The Stata Journal (2019)
19, Number 3, pp. 698–718 DOI: 10.1177/1536867X19874246

gidm: A command for generalized inflated discrete models

Yiwei Xia
Southwestern University of Finance and Economics
Chengdu, China

Yisu Zhou
Faculty of Education
University of Macau
Macau, China

Tianji Cai
Department of Sociology
University of Macau
Macau, China
tjcai@um.edu.mo

Abstract. In this article, we describe the `gidm` command for fitting generalized inflated discrete models that deal with multiple inflated values in a distribution. Based on the work of Cai, Xia, and Zhou (Forthcoming, *Sociological Methods & Research*: Generalized inflated discrete models: A strategy to work with multimodal discrete distributions), generalized inflated discrete models are fit via maximum likelihood estimation. Specifically, the `gidm` command fits Poisson, negative binomial, multinomial, and ordered outcomes with more than one inflated value. We illustrate this command through examples for count and categorical outcomes.

Keywords: st0574, gidm, generalized inflated discrete models, multiple inflated values, maximum likelihood

1 Introduction

Social science researchers have long recognized the inflation of certain values for discrete variables. For example, the number of children born within a family is concentrated on values of 0, 1, and 2 (Poston and McKibben 2003). Inflation brings challenges to traditional discrete models. For instance, the observed proportions for the inflated values exceed the probabilities that regular distributions would allow. If not modeled properly, inflations might lead to biased estimates and incorrect inferences (Lambert 1992).

Past decades have witnessed a rapid development of inflated models. Scholars have extended the inflation models not only on the forms of discrete distributions but also on allowing the number of inflation points to be more than one. To address an excess of zero counts in data, Lambert (1992) proposed a zero-inflated Poisson (ZIP) model that implements two separate models—a Poisson count model and a logit model for predicting excessive zeros. In the same vein, the zero-inflated framework has been applied to other discrete distributions, such as zero-inflated negative binomial (ZINB) regression (Ridout, Hinde, and DeméAtrio 2001); the zero-inflated binomial model (Diop, Diop, and Dupuy 2016; Hall 2000; Vieira, Hinde, and Demetrio 2000); zero-inflated multinomial regression (Diallo, Diop, and Dupuy 2018); the inflated ordered logistic model (Bagozzi and Mukherjee 2012); and the zero-inflated ordered probit (ZIOP) model (Bagozzi et al. 2015).

Recently, scholars have extended the zero-inflated models to allow for an arbitrary number of inflation points. For example, Lin and Tsai (2013) proposed a zero-k-inflated Poisson model that allows a second inflation point at the value k besides zero. Begum, Mallick, and Pal (2014) suggested a generalized inflated Poisson (GIP) model to address multiple inflations for responses in categorical forms. The most recent work by Cai, Xia, and Zhou (Forthcoming) has further extended GIP to a general form and introduced a generalized inflated discrete model (GIDM), which uses arbitrary and multiple inflations for a wide range of discrete probability distributions, such as multinomial, ordinal, Poisson, and zero truncated Poisson.

Despite the recent theoretical development of inflated models, the implementation has been lacking, especially for Stata users. Prior to Stata 15, `zip` and `zinb` were the only two available commands for modeling zero-inflated counts. Stata 15 introduced the ZIOP model via the command `zioprobit`. To the best of our knowledge, GIDM has not been integrated into Stata. To fill this gap, we developed the `gidm` command to implement the GIDMs, including the GIP, generalized inflated negative binomial, generalized inflated multinomial, and generalized inflated ordered models.

The rest of this article is organized as follows: Section 2 gives a brief introduction of GIDM, focusing on estimation. Section 3 explains the syntax and options of the `gidm` command. Sections 4 and 5 present illustrations using two well-known examples: the number of fish caught in a state park and the fictional data on smoking habits. Section 6 discusses issues such as the goodness of fit, nonconvergence problems, and further directions of development.

2 The GIDM

Following Begum, Mallick, and Pal (2014), Cai, Xia, and Zhou (Forthcoming) suggested a general framework of inflated values for discrete outcomes. Suppose Y is a discrete random variable that has inflated probabilities at values $k_1, \ldots, k_m \in \{0, 1, 2, \ldots\}$. The probability mass function (PMF) could be written as

$$p(Y = k|\lambda, \pi_i, 1 \le i \le m) = \begin{cases} \pi_i + (1 - \sum_{i=1}^{m} \pi_i) \times p(k|\lambda) & \text{if } k = k_1, \ldots, k_m \\ (1 - \sum_{i=1}^{m} \pi_i) \times p(k|\lambda) & \text{if } k \neq k_i, 1 \le i \le m \end{cases}$$

where $p(Y = k|\lambda)$ is a discrete PMF with the parameter λ for outcome k, π_i is the probability of inflation at the value k_i with $1 \le i \le m$, and $\sum_{i=1}^{m} \pi_i \in (0, 1)$.

The above parameterization suggests that the PMF can be considered a combination of probabilities for several binary outcomes and one regular discrete outcome. If the value k_i for a respondent falls in the set of inflated values $k_1, \ldots, k_m \in \{0, 1, 2, \ldots\}$, the PMF is a sum of two components: π_i denoting the chance of inflation for value k_i and $(1 - \sum_{i=1}^{m} \pi_i) \times p(k_i|\lambda)$ indicating the conditional probability for value k_i from the regular discrete PMF, for example, Poisson, negative binomial, etc. If k_i does not belong to the set of inflated values, the PMF shrinks to the conditional probability of $(1 - \sum_{i=1}^{m} \pi_i) \times p(k_i|\lambda)$. The probability of inflation at the value k_i, π_i could also depend on covariates. For example, if a logit model is specified,

$$\pi_i = \frac{1}{1+\exp(-\boldsymbol{z_s}\boldsymbol{\gamma_i})}$$

where $\boldsymbol{z_s}$ and $\boldsymbol{\gamma_i}$ is the vector of predictors for the sth observation and the vector of corresponding parameters, respectively. A probit model could be derived if a probit function is specified for π_i.

The GIDM offers a framework that covers all the inflated models commonly seen in social sciences, such as ZIP, ZINB, GIP, generalized inflated negative binomial, generalized inflated multinomial, and generalized inflated ordered. Once the GIDM is specified, the full likelihood function $L(\boldsymbol{\theta})$ can be constructed accordingly. The maximum likelihood estimator of the unknown parameter of $\boldsymbol{\theta}$ can be obtained by solving the score function:

$$\frac{\partial \log L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$$

The Fisher information matrix can be obtained by taking the second derivative of the log likelihood with respect to $\boldsymbol{\theta}$. The unknown parameters $\boldsymbol{\theta}$ can be estimated by method of moments (Hansen 1982), direct maximum likelihood (Cai, Xia, and Zhou Forthcoming), or maximum likelihood via the expectation-maximization algorithm (Begum, Mallick, and Pal 2014). Diallo, Diop, and Dupuy (2018) provided a rigorous investigation of the maximum likelihood estimator in terms of the identifiability, existence, consistency, and asymptotic normality under classical regularity conditions.

The `gidm` command maximizes the log likelihood of GIDM using Stata's `ml` command (Gould, Pitblado, and Poi 2010). The `gidm` command supports the specification of the following distributions: Poisson, negative binomial, ordinal logistic and probit, and multinomial logistic and probit. Binomial distribution is not singled out in the `gidm` command, because it can be estimated as a two-category case of multinomial distribution.

3 The gidm command

3.1 Description

The `gidm` command fits a GIDM of *depvar* on several sets of *indepvars* and *varlistN*. The *depvar* is a nonnegative integer of the response variable. The *indepvars* is a set of explanatory variables for *depvar*, whereas *varlist1* to *varlistN* are sets of explanatory variables for modeling the probabilities of inflation at each of the points corresponding to the values specified in the *numlist* in the option `inflation(`*numlist*`)`. Specifically, an intercept-only model can be specified as `(_con)`.

3.2 Syntax

`gidm` (*depvar indepvars*) (*varlist1*) [... (*varlistN*)] [*if*] [*in*] [*weight*],
`inflation(`*numlist*`)` `link(`*string*`)` [`noinitial` `vce(`*vcetype*`)` `level(`#`)`
display_options *maximize_options*]

3.3 Options

`inflation(`*numlist*`)` specifies the list of values at which the inflations are assumed. The number of elements in *numlist* must be the same as the number of equations specified by *indepvars* and *varlist1* ... *varlistN*. `inflation()` is required.

`link(`*string*`)` defines the distribution for both of the noninflated and the inflated parts. We use a four-letter combination to represent each model. The first two letters, for example, `lg` for logit and `pb` for probit, indicate the functional form for the inflated part, and the last two letters refer to the distribution of outcome. The supported distributions for the outcome are Poisson (`po`), negative binomial (`nb`), multinomial (`ml`), cumulative logit (`cl`), and cumulative probit (`cp`). For instance, the keyword `lgpo` refers to a logit-inflated Poisson, and `pbcp` is a probit-inflated cumulative probit. `link()` is required. A summary of the keywords of models supported by the `gidm` command is given in table 1.

Table 1. Link options

Outcome	Model	Option `link(`*string*`)`	
		Logit inflations	Probit inflations
Count	Poisson	`lgpo`	`pbpo`
	Negative binomial	`lgnb`	`pbnb`
Category	Multinomial	`lgml`	`pbml`
	Ordered logit	`lgcl`	`pbcl`
	Ordered probit	`lgcp`	`pbcp`

`noinitial` suppresses the default initial values that are from results of the separately fit model parts. For example, with `link(lgpo)`, the default initial values are obtained from a separately fit Poisson model for the main part and logistic regressions for the inflated parts.

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`, `opg`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] ***vce_option***.

`level(`#`)`; see [R] **Estimation options**.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

Setting the optimization type to technique(bhhh) resets the default *vcetype* to vce(opg).

3.4 Stored results

gidm stores the following in e():

Scalars

e(N)	number of observations
e(k)	number of parameters
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(chi2)	χ^2
e(p)	significance of model test
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	gidm
e(depvar)	name of dependent variable
e(chi2type)	Wald or LR; type of model chi-squared test
e(vce)	*vcetype* specified in vce()
e(opt)	type of optimization
e(ml_method)	type of ml method
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V

Matrices

e(b)	coefficient vector
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators

Functions

e(sample)	marks estimation sample

4 The number of fish caught example: The inflated count models

The number of fish caught dataset is used to showcase the capability of the `gidm` command for fitting inflated count models. The dependent variable is the number of fish caught for each individual (`count`). The independent variables include whether the individuals brought a camper (`camper`), how many adult people were in the group (`persons`), and how many children were in the group (`child`). Figure 1 shows that there are large proportions of visitors who did not harvest any fish (56.8%), or only one (12.4%), although the average number of fish caught was 3.296.

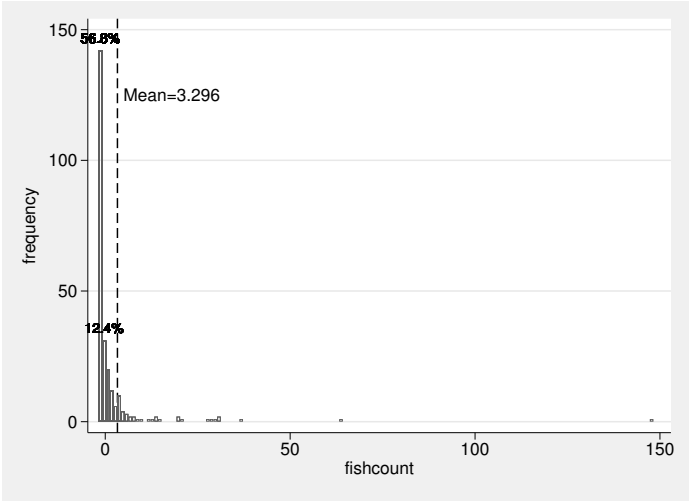

Figure 1. Histogram of fish count

We first run Stata's `zip` command with the variables `child` and `camper` as the predictors for the number of fish caught. The variable `persons` is set as the only predictor for the probability of inflation—the excess zeros. Then, the same model is fit using the `gidm` command. Notice that the `zip` command specifies the dependent variable and the predictors of the Poisson part in the main part of the command and uses the option `inflate()` to specify variables that predict the inflation. The `gidm` command allows users to specify the predictors of both the Poisson and the inflated parts in the main body and separates them by parentheses for the Poisson part and the inflated part. Two options, `inflation()` and `link()`, are used to define the value at which inflation is assumed and the distribution of outcome, respectively.

```
. webuse fish
(Fictional fishing data)
. zip count child camper, inflate(persons)
Fitting constant-only model:
Iteration 0:   log likelihood =  -1347.807
Iteration 1:   log likelihood = -1315.5343
Iteration 2:   log likelihood = -1126.3689
Iteration 3:   log likelihood = -1125.5358
Iteration 4:   log likelihood = -1125.5357
Iteration 5:   log likelihood = -1125.5357
Fitting full model:
Iteration 0:   log likelihood = -1125.5357
Iteration 1:   log likelihood = -1044.8553
Iteration 2:   log likelihood = -1031.8733
Iteration 3:   log likelihood = -1031.6089
Iteration 4:   log likelihood = -1031.6084
Iteration 5:   log likelihood = -1031.6084
Zero-inflated Poisson regression               Number of obs     =        250
                                               Nonzero obs       =        108
                                               Zero obs          =        142
Inflation model = logit                        LR chi2(2)        =     187.85
Log likelihood  = -1031.608                    Prob > chi2       =     0.0000
------------------------------------------------------------------------------
       count |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
count        |
       child |  -1.042838   .0999883   -10.43   0.000    -1.238812    -.846865
      camper |   .8340222   .0936268     8.91   0.000      .650517    1.017527
       _cons |   1.597889   .0855382    18.68   0.000     1.430237     1.76554
-------------+----------------------------------------------------------------
inflate      |
     persons |  -.5643472   .1629638    -3.46   0.001    -.8837503    -.244944
       _cons |   1.297439   .3738522     3.47   0.001     .5647022    2.030176
------------------------------------------------------------------------------
. gidm (count child camper) (persons), inflation(0) link(lgpo)
Iteration 0:   log likelihood = -1143.8173
Iteration 1:   log likelihood =  -1052.681
Iteration 2:   log likelihood = -1031.6311
Iteration 3:   log likelihood = -1031.6084
Iteration 4:   log likelihood = -1031.6084
                                               Number of obs     =        250
                                               Wald chi2(2)      =     177.74
Log likelihood = -1031.6084                    Prob > chi2       =     0.0000
------------------------------------------------------------------------------
       count |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
count        |
       child |  -1.042838   .0999883   -10.43   0.000    -1.238812   -.8468651
      camper |   .8340222   .0936268     8.91   0.000      .650517    1.017527
       _cons |   1.597889   .0855382    18.68   0.000     1.430237     1.76554
-------------+----------------------------------------------------------------
inf_at_0     |
     persons |  -.5643472   .1629638    -3.46   0.001    -.8837504   -.2449441
       _cons |   1.297439   .3738523     3.47   0.001     .5647023    2.030176
------------------------------------------------------------------------------
```

Comparing the estimates obtained from the two commands, we see that the coefficients, the standard errors, the p-values, and the confidence intervals are the same past the fourth decimal place. If a ZINB model is specified, the `gidm` command generates the same estimates, standard errors, and p-values compared with those obtained from the `zinb` command, although the latter outputs more information. Users can get exponentiated coefficients and a description of the dependent variable. Thus, the `gidm` command provides a convenient and integrated alternative to estimate both ZIP and ZINB models.

```
. zinb count child camper, inflate(persons)
Fitting constant-only model:
Iteration 0:   log likelihood = -519.33992
Iteration 1:   log likelihood = -471.96077
Iteration 2:   log likelihood = -465.38193
Iteration 3:   log likelihood = -464.39882
Iteration 4:   log likelihood = -463.92704
Iteration 5:   log likelihood = -463.79248
Iteration 6:   log likelihood = -463.75773
Iteration 7:   log likelihood =  -463.7518
Iteration 8:   log likelihood = -463.75119
Iteration 9:   log likelihood = -463.75118
Fitting full model:
Iteration 0:   log likelihood = -463.75118  (not concave)
Iteration 1:   log likelihood = -440.43162
Iteration 2:   log likelihood = -434.96651
Iteration 3:   log likelihood = -433.49903
Iteration 4:   log likelihood = -432.89949
Iteration 5:   log likelihood = -432.89091
Iteration 6:   log likelihood = -432.89091
Zero-inflated negative binomial regression      Number of obs     =        250
                                                Nonzero obs       =        108
                                                Zero obs          =        142
Inflation model = logit                         LR chi2(2)        =      61.72
Log likelihood  = -432.8909                     Prob > chi2       =     0.0000
```

count	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
count						
child	-1.515255	.1955912	-7.75	0.000	-1.898606	-1.131903
camper	.8790514	.2692731	3.26	0.001	.3512857	1.406817
_cons	1.371048	.2561131	5.35	0.000	.8690758	1.873021
inflate						
persons	-1.666563	.6792833	-2.45	0.014	-2.997934	-.3351922
_cons	1.603104	.8365065	1.92	0.055	-.036419	3.242626
/lnalpha	.9853533	.17595	5.60	0.000	.6404975	1.330209
alpha	2.678758	.4713275			1.897425	3.781834

```
. gidm (count child camper) (persons), inflation(0) link(lgnb)
Iteration 0:   log likelihood = -483.16863
Iteration 1:   log likelihood = -441.51271
Iteration 2:   log likelihood = -433.92875
Iteration 3:   log likelihood = -432.98746
Iteration 4:   log likelihood = -432.89348
Iteration 5:   log likelihood = -432.89092
Iteration 6:   log likelihood = -432.89091
Log likelihood = -432.89091                     Number of obs     =        250
```

count	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
count						
child	-1.515255	.1955913	-7.75	0.000	-1.898607	-1.131903
camper	.8790513	.2692732	3.26	0.001	.3512856	1.406817
_cons	1.371049	.256113	5.35	0.000	.8690765	1.873021
inf_at_0						
persons	-1.666554	.6792729	-2.45	0.014	-2.997904	-.3352032
_cons	1.603094	.8364998	1.92	0.055	-.0364153	3.242604
lnalpha						
_cons	.9853527	.1759497	5.60	0.000	.6404975	1.330208

Including extra inflation points is straightforward—by adding sets of predictors in the main part of the command and specifying additional inflation points in the `inflation()` option. For example, a zero-one-inflated Poisson model (Melkersson and Olsson 1999) can be specified as follows:

```
. gidm (count child camper) (persons) (persons), inflation(0 1) link(lgpo)
Iteration 0:   log likelihood = -1109.1181
Iteration 1:   log likelihood = -993.45106
Iteration 2:   log likelihood = -909.73629
Iteration 3:   log likelihood =  -908.6897
Iteration 4:   log likelihood = -908.68566
Iteration 5:   log likelihood = -908.68566
                                                Number of obs     =        250
                                                Wald chi2(2)      =     137.48
Log likelihood = -908.68566                     Prob > chi2       =     0.0000
```

count	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
count						
child	-1.031535	.1164113	-8.86	0.000	-1.259697	-.8033725
camper	.8208807	.0977577	8.40	0.000	.6292792	1.012482
_cons	1.861406	.0893422	20.83	0.000	1.686299	2.036514
inf_at_0						
persons	-.4432465	.1484102	-2.99	0.003	-.734125	-.1523679
_cons	1.134845	.3550344	3.20	0.001	.4389906	1.8307
inf_at_1						
persons	-.6008594	.2193122	-2.74	0.006	-1.030703	-.1710154
_cons	-.8175616	.4524518	-1.81	0.071	-1.704351	.0692277

The results indicate that the number of people contributes not only to the inflation at value 0 but also to that at value 1. The more people in a group, the less likely the number of fish caught will be 0 or 1.

5 The tobacco example: The inflated cumulative probit and logit models

We use the tobacco data to show how to implement the inflated ordered logistic and probit models. Suppose we are interested in factors that contribute to the number of cigarettes smoked per day by an individual. The dependent variable `tobacco` measures the number of cigarettes smoked in a day and has been grouped into four levels: `0 cigarettes`, `1 to 7 cigarettes/day`, `8 to 12 cigarettes/day`, and `more than 12 cigarettes/day` coded as `0`, `1`, `2`, and `3`, accordingly. The independent variables included are `female`, `income`, and `age`. Usually, the natural choice for the ordered outcomes is either an ordered logistic or a probit model with a proportional odds assumption, which assumes that the effects of independent variables are the same for different levels of responses and that the only difference lies in the intercepts and thresholds. Because the descriptive statistics show that 63.1% of respondents were nonsmokers, it is reasonable to assume a zero-inflated model with probit or logit for the inflation part. We use both the command `gidm` and the Stata command `zioprobit` to fit a ZIOP. Besides the independent variables, two additional variables—whether a parent smoked (`smoking`) and whether a respondent's religion discourages smoking (`religion`)—are also enclosed to account for the inflation on zeros. The output below shows that the results obtained from both commands are identical.

```
. webuse tobacco, clear
(Fictional tobacco consumption data)

. zioprobit tobacco female income age,
> inflate(female income age parent i.religion)

Iteration 0:   log likelihood = -15393.004
Iteration 1:   log likelihood = -13583.121  (not concave)
Iteration 2:   log likelihood = -13568.745
Iteration 3:   log likelihood = -13521.797
Iteration 4:   log likelihood = -13519.044
Iteration 5:   log likelihood = -13518.998
Iteration 6:   log likelihood = -13518.998

Zero-inflated ordered probit regression         Number of obs     =     15,000
                                                Wald chi2(3)      =    1356.58
Log likelihood = -13518.998                     Prob > chi2       =     0.0000
```

tobacco	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
tobacco						
female	-.1835352	.0318024	-5.77	0.000	-.2458667	-.1212037
income	.1528552	.0042382	36.07	0.000	.1445485	.1611619
age	-.1461085	.0078566	-18.60	0.000	-.1615072	-.1307098
inflate						
female	-.1858712	.0492558	-3.77	0.000	-.2824108	-.0893316
income	-.0794141	.0079746	-9.96	0.000	-.0950441	-.0637841
age	.1393169	.0167948	8.30	0.000	.1063997	.1722341
parent	.7796176	.0496775	15.69	0.000	.6822515	.8769836
religion						
discourage..	-.3316872	.0619942	-5.35	0.000	-.4531935	-.2101808
_cons	.1640407	.0822251	2.00	0.046	.0028825	.3251988
/cut1	-.0261029	.0523262			-.1286603	.0764546
/cut2	1.20342	.0429703			1.1192	1.28764
/cut3	1.851093	.0451114			1.762676	1.939509

```
. gidm (tobacco female income age) (female income age parent i.religion),
> inflation(0) link(pbcp)
Iteration 0:   log likelihood = -16315.766
Iteration 1:   log likelihood = -13678.854  (not concave)
Iteration 2:   log likelihood = -13584.028
Iteration 3:   log likelihood = -13522.021
Iteration 4:   log likelihood = -13519.123
Iteration 5:   log likelihood = -13518.998
Iteration 6:   log likelihood = -13518.998
                                                Number of obs     =     15,000
                                                Wald chi2(3)      =    1356.58
Log likelihood = -13518.998                     Prob > chi2       =     0.0000
```

tobacco	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
tobacco						
female	-.1835352	.0318024	-5.77	0.000	-.2458667	-.1212037
income	.1528551	.0042382	36.07	0.000	.1445484	.1611618
age	-.1461085	.0078566	-18.60	0.000	-.1615072	-.1307098
inf_at_0						
female	-.1858712	.0492558	-3.77	0.000	-.2824108	-.0893316
income	-.0794142	.0079746	-9.96	0.000	-.0950442	-.0637842
age	.1393171	.0167948	8.30	0.000	.1063999	.1722343
parent	.779618	.0496775	15.69	0.000	.6822519	.8769841
religion						
discourage..	-.3316874	.0619942	-5.35	0.000	-.4531938	-.210181
_cons	.1640411	.0822251	2.00	0.046	.0028829	.3251993
cut1						
_cons	-.0261025	.0523262	-0.50	0.618	-.1286599	.076455
cut2						
_cons	1.20342	.0429703	28.01	0.000	1.1192	1.287641
cut3						
_cons	1.851093	.0451114	41.03	0.000	1.762676	1.939509

Stata does not have a command to fit a zero-inflated ordered logistic model; however, it can be easily done in the command `gidm` by changing the *string* of the `link(`*string*`)` option to `lgcl` as follows.

```
. gidm (tobacco female income age) (female income age parent i.religion),
> inflation(0) link(lgcl)
Iteration 0:   log likelihood = -16351.374
Iteration 1:   log likelihood = -13771.219  (not concave)
Iteration 2:   log likelihood = -13695.636
Iteration 3:   log likelihood = -13553.223
Iteration 4:   log likelihood = -13549.972
Iteration 5:   log likelihood = -13549.769
Iteration 6:   log likelihood = -13549.769
                                                Number of obs     =     15,000
                                                Wald chi2(3)      =    1390.29
Log likelihood = -13549.769                     Prob > chi2       =     0.0000
```

tobacco	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
tobacco						
female	-.3034075	.057619	-5.27	0.000	-.4163387	-.1904763
income	.2578177	.0070776	36.43	0.000	.243946	.2716895
age	-.2609338	.0141618	-18.43	0.000	-.2886904	-.2331771
inf_at_0						
female	-.29743	.0757984	-3.92	0.000	-.445992	-.1488679
income	-.1050113	.0128231	-8.19	0.000	-.1301442	-.0798784
age	.2049737	.0266435	7.69	0.000	.1527533	.2571941
parent	1.176933	.0756198	15.56	0.000	1.028721	1.325145
religion						
discourage..	-.5024576	.0946602	-5.31	0.000	-.6879881	-.3169271
_cons	.045011	.1238729	0.36	0.716	-.1977754	.2877975
cut1						
_cons	-.2605849	.0948355	-2.75	0.006	-.4464592	-.0747107
cut2						
_cons	1.909322	.0792913	24.08	0.000	1.753914	2.06473
cut3						
_cons	3.124181	.0852814	36.63	0.000	2.957032	3.291329

The results from the ZIOP and zero-inflated ordered logistic models suggest that the income is positively associated with tobacco use, while age and gender are negatively correlated. Furthermore, with smoking parents, one is expected to more likely be a smoker or less likely be a nonsmoker. Interestingly, the effect of income is two-fold: income increases the chance to be a nonsmoker; while once started smoking, higher income is correlated to more cigarettes consumed per day.

Because of different parameterization, the sign of coefficients using a probit link is opposite to that using a logit link, although the size of coefficients is very close (Moore 2013). As a simple illustration, we can extend the inflation part to include the value `1` to see whether `age` contributes to light smoking by using a logit link. According to

the result below, `age` also reduces the chance of inflation at the category of `1 to 7 cigarettes/day`.

```
. gidm (tobacco female income age) (income) (age), inflation(0 1) link(lgcl)
Iteration 0:   log likelihood = -16147.772
Iteration 1:   log likelihood = -15490.793  (not concave)
Iteration 2:   log likelihood =   -15236.3  (not concave)
Iteration 3:   log likelihood = -14576.662  (not concave)
Iteration 4:   log likelihood = -14508.358
Iteration 5:   log likelihood =  -14024.87  (not concave)
Iteration 6:   log likelihood = -13933.834
Iteration 7:   log likelihood = -13900.487
Iteration 8:   log likelihood = -13896.787
Iteration 9:   log likelihood = -13896.007
Iteration 10:  log likelihood = -13895.832
Iteration 11:  log likelihood = -13895.789
Iteration 12:  log likelihood = -13895.779
Iteration 13:  log likelihood = -13895.777
Iteration 14:  log likelihood = -13895.777
Iteration 15:  log likelihood = -13895.777
Iteration 16:  log likelihood = -13895.777
                                                Number of obs     =     15,000
                                                Wald chi2(3)      =      71.69
Log likelihood = -13895.777                     Prob > chi2       =     0.0000
```

tobacco	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
tobacco						
female	-.1874566	.1082809	-1.73	0.083	-.3996833	.0247701
income	.1106495	.013469	8.22	0.000	.0842507	.1370483
age	-.030557	.0270018	-1.13	0.258	-.0834795	.0223655
inf_at_0						
income	.0939271	.0030189	31.11	0.000	.0880102	.0998439
_cons	-1.127461	.0248864	-45.30	0.000	-1.176237	-1.078684
inf_at_1						
age	-.0680538	.0054307	-12.53	0.000	-.0786978	-.0574098
_cons	1.377042	.0318481	43.24	0.000	1.314621	1.439463
cut1						
_cons	-22.0358	7834.393	-0.00	0.998	-15377.16	15333.09
cut2						
_cons	-17.46215	758.249	-0.02	0.982	-1503.603	1468.679
cut3						
_cons	1.229678	.1634936	7.52	0.000	.9092367	1.55012

One may wonder what if the proportional odds assumption does not hold, for example, the effects of independent variables vary by categories of the dependent variable. We can fit an inflated multinomial logit or probit model by changing the *string* of the `link(`*string*`)` option to `lgml` or `pbml`, respectively. The output is as follows.

```
. gidm (tobacco female income age) (income) (income), inflation(0 3) link(lgml)
Iteration 0:   log likelihood =  -16528.45
Iteration 1:   log likelihood = -13998.131  (not concave)
Iteration 2:   log likelihood = -13830.753
Iteration 3:   log likelihood = -13813.182
Iteration 4:   log likelihood = -13770.342
Iteration 5:   log likelihood = -13757.115
Iteration 6:   log likelihood = -13752.886
Iteration 7:   log likelihood = -13750.136
Iteration 8:   log likelihood = -13749.212
Iteration 9:   log likelihood = -13748.643  (not concave)
Iteration 10:  log likelihood = -13748.365
Iteration 11:  log likelihood = -13748.217
Iteration 12:  log likelihood =  -13748.12
Iteration 13:  log likelihood = -13748.117
Iteration 14:  log likelihood = -13748.117
                                                Number of obs     =     15,000
                                                Wald chi2(3)      =     251.97
Log likelihood = -13748.117                     Prob > chi2       =     0.0000
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
cons_0						
female	1.215171	.158328	7.68	0.000	.9048542	1.525488
income	-.7047131	.0458645	-15.37	0.000	-.7946059	-.6148203
age	.5636431	.055056	10.24	0.000	.4557354	.6715508
_cons	1.075239	.3718305	2.89	0.004	.346465	1.804014
cons_1						
female	.4551338	.0953795	4.77	0.000	.2681934	.6420742
income	-.2465465	.0127899	-19.28	0.000	-.2716143	-.2214787
age	.1258706	.0245793	5.12	0.000	.077696	.1740453
_cons	2.902104	.142925	20.31	0.000	2.621976	3.182232
cons_2						
female	.2010796	.1076831	1.87	0.062	-.0099754	.4121345
income	-.0947244	.0139959	-6.77	0.000	-.1221558	-.0672931
age	.0287424	.0276511	1.04	0.299	-.0254528	.0829377
_cons	1.099314	.1610094	6.83	0.000	.7837415	1.414887
inf_at_0						
income	-.0108989	.0086969	-1.25	0.210	-.0279444	.0061467
_cons	.2640432	.0999661	2.64	0.008	.0681133	.4599732
inf_at_3						
income	.7523712	.4193545	1.79	0.073	-.0695484	1.574291
_cons	-17.36523	8.664376	-2.00	0.045	-34.34709	-.3833643

```
. gidm (tobacco female income age) (income) (income), inflation(0 3) link(pbml)
Iteration 0:   log likelihood = -16220.104
Iteration 1:   log likelihood = -13962.643  (not concave)
Iteration 2:   log likelihood = -13853.146
Iteration 3:   log likelihood = -13829.418
Iteration 4:   log likelihood = -13768.762
Iteration 5:   log likelihood = -13765.281
Iteration 6:   log likelihood = -13754.064
Iteration 7:   log likelihood =  -13751.16
Iteration 8:   log likelihood = -13750.203
Iteration 9:   log likelihood = -13749.161
Iteration 10:  log likelihood = -13749.134
Iteration 11:  log likelihood = -13748.623
Iteration 12:  log likelihood = -13748.554
Iteration 13:  log likelihood = -13748.525
Iteration 14:  log likelihood = -13748.517  (not concave)
Iteration 15:  log likelihood =   -13748.5  (not concave)
Iteration 16:  log likelihood = -13748.469  (not concave)
Iteration 17:  log likelihood = -13748.436  (not concave)
Iteration 18:  log likelihood = -13748.047
Iteration 19:  log likelihood =  -13747.92
Iteration 20:  log likelihood = -13747.913
Iteration 21:  log likelihood = -13747.913

                                              Number of obs     =     15,000
                                              Wald chi2(3)      =     250.85
Log likelihood = -13747.913                   Prob > chi2       =     0.0000
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
cons_0						
female	1.216532	.1584682	7.68	0.000	.9059403	1.527124
income	-.7033618	.0458969	-15.32	0.000	-.7933181	-.6134056
age	.5631134	.0551083	10.22	0.000	.4551032	.6711236
_cons	1.066075	.3724674	2.86	0.004	.3360528	1.796098
cons_1						
female	.4562997	.0955166	4.78	0.000	.2690906	.6435087
income	-.2451173	.0129632	-18.91	0.000	-.2705246	-.21971
age	.1251771	.0246397	5.08	0.000	.0768841	.1734701
_cons	2.89424	.143226	20.21	0.000	2.613522	3.174958
cons_2						
female	.2024173	.1078523	1.88	0.061	-.0089692	.4138039
income	-.0930536	.0142137	-6.55	0.000	-.120912	-.0651953
age	.0278977	.0277332	1.01	0.314	-.0264583	.0822537
_cons	1.090347	.1614318	6.75	0.000	.773946	1.406747
inf_at_0						
income	-.0068457	.0054366	-1.26	0.208	-.0175012	.0038098
_cons	.1655876	.0624756	2.65	0.008	.0431376	.2880376
inf_at_3						
income	.3094754	.1265657	2.45	0.014	.0614112	.5575396
_cons	-7.370308	2.448976	-3.01	0.003	-12.17021	-2.570403

Based on the results from both models, the effects of independent variables do vary across the categories with respect to the reference group. For instance, with respect to the reference category, the effect size of income reduces (for example, -0.705 versus -0.247 versus -0.095), while the category of frequency of smoking increases. Note that the estimated standard error of the intercept for inflation at 3 is large, although the z test reaches significance at the conventional 5% level, which is a sign that the inflation at 3 might not exist. In sum, the `gidm` command offers a flexible parameterization that allows for multiple inflated values as well as link functions.

6 Discussion

Traditionally, the Vuong (1989) test is used to evaluate goodness of fit for the inflated models. However, a recent study suggests that the Vuong test is not appropriate for testing possible inflation (Wilson 2015). Because Stata 15 removed the Vuong test from their `zip`, `zinb`, and `zioprobit` commands,[1] in the current study, we use only the Akaike information criterion (AIC) and Bayesian information criterion (BIC) as criteria to compare the goodness of fit across models. For instance, in the number of fish caught example, the AIC and BIC for the inflated and the regular models can be calculated as follows.

```
. webuse fish, clear
(Fictional fishing data)
. *ZIP model
. quietly gidm (count child camper) (persons), inflation(0) link(lgpo)
. display "AIC = " -2*e(ll)+2*e(rank) /*AIC*/
AIC = 2073.2168
. display "BIC = " -2*e(ll) + log(e(N))*e(rank) /*BIC*/
BIC = 2090.8241
. *Poisson model
. quietly poisson count child camper
. display "AIC = " -2*e(ll)+2*e(rank) /*AIC*/
AIC = 2723.1858
. display "BIC = " -2*e(ll) + log(e(N))*e(rank) /*BIC*/
BIC = 2733.7502
```

The values of the AIC and BIC for the ZIP model are much lower than that of the regular Poisson, which indicates the ZIP fits the data better.

When one optimizes the likelihood function of the GIDM, most of the models implemented in the `gidm` command require only `lf0` or `lf1` (Gould, Pitblado, and Poi 2010) to converge. However, sometimes numerical issues such as nonconvergence, or overflow, may occur if the empirical derivatives of the likelihood function are hard to evaluate numerically. The most common explanation for the numerical issues might be model misspecification, especially for the inflation part. Users should be cautious about the number of values and predictors enclosed for the inflation part. It is helpful to start with a simple model, for example, intercept only for the inflation part, and then build upon it.

1. See https://www.stata.com/help.cgi?j_vuong for details.

If the model is correctly specified, one way to improve numerical stability is to fit the model using `lf2` (Gould, Pitblado, and Poi 2010). Although it is feasible, the Hessian matrix may still be expensive or cumbersome to evaluate; for instance, the computational recourses required for calculating the second derivative for an ordinal or multinomial outcome with multiple inflations might be large. An alternative is to replace the gradient or Hessian with less expensive approximations. For instance, suppose a multinomial random variable Y has inflated probabilities at categories $k_1, \ldots, k_m \in \{0, 1, 2, \ldots\}$; the PMF for each of the observations could be written as

$$p(Y = k|\boldsymbol{\beta}_k, \pi_i, 1 \leq i \leq m) = \begin{cases} \pi_i + (1 - \sum_{i=1}^m \pi_i) \times p(k) & \text{if } k = k_1, \ldots, k_m \\ (1 - \sum_{i=1}^m \pi_i) \times p(k) & \text{if } k \neq k_i, 1 \leq i \leq m \end{cases}$$

where $p(k) = p_k = \{\exp(\boldsymbol{x}_s\boldsymbol{\beta}_k)\}/\{1 + \sum_{k=1}^K \exp(\boldsymbol{x}_s\boldsymbol{\beta}_k)\}$, $p_K = 1 - \sum_{k=1}^{K-1} p_k$, and $\pi_i = 1\{1 + \exp(-\boldsymbol{z}_s\boldsymbol{\gamma}_i)\}$. If we define the indicator as $J_i := 1_{k=k_1,\ldots,k_m}$, then the log likelihood is

$$\log L = \sum_{k=1}^K \begin{pmatrix} J_i \times I(y_s = k) \times \log\{\pi_i + (1 - \sum_{i=1}^m \pi_i) \times p_k\} + \\ (1 - J_i) \times I(y_s = k) \times \{\log(1 - \sum_{i=1}^m \pi_i) + \log(p_k)\} \end{pmatrix} + c$$

The first derivative with respect to p_j is as follows.

$$\frac{\partial \log L}{\partial p_j} = \sum_{k=1}^K \begin{pmatrix} J_i \times I(y_s = k) \times \frac{1 - \sum_{i=1}^m \pi_i}{\pi_i + \left(1 - \sum_{i=1}^m \pi_i\right) \times p_k} \frac{\partial p_k}{\partial p_j} + \\ (1 - J_i) \times I(y_s = k) \times \left(\frac{1}{p_k} \frac{\partial p_k}{\partial p_j}\right) \end{pmatrix}$$

Because the term

$$\frac{1 - \sum_{i=1}^m \pi_i}{\pi_i + (1 - \sum_{i=1}^m \pi_i) \times p_k} = \frac{1}{\frac{\pi_i}{1 - \sum_{i=1}^m \pi_i} + p_k} \leq \frac{1}{p_k}$$

then

$$\frac{\partial \log L}{\partial p_j} \leq \sum_{k=1}^K \left\{ I(y_s = k) \times \frac{1}{p_k} \frac{\partial p_k}{\partial p_j} \right\}$$

It reaches to equality if π_i is zero. Thus, a majorization of the first derivative can be derived. If we look further, the diagonal elements of the second derivative yield

$$\frac{\partial^2 \log L}{\partial p_j^2} = \sum_{k=1}^K \left(\begin{array}{c} J_i \times I(y_s = k) \times \\ \left\{ \frac{1 - \sum_{i=1}^m \pi_i}{\pi_i + \left(1 - \sum_{i=1}^m \pi_i\right) \times p_k} \frac{\partial p_k}{\partial p_j^2} - \left(\frac{1 - \sum_{i=1}^m \pi_i}{\pi_i + \left(1 - \sum_{i=1}^m \pi_i\right) \times p_k} \frac{\partial p_k}{\partial p_j} \right)^2 \right\} + \\ (1 - J_i) \times I(y_s = k) \times \frac{1}{p_k} \times \frac{\partial p_k}{\partial p_j^2} \end{array} \right)$$

However, $(\partial^2 \log L)/(\partial p_j^2)$ is negative only if $p_k < (1/2)$. If $p_k \geq (1/2)$, the Hessian might not be positive definite. Therefore, it is necessary to check model specification to make sure that $p_k < (1/2)$. If we define the inflation factor as $F = \pi_i/(1 - \sum_{i=1}^m \pi_i)$,

figure 2 summarizes the relationship between the size of the inflation factor and the second derivative. Numerical issues may emerge if the inflation factor is small and the probability p_i is large. In other words, if data show a high percentage for one category, but the relative chance of inflation for that category is small, the model is likely to have nonconverge issues.

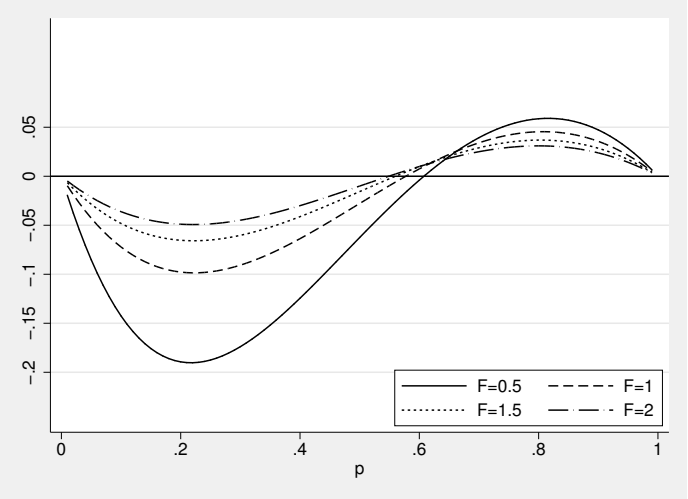

Figure 2. Relationship between the inflation factor and the second derivative

According to our experiment, even under a mild or severe situation, for example, the percentage of $p_k \geq (1/2)$ is higher than 65%, all three methods, `lf0`, `lf1`, and `lf2`, yield reasonable estimates, with a convergence rate of about 90%, especially for the `lf1` method, which requires fewer numbers of iterations and Hessian calls.

The current implementation can be further extended in several ways. A direct extension is to allow for continuous or truncated outcomes such as linear or zero-truncated models. Another useful extension might be to support predictions after estimation. At present, we are working on adding predictions, residuals, influence statistics, and the like. In addition, although the option `vce()` offers robust standard errors to account for clustering, a more efficient method would be to allow random effects. Therefore, future work will also cover random- and fixed-effects models.

Nevertheless, the GIDM framework is a powerful tool for analyzing inflated outcomes. Despite the recent development of the generalized inflation models, the modeling choices for Stata users are limited. We developed the `gidm` command, which supports various distributions, to fit the most recent version of the inflated models—the GIDM. Meanwhile, the traditional single-value inflated models are also available as special cases in the `gidm` command. We hope the `gidm` command developed in this article will be useful for Stata users when dealing with inflation problems.

7 Acknowledgment

This work was partially supported by the Multiple Year Research Grant (MYRG2018-00222-FSS) funded by RSKTO, University of Macau. Views expressed are those of the authors.

8 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 19-3
. net install st0574     (to install program files, if available)
. net get st0574         (to install ancillary files, if available)
```

9 References

Bagozzi, B. E., D. W. Hill, W. H. Moore, and B. Mukherjee. 2015. Modeling two types of peace: The zero-inflated ordered probit (ZiOP) model in conflict research. *Journal of Conflict Resolution* 59: 728–752.

Bagozzi, B. E., and B. Mukherjee. 2012. A mixture model for middle category inflation in ordered survey responses. *Political Analysis* 20: 369–386.

Begum, M., A. Mallick, and N. Pal. 2014. A generalized inflated Poisson distribution with application to modeling fertility data. *Thailand Statistician* 12: 135–159.

Cai, T., Y. Xia, and Y. Zhou. Forthcoming. Generalized inflated discrete models: A strategy to work with multimodal discrete distributions. *Sociological Methods & Research*.

Diallo, A. O., A. Diop, and J.-F. Dupuy. 2018. Analysis of multinomial counts with joint zero-inflation, with an application to health economics. *Journal of Statistical Planning and Inference* 194: 85–105.

Diop, A., A. Diop, and J.-F. Dupuy. 2016. Simulation-based inference in a zero-inflated Bernoulli regression model. *Communications in Statistics—Simulation and Computation* 45: 3597–3614.

Gould, W., J. Pitblado, and B. Poi. 2010. *Maximum Likelihood Estimation with Stata*. 4th ed. College Station, TX: Stata Press.

Hall, D. B. 2000. Zero-inflated Poisson and binomial regression with random effects: A case study. *Biometrics* 56: 1030–1039.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054.

Lambert, D. 1992. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* 34: 1–14.

Lin, T. H., and M.-H. Tsai. 2013. Modeling health survey data with excessive zero and K responses. *Statistics in Medicine* 32: 1572–1583.

Melkersson, M., and C. Olsson. 1999. Is visiting the dentist a good habit?: Analyzing count data with excess zeros and excess ones. Umeå Economic Studies No. 492, Umeå University, Umeå, Sweden.

Moore, C. 2013. Lecture notes: An introduction to logistic and probit regression models. University of Texas, Austin. https://liberalarts.utexas.edu/prc/_files/cs/Fall2013_Moore_Logistic_Probit_Regression.pdf.

Poston, D., Jr., and S. L. McKibben. 2003. Using zero-inflated count regression models to estimate the fertility of U.S. women. *Journal of Modern Applied Statistical Methods* 2(2): Article 10.

Ridout, M., J. Hinde, and C. G. B. DeméAtrio. 2001. A score test for testing a zero-inflated Poisson regression model against zero-inflated negative binomial alternatives. *Biometrics* 57: 219–223.

Vieira, A. M. C., J. P. Hinde, and C. G. B. Demetrio. 2000. Zero-inflated proportion data models applied to a biological control assay. *Journal of Applied Statistics* 27: 373–389.

Vuong, Q. H. 1989. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica* 57: 307–333.

Wilson, P. 2015. The misuse of the Vuong test for non-nested models to test for zero-inflation. *Economics Letters* 127: 51–53.

About the authors

Yiwei Xia is an assistant professor at the School of Law, Southwestern University of Finance and Economics, Chengdu, China. His research interests include quantitative methodology, juvenile delinquency, and substance abuse.

Yisu Zhou is an associate professor in the Faculty of Education at the University of Macau. He is interested in quantitative social sciences in general. His research expertise is educational policy in the greater China region. His past research projects use large-scale assessment data both internationally and domestically, covering issues such as the social environment of learning, teacher education and labor market, and social stratification in schools. He is currently working on school social segregation and achievement gaps in Chinese societies.

Tianji Cai is an assistant professor of sociology at the University of Macau. Cai's research focuses on quantitative research methods, text mining, substance abuse, and social networks.