Implementing Factor Models for Unobserved Heterogeneity in Stata: The `heterofactor` command*

Miguel Sarzosa†
Purdue University

Sergio Urzúa‡
University of Maryland

September 22, 2015

Abstract

We introduce a new Stata command for the Maximum Likelihood estimation of models with unobserved heterogeneity including a Roy Model. Our command estimates models with up to four latent factors. It allows the unobeserved heterogeneity to follow general distributions. In that regard, it differs from the `SEM` module in Stata as our command 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 using a mixture of normals in a Gauss-Hermite quadrature procedure. Our command delivers consistent estimates including the unobserved factor loadings in a wide array of model structures.

Key words: unobserved heterogeneity, factor models, maximum likelihood, numerical integration

*We would like to thank Maria Prada for all her contributions, especially the development of the triangular loadings structure. We would also like to thank Koji Miyamoto, Katarzyna Kubacka and the rest of the OECD-ESP team for their useful comments. All mistakes are ours.

†msarzosa@purdue.edu

‡urzua@econ.umd.edu

1 Introduction

Unobserved heterogeneity has become a particularly relevant topic in modern applied microeconomics (Keane and Wolpin, 1997; Cameron and Heckman, 1998, 2001; Carneiro et al., 2003; Heckman et al., 2006; Urzua, 2008; Sarzosa and Urzua, 2014). However, its adequate analysis requires the use of structural models that are often tailored to the needs of each particular research project. This reflects the fact that the research community lacks the tools needed for the systematic inclusion of unobserved heterogeneity in practical analyses. At the same time, the advances in computational capability has facilitated the estimation of structural models to a point where is now conceivable to run some of these models in standard available computers.

In this article, we discuss the implementation of factor models for the estimation of structural equations in the presence of unobserved heterogeneity, and a new Stata command that estimates them. The routines we introduce allow the calculation of consistent estimates including the loadings of the unobserved factors in a number of structures. The family of structural models we address is related the structures used in Carneiro et al. (2003); Heckman et al. (2006) and first introduced by Jöreskog and Goldberger (1972), Cameron and Heckman (1998) and Cameron and Heckman (2001). The most salient feature of these models is the presence of a factor structure that provides a parsimonious specification to identify unobserved heterogeneity and its effects on the outcomes of interest. Unlike the `SEM` module in Stata, 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 non-parametrically relying on the contributions of Kotlarski (1967).

As it will be shown below, this type of structural models have a wide range of applications. Recently, the treatment effect literature has embraced this kind of models not

only because they provide a way for estimating treatment effects depending on the level of unobservables, but also because controlling for unobserved heterogeneity allows the simulation of counterfactuals. This procedures 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 non-cognitive 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 Urzua, 2014).

The remainder of the article is organized as follows. In section 2, we review the factor model structure and discuss the mechanisms that allows us to identify key parameters. The implementation of our estimation routines, including thy syntax of the command is discussed in section 3. Then, in section 4, we provide some examples using both simulated data and the NLSY79. Finally in section 5, we conclude.

2 Factor Model Estimation

The type of structural models 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 models considered by Hansen et al. (2004), Heckman et al. (2006), Heckman and Navarro (2007), Heckman et al. (2006) and Sarzosa and Urzua (2014). In a general setup, suppose we face the following linear system:

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

where $\mathbf{Y}$ is a $M \times 1$ 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 one of the M

outcome equations with a factor structure of the form $\mathbf{U^Y} = \mathbf{\Lambda^Y \Theta} + \mathbf{e^Y}$. Hence, we can expand the linear system to

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

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

The measurement system (2) can be used to identify vectors $\alpha^{\mathbf{Y,A}}$ and $\alpha^{\mathbf{Y,B}}$, albeit under very stringent constraints and assumptions Aakvik et al. (2000). As indicated by Carneiro et al. (2003), the estimations that come from the factor structure will gain interpretability and will require less 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 following form:

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

where $\mathbf{T}$ is a $L \times 1$ vector of measurements (e.g., test scores), $\mathbf{X}_T$ is a matrix with all observable controls for each measurement and $\mathbf{\Lambda^Y}$ is a $L \times q$ matrix that holds the loadings of the q unobserved factors. Again, we assume that $(\mathbf{\Theta}, \mathbf{X}_T) \perp \mathbf{e^T}$, 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

The command can handle up to four factors. However, for presentation purposes we will describe the estimation process using a two-factor model.[2] In the two-factor case equation (1) becomes

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

and equation (2) becomes

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

To explain how the parameters of the adjunct measurement system (4) are identified, let us focus on the matrix $COV(\mathbf{T}|\mathbf{X}_T)$ whose elements in the diagonal are of the form:

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

and the off-diagonal elements are of the form:

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

As it is, the model is underidentified (Carneiro et al., 2003). Therefore, identification requires some assumptions. First, we need $\theta^A \perp \theta^B$, so $\sigma_{\theta^A\theta^B} = 0$ in (5) and (6).[3] The

[1] For the Maximum Likelihood procedure we describe below, we assume $f_{e^h}(\cdot)$ are normal distributions. This is a relatively mild assumption as these come from the idiosyncratic variation that remains after controlling for observed controls und unobserved heterogeneity.

[2] The extension to three and four factors is straightforward.

[3] Using higher moments of the distributions Heckman and Navarro (2007) show that identification

second assumption relates to the minimum number of measurements we need to have per factor. Notice that the diagonal elements of $COV\left(\mathbf{T}\,|\mathbf{X}_T\right)$ have the variances of the idiosyncratic errors, while the ones the off-diagonal 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 et al. (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 $k * L$ loadings. We then need 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 6 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 the rest of the loadings should be interpreted as relative to those used as nummeraire.[4] To incorporate this into our notation, let us expand (4) into k blocks of size m_κ such that $\sum_\kappa m_\kappa = L$. That way, with out loss of generality, we set equal to one the first loading in the first equation in each block. Therefore, in our example we

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 impossed on the factor loadings structure.

[4]This normalizations reduce by k the number of parameters to estimate. Therefore, the number of measurements needed L is given by $\frac{L(L-1)}{2} \geq Lk+k-k$, 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 we present in this paper, we will assume that we have at least $3k$ measurements.

get two blocks a and b. That is, we write (4) as

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

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

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 $COV\left(\mathbf{T}\,|\mathbf{X}_T\right)$ matrix follow one of the following cases:

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

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

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

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

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

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

for $\kappa = \{a, b\}$ and $i \neq j$. These elements show that we are not able to identify $\sigma^2_{\theta^A}$ and $\sigma^2_{\theta^B}$ and the loadings without further restrictions. Carneiro et al. (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 only be affected by the first factor. Then

$$COV\left(T_1^a, T_2^a\,|\mathbf{X}_T\right) = \alpha^{T_2^a,A}\sigma^2_{\theta^A}$$

$$COV\left(T_1^a, T_3^a\,|\mathbf{X}_T\right) = \alpha^{T_3^a,A}\sigma^2_{\theta^A}$$

$$COV\left(T_2^a, T_3^a\,|\mathbf{X}_T\right) = \alpha^{T_2^a,A}\alpha^{T_3^a,A}\sigma^2_{\theta^A}$$

Then,

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

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

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

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

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

respectively. Given that we already know $\sigma^2_{\theta^A}$, we are able to 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 $\alpha^{T^b, B}$ as 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 $COV\left(\mathbf{T} \left| \mathbf{X}_T \right.\right)$ matrix, we are able to identify the parameters in the diagonal. From (5) and the restrictions we have imposed we get that the typical diagonal element of $COV\left(\mathbf{T} \left| \mathbf{X}_T \right.\right)$ is

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

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, together with the fact that the means of θ^A, θ^B and $\mathbf{e^T}$ are finite—in fact, equal to zero because we allow the measurement system (4) to have intercepts—we can invoke the Kotlarski Theorem to use the manifest variables $\mathbf{T}$ to non-parametrically identify the distributions of $f_{\theta^A}(\cdot)$ and $f_{\theta^B}(\cdot)$ (Kotlarski, 1967).[5]

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 of 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 data available. The triangular structure presented in (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 non-cognitive one.

[5]The basic idea of the Kotlarski Theorem is that if there are three independent random variables e_{T_1}, e_{T_2} and θ and 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. Note that, 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 et al. (2003).

If data permits 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 speed the estimation process as it requires the procedure to estimate less parameters.

2.3 Estimation

We estimate the model (4) using maximum likelihood estimation (MLE). 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 due to their unobservable nature, obtaining $\hat{\beta}^T, \alpha^{T,A}, \alpha^{T,B}, \hat{F}_{\theta^A}(\cdot)$ and $\hat{F}_{\theta^B}(\cdot)$. 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 recreate 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 are able to identify a very wide range 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 are able to move on to estimate model (3). The likelihood function in this case 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 MLE will yield $\hat{\beta}^Y, \alpha^{Y,A}$ and $\alpha^{Y,B}$.[6]

Note that 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 estimating a model with two factors.[7]

2.4 The Treatment Effect Setting, a Roy Model

In this subsection, we go over the especial case of model (3) where there is a binary treatment (e.g., to go to college) and a later outcome (e.g., wages earned at age 30). This is one of the settings where the factor structure has received more attention (Heck-

[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, 2000).

[7] For the one factor case, equations (3) and (4) become $\mathbf{Y} = \mathbf{X}_Y \beta^Y + \alpha^Y \theta + \mathbf{e}^\mathbf{Y}$ and $\mathbf{T} = \mathbf{X}_T \beta^T + \alpha^\mathbf{T} \theta + \mathbf{e}^\mathbf{T}$, respectively. And the likelihood function would be

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

man et al., 2006; Urzua, 2008; Heckman et al., 2011; Prada and Urzua, 2013). The great advantage the factor structure has in this setting is that potential outcomes are separable in observables and unobservables. That is, conditional on θ 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 ATE, ATT and $ATUT$ 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, for example, treated and not treated, or high school and college. The choice is based on the following 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 (i.e., $D = 1$ 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 wage 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, $\mathbf{Y} = [Y_1, Y_0, D]'$. In this case the system

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

would be:

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

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

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

The second step likelihood function is given by:

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

In consequence, we obtain different parameter values for each potential outcomes. 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 the vector of outcomes is comprised only by the choice or treatment decision. That is, vector $\mathbf{Y} = D$, meaning that no potential outcome equations enter the second step. The

outcome equation to be estimated is (17). The complete likelihood would be:

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

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

In a similar way, we can arrive to a particular case where there is no choice or treatment equation and there is only one outcome Y. Then, the outcomes vector $\mathbf{Y} = Y$, and the outcome equation of interest will be:

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

In this case, the complete likelihood would be:

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

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

3 Implementation

3.1 The syntax of `heterofactor`

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*) factors(#) nodes(#) initialreg

exp1(*varlist*) exp2(*varlist*) exp3(*varlist*) exp4(*varlist*) triangular

fdistonly scndstponly twostep nohats nochoice choiceonly

sigmamixt11(#) sigmamixt12(#) mixtprob1(#) mumixt1(#)

sigmamixt21(#) sigmamixt22(#) mixtprob2(#) mumixt2(#)

sigmamixt31(#) sigmamixt32(#) mixtprob3(#) mumixt3(#)

sigmamixt41(#) sigmamixt42(#) mixtprob4(#) mumixt4(#)

st2(#) st3(#) st6(#) st4(#) st5(#) st9(#) st12(#)

resvar2(*varname*) resvar3(*varname*) resvar6(*varname*) resvar4(*varname*)

resvar5(*varname*) resvar9(*varname*) resvar12(*varname*) numf1tests(#)

numf2tests(#) firstloads(*matname*) firstgrad(*matname*) firstvarmat(*matname*)

level(#) log *mle_ 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**). Weights are not allowed. A description of the options that are specific to `heterofactor` is provided below.

Options

Main

`scores`(*varlist_ T*) specifies the variables that contains the scores of the measurement system. That is, vector **T** in (4). There needs to be at least three

variables specified in *varlist_T* per factor. Users may specify more than three variables per factor for models with one and two factors. If the model has three or four factors the user needs to specify three variables for the third and fourth factors. The order of *varlist_T* matters. Users must list them in blocks, where each block should be affected by the same factor or factors. Identification requires one loading normalization per factor. In consequence, 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, the loadings of the third, sixth, ninth and twelfth variable will be normalized to one. This arrangement is somewhat different if option `triangular` 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 only depend on one factor each, while last set of three measures will be affected by all factors.

`indvarsc`(*varlist_Q*) specify the observed variables that affect all test scores regressions. That is, $\mathbf{X}_T$ in (4). Note that *varlist_Q* can be the same as *varlist_X*. However, the user needs to specify both. There is no limit for the number of variables that can be specified in *varlist_Q*. If the user wants to specify different controls for each set of three measures, options `exp1()`, `exp2()`, `exp3()` and `exp4()` should be used.

`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 (i.e., $\mathbf{X}_D$ in (16)).

Model

`exp1(`*varlist*`), exp2(`*varlist*`), exp3(`*varlist*`) and exp4(`*varlist*`)` includes more controls to each set of three test equations in addition to those specified in *varlist_*Q which are common to all. This way, the user can add regressors that are believed to affect only one set of three scores and not the other ones.

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

`fdistonly` indicates Stata to only estimate the first step. That is, to estimate only measurement system (4), and obtain the parameters that describe the factors' distribution $F_\theta(\cdot)$ and the factor loadings. No outcome equation will be estimated. However, *depvar* and *varlist_*X should be provided even if they are not going to be used in the calculation.

`scndstponly` indicates Stata to only estimate the outcome equations. That is, to estimate the system provided by (3). No factor distribution identification take place. If `scndstponly` is specified, all the parameters that describe the factors' distributions $F_\theta(\cdot)$, the residuals of *varlist_*T, their variances and loadings should be provided by the user through additional options. This option is useful if the user did the first step before and now she only needs 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 only comprises a choice equation. The likelihood to be estimated is (17). No other outcomes are estimated. Not even the potential outcomes 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 do not include the binary treatment one. It indicates Stata that the model is not of the treatment effect nature described in subsection 2.4. The 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 Stata that the measurement system in the first step has a triangular loading structure like in (12). If `triangular` is specified, the structure assumed for the measurement system is one that has one block of scores that depend on all factors, and the rest of the block of scores depend only on one factor each. This option is only valid for the two-factor case. It should be noted 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(#)` specifies the number of tests used in each block of *varlist_T*. Specify only if the number 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 the first block and the last three variables are the second block.

Estimation

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

`twostep` divides the estimation process in two parts. First the factor identification

part (4) and then the outcome equations part (3). This option is only available for the one-factor case. If `factor`(1) option is specified, the default is no `twostep`.

`initialreg` makes the program calculate initial values using OLS regressions of each equation separately. These initial values are different to the ones provided by Stata in the absence of the `initialreg` option.

`nohats` estimated factor values $\hat{\theta}$ are not saved in data. This option speeds the command execution, especially in big data sets.

Sometimes, the user needs to estimate several models that are based on the same factor structure. In that case, the user needs to run the first step only once and save time by running all the required models using only the second step. For that, the user needs to specifying `scndstponly` and parameters that describe the distributions, the residuals, the var-cov matrix, and the gradient of the first stage. The distributions $F_\theta(\cdot)$ of the factors are obtained using a mixture of two normals. Therefore, to fully describe each factor's distribution we need the standard deviation and 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. They should be provided using:

`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, specify the standard deviations of the two distributions used in the mixture of the first, second, third and fourth factors respectively. Note that 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 option to be provided should be `sigmamixt11(0)`. Note that the values displayed int he output in the first stage are untransformed, and are the ones that need to be provided as they are to these options.

`mumixt1`(#), `mumixt2`(#), `mumixt3`(#) and `mumixt4`(#) specifies the mean of the first part of the mixture of each factor. The factor is centered at zero, therefore 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 probability used to combine the mixtures, provided by $\exp($`mixtprob1`$(\#))/(1 + \exp($`mixtprob1`$(\#)))$ in the case of factor 1, $\exp($`mixtprob2`$(\#))/(1 + \exp($`mixtprob2`$(\#)))$, in the case of factor 2, and so on.

`mixtprob1`(#), `mixtprob2`(#), `mixtprob3`(#) and `mixtprob4`(#) specifies the probability used to combine the two normal distributions into the mixture of normals for the distribution each factor. As in the case of the standard deviations, the value in `mixtprob1` is a transformed one: 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:

`resvar2`(*varname*), `resvar3`(*varname*), `resvar4`(*varname*), `resvar5`(*varname*), `resvar6`(*varname*), `resvar9`(*varname*) and `resvar12`(*varname*) are options that should be used only when the `scndstponly` option has been specified. They contain the residuals of the second to last and last variable of the first block (`resvar2`(*varname*), `resvar3`(*varname*)), the third to last, the

second to last and last variable of the second block (`resvar4`(*varname*), `resvar5`(*varname*), `resvar6`(*varname*)), the last variable of the third block (`resvar9`(*varname*)), and the last variable of the fourth block (`resvar12`(*varname*)). These residuals are given by the first step procedure under the names of `__res2`, `__res3`, `__res4`, `__res5`, `__res6`, `__res9` and `__res12`. The user can also provide them by constructing $\mathbf{res} = \mathbf{T} - \mathbf{X}_T\beta^T$ where $\mathbf{X}_T$ are the observable controls used in the first stage and β^T is the vector of coefficients estimated in the first stage.

`st2`(#), `st3`(#), `st4`(#), `st5`(#), `st6`(#), `st9`(#) and `st12`(#) are options that should be used only when the `scndstponly` option has been specified. These options allow the user to provide the standard deviations of the residuals specified in `resvar`#. These variances are given by the first step procedure. They should also be provided using the logarithmic transformation (i.e., as the were reported in the first stage).

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

`firstloads`(*namelist*) 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 var-cov matrix of the first stage is stored.

3.2 Further Remarks

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

2. `heterofactor` is computationally demanding because of the non-parametric way the unobserved factors' distributions are estimated. The use of numerical integration in a complex likelihood function, together with the numerical calculation of the gradient and Hessian during the optimization processes put pressure on the computational resources available. Consequently, 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, estimating the one-factor example presented in Section 4.1.1 took a MacBook Pro with 3.1GHz Intel Core i7 and 16GB memory 302.51 seconds. The same machine took 2155.38 seconds estimating the two-factor model presented in Section 4.1.2.

3. There are trade-offs between estimation time and precision, smoothness and concavity of the likelihood function. Larger samples and more nodes increase precision but increase estimation time. Analogously, the use of more observable controls increases smoothness and concavity in the likelihood function. That is the case, because the factors are being estimated from the residuals left after the observed variables have been controlled for. Therefore, a "cleaner" residual leads to an easier estimation of the factors and therefore a smoother likelihood to maximize. However, more observable controls imply 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 taking on unfeasible values. In particular, standard deviations should be always 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. In consequence, 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. Some practical recommendations for `heterofactor` users are the following:

   (a) When doing the first exploratory analyses, the user should try with few integration nodes. This will decrease estimation time, but will give a very well informed indication on how the estimations will look like.

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

6. `heterofactor` requires the `matdelrc` command that can be downloaded from the

web by typing `-findit matdelrc-` in the command window.

7. The heterofactor routines are written in `mata`. Therefore, the are compiled and kept in a library called lheterofactor.mlib (see [M-3] `mata mlib`). Make sure that you place the library in a folder where Stata looks for it. However, that is not enough. Before you call the library for the very first time, you need to type `-mata mlib index-` in the mata prompt. See [M-3] `mata mlib` for details and further explanation.

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

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

   (b) `mixt#`. A random draw of the estimated distributions of θ. They provide a way to explore the shape of the distributions using. # represents the factor number.

3.3 Post Estimation Stored Results

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

```
scalars:
                  e(N)
              e(sf11)
              e(sf12)
              e(mu11)
                e(p1)
              e(sf21)
```

```
                e(p2)
              e(mu21)
              e(sf22)

matrices:
                e(b)
                e(V)
              e(g11)
              e(V11)
              e(sT2)
              e(aT2)
              e(sT1)
              e(aT1)
         e(coeff_T6)
         e(coeff_T5)
         e(coeff_T4)
         e(coeff_T3)
         e(coeff_T2)
         e(coeff_T1)
             e(ilog)
         e(gradient)

functions:
           e(sample)
```

Where `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 var-cov of the parameters in the first stage. `e(aT1)` and `e(aT2)` are matrices that collect the loadings associated to 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* (i.e., β^T).

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

```
scalars:
                e(N)
```

```
                e(av1)
                e(av2)
                e(sY0)
                e(a01)
                e(a02)
                e(a12)
                e(a11)
                e(sY1)

matrices:
                  e(b)
                  e(V)
           e(coeff_Y1)
           e(coeff_Y0)
            e(coeff_D)
               e(ilog)
           e(gradient)

functions:
             e(sample)
```

Where `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 $D = 1$ respectively. In the same way, `e(sY0)` and `e(sY1)` are scalars storing the variance of the residual of the outcome equations. Finally, 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 the choice equation respectively (i.e.., β^{Y_0}, β^{Y_1} and β^{D}).

`heterofactor` estimates multiple equations and it is not compatible with the `predict` postestimation command. Instead, it provides the 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 present illustrations of the `heterofactor` command using both simulated and real data (i.e., the NLSY79). We use simulated data in order to have a benchmark for the precision of the estimates in different structures.

4.1 Examples with simulated data

We will present three examples using simulated data, all of them using the treatment effect structure. First, a one-factor model. Then, 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).

In order to show how to empirically recover the parameters from this model, consider the following parameterization:

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

We create our data using the following Stata code:

```
. set seed 12345
. set obs 5000
obs was 0, now 5000
. gen u1=uniform()
. gen u2=uniform()
. gen 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)
. gen f2     = invnorm(uniform())*sqrt(1) + 0.5 if u2<0.5
(2536 missing values generated)
. replace f2 = invnorm(uniform())*0.70710678 -0.5 if u2>=0.5
(2536 real changes made)

. gen X=rnormal()
. gen Q=rnormal()
. gen Z=rnormal()
. gen uv=rnormal()
. gen u1=rnormal()
. gen u0=rnormal()
. forvalues i=1/12
  2. gen e`i´=rnormal()
```

```
  3.

. gen t1 =0.1 +0.1 *Q  +1.1*f1+e1
. gen t2 =0.5 +0.1 *Q  +1.4*f1+e2
. gen t3 =0.4 +0.3 *Q  +    f1+e3
. gen t4 =0.3 +0.11*Q +3  *f2 +e7
. gen t5 =0.4 +0.21*Q +1.6*f2 +e8
. gen t6 =0.1 +0.31*Q +    f2 +e9
. gen t7 =0.3 +0.11*Q +3.1*f1 +3  *f2 +e7
. gen t8 =0.4 +0.21*Q +1.2*f1 +1.6*f2 +e8
. gen t9 =0.1 +0.31*Q +2  *f1 +    f2 +e9

. gen D=(0.5*Z + f1 + uv>0)
. gen Y1 = 2   +2*X + 2*f1 + u1
. gen Y0 = 1.5 +    X +   f1 + u0
. gen  Y  = D*Y1   + (1-D)*Y0

. gen D2=(0.5*Z + f1 + f2 + uv>0)
. gen Y21 = 2   +2*X + 2*f1 + 2*f2 + u1
. gen Y20 = 1.5 +  X +   f1 +   f2 + u0
. gen  Y2 = D2*Y21 + (1-D2)*Y20
```

4.1.1 One Factor

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

```
heterofactor Y X, treatind(D1) instrum(Z) scores(t1 t2 t3) indvarsc(Q) difficult
initialreg
```

Resulting in

```
. heterofactor Y1 X, treat(D1) 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)
...
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. In the same way, `/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. Note that the reported standard deviations (i.e., `/sigf1` and `/sigf2`) and the mixture combining probability (i.e., `/p1`) are transformed. To retrieve the actual values we need to transform them back:

```
. di exp(_b[sigf1:_cons])
.90781715

. di exp(_b[sigf2:_cons])
.62749649

. di invlogit(_b[p1:_cons])
.24802025
```

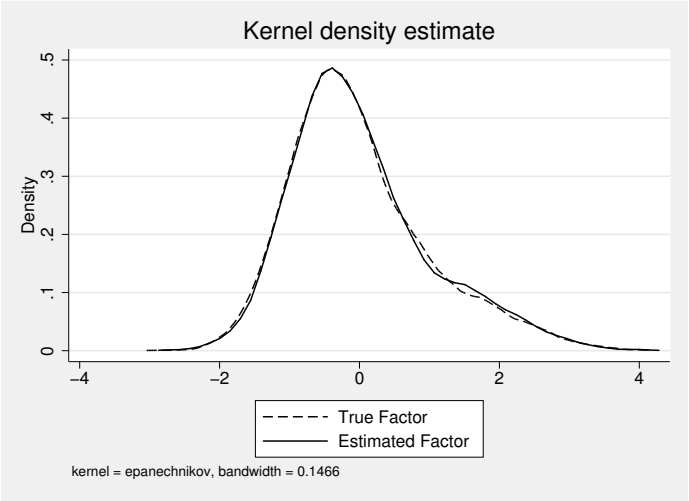
The command provides a random draw from the estimated factor distribution under the name `mixt`. That is, the program creates a variable for the user to plot the distribution that results from the estimated mixture of normals. Here, we use this variable to show the accuracy of the estimation by comparing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

```
kdensity mixt, addplot(kdensity f1) scheme(sj) ///
legend(order(2 1) lab(1 "Estimated Factor")  lab(2 "True Factor")) xtitle("")
```

4.1.2 Two Factors

Now we move on to the two-factor case assuming the loading structure presented in (13). Therefore, we will use measures T_1 to T_6. The output will be divided in three parts: one for the estimation of each factor's distribution, and one for the estimation of the outcomes and choice equations. We estimate the model using the following command:

Figure 1: Actual and Estimated Factor 1

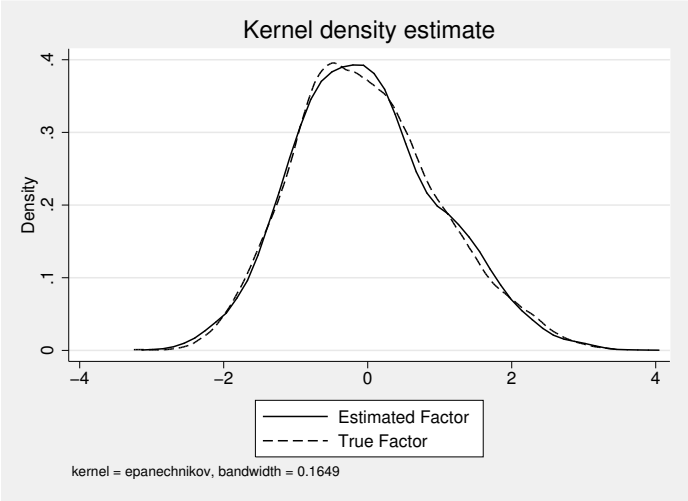



```
heterofactor Y2 X, treat(D2) instrum(Z) scores(t1 t2 t3 t4 t5 t6) indvarsc(Q)
       factors(2) initialreg difficult nohats
```

```
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)
...
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)
...
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)
...
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          |
```

Figure 2: Actual and Estimated Factor 2 Using Structure (13)




```
           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 the model to estimate has 2 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 $\alpha^{Y_1,B}$. In the same way, `/a01` indicates $\alpha^{Y_0,A}$ and `/a02` indicates $\alpha^{Y_0,B}$. In order to show the accuracy of our estimates of $F_{\theta^B}(\zeta)$, we plot it together with the real one in Figure 2.

```
kdensity mixt2, addplot(kdensity f2) scheme(sj) ///
legend(lab(1 "Estimated Factor")  lab(2 "True Factor")) xtitle("")
```

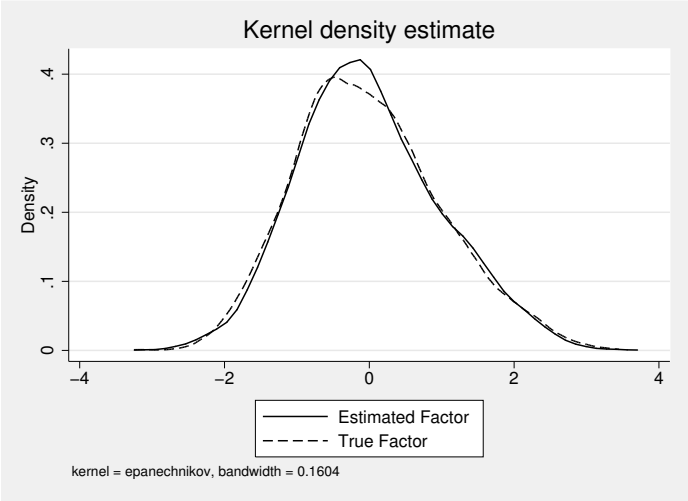
4.1.3 Two Factors - Triangular Loadings Structure

Now, we run a model that assumes that the measurement system (4) has a triangular loading 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 4.1.2. 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 difficult nohats
```

```
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)
...
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)
...
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
```

Figure 3: Actual and Estimated Factor 2 Using Triangular Structure (12)




```
------------------------------------------------------------------------------
             |      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(lab(1 "Estimated Factor")  lab(2 "True Factor")) xtitle("")
```

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

4.2 Example using the NLSY79

In this Section, we will present an example with real data. The data set used is the National Longitudinal Survey of Youth (NLSY79). It is dataset that is widely used by the research community (see for instance Heckman et al. (2006); Urzua (2008); Prada and Urzua (2013)). In our example, we will estimate a Roy model where the endogenous choice is whether or not the person went to college by age 25, and the potential outcomes are the log of earnings by age 30. The adjunct measurement system is comprised by the Armed Services Vocational Aptitude Battery (ASVAB) tests recorded during their 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.

```
heterofactor lnincome blackwhite ExperienceF Experience2, treat(HR_5)
instrum(HGC_MOTHER) scores(stASVAB_6 stASVAB_10 stASVAB_8) indvarsc(blackwhite
HGC_MOTHER)
```

```
. 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)
...
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 paper, we presented a Stata code that is able to fit numerous models whose common feature is that they are systems of equations liked by 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.

References

Aakvik, A., Heckman, J. J., and Vytlacil, E. (2000). Treatment Effects for Discrete Outcomes when Responses to Treatment Vary Among Observationally Identical Persons. page 53.

Cameron, S. V. and Heckman, J. J. (1998). Life Cycle Schooling and Dynamic Selection Bias: Models and Evidence for Five Cohorts of American Males. *Journal of Political Economy*, 106(2):262–73.

Cameron, S. V. and Heckman, J. J. (2001). The Dynamics of Educational Attainment for Black, Hispanic, and White Males. *Journal of Political Economy*, 109(3):455–499.

Carneiro, P., Hansen, K. T., and Heckman, J. (2003). 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(2):361–422.

Greene, W. H. (2000). *Econometric analysis*. Prentice Hall, Upper Saddle River, New Jersey, 4th edition.

Hansen, K. T., Heckman, J. J., and Mullen, K. J. (2004). The effect of schooling and ability on achievement test scores. *Journal of Econometrics*, 121(1-2):39–98.

Heckman, J., Stixrud, J., and Urzua, S. (2006). The Effects of Cognitive and Noncognitive Abilities on Labor Market Outcomes and Social Behavior. *Journal of Labor Economics*, 24(3):411–482.

Heckman, J. J., Humphries, J. E., Urzua, S., and Veramendi, G. (2011). The Effects of Educational Choices on Labor Market, Health, and Social Outcomes. *Human Capital and Economic Opportunity: A Global Working Group*, (2011-002):1–63.

Heckman, J. J. and Navarro, S. (2007). Dynamic discrete choice and dynamic treatment effects. *Journal of Econometrics*, 136(2):341–396.

Jöreskog, K. and Goldberger, A. (1972). Factor analysis by generalized least squares. *Psychometrika*, 37(3):243–260.

Judd, K. L. (1998). *Numerical Methods in Economics*. The MIT Press, Cambridge, Massachusetts.

Keane, M. P. and Wolpin, K. I. (1997). The Career Decisions of Young Men. *Journal of Political Economy*, 105(3):473–51.

Kotlarski, I. (1967). On characterizing the gamma and the normal distribution. *Pacific Journal of Mathematics*, 20(1):69–76.

Prada, M. F. and Urzua, S. (2013). One Size Does Not Fit All: The Role of Vocational Ability on College Attendance and Labor Market Outcomes.

Roy, A. D. (1951). Some thoughts on the distribution of earnings. *Oxford Economic Papers*, 3(2):135–146.

Sarzosa, M. (2015). The Dynamic Consequences of Bullying on Skill Accumulation.

Sarzosa, M. and Urzua, S. (2014). Bullying and Cyberbullying in Teenagers, The Role of Cognitive and Non-Cognitive Skills.

Urzua, S. (2008). Racial Labor Market Gaps. *Journal of Human Resources*, 43(4):919.