The Stata Journal (2008)
8, Number 2, pp. 170–189

The Stata command felsdvreg to fit a linear model with two high-dimensional fixed effects

Thomas Cornelissen
University of Hannover
Hannover, Germany
cornelissen@ewifo.uni-hannover.de

Abstract. This article proposes a memory-saving decomposition of the design matrix to facilitate the estimation of a linear model with two high-dimensional fixed effects. A common way to fit such a model is to take into account one of the effects by including dummy variables and to sweep out the other effect by the within transformation (fixed-effects transformation). If the number of panel units is high, creating and storing the dummy variables can involve prohibitively large computer-memory requirements. The memory-saving procedure to set up the moment matrices for estimation presented in this article can reduce the memory requirements considerably. The companion Stata ado-file `felsdvreg` implements the estimation method, takes care of identification issues, and provides useful summary statistics.

Keywords: st0143, felsdvreg, linked employer–employee data, fixed effects, three-way error-components model

1 Introduction

Fixed-effects models are popular in applied econometric work because they allow us to take into account time-constant unobserved heterogeneity that may be correlated with observed characteristics. In recent years, large-scale linked employer–employee data, linked student–teacher data, and other types of linked data have become available. Such data allow us to include at least two fixed effects into the analysis, for example, person and firm effects or student and teacher effects. Because the datasets involved usually include high numbers of observations, these fixed effects are often high dimensional, i.e., there is a high number of panel units (workers, firms, teachers, students). Applications of such models can be found in the fields of labor economics and educational economics. For example, Abowd, Kramarz, and Margolis (1999); Abowd, Creecy, and Kramarz (2002); and Andrews, Schank, and Upward (2006) estimate wage equations including person and firm effects, and Harris and Sass (2007) fit a model of student achievement including student, teacher, and school effects. However, the problem is not confined to matched datasets. For example, in a panel dataset with a high number of individuals and many geographical regions, the two high-dimensional fixed effects may consist of individual effects and region dummies.

Because of the size of the datasets involved, the researcher often encounters computer restrictions in terms of memory space and computing time. This article and the companion Stata command, `felsdvreg`, deal with the first restriction, the limitation of computer memory. I present a memory-saving way to fit a fixed-effects model with two high-dimensional fixed effects. It relies on the idea that a typical dummy-variable matrix of fixed effects, such as firm or teacher effects, is a sparse matrix. Sparse matrices can be stored efficiently in compressed form. The method is implemented in a ready-to-use Stata ado-file. This program solves the identification problem, computes the estimates, and provides useful summary statistics.

The article is organized as follows: Section 2 presents the model. Section 3 points out the computer restrictions and describes how the estimation can be organized in a memory-efficient way. Section 4 summarizes the steps of the estimation. Section 5 presents the implementation of the method in a Stata ado-file and comments on the output of the program. Section 6 concludes the article.

Throughout the article, I refer to linked employer–employee datasets, calling the two effects to be estimated person and firm effects. I refer to "stayers" as those individuals who are observed in only one firm and to "movers" as those who are observed in several firms. Despite the terminology used, the method can be directly transferred to other types of datasets.

2 A linear fixed-effects model with two high-dimensional fixed effects

Consider the following model, which can be applied to linked employer–employee panel data containing data about individuals and firms. The model is

$$\widetilde{\mathbf{y}} = \widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{D}}\boldsymbol{\theta} + \widetilde{\mathbf{F}}\boldsymbol{\psi} + \widetilde{\boldsymbol{\epsilon}} \tag{1}$$

where $\widetilde{\mathbf{X}}$ $(N^* \times K)$ is the design matrix of time-varying characteristics; $\widetilde{\mathbf{D}}$ $(N^* \times N)$ is the design matrix for the person effects; and $\widetilde{\mathbf{F}}$ $(N^* \times J)$ is the design matrix for the firm effects. N^* is the number of person-years in the dataset, J is the number of firms, N is the number of persons, and K is the number of time-varying regressors. The $\widetilde{\ }$ reflects that (1) is the untransformed model.

Further effects, such as fixed time effects, are subsumed in $\widetilde{\mathbf{X}}$ together with the other time-varying regressors. In a student–teacher context, further effects subsumed in $\widetilde{\mathbf{X}}$ might be school effects. But any fixed effects remaining in $\widetilde{\mathbf{X}}$ should not be high dimensional (relative to the computer memory available), because only the effects in $\widetilde{\mathbf{D}}$ and $\widetilde{\mathbf{F}}$ are, in the following, treated as high dimensional.

A common way to fit such a model is to include one of the effects (here the firm effect) as dummy variables and to sweep out the other effect (here the person effect) by the within transformation or fixed-effects transformation. This transformation consists of subtracting the group mean (here the person mean) for all observations. The $\widetilde{\mathbf{D}}$

matrix becomes the null matrix, and the person effects are eliminated from the model. Write the transformed model as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{F}\boldsymbol{\psi} + \boldsymbol{\epsilon} \tag{2}$$

where $\boldsymbol{\epsilon}$ is an error term satisfying the assumptions of the classical linear regression model. Abowd, Kramarz, and Margolis (1999) note that this procedure is algebraically equivalent to the full dummy-variable model. Andrews, Schank, and Upward (2006) call this procedure the "FEiLSDVj" method in order to emphasize that the model combines the classical fixed-effects (FE) model and the least-squares dummy-variable model (LSDV), because one effect is eliminated by the fixed-effects transformation and the other is included as dummy variables. This procedure is adequate for balanced and unbalanced panels alike[1] (Greene 2003, 293).

The system of normal equations is

$$\mathbf{A}\begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\psi} \end{pmatrix} = \mathbf{B}$$

with

$$\mathbf{A} = \begin{pmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{F} \\ \mathbf{F}'\mathbf{X} & \mathbf{F}'\mathbf{F} \end{pmatrix} \tag{3}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{F}'\mathbf{y} \end{pmatrix} \tag{4}$$

Solving this system of equations delivers the coefficient estimates.

3 Creating the moment matrices in a memory-efficient way

In big datasets, the design matrix $(\mathbf{X}, \mathbf{F})$ can be too large to fit into memory, because most software packages such as Stata require the design matrix be stored in memory. To illustrate the memory requirement with the explicit creation of all dummy variables, consider the following example: A dataset contains $N^* = 2,000,000$ observations, $N = 100,000$ persons, $J = 20,000$ firms, and $K = 50$ further right-hand-side regressors. Assume that one cell of the data matrix consumes 8 bytes (which is the case when working in high-precision mode). The creation of the time-demeaned firm dummies implies storing the design matrix $(\mathbf{X}, \mathbf{F})$ in the computer memory. The size of this matrix is $N^* \cdot (K + J) \cdot 8$ bytes = 320.8 gigabytes. This is far more than the computer memory available at present to most researchers. It therefore seems that the estimation of person and firm effects by using the "FEiLSDVj" method with several millions of observations and several thousands of firms would be impossible because of restricted memory resources.

1. For a more general treatment of the matrix algebra involved in representing multiple-way error-components models with unbalanced data structures, see Davis (2001).

However, while the design matrix $(\mathbf{X}, \mathbf{F})$ is of dimension $N^* \times (K + J)$, the cross-product matrices $\mathbf{A}$ and $\mathbf{B}$, given in (3) and (4), are of dimension $(K + J) \times (K + J)$ and $(K + J) \times 1$ only. They require much less storage space. In the above example, the memory requirement for $\mathbf{A} = (\mathbf{X}, \mathbf{F})'(\mathbf{X}, \mathbf{F})$ is only $(K + J)^2 \cdot 8$ bytes $= 3.21$ gigabytes, which is considerably smaller.[2]

In fact, $\mathbf{A}$ and $\mathbf{B}$ can be computed without explicitly creating the full design matrix $(\mathbf{X}, \mathbf{F})$. A solution for that problem lies in the fact that each element of $\mathbf{A}$ and $\mathbf{B}$ is a cross-product sum of no more than two regressors. This implies that for computing one element of $\mathbf{A}$ or $\mathbf{B}$, only two regressors need to be stored in memory. While the $\mathbf{X}$ part of the design matrix is provided as a dataset, the $\mathbf{F}$ part of the cross-product matrix can be created during the estimation process without actually generating the $\mathbf{F}$ part of the design matrix, i.e., the dummy variables. The information needed for that purpose is condensed in the group identifiers. In other words, the group identifiers provide a compressed storage format of the sparse dummy-variable matrices. The following decomposition is based on the fact that the $\mathbf{F}$ matrix is a sparse matrix, i.e., large parts of it are null submatrices, which deliver no contribution to $\mathbf{A}$ or $\mathbf{B}$. Therefore, in the process of the formation of $\mathbf{A}$ and $\mathbf{B}$, only certain parts of the $\mathbf{F}$ matrix need to be created, and time and memory can be saved.

Let the persons in the dataset be indexed by i $(i = 1, \ldots, N)$ and the time periods for each individual be indexed by t $(t = 1, \ldots, T_i)$. T_i is the number of time periods that individual i is observed. The total number of observations is then $N^* = \sum_i T_i$. The vector $\mathbf{y}$ and the design matrices $\mathbf{X}$ and $\mathbf{F}$ in (2) have row dimension N^*, and rows are indexed by the index it. The columns of $\mathbf{X}$ are indexed k $(k = 1, \ldots, K)$, and the columns of $\mathbf{F}$ are indexed j $(j = 1, \ldots, J)$.

The memory-saving way to create $\mathbf{A}$ and $\mathbf{B}$ starts from the idea that these matrices can be decomposed by observations or subsets of observations.[3] For example, $\mathbf{A}$ $(\mathbf{B})$ can be represented as a sum of matrices $\mathbf{A}_i$ $(\mathbf{B}_i)$ for each individual:

$$\begin{aligned} \mathbf{A} &= \sum_i \mathbf{A}_i = \sum_i \begin{pmatrix} \mathbf{X}_i'\mathbf{X}_i & \mathbf{X}_i'\mathbf{F}_i \\ \mathbf{F}_i'\mathbf{X}_i & \mathbf{F}_i'\mathbf{F}_i \end{pmatrix} \\ \mathbf{B} &= \sum_i \mathbf{B}_i = \sum_i \begin{pmatrix} \mathbf{X}_i'\mathbf{y}_i \\ \mathbf{F}_i'\mathbf{y}_i \end{pmatrix} \end{aligned}$$

where $\mathbf{X}_i$ is $(T_i \times K)$, $\mathbf{F}_i$ is $(T_i \times J)$, and $\mathbf{y}_i$ is $(T_i \times 1)$. The matrices involve only those observations that are associated with individual i. For the current purpose, it makes sense to do the individual-wise decomposition only for those parts of the matrices where the $\mathbf{F}$ matrix is involved, that is,

2. The cross-product matrix $\mathbf{B} = (\mathbf{X}, \mathbf{F})'\mathbf{y}$ is negligibly small compared to the matrix $\mathbf{A}$.
3. The possibility of an observation-wise computation of a cross-product matrix is usually presented in econometrics textbooks by two alternative ways of writing the ordinary least-squares estimator. For example, in Wooldridge (2002, 53) it is stated that $\mathbf{X}'\mathbf{X} = \sum_i \mathbf{x}_i'\mathbf{x}_i$, where $\mathbf{x}_i$ are the rows of $\mathbf{X}$.

$$\mathbf{A} = \begin{pmatrix} \mathbf{X}'\mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_i \begin{pmatrix} \mathbf{0} & \mathbf{X}_i'\mathbf{F}_i \\ \mathbf{F}_i'\mathbf{X}_i & \mathbf{F}_i'\mathbf{F}_i \end{pmatrix} \tag{5}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{0} \end{pmatrix} + \sum_i \begin{pmatrix} \mathbf{0} \\ \mathbf{F}_i'\mathbf{y}_i \end{pmatrix} \tag{6}$$

The decomposition continues with the idea that the $\mathbf{F}$ matrix has a different structure for stayers and for movers. In this context, movers are defined as workers who change employers at least once during the whole observation period, and stayers are those workers who never change employers.

Recall that the model is a transformed model. Group means by person have been subtracted ("time-demeaning"/"within-transformation"). Because stayers never change firms, the time-demeaned firm dummies are all zero. The $\mathbf{F}$ matrix for stayers is the null matrix.

Therefore, we get

$$\mathbf{A} = \begin{pmatrix} \mathbf{X}'\mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \sum_{i\in \text{ Movers}} \begin{pmatrix} \mathbf{0} & \mathbf{X}_i'\mathbf{F}_i \\ \mathbf{F}_i'\mathbf{X}_i & \mathbf{F}_i'\mathbf{F}_i \end{pmatrix} \tag{7}$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{0} \end{pmatrix} + \sum_{i\in \text{ Movers}} \begin{pmatrix} \mathbf{0} \\ \mathbf{F}_i'\mathbf{y}_i \end{pmatrix} \tag{8}$$

Equations (7) and (8) are important simplifications of (5) and (6). Because the $\mathbf{F}$ matrix is the null matrix in the subsample of stayers, the cross-product submatrices $\mathbf{X}'\mathbf{F}$, $\mathbf{F}'\mathbf{F}$, and $\mathbf{F}'\mathbf{y}$ need to be computed only for movers.[4] As these matrices can be computed individual by individual, the $\mathbf{F}$ matrix does not need to exist completely at any point of time. For example, it suffices to create the matrix $\mathbf{F}_i$ for one individual and to compute $\mathbf{X}_i'\mathbf{F}_i$, $\mathbf{F}_i'\mathbf{F}_i$, and $\mathbf{F}_i'\mathbf{y}_i$. $\mathbf{F}_i$ is of dimension $(T_i \times J)$ so it should fit into memory. However, by analyzing the structure of $\mathbf{F}_i$ more precisely, the matrix can be reduced further, and more memory space can be saved.

Look at $\mathbf{F}_{i^*}$ for a worker i^* who is observed at $T_{i^*} = 3$ different points in time and changes firms once. The non–time-demeaned matrix $\widetilde{\mathbf{F}}_{i^*}$ is

$$\widetilde{\mathbf{F}}_{i^*} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{pmatrix}$$

Worker i^* is employed during two time periods in firm 1 and during a third time period in firm 4. He is never employed in any other firm, which means that to the right the individual $\mathbf{F}$ matrix is filled up with zeros.

4. The matrix $\mathbf{F}'\mathbf{X}$ is the transpose of $\mathbf{X}'\mathbf{F}$, and so in what follows it is not discussed separately.

The corresponding time-demeaned design matrix of the firm effects for individual i^* is

$$\mathbf{F}_{i^*} = \begin{pmatrix} 1/3 & 0 & 0 & -1/3 & 0 & \dots & 0 \\ 1/3 & 0 & 0 & -1/3 & 0 & \dots & 0 \\ -2/3 & 0 & 0 & 2/3 & 0 & \dots & 0 \end{pmatrix}$$

For each worker, only very few columns of $\mathbf{F}_{i^*}$ will be different from null vectors, because a given worker is employed in very few firms relative to the total set of firms. Consequently, many elements of the cross-product matrices $\mathbf{X}_i'\mathbf{F}_i$, $\mathbf{F}_i'\mathbf{F}_i$, and $\mathbf{F}_i'\mathbf{y}_i$ are equal to zero.

In the appendix (section 7), $(\mathbf{X}_{i^*}'\mathbf{F}_{i^*})$, $(\mathbf{F}_{i^*}'\mathbf{F}_{i^*})$, and $(\mathbf{F}_{i^*}'\mathbf{y}_{i^*})$ are computed for the above example. In $(\mathbf{F}_{i^*}'\mathbf{F}_{i^*})$, the only nonzero elements are those where both row and column indices refer to a firm where worker i was employed at some moment of time. In $(\mathbf{X}_{i^*}'\mathbf{F}_{i^*})$, only the columns that are indexed with reference to a firm where worker i was employed are nonzero. In $(\mathbf{F}_{i^*}'\mathbf{y}_{i^*})$, only the rows that are indexed with reference to a firm where worker i was employed are nonzero.

A typical worker is usually employed in very few firms and thus contributes to only a very few elements of the cross-product matrices. $\mathbf{F}_i$ is a sparse matrix, and so are $(\mathbf{X}_{i^*}'\mathbf{F}_{i^*})$, $(\mathbf{F}_{i^*}'\mathbf{F}_{i^*})$, and $(\mathbf{F}_{i^*}'\mathbf{y}_{i^*})$. One can write $\mathbf{F}_i$ more compactly by leaving out the zero columns. Call this reduced matrix $\mathbf{F}_i^S$. This is a $T_i \times s$ matrix, where s is the number of firms in which individual i was employed. In the above example, $\mathbf{F}_i^S$ would be a (3×2) matrix that reads

$$\mathbf{F}_{i^*}^S = \begin{pmatrix} 1/3 & -1/3 \\ 1/3 & -1/3 \\ -2/3 & 2/3 \end{pmatrix}$$

Instead of computing $(\mathbf{X}_i'\mathbf{F}_i)$, $(\mathbf{F}_i'\mathbf{F}_i)$, and $(\mathbf{F}_i'\mathbf{y}_i)$, one can compute $(\mathbf{X}_i'\mathbf{F}_i^S)$, $(\mathbf{F}_i^{S\prime}\mathbf{F}_i^S)$, and $(\mathbf{F}_i^{S\prime}\mathbf{y}_i)$, which saves memory and time. However, one needs the information to which firm the columns of $\mathbf{F}_i^S$ refer, because once the cross products are computed, the results need to be added to the correct elements of the $\mathbf{A}$ and the $\mathbf{B}$ matrix, which is not a problem because this information is stored in the group identifiers. The next section summarizes the algorithm for the fixed-effects estimation of the linear three-way error-components model in a memory-saving way.

4 The algorithm to compute the least-squares solution

The memory-saving way to compute the matrices $\mathbf{A}$ and $\mathbf{B}$ of the normal equations uses the information in which firm a given worker is employed. This allows us to compute only those elements of $\mathbf{A}$ and $\mathbf{B}$ to which the worker contributes. The zero elements of the sparse matrices involved are dropped from the computations.

Use the following steps:

1. Create null matrices $\mathbf{A}$ of dimension $(K+J)\times(K+J)$ and $\mathbf{B}$ of dimension $(K+J)\times 1$.

2. Compute $\mathbf{X}'\mathbf{X}$ and $\mathbf{X}'\mathbf{y}$ on the combined sample of movers and stayers. Fill in these cross products at the appropriate submatrices of $\mathbf{A}$ and $\mathbf{B}$ as shown in (5) and (6).

3. For each mover $i(i \in$ Mover), create the time-demeaned matrix $\mathbf{F}_i$ but omit columns that are zero; call this reduced matrix $\mathbf{F}_i^S$. This is a $T_i \times s$ matrix, where s is the number of firms in which individual i was employed. Now,

   a. form $\mathbf{F}_i^{S\prime}\mathbf{F}_i^S$ and update the $\mathbf{A}$ matrix by adding the resulting cross products to the appropriate elements of $\mathbf{A}$,

   b. form $\mathbf{X}_i'\mathbf{F}_i^S$, as well as its transpose, $(\mathbf{X}_i'\mathbf{F}_i^S)' = \mathbf{F}_i^{\mathbf{S}\prime}\mathbf{X}_i$, and update the $\mathbf{A}$ matrix by adding the resulting cross products to the appropriate elements of $\mathbf{A}$, and

   c. form $\mathbf{F}_i^{S\prime}\mathbf{y}$ and update the $\mathbf{B}$ matrix by adding the resulting cross products to the appropriate elements of $\mathbf{B}$.

4. Once $\mathbf{A}$ and $\mathbf{B}$ are completed, solve for the coefficient vector $(\boldsymbol{\beta}, \boldsymbol{\psi})$.

5 Implementation in Stata

The method is implemented in Stata in the ado-file `felsdvreg`, the core of which is programmed in Mata. Using Mata in the context of large datasets is an advantage. First, provided that there is enough computer memory, Mata can handle matrices of a dimension of up to 2 billion × 2 billion compared with only 11,000 × 11,000 in the Stata environment (Stata/SE). Second, Mata provides computer routines with high numerical precision, which is more important in large datasets than in small datasets.[5]

Other ways to handle the estimation problem are the approximate procedures, as well as two-step and iterative solutions to the exact problem, presented in Abowd, Kramarz, and Margolis (1999), Abowd, Creecy, and Kramarz (2002), Andrews, Schank, and Upward (2006), and Grütter (2006).[6] If the sole aim is to control for unobserved heterogeneity and not to compute the person and firm effects explicitly, the "spell fixed-

5. In addition to using high-precision routines, cross-checking the results obtained with those obtained in similar but smaller datasets is another way to test whether the size of the dataset poses problems of numerical precision. Helpful comments about that topic can be found under the thread "dataset larger than RAM" on the Statalist discussion board, at http://www.stata.com/statalist/archive/.
6. The classical minimum-distance estimator proposed by Andrews, Schank, and Upward (2006) delivers the same coefficient estimates as the "FEiLSDVj" method, but it delivers different standard errors because it is based on separate estimations for movers and stayers, and the error-term variance of both estimations is not constrained to be equal.

effects" method proposed in Andrews, Schank, and Upward (2006) is a good alternative to the "FEiLSDVj" method.[7]

In the following, a small simulated linked employer–employee dataset is used to illustrate the Stata implementation of the estimation method presented in the previous sections. The dataset used for the illustration has 100 observations. It comprises 20 workers, for which the dummy variables `p1`, ..., `p20` have been created, and 15 firms, for which the dummy variables `f1`, ..., `f15` have been created. The dependent variable is called `y`, and the two independent time-varying regressors are called `x1` and `x2`. The pattern of worker mobility between firms is important because it determines whether person and firm effects can be identified. The firms with movers can be divided into groups (shown in table 1) within which there is worker mobility, but between which there is no mobility.[8]

Table 1. Group that the firms with movers belong to

Group	Firms
1	3, 4, 5
2	6, 7, 8, 9
3	10, 11, 12
4	13, 14, 15

Within each group, one effect is not identified and serves as the reference; for example, if the firm with the smallest firm ID is chosen as the reference firm in each group, the effects of firms 3, 6, 10, and 13 are not identified. The effects of firms without movers (firms 1 and 2) are not identified because they can be thought of as forming single groups with only one firm per group.

A common way to fit a model with person and firm fixed effects is to include the firm effects as dummies and to eliminate the person effects by the within transformation (the "FEiLSDVj" method). Knowing that the effects of firms 1, 2, 3, 6, 10, and 13 are not identified in the given example, this can be implemented as follows, where `i` is a variable containing the person identifier:

7. An alternative Stata command to compute a model with two high-dimensional fixed effects is `a2reg` by Amine Ouazad, based on Abowd, Creecy, and Kramarz (2002). Ouazad's two-way fixed-effects regressions are available at http://vrdc.ciser.cornell.edu/guides/cg2/html/index.html or by typing `net from http://repository.ciser.cornell.edu/viewcvs-public/cg2/branches/stata/` in the Stata command line. This command solves for the coefficient estimates by using a solver algorithm suitable for sparse matrices. More generally, all mathematics or statistics packages that include sparse-matrix functions could be used as alternatives to the method described here.
8. An algorithm to determine the groups is derived in Abowd, Creecy, and Kramarz (2002).

```
. use felsdvsimul
. tabulate i, generate(p)
  (output omitted)
. tabulate j, generate(f)
  (output omitted)
. xtset i
       panel variable:  i (unbalanced)
. xtreg y x1 x2 f4-f5 f7-f9 f11-f12 f14-f15, fe
Fixed-effects (within) regression               Number of obs      =       100
Group variable: i                               Number of groups   =        20
R-sq:  within  = 0.6518                         Obs per group: min =         1
       between = 0.0015                                        avg =       5.0
       overall = 0.0913                                        max =         9
                                                F(11,69)           =     11.74
corr(u_i, Xb)  = -0.5330                        Prob > F           =    0.0000
```

y	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
x1	1.029258	.2151235	4.78	0.000	.6000987	1.458418
x2	-.709482	.2094198	-3.39	0.001	-1.127263	-.2917009
f4	13.2617	3.258081	4.07	0.000	6.762004	19.76139
f5	13.95499	2.818964	4.95	0.000	8.331314	19.57867
f7	8.559977	3.882525	2.20	0.031	.8145504	16.3054
f8	5.433107	3.908214	1.39	0.169	-2.363566	13.22978
f9	11.44951	4.792492	2.39	0.020	1.888749	21.01027
f11	16.76837	3.245567	5.17	0.000	10.29364	23.2431
f12	10.01551	3.407205	2.94	0.004	3.218319	16.8127
f14	-10.19694	3.074528	-3.32	0.001	-16.33046	-4.063427
f15	2.526721	3.844219	0.66	0.513	-5.142287	10.19573
_cons	-6.044057	1.03021	-5.87	0.000	-8.09927	-3.988844
sigma_u	10.169633					
sigma_e	5.4861156					
rho	.77458273	(fraction of variance due to u_i)				

```
F test that all u_i=0:     F(19, 69) =     8.64              Prob > F = 0.0000
```

The estimated firm effects appear in the regression output. The estimates of the person effects can be displayed by

```
. predict peffxt, u
. table i, contents(m peffxt)
```

i	mean(peffxt)
1	-9.345165
2	-3.751444
3	12.98728
4	-4.665943
5	-3.879235
6	1.137969
7	-.4461367
8	.4524156
9	-16.23423
10	-12.18615
11	-.4041495
12	-3.953967
13	-11.94854
14	-4.272363
15	1.732473
16	-11.58673
17	-13.57038
18	21.66491
19	14.42718
20	11.0613

As described in the previous section, the explicit creation of all firm dummies, combined with the use of `xtreg` (let alone the creation of all person and firm dummies with the use of `regress`), can require more computer memory than is available. In the case where there is a large number of firms, it can therefore be necessary to apply a memory-saving way to the solution of the "FEiLSDVj" estimator. I have programmed the algorithm presented in the preceding section as a Stata ado-file called `felsdvreg`. This routine can be applied to the present dataset as follows:

```
. felsdvreg y x1 x2, ivar(i) jvar(j) feff(feffhat) peff(peffhat) xb(xb) res(res)
> mover(mover) group(group) mnum(mnum) pobs(pobs)
```

The options are the following: The `ivar()` option is used to pass the variable name of the person ID, and the `jvar()` option does the same for the firm ID. The `feff()` and `peff()` options define the names of new variables to be created to store the firm and person effects after estimation. The `xb()` and `res()` options store the linear combinations $x'\widehat{\beta}$ and the residual $\widehat{\epsilon}$. The remaining options define the names of the new variables that store a dummy variable indicating a person who is a mover, `mover()`; a group variable indicating the groups of firms connected through mobility, `group()`; a variable containing the number of movers per firm, `mnum()`; and a variable indicating the number of observations per persons, `pobs()`. The output is

```
. felsdvreg y x1 x2, ivar(i) jvar(j) feff(feffhat) peff(peffhat) xb(xb) res(res)
> mover(mover) group(group) mnum(mnum) pobs(pobs)
Memory requirement for moment matrices in GB:
  2.17600e-06

Computing generalized inverse, dimension: 11
  Start: 6 Mar 2008 18:06:02
  End:   6 Mar 2008 18:06:02
N=100
```

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
x1	1.029258	.2151235	4.78	0.000	.6000987	1.458418
x2	-.7094819	.2094198	-3.39	0.001	-1.127263	-.2917009

```
F-test that person and firm effects are equal to zero: F(28,69)=9.81 Prob > F = 0
F-test that person effects are equal to zero:          F(19,69)=8.64 Prob > F = 0
F-test that firm effects are equal to zero:            F(9,69)=9.97 Prob > F = 0
```

In big datasets, the crucial steps of the estimation concerns the question of whether the moment matrices fit into memory, and how much computing time is required when solving for the coefficients (computing the inverse). The above default output contains information on these points. The firm and person effects can be displayed as follows:

```
. table j, contents(m feffhat)
```

j	mean(feffhat)
1	0
2	0
3	0
4	13.2617
5	13.95499
6	0
7	8.559977
8	5.433106
9	11.44951
10	0
11	16.76837
12	10.01551
13	0
14	-10.19694
15	2.526721

```
. table i, contents(m peffhat)
```

i	mean(peffhat)
1	-15.38922
2	-9.795502
3	6.943222
4	-10.71
5	-9.923292
6	-4.906089
7	-6.490194
8	-5.591642
9	-22.27829
10	-18.23021
11	-6.448206
12	-9.998024
13	-17.99259
14	-10.31642
15	-4.311584
16	-17.63079
17	-19.61444
18	15.62085
19	8.383124
20	5.017239

The firm effects of the firms without movers and the reference firm in each group are set to zero. The firm effects are exactly the same as in the `xtreg` estimation (p. 178). The person effects differ from the effects of the `xtreg` regression only by the constant −6.044057 of the `xtreg` model, because `felsdvreg` does not, by default, normalize the sum of the person effects to zero.[9]

Using the option `noisily` allows us to generate the following additional output:

```
. felsdvreg y x1 x2, ivar(i) jvar(j) feff(feffhat) peff(peffhat) xb(xb) res(res)
> mover(mover) group(group) mnum(mnum) pobs(pobs) noisily
Unique worker-firm combinations: 41
Number of firms workers are employed in:
```

Number of firms	Freq.	Percent	Cum.
1	7	35.00	35.00
2	7	35.00	70.00
3	4	20.00	90.00
4	2	10.00	100.00
Total	20	100.00	

9. If the option `cons` is chosen in `felsdvreg`, it does normalize the sum of the person effects to zero and displays a regression constant.

```
Number of movers (0=Stayer, 1=Mover):
      Mover |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |          7       35.00       35.00
          1 |         13       65.00      100.00
------------+-----------------------------------
      Total |         20      100.00

Number of observations per person:
  Obs. per |
     person |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |          3       15.00       15.00
          2 |          3       15.00       30.00
          4 |          3       15.00       45.00
          5 |          1        5.00       50.00
          6 |          4       20.00       70.00
          7 |          1        5.00       75.00
          8 |          2       10.00       85.00
          9 |          3       15.00      100.00
------------+-----------------------------------
      Total |         20      100.00

Number of movers per firm:
 Movers per |
       firm |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |          2       13.33       13.33
      1-  5 |          7       46.67       60.00
      6- 10 |          5       33.33       93.33
     11- 20 |          1        6.67      100.00
------------+-----------------------------------
      Total |         15      100.00
```

This output provides additional tables, which are interesting. For example, in the context of matched employer–employee or student–teacher data, the tables give details on the mobility pattern in the dataset. The first table summarizes in how many firms the workers are employed. The 7 workers employed in only one firm are stayers. Out of the remaining 13 workers, 7 are observed in two firms; 4, in three firms; and 2, in four firms. The second table is a summary of the first and gives the total number of stayers and movers. The third table indicates the number of observations per person. For example, 3 workers are observed at only one point in time, 3 workers are observed nine times, etc. The fourth table shows the distribution of the number of movers per firm. The purpose of this table is to get an impression of the quality of the estimation of the firm effects. The estimation of the firm effects is better the more movers there are, and one might think of the firm effects that are identified by few movers as effects that are poorly estimated. In this example dataset, two firms have no movers, and all 15 firms have less than 20 movers.

The 15 firms can be divided into groups within which there is worker mobility, but between which there is no mobility. Within each group, one firm effect is not identified, i.e., one firm effect has to be taken as the reference, and all other firm effects are expressed as differences from the reference. The `felsdvreg` program goes on by defining these groups:[10]

```
Groups of firms connected by worker mobility:
```

	Person-years	Persons	Movers	Firms
group	N(__000000)	N(__000009)	sum(__00000D)	N(__000008)
0	10	5	0	2
1	26	5	3	3
2	15	2	2	4
3	24	5	5	3
4	25	3	3	3
Total	100	20	13	15

```
Note: Group 0 in the table regroups firms without movers.

No firm effect in group 0 is identified.
15-2-4 = 9 firm effects are identified.
(number of firms - number of firms without movers - number of groups excl. group 0)
```

The two firms without movers are gathered in group 0. The remaining firms of the sample are divided into 4 groups. The table shows the number of person-years, persons, movers, and firms in each of the groups. As indicated, only 9 of the 15 firm effects are identified: 2 firms have no movers and their firm effects cannot be identified, and 4 more firm effects are not identified because they serve as reference in their groups.[11]

The option `noisily` also generates the following output:

```
If the covariances are positive, the following may indicate the importance in
> explaining the variance of y:
Cov(y, xb) / Var(y):                          .10029458
Cov(y, peffhat) / Var(y):                     .56511312
Cov(y, feffhat) / Var(y):                     .15341486
Cov(y, res) / Var(y):                         .18117743
```

This variance decomposition gives an indication of how strongly the four components (i) observed time-varying characteristics, (ii) person effects, (iii) firm effects, and (iv) the residual contribute to explaining the variance of the dependent variable. The shares sum to 1; however, the covariances indicated can become negative, and then it becomes difficult to interpret the numbers as shares.

10. The grouping algorithm incorporated in `felsdvreg` draws heavily on `a2group`, which is a Stata command by Amine Ouazad of the original FORTRAN code written by Robert Creecy and Lars Vilhuber (see footnote 7 for a link to access Ouazad's two-way fixed-effects regression package).
11. One of the simplest configurations of that table would be that there is only one group, because all panel units of the second effect are connected by mobility of the units of the first effect. For example, if the second effect cover's geographical regions of a country, which might all be connected by worker mobility, then they would all belong to a single mobility group.

After the estimation, the researcher may be interested in correlating the person and firm effects with each other or with other regressors. However, what is actually identified are relative person and firm effects within each group, and person and firm effects of different groups can be compared only if one is willing to make certain assumptions. This can be illustrated by computing the correlation of person and firm effects over all groups with different normalizations. The first command correlates the person and firm effects over all groups, while the second command correlates only the effects of group 1:

```
. correlate feffhat peffhat
(obs=100)
             | feffhat  peffhat
-------------+------------------
    feffhat  |  1.0000
    peffhat  | -0.5645   1.0000

. correlate feffhat peffhat if group==1
(obs=26)
             | feffhat  peffhat
-------------+------------------
    feffhat  |  1.0000
    peffhat  | -0.2006   1.0000
```

Now the firm and person effects are normalized so that they sum to zero within each group by subtracting the average group firm effect and the average group person effect. A new variable, `gmean`, captures the sum of the mean firm and the mean person effect of each group. After this normalization, the person and firm effects are deviations from the group means. After this, the correlation over all groups and the correlation using only the effects of group 1 are again computed:

```
. sort group
. by group: egen pmean=mean(peffhat)
. by group: egen fmean=mean(feffhat)
. generate peffnorm=peffhat-pmean
. generate feffnorm=feffhat-fmean
. generate gmean=pmean+fmean
. table group, contents(m gmean)
```

group	mean(gmean)
0	-7.385707
1	-1.236314
2	-2.610044
3	-7.291342
4	4.825048

```
. correlate feffnorm peffnorm
(obs=100)
             | feffnorm peffnorm
-------------+------------------
    feffnorm |   1.0000
    peffnorm |   0.0227   1.0000

. correlate feffnorm peffnorm if group==1
(obs=26)
             | feffnorm peffnorm
-------------+------------------
    feffnorm |   1.0000
    peffnorm |  -0.2006   1.0000
```

The normalization has changed the result from the correlation over all groups.[12] It is now 0.0227 whereas before it was −0.5645. The result of the correlation within the group of −0.2006 is unchanged. One could argue that the normalization of person and firm effects to an equal group mean makes comparison across groups more appropriate, and, therefore, the correlation over all groups after normalization is appropriate, whereas the one before normalization was not. However, it seems difficult to argue that a deviation of +1 from a group mean of −7.29 of group 3 means the same as a deviation of +1 from the group mean of 4.83 in group 4. The normalization does not change the fact that relative firm effects within groups are identified but relative firm effects between groups are not identified. It is therefore preferable to correlate only effects of the same group.

Andrews et al. (2008) show that the correlation between worker and firm effects is biased and that the bias is greater the lower the observed worker mobility between firms. After estimation, one may therefore want to select firm and person effects that fulfill certain minimum requirements with respect to the minimum number of movers per firm or the minimum number of observations per person. This is possible with the variables defined in the `mnum()` and `pobs()` options and returned by `felsdvreg`.

Even though the algorithm described above is memory saving, some applications in large datasets will still reach the limit of available memory. It is therefore important to observe the following remarks: The memory-intensive part of the program runs in Mata. Mata can use only memory which is not allocated to Stata by the `set memory` command. The user should therefore not allocate too much memory to Stata. The error message "unable to allocate real" indicates that Mata is running out of memory; in this case, memory allocated to Stata by `set memory` should be reduced. The error message "no room to add more observations/variables" indicates that Stata is running out of memory; in this case, the memory allocated to Stata by `set memory` should be increased. If there is not enough memory available to run `felsdvreg` on the complete sample, it might be worthwhile to run it on a subsample. To maximize the number of

12. This normalization is not exactly implemented in `felsdvreg`, but the program has two options for normalization: The option `normalize` normalizes the firm effects to mean zero within each group and adds to the person effects the mean firm effects that are subtracted in each group. The option `cons` normalizes the person effects to sum to zero over all observations and displays the overall mean person effect as the regression constant. Both options can be combined.

identified firm effects in a subsample, one could choose subsamples so that the mobility groups remain intact. For example, one might choose a large mobility group as a subsample and remove the remaining groups. In this case, `felsdvreg` should first be run with the option `grouponly`. This runs only the grouping algorithm, creating the group variable. This variable can be used to choose a subsample of the original sample on which `felsdvreg` can then be run.

The program `felsdvreg` can also be used for instrumental-variable (IV) estimation to cope with endogenous regressors. To produce IV estimates, the two stages of the two-stage least-squares (2SLS) estimation have to be carried out manually. In the second-stage estimation, `felsdvreg` needs to be told the names of the regressors, which have been predicted from a first stage, as well as which original regressors belong to the predicted regressors. For example, say that in a regression of `y` on `z1`, `x2`, `x3`, and two-way fixed effects, the variables `x2` and `x3` are to be instrumented by the IVs `z2` and `z3`. A 2SLS estimation can be carried out in the following way:

1. Run a first-stage regression for `x2`, and generate its prediction `x2hat`:

```
. felsdvreg x2 z1 z2 z3, ivar(i) jvar(j) xb(xb) feff(fhat) peff(phat) ...
. generate x2hat=xb+fhat+phat
```

2. Run a first-stage regression for `x3`, and generate its prediction `x3hat`:

```
. felsdvreg x3 z1 z2 z3, ivar(i) jvar(j) xb(xb) feff(fhat) peff(phat) ...
. generate x3hat=xb+fhat+phat
```

3. Run the second-stage regression:

```
. felsdvreg y z1 x2hat x3hat, ivar(i) jvar(j) xb(xb) feff(fhat) peff(phat)
> hat(x2hat x3hat) orig(x2 x3) ...
```

In the second-stage regression, `hat(x2hat x3hat)` and `orig(x2 x3)` tell `felsdvreg` that `x2hat` and `x3hat` are first-stage predictions of `x2` and `x3`. This allows `felsdvreg` to adjust the residual sum of squares and the standard errors of the second-stage regression (see, for example, Greene [2003, 400]).

The program `felsdvreg` includes the options of computing robust and clustered standard errors. However, the memory-saving design of the estimation is especially costly in terms of computing time when robust or clustered standard errors are computed. Therefore, computing robust or clustered standard errors may, in some cases, be prohibitively time consuming.

The program `felsdvreg` checks for collinearity between the explicit right-hand-side regressors right at the start. But collinearity between regressors and fixed effects also poses a problem. Sometimes it is easy to avoid regressors that are collinear with the fixed effects. For example, one can easily avoid including time-constant variables like gender in a model with individual fixed effects. But other cases are more difficult. For example, if school dummies are added as explicit right-hand-side regressors to a model including teacher and student fixed effects, it is hard to know a priori which school

effects are collinear with the teacher and student effects. Such collinearity will only be detected by `felsdvreg` at the moment when the inverse of the moment matrices is computed to solve for the coefficient vector. At that step, the program uses the Mata function `invsym()`, which automatically drops collinear regressors. This is the advantage of using `invsym()` at that stage. However, `felsdvreg` provides the option `cholsolve` to use the Mata solver `cholsolve()`. Using a solver has advantages in terms of precision, but the disadvantage would here be that it does not simply drop collinear regressors but instead fails and issues the error message "matrix has missing values".[13]

Another option is `feffse(`*varname*`)`, which allows you to pass a name of a new variable to store the standard errors of the fixed effects of the second effect (firm effect). All options are also described in detail in the help file accompanying `felsdvreg`.

6 Conclusion

This article has proposed a memory-saving decomposition of the design matrix to facilitate the estimation of a linear model with two high-dimensional fixed effects. This is applicable, for example, to linked employer–employee datasets, but it is also applicable to other data that allow us to fit multiple-way fixed-effects models, such as linked student–teacher data, etc.

A common way to fit such a model is to take into account one of the effects by including dummy variables and to sweep out the other effect by the within transformation (the fixed-effects transformation). If the number of panel units is high, creating and storing the dummy variables can require a lot of computer memory. The decomposition of the design matrix presented in this article reduces the storage requirements. The article also described the Stata ado-file `felsdvreg`, which implements the memory-saving estimation method.

13. `felsdvreg` uses the currently available Mata functions to solve for the coefficient estimates. This is not the most efficient procedure for the present problem because, not only is the design matrix $(\mathbf{X}, \mathbf{F})$ sparse, but so is the moment matrix $\mathbf{A}$. $\mathbf{A}$ has zero entries in all cells where there is no direct worker mobility between the row firm and the column firm. For the solution of systems of linear equations involving sparse matrices, there are more-efficient algorithms than the standard algorithms used here. However, as explained above, the advantage of using `invsym()` is that regressors collinear with the fixed effects can be handled.

7 Appendix

In the example in section 3, $\mathbf{X}'_{i^*}\mathbf{F}_{i^*}$, $\mathbf{F}'_{i^*}\mathbf{F}_{i^*}$, and $\mathbf{F}'_{i^*}\mathbf{y}_{i^*}$ are

$$
\begin{aligned}
\mathbf{F}'_{i^*}\mathbf{F}_{i^*} &= \begin{pmatrix} 1/3 & 1/3 & -2/3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1/3 & -1/3 & 2/3 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1/3 & 0 & 0 & -1/3 & 0 & \dots & 0 \\ 1/3 & 0 & 0 & -1/3 & 0 & \dots & 0 \\ -2/3 & 0 & 0 & 2/3 & 0 & \dots & 0 \end{pmatrix} \\
&= \begin{pmatrix} \phi_{i11} & 0 & 0 & \phi_{i14} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \phi_{i14} & 0 & 0 & \phi_{i44} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix} \qquad (9) \\
\phi_{i11} &= \left(\frac{1}{3}\right)^2 + \left(\frac{1}{3}\right)^2 + \left(-\frac{2}{3}\right)^2 \\
\phi_{i14} &= \left(\frac{1}{3}\right)\left(-\frac{1}{3}\right) + \left(\frac{1}{3}\right)\left(-\frac{1}{3}\right) + \left(-\frac{2}{3}\right)\left(\frac{2}{3}\right) \\
\phi_{i44} &= \left(-\frac{1}{3}\right)^2 + \left(-\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{X}'_{i^*}\mathbf{F}_{i^*} &= \begin{pmatrix} x_{i11} & x_{i21} & x_{i31} \\ x_{i12} & x_{i22} & x_{i32} \\ \vdots & \vdots & \vdots \\ x_{i1K} & x_{i2K} & x_{i3K} \end{pmatrix} \begin{pmatrix} 1/3 & 0 & 0 & -1/3 & 0 & \dots & 0 \\ 1/3 & 0 & 0 & -1/3 & 0 & \dots & 0 \\ -2/3 & 0 & 0 & 2/3 & 0 & \dots & 0 \end{pmatrix} \\
&= \begin{pmatrix} \xi_{i11} & 0 & 0 & \xi_{i14} & 0 & \dots & 0 \\ \xi_{i21} & 0 & 0 & \xi_{i24} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ \xi_{iK1} & 0 & 0 & \xi_{iK4} & 0 & \dots & 0 \end{pmatrix} \qquad (10) \\
\xi_{ij1} &= \left(\frac{1}{3}\right) x_{i11} + \left(\frac{1}{3}\right) x_{i21} + \left(-\frac{2}{3}\right) x_{i31} \\
\xi_{ij4} &= \left(-\frac{1}{3}\right) x_{i11} + \left(-\frac{1}{3}\right) x_{i21} + \left(\frac{2}{3}\right) x_{i31}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{F}'_{i^*}\mathbf{y}_{i^*} &= \begin{pmatrix} 1/3 & 1/3 & -2/3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1/3 & -1/3 & 2/3 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_{i1} \\ y_{i2} \\ y_{i3} \end{pmatrix} = \begin{pmatrix} \upsilon_{i1} \\ 0 \\ 0 \\ \upsilon_{i4} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad (11) \\
\upsilon_{i1} &= \left(\frac{1}{3}\right) y_{i1} + \left(\frac{1}{3}\right) y_{i2} + \left(-\frac{2}{3}\right) y_{i3} \\
\upsilon_{i4} &= \left(-\frac{1}{3}\right) y_{i1} + \left(-\frac{1}{3}\right) yx_{i2} + \left(\frac{2}{3}\right) y_{i3}
\end{aligned}
$$

8 References

Abowd, J., R. Creecy, and F. Kramarz. 2002. Computing person and firm effects using linked longitudinal employer–employee data. Technical Report 2002-06, U.S. Census Bureau. http://lehd.dsd.census.gov/led/library/techpapers/tp-2002-06.pdf..

Abowd, J., F. Kramarz, and D. Margolis. 1999. High wage workers and high wage firms. *Econometrica* 67: 251–333.

Andrews, M., T. Schank, and R. Upward. 2006. Practical fixed-effects estimation methods for the three-way error-components model. *Stata Journal* 6: 461–481.

Andrews, M. J., L. Gill, T. Schank, and R. Upward. 2008. High wage workers and low wage firms: Negative assortative matching or limited mobility bias. *Journal of the Royal Statistical Society, Series A* 171: 673–697.

Davis, P. 2001. Estimating multi-way error components models with unbalanced data structures. *Journal of Econometrics* 106: 67–95.

Greene, W. H. 2003. *Econometric Analysis*. 5th ed. Upper Saddle River, NJ: Prentice Hall.

Grütter, M. 2006. *The Anatomy of the Wage Structure*. Hamburg: Verlag Dr. Kovac.

Harris, N. H., and T. R. Sass. 2007. What makes for a good teacher and who can tell? Manuscript, Florida State University, Tallahassee.

Wooldridge, J. M. 2002. *Econometric Analysis of Cross Section and Panel Data*. Cambridge, MA: MIT Press.

About the author

Thomas Cornelissen is a research and teaching associate at the Institute of Empirical Economics of the University of Hannover, Germany. His research is about applied econometrics and empirical labor economics.