



The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.

The Stata Journal (2017)
17, Number 3, pp. 687–703

Estimating measures of multidimensional poverty with Stata

Daniele Pacifico
OECD
Paris, France
daniele.pacifico@oecd.org

Felix Poege
Max Planck Institute for Innovation and Competition
Munich, Germany
felix.poege@ip.mpg.de

Abstract. In this article, we describe the multidimensional poverty measures developed by [Alkire and Foster \(2011, *Journal of Public Economics* 95: 476–487\)](#) and show how they can be computed with Stata by using the `mpi` command.

Keywords: st0492, mpi, multidimensional poverty, Alkire–Foster method

1 Introduction

One’s poverty status is often the result of a plurality of simultaneous deprivations that go beyond the shortage of financial resources. For instance, a person who is not classified as income-poor can still suffer other deprivations, such as poor health, malnutrition, little schooling, or inadequate housing—all conditions that reduce well-being while increasing the risk of marginalization and social exclusion.

[Alkire and Foster \(2011\)](#) have proposed a methodology that summarizes a plurality of imperfectly overlapping deprivation domains into a consistent parametric class of multidimensional poverty indices. These indices can be used in a variety of policy-relevant applications, such as creating measures of well-being, monitoring and evaluating anti-poverty programs, and improving the targeting of in-kind and cash benefits.¹ The Alkire–Foster (AF) measures build on the Foster–Greer–Thorbecke (FGT) indices introduced in [Foster, Greer, and Thorbecke \(1984\)](#). Accordingly, they can be decomposed by population subgroups (for example, ethnicity or geographic area) and deprivation domains (for example, education, income, or health), a feature that makes them particularly suitable for policy analysis. Similar to the FGT measures, the AF measures depend on a parameter α that ensures they satisfy a broad range of multidimensional poverty-measurement axioms, such as replication invariance, symmetry, poverty focus, and weak monotonicity.²

1. For a review of applications, refer to the Oxford Poverty and Human Development Initiative, the United Nations Development Programme’s Human Development Reports, [Alkire \(2013\)](#), and [OECD \(2015\)](#).

2. For $\alpha \geq 0$, the AF indices satisfy decomposability, replication invariance, symmetry, poverty and deprivation focus, weak and dimensional monotonicity, nontriviality, normalization, and weak rearrangement. For $\alpha > 0$, they also satisfy monotonicity, and for $\alpha \geq 1$, they also satisfy the axiom of weak transfer. See [Alkire and Foster \(2011\)](#) for definitions and proofs.

In this article, we review the AF method and show how to apply it in Stata with the `mpi` command.³ An important feature of `mpi` is its flexibility: depending on the type of data, `mpi` estimates the whole range of AF multidimensional poverty measures for arbitrary values of α and computes their decompositions by deprivation indicators and population subgroups. `mpi` allows for an indefinite number of indicators, with the possibility to use a flexible weighting structure for each of them. The survey design is also fully accounted for when computing the indices and the corresponding variance–covariance matrices.⁴

The remainder of this article proceeds as follows. Section 2 overviews the AF method. Section 3 overviews the `mpi` command and the main options. Section 4 concludes with an empirical application based on the data used in Alkire and Foster (2011).

2 The AF method

In this section, we overview the main concepts used in the AF framework to derive the related class of poverty measures. The AF method can be divided into two sequential parts: the identification of poor individuals and the measure of poverty based on such identification.

2.1 Identifying the poor

Let us consider a sample of N individuals and $D \geq 2$ deprivation indicators. Indicators related to the same area of deprivation can be grouped into deprivation domains. For example, the health domain can be identified in the data with two indicators, say, the number of doctor visits and the distance to the closest medical facility. Let \mathbf{Y} be an $N \times D$ matrix whose entry y_{ij} denotes the level of indicator j for individual i . The $1 \times D$ vector $\mathbf{z} = (z_1, \dots, z_D)$ contains the deprivation cutoffs of the D indicators and is used to determine if a person is deprived in each of the D dimensions. Assume that, for an indicator j and individual i , the deprivation occurs when y_{ij} falls strictly below the respective cutoff, that is, $y_{ij} < z_j$.

Indicators can enter the analysis with different weights depending on their policy relevance. Weights are collected in a $1 \times D$ vector $\mathbf{w} = (w_1, \dots, w_D)$, with $0 < w_j < 1$ and $\sum_{j=1}^D w_j = 1$. For instance, if each indicator is viewed as having equal importance, all weights will be equal to $1/D$.

3. We are grateful to two anonymous referees for useful comments and suggestions on this manuscript and the related Stata command.

4. Abdelkrim and Duclos (2007) (distributive analysis Stata package) also permits the estimation of the AF indices. However, this program faces certain limitations. It permits a maximum of 10 indicators, does not allow grouping indicators into policy domains, and does not allow computing the indices with nonstandard α values. The program is not written as a proper estimation command, which means that standard postestimation commands such as `test` cannot be used. Moreover, the distributive analysis Stata package does not have a return list, and this forbids a series of applications that can be useful in applied contexts. For example, computing bootstrapped standard errors would be infeasible.

Let \mathbf{g}^0 be the $N \times D$ matrix whose entries are given by $g_{ij}^0 = w_j$ if $y_{ij} < z_j$ and by 0 otherwise. This is called the deprivation matrix in the Alkire and Foster (2011) framework because, for each individual of the population, it contains the policy relevance of each deprivation when such deprivation occurs. The row sum of \mathbf{g}^0 represents therefore the number of weighted deprivations faced by individual i : $c_i = \sum_{j=1}^D g_{ij}^0$.

With cardinal indicators, the matrix of deprivations \mathbf{g}^0 can be complemented with the matrix of normalized deprivation gaps, \mathbf{g}^1 , whose entries are given by $g_{ij}^1 = g_{ij}^0(z_j - y_{ij})/z_j$. In other words, g_{ij}^1 represents a measure of the extent to which individual i is deprived in dimension j whenever $y_{ij} < z_j$. More generally, for any α , let us define the matrix \mathbf{g}^α by raising each entry of \mathbf{g}^1 to the power of α : $g_{ij}^\alpha = g_{ij}^0 \{(z_j - y_{ij})/z_j\}^\alpha$. Hence, similarly to the FGT class of poverty measures, the higher the value of α , the higher the importance of the elements of \mathbf{g}^α with the biggest gaps, that is, the focus on the poorest among the poor in the calculation of the overall index.

Let us define $0 < k < 1$ as the poverty cutoff. This value is key in the AF method because it represents the extent of weighted deprivations a person must exceed to be classified as poor. For example, if there are 10 indicators of equal importance and $k = 0.4$, a person is considered poor if that person experiences five or more deprivations simultaneously. The use of indicator and poverty cutoffs is what justifies the term “dual-cutoff approach” when referring to the AF method.

Let us define the identification function $\rho_k(y_i, \mathbf{z}) = 1$ if $c_i > k$ and 0 otherwise. This function plays a key role in the AF framework because it modifies the entries of matrix \mathbf{g}^α as $g_{ij}^\alpha \rho_k(y_i, \mathbf{z})$; so if person i is not considered poor, then the row vector \mathbf{g}_i^α is replaced with 0s. Alkire and Foster (2011) define the resulting matrix as $\mathbf{g}^\alpha(k)$ and call it the censored deprivation matrix.

2.2 Measuring multidimensional poverty

The simplest index of multidimensional poverty in the AF framework is the multidimensional headcount ratio, which measures the incidence of poverty in the population:

$$H = \frac{\sum_{i=1}^N \rho_k(y_i, \mathbf{z})}{N} = \frac{q}{N}$$

The numerator is the number of poor individuals identified with the identification function defined in section 2.1, and N is the population size. Despite its simplicity and its widespread use in the policy debate, H does not have the desirable property of increasing when a poor person becomes deprived in a new dimension.⁵

5. This is the measurement axiom of dimensional monotonicity in the AF framework.

An index that increases with the number of deprivations experienced by the poor individuals can be derived from the censored deprivation matrix, $\mathbf{g}^0(k)$. Let $|\mathbf{g}^0(k)|$ be the sum of all entries of matrix $\mathbf{g}^0(k)$: $|\mathbf{g}^0(k)| = \sum_{i=1}^N \sum_{j=1}^D g_{ij}^0(k)$. Alkire and Foster (2011) define the index A as the ratio between the number of deprivations faced by the poor individuals, $|\mathbf{g}^0(k)|$, and the number of poor individuals (q):⁶ $A = \{|\mathbf{g}^0(k)|\}/q$.

A poverty measure that accounts for both the incidence (H) and the breadth (A) of simultaneous deprivations can be derived from the product of H and A :

$$M_0 = H \times A = \frac{|\mathbf{g}^0(k)|}{N}$$

Alkire and Foster (2011) define M_0 as the adjusted multidimensional headcount ratio, also known as the multidimensional poverty index.

The multidimensional poverty index can be computed with binary, ordinal, or real-valued data.⁷ However, when real-valued indicators are available, M_0 can be complemented with other indices that can also account for the depth and severity of each deprivation. Let $|\mathbf{g}^1(k)|$ be the sum of the poverty gaps of poor individuals. The average poverty gap across the extent of all possible deprivations faced by the poor is $G = \{|\mathbf{g}^1(k)|\}/\{|\mathbf{g}^0(k)|\}$. A poverty measure that jointly considers the incidence of poverty (H), the average range of deprivations (A), and the average depth across deprived dimensions (G) can be computed as

$$M_1 = M_0 \times G = \frac{|\mathbf{g}^1(k)|}{N}$$

More importantly, M_1 respects the traditional monotonicity axiom; that is, it increases as a poor person becomes more deprived in a given dimension.

Following Foster, Greer, and Thorbecke (1984), ideal poverty measures should also respect the transfer principle; that is, they should increase at a faster rate when the depth of deprivation gets worse for those individuals who are already highly deprived. An index with such property can be easily derived within the AF framework by simply substituting $|\mathbf{g}^1(k)|$ with $|\mathbf{g}^2(k)|$ in the calculation of the G index. This leads to a measure of average severity of deprivations: $S = \{|\mathbf{g}^2(k)|\}/\{|\mathbf{g}^0(k)|\}$.

A multidimensional poverty measure that jointly considers all the aspects defined above can be derived from the product of M_0 and S :

$$M_2 = M_0 \times S = \frac{|\mathbf{g}^2(k)|}{N}$$

6. The denominator of index A in the original Alkire and Foster (2011) contribution is qD . In the context of this article, using indicator weights that sum up to 1 implies that the extent of all possible deprivations for the generic individual i is also standardized to 1.

7. Exercise caution when using ordinal variables because the resulting indicators M_1 , M_2 , and M_α will depend on the coding of the input variables. This can be seen easily by adding a constant to an ordinal variable and adjusting the threshold accordingly.

More generally, the AF class of multidimensional poverty measures is given by

$$M_\alpha = \frac{|\mathbf{g}^\alpha(k)|}{N}, \alpha \geq 0$$

A key property of the AF measures M_α is the perfect decomposability by population subgroups and indicators. Perfect decomposability into population subgroups means that the overall measure can be obtained as the weighted average of subgroup poverty levels with weights given by the subgroup population shares

$$M_\alpha = \sum_{g=1}^G \frac{N_g}{N} M_{\alpha,g}$$

where $M_{\alpha,g}$ is the index for subgroup g and N_g is the corresponding population size. The percentage contribution of group g is therefore $C_{\alpha,g} = (N_g/N)(M_{\alpha,g}/M_\alpha)$.

The AF class of poverty measures can be further decomposed by indicators of deprivation. Let $|\mathbf{g}_j^\alpha(k)|$ be the sum of the j column entries of $\mathbf{g}^\alpha(k)$. Then, $M_\alpha = \sum_{j=1}^D |\mathbf{g}_j^\alpha(k)|/N$. The percentage contribution of each indicator to the overall measure is therefore $CI_{\alpha,j} = \{|\mathbf{g}_j^\alpha(k)|\}/(N \times M_\alpha)$. The contribution of a group of indicators follows simply as the sum of the contributions of the individual indicators.

3 The mpi command

The `mpi` command estimates the AF poverty measures described in section 2.2 and provides the exact decomposition by deprivation indicators. It computes the variance–covariance matrix of the estimates and can account for complex survey designs.⁸ Users can provide the weighting structure of the indicators in a flexible way and define the thresholds for real-valued indicators directly in the command line. `mpi` also calculates the decomposition by population subgroups as well as the contribution of each deprivation indicator in each subgroup. When real-valued indicators are available, `mpi` computes the entire parametric class of AF poverty measures for arbitrary values of α and provides the decomposition by population subgroups and indicators for each M_α .

An important characteristic of `mpi` is the possibility to group indicators into policy domains. This does not affect the statistical derivation of the AF measures, but facilitates the interpretation of the results. Let us consider two deprivation domains, say, monetary poverty and health. Monetary poverty can be identified by one indicator, for example, household income, and health by two indicators, for example, the number of doctor visits and the distance to the closest medical center. In this example, there are three deprivation indicators for two policy domains, and `mpi` provides information at both the indicator and the domain level.

8. The variance–covariance matrices are computed with the Stata built-in commands `mean` and `ratio`. This is possible because the AF measures can be derived as the result of means or ratios from suitable transformations of the indicators.

3.1 Syntax

The generic syntax for `mpi` is

```
mpi d1(varlist) [d2(varlist) ... w1(numlist) w2(numlist) ... t1(thresholds)
  t2(thresholds) ...] [if] [in] [weight], cutoff(#) [by(varname) svy
  subpop([varlist] [if])] alpha(numlist) level(#) categories(#)
  depriveddummy(varname) deprivedscore(varname) nosummary
  nodecomposition postoption]
```

`pweights` and `fweights` are allowed; see [U] 11.1.6 **weight**.

`d1(varlist)`, `d2(varlist)`, ... denote deprivation domains, for example, health, housing, or education. Users can specify an indefinite number of domains and, for each domain, an indefinite number of indicators. `d1()` is required. Observations with missing values are excluded from the estimation sample. If users do not specify deprivation thresholds in the corresponding option (`t1(thresholds)`; see below), then `mpi` treats the indicators as binary variables, taking values 1 (deprived) or 0 (not deprived). If users specify deprivation thresholds, then `mpi` considers the related indicators as real-valued variables. Deviation from these rules will result in an error message; thus, combining binary and real-valued indicators is only possible when selecting the corresponding threshold also for the binary indicators (that is, 0 or any value between 0 and 0.5).⁹

`w1(numlist)`, `w2(numlist)`, ... denote the weights of the indicators. Weights are numbers ranging between 0 and 1 and must sum up to 1; when this is not the case, `mpi` gives an error. These options use as many weights as indicators, and the order in the `numlists` must follow the order of the corresponding indicators. The default option is equal weighting, where domains are equally weighted and indicators in each domain are also equally weighted. The two command lines below are therefore equivalent:

```
mpi d1(ind1 ind2) d2(ind3), cutoff(0.74)

mpi d1(ind1 ind2) d2(ind3) w1(0.25 0.25) w2(0.5), cutoff(0.74)
```

9. Alkire and Foster (2011) suggest caution when combining binary and real-valued indicators in the calculation of multidimensional poverty measures with $\alpha > 0$. In these cases, binary indicators would automatically receive a higher weight than real-valued variables in the calculation of the M_α measures based on the normalized poverty gaps, because these gaps would always be the highest possible. If one wants to combine binary and real-valued indicators, a different weighing structure of each indicator can counterbalance the implicit higher weight of binary indicators. See Alkire and Foster (2008, 18–20) for a discussion. `mpi` detects real-valued indicators based on the number of different values characterizing the variable. Indicators with more than 20 different values are considered real-valued variables, and users get an alert when the variable has fewer values. The threshold of 20 values can be changed using the `categories()` option. According to Alkire and Foster (2011), ordinal and categorical indicators should not be included in the calculation of the AF measures with $\alpha > 0$. Instead, M_0 provides for both meaningful comparisons and favorable axiomatic properties when data are ordinal or categorical, so long as they can be recoded into dichotomous indicators.

`t1(thresholds)`, `t2(thresholds)`, ... denote the deprivation thresholds for the indicators of each domain. These options are only required when using real-valued indicators. The deprivation occurs when the indicator is strictly below the threshold.

The command line below shows an example of different deprivation thresholds: for `ind1`, `ind2`, and `ind3`, the deprivation occurs for values that are strictly below 4, 3.5, and 5, respectively.

```
mpi d1(ind1 ind2) d2(ind3) t1(4 3.5) t2(5), cutoff(0.74)
```

Mixing different types of indicators (binary/ordinal and real-valued) is generally not recommended. See [Alkire and Foster \(2008, 2011\)](#) for a discussion on this point.

3.2 Options

`cutoff(#)` is required and specifies a number between 0 and 1, above which the individual is considered poor. Following the approach outlined in section 2, `mpi` computes for each individual the weighted sum of the indicators and classifies an individual as poor only if the resulting score is higher than the selected poverty cutoff. Weights are specified in the corresponding `mpi` option (see above); if no weights are specified, then `mpi` assumes equal weights at the domain level and within each domain. Hence, when the number of indicators is equal to the number of domains and the indicators have equal weights, the poverty cutoff will simply indicate the percentage of simultaneous deprivations above which a person is considered poor.

Let us consider an example. The command line below uses three indicators and a poverty cutoff of 0.66. Because no thresholds are specified, `mpi` assumes that the indicators are all binary variables. Because no weights are specified, `mpi` assumes equal weights at the domain level and within each domain. Therefore, in the example below, a person is considered poor if that person faces at least two deprivations, because in that case, the weighted sum of two deprivations would be just above the poverty cutoff.¹⁰

```
mpi d1(ind1 ind2 ind3), cutoff(0.66)
```

Let us now consider the case of three indicators and two deprivation domains, the first containing two indicators (`ind1` and `ind2`) and the second containing one indicator (`ind3`). Because weights are not specified, `mpi` assumes equal weights at the domain level and within each domain. In the example below, each domain therefore has a weight of 0.5, and the first two indicators each have a weight of 0.25. Given a poverty cutoff of 0.74, an individual deprived in `ind1` and `ind3` would therefore be considered poor because the weighted sum of those deprivations would be just above the poverty cutoff.

```
mpi d1(ind1 ind2) d2(ind3), cutoff(0.74)
```

10. Note that the command `mpi d1(ind1 ind2 ind3), cutoff(0.66)` is equivalent to the command `mpi d1(ind1) d2(ind2) d3(ind3), cutoff(0.66)`.

by(*varname*) requires **mpi** to compute the decomposition of the AF measures by categories of *varname*. The variable must be numeric. Missing values are excluded from the estimation sample.

svy requires **mpi** to account for the survey design. The survey design must be declared through **svyset** (see [SVY] **svyset**) before using **mpi**. If the only information about the survey design relates to the sampling weights, then the user can supply **svyset** with such information and use the **svy** option of **mpi**. Equivalently, the user can specify the sampling weights in the command line by using the standard syntax.

subpop(*varlist* [*if*]) requires **mpi** to perform subpopulation estimation on the sample. See [SVY] **svyset** for details. This option requires users to specify the **svy** option. In this case, **if** and **in** are no longer allowed.

alpha(*numlist*) triggers the computation of additional nonstandard indices M_α . This option is only possible with real-valued indicators. Measures M_1 and M_2 are computed by default. Hence, the option **alpha**(3) means that **mpi** computes M_1 , M_2 , and M_3 . Decompositions by indicators and population subgroups are also computed for the new M_α 's.

level(#) specifies the confidence level, as a percentage, for confidence intervals. The default is **level**(95) or as set by **set level**.

categories(#) changes how **mpi** detects real-valued indicators by counting the number of different values characterizing the variable. The default is **categories**(20). When a variable has fewer than 20 different values, **mpi** shows an error message.¹¹

depriveddummy(*varname*) generates a binary variable called *varname* that is equal to 1 for individuals who are multidimensionally deprived and equal to 0 otherwise.

deprivedscore(*varname*) generates a variable called *varname* containing the multidimensional deprivation score.

nosummary suppresses the display of the summary table at the beginning of **mpi**'s output.

nodecomposition suppresses the computation and display of the decompositions along the lines of the domains and indicators. This slightly increases the execution speed.

postoption defines the results that **mpi** stores in **e(b)** and **e(V)**. This option is useful if one intends to run standard postestimation commands, such as **test**, after **mpi**, or handle the **mpi** output easily for creating tables and graphs.¹² Users may specify only one of the options below at a time. However, independently of the selected option, the **ereturn** list will always return the full set of results, all conveniently stored as matrices and labeled with a standardized notation (see section 3.3).

11. See section 2.2 and footnote 9 for discussions of why this is necessary.

12. When a given indicator is missing, the related standard errors and covariances will enter as 0 in **e()**, while they will be shown as missing in the displayed results. This depends on the output of the built-in commands **mean** and **ratio**, which **mpi** runs internally to compute the estimates and the related variance–covariance matrix.

`postmain` returns the main indicators H , M_0 , M_1 , M_2 , and all M_α . This is the default.

`postadditional` returns the additional indicators A , S , and G .

`postindicators` returns the decomposition by indicators.

`postdomains` returns the decomposition by domains.

`postbymain` returns the main indicators over by-groups.

`postbyproportion` returns the proportional contributions of the by-groups to the main indicators.

`postbyindicators` returns the decomposition both by indicators and over by-groups.

`postbydomains` returns the decomposition both by domains and over by-groups.

3.3 Stored results

`mpi` stores the following in `e()`:

Scalars

`e(N)` number of observations

Macros

`e(cmd)` `mpi`
`e(cmdline)` command as typed
`e(properties)` `b V`

Matrices

`e(b)` results stored in this vector depend on the selected `postoption`
`e(V)` variance-covariance matrix of `e(b)`
`e(mpi_main)` estimates of H , M_0 , M_1 , M_2 , and all M_α
`e(mpi_main.V)` variance-covariance matrix of `e(mpi_main)`
`e(mpi_add)` estimates of A , G , and S
`e(mpi_add.V)` variance-covariance matrix of `e(mpi_add)`
`e(ind)` contribution of each indicator to M_0 , M_1 , M_2 , and M_α
`e(ind.V)` variance-covariance matrix of `e(ind)`
`e(dom)` contribution of each domain to M_0 , M_1 , M_2 , and M_α
`e(dom.V)` variance-covariance matrix of `e(dom)`

Functions

`e(sample)` observations used for the computation of `mpi`

When the `by()` option is specified, `mpi` also stores the following in `e()`:

Matrices

`e(by_mpi)` H , M_0 , M_1 , M_2 , and M_α by level of the by-variable
`e(by_mpi.V)` variance-covariance matrix of `e(by_mpi)`
`e(by_mpi_pc)` proportional contribution of every level of the by-variable to the main indicators
`e(by_mpi_pc.V)` variance-covariance matrix of `e(by_mpi_pc)`
`e(by_ind)` contribution for each indicator to M_0 , M_1 , M_2 , and M_α by level of the by-variable
`e(by_ind.V)` variance-covariance matrix of `e(by_ind)`
`e(by_dom)` contribution for each domain to M_0 , M_1 , M_2 , and M_α by level of the by-variable
`e(by_dom.V)` variance-covariance matrix of `e(by_dom)`

When a given indicator is missing, the related standard errors and covariances will enter as 0 in `e()`, while they will appear as missing in the displayed results. This depends on the output of the built-in Stata commands `mean` and `ratio`, which `mpi` runs internally to compute the indices and the variance–covariance matrices.

4 Empirical applications

For the empirical application of `mpi`, we use the 2000 Indonesian Family Life Survey. Alkire and Foster (2011) use the same data, which can be freely downloaded with the related Stata codebook.¹³

The analysis applies the same settings of the original Alkire and Foster (2011) contribution. We consider all adults above 19 years old and consider three deprivation indicators: household expenditure (`exp`), body mass index (BMI), and years of schooling (`educ`). Expenditure variables are adjusted by the square root of household size. The deprivation thresholds are the following: expenditure below 150,000 Rupiah, a BMI lower than 18.5 kg/m², and less than 6 years of schooling. The final sample consists of 17,678 individuals, whereas the original Alkire and Foster (2011) contribution has 19,752. The difference is arguably due to the calculation of years of schooling from the raw data and the related treatment of implausible and missing values. The following table shows the first five rows of the dataset:

```
. use indonesia_2000.dta
. list hid id sex BMI exp ex_food ex_nofood educ weights in 1/5, table
```

	hid	id	sex	BMI	exp	ex_food	ex_nof-d	educ	weights
1.	1	1	Female	21	482947	462731	20217	3	0.85
2.	1	2	Male	23	482947	462731	20217	7	0.60
3.	2	3	Female	17	133736	131532	2205	0	0.95
4.	2	4	Male	20	133736	131532	2205	3	0.91
5.	3	5	Male	22	165981	154333	11647	4	0.95

The variables `hid` and `id` are the household and person identifiers, `weights` represents the survey weights, and `ex_food` and `ex_nofood` represent food and nonfood household expenditures. In the first application, we show how to use `mpi` with binary indicators. First, we construct three binary indicators using the deprivation thresholds defined above, and then we estimate the AF measure with a poverty cutoff equal to 0.66. This means that only persons with at least two simultaneous deprivations are considered poor.

```
. generate exp_i = (exp < 150000)
. generate educ_i = (educ < 6)
. generate BMI_i = (BMI < 18.5)
```

13. Strauss et al. (2004); material available at <http://www.rand.org/labor/FLS/IFLS/ifls3.html>.

```
. mpi d1(exp_i) d2(educ_i) d3(BMI_i) [pw=weights], cutoff(0.66)
```

Summary of mpi indicators

Indicator	Type	Weight	Deprived
Domain 1			
exp_i	Binary	.33	31.477 %
Domain 2			
educ_i	Binary	.33	38.318 %
Domain 3			
BMI_i	Binary	.33	16.006 %

Deprived: Percentage of individuals whose indicator values are below the threshold.

Main results

N = 17678

		Coef.	Std. Err.	[95% Conf. Interval]	
Main					
	H	0.229	0.004	0.222	0.237
	M0	0.166	0.003	0.161	0.172
Additional					
	A	0.725	0.002	0.720	0.729

Note: Adjusted Multidimensional Headcount

M0 = H*A

Indicator	M0
domain 1	
exp_i	0.383
domain 2	
educ_i	0.412
domain 3	
BMI_i	0.204
Total	1.000

Contribution of each indicator (%)

Type ereturn list to see the list of saved results and more information on the estimation sample.

The first part of the `mpi` output is a table with a summary of the deprivation indicators, which are organized in deprivation domains. For each indicator, `mpi` shows the type (for example, binary or real-valued), the policy weight (equal weights in this example, the default option), and the share of deprived individuals.

The second table shows the AF poverty measures with the related standard errors. This table is divided into two parts, containing the estimated H and M_α parameters (upper part) and the related subindices (bottom part). Because there are only binary indicators in this example, `mpi` computes only M_0 , which is derived as the product of indices H (the incidence of the poor in the population) and the subindex A (the average intensity of simultaneous deprivations among the poor).

The third table shows the proportional contribution of each indicator to the overall index. In this example, the deprivation in household expenditures accounts for 38.3% of the overall value of M_0 .

The next example shows how to allow for a different weighting structure in the policy relevance of each deprivation indicator. It also shows how to decompose the AF measures by population subgroups and how to run statistical tests on the estimates by using *postoption*, described in section 3.2.

```
. mpi d1(exp_i) w1(0.5) d2(educ_i) w2(0.3) d3(BMI_i) w3(0.2) [pw=weights],
> cutoff(0.66) by(sex) postbymain
```

Summary of mpi indicators

Indicator	Type	Weight	Deprived
Domain 1			
exp_i	Binary	.5	31.477 %
Domain 2			
educ_i	Binary	.3	38.318 %
Domain 3			
BMI_i	Binary	.2	16.006 %

Deprived: Percentage of individuals whose indicator values are below the threshold.

Main results

N = 17678

		Coef.	Std. Err.	[95% Conf. Interval]	
Main	H	0.191	0.003	0.185	0.198
	M0	0.159	0.003	0.153	0.164
Additional					
	A	0.829	0.002	0.826	0.833

Note: Adjusted Multidimensional Headcount

M0 = H*A

Indicator	M0
domain 1 exp_i	0.603
domain 2 educ_i	0.317
domain 3 BMI_i	0.080
Total	1.000

Contribution of each indicator (%)

Decomposition by subgroups

MPI by: sex

	Male	Female	Total
H	0.160	0.217	0.191
M0	0.131	0.181	0.159
pop share	0.454	0.546	1.000

Indices by subgroup (absolute)

	Male	Female	Total
H	0.380	0.620	1.000
M0	0.376	0.624	1.000

Contribution of subgroups to indices (%)

	Male	Female	Total
M0			
exp_i	0.610	0.599	0.603
educ_i	0.298	0.328	0.317
BMI_i	0.093	0.073	0.080
Total	1.000	1.000	1.000

Contribution of each indicator (%)

Type `ereturn` list to see the list of saved results and more information on the estimation sample.

The indicator `exp_i` enters the analysis with a relevance of 50%, whereas `educ` and `BMI` have weights of 30% and 20%, respectively. The different weighting structure affects both the identification of the poor and the measurement of poverty: H , the share of poor in the population, is now lower, whereas A is slightly higher, implying an overall lower value of M_0 .

When the user specifies the `by(varname)` option, `mpi` computes the related decomposition by categories of `varname`. In this example, `mpi` provides the decomposition by gender.

The three tables at the end of the output relate to the decomposition by population subgroups. The first shows the absolute value of the indices in each subgroup and, in the last row, shows the related population shares. In this example, 54.6% of the population are women. In this subgroup, the incidence of poverty is significantly higher (21.7% against 16.0%), as well as the overall level of M_0 (0.181 against 0.131). The last column shows the overall value of the indices in the population, which is given by the weighted sum of the indices in the two subgroups with weights given by the related population shares: $0.131 \times 0.454 + 0.181 \times 0.546 = 0.159$.

The second table shows the proportional contributions of each subgroup of the population to the overall index. In this example, 62.4% of M_0 is attributable to the group of women. These values are computed by dividing the weighted indices of each subgroup by the overall index, with weights given by the related population share.

The third and last table shows the proportional contributions of the indicators in each subgroup. In this example, there are clear differences by gender. For women, education contributes more to poverty, while for men, expenditure and BMI are relatively more important. However, for both genders overall, expenditure and education are much stronger drivers of poverty than BMI.

`mpi` allows statistical tests on the estimates with standard postestimation commands, such as `test`. For instance, one may want to test whether M_0 is statistically different for men and women. To run postestimation tests, the user must specify one of the `postoption` options outlined in section 3.2. If the aim is to test differences of the main indices (that is, M_0 and H in the example above) between population subgroups (for example, by gender) the proper `postoption` would be `postbymain`. In this case, `mpi` stores in `e(b)` the results from the table “Indices by subgroup (absolute)” and stores in `e(V)` the related variance–covariance matrix. This allows the user to easily run standard postestimation tests, for example,

```
. matrix list e(b)
e(b) [1,4]
      H:      M0:      H:      M0:
    sex_1  sex_1  sex_2  sex_2
y1  .15998123 .21709587 .13123019 .18127231
. test [H]_b[sex_1] = [H]_b[sex_2]
( 1)  [H]sex_1 - [H]sex_2 = 0
      chi2( 1) = 857.00
      Prob > chi2 = 0.0000
. test [M0]_b[sex_1] = [M0]_b[sex_2]
( 1)  [M0]sex_1 - [M0]sex_2 = 0
      chi2( 1) = 1526.92
      Prob > chi2 = 0.0000
```

Depending on the `postoption` option included in the command line, one can make inference on different types of `mpi` estimates. For instance, the `postbyindicator` option allows running statistical tests on the contribution of each indicator by population subgroup.

The next example shows an application of `mpi` with real-valued indicators and with more than one indicator in the first deprivation domain. Specifically, the indicator of household expenditure is replaced with two indicators: household food expenditures and other (nonfood) expenditures. The overall relevance of this domain is kept at 50%, and equal weights are assigned to the two indicators, which implies that each indicator of this domain enters with a weight of 25% in the `mpi` analysis. The relevance of the other two domains is also the same as before: 30% for years of education and 20% for BMI.

When using real-valued indicators, users must specify the related deprivation thresholds in the command syntax. In what follows, the chosen thresholds for the first domain are 100,000 Rupiah for food and 12,000 Rupiah for nonfood expenditure. The thresholds for the other domains are the same as before: less than 6 years of schooling and a BMI lower than 18.5 kg/m². With real-valued indicators, `mpi` computes all the AF poverty measures: M_0 , M_1 , and M_2 . The user may specify additional values of α by using the related option (`alpha(3)` in the example below).

```
. mpi d1(ex_food ex_nofood)   t1(100000 12000)   w1(0.4 0.1)
>   d2(educ)                  t2(6)              w2(0.3)
>   d3(BMI)                   t3(18.5)           w3(0.2)
>   [pw=weights], cutoff(0.66) alpha(3)
```

Summary of mpi indicators

Indicator	Type	Weight	Threshold	Deprived
Domain 1				
ex_food	Real-valued	.4	100000	19.941 %
ex_nofood	Real-valued	.1	12000	25.709 %
Domain 2				
educ	Real-valued	.3	6	38.318 %
Domain 3				
BMI	Real-valued	.2	18.5	16.006 %

Deprived: Percentage of individuals whose indicator values are below the threshold.

Main results

N = 17678

		Coef.	Std. Err.	[95% Conf. Interval]	
Main					
	H	0.117	0.003	0.111	0.122
	M0	0.094	0.002	0.089	0.098
	M1	0.043	0.001	0.041	0.045
	M(2)	0.031	0.001	0.029	0.033
	M(3)	0.026	0.001	0.025	0.028
Additional					
	A	0.804	0.003	0.799	0.809
	G	0.460	0.004	0.451	0.468
	S(2)	0.330	0.005	0.321	0.339
	S(3)	0.279	0.005	0.270	0.288

Note: Adjusted Multidimensional Headcount M0 = H*A
Adjusted Poverty Gap M1 = H*A*G
Adjusted Foster-Greer-Thorbecke (FGT) Measure M(a) = H*A*S(a)

Indicator	M0	M(1)	M(2)	M(3)
domain 1				
ex_food	0.497	0.318	0.194	0.122
ex_nofood	0.080	0.091	0.083	0.072
domain 2				
educ	0.346	0.579	0.721	0.806
domain 3				
BMI	0.076	0.012	0.002	0.000
Total	1.000	1.000	1.000	1.000

Contribution of each indicator (%)

Domain	M0	M(1)	M(2)	M(3)
domain 1	0.578	0.409	0.277	0.194
domain 2	0.346	0.579	0.721	0.806
domain 3	0.076	0.012	0.002	0.000
Total	1.000	1.000	1.000	1.000

Contribution of each domain (%)

Type `ereturn list` to see the list of saved results and more information on the estimation sample.

The summary table at the top now shows that the indicators are real-valued together with the related thresholds and the direction of each deprivation, which in this example is always for values below the threshold.

The second table shows the values of the AF class of poverty measures with the related subindices. Because the command line includes the `alpha(3)` option, the table also shows the value of M_3 and the related subindex S_3 as defined in section 2.2.

The third table shows the contribution of the indicators to each poverty measure, M_0 , M_1 , M_2 , and M_3 , whereas the last table provides the contribution of the deprivation domains.

5 Conclusions

In this article, we outlined `mpi`, a new command for estimating the entire class of Alkire and Foster (2011) multidimensional poverty measures. A key feature of `mpi` is its flexibility. Depending on the nature of the deprivation indicators, `mpi` allows for arbitrary values of α in the calculation of the poverty indices. Results are organized and displayed in formatted tables, and all poverty measures are conveniently decomposed by deprivation indicators and population subgroups. Users can also specify an indefinite number of indicators and use a flexible weighting structure for each of them. Indicator thresholds can be set directly within the command line by using a flexible notation for identifying who is below the threshold. Deprivation indicators can be grouped into broader policy

domains, thus simplifying the empirical analysis. The survey design can be fully accounted for when computing standard errors and covariances. `mpi` comes as a standard e-type command, which means that bootstrapping and postestimation programs, such as `test`, can be easily used to make inference on the estimated parameters.

6 Acknowledgment

The views expressed in this paper should not be reported as representing the official views of the OECD or of its member countries. Any errors made, opinions expressed, or arguments used are those of the authors.

7 References

- Abdelkrim, A., and J.-Y. Duclos. 2007. DASP: Distributive analysis Stata package. <http://dasp.ecn.ulaval.ca/>.
- Alkire, S. 2013. How to measure the many dimensions of poverty? In *Development Co-operation Report 2013: Ending Poverty*, 43–50. Paris: OECD Publishing.
- Alkire, S., and J. Foster. 2008. Counting and multidimensional poverty measurement. OPHI Working Paper 7. <http://www.ophi.org.uk/wp-content/uploads/ophi-wp7.pdf>.
- . 2011. Counting and multidimensional poverty measurement. *Journal of Public Economics* 95: 476–487.
- Foster, J., J. Greer, and E. Thorbecke. 1984. A class of decomposable poverty measures. *Econometrica* 52: 761–766.
- OECD. 2015. Keeping the multiple dimensions of poverty at the heart of development. <http://www.oecd.org/dac/POST-2015%20multidimensional%20poverty.pdf>.
- Strauss, J., K. Beegle, B. Sikoki, A. Dwiyanto, Y. Herawati, and F. Witoelar. 2004. The third wave of the Indonesia Family Life Survey (IFLS): Overview and field report.

About the authors

Daniele Pacifico is an economist at the OECD—Directorate for Employment, Labour and Social Affairs.

Felix Poege is a PhD student at the Max Planck Institute for Innovation and Competition in Munich, Germany.