Monitoring, Sanctions and Front-Loading of Job Search in a Non-Stationary Model: How to Use the Code

Bart Cockx[†], Muriel Dejemeppe[‡], Andrey Launov[*] and Bruno Van der Linden[⋆]

[†] Sherppa, Ghent U., IZA and CESIfo, [‡] IRES, Université catholique de Louvain,
[*] U. of Mainz, IZA and CESIfo, [⋆]IRES, Université catholique de Louvain and FNRS

Date: November 21, 2011

Summary

This file provides instructions for replication of all our key results. In Section 1 we describe the contents of the package and address the initial steps necessary to repeat these results. In Section 2 we review the estimation of the unconditional specification of our econometric model. Section 3 provides details on the estimation of the conditional specification as well as on the numerical analysis based on this specification. This includes goodness of fit analysis, external validation exercise and policy evaluation using structural ATT. Current file provides only a short description. More detailed explanation can be provided upon request, should such a necessity arise.

1 Contents of the Package and Technical Preparation

1.1 Structure of the Package

The entire package can be found in the "CDLV.zip" file. To be able to run all our programs extract the contents of this .zip-file to a hard drive, e.g. to C:\.

Root directory of the package is called "**CDLV**". It contains two folders: "**code**" and "**results**". The "**code**" folder stores the code for estimation of the unconditional specification "**code****code_u**", and the code for estimation of the conditional specification "**code****code_c**" of our model, as well as programs for all key simulations and predictions based on the latter specification. Similarly, the "**results**" folder stores the estimation results for the unconditional specification "**results****results_u**" and the estimation results for the conditional specification "**results****results_c**", the latter along with the set of results of numerical simulations and policy evaluations based on the conditional specification.

1.2 Technical Preparation

All programs are written for Matlab, and the model is estimated using Matlab 7.10.0.499.

The code for estimation of the unconditional specification and the code for estimation of the conditional specification are two independent suites with some of the functions having the same name, e.g. the likelihood function, but different contents by virtue of the difference in specification. Therefore, to repeat the estimation of the unconditional model use Matlab "File"-menu to add to Matlab library the folder "**code_u**":

"File" → "Set Path..." → "Add with subfolders" → [select the ...**code****code_u**] → "Save"

To repeat the estimation of the conditional specification and all numerical simulations, the previous path should be deleted from the library using Matlab menu "File" → "Set Path...", selecting all the paths with the root "**CDLV****code****code_u**" and pressing the "Remove" button. After that use the same procedure with the "File"-menu to add to Matlab library the folder "**code_c**":

"File" → "Set Path..." → "Add with subfolders" → [select the ...**code****code_c**] → "Save"

In other words, "**code_u**\ **...**" and "**code_c**\ **...**" directories may not appear in Matlab library simultaneously.

2 Estimation of the Unconditional Specification

2.1 Data and Results File

All data and estimation results for the unconditional specification of the model are conveniently stored in the file "*unconditional_final.mat*" which can be found in the "**results****results_u**" folder. This file contains following objects.

A: Data objects

dmxn - data matrix with duration, benefit, evaluation and earnings information for all selected individuals whose behavior is affected by the program (i.e. is nonstationary); see Appendix for the explanation of the structure of this matrix

dmxs - data matrix with duration, benefit and earnings information for all selected individuals whose behavior is not affected by the program (i.e. is stationary); see Appendix for the explanation of the structure of this matrix

B: Solution objects

b - vector that maximizes the total log-likelihood defined by $\ell\,(d_u, d_e, w^o)$ conditional on the solution for the total log-likelihood defined by $\ell\,(O_1, e)$ stored in **bprob** [1]

bprob - vector that maximizes the total log-likelihood defined by $\ell\,(O_1, e)$ conditional on the solution for the total log-likelihood defined by $\ell\,(d_u, d_e, w^o)$ stored in **b**

hess - Hessian matrix for the objective function based on $\ell\,(d_u, d_e, w^o)$ evaluated in **b**

hessprob - Hessian matrix for the objective function based on $\ell\,(O_1, e)$ evaluated in **bprob**

cov - covariance matrix for the parameter estimates in **b**

covprob - covariance matrix for the parameter estimates in **bprob** [2]

kappa - solution for the set of evaluation-related parameters $\{\kappa_k\}_{k=1}^{3}$ implied by **b** and **bprob** [3]

stats - standard regression output for **b** as shown in Table 5: first column is the parameter vector, second is the standard error, third is the z-test statistic for the significance test and fourth is the p-value implied by the z-test.

statsprob - standard regression output for **bprob** as shown in Table 5: first column is the parameter vector, second is the standard error, third is the z-test statistic for the significance test and fourth is the p-value implied by the z-test.

f - the value of the negative total log-likelihood defined by $\ell(d_u, d_e, w^o)$ evaluated at the solution

grad - the gradient of the negative total log-likelihood defined by $\ell(d_u, d_e, w^o)$ evaluated at the solution

C: Optimization objects

xst - vector of initial values for **b**

xpr - vector of initial values for **bprob** [4]

xkp - vector of initial values for **kappa** [5]

bds - matrix of the lower and upper bounds for each parameter in **b**, defining the range within which the solution for **b** is looked for

opt_ml - Matlab structural object with optimization settings for "fmincon.m"-routine used at the first stage of the two-stage estimation procedure

opt_wr - Matlab structural object with settings for the numerical solution of a root-finding problem used by all the routines that require computation of search effort and reservation wage in the stationary environment; it is a necessary input both at the first and at the second stages of the two-stage estimation procedure

In the folder "**results_u**" one can also find the file "*log_unconditional_final.txt*". It contains the detailed iteration log, documenting the estimation of the unconditional specification.

2.2 Replicating the Estimation Results

Unconditional model has been estimated on a server for high-performance computing that uses Intel Xeon X5570 2.93 GHz processor with eight cores available simultaneously. The total estimation time is about 15 hours. For replication purpose it is strongly advisable to use computer cluster.

To repeat the estimation one should use the function

```
[b,f,grad,hess,bprob,hessprob,kappa]=alg_itp(xst,xpr,xkp,bds,opt_ml,opt_wr,dmxn,dmxs)
```

which can be found in the "**code****code_u**" folder. This is a function that completely specifies the iterative algorithm for the two-stage procedure described in Section 4.3 of the paper. It has eight inputs. First three are the vectors of the initial guesses for the parameters of the model and the evaluation technology (**xst**, **xpr** and **xkp**). Next three are the optimization-related settings, namely the approximate range for the maximizer (**bds**) and the defaults for maximization and root-finding problems (**opt_ml** and **opt_wr**). Last two inputs are the data matrices containing information about individuals affected (**dmxn**) and unaffected (**dmxs**) by the reform.

The estimation routine proceeds as follows. At the initiation step the starting values **xst**, **xpr** and **xkp** are made consistent with each other. Namely, for a given vector **xst** we use the initial values **xpr** to estimate the probability model on the outcomes of the first interview in order to obtain the initial value for β_1^0 that conforms with the average search intensity implied by parameters in **xst**. This is made iteratively (see: ".. probability part (balancing)" in the iteration log). Once solved, the parameter vector **xst** and the updated parameter vector **xpr** are used to find the corresponding starting values of $\{\kappa_k\}_{k=1}^3$ using the initial guess for **xkp** (see: ".. solve for kappas" in the iteration log). This provides consistent starting values which are then used to perform the two-stage estimation as described in Section 4.3.[6]

Function "*alg_itp.m*" has seven outputs, namely the parameter estimates of both stages (**b** and **bprob**), the corresponding Hessian matrices (**hess** and **hessprob**), the values of the evaluation technology parameters (**kappa**) at the solution and the summary measures (function value **f** and gradient

grad) for the negative total log-likelihood of the first stage of the estimation procedure evaluated at the solution. The solution for **b** always lies in the interior, which can be seen from the iteration log in the output summary after each completed optimization.

Using parameter estimates **b** and **bprob** and their respective Hessian matrices we then compute the statistics reported in Table 5.

3 Estimation of the Conditional Specification, Numerical Analysis and Policy Evaluation

3.1 Data and Results File

All data and estimation results for the conditional specification of the model are conveniently stored in the file "*conditional_final.mat*" which can be found in the folder "**results****results_c**". The structure of this file is identical to the structure of the file "*unconditional_final.mat*". For the meaning of the objects stored in this file please consult Section 2.1.

The differences are only in the contents of the data objects, namely, data matrices **dmxn** and **dmxs** contain now five more columns to accommodate covariates (see Appendix for exposition). Clearly the initial values and the estimation results are different as well. Optimization options remain unchanged.

3.2 Replicating the Estimation Results

Conditional model has been estimated on a server for high-performance computing that uses Intel Xeon X5570 2.93 GHz processor with eight cores available simultaneously. Total estimation time is about 55 hours. This means that when repeating the estimation of the conditional specification use of computer cluster is indispensable.

Estimation procedure is identical to the one described in Section 2.2. Syntax is, likewise, the same with

```
[b,f,grad,hess,bprob,hessprob,kappa]=alg_itp(xst,xpr,xkp,bds,opt_ml,opt_wr,dmxn,dmxs)
```

(see "**code****code_c**" folder) delivering all the results. Contents of all inputs and outputs is identical to that of inputs and outputs discussed in Section 2.2.

The entire estimation procedure is summarized in the iteration log "*log_conditional_final.txt*" which can be found in the "**results_c**" folder. Like with the unconditional specification, the maximizer **b** always lies in the interior, which can be seen from the iteration log.

For repeating the estimation of the conditional specification please do not forget to update paths in the Matlab library as described in Section 1.2.

3.3 Replicating Numerical Analysis

All results of the numerical analysis can be found in the folder "**results****results_c****numerics**". This folder contains three Matlab files:

results_gof_1m.mat - summarizes results of the goodness of fit analysis

results_validate.mat - summarizes results of the external validation of the structural model

results_att_xm.mat - contains computed structural ATT for all stages of the reform

together with corresponding log-files. Programs that are used to perform this analysis are collected in the "**code****code_c****pred**"directory.

3.3.1 Goodness of Fit Analysis

Goodness of fit analysis has three aspects: a) fit to the empirical cumulative probability of exit to full-time employment, b) fit to the empirical distribution of wages observed upon transition to full-time employment, and c) fit to the empirical probability of negative evaluation at the first interview. The results for all the aspects of this analysis can be found in the "*results_gof_1m.mat*" file. This file contains following objects.

A: Data, parameter estimates and optimization defaults as described in Section 2.1

dmxn - data matrix

b, **bprob** and **kappa** - parameter estimates

opt_wr - settings for the numerical solution of a root-finding problem

B: Other inputs

b3 - vector of parameter estimates from the conditional exponential regression where exit to the residual state is defined as a failure[7]

we_data - vector of wages observed upon transition to full-time employment (extracted from **dmxn**)

km_we - matrix of Kaplan-Meier estimates of the survivor function for the distribution of observed wages, along with the confidence interval around it, estimated on **we_data**[8]

C: Results

cfr_mod_e - cumulative probability of exit to full-time employment predicted by the model

cfr_non_e - empirical cumulative probability of exit to full-time employment, along with bootstrap confidence interval

sf_we_mod - model prediction of the survivor function for the distribution of wages observed upon transition to full-time employment

sf_we_mod_plot - the same as **sf_we_mod**, augmented by the estimates from **km_we** and arranged in a way convenient for plotting

Programs for performing the goodness of fit analysis can be found in "**\code\code_c\pred\gof_1m**" folder.

To analyze the fit of the cumulative probability of exit to full-time employment we apply the function "*fit_cfr.m*"

```
[cfr_mod_e,cfr_non_e]=fit_cfr(b,b3,bprob,kappa,opt_wr,dmxn,'empl',NBr)
```

This function has eight inputs. First six of them have already been discussed in detail. The seventh input is the field that tells the procedure the destination state upon exit from unemployment. Field 'empl' indicates that we look at transitions to full-time employment.[9] The last input (NBr) gives the number of bootstrap replications of the original sample **dmxn** in order to compute confidence intervals around the empirical exit probability. There are two outputs: **cfr_mod_e** and **cfr_non_e**, which represent the theoretical and empirical cumulative exit probabilities, the latter along with the confidence interval. The results contained in these two output matrices are plotted in Figure 3(a). Log-file "*log_gof_1m.txt*" in "**\results\results_c\numerics**" folder documents the computation.

Function "*fit_we.m*" is applied to analyze the fit of the distribution of observed wages upon the transition to full-time employment

```
[sf_we_mod,sf_we_mod_plot]=fit_we(b,b3,bprob,kappa,opt_wr,dmxn,km_we,'error',NIr)
```

It has nine inputs. First six of them have already been discussed above. The seventh input is the matrix of Kaplan-Meier estimates of the survivor function obtained from observed wages (along with the confidence interval). The eighth input is the field which tells the procedure that the distribution of observed wages should account for measurement error.[10] The last input (NIr) shows how many times each individual in the sample should be repeated in order to draw a simulated sample from the theoretical earnings distribution (see Section 6.2 in the paper for more details). The procedure returns two outputs: **sf_we_mod** - the theoretical prediction of the survivor function for the distribution of observed wages, and **sf_we_mod_plot** - theoretical and empirical estimates of the survivor function arranged in the form convenient for plotting. The results contained in **sf_we_mod_plot** are plotted in Figure 3(b). Again, log-file "*log_gof_1m.txt*" documents the computation.

Finally, to investigate the fit of the probability of negative evaluation at the first interview we apply the function "*gof_test_pne.m*"

```
gof_stat_pne=gof_test_pne(b,bprob,kappa,opt_wr,dmxn)
```

All five inputs of this function are as discussed above. The function has a single output: a vector of three elements, first of which shows the number of observations used in the test, second reports the $\chi^2_{(1)}$-test statistic and third provides the corresponding p-value. Computation of this test is likewise documented in "*log_gof_1m.txt*".

3.3.2 External Validation

Results on the external validation of the model are stored in "*results_validate.mat*" file in the "**\results \results_c\numerics**" folder. Apart from **b** and **opt_wr** already defined above, this file contains following objects.

A: Data

vmx_03 - data matrix with information on benefits, unemployment duration and destinations states in the sample of theoretically notified individuals in 2003; these data are used for validation of the cumulative probability of exit to full-time employment

rmx_0304 - data matrix with information on unemployment duration and destinations states for the individuals between 30 and 32 years of age theoretically notified in 2003-2004; these data are used to estimate the adjustment factor for business cycle conditions (see Section 6.3 in the paper for more discussion)

vmx_03_we - subset of **vmx_03** containing the data for those individuals who exited to full-time employment and whose wage has been recorded upon exit; these data, together with **wmx_03_we** below, are used for validation of the distribution of observed wages

wmx_03_we - data matrix with observed wages corresponding to **vmx_03_we**

km_we - matrix of Kaplan-Meier estimates of the survivor function for the distribution of observed wages, along with the confidence interval around it, estimated for **wmx_03_we**[11]

B: Results

cfr_mod - cumulative probability of exit to full-time employment predicted by the model

cfr_dta - empirical cumulative probability of exit to full-time employment, along with bootstrap confidence interval, adjusted for business cycle conditions

sf_we - model prediction of the survivor function for the distribution of wages observed upon transition to full-time employment

sf_we_plot - the same as **sf_we_mod**, augmented by the estimates from **km_we** and arranged in a way convenient for plotting

In addition, three more outputs - the estimates of the parameters that determine the hazard to the residual state in the validation sample (**b3v**), as well as estimates of piecewise-constant adjustment factors for transition to employment (**delta_e**) and to residual state (**delta_0**) - are stored in the workspace, but these were not reported in the paper for brevity considerations.

Programs for performing the validation analysis can be found in "**\code\code_c\pred\gof_1m \validation**" folder.

Function "*validate_cfr.m*" performs the validation exercise for the cumulative probability of exits to full-time employment. It has the following syntax

```
[cfr_mod,cfr_dta,b3v,delta_e,delta_0]=validate_cfr(b,opt_wr,vmx_03,rmx_0304,NBr)
```

All inputs but the last have already been described above. The last input in this function shows the number of bootstrap replications of the sample **vmx_03** for the estimation of the confidence interval around the cumulative probability of exits to employment.

Function "*validate_cfr.m*" will first estimate the duration model for exits to the residual state, then estimate the adjustment factors, and with this information proceed to estimating model-based and data-based cumulative probability of exit to full-time employment. Two first outputs (**cfr_mod** and **cfr_dta**) contain the model-based and data-based estimates, the latter with the confidence interval, and are used to create Figure 3(a) in the paper. The entire exercise is documented in the log-file "*log_validate.txt*" which lies in the "**\results\results_c\numerics**" folder.

To perform the validation exercise for the distribution of wages observed upon exit to full-time employment we use function "*validate_we.m*". It has the following syntax

```
[sf_we,sf_we_plot]=validate_we(b,opt_wr,vmx_03_we,km_we_03)
```

All inputs and outputs have already been defined. The matrix **sf_we_plot** provides the results of the analysis plotted in Figure 3(b) in the text. Again, log-file "*log_validate.txt*" documents the computation.

3.3.3 Average Treatment Effects on the Treated

Computed ATT for cumulative probability of exit to full-time employment and for expected accepted wages are stored in the folder "**\results\results_c\numerics**", file "*results_att_xm.dta*". Log-file "*log_att_xm.txt*" in the same folder summarizes the computation.

There are four sets of results for the first statistic and four sets of results for the second one. The most obvious way to repeat the computation of all the ATT would be to open the stored workspace file "*results_att_xm.dta*" and enter the command

```
get_att_all
```

This command corresponds to the .m-file "*get_att_all.m*" in the "**\code\code_c\pred\att_xm**" folder and collects all functions for computing the structural average treatment effect on the treated for any of the two statistics between any of the two interviews. For instance, function

```
[att_cfr_1m_d,att_cfr_1m_p,cfr_tre_1m,cfr_ntr_1m]=att_cfr_0_to_1(b,b3,bprob,kappa,opt_wr,dmxn)
```

in "*get_att_all.m*" is used for computation of the effect on the cumulative probability of exit to full-time employment between the notification and the first interview (0_to_1). The rest of the functions are defined in the same way, differing only in the interview designation ("1_to_2", "2_to_3" and "after_3") in the name of the function. All inputs of these functions have been already discussed by now. The last input is always the data matrix. For computation of the effect between the notification and the first interview this is the original data matrix (**dmxn**) which we use for the estimation of the model. For computation of the effects between any other two adjacent interviews these are the simulated data matrices which take into account dynamic selection by the time of each next evaluation. The simulated data matrices (**dmxn_for_2m**, **dmxn_for_3m**, and **dmxn_for_a3**) for respective intervals are available in the same workspace.[12]

Function "*att_cfr_0_to_1.m*", like the rest of the functions for calculation of the the ATT for the cumulative exit probability to employment, has four outputs:

att_cfr_1m_d - ATT expressed as a difference; this statistic is plotted in Figure 5, upper-left corner

att_cfr_1m_p - ATT expressed as a percentage change

cfr_tre_1m - cumulative probability of exit for treated

cfr_ntr_1m - counterfactual cumulative probability of exit for treated in absence of policy

For ATT of all other interviews, we get the same four output objects differing only in the interview designation ("_2m", "_3m" or "_a3").

For computation of the ATT for mean observed wages upon transition to full-time employment the same structure applies. For instance, average treatment affect on the treated for this statistic between notification and the first interview can be obtained using the function

```
[att_aw_1m_d,att_aw_1m_p,aw_tre_1m,aw_ntr_1m]=att_aw_0_to_1(b,bprob,kappa,opt_wr,dmxn)
```

which likewise appears in the "*get_att_all.m*" file. The rest of the functions are defined in the same way, differing only in the interview designation ("1_to_2", "2_to_3" and "after_3"). Inputs of these functions have been just discussed. Outputs, using the example of "*att_aw_0_to_1.m*", are:

att_aw_1m_d - ATT expressed as a difference; this statistic is plotted in Figure 6, upper-left corner

att_aw_1m_p - ATT expressed as a percentage change

aw_tre_1m - expected wage upon exit for treated

aw_ntr_1m - counterfactual expected wage upon exit for treated in absence of policy

For all other interviews we again receive the same four output objects differing only in the interview designation ("_2m", "_3m" or "_a3").

Notes

[1]Note that Matlab optimization routines minimize functions instead of maximizing them.

[2]Note that this is a (1×1)-matrix which does not contain the information about the standard error of the parameter estimates of β_1^1 also reported in Table 5. The reason is that the estimate of β_1^1 can be obtained simply by regressing the outcomes of the evaluation of all individuals, who were evaluated independently of their search effort, on a constant. The specification of the probability

model is as shown in eq.(20). This regression has been run externally and its replication is not offered here because of its triviality.

[3]The first element is always set to one ($\kappa_1 = 1$). See Section 4.2.1 in the paper for discussion.

[4]Note that the second element already contains the parameter estimate of β_1^1 obtained externally; see Endnote 2 for details.

[5]Note that the first element is always set to one ($\kappa_1 = 1$); see Endnote 3 for details.

[6]Making the initial values consistent with each other can also be skipped. However in this case we would need much higher number of the iterations of Stages 1 and 2 to balance the estimated parameters before they converge. Since Stage 1 includes maximization of the function that is extremely complex numerically, this this would make the entire estimation procedure prohibitively long.

[7]Regression performed externally.

[8]Estimation performed externally.

[9]Option 'meet', indicating exits to the interview, is also programmed, but the results are not reported in the paper, though available upon request.

[10]Alternatively, option 'true', which tells the programme to ignore measurement error, is available. However its application is less meaningful in the present context.

[11]Estimation performed externally.

[12]We create these simulated data sets with the help of the programs stored in "**\code\code_c\pred \att_xm\sim_xm**". For the sake of brevity we do not describe how these programs work, but their inputs and outputs should be self-speaking.

Appendix

Here we schematically outline the structure of the data matrices.

dmxn For individuals whose behavior is affected by the reform the data matrix (**dmxn**) is arranged as follows:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)-(20)	(21)-(27)	(28)	(29)	(30)	(31)	(32)	(33)	(34)-(36)	(37)
ID	b_h	b_l	$e=0,1$			d_{sch}	d_{del}	I_E	w^o	d_{job}	I_U	O_1			d_{rest}	I_E	w^o	d_{job}	I_U			
62	325	0	0	5	2	6	NaN	1	1248	20	0	NaN			NaN	NaN	NaN	NaN	NaN	1		1001
70	1005	802	0	15	2	8	6	NaN	NaN	NaN	NaN	1			NaN	NaN	NaN	NaN	NaN	1		1001
...	...	...	...	...	...	...	...	...	...	...	...	...			...	...	...	...	...	...		...
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)-(18)		...							(33)	(34)-(36)	(37)
ID	b_h	b_l	$e=0,1$	d_{job}	I_U	d_{sch}	d_{del}	I_E	w^o	O_1												
11841	835	601	1	2	1	6	NaN	1	1155	NaN										1		1100
...	...	...	...	...	...	...	...	...	...	...										...		...

Column (1) shows the individual ID number. Next two columns contain current benefit level (2) and the benefit level if sanction applies (3). Column (4) tells whether the person is evaluated on the basis of search effort ($e = 0$) or independent of search effort ($e = 1$). Columns (5) and (6) report duration until selection, and between selection and notification, respectively.

Considering those who are evaluated on the basis of search effort (value "0" in the fourth column), column (7) displays the duration of unemployment since notification and at most up to the scheduled date of the 1st interview, and column (8) reports the length of the interview delay. If exit out of unemployment is before the scheduled date of the interview, column (8) cannot assume any value. Once exit out of unemployment occurs faster than transition to the interview, column (9) tells whether exit out of unemployment is to full-time employment ($I_E = 1$) or to a residual state ($I_E = 0$). Whenever the transition is to employment, column (10) reports the accepted wage, column (11) reports the subsequent job duration and column (12) tells whether job loss results in a transition back to unemployment ($I_U = 1$) or to a residual state ($I_U = 0$). Once transition to interview is faster than transition to full-time job, columns (9)-(12) cannot assume any value and individual gets evaluated. The result of evaluation at the first interview (O_1) is reported in column (13). It takes value "1" if evaluation is positive and value "2" if evaluation is negative. It assumes no value if evaluation never comes.

The structure of columns (7)-(13) is repeated for the span between the 1st and the 2nd interview [columns (14) to (20)] and the span between the 2nd and the 3rd interviews [columns (21)-(27)]. After the third, i.e. the last, evaluation the programme terminates, i.e. no more evaluations and no more delays thereof are possible. As a result, beyond the last evaluation [column (27)] we have only five columns: (28)-(32) left, where the columns corresponding to (8) and (13) drop out. This completes the structure of the data matrix for the estimation of the unconditional specification. For the conditional specification this matrix has five more columns: (33) contains an intercept, (34)-(36) contain dummy variables used in the analysis and (37) is the identifier of unique homogeneous groups defined with respect to observed characteristics.

Considering those who are evaluated independent of search effort (value “1” in the fourth column), meaning that there has been a transition to full-time employment between selection and notification, the structure is slightly different in the beginning. Column (5) shows the duration of this employment spell and column (6) is a dummy variable that takes value “1” if transition is back to unemployment. Columns (7)-(10) are identical in meaning to columns (7)-(10) of the former case ($e = 0$). Column (11) reports the result of evaluation at the first interview (O_1) if the first interview ever occurs. Once evaluation is negative, the person looses the privilege of being evaluated independently of search effort, so columns (12)-(18) repeat the structure of columns (7)-(13), and so on.

dmxs For individuals whose behavior is not affected by the reform, arrangement of the data matrix (**dmxs**) is simpler:

(1) ID	(2) b_h	(3)	(4)	(5) I_E	(6) w^o	(7) d_{job}	(8) I_U	(9)	(10)-(12)	(13)
88	865	2	2	1	1184	26	0	1		1110
1307	325	5	3	0	NaN	NaN	NaN	1		1101
...	...	...	...	...	...	...		...		...

Column (1) shows the individual ID number. Column (2) reports current benefit level. Columns (3) and (4) record the elapsed duration of unemployment at the moment of notification and residual duration of unemployment since notification. Column (5) is a dummy variable that takes value “1” if transition from unemployment is to full-time employment. Whenever transition is to full-time employment, we record the accepted wage [column (6)], the length of job spell [column (7)] and the destination state upon match dissolution. If this destination state is unemployment, column (8) shows value “1” (and value “0” otherwise). Columns (1)-(8) make the data matrix that is used for the estimation of the unconditional specification. For the conditional specification this data matrix is augmented by columns (9)-(13) which contain information about covariates. The meaning of these columns is identical to the meaning of columns (33)-(37) of **dmxn**.