   .0029816   -63.00   0.000    -.1936833   -.1819955
       _cons |   4.907224   .0322234   152.29   0.000     4.844067     4.97038
-------------+----------------------------------------------------------------
  var(e.logy)|   .0089194   .0009488                      .0072409     .010987
------------------------------------------------------------------------------

. lrtest free .

Likelihood-ratio test
Assumption: . nested within free

 LR chi2(1) =   3.03
Prob > chi2 = 0.0817
```

# **sem** example

```
. estat ginvariant
Tests for group invariance of parameters
```

| | Wald test chi2 | df | P>chi2 | Score test chi2 | df | P>chi2 |
|---|---|---|---|---|---|---|
| Structural | | | | | | |
| logy | | | | | | |
| x1 | 2180.641 | 1 | 0.0000 | . | . | . |
| x2 | . | . | . | 3.010 | 1 | 0.0827 |
| _cons | 6413.147 | 1 | 0.0000 | . | . | . |
| var(e.logy) | 6.268 | 1 | 0.0123 | . | . | . |

# `gsem` command

## Description

- Fit multiple outcome models across subgroups of the data while allowing some parameters to vary and constraining others to be equal across subgroups.

## Features

- Easy syntax for constraints, and option `ginvariant()`.
- `test` equality of parameters between groups.
- Use `lrtest` to compare model fits with different group constraint patterns.
- Fit multiple outcomes simultaneously.
- Fit the outcome model of interest.
- Group-specific outcome models.
- Support for CFA, IRT, generalized SEM, random effects, multilevel latent variables.

# gsem example

```
. gsem (y <- x1 x2), poisson group(grp) ginvariant(none) nolog noheader
Group: 1                                              Number of obs = 122

------------------------------------------------------------------------------
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
y            |
          x1 |   .0962749   .0127086     7.58   0.000     .0713666    .1211832
          x2 |  -.1814847   .0206201    -8.80   0.000    -.2218993   -.1410701
       _cons |   2.956803   .2116169    13.97   0.000     2.542041    3.371564
------------------------------------------------------------------------------

Group: 2                                              Number of obs = 178

------------------------------------------------------------------------------
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
y            |
          x1 |  -.0974296   .0062866   -15.50   0.000    -.1097511   -.0851081
          x2 |  -.1929588   .0099521   -19.39   0.000    -.2124646    -.173453
       _cons |   4.968026    .095185    52.19   0.000     4.781467    5.154585
------------------------------------------------------------------------------

. estimates store free
```

# gsem example

```
. gsem, coeflegend noheader nodvheader
Group: 1                                             Number of obs = 122

-------------------------------------------------------------------------------
             |  Coefficient  Legend
-------------+-----------------------------------------------------------------
y            |
          x1 |    .0962749  _b[y:1.grp#c.x1]
          x2 |   -.1814847  _b[y:1.grp#c.x2]
       _cons |    2.956803  _b[y:1.grp]
-------------------------------------------------------------------------------

Group: 2                                             Number of obs = 178

-------------------------------------------------------------------------------
             |  Coefficient  Legend
-------------+-----------------------------------------------------------------
y            |
          x1 |   -.0974296  _b[y:2.grp#c.x1]
          x2 |   -.1929588  _b[y:2.grp#c.x2]
       _cons |    4.968026  _b[y:2.grp]
-------------------------------------------------------------------------------

. test _b[y:1.grp#x1] = _b[y:2.grp#x1]
 ( 1)  [y]1bn.grp#c.x1 - [y]2.grp#c.x1 = 0

           chi2(  1) =  186.65
         Prob > chi2 =    0.0000

. test _b[y:1.grp#x2] = _b[y:2.grp#x2]
 ( 1)  [y]1bn.grp#c.x2 - [y]2.grp#c.x2 = 0

           chi2(  1) =    0.25
         Prob > chi2 =    0.6163
```

# gsem example

```
. gsem (y <- x1 x2@a), poisson group(grp) ginvariant(none) nolog noheader
Group: 1                                            Number of obs = 122

------------------------------------------------------------------------------
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
y            |
          x1 |   .0966158   .0126846     7.62   0.000     .0717545    .1214771
          x2 |  -.1907964   .0089568   -21.30   0.000    -.2083513   -.1732414
       _cons |    3.04447   .1183256    25.73   0.000     2.812556    3.276384
------------------------------------------------------------------------------

Group: 2                                            Number of obs = 178

------------------------------------------------------------------------------
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
y            |
          x1 |  -.0973955   .0062872   -15.49   0.000    -.1097183   -.0850728
          x2 |  -.1907964   .0089568   -21.30   0.000    -.2083513   -.1732414
       _cons |   4.948426   .0867631    57.03   0.000     4.778373    5.118478
------------------------------------------------------------------------------

. lrtest free .

Likelihood-ratio test
Assumption: . nested within free

 LR chi2(1) =   0.25
Prob > chi2 = 0.6164
```

# gsem example

```
. gsem (y <- x1 x2), poisson group(grp) ginvariant(coef) noheader nodvheader
Group: 1                                          Number of obs = 122
```

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| y | | | | | | |
| x1 | -.0588148 | .0055526 | -10.59 | 0.000 | -.0696977 | -.0479319 |
| x2 | -.185728 | .0089035 | -20.86 | 0.000 | -.2031786 | -.1682775 |
| _cons | 3.773633 | .0969348 | 38.93 | 0.000 | 3.583644 | 3.963621 |

```
Group: 2                                          Number of obs = 178
```

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| y | | | | | | |
| x1 | -.0588148 | .0055526 | -10.59 | 0.000 | -.0696977 | -.0479319 |
| x2 | -.185728 | .0089035 | -20.86 | 0.000 | -.2031786 | -.1682775 |
| _cons | 4.749134 | .0859279 | 55.27 | 0.000 | 4.580718 | 4.917549 |

```
. lrtest free .

Likelihood-ratio test
Assumption: . nested within free

 LR chi2(2) = 192.38
Prob > chi2 = 0.0000
```

# gsem example

```
. gsem (1: y <- x1 x2, poisson) ///
>      (2: y <- x1 x2, nbreg) ///
>      , group(grp) ginvariant(none) nolog

Generalized structural equation model             Number of obs     = 300
Grouping variable: grp                            Number of groups  =   2
Log likelihood = -623.8088

Group:     1                                        Number of obs = 122
Response:  y
Family:    Poisson
Link:      Log
```

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| y | | | | | | |
| x1 | .0962749 | .0127086 | 7.58 | 0.000 | .0713666 | .1211832 |
| x2 | -.1814847 | .0206201 | -8.80 | 0.000 | -.2218993 | -.1410701 |
| _cons | 2.956803 | .2116169 | 13.97 | 0.000 | 2.542041 | 3.371564 |

```
Group:       2                                      Number of obs = 178
Response:    y
Family:      Negative binomial
Dispersion:  mean
Link:        Log
```

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| y | | | | | | |
| x1 | -.0974357 | .0062867 | -15.50 | 0.000 | -.1097574 | -.085114 |
| x2 | -.1929708 | .0099523 | -19.39 | 0.000 | -.2124769 | -.1734646 |
| _cons | 4.968134 | .0951861 | 52.19 | 0.000 | 4.781573 | 5.154696 |
| /y | | | | | | |
| lnalpha | -18.31174 | 208.4945 | | | -426.9534 | 390.3299 |

# `fmm`: prefix

## Description

- If the group membership is not observed, you can use a finite mixture model.

## Features

- Prefix syntax using familiary estimation commands.
- Easy syntax for constraints, and option **`lcinvariant()`**.
- Fit the outcome model of interest.
- Class specific outcome models.
- Predict class membership probabilities.
- Use **`lcstats`** to test for the number of classes.

# fmm: example

```
. fmm 2, nolog nodvheader : poisson y x1 x2

Finite mixture model                                Number of obs = 300
Log likelihood = -765.56337

------------------------------------------------------------------------------
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
1.Class      |  (base outcome)
-------------+----------------------------------------------------------------
2.Class      |
       _cons |  -.5790036   .1573549    -3.68   0.000    -.8874136   -.2705936
------------------------------------------------------------------------------

Class: 1

------------------------------------------------------------------------------
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
y            |
          x1 |  -.1051311   .0064635   -16.27   0.000    -.1177994   -.0924629
          x2 |  -.2056394   .0105883   -19.42   0.000     -.226392   -.1848868
       _cons |   5.084371   .0987573    51.48   0.000     4.890811    5.277932
------------------------------------------------------------------------------

Class: 2

------------------------------------------------------------------------------
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
y            |
          x1 |   .1106521   .0135704     8.15   0.000     .0840547    .1372495
          x2 |  -.1849162   .0227743    -8.12   0.000    -.2295531   -.1402794
       _cons |   2.981518    .228961    13.02   0.000     2.532762    3.430273
------------------------------------------------------------------------------

. matrix b = e(b)

. estimates store fmm2
```

# fmm: example

```
. quietly fmm 1 : poisson y x1 x2
. estimates store fmm1
. lcstats fmm1 fmm2
Latent class statistics
```

| | Classes | N | ll | Rank | Entropy | df | LMR | P>LMR |
|---|---|---|---|---|---|---|---|---|
| fmm1 | 1 | 300 | -1,013.03 | 3 | | | | |
| fmm2 | 2 | 300 | -765.56 | 7 | 0.6030 | 4 | 474.15 | <0.001 |

```
LMR is the Lo-Mendell-Rubin-adjusted likelihood-ratio test statistic.
Likelihood-ratio tests compare the given model versus the same model with
one less latent class.
```

# **fmm**: example

```
. quietly fmm 3 : poisson y x1 x2
. estimates store fmm3
. lcstats fmm1 fmm2 fmm3
Latent class statistics
```

| | Classes | N | ll | Rank | Entropy | df | LMR | P>LMR |
|---|---|---|---|---|---|---|---|---|
| fmm1 | 1 | 300 | -1,013.03 | 3 | | | | |
| fmm2 | 2 | 300 | -765.56 | 7 | 0.6030 | 4 | 474.15 | <0.001 |
| fmm3 | 3 | 300 | -765.56 | 11 | 0.7060 | 4 | 0.00 | 1.000 |

```
LMR is the Lo-Mendell-Rubin-adjusted likelihood-ratio test statistic.
Likelihood-ratio tests compare the given model versus the same model with
one less latent class.
```

# **fmm**: example

```
. etable, estimates(fmm1 fmm2 fmm3) column(estimates) showeq mstat(none)
```

| | fmm1 | fmm2 | fmm3 |
|---|---|---|---|
| Class | | | |
| 2 | | | |
| Intercept | | -0.579 | -4.101 |
| | | (0.157) | (27050.782) |
| 3 | | | |
| Intercept | | | -0.609 |
| | | | (823.820) |
| y | | | |
| Class # x1 | | | |
| 1 | -0.060 | -0.105 | -0.105 |
| | (0.006) | (0.006) | (0.006) |
| 2 | | 0.111 | 0.111 |
| | | (0.014) | (0.082) |
| 3 | | | 0.111 |
| | | | (0.014) |
| Class # x2 | | | |
| 1 | -0.213 | -0.206 | -0.206 |
| | (0.009) | (0.011) | (0.011) |
| 2 | | -0.185 | -0.185 |
| | | (0.023) | (0.139) |
| 3 | | | -0.185 |
| | | | (0.023) |
| Class | | | |
| 1 | 4.737 | 5.084 | 5.084 |
| | (0.086) | (0.099) | (0.099) |
| 2 | | 2.982 | 2.982 |
| | | (0.229) | (1.397) |
| 3 | | | 2.982 |
| | | | (0.233) |

# `fmm`: example

```
. quietly fmm 2, from(b) : poisson y x1 x2@a

. lrtest fmm2 .

Likelihood-ratio test
Assumption: . nested within fmm2

 LR chi2(1) =   0.67
Prob > chi2 = 0.4117
```

# fmm: example

```
. quietly fmm 2, lcinvariant(coef) from(b) : poisson y x1 x2

. lrtest fmm2 .

Likelihood-ratio test
Assumption: . nested within fmm2

 LR chi2(2) = 108.68
Prob > chi2 = 0.0000
```

# Latent class models

## Wikipedia

- A latent class model (LCM) relates a set of observed (usually discrete) multivariate variables to a set of latent variables. It is a type of latent variable model. It is called a latent class model because the latent variable is discrete (categorical).
- Latent class analysis (LCA) is a subset of structural equation modeling, used to find groups or subtypes of cases in multivariate categorical data.

# LCA via `gsem` command

## Features

- Specify categorical latent variables in option `lclass()`.
- Easy syntax for constraints, and option `lcinvariant()`.
- `test` equality of parameters between classes.
- Use `lrtest` to compare model fits with different class constraint patterns.
- Fit multiple outcomes simultaneously.
- Class-specific outcome models.
- Use `lcstats` to test for the number of classes.

## Limitations

- Support is restricted to model specifications that do not include continuous latent variables.

# LCA example dataset

## Observed variables

- **y1**, **y2**, and **y3** are binary dependent variables of interest.
- **x** is a continuous independent variable.
  Use it to predict class membership and the dependent variables.

# LCA example

```
. gsem (y* <- x) (2.C <- x), logit lclass(C 2) nolog noheader
```

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| 1.C | (base outcome) | | | | | |
| 2.C | | | | | | |
| x | -.3669334 | .0538679 | -6.81 | 0.000 | -.4725124 | -.2613543 |
| _cons | -.2821683 | .0498944 | -5.66 | 0.000 | -.3799596 | -.184377 |

Class: 1

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| y1 | | | | | | |
| x | .9722039 | .1006049 | 9.66 | 0.000 | .775022 | 1.169386 |
| _cons | -1.98065 | .1169993 | -16.93 | 0.000 | -2.209965 | -1.751336 |
| y2 | | | | | | |
| x | 1.073417 | .1033773 | 10.38 | 0.000 | .8708013 | 1.276033 |
| _cons | -2.011822 | .1237851 | -16.25 | 0.000 | -2.254436 | -1.769208 |
| y3 | | | | | | |
| x | 1.328942 | .1152357 | 11.53 | 0.000 | 1.103084 | 1.5548 |
| _cons | -2.315209 | .1443729 | -16.04 | 0.000 | -2.598175 | -2.032244 |

Class: 2

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| y1 | | | | | | |
| x | 1.088556 | .1259243 | 8.64 | 0.000 | .841749 | 1.335363 |
| _cons | 2.105665 | .1543028 | 13.65 | 0.000 | 1.803237 | 2.408093 |
| y2 | | | | | | |
| x | .8507127 | .1265496 | 6.72 | 0.000 | .60268 | 1.098745 |
| _cons | 2.040876 | .148814 | 13.71 | 0.000 | 1.749206 | 2.332546 |
| y3 | | | | | | |
| x | 1.11105 | .1274936 | 8.71 | 0.000 | .8611668 | 1.360933 |
| _cons | 2.231151 | .1626808 | 13.71 | 0.000 | 1.912303 | 2.55 |

```
. matrix b = e(b)
. estimates store lca2
```

# LCA example

```
. gsem (y* <- x) (2.C <- x), logit lclass(C 2) nolog nodvheader
```

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| 1.C | (base outcome) | | | | | |
| 2.C | | | | | | |
| x | -.3669334 | .0538679 | -6.81 | 0.000 | -.4725124 | -.2613543 |
| _cons | -.2821683 | .0498944 | -5.66 | 0.000 | -.3799596 | -.184377 |

...

# LCA example

```
. gsem (y* <- x) (2.C <- x), logit lclass(C 2) nolog nodvheader
...
Class: 1
```

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| y1 | | | | | | |
| x | .9722039 | .1006049 | 9.66 | 0.000 | .775022 | 1.169386 |
| _cons | -1.98065 | .1169993 | -16.93 | 0.000 | -2.209965 | -1.751336 |
| y2 | | | | | | |
| x | 1.073417 | .1033773 | 10.38 | 0.000 | .8708013 | 1.276033 |
| _cons | -2.011822 | .1237851 | -16.25 | 0.000 | -2.254436 | -1.769208 |
| y3 | | | | | | |
| x | 1.328942 | .1152357 | 11.53 | 0.000 | 1.103084 | 1.5548 |
| _cons | -2.315209 | .1443729 | -16.04 | 0.000 | -2.598175 | -2.032244 |

...

# LCA example

```
. gsem (y* <- x) (2.C <- x), logit lclass(C 2) nolog nodvheader
...
Class: 2
```

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| y1 | | | | | | |
| x | 1.088556 | .1259243 | 8.64 | 0.000 | .841749 | 1.335363 |
| _cons | 2.105665 | .1543028 | 13.65 | 0.000 | 1.803237 | 2.408093 |
| y2 | | | | | | |
| x | .8507127 | .1265496 | 6.72 | 0.000 | .60268 | 1.098745 |
| _cons | 2.040876 | .148814 | 13.71 | 0.000 | 1.749206 | 2.332546 |
| y3 | | | | | | |
| x | 1.11105 | .1274936 | 8.71 | 0.000 | .8611668 | 1.360933 |
| _cons | 2.231151 | .1626808 | 13.71 | 0.000 | 1.912303 | 2.55 |

# LCA example

```
. gsem (y* <- x), logit lclass(C 1) nolog noheader
```

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| 1.C | (base outcome) | | | | | |

Class: 1

| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| y1 | | | | | | |
| x | .2762888 | .037255 | 7.42 | 0.000 | .2032703 | .3493072 |
| _cons | -.1812948 | .037028 | -4.90 | 0.000 | -.2538683 | -.1087212 |
| y2 | | | | | | |
| x | .2507082 | .0370367 | 6.77 | 0.000 | .1781176 | .3232987 |
| _cons | -.1462605 | .0369115 | -3.96 | 0.000 | -.2186058 | -.0739153 |
| y3 | | | | | | |
| x | .3208075 | .0375854 | 8.54 | 0.000 | .2471414 | .3944736 |
| _cons | -.1789673 | .037144 | -4.82 | 0.000 | -.2517682 | -.1061664 |

```
. estimates store lca1
```

# LCA example

```
. lcstats lca1 lca2
Latent class statistics
```

| | Classes | N | ll | Rank | Entropy | df | LMR | P>LMR |
|---|---|---|---|---|---|---|---|---|
| lca1 | 1 | 3,000 | −6,119.51 | 6 | | | | |
| lca2 | 2 | 3,000 | −5,088.32 | 14 | 0.7879 | 8 | 2,030.67 | <0.001 |

```
LMR is the Lo–Mendell–Rubin-adjusted likelihood-ratio test statistic.
Likelihood-ratio tests compare the given model versus the same model with one
less latent class.
```

# LCA example

```
. quietly gsem (y* <- x) (2.C <- x), logit lclass(C 2) lcinvariant(coef) from(b)
. lrtest lca2 .
Likelihood-ratio test
Assumption: . nested within lca2
 LR chi2(3) =    4.09
Prob > chi2 = 0.2523
```

*That's all folks!*