PENNSTATE

The Methodology Center
advancing methods, improving health

LCA Stata Plugin Users' Guide Version 1.2

Stephanie T. Lanza
John J. Dziak
Liying Huang
Aaron T. Wagner
Linda M. Collins

Please send questions and comments to *MChelpdesk@psu.edu*.
The development of the LCA Stata Plugin was supported by the National Institute on Drug Abuse Grant P50-DA10075 to The Center for Prevention and Treatment Methodology.

The suggested citation for this users' guide is
Lanza, S. T., Dziak, J. J., Huang, L., Wagner, A. T. , & Collins, L. M. (2015). *LCA Stata plugin users' guide* (Version 1.2). University Park: The Methodology Center, Penn State. Available from methodology.psu.edu. The authors would like to thank Daniele Pacifico for feedback and Stata code that were used to improve the plugin.

Table of Contents

1 PLUGIN FEATURES AND LIMITATIONS 3

1.1 Overview 3

1.2 Version modifications 3

2 LCA MATHEMATICAL MODEL 5

3 TECHNICAL DETAILS 7

3.1 Missing Data 7

3.2 Standard Errors 7

3.3 Clusters and Weights 7

4 PREPARING TO USE THE LCA STATA PLUGIN 9

4.1 Managing Files 9

4.2 Preparing Data 9

4.3 Calling the Plugin 9

5 LCA STATA PLUGIN SYNTAX 10

5.1 Mandatory Options 11

5.2 Non-Mandatory Options 11

5.3 Output 20

6 EXAMPLES 23

REFERENCES 24

1 Plugin Features and Limitations

1.1 Overview

The LCA Stata Plugin was developed for Stata for Windows (version 11.0 or higher). The plugin allows Stata users to employ the same functionality in the SAS procedure PROC LCA (Lanza, Dziak, Huang, Xu, & Collins, 2011). Both pieces of software were developed by The Methodology Center for conducting latent class analysis (LCA). This plugin can be used to estimate latent classes that are measured by categorical indicators. Key features of the LCA Stata Plugin include

- multiple-groups LCA,
- option to impose measurement invariance across groups,
- LCA with covariates (prediction of latent class membership),
- binary and multinomial logistic regression options for predicting latent class membership,
- the ability to take into account sampling weights and clusters,
- option for automatic starting values,
- option for applying data-derived prior in order to stabilize logistic regression (betaprior),
- posterior probabilities matrix generated in the output,
- parameter estimates generated in the output, and
- input data can be in aggregated (response-pattern data) form or one record per case.

This guide assumes the user has a working knowledge of LCA; an introduction can be found in Lanza, Bray, and Collins (2013) and Collins and Lanza (2010). A detailed empirical demonstration of PROC LCA (which employs nearly identical functionality) appears in Lanza, Collins, Lemmon, and Schafer (2007).

This document is intended for experienced Stata users; general Stata instructions are not included.

1.2 Version modifications

Important changes from version 1.1

- The LCA Stata Plugin now can accommodate larger and more complex analyses in lower versions of Stata. As a result, the plugin poses no limit based on matrix size.
- Three matrices, post_prob, madvec, and llvec no longer appears in the return list of matrices. The posterior probabilities, the "BestIndex" column, and the psuedoclass draws are now appended as additional columns in the users' data set. (Psuedoclass draws will only be present if the "seed_draws" option is used.) Results are not impacted.
- The covb matrix now appears in the return list of matrices when covariates are included in the model. This change was implemented to facilitate additional functionality for a future release.

Important changes from version 1.0

- The NSTARTs option is now available for use in models with covariates.

- The "BestIndex" column has been added to the r(post_prob) matrix. This column indicates which latent class is the best match for each individual based on posterior probabilities (i.e., maximum-probability assignment, also called modal assignment).
- The new seed_draws option allows users to generate 20 random simulations for each individual's potential class membership based on posterior probabilities (i.e., pseudo-class draws) and save them to the r(post_prob) matrix.

2 LCA Mathematical Model

Up to three sets of parameters are provided in the LCA Stata Plugin output.

- Gamma (γ) parameters: latent class membership probabilities
- Rho (ρ) parameters: item-response probabilities conditional on latent class membership
- Beta (β) parameters: logistic regression coefficients for covariates, predicting class membership

The ρ parameters express the correspondence between the observed items and the latent classes and form the basis for interpretation of the latent classes. When no covariates are included, only ρ and γ parameters are estimated. When covariates are included, only ρ and β parameters are estimated; in this case, the γ parameters are calculated as functions of β parameters and the covariates, and are provided in the LCA Stata Plugin output. If a grouping variable is included, all sets of parameters (γ, ρ, β) can be conditioned on group.

Suppose we estimate a latent class model with n_c classes from a set of M categorical items and include a covariate denoted X, which may be either continuous or dichotomous (zero/one coded). Let the vector $\mathbf{Y}_i=(Y_{i1},\ldots,Y_{iM})$ represent individual i's responses to the M items, where the possible values of Y_{im} are $1,\ldots,r_m$. Let $L_i=1,2,\ldots,n_c$ be the latent class membership of individual i, and let $I(y = k)$ be the indicator function; that is, a function that equals 1 if y equals k, and 0 otherwise. Suppose we let the last class be the reference class. Let X_i represent the value of the covariate for individual i; the covariate may be related to the probability of membership in each latent class, γ, but is assumed to be otherwise unrelated to $\mathbf{Y}_i$. Then the contribution by individual i to the likelihood is

$$P\left(\mathbf{Y}_i = \mathbf{y} \mid X_i = x\right) = \sum_{l=1}^{n_c} \gamma_l(x) \prod_{m=1}^{M} \prod_{k=1}^{r_m} \rho_{mk|l}^{I(y_m=k)} \quad . \tag{1}$$

The β parameters are the coefficients in logistic regressions using the covariate X to model the class membership parameters γ. The γ parameters can be expressed as

$$\gamma_l(x) = P(L_i = l \mid X_i = x) = \frac{\exp\left(\beta_{0l} + x\beta_{1l}\right)}{\sum_{j=1}^{n_c} \exp\left(\beta_{0j} + x\beta_{1j}\right)} = \frac{\exp\left(\beta_{0l} + x\beta_{1l}\right)}{1 + \sum_{j=1}^{n_c-1} \exp\left(\beta_{0j} + x\beta_{1j}\right)} \tag{2}$$

for $l=1,2,\ldots,n_c$. Note that the latter two terms on the right are equal because we assume that the last (i.e., the n_c-th) class is used as the reference class. The reference class has its βs constrained to zero, since the relative probabilities of being in the other classes are being compared to the probability of this reference class. It is necessary to set the βs for some class to zero for the sake of model identifiability, because of the natural constraint that the probabilities for all classes must sum to one for each individual, but it need not be the last class. The choice of reference class does not affect the final fitted probability estimates for any individual or class.

This model allows us to estimate the log odds that individual i falls in latent class l relative to the reference class. For example, if class 2 is the reference class, then the log odds of membership in class 1 relative to class 2 for an individual with value x on the covariate is

$$\log\left(\frac{\gamma_1(x)}{\gamma_2(x)}\right) = \beta_{01} + \beta_{11}x \quad . \tag{3}$$

Exponentiated β parameters are odds ratios, reflecting the increase in odds of class membership (relative to reference class n_c) corresponding to a one-unit increase in the covariate. Note that multiple covariates can be included simultaneously, just as in logistic regression. For models involving three or more latent classes, the LCA Stata Plugin also includes an option to conduct binary logistic regression, as opposed to baseline-category multinomial logistic regression, when predicting latent class membership. A comparison class is specified by the user, and all other latent classes are combined into one reference group. Covariates are then used to predict membership in the specified class relative to the others. This option provides a more parsimonious prediction model and may be useful in some cases in which the multinomial logistic regression model is not estimable due to sparseness.

3 Technical Details

In the LCA Stata Plugin, parameters are estimated by maximum likelihood using the EM algorithm, with Newton-Raphson incorporated into the estimation of regression coefficients for covariates. The convergence index used is the maximum absolute deviation (MAD). The MAD associated with a particular iteration of the estimation procedure is computed by calculating the absolute value of the difference between the current iteration parameter estimates and those corresponding to the previous iteration; the value assigned to MAD for that iteration is the largest number in this array. Ordinarily the value of MAD becomes smaller with each iteration of the estimation procedure, although there are conditions under which this may not hold. The estimation procedure iterates until either a previously specified criterion value of MAD (the convergence criterion) or a previously specified maximum number of iterations is reached.

3.1 Missing Data

Missing data on the latent class indicators are permitted in this plugin. Missing values should be represented as Stata system missing (".") as usual. When there are missing data the models expressed in Equation 1 are modified so that the product over $m = 1,\ldots,M$ is replaced by a product over the items observed for that individual.

Data are assumed to be missing at random (MAR). A test of the null hypothesis that data are missing completely at random (MCAR) also appears in the output. Missing data on covariates, groups, clusters, or weights (if these features are included in the model) are not allowed. That is, any record with missing data on a covariates, groups, clusters or weights variable specified in the model is eliminated from the analysis.

3.2 Standard Errors

Asymptotic standard errors for LCA parameter estimates are provided when available. For models without weights or clustering, standard errors are found by inverting the Hessian matrix of the log-likelihood (see the "standard" option in LatentGOLD; Vermunt & Magidson, 2005a, pp. 98-100, for technical details). For models with weights or clustering, a "robust" or "sandwich" standard error based on Taylor linearization is used (see the "robust" option in LatentGOLD).

3.3 Clusters and Weights

In many contexts in the social sciences, data arise from a sampling scheme more complicated than a simple random sample. Very often, participants are selected with unequal probabilities, so that in order to accurately describe population proportions, observations need to be given different weights. Also, instead of being independent, participants are often nested within clusters ("primary sampling units") such as schools, clinics or neighborhoods.

The LCA Stata Plugin accommodates clusters and weights using the pseudo-maximum-likelihood approach (Skinner, 1989; Vermunt & Magidson, 2005b, pp. 98-100). Under this approach, sampling weights are first standardized to have an average value of 1 over all of the individuals being analyzed; they are then used as if they were frequency weights in calculating the estimates. Clustering is ignored for estimation purposes, but is taken

into account in calculating standard errors by using a "robust" or "sandwich" style covariance estimate.

Note: The LCA Stata Plugin assumes that all of the data are from the same stratum in the sampling sense.

Note: Even if the **GROUPs** option is used, the weights are standardized to average to 1 across the whole analyzed data set, not within each group separately. Users who wish to take a different approach may standardize weights as they wish prior to conducting the latent class analysis, and then use the **origweights** option (see page 18) to specify that original weights be used.

Note: Latent GOLD (Vermunt & Magidson, 2005a, 2005b) uses the pseudo-likelihood approach by default to handle sampling weights and clustering. The pseudo-likelihood approach is also one of the two approaches available in MPlus for complex survey data (see Asparouhov 2005; Muthen & Muthen 2010, p. 233).

Note: When weights or clusters are present, inference is done using the "pseudo" or "weighted" log-likelihood function, since the true likelihood taking sampling into account may be difficult to find. Therefore, in the LCA Stata Plugin the G^2, AIC, BIC, CAIC, ABIC, and entropy statistics are also based on the log–pseudo-likelihood. However, the classic literature on these criteria generally assumes that they are based on a true log-likelihood from a model with equally weighted independent observations. This may mean that they are more difficult to interpret or must be interpreted with more caution because their statistical properties are largely unknown (Vermunt & Magidson, 2007). However, they may still be useful as heuristics (Wedel, ter Hofstede, & Steenkamp, 2008).

Note: When weights or clusters are present, the log-likelihood test for the significance of a covariate is corrected for the effects of the weights and clusters as recommended by Satorra and Bentler (1988) and Asparouhov and Muthén (2005).

4 Preparing to Use the LCA Stata Plugin

4.1 Managing Files

Three steps are required to set up the plugin before use.

1. Unzip the folder downloaded from methodology.psu.edu and place all the files in the desired folder on your computer.
2. Alter the folder path in the lca.do file:
    a. Open the file "lca.do"
    b. In the **4th line of code,** modify the path "D:\project\Stata_lca\Release\" to match the folder path where you placed the files. (This line has a comment, "/*CHANGE THIS PATH TO MATCH THE FILE LOCATION ON YOUR MACHINE*/")
    c. Save the changes.
3. Alter the folder path in the doLCA.ado file:
    a. Open the file "doLCA.ado"
    b. In the **final line of code,** modify the path "D:\project\Stata_lca\Release\lca.dll" to match the location of the lca.dll file. **Note: your edit should end with "lca.dll," not the folder location.**
4. Save the changes.

The plugin is ready for use.

4.2 Preparing Data

The data set can contain more variables than will be used in the analysis. It must contain at least two categorical variables to be used as indicators for the latent class model. The data set can be organized using one record per individual or aggregated with one record per response pattern. If data are aggregated, the data set must contain a frequency count variable.

4.3 Calling the Plugin

Run the examples in section 6 to see how to specify the options and run the plugin.

5 LCA Stata Plugin Syntax

The LCA Stata Plugin syntax is described below. Lines outside of the brackets are mandatory; inside brackets are non-mandatory. Capital letters indicate abbreviated version of the options when available.

```
program doLCA, rclass
 version 11.0
 syntax varlist,
   NClass (integer)                ///
   Categories (numlist >0 integer)    ///
   [ id (variables)                ///
     GROUPs (variables)            ///
     groupnames(string)            ///
     measurement(string)           ///
     COVariates (variables)        ///
     REFerence(integer)            ///
     binary(integer)               ///
     COREs(integer 1)              ///
     betaprior(real 0.0)           ///
     gammaprior(real 0.0)          ///
     rhoprior(real 0.0)            ///
     freq(varlist)                 ///
     weight(varlist)               ///
     clusters(varlist)             ///
     seed(real)                    ///
     seed_draws (real)             ///
     NSTARTs(integer 1)            ///
     maxiter(integer 5000)         ///
     criterion (real 0.000001)     ///
     gammastart(numlist >=0)       ///
     betastart(numlist >=0)        ///
     rhostart(numlist >=0)         ///
     gammarestrict(numlist >=0)    ///
     rhorestrict(numlist >=0)      ///
     nobetatest(string)            ///
     origweights(string) ]
   …
   end
```

Note: To generate output, include a "return list" command when the plugin is called. A list of returned scalars and matrices will be displayed after the plugin runs.

5.1 Mandatory Options

The following options are required in order to specify the model to be fit by the LCA Stata Plugin.

About the variable list

The variable list includes the categorical variables to be used as indicators of the latent classes. Two or more variables must be specified, and the number of arguments in the **Categories** option must equal the number of variables in this list. Each indicator must be coded with sequential integer values from 1 to *R*, where *R* is the number of response categories for that particular item. Missing values are permitted and should be coded as Stata system missing values (".."). **Note: covariates are either binary or continuous and are coded differently than the variables in this list.**

NClass (integer)

This option specifies the number of latent classes in the model to be estimated. Valid values are integers greater than or equal to 1.

Categories (number list >0 integer)

This option lists the number of response categories in each item in the variable list that appears after the **doLCA** option. Integer values must be listed in the same order as the variables listed after the **doLCA** option. Values must be between 2 and 99.

5.2 Non-Mandatory Options

Acceptable inputs and default values (if applicable) are listed inside the parentheses.

id (variable)

The **id** option is used to specify a variable in the analysis data that serves as the row names for the posterior probabilities matrix in the output. When data are not aggregated, the posterior probabilities matrix contains the following variables: items indicating the latent class variable (listed in the variable list), the grouping variable, the covariates, the posterior probabilities, and the variable listed in the **id** option. Typically, when data are not aggregated, a case identifier exists in the analysis data. By listing the case identifier in the **id** option, this identifier is carried through to the posterior probabilities matrix as row names. Examples 2 through 5 in Section 6 use the **id** variable. Note that only one variable can be specified in the **id** option.

GROUPs (variable)

Multiple-groups latent class analysis can be conducted using the LCA Stata Plugin. The grouping variable is specified in the **GROUPs** option. Only one grouping variable may be specified, although the user can cross several categorical variables to create a single grouping variable. The grouping variable must be coded with sequential integer values from 1 to the number of groups. When the **GROUPs** option is used, the user may wish to label the groups using the **groupnames** option. Cases with missing data for the grouping variable will be deleted automatically. The number of cases used in the analysis will be noted in the output and the number of cases read in and the number of deleted cases will be noted in the output.

User Tip: If the **GROUPs** and **clusters** options are both used, then the **measurement** option must also be used to specify measurement invariance. In other words, if the data arises from a cluster sampling scheme, then the LCA Stata Plugin requires the assumption of measurement invariance across groups.

groupnames (string)

This option allows the user to specify labels (up to 12 characters for each label) for the different levels of the grouping variable specified in the **GROUPs** option. The number of labels listed in the **groupnames** option must be equal to the number of groups, and the order of the labels must correspond to the order of the integers denoting the groups. This option should only be used in conjunction with the **GROUPs** option.

measurement (string)

When a grouping variable is provided in the **GROUPs** option, the user can use the **measurement** option to impose measurement invariance across all groups, without having to use the **rhorestrict** and **gammarestrict** options. The keyword **GROUPs** restricts estimation so that all ρ parameters (class-specific item response probabilities) are equal across groups. Example 2 in Section 6 demonstrates the use of the **measurement** option.

COVariates (variables)

One or more covariates can be incorporated in the latent class model by specifying the variable names in the **COVariates** option. The γ parameters (probabilities of latent class membership) will depend on the values or levels of the covariates. (The ρ parameters [item-response probabilities] will not depend on the values or levels of the covariates.) It is strongly recommended that the user first run the model without covariates to determine the latent structure (to select the number of latent classes), explore issues such as measurement invariance, and assess model fit. Note that covariates are treated as numeric (continuous variables and dummy-coded [i.e., dichotomous] variables are recommended). Cases with missing data for a covariate will be deleted. The number of cases used in the analysis will be noted in the output. See Example 3 in Section 6 for an example that includes covariates.

Note that when the **COVariates** option is specified, it is not possible to specify equivalence sets in the γ parameters. However, individual γ parameters may be fixed to their corresponding starting values.

REFerence (integer)

Use only in conjunction with the **COVariates** option. The **REFerence** option specifies the number of the latent class (an integer) to serve as the reference class for logistic regression. The minimum value is 1 and the maximum value is the number of classes specified in the **NClass** option.

Note: When random starting values are used, the order of the latent classes is random. When using the **seed** option, the user may wish to estimate a model, examine the output to choose the reference class, then specify that reference class in the syntax and rerun the model using the same **seed** value. Alternatively, the user may wish to use the **rhostart** and **gammastart** options so that the expected ordering of the latent classes can be known.

binary (integer)

Use only in conjunction with the **COVariates** option, in place of the **REFerence** option.

By default, the LCA Stata Plugin uses baseline-category multinomial logistic regression to predict latent class membership. However, a binary logistic model may be specified using the **binary** option.

Use this option to specify the number of the latent class to serve as the comparison group for binary logistic regression. The remaining latent statuses will be combined to form the reference group for binary logistic regression. The minimum value is 1 and the maximum value is the number of classes specified in the **NClass** option.

COREs (integer 1)

Specifies that the computational work should be divided among *value* different processors (cores), for multicore computers. The default value of 1 is assumed if this option is not specified. Other common values are 2 or 4 depending on your computer.

betaprior (real 0.0)

Use only in conjunction with the **COVariates** option, to invoke a stabilizing prior distribution on the β parameters. It creates a data-derived prior which is used in the estimation of each logistic model specified by the user. The value provided, which must be a positive real number, controls the strength of the prior (and how strongly we want it to influence the β estimates). A strength of 0 would mean no prior (ordinary maximum-likelihood estimation). A strength of 1 is recommended if a prior is desired.

User Tip: If estimation of a logit model fails, the problem is likely due to sparseness in the data. The recommended course of action is to invoke the **betaprior** option, which will solve most sparseness-related estimation problems. In extreme cases, this approach may not suffice and additional measures must be taken. One option is to reduce the number of parameters in the logit model by switching from a baseline-category multinomial logit model to a binary logit model. Also, be sure to check that no class membership probability is estimated at zero for one of the groups. (These should be examined in the model with no covariates, fit only to individuals who provided data on the covariate(s).) Any class membership probabilities that are estimated at a value very close to 0 can be fixed to 0 using the **gammarestrict** option. This eliminates the empty class from the logistic regression for that group.

Note: Use of the **betaprior** option is a practical solution for stabilizing the estimation of logit parameters when one or more of the β estimates diverge to infinity due to insufficient information for estimation. Sparseness is more likely to cause estimation problems when the sample size is small, one or more groups is small, one of the latent classes has a very small class membership probability, or when membership in one of the classes is essentially zero for some level of a covariate. This last condition can be difficult to identify, as true class membership is unknown. For more information about the prior used here, see Clogg, Rubin, Schenker, Schultz, and Weidman (1991).

gammaprior (real 0.0)

This option invokes a data-derived prior on the γ estimates. The value provided with **gammaprior**, which must be a positive real number, controls the strength of the prior (and how strongly we want it to influence the γ estimates). We recommend a strength of 1 as standard. The γ-stabilizing prior strength in LCA Stata Plugin is similar to the "Bayes constant" for "latent variables" in the latent class clustering functionality in LatentGOLD (Vermunt and Magidson 2005). It essentially adds a small number of pseudo-cases to each class, in order to improve estimation overall by biasing γ estimates away from zero. The specified strength is the total

number of pseudo-cases (if there is no grouping variable) or the total number per group (if there is). The **gammaprior** option can be used when there are no covariates; if your model has covariates use **betaprior** instead.

rhoprior (real 0.0)

This option invokes a data-derived prior on the ρ estimates. The ρ-stabilizing prior strength in the LCA Stata Plugin is somewhat similar to the "Bayes constant" for "categorical variables" in the latent class clustering functionality in LatentGOLD (Vermunt and Magidson, 2005). The value must be a positive real number. We recommend a strength of 1 as standard, although smaller or larger values can also be used. This prior acts somewhat like adding a small number of pseudo-cases to each response category for each class, in order to improve estimation overall by biasing it away from solutions in which some ρs are zero or one. This is important if standard errors are desired, because standard errors cannot be calculated for models with estimates on the boundary of the parameter space (parameters estimated at zero or one without a prior). The strength given is the total number of pseudo-cases (if there is no grouping variable) or the total number of pseudo-cases per group (if there is).

freq (varlist)

The LCA Stata Plugin can analyze data with one record per case or data that are aggregated into response patterns (with a count variable). The **freq** option must be used if data are aggregated. The variable containing the count variable is specified here. If data are not in aggregated form, this option should not be used. Frequency weights will usually be integers (whole numbers) but are not required to be. They are required to be greater than zero.

For aggregated data with 4 indicator variables (**Ind**) and a **Count** variable, the data would be coded like this. The **freq** option would be used to indicate that the data are aggregated.

Ind 1	Ind 2	Ind 3	Ind 4	Count
1	1	1	1	44
1	1	1	1	162
1	1	2	2	73

weight (varlist)

This option indicates that inverse-probability sampling weights should be used to adjust the data. By default, sampling weights are standardized before using them, so that they average to 1 over the subjects used in the analysis. (Cases that are deleted because they are missing covariates, are missing the grouping variable, are missing weights or frequencies, or are missing all of the indicators, are not included in this averaging.) Users who do not want weights to be standardized, or wish to do this manually, can use the **origweights** option.

clusters (varlist)

This option tells the LCA Stata Plugin that the subjects are not independent random draws, but are nested within clusters (primary sampling units) such as schools or classrooms. The schools or classrooms, rather than the individuals, are then assumed to be independent of each other. The **clusters** option identifies a variable which must consist of positive integers

(whole numbers), used as identification numbers for the cluster. For example, everyone having 2 in their cluster ID variable is assumed to be nested inside cluster 2.

seed (real)

Random starting values for the ρ parameters can be generated in the LCA Stata Plugin by specifying a positive integer value in the **seed** option. (Default starting values of 1/**NClass** for γ parameters and 0 for β parameters are used.) An integer seed to generate the random values allows the user to replicate the analysis at a later date. This option can't be used in conjunction with the **gammastart** and **rhostart** options. If the **gammastart** and **rhostart** options are not used, then a random number generator seed must be provided in the **seed** option. See Example 1 in Section 6 for a demonstration of this option.

Note: The seed should be an integer (whole number) greater than 1 and less than 2,000,000,000. We use arbitrary nine-digit seeds in the examples in Section 6. The value of the seed has no substantive meaning and is used to generate random values. We ask the user to specify the seed instead of simply using an automatically generated seed as many software packages do, because by saving the seed the user can generate the same sequence of random values when an analysis is completed. If the **seed** option is not included, the **gammastart** and **rhostart** options in the LCA Stata Plugin option must be included.

Note: For technical reasons, the **seed** option accepts any real number, but an integer should be used.

seed_draws (real)

The user provides an integer that serves as a random seed for generating posterior class membership draws for each individual, which are appended to the data set used in the analysis. If **seed_draws** is not used, the random posterior draws are not generated.

Note: The seed should be an integer (whole number) greater than 1 and less than 9,999,999,999.

NSTARTs (integer 1)

This command allows the model to be fit several times, in order to try to find the best estimates and avoid suboptimal local maxima of the likelihood function.

Change since version 1.0: NSTARTs can now be used when covariates are present in the model.

User Tip: It is good practice to check the identification of all models, both those without and with covariates. **NSTARTs** can be used to find the optimal seed or a good set of starting values, which can then be used to replicate the model at a later date.

Note: If the user specifies an **NSTARTs** value greater than one, a starting value in **seed** is still needed. However, this random seed is not used directly for creating starting values, but instead is used for generating a set of seeds to be used in the repeated estimation.

maxiter (integer 5000)

The **maxiter** option allows the user to specify the maximum number of iterations in the EM estimation procedure. The default value is 5000. If convergence is reached before the value specified in the **maxiter** option, the procedure will terminate normally.

criterion (real 0.000001)

The **criterion** option allows the user to specify the maximum absolute deviation (MAD) convergence criterion for the estimation procedure. The default value is 0.000001.

gammastart (numlist >=0)

The **gammastart** option allows the user to specify a list containing starting values for the γ parameters. This option must be used in conjunction with the **rhostart** option; the **betastart** option is optional. If starting values for the ρ parameters are of main interest, then the user can simply provide "flat" starting values (1/**NClass**) for the γ parameters. If a **GROUPs** option is not used, the list will contain **NClass** elements. If a **GROUPs** option is used, the list will include **NClass** elements for each group (group 1, then group 2, and so on). Example 4 in section 6 uses the **gammastart** and **rhostart** options.

Note : If the **gammastart** option is not invoked, the **seed** option (see page 14) must be included. If the **gammastart** option is invoked, the **seed** option and the **NSTARTs** option may not be included. Both **seed** and **gammastart** may not be specified together.

rhostart (numlist >=0)

The **rhostart** option allows the user to specify a list containing starting values for the ρ parameters. This option must be used in conjunction with the **gammastart** option; the **betastart** option is optional. If starting values for the ρ parameters are of main interest, then the user can simply provide "flat" starting values (1/**NClass**) for the γ parameters. This list must be structured as follows. There will be one column for each indicator variable. There will be a row for each latent class in each response category in each group. Graphically represented, it should look like this:

The chart below shows the meaning of each value in the **rhostart** list.

			Indicator variables 1 - 6					
Group 1	**Category 1**	**Class 1**	0.2	0.2	0.2	0.2	0.2	0.2
		Class 2	0.8	…	…	…	…	…
		Class 3	0.8	…	…	…	…	…
	Category 2	**Class 1**	0.8	…	…	…	…	…
		Class 2	0.2	…	…	…	…	…
		Class 3	0.2	…	…	…	…	…
Group 2	**Category 1**	**Class 1**	0.2	…	…	…	…	…
		Class 2	0.2	…	…	…	…	…
		Class 3	0.8	…	…	…	…	…
	Category 2	**Class 1**	0.8	…	…	…	…	…
		Class 2	0.8	…	…	…	…	…
		Class 3	0.2	…	…	…	…	…

```
rhostart ( 0.2 0.2 0.2 0.2 0.2 0.2 ///
           0.2 0.8 0.8 0.2 0.2 0.8 ///
           0.8 0.8 0.8 0.8 0.8 0.8 ///
           0.8 0.8 0.8 0.8 0.8 0.8 ///
           0.8 0.2 0.2 0.8 0.8 0.2 ///
           0.2 0.2 0.2 0.2 0.2 0.2 ///
           0.2 0.2 0.2 0.2 0.2 0.2 ///
           0.2 0.8 0.8 0.2 0.2 0.8 ///
           0.8 0.8 0.8 0.8 0.8 0.8 ///
           0.8 0.8 0.8 0.8 0.8 0.8 ///
           0.8 0.2 0.2 0.8 0.8 0.2 ///
           0.2 0.2 0.2 0.2 0.2 0.2)
```

Example 4 in Section 6 uses the **gammastart** and **rhostart** options.

Note : If the **rhostart** and **gammastart** options are not invoked, the **seed** option (see page 15) must be included. If the **rhostart** and **gammastart** options are invoked, the **seed** option and the **NSTARTs** option may not be included.

gammarestrict (numlist >=0)

The **gammarestrict** option allows the user to specify a list containing γ parameter restrictions. Parameter restrictions for the γ parameters can be used to test hypotheses about the prevalence of latent classes, or to fix the probability of membership in a latent class to zero for a particular group. When the **gammarestrict** option is used, the **rhorestrict** option must also be used. If a **GROUPs** option is not used, the list will contain **NClass** elements. If a **GROUPs** option is used, the list will include **NClass** elements for each group (group 1, then group 2, and so on). The list must specify a restriction option, indicated by an integer of value 0 or higher, corresponding to each parameter.

rhorestrict (numlist >=0)

The **rhorestrict** option allows the user to specify a list containing ρ parameter restrictions. Parameter restrictions for the ρ parameters can be useful to help achieve model identification or to test specific hypotheses about the measurement of the latent class variable. When the **rhorestrict** option is used, the **gammarestrict** option must also be used. The user must specify a list of parameter restrictions, indicated by an integer of value 0 or higher.

One column for each indicator. One row for each latent class in each response category in each group.

```
rhorestrict   ( 1 1 1 1 1 1 ///
              1 1 1 2 2 1 ///
              2 1 1 1 1 1 ///
              1 1 1 1 1 1 ///
              1 1 1 1 1 1 ///
              1 1 1 1 1 1 ///
              1 1 1 1 1 1 ///
              1 1 1 2 2 1 ///
              2 1 1 1 1 1 ///
              1 1 1 1 1 1 ///
              1 1 1 1 1 1 ///
              1 1 1 1 1 1 )
```

The chart below shows the meaning of each value in the **rhorestrict** list.

			Indicator Variables 1 - 6					
Group 1	**Category 1**	**Class 1**	1	1	1	1	1	1
		Class 2	1	…	…	…	…	…
		Class 3	2	…	…	…	…	…
	Category 2	**Class 1**	1	…	…	…	…	…
		Class 2	1	…	…	…	…	…
		Class 3	1	…	…	…	…	…
Group 2	**Category 1**	**Class 1**	1	…	…	…	…	…
		Class 2	1	…	…	…	…	…
		Class 3	2	…	…	…	…	…
	Category 2	**Class 1**	1	…	…	…	…	…
		Class 2	1	…	…	…	…	…
		Class 3	1	…	…	…	…	…

User Tip : For convenience, the **measurement** option (see page 12) can be used to restrict ρ parameters to be invariant across groups without using the **rhorestrict** option. If both the **rhorestrict** option and the **measurement** option are used, restrictions corresponding to ρ parameters for Group 1 that are provided in the matrix are applied to all subsequent groups. Additional information on the use of parameter restrictions can be found in separate documentation (WinLTA main manual, available at methodology.psu.edu).

rhosrestrict and **gammarestrict** restrictions

The following restrictions with each set of ρ and γ parameters are possible.

- *A parameter may be fixed to a specific value.* A value of 0 in the parameter restriction list indicates that the parameter is to be fixed. A parameter that is fixed is not estimated but remains at the starting value provided. If the user wishes to fix parameter estimates to a specific value, then the **gammastart** and **rhostart** options must be used in conjunction with the **rhorestrict** and **gammarestrict** options.
- *A parameter may be freely estimated with no restrictions.* A value of 1 in the parameter restriction list indicates that the parameter is to be freely estimated (this is also the default when the **gammarestrict** and **rhorestrict** options are not used).
- *A parameter may form part of an equivalence set.* Integers of value 2 or greater specify an equivalence set; estimates for all parameters with the same value are constrained equal to one another and only one parameter is estimated for each set.

Note : If an equivalence set is imposed in the γ parameters, then covariates may not be used to predict class membership.

Note : There are a few kinds of restrictions which still allow estimates to be computed but for which standard errors are unavailable. These are: (1) One or more γs are preset to constants. (2) Some, but not all, γs are put in equivalence sets. (3) A ρ in a polychotomous item (>2 categories) is constrained but another ρ in the same item is free.

Note : If the **gammarestrict** option is used then the **clusters** option may not be used.

nobetatest (string)

This option with the value "yes" suppresses tests of significance for covariates. This option has meaning only when the **covariates** option is used. If significance tests for covariates are not of interest, then invoking this option is recommended as it speeds up model estimation.

origweights (string)

The **origweights** option with the value "yes" can be used if it is desired that the weights be implemented "as-is." Users may wish to use this option if, for example, they wish to standardize weights to average to 1 within each **group** separately. When this statement is not used, the weights are standardized to average to 1 over the subjects included in the analysis. Thus, they are assumed to express the relative importance of each subject, but don't change the overall sample size.

Table 1: Summary of LCA Stata Plugin Options

Option	Description & Default (if applicable)
doLCA	Invokes the plugin with a list of indicator variables
NClass	Specifies number of latent classes
Categories	Specifies number of response categories in items
[ID	Declares identifier for posterior probabilities matrix
GROUPs	Declares categorical grouping variable
groupnames	Specifies a label for each group
measurement	Invokes measurement invariance across groups when value ="groups"
COVariates	Declares variables to include as covariates
REFerence	Specifies latent class to use as reference class in prediction from covariates
binary	Specifies latent class to use as comparison group in prediction from covariates, and specifies that binary logistic regression is to be used
CORES	Divides work between multiple cores on a multiprocessor computer. Default: 1
betaprior	Invokes a stabilizing prior for the β parameters. Prior strength must be specified; as a standard we recommend betaprior 1
gammaprior	Invokes a stabilizing prior for the γ parameters. Prior strength must be specified; as a standard we recommend gammaprior 1
rhoprior	Invokes a stabilizing prior for the ρ parameters. Prior strength must be specified; as a standard we recommend rhoprior 1
clusters	Declares a cluster ID (primary sampling unit) and tells LCA Stata Plugin that the data are clustered
freq	Identifies the frequency count variable, to use when data are aggregated
weight	Identifies the sampling weight variable, to use with complex survey data
seed	Specifies seed for random number generator *
seed_draws	Specifies seed for generating posterior class membership draws
NSTARTs	Specifies the number of different random starting values to use
maxiter	Specifies maximum number of iterations. Default: 5000
criteron	Specifies convergence criterion for maximum absolute deviation. Default: 0.000001
gammastart	Specifies the starting values for the γ parameters
rhostart	Specifies the starting values for the ρ parameters
gammarestrict	Specifies a list containing γ parameter restrictions
rhorestrict	Specifies a list containing ρ parameter restrictions
nobetatest	Suppresses tests of significance for covariates when input is "yes"
origweights]	Prevents the plugin from standardizing the weights to average to 1 across all subjects in the analysis when input is "yes"

- The **seed** option is required if the **gammastart** and **rhostart** options are not used. It is also required if the **NSTARTs** option is used. The **gammastart** and **rhostart** options may not be used with the **seed** option.

5.3 Output

The output for the plugin is extensive. Model information will be displayed once the plugin runs. You can include a "return list" command after calling the plugin to generate the list of returned scalars and matrices.

Returned model information includes

- number of subjects,
- number of measurement items,
- number of response categories per item,
- number of groups,
- number of latent classes,
- whether a seed was used to randomly generate rho starting values,
- parameter restrictions,
- percentage of seeds associated with best fitted model,
- maximum number of iterations,
- convergence method, and
- convergence criterion.

The fit statistics in the scalars list are also provided.

Table 2: Returned scalars

r (df)	The degrees of freedom of the fitted model. In models with no covariates, this is the number of cells in the contingency table, minus the number of parameters that are freely estimated, minus 1.
r (EntropyRsqd)	The scaled relative entropy $1 - S/(n \log K)$ (Ramaswamy et al., 1993).
r(EntropyRaw)	The mathematical entropy of the class partitioning, equal to $S = -\sum_{i=1}^{n}\sum_{k=1}^{L} p_{ik} \log p_i k$.
r(AdjustedBIC)	The adjusted BIC using Rissanen's sample size adjustment (see Sclove, 1987)
r(bic)	The Schwarz Bayesian information criterion (Schwarz, 1974; Lin & Dayton 1997)
r(aic)	The Akaike information criterion (Akaike, 1973; see Lin & Dayton 1997)
r(Gsquared)	The G^2 deviance statistic
r(loglikelihood)	The log-likelihood of the fitted model
r(iteration)	The number of iterations required to reach convergence
r(DesignEffect)	(Only when **weight** and/or **clusters** are used.) The multivariate design effect based on the matrices in the sandwich covariance estimate, if available (Skinner, Holt and Smith 1989; Vermunt and Magidson 2005b)

Note: Except for r(loglikelihood), these fit statistics are not provided when covariates are included in the model.

Note: When **weight** and **clusters** are used, the fit statistics are based on a pseudo-likelihood rather than a true likelihood, which may complicate their interpretation (see Vermunt & Magidson, 2007; Wedel, ter Hofstede, & Steenkamp, 2008).

Table 3: Returned matrices

r(covb)	This matrix is returned when there are covariates in the model. It contains the estimated covariance of the beta parameter estimates. It will facilitate additional functionality that we are developing for a future release.
r (gammas)	This matrix represents the membership probabilities for each seed when the **NSTARTs** option is used. In this matrix, each row represents the membership probabilities for a different seed.
r (logliks)	This matrix represents the log-likelihood for each starting value when the **NSTARTs** option is used. In this matrix, each row represents the log-likelihood for each seed.
r(seeds)	Lists the seeds used for each run when the **NSTARTs** option is used.
~~r(llvec)~~	Removed in version 1.2.
~~r(madvec)~~	Removed in version 1.2.
~~r(post_prob)~~	***Change since version 1.2:*** This matrix is no longer returned. The BestIndex column and 20 columns of random draws (when seed_draws is used) are now appended to the data set. The posterior probabilities for each latent class. In this matrix, each row represents a subject, and each column represents a latent class. ***Change since version 1.1:*** The BestIndex column indicates the latent class for which each individual has the highest posterior probability of membership. Also, if used in conjuction with the **seed_draws** option, this matrix will include 20 columns of random draws from the multinomial distribution defined by each individual's posterior probabilities (Bandeen-Roche et al., 1997; Wang, Brown, & Bandeen-Roche, 2005). This is often refered to as assignment based on multiple pseudo-class draws. The 20 pseudo-class draw assignments are indicated in newly created columns named Draw_1 through Draw_20.
r(rhoSTD)	The standard errors for the item response probabilities for each latent class. In this matrix, each row represents an indicator variable within a category within a group, and each column represents a latent class. (See figure on following page)
r(rho)	The item response probabilities for each latent class. In this matrix, each row represents an indicator variable within a category within a group, and each column represents a latent class. (See figure on following page)
r(gammaSTD)	The standard errors for the membership probabilities for each latent class for the best selected seed. In this matrix, each row represents a group and each column represents a latent class.

r(gamma)	The membership probabilities for each latent class for the best selected seed. In this matrix, each row represents a group and each column represents a latent class.
r(beta)	The logistic regression coefficients for predicting latent class membership. In this matrix, each row represents a covariate within a group and each column represents a latent class. (See figure below)
r(betaSTD)	The standard errors for the logistic regression coefficients for predicting latent class membership. In this matrix, each row represents a covariate within a group and each column represents a latent class. (See figure below)

The meaning of each cell in the r(rho) and r(rhoSTD) matrices (assuming 2 groups, 2 response categories for each item, 4 indicator variables (Var) , and 3 classes in the model).

			Classes 1 -3		
Group 1	**Category 1**	**Var 1**	…	…	…
		Var 2	…	…	…
		Var 3	…	…	…
		Var 4	…	…	…
	Category 2	**Var 1**	…	…	…
		Var 2	…	…	…
		Var 3	…	…	…
		Var 4	…	…	…
Group 2	**Category 1**	**Var 1**	…	…	…
		Var 2	…	…	…

The meaning of each cell in the r(beta) and r(betaSTD) matrices (assuming 2 groups, 2 response categories for each item, 4 indicator variables (Var) , and 3 classes in the model).

		Classes 1 -3		
Group 1	**Intercept**	…	…	…
	Cov 1	…	…	…
	Cov 2	…	…	…
	Cov 3	…	…	…
Group 2	**Intercept**	…	…	…
	Cov 1	…	…	…

6 Examples

The following examples are available in the "LcaSampleDataset" file which was downloaded with the plugin. By running the models specified in the lca.do file, you can see examples of the following models. (The example data set is loosely based on the Youth Risk Behavior Survey 2004 data set).

Example 1 Basic LCA model
Example 2 LCA model with groups, measurement invariance, and a rho prior
Example 3 LCA model with groups, covariates and a beta prior
Example 4 LCA model with groups, rho starts, and gamma starts
Example 5 LCA model with groups, rho starts, gamma starts, and restrictions

All examples rely on the same set of 12 indicators of youth risk behavior: smoking before age 13, daily smoking, ever having driven drunk, drinking alcohol before age 13, recent binge drinking, using marijuana before age 13, lifetime use of cocaine, lifetime use of glue, lifetime use of meth, lifetime use of ecstasy, having sex before age 13, and having a high number of lifetime sexual partners. Covariates included in the model are hours of TV watched per day and mother's education. Gender is used as the grouping variable.

All examples are shown with a 5-class solution. In example 1, the item-response probabilities indicate that the classes could be interpreted as follows:

- Class 1: Non-Users
- Class 2: Early Experimenters
- Class 3: Binge Drinkers
- Class 4: High Risk
- Class 5: Sexual Risk Takers

In other examples, the order of these classes is not always consistent with the above labels. This is because the order in which LCA classes are labeled as 1, 2, and so on is essentially random; the meaning of the classes is conveyed by the ρ parameters.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Csake (Eds.), *Second international symposium on information theory* (pp. 267-281). Budapest, Hungary: Akademiai Kiado.

Asparouhov, T. (2005). Sampling weights in latent variable modeling. *Structural Equation Modeling, 12,* 411-434.

Asparouhov, T., & Muthén, B. (2005). Multivariate statistical modeling with survey data. *Proceedings of the Federal Committee on Statistical Methodology (FCSM) Research Conference*. Washington, DC: Office of Management and Budget.

Bandeen-Roche, K., Miglioretti, D. L., Zeger, S. L., & Rathouz, P. J. (1997). Latent variable regression for multiple discrete outcomes. *Journal of the American Statistical Association, 92* (440), 1375-1386.

Bozdogan, H., (1987). Model-selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika, 52,* 345-370.

Bray, B. C., Lanza, S. T., & Tan, X. (in press). Eliminating bias in classify-analyze approaches for latent class analysis. *Structural Equation Modeling: A Multidisciplinary Journal.*

Clogg, C. C., Rubin, D. B., Schenker, N., Schultz, B., & Weidman, L. (1991). Multiple imputation of industry and occupation codes in census public-use samples using Bayesian logistic regression. *Journal of the American Statistical Association, 86,* 68-78.

Collins, L. M., & Lanza, S. T. (2010). *Latent class and latent transition analysis: With applications in the social, behavioral, and health sciences.* New York, NY: Wiley.

Lanza, S. T., Bray, B. C., & Collins, L. M. (2012). Latent class and latent transition analysis. In J. A. Schinka, W. F. Velicer & I. B. Weiner (Eds.), *Handbook of psychology* (2nd ed., Vol. 2, pp. 691-716). Hoboken, NJ: Wiley.

Lanza, S. T., & Collins, L. M. (2008). A new SAS procedure for latent transition analysis: Transitions in dating and sexual risk behavior. *Developmental Psychology, 44(2),* 446-456.

Lanza, S. T., Collins, L. M., Lemmon, D., & Schafer, J. L. (2007). PROC LCA: A SAS procedure for latent class analysis. *Structural Equation Modeling, 14(4),* 671-694.

Lanza, S. T., Dziak, J. J., Huang, L., Xu, S., & Collins, L. M. (2011). *PROC LCA & PROC LTA users' guide* (Version 1.2.7). University Park: The Methodology Center, Penn State. Retrieved from http://methodology.psu.edu

Lin, T. H., & Dayton, C. M. (1997). Model selection information criteria for non-nested latent class models. *Journal of Educational and Behavioral Statistics*, 22, 249-264.

Muthén, L.K., & Muthén, B.O. (2010). *Mplus user's guide* (6th ed.). Los Angeles, CA: Muthén & Muthén.

Ramaswamy, V., Desarbo, W. S., Reibstein, D. J., & Robinson, W. T. (1993). An empirical pooling approach for estimating marketing mix elasticities with PIMS data. *Marketing Science, 12,* 103-124.

Satorra, A. & Bentler, P. M. (1988). Scaling corrections for chi-square statistics in covariance structure analysis. *Proceedings of the Business and Economic Statistics Section of the American Statistical Association*, 308-313.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics, 6,* 461-464.

Sclove, L. S., (1987). Application of model-selection criteria to some problems in multivariate analysis. *Psychometrika, 52,* 333-343.

Skinner, C. J., Holt, D., & Smith, T. M. F. (eds.) (1989), *Analysis of complex surveys.* New York, NY: Wiley.

Skinner, C. J. (1989). Domain means, regression and multivariate analysis. In C. J. Skinner, D. Holt, & T. M. F. Smith (Eds.), *Analysis of complex surveys.* New York, NY: Wiley.

Vermunt, J. K., & Magidson, J. (2005a). *Latent GOLD 4.0 user's guide.* Belmont, Massachusetts: Statistical Innovations, Inc.

Vermunt, J. K., & Magidson, J. (2005b). *Technical guide for Latent GOLD 4.0: Basic and advanced.* Belmont, Massachusetts: Statistical Innovations Inc.

Wang, C., Brown, C. H., & Bandeen-Roche, K. (2005). Residual diagnostics for growth mixture models: Examining the impact of a preventive intervention on multiple trajectories of aggressive behavior. *Journal of the American Statistical Association, 100*(471), 1054-1076.

Index

Adjusted BIC, 19
AIC, 23
Baseline class, 5, *See also* Reference class
Bayes constant, 13
Beta, 5
Beta parameters, 5, 12, 14, 18
BETA PRIOR, 3, 12, 13, 18
BIC, 19
Binary, 3, 12, 18
CATEGORIES, 11, 18
CLUSTERS, 11, 14, 18
Convergence, 7, 15, 18
CORES, 12, 18
Covariates, 3, 5, 6, 7, 11, 12, 13, 14, 15, 17, 18, 19
CRITERION, 15, 18, *See also* Convergence Criterion
EM algorithm, 7
FREQ, 13
Gamma parameters, 15
GAMMA PRIOR, 13, 18
Grouping variable, 5, 11, 12, 13, 14, 18
GROUPNAMES, 11, 18
Groups, 3, 7, 11, 12, 13, 17, 18
ID, 11, 14, 18
Identification, 14, 16
LatentGOLD, 13
LOG_LIKELIHOOD, 19
Maximum absolute deviation, 7, 15, 18
Maximum likelihood, 7, 23
MAXITER, 15, 18
MEASUREMENT, 11, 12, 17, 18
Measurement invariance, 3, 11, 12, 18
Missing data, 7, 11, 12
NCLASS, 11, 12, 14, 15, 18
NOBETATEST, 17
NSTARTS, 15, 16, 18
ORIG_WEIGHTS, 8, 14, 17
Posterior probabilities, 3
Prior, 3, 8, 12, 13, 18
PROC LCA Syntax, 10, 18
Pseudo-cases, 13
Pseudo-maximum-likelihood, 7
REFERENCE, 12, 18
Reference class, 5, 12
RESTRICT, 13, 16
Rho, 5
RHO PRIOR, 13, 18
Sampling weights, 3
SEED, 12, 14, 15, 18
Sparseness, 6, 13
START, 12, 15, 18
Starting values, 3, 12, 14, 15, 18
Taylor linearization, 7
Weights, 7, 13, 14