The Stata Journal (2016)
16, Number 1, pp. 197–228

Implementing factor models for unobserved heterogeneity in Stata

Miguel Sarzosa
Purdue University
West Lafayette, IN
msarzosa@purdue.edu

Sergio Urzúa
University of Maryland
College Park, MD
and National Bureau of Economic Research
Cambridge, MA
urzua@econ.umd.edu

Abstract. We introduce a new command, `heterofactor`, for the maximum likelihood estimation of models with unobserved heterogeneity, including a Roy model. `heterofactor` fits models with up to four latent factors and allows the unobserved heterogeneity to follow general distributions. Our command differs from Stata's `sem` command in that it does not rely on the linearity of the structural equations and distributional assumptions for identification of the unobserved heterogeneity. It uses the estimated distributions to numerically integrate over the unobserved factors in the outcome equations by using a mixture of normals in a Gauss–Hermite quadrature. `heterofactor` delivers consistent estimates, including the unobserved factor loadings, in a variety of model structures.

Keywords: st0431, heterofactor, unobserved heterogeneity, factor models, Roy model, maximum likelihood, numerical integration

1 Introduction

Unobserved heterogeneity has become a particularly relevant topic in modern applied microeconomics (Keane and Wolpin 1997; Cameron and Heckman 1998, 2001; Carneiro, Hansen, and Heckman 2003; Heckman, Stixrud, and Urzua 2006; Urzúa 2008; Sarzosa and Urzua 2015). However, its adequate analysis requires structural models often tailored to the needs of each particular research project. This reflects the fact that the research community lacks the tools for the systematic inclusion of unobserved heterogeneity in practical analyses. However, advances in computational capability have facilitated the estimation of structural models so that it is now conceivable to run some of these models in standard computers.

In this article, we discuss the implementation of factor models when estimating structural equations in the presence of unobserved heterogeneity and a new command, `heterofactor`, for fitting such models. Our routines allow the calculation of consistent estimates, including the loadings of the unobserved factors, in multiple structures. The structural models that we address are related to the model used in Carneiro, Hansen, and Heckman (2003) and Heckman, Stixrud, and Urzua (2006) and first introduced by Jöreskog and Goldberger (1972) and Cameron and Heckman (1998, 2001). The most salient feature of these models is their factor structure that provides a parsimonious

specification to identify unobserved heterogeneity and its effects on the outcomes of interest. Unlike Stata's `sem` command, our command does not rely on the linearity of the structural equations and distributional assumptions for identification. Instead, the distributions of the unobserved factors are identified nonparametrically following the contributions of Kotlarski (1967).

As shown below, the structural models we refer to here have a variety of applications. Recently, the treatment-effect literature has embraced these models not only because they provide a method for estimating treatment effects depending on the level of unobservables but also because controlling for unobserved heterogeneity allows the simulation of counterfactuals. This method can also be used to estimate the parameters of a measurement system that contains unobserved attributes. In particular, this setting could relate to the skills literature, where cognitive and noncognitive skills are unobservable characteristics of individuals that influence their decisions and outcomes later in life (see Heckman et al. [2011]; Prada and Urzua [2013]; Sarzosa and Urzúa [2015]).

This article is organized as follows. In section 2, we review the factor model structure and discuss the mechanisms that allow us to identify key parameters. In section 3, we discuss the implementation of our estimation routines, including the syntax of `heterofactor`. In section 4, we provide examples using both simulated data and data from the National Longitudinal Survey of Youth (NLSY79). In section 5, we conclude.

2 Factor model estimation

The type of structural model that our command can handle can be described as a set of measurement systems that are linked by a factor structure. This is the type of model considered by Hansen, Heckman, and Mullen (2004), Heckman, Stixrud, and Urzua (2006), Heckman and Navarro (2007), and Sarzosa and Urzúa (2015). In a general setup, suppose we face the following linear system,

$$\mathbf{Y} = \mathbf{X}_Y \beta^Y + \mathbf{U}^{\mathbf{Y}}$$

where $\mathbf{Y}$ is an $M \times 1$ vector of outcome variables, $\mathbf{X}_Y$ is a matrix with all observable controls, and $\mathbf{U}^{\mathbf{Y}}$ is a vector that contains the unobservables for each outcome equation with a factor structure of the form $\mathbf{U}^{\mathbf{Y}} = \mathbf{\Lambda}^{\mathbf{Y}}\mathbf{\Theta} + \mathbf{e}^{\mathbf{Y}}$. Hence, we can expand the linear system to

$$\mathbf{Y} = \mathbf{X}_Y \beta^Y + \mathbf{\Lambda}^{\mathbf{Y}}\mathbf{\Theta} + \mathbf{e}^{\mathbf{Y}} \tag{1}$$

where $\mathbf{\Theta}$ is a $q \times 1$ vector that contains the q dimensions of unobserved heterogeneity (that is, q latent factors), $\mathbf{\Lambda}^{\mathbf{Y}}$ is an $M \times q$ matrix that contains the factor loadings for each type of unobserved heterogeneity, and $\mathbf{e}^{\mathbf{Y}}$ is a vector of error terms with distributions $f_{e^{y_m}}(\cdot)$ for every $m = 1, \ldots, M$. We assume that $\mathbf{e}^{\mathbf{Y}} \perp (\mathbf{\Theta}, \mathbf{X}_Y)$ and also that $e^{y_i} \perp e^{y_j}$ for $i, j = 1, \ldots, M$ and $i \neq j$. Furthermore, we assume the vector $\mathbf{\Theta}$ has the associated distribution $f_\theta(\cdot)$. Hence, the econometrician does not observe the actual value of $\mathbf{\Theta}$ for each observation. Instead, he or she knows or estimates the distributions from which they are drawn.

The measurement system (2) can be used to identify the components of matrix $\mathbf{\Lambda^Y}$, albeit under very stringent constraints and assumptions (Aakvik, Heckman, and Vytlacil 2000). As indicated by Carneiro, Hansen, and Heckman (2003), the estimations that come from the factor structure will gain interpretability and will require fewer restrictions for its identification if a measurement system—also linked by the same factor structure—is adjoined to the system (1). This system can be used to identify the distributional parameters of the unobserved factors. This adjoined measurement system would have the form

$$\mathbf{T} = \mathbf{X}_T \beta^T + \mathbf{\Lambda^T \Theta} + \mathbf{e^T} \tag{2}$$

where $\mathbf{T}$ is an $L \times 1$ vector of measurements (for example, test scores), $\mathbf{X}_T$ is a matrix with all observable controls for each measurement, and $\mathbf{\Lambda^T}$ is an $L \times q$ matrix that holds the loadings of the q unobserved factors. Again we assume that $(\mathbf{\Theta}, \mathbf{X}_T) \perp \mathbf{e^T}$ and that all the elements of the $L \times 1$ vector $\mathbf{e^T}$ are mutually independent and have associated distributions $f_{e^h}(\cdot)$ for every $h = 1, \ldots, L$.[1]

2.1 Identification of the adjunct measurement system

`heterofactor` can handle up to four factors. However, for presentation purposes, we will demonstrate estimation using a two-factor model.[2] In the two-factor model, (1) becomes

$$\mathbf{Y} = \mathbf{X}_Y \beta^Y + \boldsymbol{\alpha}^{Y,A} \theta^A + \boldsymbol{\alpha}^{Y,B} \theta^B + \mathbf{e^Y} \tag{3}$$

and (2) becomes

$$\mathbf{T} = \mathbf{X}_T \beta^T + \boldsymbol{\alpha}^{\mathbf{T,A}} \theta^A + \boldsymbol{\alpha}^{\mathbf{T,B}} \theta^B + \mathbf{e^T} \tag{4}$$

To explain how the parameters of the adjunct measurement system (4) are identified, let's focus on the matrix $\text{Cov}(\mathbf{T} | \mathbf{X}_T)$, whose elements in the diagonal are of the form

$$\text{Cov}(T_i, T_i | \mathbf{X}_T) = \left(\alpha^{T_i,A}\right)^2 \sigma^2_{\theta^A} + \alpha^{T_i,A} \alpha^{T_i,B} \sigma_{\theta^A \theta^B} + \left(\alpha^{T_i,B}\right)^2 \sigma^2_{\theta^B} + \sigma^2_{e^{T_i}} \tag{5}$$

and in the off diagonal are of the form

$$\begin{aligned} \text{Cov}(T_i, T_j | \mathbf{X}_T) = {} & \alpha^{T_i,A} \alpha^{T_j,A} \sigma^2_{\theta^A} + \left(\alpha^{T_i,A} \alpha^{T_j,B} + \alpha^{T_i,B} \alpha^{T_j,A}\right) \sigma_{\theta^A \theta^B} \\ & + \alpha^{T_i,B} \alpha^{T_j,B} \sigma^2_{\theta^B} \end{aligned} \tag{6}$$

As it is, the model is underidentified (Carneiro, Hansen, and Heckman 2003). Therefore, identification requires some assumptions. First, we need $\theta^A \perp \theta^B$, so $\sigma_{\theta^A \theta^B} = 0$ in

1. For the maximum likelihood procedure we describe below, we assume $f_{e^h}(\cdot)$ are normal distributions. This is a relatively mild assumption because these come from the idiosyncratic variation that remains after controlling for observed controls and unobserved heterogeneity.
2. The extension to three and four factors is straightforward.

(5) and (6).[3] The second assumption relates to the minimum number of measurements we need to have per factor. Notice that the diagonal elements of $\text{Cov}\left(\mathbf{T}\,|\mathbf{X}_T\right)$ have the variances of the idiosyncratic errors, while the off-diagonal elements do not. Hence, once we identify the rest of the model parameters, the diagonals will identify $\sigma^2_{e^{T_h}}$ for $h = 1, \ldots, L$. Then, following Carneiro, Hansen, and Heckman (2003), we can use the $\{L(L-1)\}/2$ off-diagonal elements to identify the variances of the factors and their associated factor loadings. If we let k be the number of factors we are using in the model—in the present example, $k = 2$—then we have $k \times L$ loadings. It should then follow that

$$\frac{L\left(L-1\right)}{2} \geq Lk + k \quad \text{thus} \quad \frac{L\left(L-1\right)}{2\left(L+1\right)} \geq k$$

In our example where $k = 2$, this restriction tells us that $L \geq 6$. That is, we need at least six test scores to identify the parameters of the measurement system with two factors.

The next step for identification is to acknowledge that latent factors have no metric or scale of their own. Hence, we need to normalize to unity one loading per factor, and the estimation of all other loadings should be interpreted as relative to those used as numeraire.[4] To incorporate this into our notation, we expand (4) into k blocks of size m_κ such that $\sum_\kappa m_\kappa = L$. That way, without loss of generality, we set the first loading in the first equation in each block to one. In our example, we get two blocks, a and b. That is, we write (4) as

$$\mathbf{T}^a = \mathbf{X}_{T^a}\beta^{T^a} + \boldsymbol{\alpha}^{\mathbf{T^a},\mathbf{A}}\theta^A + \boldsymbol{\alpha}^{\mathbf{T^a},\mathbf{B}}\theta^B + \mathbf{e}^{\mathbf{T^a}}$$
$$\mathbf{T}^b = \mathbf{X}_{T^b}\beta^{T^b} + \boldsymbol{\alpha}^{\mathbf{T^b},\mathbf{A}}\theta^A + \boldsymbol{\alpha}^{\mathbf{T^b},\mathbf{B}}\theta^B + \mathbf{e}^{\mathbf{T^b}}$$

3. Using higher moments of the distributions, Heckman and Navarro (2007) show that identification can be achieved even if the factor independence assumption is relaxed. Also, Sarzosa (2015) shows that models with correlated factors can be identified if additional restrictions are imposed on the factor loadings structure.
4. These normalizations reduce by k the number of parameters to estimate. Hence, L, the number of measurements needed, is given by $\{L\left(L-1\right)\}/2 \geq Lk + k - k$, which simplifies to $L \geq 2k + 1$. Therefore, the presence of two factors in (3) implies that there should be at least five measures in (4). Throughout the routines in this article, we will assume that we have at least $3k$ measurements.

with $\alpha^{T_1^a,A} = 1$ and $\alpha^{T_1^b,B} = 1$, where T_1^κ indicates the first test in block κ and T_i^κ indicates all tests different from the first one in block κ. Then the off-diagonal elements of the $\text{Cov}\left(\mathbf{T}\,|\mathbf{X}_T\right)$ matrix follow one of the following cases,

$$\text{Cov}\left(T_1^a, T_i^b\,|\mathbf{X}_T\right) = \alpha^{T_i^b,A}\sigma^2_{\theta^A} + \alpha^{T_1^a,B}\alpha^{T_i^b,B}\sigma^2_{\theta^B} \tag{7}$$

$$\text{Cov}\left(T_i^a, T_i^b\,|\mathbf{X}_T\right) = \alpha^{T_i^a,A}\alpha^{T_i^b,A}\sigma^2_{\theta^A} + \alpha^{T_i^a,B}\alpha^{T_i^b,B}\sigma^2_{\theta^B}$$

$$\text{Cov}\left(T_1^a, T_1^b\,|\mathbf{X}_T\right) = \alpha^{T_1^b,A}\sigma^2_{\theta^A} + \alpha^{T_1^a,B}\sigma^2_{\theta^B} \tag{8}$$

$$\text{Cov}\left(T_i^a, T_1^b\,|\mathbf{X}_T\right) = \alpha^{T_i^a,A}\alpha^{T_1^b,A}\sigma^2_{\theta^A} + \alpha^{T_i^a,B}\sigma^2_{\theta^B} \tag{9}$$

$$\text{Cov}\left(T_1^\kappa, T_i^\kappa\,|\mathbf{X}_T\right) = \alpha^{T_i^\kappa,A}\sigma^2_{\theta^A} + \alpha^{T_1^\kappa,B}\alpha^{T_i^\kappa,B}\sigma^2_{\theta^B} \tag{10}$$

$$\text{Cov}\left(T_i^\kappa, T_j^\kappa\,|\mathbf{X}_T\right) = \alpha^{T_i^\kappa,A}\alpha^{T_j^\kappa,A}\sigma^2_{\theta^A} + \alpha^{T_i^\kappa,B}\alpha^{T_j^\kappa,B}\sigma^2_{\theta^B} \tag{11}$$

for $\kappa = \{a, b\}$ and $i \neq j$. These elements show that we cannot identify $\sigma^2_{\theta^A}$ and $\sigma^2_{\theta^B}$ and the loadings without further restrictions. Carneiro, Hansen, and Heckman (2003) suggest that the first restrictions should be $\alpha^{T_1^a,B} = 0$, $\alpha^{T_2^a,B} = 0$, and $\alpha^{T_3^a,B} = 0$. That is, the first three tests in the first block can be affected by only the first factor. Then

$$\begin{aligned}
\text{Cov}\left(T_1^a, T_2^a\,|\mathbf{X}_T\right) &= \alpha^{T_2^a,A}\sigma^2_{\theta^A} \\
\text{Cov}\left(T_1^a, T_3^a\,|\mathbf{X}_T\right) &= \alpha^{T_3^a,A}\sigma^2_{\theta^A} \\
\text{Cov}\left(T_2^a, T_3^a\,|\mathbf{X}_T\right) &= \alpha^{T_2^a,A}\alpha^{T_3^a,A}\sigma^2_{\theta^A}
\end{aligned}$$

Then

$$\frac{\text{Cov}\left(T_2^a, T_3^a\,|\mathbf{X}_T\right)}{\text{Cov}\left(T_1^a, T_2^a\,|\mathbf{X}_T\right)} = \alpha^{T_3^a,A}, \quad \frac{\text{Cov}\left(T_2^a, T_3^a\,|\mathbf{X}_T\right)}{\text{Cov}\left(T_1^a, T_3^a\,|\mathbf{X}_T\right)} = \alpha^{T_2^a,A}, \quad \frac{\text{Cov}\left(T_2^a, T_i^\kappa\,|\mathbf{X}_T\right)}{\text{Cov}\left(T_1^a, T_2^a\,|\mathbf{X}_T\right)} = \alpha^{T_i^\kappa,A}$$

and hence, we identify $\sigma^2_{\theta^A}$ from

$$\text{Cov}\left(T_1^a, T_3^a\,|\mathbf{X}_T\right) = \frac{\text{Cov}\left(T_2^a, T_3^a\,|\mathbf{X}_T\right)}{\text{Cov}\left(T_1^a, T_2^a\,|\mathbf{X}_T\right)}\sigma^2_{\theta^A}$$

Identification of the loadings and variances associated with the subsequent factors requires fewer restrictions. Note that under the assumption of $\alpha^{T_1^a,B} = 0$, (7) and (8) become

$$\text{Cov}\left(T_1^a, T_i^b\,|\mathbf{X}_T\right) = \alpha^{T_i^b,A}\sigma^2_{\theta^A} \quad \text{and} \quad \text{Cov}\left(T_1^a, T_1^b\,|\mathbf{X}_T\right) = \alpha^{T_1^b,A}\sigma^2_{\theta^A}$$

respectively. Given that we already know $\sigma^2_{\theta^A}$, we can identify all the loadings associated with the first factor in all the subsequent blocks. This allows us to use (9), (10), and (11) when $\kappa = b$ to identify $\sigma^2_{\theta^B}$ and $\alpha^{T^b,B}$ because we already know the first part of the right-hand side of those expressions.

Finally, having identified all the parameters from the off-diagonal elements of the $\text{Cov}\left(\mathbf{T}\,|\mathbf{X}_T\right)$ matrix, we can identify the parameters in the diagonal. From (5) and the restrictions we have imposed, we find that the typical diagonal element of $\text{Cov}\left(\mathbf{T}\,|\mathbf{X}_T\right)$ is

$$\text{Cov}\left(T_i, T_i\,|\mathbf{X}_T\right) = \left(\alpha^{T_i,K}\right)^2 \sigma^2_{\theta^K} + \sigma^2_{e^{T_i}}$$

for $K = \{A, B\}$. Given that we have already identified the first part of the right-hand side of this equation, we can use the diagonal elements to identify $\sigma^2_{e^{T_i}}$.

Now that we have identified all the loadings, factor variances, and measurement residual variances and that we know that the means of θ^A, θ^B, and $\mathbf{e^T}$ are finite—in fact, they are equal to zero because we allow the measurement system (4) to have intercepts—we can invoke the Kotlarski Theorem (Kotlarski 1967)[5] to use the manifest variables $\mathbf{T}$ to nonparametrically identify the distributions of $f_{\theta^A}(\cdot)$ and $f_{\theta^B}(\cdot)$.

2.2 Loadings structures in the measurement system

We have shown that identification requires some restrictions in the loadings structure. The more general structure requires one normalization per factor and the first three measurements of the first block to be affected only by the first factor. In our example with two factors and using three measurements per block, the loadings structure can be represented as

$$\mathbf{\Lambda^T} = \begin{bmatrix} \alpha^{T_1,A} & \alpha^{T_1,B} \\ \alpha^{T_2,A} & \alpha^{T_2,B} \\ \alpha^{T_3,A} & \alpha^{T_3,B} \\ \alpha^{T_4,A} & \alpha^{T_4,B} \\ \alpha^{T_5,A} & \alpha^{T_5,B} \\ \alpha^{T_6,A} & \alpha^{T_6,B} \end{bmatrix} = \begin{bmatrix} \alpha^{T_1,A} & 0 \\ \alpha^{T_2,A} & 0 \\ 1 & 0 \\ \alpha^{T_4,A} & \alpha^{T_4,B} \\ \alpha^{T_5,A} & \alpha^{T_5,B} \\ \alpha^{T_6,A} & 1 \end{bmatrix} \tag{12}$$

Provided that the loadings structure fulfills the required restrictions, the choice of structure depends entirely on the available data. The triangular structure (12) allows for a block of measures that depend on both factors. For instance, grades and education achievement scores depend not only on a cognitive factor but also on a noncognitive one.

5. The Kotlarski Theorem states that if there are three independent random variables, e_{T_1}, e_{T_2}, and θ, and we define $T_1 = \theta + e_{T_1}$ and $T_2 = \theta + e_{T_2}$, the joint distribution of (T_1, T_2) determines the distributions of e_{T_1}, e_{T_2}, and θ, up to one normalization. Given that we have already identified all the loadings, we can write (4) in terms of $T_\tau = \theta + e_{T_\tau}$ by dividing both sides by the loading. See more details in Carneiro, Hansen, and Heckman (2003).

If data permit, the researcher can use a more restrictive loadings structure in which only one factor affects each block of measurements. It will take the following form:

$$\mathbf{\Lambda^T} = \begin{bmatrix} \alpha^{T_1,A} & \alpha^{T_1,B} \\ \alpha^{T_2,A} & \alpha^{T_2,B} \\ \alpha^{T_3,A} & \alpha^{T_3,B} \\ \alpha^{T_4,A} & \alpha^{T_4,B} \\ \alpha^{T_5,A} & \alpha^{T_5,B} \\ \alpha^{T_6,A} & \alpha^{T_6,B} \end{bmatrix} = \begin{bmatrix} \alpha^{T_1,A} & 0 \\ \alpha^{T_2,A} & 0 \\ 1 & 0 \\ 0 & \alpha^{T_4,B} \\ 0 & \alpha^{T_5,B} \\ 0 & 1 \end{bmatrix} \tag{13}$$

This type of loadings structure will increase the speed of estimation because it requires the estimation of fewer parameters.

2.3 Estimation

We fit the model (4) using maximum likelihood estimation. The likelihood is

$$\mathcal{L} = \prod_{i=1}^{N} \int \int \left\{ f_{e^1}\left(\mathbf{X}_{T_1}, T_1, \zeta^A, \zeta^B\right) \times \cdots \times f_{e^L}\left(\mathbf{X}_{T_L}, T_L, \zeta^A, \zeta^B\right) \right\} dF_{\theta^A}\left(\zeta^A\right) dF_{\theta^B}\left(\zeta^B\right)$$

where we integrate over the distributions of the factors because of their unobservable nature, obtaining $\widehat{\beta}^T$, $\alpha^{T,A}$, $\alpha^{T,B}$, $\widehat{F}_{\theta^A}(\cdot)$, and $\widehat{F}_{\theta^B}(\cdot)$. All the integrals are calculated numerically using a Gauss–Hermite quadrature within a mixture of normals (Judd 1998). This guarantees the flexibility required to appropriately re-create the unobserved distributions in the estimation. Our routine does not impose normality on $F_{\theta^A}(\cdot)$ and $F_{\theta^B}(\cdot)$. Instead, it assumes they are distributed according to mixtures of two normal distributions. Therefore, we estimate the distributional parameters of the normals and the mixing probability. This way, we can identify an array of possible functional forms for $F_{\theta^A}(\cdot)$ and $F_{\theta^B}(\cdot)$.

Having identified the distributional parameters of $F_{\theta^A}(\cdot)$ and $F_{\theta^B}(\cdot)$ from (4), we can proceed to fit model (3). The likelihood function here is

$$\mathcal{L} = \prod_{i=1}^{N} \int \int \left\{ f_{e^{y_1}}\left(\mathbf{X}_{Y_1}, Y_1, \zeta^A, \zeta^B\right) \times \cdots \times f_{e^{y_M}}\left(\mathbf{X}_{Y_M}, Y_M, \zeta^A, \zeta^B\right) \right\} dF_{\theta^A}\left(\zeta^A\right) dF_{\theta^B}\left(\zeta^B\right)$$

This maximum likelihood estimation will yield $\widehat{\beta}^Y$, $\alpha^{Y,A}$, and $\alpha^{Y,B}$.[6]

6. In this two-step procedure, we use a limited-information maximum likelihood and correct the variance–covariance matrix of the second stage, incorporating the estimated variance–covariance matrix and gradient of the first stage (Greene 2012).

Also, the two steps presented above can be joined and calculated in one likelihood of the form

$$\mathcal{L} = \prod_{i=1}^{N} \int \int \left\{ \begin{array}{c} f_{e^{y_1}}\left(\mathbf{X}_{Y_1}, Y_1, \zeta^A, \zeta^B\right) \times \cdots \times f_{e^{y_M}}\left(\mathbf{X}_{Y_M}, Y_M, \zeta^A, \zeta^B\right) \\ \times f_{e^1}\left(\mathbf{X}_{T_1}, T_1, \zeta^A, \zeta^B\right) \times \cdots \times f_{e^L}\left(\mathbf{X}_{T_L}, T_L, \zeta^A, \zeta^B\right) \end{array} \right\} dF_{\theta^A}\left(\zeta^A\right) dF_{\theta^B}\left(\zeta^B\right)$$

However, the two-step procedure is less computationally burdensome, especially if we are fitting a model with two or more factors.[7]

2.4 The treatment-effect setting: A Roy model

In this subsection, we discuss the special case of model (3), where there is a binary treatment (for example, to go to college) and a later outcome (for example, wages earned at age 30). This is one of the settings where the factor structure has received more attention (Heckman, Stixrud, and Urzua 2006; Urzúa 2008; Heckman et al. 2011; Prada and Urzua 2013). The advantage that the factor structure has here is that potential outcomes are separable in observables and unobservables. That is, conditional on $\boldsymbol{\theta}$ and $\mathbf{X}_Y$, potential outcomes are independent because any selection on unobservables is already accounted for.[8] This allows researchers to simulate observationally identical counterfactuals, permitting the calculation of treatment parameters like average treatment effect, average treatment effect for the treated, and average treatment effect on the untreated, for every level of the unobserved heterogeneity.

Consider a model of potential outcomes inspired by the Roy model (Roy 1951). Individuals must choose between two sectors such as treated and not treated or high school and college. The choice is based on the decision model

$$D = \mathbb{1}\left(\mathbf{X}_D \beta^{Y_D} + \alpha^{Y_D, A}\theta^A + \alpha^{Y_D, B}\theta^B + e^D > 0\right)$$

where $\mathbb{1}(A)$ denotes an indicator function that takes a value of 1 if A is true. Then D is the binary treatment variable, and $\mathbf{X}_D$ represents a set of exogenous observable variables. Depending on the selected sector (that is, $D = 1$ or $D = 0$), individuals will experience different outcomes. We denote these potential outcomes by Y_1 and Y_0, respectively. Y_1 can represent, for instance, the wages earned at age 30 by a college graduate, while Y_0 represents the wages earned at age 30 by a person that did not go to college. Therefore, in a treatment-effect setting, the system of equations (3) will represent both potential outcomes and the choice equation. That is, $\mathbf{Y} = (Y_1, Y_0, D)'$. Here the system would be

7. For one-factor models, (3) and (4) become $\mathbf{Y} = \mathbf{X}_Y \beta^Y + \boldsymbol{\alpha}^Y \theta + \mathbf{e}^{\mathbf{Y}}$ and $\mathbf{T} = \mathbf{X}_T \beta^T + \boldsymbol{\alpha}^{\mathbf{T}} \theta + \mathbf{e}^{\mathbf{T}}$, respectively. The likelihood function would be

$$\mathcal{L} = \prod_{i=1}^{N} \int \int \left\{ \begin{array}{c} f_{e^{y_1}}\left(\mathbf{X}_{Y_1}, Y_1, \zeta\right) \times \cdots \times f_{e^{y_M}}\left(\mathbf{X}_{Y_M}, Y_M, \zeta\right) \\ \times f_{e^1}\left(\mathbf{X}_{T_1}, T_1, \zeta\right) \times \cdots \times f_{e^L}\left(\mathbf{X}_{T_L}, T_L, \zeta\right) \end{array} \right\} dF_\theta(\zeta)$$

8. Recall that $e^{y_i} \perp e^{y_j}$ for $i, j = 1, \ldots, M$ and $i \neq j$.

$$Y_1 = \begin{cases} \mathbf{X}_Y \beta^{Y_1} + \alpha^{Y_1,A}\theta^A + \alpha^{Y_1,B}\theta^B + e^{Y_1} & \text{if } D = 1 \\ 0 & \text{if } D = 0 \end{cases} \tag{14}$$

$$Y_0 = \begin{cases} \mathbf{X}_Y \beta^{Y_0} + \alpha^{Y_0,A}\theta^A + \alpha^{Y_0,B}\theta^B + e^{Y_0} & \text{if } D = 0 \\ 0 & \text{if } D = 1 \end{cases} \tag{15}$$

$$D = \mathbb{1}\left(\mathbf{X}_D \beta^{Y_D} + \alpha^{Y_D,A}\theta^A + \alpha^{Y_D,B}\theta^B + e^D > 0\right) \tag{16}$$

The second-step likelihood function is given by

$$\mathcal{L} = \prod_{i=1}^{N} \int\int \left\{ \begin{array}{c} f^{Y_0}\left(\mathbf{X}_Y, Y_0, \zeta^A, \zeta^B\right)^{1-D} f^{Y_1}\left(\mathbf{X}_Y, Y_1, \zeta^A, \zeta^B\right)^{D} \\ \times f^{D}\left(\mathbf{X}_D, Y_D, \zeta^A, \zeta^B\right) \end{array} \right\} \\ d\widehat{F}_{\theta^A}\left(\zeta^A\right) d\widehat{F}_{\theta^B}\left(\zeta^B\right)$$

Thus we obtain different parameter values for each potential outcome. That is, the measures of the effects of observable and unobservable features on the outcome differ depending on D.

2.5 Probit and normal regressions with unobserved heterogeneity

An especial case related to the one presented above is one in which the vector of outcomes is composed by only the choice or treatment decision. That is, vector $\mathbf{Y} = D$, meaning there are no potential-outcome equations in the second step. The outcome equation to be estimated is (17). The complete likelihood would be

$$\mathcal{L} = \prod_{i=1}^{N} \int\int \left\{ \begin{array}{l} f^{D}\left(\mathbf{X}_D, Y_D, \zeta^A, \zeta^B\right) \\ \quad \times f_{e^1}\left(\mathbf{X}_{T_1}, T_1, \zeta^A, \zeta^B\right) \times \cdots \times f_{e^L}\left(\mathbf{X}_{T_L}, T_L, \zeta^A, \zeta^B\right) \end{array} \right\} \\ dF_{\theta^A}\left(\zeta^A\right) dF_{\theta^B}\left(\zeta^B\right) \tag{17}$$

This structure should be interpreted as a probit of D on $\mathbf{X}_D$ that allows for unobserved heterogeneity.

Similarly, we may have a case where there is no choice or treatment equation, and there is only one outcome Y. Then the outcome vector $\mathbf{Y} = Y$ and the outcome equation of interest will be

$$Y = \mathbf{X}_Y \beta^Y + \alpha^{Y,A}\theta^A + \alpha^{Y,B}\theta^B + e^Y$$

Here the complete likelihood would be

$$\mathcal{L} = \prod_{i=1}^{N} \int\int \left\{ \begin{array}{l} f_{e^y}\left(\mathbf{X}_Y, Y, \zeta^A, \zeta^B\right) \\ \quad \times f_{e^1}\left(\mathbf{X}_{T_1}, T_1, \zeta^A, \zeta^B\right) \times \cdots \times f_{e^L}\left(\mathbf{X}_{T_L}, T_L, \zeta^A, \zeta^B\right) \end{array} \right\} \\ dF_{\theta^A}\left(\zeta^A\right) dF_{\theta^B}\left(\zeta^B\right) \tag{18}$$

This structure should be interpreted like a normal regression of Y on $\mathbf{X}_Y$ that allows for unobserved heterogeneity.

3 The heterofactor command

3.1 Syntax

The syntax of the command is as follows:

```
heterofactor depvar varlist_X [if] [in], scores(varlist_T)
   indvarsc(varlist_Q) [treatind(varname_D) instrum(varlist_Z) exp1(varlist)
   exp2(varlist) exp3(varlist) exp4(varlist) factors(#) fdistonly
   scndstponly choiceonly nochoice triangular numf1tests(#)
   numf2tests(#) nodes(#) twostep initialreg nohats sigmamixt11(#)
   sigmamixt12(#) sigmamixt21(#) sigmamixt22(#) sigmamixt31(#)
   sigmamixt32(#) sigmamixt41(#) sigmamixt42(#) mumixt1(#)
   mumixt2(#) mumixt3(#) mumixt4(#) mixtprob1(#) mixtprob2(#)
   mixtprob3(#) mixtprob4(#) st2(#) st3(#) st4(#) st5(#) st6(#)
   st9(#) st12(#) resvar2(varname) resvar3(varname) resvar4(varname)
   resvar5(varname) resvar6(varname) resvar9(varname) resvar12(varname)
   firstloads(matname) firstgrad(matname) firstvarmat(matname)
   level(#) maximize_options]
```

`heterofactor` is implemented for Stata 11 by using the `d0` estimator of `ml`. All likelihood routines are coded in Mata. These commands share the same features of most of the Stata estimation commands that use maximum likelihood, including access to the last estimation results and the options for the maximization process (see [R] **maximize**).

3.2 Options

`scores(`*varlist_T*`)` specifies the variables that contain the scores of the measurement system [that is, vector **T** in (4)]. There must be at least three variables specified in *varlist_T* per factor. Users may specify more than three variables per factor for models with one or two factors. If the model has three or four factors, users must specify three variables for the third and fourth factors. The order of *varlist_T* matters. Users must list variables in blocks, where each block should be affected by the same factor or factors. Identification requires one loading normalization per factor. Thus the loadings of the last test score in each block will be normalized. For instance, if the model chosen has four factors and *varlist_T* contains exactly three variables per factor, then the loadings of the third, sixth, ninth, and twelfth variable will be normalized to one. This arrangement is somewhat different if the `triangular` option is specified. In that case, the factor structure is the one presented in (12). That is, if f is the number of factors, the first $f-1$ sets of three measures provided in *varlist_T* should depend on only one factor each, while the last set of three measures will be affected by all factors. `scores()` is required.

`indvarsc(`*varlist_Q*`)` specifies the observed variables that affect all test-score regressions [that is, $\mathbf{X}_T$ in (4)]. *varlist_Q* can be the same as *varlist_X*, but users must specify both. There is no limit for the number of variables that can be specified in *varlist_Q*. If users want to specify different controls for each set of three measures, the `exp1()`, `exp2()`, `exp3()`, and `exp4()` options should be used. `indvarsc()` is required.

`treatind(`*varname_D*`)` specifies the choice variable when there is a choice equation in the model. *varname_D* represents variable $\mathbf{D}$ in (16). *varname_D* indicates the assignment to treatment, and it needs to be a binary variable.

`instrum(`*varlist_Z*`)` specifies the observed variables that affect the binary choice equation [that is, $\mathbf{X}_D$ in (16)].

`exp1(`*varlist*`)`, `exp2(`*varlist*`)`, `exp3(`*varlist*`)`, and `exp4(`*varlist*`)` include more controls in each set of three measures in addition to those specified in *varlist_Q*, which are common to all. Users can add regressors that are believed to affect only one set and not the other ones.

`factors(#)` specifies the number of factors used in the model. # can be any integer between 1 and 4. The default is `factors(1)`.

`fdistonly` specifies to estimate only the first step. Stata will estimate only the factors' distribution parameters and factor loadings on the test scores. No outcome equation will be estimated. However, *depvar* and *varlist_X* should be provided even if they are not going to be used.

`scndstponly` specifies to estimate only the second step. Stata will estimate only the outcome equations. No factor-distribution identification takes place. If `scndstponly` is specified, all the parameters that describe the factors' distributions $F_\theta(\cdot)$, the residuals of *varlist_T*, and their variances and loadings should be provided by users through additional options. This option is useful if users did the first step before and now need only to estimate a new set of outcome equations based on the same factors.

`choiceonly` specifies that the model to be estimated in the second step include only a choice equation. That is, it will estimate only the equation described by *varlist_D* and *varlist_Z*. No other outcomes are estimated, including the potential-outcome equations (14) and (15). The estimation using this option should be interpreted as running a probit estimation allowing for the presence of unobserved heterogeneity.

`nochoice` specifies that the outcome equations in the model not include the binary treatment equation. It indicates to Stata that the model is not of the treatment-effect nature described in subsection 2.4. This likelihood is described in (18) and has a unique outcome equation, $Y = \mathbf{X}_Y\beta^Y + \alpha^{Y,A}\theta^A + \alpha^{Y,B}\theta^B + e^Y$. This should be interpreted as a linear regression allowing for the presence of unobserved heterogeneity.

`triangular` indicates that the measurement system in the first step has a triangular loading structure. If `triangular` is specified, the structure assumed for the measurement system is one that has one block of scores that depends on all factors; the

other blocks of scores depend on only one factor each. This option is valid only for the two-factor case. Note that this option increases the computational time needed for calculation. If `triangular` is not specified, the loading structure assumed is the one presented in (13).

`numf1tests(#)` and `numf2tests(#)` specify the number of tests used in each block of *varlist_T*. These options should be specified only if the number of tests is different from three. For instance, if the user lists seven variables in *varlist_T*, `numf1tests(4)` and `numf2tests(3)` are specified to indicate that the first four variables are in the first block and the last three variables are in the second block.

`nodes(#)` defines the number of points used in the Gauss–Hermite quadrature for integration. The number defined can be either 4 or 10. While using 10 nodes provides more accuracy, integrating with 4 nodes is faster.

`twostep` divides estimation into two parts: the factor-identification part (4) and the outcome-equations part (3). If the `factor(#)` option is specified with $\# > 1$, `twostep` is assumed. If the `factor(1)` option is specified, `twostep` is not used.

`initialreg` specifies to calculate initial values using ordinary least-squares regression of each equation separately. These initial values are different from the ones provided by Stata in the absence of the `initialreg` option.

`nohats` specifies that estimated factor values $\widehat{\theta}$ not be saved in the data. This option speeds the command execution, especially in big datasets.

Sometimes, the user needs to fit several models that use the same factor structure. In that case, the user needs to run the first step only once and can save time by running all the required models using only the second step. To do so, the user needs to specify `scndstponly` and the parameters that describe the distributions, the residuals, the variance–covariance matrix, and the gradient of the first stage. The distributions $F_\theta(\cdot)$ of the factors are obtained using a mixture of two normals. To fully describe each factor's distribution, we need the standard deviation, the mean of each of the normals, and the weight (probability) with which the two normals are combined.

If `scondstponly` is specified, the user needs to provide the parameters that describe the distributions. The parameters should be provided using the following:

`sigmamixt11(#)`, `sigmamixt12(#)`, `sigmamixt21(#)`, `sigmamixt22(#)`, `sigmamixt31(#)`, `sigmamixt32(#)`, `sigmamixt41(#)`, and `sigmamixt42(#)` specify the standard deviations of the two distributions used in the mixture of normals that describe the distribution of the first and specify the standard deviations of the two distributions used in the mixture of the first, second, third, and fourth factors. Given the transformations done in the code to ensure the parameters remain in the valid range, the user needs to provide the natural logarithm of the actual standard deviations. That is, if the standard deviation of the first normal of the first mixture is $\sigma_{11} = 1$, the `sigmamixt11(0)` option should be specified. Note that the values displayed in the output in the first stage are untransformed, and they must be provided with these options.

`mumixt1(#)`, `mumixt2(#)`, `mumixt3(#)`, and `mumixt4(#)` specify the mean of the first part of the mixture of each factor. The factor is centered at zero, so the mean of the second part of the mixture can be obtained from the equation $\varrho\mu_1 + (1-\varrho)\,\mu_2 = 0$, where ϱ is the probability used to combine the mixtures, provided by $\exp(\texttt{mixtprob1(\#)})/\{1+\exp(\texttt{mixtprob1(\#)})\}$ for factor 1, $\exp\{\texttt{mixtprob2(\#)}\}/\{1+\exp(\texttt{mixtprob2(\#)})\}$ for factor 2, and so on.

`mixtprob1(#)`, `mixtprob2(#)`, `mixtprob3(#)`, and `mixtprob4(#)` specify the probability used to combine the two normal distributions into the mixture of normals for the distribution of each factor. As with standard deviations, the value in `mixtprob#(#)` is the logit transformation of the actual mixing probability.

When `scndstponly` is specified, the user also needs to specify some of the variables where the estimated residuals of *varlist_T* are stored and their variances. This is done using the following:

`st2(#)`, `st3(#)`, `st4(#)`, `st5(#)`, `st6(#)`, `st9(#)`, and `st12(#)` allow the user to provide the standard deviations of the residuals specified in `resvar#()`. These variances are given in the first step. They should also be provided using the logarithmic transformation. `st2(#)`, `st4(#)`, and `st5(#)` have to be provided only in the two-factor case. These options should be used only when `scndstponly` has been specified.

`resvar2(`*varname*`)`, `resvar3(`*varname*`)`, `resvar4(`*varname*`)`, `resvar5(`*varname*`)`, `resvar6(`*varname*`)`, `resvar9(`*varname*`)`, and `resvar12(`*varname*`)` contain the residuals of the test equations' estimations. `resvar2(`*varname*`)` and `resvar3(`*varname*`)` refer to the second to last and last variable of the first block of tests. `resvar4(`*varname*`)`, `resvar5(`*varname*`)`, and `resvar6(`*varname*`)` refer to the third to last, the second to last, and last variable of the second block of tests. `resvar9(`*varname*`)` and `resvar12(`*varname*`)` refer to the last variable of the third block and the last variable of the fourth block. These residuals are given by the first step under the names `__res2`, `__res3`, `__res4`, `__res5`, `__res6`, `__res9`, and `__res12`. `resvar2(`*varname*`)`, `resvar4(`*varname*`)`, and `resvar5(`*varname*`)` have to be provided only in the two-factor case. These options should be used only when `scndstponly` has been specified.

When `scndstponly` is specified, the user also needs to specify some of the matrices reported in the first stage to correct the standard errors of the second stage because there was a previous step in the estimation. This is done using the following options:

`firstloads(`*matname*`)` provides the name of the matrix where the loadings of the first stage are stored.

`firstgrad(`*matname*`)` provides the name of the matrix where the gradient of the first stage is stored.

`firstvarmat(`*matname*`)` provides the name under which the variance–covariance matrix of the first stage is stored.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals of the coefficients. The default is `level(95)` or as set by `set level`; see [R] **level**.

maximize_options: `difficult`, `iterate(#)`, [`no`]`log`, `trace`, `gradient`, `showstep`, `hessian`, `shownrtolerance`, `tolerance(#)`, `ltolerance(#)`, `gtolerance(#)`, `nrtolerance(#)`, `nonrtolerance`; see [R] **maximize**. These options are seldom used.

3.3 Further remarks

1. `heterofactor` typically requires relatively large samples, especially if it is used in a setting with more than one factor. The structural model is estimating not only several parameters (that is, $\widehat{\beta}^{Y}, \alpha^{Y,A}, \alpha^{Y,B}, \widehat{\beta}^{T}, \alpha^{T,A}, \alpha^{T,B}$) but also the distributions of unobservable attributes $\widehat{F}_{\theta}(\cdot)$.

2. `heterofactor` is computationally demanding because of the nonparametric way the unobserved factors' distributions are estimated. Numerical integration in a complex likelihood function, together with the numerical calculation of the gradient and Hessian during optimization, puts pressure on the computational resources available. Thus the estimation time increases with sample size, the number of observable controls, and the number of nodes used in the Gauss–Hermite quadrature for the numerical integration. For instance, it took 302.51 seconds for a MacBook Pro with 3.1 GHz Intel Core i7 and 16 GB memory to estimate the one-factor example presented in section 4.1. The same machine took 2155.38 seconds to fit the two-factor model presented in section 4.1.

3. There are trade-offs between estimation time and precision and smoothness and concavity of the likelihood function. Using larger samples and more nodes increases precision but also increases estimation time. Analogously, using more observable controls increases smoothness and concavity in the likelihood function. This is because the factors are being estimated from the residuals left after controlling for the observed variables. Therefore, a "cleaner" residual leads to an easier estimation of the factors and thus a smoother likelihood to maximize. However, more observable controls implies higher dimensions of the Hessian of the likelihood.

4. As explained in subsection 3.1, the estimated standard deviations and mixing probabilities are in transformed terms. This is done to avoid the optimization routine using unfeasible values. In particular, standard deviations should always be positive, and the mixing probabilities should always be in the $(0, 1)$ interval. Therefore, the standard deviations are transformed using the exponential function, and the mixing probabilities are transformed using a logit function. Thus, if s is the number provided by the estimation results for the standard deviations, then the actual standard deviation value is $\sigma = \exp(s)$. If the number provided by the estimation results for the mixing probability is p, then the actual mixing probability value is $\varrho = \exp(p)/\{1 + \exp(p)\}$.

5. Here are some practical recommendations for using `heterofactor`:

    a. When doing the first exploratory analyses, users should try using few integration nodes. This will decrease estimation time but will give a very well-informed indication on how the estimations will look.

    b. Given that the likelihood function is complex, convergence can be difficult. Recall that more control variables (sensible and informative) facilitate convergence. When convergence has been elusive, users are encouraged to use all the available tools in maximum likelihood estimation to improve the chances of convergence (see [R] **maximize**). For instance, Stata's maximum likelihood option `difficult` can be helpful in this case. `heterofactor` also offers the `initialreg` option, which provides a set of initial values different from the ones provided by Stata. As in any complicated likelihood, convergence might depend on the initial values. Trying with different sets of initial values is encouraged when convergence is difficult.

6. `heterofactor` requires the `matdelrc` command (Cox 1999), which can be downloaded by typing `search matdelrc` in the Command window.

7. The `heterofactor` routines are written in Mata and thus compiled in a library called `lheterofactor.mlib` (see [M-3] **mata mlib**). The library must be placed in a folder where Stata will look for it. However, before you call the library for the first time, you must type `mata mlib index` in the Mata prompt. See [M-3] **mata mlib** for details.

8. `heterofactor` creates the following variables every time it runs the first stage:

    a. `__res#`: the estimated residuals for each variable in *varlist_T* (that is, $\mathbf{res} = \mathbf{T} - \mathbf{X}_T\beta^T$), where # is given according to the order in *varlist_T*

    b. `mixt#`: the random draws of the estimated distributions of θ, which provide a way to explore the shape of the distributions, where # represents the factor number

3.4 Stored results

`heterofactor` saves numerous results in `ereturn`. The ones produced during the first stage are crucial because they will be used in a future second-stage estimation, if needed. For instance, for a two-factor model, the main stored results after the first stage are the following:

Scalars

`e(N)`	number of observations
`e(sf11)`	standard deviation of first normal used in mixture defining first factor
`e(sf12)`	standard deviation of second normal used in mixture defining first factor
`e(mu11)`	mean of first normal used in mixture defining first factor
`e(p1)`	mixing probability for mixture of normals defining first factor
`e(sf21)`	standard deviation of first normal used in mixture defining second factor
`e(p2)`	mixing probability for mixture of normals defining second factor
`e(mu21)`	mean of first normal used in mixture defining second factor
`e(sf22)`	standard deviation of second normal used in mixture defining second factor

Matrices

`e(b)`	coefficient vector
`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
`e(g11)`	gradient vector of the first step
`e(V11)`	variance–covariance matrix of the first step
`e(sT2)`	standard deviation of residuals of the second block of tests
`e(aT2)`	factor loadings of the second block of tests
`e(sT1)`	standard deviation of residuals of the first block of tests
`e(aT1)`	factor loadings of the first block of tests
`e(coeff_T6)`	coefficient vector of test 6
`e(coeff_T5)`	coefficient vector of test 5
`e(coeff_T4)`	coefficient vector of test 4
`e(coeff_T3)`	coefficient vector of test 3
`e(coeff_T2)`	coefficient vector of test 2
`e(coeff_T1)`	coefficient vector of test 1

Functions

`e(sample)`	marks estimation sample

`e(sf11)`, `e(sf12)`, `e(mu11)`, `e(p1)`, `e(sf21)`, `e(p2)`, `e(mu21)`, and `e(sf22)` provide the distributional parameters of the two factors. `e(g11)` and `e(V11)` provide the gradient and the variance–covariance of the parameters in the first stage. `e(aT1)` and `e(aT2)` are matrices that collect the loadings associated with each block in *varlist_T*. `e(sT1)` and `e(sT2)` are matrices that collect the variances of the estimated residuals for each block in *varlist_T*. Finally, `e(coeff_T#)` are vectors that collect the coefficients of the observable controls for each variable in *varlist_T* (that is, β^T).

The main results stored after a second stage are the following:

Scalars

`e(N)`	number of observations
`e(av1)`	loading of first factor on the choice equation
`e(av2)`	loading of second factor on the choice equation
`e(sY0)`	standard deviation of residuals of Y_0 equation
`e(a01)`	loading of first factor on Y_0 equation
`e(a02)`	loading of second factor on Y_0 equation
`e(a12)`	loading of second factor on Y_1 equation
`e(a11)`	loading of first factor on Y_1 equation
`e(sY1)`	standard deviation of residuals of Y_1 equation

Matrices

`e(b)`	coefficient vector
`e(ilog)`	iteration log
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
`e(coeff_Y1)`	coefficient vector of equation Y_1
`e(coeff_Y0)`	coefficient vector of equation Y_0
`e(coeff_D)`	coefficient vector of choice equation

Functions

`e(sample)`	marks estimation sample

`e(av1)` and `e(av2)` are scalars that collect the loadings of each factor in the choice equation. `e(a01)`, `e(a02)`, `e(a12)`, and `e(a11)` are the scalars that store the loadings of each factor for the outcome equations when $D = 0$ and $D = 1$. Likewise, `e(sY0)` and `e(sY1)` are scalars that store the variance of the residual of the outcome equations. Matrices `e(coeff_Y1)`, `e(coeff_Y0)`, and `e(coeff_D)` collect the coefficients of the observable controls for the outcome equations when $D = 0$ and $D = 1$ and for the choice equation, respectively (that is, β^{Y_0}, β^{Y_1}, and β^{D}).

`heterofactor` estimates multiple equations; it is not compatible with the `predict` postestimation command. Instead, it provides users with vectors stored in `e(.)` to create the predicted values of the desired equations. (See [P] **matrix score** for details on how vectors can be used to create variables with predicted values.)

4 Examples

In this section, we illustrate the `heterofactor` command using both simulated and real data (that is, the NLSY79). We use simulated data as a benchmark for the precision of the estimates in different structures.

4.1 Examples with simulated data

We present three examples using simulated data, all of which use the treatment-effect structure. First, we present a one-factor model. Then, we present a two-factor model assuming a loadings structure as in (13). Finally, we present a two-factor model assuming a triangular loadings structure as in (12).

To show how to empirically recover the parameters from this model, we present the following parameterization:

$$
\begin{aligned}
\theta^A &\sim 0.3\mathcal{N}(0,1) + 0.7\mathcal{N}(-0.428, 0.387) \\
\theta^B &\sim 0.5\mathcal{N}(0,1) + 0.5\mathcal{N}(-0.5, 0.5) \\
(\mathbf{e}_T, \mathbf{e}_Y, X, Z, Q) &\sim \mathcal{N}(0,1) \\
T_1 &= 0.1 + 0.1Q + 1.1\theta^A + e_1 \\
T_2 &= 0.5 + 0.1Q + 1.4\theta^A + e_2 \\
T_3 &= 0.4 + 0.3Q + \theta^A + e_3 \\
T_4 &= 0.3 + 0.11Q + 3\theta^B + e_4 \\
T_5 &= 0.4 + 0.21Q + 1.6\theta^B + e_5 \\
T_6 &= 0.1 + 0.31Q + \theta^B + e_6 \\
T_7 &= 0.3 + 0.11Q + 3.1\theta^A + 3\theta^B + e_7 \\
T_8 &= 0.4 + 0.21Q + 1.2\theta^A + 1.6\theta^B + e_8 \\
T_9 &= 0.1 + 0.31Q + 2\theta^A + \theta^B + e_9 \\
D &= \begin{cases} 1 & \text{if } 0.5Z + \theta^A + e_D > 0 \\ 0 & \text{otherwise} \end{cases} \\
Y_1 &= 2 + 2X + 2\theta^A + e_{Y^1} \\
Y_0 &= 1.5 + X + \theta^A + e_{Y^0} \\
D_2 &= \begin{cases} 1 & \text{if } 0.5Z + \theta^A + \theta^B + e_{D_2} > 0 \\ 0 & \text{otherwise} \end{cases} \\
Y_{2,1} &= 2 + 2X + 2\theta^A + 2\theta^B + e_{Y_2^1} \\
Y_{2,0} &= 1.5 + X + \theta^A + \theta^B + e_{Y_2^0}
\end{aligned}
$$

The results were produced using Stata/MP 8 14.1. Your results may vary if you are using a different flavor (that is, Stata/SE, Stata/MP 2, etc.) of Stata. We create our data using the following:

```
. set seed 12345
. set obs 5000
obs was 0, now 5000
. generate u1=runiform()
. generate u2=runiform()
. generate f1 = rnormal()*sqrt(1)+1 if u1<0.3
(3486 missing values generated)
. replace f1 = rnormal()*0.622269-0.42857143 if u1>=0.3
(3486 real changes made)
. generate f2 = invnormal(runiform())*sqrt(1) + 0.5 if u2<0.5
(2536 missing values generated)
```

```
. replace f2 = invnormal(runiform())*0.70710678 -0.5 if u2>=0.5
(2536 real changes made)
. drop u?
. generate X=rnormal()
. generate Q=rnormal()
. generate Z=rnormal()
. generate uv=rnormal()
. generate u1=rnormal()
. generate u0=rnormal()
. forvalues i=1/12{
  2.      generate e`i´=rnormal()
  3. }
. generate t1=0.1 +0.1 *Q +1.1*f1+e1
. generate t2=0.5 +0.1 *Q +1.4*f1+e2
. generate t3=0.4 +0.3 *Q + f1+e3
. generate t4=0.3 +0.11*Q +3 *f2 +e7
. generate t5=0.4 +0.21*Q +1.6*f2 +e8
. generate t6=0.1 +0.31*Q + f2 +e9
. generate t7=0.3 +0.11*Q +3.1*f1 + 3*f2 +e7
. generate t8=0.4 +0.21*Q +1.2*f1 + 1.6*f2 +e8
. generate t9=0.1 +0.31*Q +2*f1 + f2 +e9
. generate D=(0.5*Z + f1 + uv>0)
. generate Y11=2 +2*X + 2*f1 + u1
. generate Y10=1.5 + X + f1 + u0
. generate Y1=D*Y11 + (1-D)*Y10
. generate d2=(0.5*Z + f1 + f2 + uv>0)
. generate Y21=2 +2*X + 2*f1 + 2*f2 + u1
. generate Y20=1.5 + X + f1 + f2 + u0
. generate Y2=d2*Y21 + (1-d2)*Y20
```

One-factor model

Here we present a case where the system is described by only one factor, as in footnote 7. The command is

```
. heterofactor Y1 X, treatind(D) instrum(Z) scores(t1 t2 t3) indvarsc(Q)
> factors(1) difficult initialreg
Estimating Initial Values Vector
Running Factor Model
Iteration 0:   log likelihood = -40882.22 (not concave)
Iteration 1:   log likelihood = -38557.065 (not concave)
  (output omitted)
Iteration 6:   log likelihood = -35931.868
Iteration 7:   log likelihood = -35931.69
Iteration 8:   log likelihood = -35931.69
                                                Number of obs     =      5,000
                                                Wald chi2(1)      =     451.15
Log likelihood = -35931.69                      Prob > chi2       =     0.0000
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
D1						
Z	.5068783	.0238641	21.24	0.000	.4601055	.5536512
_cons	-.0237893	.0251811	-0.94	0.345	-.0731434	.0255648
Y11						
xw	1.975667	.0276118	71.55	0.000	1.921549	2.029785
_cons	1.997345	.042107	47.44	0.000	1.914817	2.079873
Y10						
xw	1.015075	.020827	48.74	0.000	.9742545	1.055895
_cons	1.468498	.0315469	46.55	0.000	1.406667	1.530329
t1						
x	.0998038	.01589	6.28	0.000	.0686601	.1309476
_cons	.0861379	.0206553	4.17	0.000	.0456543	.1266216
t2						
x	.0997367	.0170335	5.86	0.000	.0663516	.1331219
_cons	.5030196	.0242277	20.76	0.000	.4555342	.5505049
t3						
x	.302106	.0157761	19.15	0.000	.2711855	.3330266
_cons	.4174916	.0196586	21.24	0.000	.3789615	.4560217
/a1	2.122694	.0449329	47.24	0.000	2.034627	2.210761
/a0	1.043721	.0433614	24.07	0.000	.9587339	1.128708
/av	1.019103	.0409113	24.91	0.000	.9389189	1.099288
/aT1	1.129494	.0242495	46.58	0.000	1.081966	1.177022
/aT2	1.482229	.0290633	51.00	0.000	1.425266	1.539192
/sig1	.0351039	.0258972	1.36	0.175	-.0156537	.0858614
/sig0	.0098692	.0170994	0.58	0.564	-.023645	.0433833
/sigT1	-.0006658	.0120529	-0.06	0.956	-.0242892	.0229575
/sigT2	-.0093154	.0145453	-0.64	0.522	-.0378236	.0191928
/sigT3	.0210075	.0114337	1.84	0.066	-.001402	.0434171
/sigf1	.0030812	.033485	0.09	0.927	-.0625482	.0687106
/sigf2	-.4976842	.0340155	-14.63	0.000	-.5643534	-.431015
/p1	-.666261	.1045168	-6.37	0.000	-.8711102	-.4614118
/mu1	.8015842	.0598374	13.40	0.000	.684305	.9188635

```
Done Estimating Factor Model
```

In this output, `/a1` and `/a0` indicate the factor loadings for the equation of Y_1 and Y_0, respectively. Likewise, `/av` indicates the estimand of the factor loading in the choice equation, while `/aT1` and `/aT2` are the factor loadings for measures T_1 and T_2, respectively. Note that the reported standard deviations (that is, `/sigf1` and `/sigf2`) and the mixture-combining probability (that is, `/p1`) are transformed. To retrieve the actual values, we need to transform them back as follows:

```
. display exp(_b[sigf1:_cons])
.90781715
. display exp(_b[sigf2:_cons])
.62749649
. display invlogit(_b[p1:_cons])
.24802025
```

The command provides a random draw from the estimated factor distribution under the name `mixt`. That is, the program creates a variable for the user to plot the distribution that results from the estimated mixture of normals. Here we use this variable to show the accuracy of the estimation by comparing it with the true distribution of θ^A.

```
. kdensity mixt, addplot(kdensity f1) scheme(sj)
> legend(order(2 1) label(1 "Estimated factor") label(2 "True factor"))
> xtitle("")
```

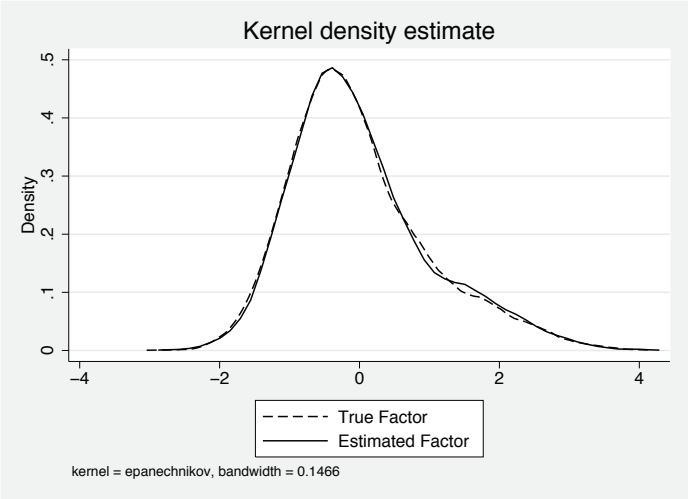

Figure 1. Actual and estimated factor 1

Two-factor model

Here we present a two-factor model assuming the loadings structure presented in (13). We use measures T_1 to T_6. The output will be divided into three parts: one part for the estimation of the first factor's distribution, one for the estimation of the second factor's

distribution, and one for the estimation of the outcomes and choice equations. We fit the model using the following command:

```
. heterofactor Y2 X, treat(d2) instrum(Z) scores(t1 t2 t3 t4 t5 t6) indvarsc(Q)
> factors(2) initialreg difficult nohats
Estimating Initial Values Vector
Running Factor Model

Twostep option specified
   Step: 1

Factor:  1

Iteration 0:   log likelihood = -27963.837  (not concave)
Iteration 1:   log likelihood = -26648.695  (not concave)
  (output omitted)
Iteration 9:   log likelihood = -25333.818
Iteration 10:  log likelihood = -25333.818
                                                Number of obs     =      5,000
                                                Wald chi2(1)      =      31.72
Log likelihood = -25333.818                     Prob > chi2       =     0.0000
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
t1						
Q	.1106338	.0196421	5.63	0.000	.072136	.1491316
_cons	.0934989	.0209643	4.46	0.000	.0524096	.1345882
t2						
Q	.1140862	.0227646	5.01	0.000	.0694685	.1587039
_cons	.512667	.0246802	20.77	0.000	.4642947	.5610394
t3						
Q	.3116778	.0188027	16.58	0.000	.2748252	.3485303
_cons	.4240101	.0199136	21.29	0.000	.3849802	.46304
/aT11	1.131651	.0266184	42.51	0.000	1.079479	1.183822
/aT21	1.500883	.0346451	43.32	0.000	1.43298	1.568786
/sigT1	.0054918	.0143887	0.38	0.703	-.0227096	.0336931
/sigT2	-.0210072	.0212032	-0.99	0.322	-.0625647	.0205502
/sigT3	.0273248	.0127516	2.14	0.032	.0023321	.0523174
/sigf11	-.0382022	.0866956	-0.44	0.659	-.2081224	.131718
/sigf12	-.4788531	.04016	-11.92	0.000	-.5575653	-.4001409
/p1	-1.000756	.2626884	-3.81	0.000	-1.515616	-.4858966
/mu1	1.043295	.2033173	5.13	0.000	.6448008	1.44179

```
Factor:  2
Iteration 0:   log likelihood =  -32280.37  (not concave)
Iteration 1:   log likelihood = -30724.776  (not concave)
  (output omitted)
Iteration 9:   log likelihood = -27822.353
Iteration 10:  log likelihood = -27822.351
                                                Number of obs     =      5,000
                                                Wald chi2(1)      =      11.22
Log likelihood = -27822.351                     Prob > chi2       =     0.0008
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
t4						
Q	.1498471	.0447385	3.35	0.001	.0621612	.237533
_cons	.2250401	.0443057	5.08	0.000	.1382024	.3118777
t5						
Q	.2303199	.0266435	8.64	0.000	.1780997	.2825401
_cons	.3583129	.026471	13.54	0.000	.3064307	.4101952
t6						
Q	.3066849	.0202724	15.13	0.000	.2669518	.3464181
_cons	.0861055	.0202484	4.25	0.000	.0464193	.1257916
/aT42	2.887153	.0466271	61.92	0.000	2.795765	2.97854
/aT52	1.564409	.0270325	57.87	0.000	1.511426	1.617392
/sigT4	.0762297	.029669	2.57	0.010	.0180796	.1343799
/sigT5	-.0180211	.0153144	-1.18	0.239	-.0480368	.0119946
/sigT6	.0060149	.0111457	0.54	0.589	-.0158304	.0278601
/sigf21	-.1622406	.0272878	-5.95	0.000	-.2157236	-.1087575
/sigf22	-.3072059	.0274006	-11.21	0.000	-.36091	-.2535018
/p2	-.625175	.1279547	-4.89	0.000	-.8759617	-.3743883
/mu2	.8941899	.0585462	15.27	0.000	.7794415	1.008938

```
Second Stage: Estimation
Iteration 0:   log likelihood = -56305.239  (not concave)
Iteration 1:   log likelihood = -55062.643  (not concave)
  (output omitted)
Iteration 5:   log likelihood = -53286.876
Iteration 6:   log likelihood = -53286.876
                                                Number of obs     =      5,000
                                                Wald chi2(1)      =     354.40
Log likelihood = -53286.876                     Prob > chi2       =     0.0000
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
d2						
Z	.5071589	.0269401	18.83	0.000	.4543573	.5599605
_cons	-.0062522	.0246106	-0.25	0.799	-.054488	.0419836
Y21						
X	1.968805	.0306787	64.17	0.000	1.908676	2.028934
_cons	2.009102	.0417183	48.16	0.000	1.927335	2.090868
Y20						
X	1.017017	.0221343	45.95	0.000	.9736349	1.0604
_cons	1.457208	.0335403	43.45	0.000	1.39147	1.522946
/a11	2.084713	.0370919	56.20	0.000	2.012014	2.157412
/a12	1.910914	.0361597	52.85	0.000	1.840042	1.981786
/a01	1.066335	.0420311	25.37	0.000	.9839559	1.148715
/a02	.9777517	.0296384	32.99	0.000	.9196615	1.035842
/av1	1.034998	.0443362	23.34	0.000	.9481003	1.121895
/av2	.9655901	.0363247	26.58	0.000	.8943949	1.036785
/aT21	1.484669	.0199244	74.51	0.000	1.445618	1.52372
/aT42	2.884038	.0252152	114.38	0.000	2.834617	2.933459
/aT52	1.56073	.0176811	88.27	0.000	1.526076	1.595384
/sig1	.0673842	.0321999	2.09	0.036	.0042735	.1304948
/sig0	.001	.0196342	0.05	0.959	-.0374824	.0394823

When you fit a model with two factors, the output includes an extra digit to identify the factor it is referring to. For instance, in the second-stage estimation, `/a11` indicates the loading of the first factor in the equation for Y_1, and `/a12` indicates the loading of the second factor in the same equation. That is, $\alpha^{Y_1,A}$ and $\alpha^{Y_1,B}$. Similarly, `/a01` indicates $\alpha^{Y_0,A}$, and `/a02` indicates $\alpha^{Y_0,B}$. To show the accuracy of our estimates of $F_{\theta^B}(\zeta)$, we plot it together with the true factor in figure 2.

```
. kdensity mixt2, addplot(kdensity f2) scheme(sj)
> legend(label(1 "Estimated Factor") label(2 "True Factor")) xtitle("")
```

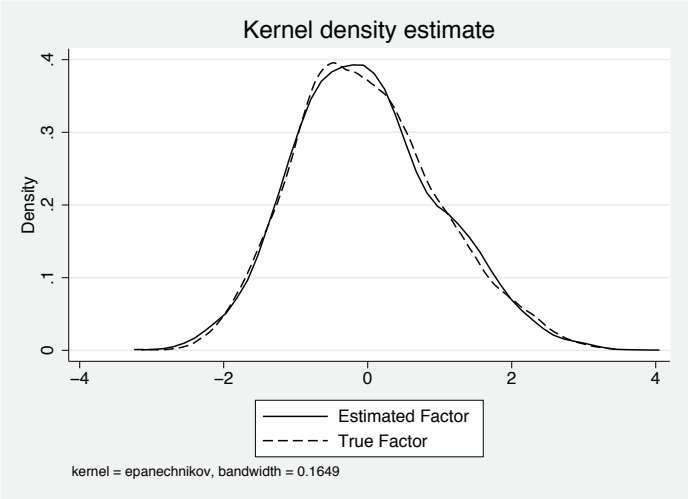

Figure 2. Actual and estimated factor 2 using structure (13)

Two factors—Triangular loadings structure

Now, we run a model that assumes that the measurement system (4) has a triangular loadings structure as in (12). Note that the estimation of the system that is affected by the first factor is exactly the same as in subsection 4.1. Therefore, we omit that part of the output.

```
. heterofactor Y2 X, treat(d2) instrum(Z) scores(t1 t2 t3 t7 t8 t9) indvarsc(Q)
> factors(2) triangular initialreg difficult nohats
Estimating Initial Values Vector
Running Factor Model
Twostep option specified
   Step: 1
Factor:  1
  (output omitted)
Factor:  2
Iteration 0:   log likelihood = -62751.068  (not concave)
Iteration 1:   log likelihood = -60586.731  (not concave)
  (output omitted)
Iteration 13:  log likelihood = -53991.903
Iteration 14:  log likelihood = -53991.789
Iteration 15:  log likelihood = -53991.789

                                                Number of obs     =      5,000
                                                Wald chi2(1)      =      16.62
Log likelihood = -53991.789                     Prob > chi2       =     0.0000
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
t7						
Q	.1775851	.0435664	4.08	0.000	.0921966	.2629736
_cons	.2291283	.0483811	4.74	0.000	.134303	.3239536
t8						
Q	.2378664	.0253233	9.39	0.000	.1882337	.2874991
_cons	.3582543	.0275752	12.99	0.000	.3042079	.4123007
t9						
Q	.329379	.0221729	14.86	0.000	.2859209	.3728371
_cons	.0928807	.0236972	3.92	0.000	.0464349	.1393264
/aT41	3.239198	.0510511	63.45	0.000	3.139139	3.339256
/aT51	1.229221	.0290314	42.34	0.000	1.17232	1.286121
/aT61	2.060524	.0270328	76.22	0.000	2.007541	2.113508
/aT42	2.909617	.0534597	54.43	0.000	2.804838	3.014396
/aT52	1.552329	.0324726	47.80	0.000	1.488684	1.615974
/aT11	1.118679	.0179589	62.29	0.000	1.08348	1.153878
/aT21	1.481497	.0191889	77.21	0.000	1.443887	1.519107
/sigT4	-.0069295	.0359718	-0.19	0.847	-.0774329	.0635739
/sigT5	.0012072	.0156359	0.08	0.938	-.0294387	.0318531
/sigT6	.0145307	.0132415	1.10	0.272	-.0114222	.0404837
/sigf21	-.1259575	.0345348	-3.65	0.000	-.1936446	-.0582704
/sigf22	-.3364971	.0361957	-9.30	0.000	-.4074393	-.2655549
/p2	-.3397048	.1032454	-3.29	0.001	-.5420622	-.1373475
/mu2	.7849042	.0503796	15.58	0.000	.6861619	.8836465

```
Second Stage: Estimation
Iteration 0:   log likelihood = -57322.997  (not concave)
Iteration 1:   log likelihood = -55183.622  (not concave)
  (output omitted)
Iteration 5:   log likelihood = -53153.727
Iteration 6:   log likelihood = -53153.727
                                                Number of obs     =      5,000
                                                Wald chi2(1)      =     379.62
Log likelihood = -53153.727                     Prob > chi2       =     0.0000
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
d2						
Z	.4985125	.025586	19.48	0.000	.4483649	.5486602
_cons	-.0167719	.0231616	-0.72	0.469	-.0621677	.028624
Y21						
X	1.993162	.0229973	86.67	0.000	1.948088	2.038236
_cons	1.97525	.029863	66.14	0.000	1.91672	2.033781
Y20						
X	1.023553	.0200909	50.95	0.000	.984175	1.06293
_cons	1.456468	.0269527	54.04	0.000	1.403642	1.509294
/a11	2.068437	.0380411	54.37	0.000	1.993878	2.142996
/a12	1.91965	.0301661	63.64	0.000	1.860526	1.978775
/a01	1.050172	.035932	29.23	0.000	.9797465	1.120597
/a02	.9953106	.0291753	34.11	0.000	.9381281	1.052493
/av1	1.002611	.0376846	26.61	0.000	.928751	1.076472
/av2	.9590321	.0351233	27.30	0.000	.8901916	1.027873
/aT21	1.480445	.019431	76.19	0.000	1.442361	1.51853
/aT41	3.22221	.0483981	66.58	0.000	3.127352	3.317069
/aT42	2.893978	.0283124	102.22	0.000	2.838487	2.949469
/aT51	1.217317	.0278293	43.74	0.000	1.162773	1.271862
/aT52	1.54802	.0195086	79.35	0.000	1.509784	1.586256
/aT61	2.053388	.0256322	80.11	0.000	2.00315	2.103626
/sig1	.0266306	.0182996	1.46	0.146	-.0092359	.0624972
/sig0	-.0118307	.0150991	-0.78	0.433	-.0414245	.017763

```
. kdensity mixt2, addplot(kdensity f2) scheme(sj)
> legend(label(1 "Estimated Factor")  label(2 "True Factor")) xtitle("")
```

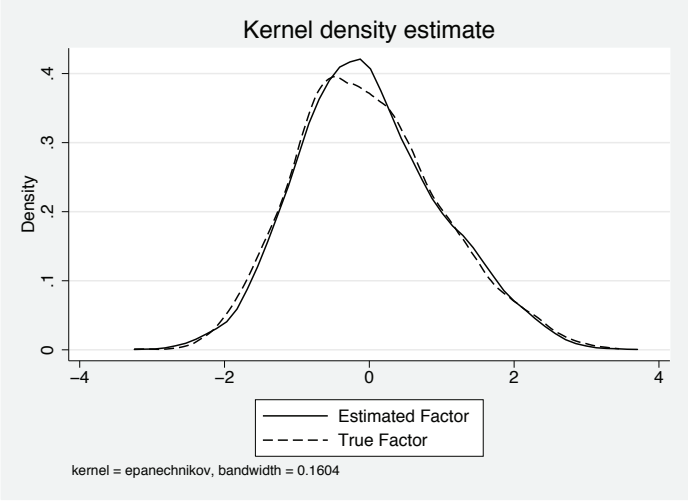

Figure 3. Actual and estimated factor 2 using triangular structure (12)

Again, to show the accuracy of our estimates of $F_{\theta^B}(\zeta)$ in this more complicated setting, we plot it together with the actual factor in figure 3.

4.2 Example using the NLSY79

In this section, we present an example using real data from NLSY79. The dataset is widely used by the research community (see, for instance, Heckman, Stixrud, and Urzua [2006], Urzúa [2008], Prada and Urzua [2013]). In our example, we fit a Roy model where the endogenous choice is whether the person went to college by age 25 and the potential outcomes are the log of earnings by age 30. The adjunct measurement system comprises the armed services vocational aptitude battery tests recorded during the participants' teenage years. The observable controls used in the test equations are race and mother's education. This last control is also used in the college enrollment decision. Finally, in the earning equations, we control for race and experience.

```
. use nlsyforfactor.dta, clear
. heterofactor lnincome blackwhite ExperienceF Experience2, treat(HR_5)
> instrum(HGC_MOTHER) scores(stASVAB_6 stASVAB_10 stASVAB_8)
> indvarsc(blackwhite HGC_MOTHER)
Running Factor Model
initial:       log likelihood = -23908.226
alternative:   log likelihood = -19308.277
rescale:       log likelihood = -19308.277
rescale eq:    log likelihood = -12760.811
Iteration 0:   log likelihood = -12760.811  (not concave)
```

(*output omitted*)

```
Iteration 21:  log likelihood = -10878.232
Iteration 22:  log likelihood = -10878.231
                                                  Number of obs   =       2188
                                                  Wald chi2(1)    =     152.51
Log likelihood = -10878.231                       Prob > chi2     =     0.0000
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
HR_5						
HGC_MOTHER	.2440448	.0197617	12.35	0.000	.2053125	.2827771
_cons	-3.977122	.2557827	-15.55	0.000	-4.478447	-3.475797
lnincome1						
blackwhite	.0545235	.1332389	0.41	0.682	-.2066198	.3156669
ExperienceF	.0582035	.0129243	4.50	0.000	.0328723	.0835347
Experience2	-.000468	.000138	-3.39	0.001	-.0007385	-.0001975
_cons	1.627223	.3299871	4.93	0.000	.9804607	2.273986
lnincome0						
blackwhite	.3306208	.0602065	5.49	0.000	.2126183	.4486233
ExperienceF	.0629075	.0069841	9.01	0.000	.049219	.0765961
Experience2	-.0003744	.0000741	-5.05	0.000	-.0005197	-.0002291
_cons	.4799472	.1643739	2.92	0.004	.1577802	.8021143
stASVAB_6						
blackwhite	.6175045	.0523823	11.79	0.000	.5148372	.7201719
HGC_MOTHER	.0933385	.0072596	12.86	0.000	.0791098	.1075671
_cons	-1.579435	.0983605	-16.06	0.000	-1.772218	-1.386652
stASVAB_10						
blackwhite	.4188891	.0469253	8.93	0.000	.3269172	.510861
HGC_MOTHER	.0880278	.0070668	12.46	0.000	.0741772	.1018785
_cons	-1.345903	.0993812	-13.54	0.000	-1.540687	-1.15112
stASVAB_8						
blackwhite	.6111189	.0568265	10.75	0.000	.499741	.7224968
HGC_MOTHER	.0676339	.0078114	8.66	0.000	.0523238	.082944
_cons	-1.285871	.105387	-12.20	0.000	-1.492425	-1.079316
/a1	.2686718	.0924776	2.91	0.004	.087419	.4499247
/a0	.2871036	.0478504	6.00	0.000	.1933186	.3808887
/av	1.829669	.1091898	16.76	0.000	1.61566	2.043677
/aT1	1.064874	.0456752	23.31	0.000	.975352	1.154396
/aT2	1.663054	.0630574	26.37	0.000	1.539464	1.786644
/sig1	-.3036387	.0300436	-10.11	0.000	-.3625231	-.2447543
/sig0	-.2164866	.0175662	-12.32	0.000	-.2509156	-.1820575
/sigT1	-.4138092	.0171057	-24.19	0.000	-.4473358	-.3802827
/sigT2	-1.311971	.076692	-17.11	0.000	-1.462285	-1.161657
/sigT3	-.2876287	.0162171	-17.74	0.000	-.3194136	-.2558439
/sigf1	-1.624681	.1030396	-15.77	0.000	-1.826635	-1.422727
/sigf2	-1.172679	.0643634	-18.22	0.000	-1.298829	-1.046529
/p1	-.6149592	.1085965	-5.66	0.000	-.8278045	-.4021139
/mu1	.5898844	.0333342	17.70	0.000	.5245506	.6552181

```
Done Estimating Factor Model
```

The results of this example indicate that people with higher levels of latent ability are more likely to go to college and to earn more.

5 Conclusions

Models of unobserved heterogeneity are becoming increasingly popular. However, their implementation is difficult and often tailored to the needs of each particular project. In this article, we presented code that can fit many models whose common feature is that they are systems of equations with latent-factor structures. Our code is flexible enough to incorporate different features of the data while keeping the distributional assumptions to the minimum. Although these models are computationally demanding, most estimations can be done using personal computers.

6 Acknowledgments

We thank Maria Prada for all of her contributions, especially for the development of the triangular loadings structure. We also thank Ricardo Espinoza, the Stata reviewer, as well as Koji Miyamoto, Katarzyna Kubacka, and the rest of the OECD-ESP team for their useful comments. All mistakes are ours.

7 References

Aakvik, A., J. J. Heckman, and E. J. Vytlacil. 2000. Treatment effects for discrete outcomes when responses to treatment vary among observationally identical persons: An application to Norwegian vocational rehabilitation programs. NBER Technical Working Paper No. 262. http://www.nber.org/papers/t0262.

Cameron, S. V., and J. J. Heckman. 1998. Life cycle schooling and dynamic selection bias: Models and evidence for five cohorts of American males. *Journal of Political Economy* 106: 262–333.

———. 2001. The dynamics of educational attainment for black, Hispanic, and white males. *Journal of Political Economy* 109: 455–499.

Carneiro, P., K. T. Hansen, and J. J. Heckman. 2003. 2001 Lawrence R. Klein lecture: Estimating distributions of treatment effects with an application to the returns to schooling and measurement of the effects of uncertainty on college choice. *International Economic Review* 44: 361–422.

Cox, N. J. 1999. dm69: Further new matrix commands. *Stata Technical Bulletin* 50: 5–9. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 28–34. College Station, TX: Stata Press.

Greene, W. H. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

Hansen, K. T., J. J. Heckman, and K. J. Mullen. 2004. The effect of schooling and ability on achievement test scores. *Journal of Econometrics* 121: 39–98.

Heckman, J. J., J. E. Humphries, S. Urzua, and G. Veramendi. 2011. The effects of educational choices on labor market, health, and social outcomes. Working Paper No. 2011-002, Human Capital and Economic Opportunity: A Global Working Group. http://humcap.uchicago.edu/RePEc/hka/wpaper/HHUV_2010_effect-edu-choice.pdf.

Heckman, J. J., and S. Navarro. 2007. Dynamic discrete choice and dynamic treatment effects. *Journal of Econometrics* 136: 341–396.

Heckman, J. J., J. Stixrud, and S. Urzua. 2006. The effects of cognitive and noncognitive abilities on labor market outcomes and social behavior. *Journal of Labor Economics* 24: 411–482.

Jöreskog, K. G., and A. S. Goldberger. 1972. Factor analysis by generalized least squares. *Psychometrika* 37: 243–260.

Judd, K. L. 1998. *Numerical Methods in Economics*. Cambridge, MA: MIT Press.

Keane, M. P., and K. I. Wolpin. 1997. The career decisions of young men. *Journal of Political Economy* 105: 473–522.

Kotlarski, I. I. 1967. On characterizing the gamma and the normal distribution. *Pacific Journal of Mathematics* 20: 69–76.

Prada, M. F., and S. Urzua. 2013. One size does not fit all: The role of vocational ability on college attendance and labor market outcomes. Working Paper. http://lacer.lacea.org/bitstream/handle/123456789/48629/lacea_2013_role_vocational_ability.pdf.

Roy, A. D. 1951. Some thoughts on the distribution of earnings. *Oxford Economic Papers* 3: 135–146.

Sarzosa, M. 2015. The dynamic consequences of bullying on skill accumulation. http://krannert.purdue.edu/faculty/msarzosa/Research/DynBullying.pdf.

Sarzosa, M., and S. Urzúa. 2015. Bullying among adolescents: The role of cognitive and noncognitive skills. NBER Working Paper No. 21631, The National Bureau of Economic Research. http://www.nber.org/papers/w21631.

Urzúa, S. 2008. Racial labor market gaps: The role of abilities and schooling choices. *Journal of Human Resources* 43: 919–971.

About the authors

Sergio Urzúa is an associate professor of economics at the University of Maryland. His primary research focuses on the role of cognitive ability, noncognitive skills, and early health status as determinants of schooling, labor market, and adult social behaviors. His research in applied

econometrics is mainly concerned with the estimation and identification of selection models with unobserved heterogeneity.

Miguel Sarzosa is an assistant professor of economics at Purdue University. His research focuses on the estimation of the effects that cognitive and noncognitive skills have on social behaviors, specifically, the effect that skill endowments have on in-school victimization and workplace discrimination.