PennState The Methodology Center
advancing methods, improving health

LCA_Distal_BCH Stata function Users' Guide (Version 1.1)

Liying Huang
John J. Dziak
Bethany C. Bray
Aaron T. Wagner
Penn State

Please send questions and comments to *MChelpdesk@psu.edu*.

The development of the LCA_Distal_BCH Stata function was supported by National Institute on Drug Abuse Grants P50 DA10075 and P50 DA039838. The authors would like to thank Amanda Applegate for helpful comments.

Thank you for citing this users' guide when you use this function. Suggested citation:

Huang, L., Dziak, J. J., Bray, B. C., & Wagner, A. T. (2017). *LCA_Distal_BCH Stata function users' guide* (Version 1.1). University Park, PA: The Methodology Center, Penn State. Retrieved from http://methodology.psu.edu

Contents

1 About the LCA_Distal_BCH Stata function

The LCA_Distal_BCH Stata function estimates the association between a latent class variable and a distal outcome using the approach of Bolck, Croon, and Hagenaars (2004), as adapted by Vermunt (2010). The LCA_Distal_BCH Stata function is designed to work with Stata Version 11.0 or higher and the Stata LCA plugin.

The LCA_Distal_BCH Stata function

- uses simple, minimal syntax;
- estimates class-specific response probabilities and standard errors for distal outcomes;
- estimates class-specific means and standard errors for continuous and count distal outcomes;
- provides significance tests to compare distal outcome means or proportions between classes; and
- can accommodate distal outcomes for multiple demographic groups.

This guide assumes the user has a working knowledge of latent class analysis and the LCA Stata plugin. The book, *Latent class and latent transition analysis: With applications in the social, behavioral, and health sciences* (Collins & Lanza, 2010), provides a comprehensive introduction to the use of latent class analysis in applied research.

Note: To use this function, you must also use the LCA Stata plugin v. 1.2.1 or higher. The LCA Stata plugin and the accompanying users' guide can be downloaded from http://methodology.psu.edu/downloads.

2 System Requirements

The LCA_Distal_BCH Stata function requires

- Stata Version 11.0 or higher (Windows version) and
- LCA Stata plugin Version 1.2.1 or higher (to fit LCA models).

3 The BCH Approach to LCA With a Distal Outcome

Researchers are often interested in the relationship between a latent class variable, *C,* and a distal outcome, *Z*. Often, they wish to compare the class-specific expected value E(Z|C=c) for each class *c*. This expected value is the same as the mean (average) for count or continuous variables. For binary variables coded as 0 or 1, the expected value is the proportion of 1s in the population, or equivalently, the probability of a 1 rather than a 0 for a single randomly selected population member.

The BCH method is a kind of “three-step” method. This means that (1) the parameters of the LCA model first are estimated without the distal outcome, then (2) the posterior probabilities of class membership based on this model are used to compute a special weighting variable, and finally (3) the weighting variable is used to calculate a weighted average of *Z* for each class. The simplest approach to creating weights is to use either the posterior probabilities themselves as weights (“proportional assignment”) or to round the highest probability for each subject to 1 and the others to zero (“modal assignment”), and to apply no further adjustment. However, this treats the posterior probabilities as if they were known quantities measuring degrees of class membership, and does not take into account uncertainty introduced by possible misclassification when estimating the model parameters. Bolck, Croon, and Hagenaars (2004) proposed a more accurate method that accounts for misclassification probabilities. Although they first proposed this method only in the case of categorical outcomes, Vermunt (2010) explained how to adapt it to continuous outcomes as well.

This function will calculate distal outcome estimates with either modal or proportional assignment, and either with BCH adjustment (“BCH” estimates) or without it (“naïve” or “unadjusted” estimates). It is generally better to use BCH adjustment rather than unadjusted estimates. However, as long as BCH adjustment is used, it usually does not matter very much whether modal assignment or proportional assignment is used. Occasionally, BCH assignment has been found to give an uninterpretable value (such as a negative probability); in this case, it is better to revert to the unadjusted assignment. The three steps followed by this function are described further below.

Step 1. Fit the LCA model to define latent class memberships, using only the indicator variables $\mathrm{Y}=Y_{1,\ldots,},Y_{\mathrm{m}}$, without including the distal outcome *Z* in the model. This will provide posterior probabilities of class membership, $\omega_{ic} = \mathrm{P}(C = c|Y = y_i)$, for each individual *i*=1,…,*N* in the dataset and each class *c*=1,…,n_c.

Step 2. Construct the weights for use in calculating weighted averages for each class on the distal outcome. The details depend on the options chosen.

- Unadjusted Modal Assignment. For each individual *i* and each possible class *c*, define the class weights w_{ic}. Specifically, let $\mathrm{w}_{ic} = 1$ if *c* is the most likely class (the maximum among $\omega_{i1}, \dots, \omega_{in_c}$) for a given individual, and 0 otherwise. For example, if individual *i* is estimated to have a 60% chance of belonging to class 2, then individual *i* will count as 100% of a member of class 2 and 0% of other classes.
- Unadjusted Proportional Assignment. Define the class weights as $\mathrm{w}_{ic} = \omega_{ic}$ for each individual *i* and each possible class *c*. For example, if individual *i* is estimated to have a 60% chance of belonging to class 2, then individual *i* will count as 60% of a member of class 2 when calculating weighted averages; the remaining 40% of the membership of individual *i* is divided among the remaining classes.
- BCH-Adjusted Modal Assignment. Calculate the misclassification matrix **D**. The entry in row *a* and column *b* of **D** represents the estimated probability that a subject who truly belongs to class *a* would be labeled as belonging to class *b*. Specifically, $\mathbf{D}_{ab}$ is calculated as $\sum_{i=1}^{N} \omega_{ia} w_{ib}^{unadj} / N\, \gamma_a$, where *N* is the number of subjects, w_{ib}^{unadj} is the unadjusted modal weight for individual *i* in class *b*, and γ_a is the estimated overall class probability P(*C*=*a*). Then calculate the vector of BCH weights using linear algebra as $\mathbf{w}^{BCH} = \mathbf{w}^{unadj}\mathbf{D}^{-1}$, where $\mathbf{w}^{unadj}$ is the *N*×n_c matrix of unadjusted modal weights *w*.
- BCH-Adjusted Proportional Assignment**.** Same as BCH-adjusted modal, but use the proportional weights for *w* instead of using the modal weights.

Step 3. Estimate the expected value of the distal outcome within each latent class by taking a weighted average of the observed values for all participants, weighted by each participant's value of $\mathbf{w}^{unadj}$ or $\mathbf{w}^{BCH}$, as requested by the user. Standard errors are calculated using Taylor linearization ("sandwich" covariance estimation).

Standard errors and tests. In principle, there are two ways of doing tests, or obtaining standard errors or confidence intervals, for non-normal distal outcomes. One is to treat them as simply averages and ignore the fact that they are not normally distributed. This is convenient and asymptotically valid, although not the most statistically efficient. The other is to assume a non-normal distribution (here we use Bernoulli for binary and Poisson for count) and construct the confidence intervals or tests for the underlying parameter (the logit probability or log mean) of this distribution. This function mostly imitates the behavior of the LatentGOLD software, in that standard errors are provided using the simpler method, and tests are performed using the more complicated method. For the binary case, non-symmetric confidence intervals are additionally provided using the more complicated method (calculating standard errors and confidence interval limits for the logit, and then back-transforming the ends of this confidence interval to describe the observed mean).

Pairwise and omnibus tests. The function provides Wald tests and p-values for comparing the expected values of the distal outcome between each pair of latent classes, testing the null hypothesis that the expected values are equal. The p-values are not adjusted for multiple comparisons, but a user who wishes to apply a Bonferroni correction could simply divide the alpha level used for comparison (e.g., .05) by the number of pairs being compared: specifically, by $n_c(n_c - 1)/2$. In addition to these tests, an omnibus test simultaneously comparing all of the expected values is also performed. For categorical outcomes in the current version of the function, only an omnibus test, rather than pairwise tests, is performed.

Sampling weights. If complex survey sample weights are used in the LCA (the `weight` option in the Stata LCA plugin) then these must be specified in this function also (using the `weight=` optional argument). Sampling weights are implemented by multiplying each w_{ic} by the corresponding sampling weight *s*$_i$. This is done before postmultiplying by $\mathbf{D}^{-1}$ in the BCH method. Note that although survey weights can be accommodated, the current version of the function does not account for clustering when calculating standard errors.

Grouping variable. The calculations of the function can accommodate an observed grouping variable (usually gender or other demographic categories) as in the `groups` option in the Stata LCA plugin. The function assumes measurement invariance across groups and performs calculations separately for each group. Separate output is also provided for each group.

4 Using the LCA_Distal_BCH Stata function

Table 1. Argument Definitions for the LCA_Distal_BCH Stata Function.

Option	Required	Description
`varname`	Yes	Distal outcome variable.
`gammaList`	Yes	List of estimates of the gamma parameters. Generated from the gamma matrix generated by the LCA Stata plugin.
`metric`	Yes	Metric assumed for the within-class distribution of the distal outcome variable. This may be the word “binary,” “categorical,” “count,” or “numerical.”
`groups`	No	Variable for multiple groups. If no group argument is supplied, the function assumes there is only one group.
`alpha`	No	Significance level. Default = 0.05.
`weight`	No	Name of the variable specifying survey weight. This option only works in the binary outcome case. It assumes that `weight` has also been used in the run of the LCA Stata plugin.
`adjustment _method`	No	The method, if any, of adjusting the class membership weights for the possibilitiy of misclassification. This may be “BCH” (default, recommended) or “unadjusted.”
`assignment`	No	The method of generating class membership weights based on the posterior probabilities, before doing the BCH adjustment, if any. This may be “modal” (default) or “proportional.”

4.1 Managing files and preparing data

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

1. Set up the LCA Stata plugin as described in the *LCA Stata plugin users’ guide.*
2. Unzip the files in the LCA_Distal_BCH Stata function folder downloaded from methodology.psu.edu/downloads and place all the files in the same file location where you placed the LCA Stata plugin files in step 1.
3. Run an example:
    a. Open relevant “.do” file
    b. In the **4th line of code,** modify the path “D:\project\Stata_lca\LCA_Distal_BCH_64-bit ” to match the folder path where you placed the files. **NOTE: If there are any spaces in your directory path, you will need to put the path in double quotation marks, per Stata convention.**
    c. Save the changes.

The function is ready to use.

Note: Missingness in the distal outcome variable should be imputed (e.g., multiple imputation; Schafer, 1997). Otherwise, cases with missing values in the distal outcome variable must be removed from the analysis.

4.2 Estimation of the Latent Class Model in the Stata LCA plugin

Use the Stata LCA plugin to generate the output needed for use by the LCA_Distal_BCH function. First, you must select the LCA model. This process is described in the *LCA Stata plugin users' guide* (Lanza, Dziak, Huang, Wagner, & Collins, 2015).

Once model selection is complete, generate the matrices containing the parameter estimates to be used in the plugin by estimating the latent class model.

```
qui doLCA Item001  Item002 Item003 Item004 Item005 Item006
   Item007 Item008, ///
     nclass(5) ///
     id(ID)                 ///
     maxiter(5000)              ///
     seed(12345)                ///
     nstarts (20)       ///
     categories(2 2 2 2 2 2 2 2)///
     criterion(0.000001)        ///
     rhoprior(1.0)
```

4.3 Converting matrices to lists

Note: Section 4.3 is a basic Stata operation and not part of our function, but we include it here as a convenient review for some users.

The LCA_Distal_BCH Stata function relies on inputs from the LCA Stata plugin, but the LCA Stata plugin generates matrices, and the LCA_Distal_BCH Stata function (by Stata convention) cannot accept matrices as input for options. This means that the r(gamma) matrix must be converted to a list after running the LCA Stata plugin and before running the LCA_Distal_BCH Stata function. The following code can be used for this purpose.

```
mat G = r(gamma)
```

```
forvalues i=1/`=rowsof(G)' {
      forvalues j=1/`=colsof(G)' {
      local glist `glist' `=G[`i', `j']'
      }
}
```

Then, the list, ‘`glist,`’ can be used in the LCA_Distal_BCH Stata function.

4.4 Function options and input

Run the function with the user-defined options in parentheses. The function parameters are shown below.

```
      doLCA_Distal_BCH varname,  ///
      gammaList(numlist) ///
      metric (string) ///
      [adjustment_method (string) ///
      assignment (string) ///
      GROUPs(varlist) /*INCLUDED ONLY WHEN GROUPS ARE IN THE DATA*/ ///
      alpha (real) ///
      weight (var list) ]
```

4.5 Output

The function produces both screen output and matrices. The output will always contain a set of estimates and standard erros for the expected value of the distal outcome within each class. In addition, for binary distal outcomes, a table of log odds estimates and asymmetric confidence intervals is provided. The function then provides a table of Wald chi-squared tests for testing the equality of expected values between classes. These include both pairwise and omnibus tests, except for categorical distal outcomes, for which only omnibus tests are provided.

The output is also contained in in two or three new matrices, depending on the `metric` used in the analysis.

- r(distal_tests)
- r(distal_estimates)
- r(distal_log_odds) This matrix is produced only for binary outcomes.

5 Demonstrations of the LCA_Distal_BCH Stata function

In this section, we first describe the structure of the data sets and the variables to be analyzed. Then, we illustrate how to estimate the distribution of the distal outcome within each latent class using the LCA_Distal_BCH Stata function and describe the output of the function. Section 5.1 describes use of the function with a binary distal outcome.

Examples of continuous, count, and categorical outcomes are avalable in the files included with the download of the LCA_Distal_BCH Stata function, available at methodology.psu.edu/downloads. This section of the users' guide is based on testBCH-Binary-BCH-Modal.do.

5.1 Example Data

Below are the first 10 observations from the data set **simdata_binary.dta**, which is contained in the files downloaded with the function.

ID	Item001	Item002	Item003	Item004	Item005	Item006	Item007	Item008	Z
1	2	2	1	2	2	2	2	2	1
2	1	1	2	2	2	2	2	2	0
3	2	1	2	1	1	1	1	1	0
4	2	2	2	2	2	2	2	2	1
5	2	2	2	2	2	2	2	2	1
6	1	1	1	2	2	2	2	2	1
7	2	2	1	2	2	2	2	2	1
8	2	2	2	2	2	2	2	2	1
9	2	2	2	2	2	2	2	2	1
10	2	2	2	2	1	2	2	2	1

ID = subject's identification variable,

Item001,…, Item008 = 8 items used to measure the latent class variable

Z = the distal outcome (Note: binary distal outcome should be coded using 0s and 1s.)

5.2 Example Syntax

First, estimate the latent class model in the LCA Stata plugin.

```
qui doLCA Item001  Item002 Item003 Item004 Item005 Item006 Item007
Item008, ///
      nclass(5) ///
        id(ID)                 ///
        maxiter(5000)  ///
        seed(12345) ///
        nstarts(20)   ///
        categories(2 2 2 2 2 2 2 2)///
        criterion(0.000001)  ///
        rhoprior(1.0)
```

The output is described in the *LCA Stata Plugin Users' Guide*.

Then, the r(gamma) matric needs to be converted to lists so it can be used as an input in the LCA_Distal_BCH Stata function.

```
        mat G = r(gamma)

        forvalues i=1/`=rowsof(G)' {
                       forvalues j=1/`=colsof(G)' {
                                     local glist `glist' `=G[`i', `j']'
                                    }
        }
```

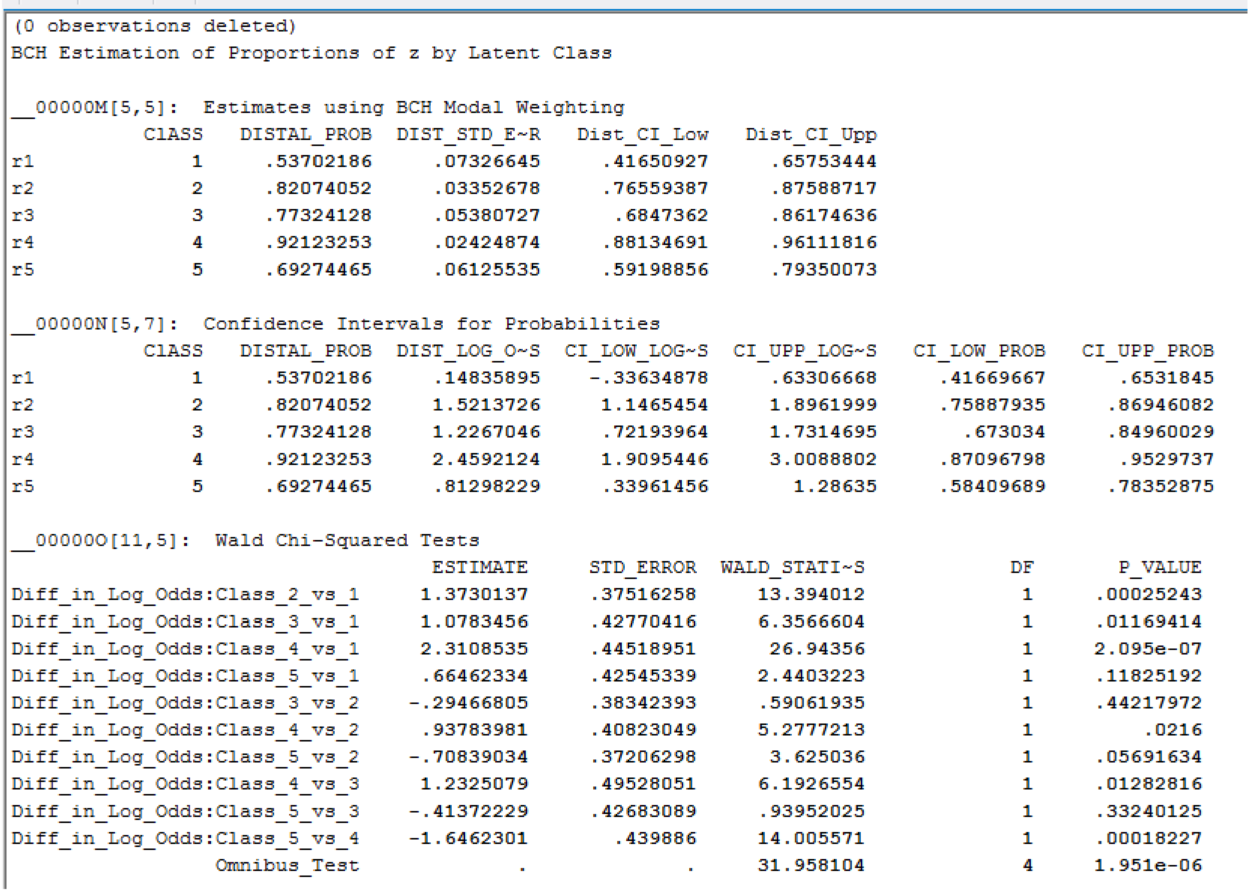
Now the distal outcomes function can be run. Include the function and enter the proper syntax in SAS.

```
doLCA_Distal_BCH z,  ///
        gammaList(`glist') ///
        metric("binary") ///
        alpha(0.1)
return list
```

The option `z` is the name of the distal outcome variable in this code. The `gammaList` option directs the function to the parameters in the list you just created. The `metric` option identifies the outcome as binary. The `alpha` option defines the significance level.

5.3 Example Output

Below is the onscreen output. It includes the class-specific distribution estimates for the distal outcome, the estimated class-conditional probabilities, the confidence intervals, and the Wald test statistic on class-conditional probabilities.



```
(0 observations deleted)
BCH Estimation of Proportions of z by Latent Class

__00000M[5,5]:  Estimates using BCH Modal Weighting
            ClASS   DISTAL_PROB  DIST_STD_E~R   Dist_CI_Low   Dist_CI_Upp
r1              1     .53702186     .07326645     .41650927     .65753444
r2              2     .82074052     .03352678     .76559387     .87588717
r3              3     .77324128     .05380727      .6847362     .86174636
r4              4     .92123253     .02424874     .88134691     .96111816
r5              5     .69274465     .06125535     .59198856     .79350073

__00000N[5,7]:  Confidence Intervals for Probabilities
            ClASS   DISTAL_PROB  DIST_LOG_O~S  CI_LOW_LOG~S  CI_UPP_LOG~S   CI_LOW_PROB   CI_UPP_PROB
r1              1     .53702186     .14835895    -.33634878     .63306668     .41669667      .6531845
r2              2     .82074052     1.5213726     1.1465454     1.8961999     .75887935     .86946082
r3              3     .77324128     1.2267046     .72193964     1.7314695       .673034     .84960029
r4              4     .92123253     2.4592124     1.9095446     3.0088802     .87096798      .9529737
r5              5     .69274465     .81298229     .33961456       1.28635     .58409689     .78352875

__00000O[11,5]:  Wald Chi-Squared Tests
                                  ESTIMATE     STD_ERROR  WALD_STATI~S            DF       P_VALUE
Diff_in_Log_Odds:Class_2_vs_1    1.3730137     .37516258     13.394012             1     .00025243
Diff_in_Log_Odds:Class_3_vs_1    1.0783456     .42770416     6.3566604             1     .01169414
Diff_in_Log_Odds:Class_4_vs_1    2.3108535     .44518951      26.94356             1     2.095e-07
Diff_in_Log_Odds:Class_5_vs_1    .66462334     .42545339     2.4403223             1     .11825192
Diff_in_Log_Odds:Class_3_vs_2   -.29466805     .38342393     .59061935             1     .44217972
Diff_in_Log_Odds:Class_4_vs_2    .93783981     .40823049     5.2777213             1         .0216
Diff_in_Log_Odds:Class_5_vs_2   -.70839034     .37206298      3.625036             1     .05691634
Diff_in_Log_Odds:Class_4_vs_3    1.2325079     .49528051     6.1926554             1     .01282816
Diff_in_Log_Odds:Class_5_vs_3   -.41372229     .42683089     .93952025             1     .33240125
Diff_in_Log_Odds:Class_5_vs_4   -1.6462301       .439886     14.005571             1     .00018227
                 Omnibus_Test            .             .     31.958104             4     1.951e-06
```

The same information will also be returned as matrices:

- r(distal_tests)
- r(distal_log_odds)
- r(distal_estimates)

The matrices are returned for technical use. They contain the data summarized in the onscreen output.

References

Bakk, Z., Oberski, D. L., & Vermunt, J. K. (2014). Relating latent class assignments to external variables: Standard errors for corrected inference. *Political Analysis, 22*, 520-540. doi:10.1093/pan/mpu003

Bakk, Z., & Vermunt, J. K. (2016). Robustness of stepwise latent class modeling with continuous distal outcomes. *Structural Equation Modeling, 23,* 20-31*.* doi:10.1080/10705511.2014.955104

Bolck, A., Croon, M., & Hagenaars, J. (2004). Estimating latent structure models with categorical variables: One-step versus three-step estimators. *Political Analysis*, *12*(1), 3–27.

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.

Dziak, J. J., Bray, B. C., & Wagner, A. T.(2017). *LCA_Distal_BCH SAS macro users' guide* (Version 1.1). University Park, PA: The Methodology Center, Penn State. Retrieved from http://methodology.psu.edu

Dziak, J. J., Bray, B. C., Zhang, J. - T., Zhang, M., & Lanza, S. T. (2016). Comparing the performance of improved classify-analyze approaches in latent profile analysis. Methodology: *European Journal of Research Methods for the Behavioral and Social Sciences, 12,* 107-116. http://doi.org/10.1027/1614-2241/a000114

Lanza, S. T., Dziak, J. J., Huang, L., Wagner, A. T., & Collins, L. M. (2015). *Proc LCA & Proc LTA users' guide* (Version 1.3.2). University Park: The Methodology Center, Penn State. Available from methodology.psu.edu

Lanza, S. T., Tan, X., & Bray, B. C. (2013). Latent class analysis with distal outcomes: A flexible model-based approach. *Structural Equation Modeling: A Multidisciplinary Journal, 20*, 1-20.

Schafer, J. L. (1997) *Analysis of incomplete multivariate data*. Boca Raton, FL: Chapman and Hall/CRC.

Vermunt, J. K. (2010). Latent class modeling with covariates: Two improved three-step approaches. *Political Analysis, 18,* 450–469.