Multilevel Modelling of Country Effects: A Cautionary Tale

Supplementary Material

Mark L. Bryan
(ISER, University of Essex)

Stephen P. Jenkins
(London School of Economics, ISER, and IZA, Bonn)

This document contains the Supplementary Material referred to in the main text.

Further details can be found in:

> Bryan, M. L. and Jenkins, S. P. (2013). 'Regression analysis of country effects using multilevel data: a cautionary tale', Working Paper 2013-14**.** Colchester: ISER, University of Essex. https://www.iser.essex.ac.uk/publications/working-papers/iser/2013-14

A full list of References for work cited in this appendix is found in the main text or in Bryan and Jenkins (2013, op. cit.)

Contents

§1. Regression analysis of multilevel country data: four approaches

We review four regression approaches that an analyst might use with multilevel country data, based on the following linear model for a metric outcome variable:[1]

$$y_{ic} = \boldsymbol{X}_{ic}\boldsymbol{\beta} + \boldsymbol{Z}_c\boldsymbol{\gamma} + u_c + \varepsilon_{ic}, \quad \text{with } i = 1, \ldots, N_c;\ c = 1, \ldots, C. \tag{1}$$

Outcome y_{ic} for each person i in country c is assumed to depend on both observed predictors and unobserved factors. $\boldsymbol{X}_{ic}$ contains variables that summarise individual-level characteristics such as age, education or marital status; $\boldsymbol{Z}_c$ contains variables summarising country-level features such as socio-economic institutions or labour markets. There are also unobserved individual effects (ε_{ic}) and country effects (u_c) that are each assumed to be normally distributed and uncorrelated with $\boldsymbol{X}_{ic}$ and $\boldsymbol{Z}_c$.

Pooling the data for all countries (and using cluster-robust standard errors)

A first approach is to simply pool the data from all of the country surveys. In this approach the unobserved individual and country effects are combined into a single composite error term. The outcome is no longer modelled conditional on a country's unobserved characteristics, instead the $\boldsymbol{\beta}$ parameters are interpreted as the effect of $\boldsymbol{X}_{ic}$ averaged over all possible values of u_c. This is sometimes called the population-averaged (PA) effect, to be distinguished from the cluster-specific (country-specific) effect that conditions on u_c (Neuhaus et al. 1991). In linear models, the PA and cluster-specific effects are identical because the effect of $\boldsymbol{X}_{ic}$ does not depend on the other variables (unlike in non-linear models). Whatever the case, there are consequences for the standard errors if one disregards the nesting of observations within countries and the fact that individuals within a country share unobserved characteristics (u_c is an omitted variable). This leads to underestimation of the standard errors of $\boldsymbol{\beta}$ because the within-group (intra-class) correlation across individual units is not accounted for (Moulton 1986).

Fortunately, it is straightforward to apply a 'Moulton correction' or, more commonly, to allow for more general correlation structure among individuals within countries using estimates of cluster-robust standard errors where the clusters are the countries (Angrist and

[1] Our discussion is limited to the classical statistical framework favoured by most applied researchers. Bayesian methods offer a potential way to address the small numbers issues, contingent on making assumptions about 'prior distributions' of parameters including regarding country effects. See inter alia Browne and Draper (2006 and Gelman (2006). Bayesian methods are not yet widely used by social science researchers. One exception is the application by Kedar (2005) in which the number of second level units is 14.

Pischke 2009: 312–3).[2] Another possibility is to derive standard errors with block-bootstrap techniques (Angrist and Pischke 2009: 315; Cameron, Gelbach, and Miller 2008). Although cluster-robust standard errors are easy to derive nowadays, reliance on them is a conservative strategy because the within-country correlation is controlled for but not explicitly modelled. There are no estimates of parameters describing the distributions of the unobserved factors.

The other three approaches (which are cluster-specific) account for the hierarchical nature of the data explicitly.

Separate models for each country

Researchers can fit a separate model to each country's dataset. In this case, any country effect (u_c) is absorbed into, and cannot be identified separately from, the intercept term in each country's regression model (and so is a fixed parameter included as an element of $\boldsymbol{\beta}$). This approach has the advantage of allowing the estimates of the coefficients on individual-level characteristics (the elements of $\boldsymbol{\beta}$ other than the intercept) to differ across countries. In addition, the variance of the individual-specific error is also country-specific.

Country fixed effects (FE) models

In a fixed effects (FE) approach, the data from the country surveys are pooled but the model specification includes distinct country intercepts (estimated as the coefficients on country binary indicator variables). Again, the country effects are treated as fixed parameters rather than random terms, with each country intercept representing the effects of unobserved factors that are shared within each country. In the simplest case, the individual effects (the non-intercept elements of $\boldsymbol{\beta}$) are constrained to be equal across countries, but they can be allowed to differ between countries by interacting subsets of individual-level characteristics with the country indicator variables. Estimates from a model that includes a full set of interactions between individual characteristics and the country dummies are not equivalent to the estimates derived from distinct country regressions because the residual error variance is constrained to be the same across countries in the former case but not in the latter.

[2] In Stata, one would use the regression command option `vce(cluster` *`country_identifier`*`)`.

Country random effects (RE) models

The random effects (RE) approach also pools the data and allows for country effects. However, rather than treating these as distinct values each of which can be estimated, they are modelled as random draws from a distribution (usually normal) with mean zero and variance which is estimated (estimates of the separate country effects can be obtained from the estimated parameters using Empirical Bayes techniques; see Raudenbush and Bryk 2002: 47–8). Although this approach is termed 'random effects', the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ remain fixed in the simple case; in a more complicated RE model they may also be allowed to vary randomly (see Bryan and Jenkins 2013: section 4). One of the attractions of the RE approach is that country-level regressors can also be used as model predictors (see below). By contrast, in the FE approach, country differences are fully characterised by the country indicator variables.

The RE model is the prototypical multilevel (hierarchical) model with random intercepts. A key parameter is the intra-class correlation $\rho = \sigma_u^2 / (\sigma_\varepsilon^2 + \sigma_u^2)$, where σ_ε^2 and σ_u^2 are the variances of the individual and country random effects respectively. (Individual random effects (ε_{ic}) and country random effects (u_c) are assumed to be uncorrelated with $\boldsymbol{X}_{ic}$ and $\boldsymbol{Z}_c$ and with each other.) The intra-class correlation summarises the extent to which unobserved factors within each country are shared by individuals. It tends to zero as $\sigma_u^2 \to 0$. Assuming that the correlation structure of the random effects has a particular form leads to more efficient estimates of the individual-level effects represented by $\boldsymbol{\beta}$, i.e. estimates with standard errors smaller than the cluster-robust ones. (Of course, the efficiency gain is conditional on the model being correct.) Estimation methods for this type of model include generalised least squares (GLS), full maximum likelihood (FML) and restricted maximum likelihood (REML): see Hox (2010) for a comparative discussion. Estimates of the fixed parameters are unbiased (Kackar and Harville 1981, 1984). However, if the number of groups is small, and even if the group sizes are large, estimates of the variance components and of their standard errors are imprecise and likely to be biased downwards (Hox 2010: 233, Raudenbush and Bryk 2002: 283). Estimates of the standard errors of fixed parameters are also affected by the uncertainty in the variance estimates when the number of groups is small: they are biased downwards and the distribution of test statistics is unknown (Raudenbush and Bryk 2002: 282). However, adjustments are available for inference about the fixed effects

(Kenward and Roger 1997) or if the random effects are not normally distributed (Carpenter et al. 2003).

Bryan and Jenkins (2013) discuss extensions of the model to include country-specific slope parameters (Section 4) and non-linear models of binary dependent outcomes (Section 6).

§2. Choice of approach

Because the four approaches differ in fundamental ways, one cannot straightforwardly recommend one approach over another. First of all, there are conceptual distinctions between them that go beyond questions of statistical specification. The pooled approach differs from the other three in that it yields population-averaged effects rather than cluster-specific effects. The two are identical in a linear model, but in non-linear models the effect of $\boldsymbol{X}_{ic}$ on y_{ic} will depend on the values of the other variables (including u_c): thus the effect in one country will differ from that in another country (all else equal) and also from the average effect over countries. While the pooled approach is approach is straightforward to implement, much cross-country research includes an explicit focus on the distinct contributions of unobserved country effects and so researchers usually implement a cluster-specific approach.

Within the cluster-specific approaches there are also conceptual distinctions between FE and RE, especially when using multilevel country data for only a few countries. In the FE approach, the emphasis is on the uniqueness of each country: the country effect (e.g. national culture or institutions) is treated as a characteristic that cannot be transferred to another national context. It is an effect that needs to be included as a control in the model, but each country's estimate has no particular meaning regarding another country. That is, estimates from an FE approach (intercepts and coefficients) relate specifically to the set of countries included in the sample and cannot be generalised out of sample. As an example, FE estimates from a dataset including respondents from the original 15 European Union member states could not be applied to describe outcomes for the 12 new member states with their very different institutions and history. (The post-war experience of Slovenia is very different to that of France, for instance.)

Another consequence of the FE approach is that country-level variables cannot be included as additional predictors (e.g. parental leave laws affecting couples' division of childcare time) because the country intercepts already fully encapsulate cross-country differences (Snijders and Bosker 2012: 46). The limited conclusions in this case are a consequence of the agnostic view about the nature of country effects. To say more, additional assumptions have to be made.

The emphasis in RE models is very different: the set of countries included in the analysis is modelled as a sample from a larger population of countries defined in terms of observed country characteristics. Any remaining unobserved country effects are treated as being generated by some common mechanism and so are 'exchangeable' between countries (Snijders and Bosker 2012: 46–47). The regression intercept is a population average (a common European intercept in the EU example) and deviations from this average are assumed to be uncorrelated with country-level variables included in the model. With these assumptions, the RE results can be generalised to other countries with different policies and institutions. For example, estimate of the effects of parental leave legislation on childcare time based on the old EU countries may be applied to possible legislative changes in the new member states. Clearly the exchangeability assumption is not only strong but also potentially unrealistic (depending on the research context).

The second consideration relates to statistical performance. Provided there is no correlation between the unobserved group-specific effect and the regressors, FE and RE both deliver consistent estimates of $\boldsymbol{\beta}$ but the RE approach is more efficient because it 'borrows strength' from between-group variation (FE uses only within-group variation). However, in practice, the difference between the RE and FE estimates is likely to be negligible when using cross-country data that contain many more observations within countries than there are countries (large N_C, small C). This is because, with large N_C, almost all the variation used in RE estimation is from within, rather than between, countries.[3] Thus the efficiency loss from using FE rather than RE (to estimate $\boldsymbol{\beta}$) may be negligible: with only a few countries there is little potential to 'borrow strength' across them.

Because the differences between the FE and RE estimates of $\boldsymbol{\beta}$ are likely to be minor when using cross-county data, the choice between the two approaches (and the other methods) may largely depend on which parameters are the substantive focus of interest. Analysts primarily interested in the individual effects associated with observed predictors ($\boldsymbol{\beta}$) may favour the FE approach or separate equations. On the other hand the RE approach is the natural choice if the focus is on the effects ($\boldsymbol{\gamma}$) of country-level predictors or the variance component structure. To some extent this aspect is related to disciplinary conventions.

[3] For example GLS estimation of (1) weights between- and within-country variation as a function of $\sigma_\varepsilon^2/(\sigma_\varepsilon^2 + N_C\,\sigma_u^2)$. As N_C becomes large, the fraction of between-country variation used tends to zero and GLS converges to the within-country (FE) estimator.

Economists have conventionally avoided RE approaches, preferring to use one of the other three approaches. Other social scientists, including quantitative sociologists, have tended to favour the multilevel or hierarchical RE modelling approach. We also focus the discussion on a RE framework, given our interest in the effects of both individual- and country-level predictors (and random country-specific parameters).

§3. The two-step approach

The model specified in equation (1) of the main text may also be estimated in two steps. This two-step approach has several advantages: first, it highlights the sources of variation in the data and illustrates why a small number of countries affects the reliability of estimates; second, given large enough sample sizes in each country, the estimates have the correct standard errors (and are unbiased) and so can be used as a benchmark for the other methods; and third, the two-step method leads naturally to an alternative (or complementary) graphical approach that provides a non-statistical view of country-level variation.

The two-step approach consists of one regression at the individual level and another regression at the country level (for discussions see Hanushek 1974, Saxonhouse 1976, Borjas and Sueyoshi 1994, Card 1995, a special issue of *Political Analysis* introduced by Kedar and Shively 2005, Donald and Lang 2007, and a textbook discussion by Wooldridge 2010: chapter 20).

In the first (within country) step, we estimate

$$y_{ic} = \boldsymbol{X}_{ic}\boldsymbol{\beta} + v_c + \varepsilon_{ic}, \quad \text{with } i = 1, \ldots, N_c;\ c = 1, \ldots, C \qquad \text{(SM1)}$$

where v_c is a fixed effect for country c that combines both observed and unobserved country characteristics, i.e. $v_c = \mathbf{Z}_c\boldsymbol{\gamma} + u_c$. In practice, this is fitted either by letting v_c be a country-specific binary indicator variable in an OLS regression (cf. approach 2 above) or by using the within-group estimator with the country as the group (for textbook discussion of both estimation approaches, and their equivalence, see Hsiao 2003: section 3.2). In the second step we estimate

$$\hat{v}_c = \alpha + Z_c\boldsymbol{\gamma} + \eta_c \ , \quad \text{with } c = 1, \ldots, C. \qquad \text{(SM2)}$$

where $\hat{v}_c$ is an estimate of the country-specific fixed effect and η_c is a residual error term. Depending on the first-step estimation method, $\hat{v}_c$ is either the coefficient on the country indicator variable or is derived from the estimates as $\hat{v}_c = \bar{y}_c - \bar{X}_c\hat{\boldsymbol{\beta}}$, where the bars over variables denote means taken over all individuals within a country. In general, η_c will be heteroscedastistic because sampling error varies across countries and so the second step would normally be estimated by GLS. However, with large N_c, the sampling error should be sufficiently small that the second step can be estimated by applying OLS to the C country-level observations (Donald and Lang 2007, Wooldridge 2010: 891–2).[4] It may be noted that the two-step method is very similar to a meta-analytical approach in which each country corresponds to a separate country study. The meta-analysis combines the separate effect estimates from each 'study' ($\hat{v}_c$), taking account of their precision, to produce an overall effect estimate (α) and estimates of the impact (γ) of study characteristics (Snijders and Bosker 2012: 37–9).

Under the assumptions of the basic model (Section 2) and with large N_c, the estimates of both $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are unbiased and have the correct standard errors. In addition the t statistics and p-values reported as standard by software packages will lead to reliable hypothesis tests. Moreover, OLS at step 2 provides an unbiased estimate of the variance of the country effects, σ_u^2. These properties apply even if there are few countries (small C), and so the two-step method can be seen as a useful benchmark for comparison with the other approaches. Closer consideration of the two-step method also highlights a number of issues that apply more generally to estimation using clustered data with few groups.

First, step 1 uses only within-country variation to estimate the individual-level parameters, $\boldsymbol{\beta}$, in contrast to the RE (and pooled) approach, which also uses between-country variation. The ability to 'borrow strength' from across groups (countries) is often cited as an advantage (increasing efficiency) of the RE approach in estimating $\boldsymbol{\beta}$. But, as noted by Aachen (2005), with only a small number of groups but large numbers of individual units within groups, there is much less need (and less potential) to borrow strength across groups. In this case the RE approach uses mainly within-country variation and the resulting $\boldsymbol{\beta}$ estimates will in practice be close to the two-step (or equivalently FE) estimates (as illustrated in Section 8).

Second, the second-step regression makes clear that estimation of the $\boldsymbol{\gamma}$ parameters associated with country-level predictors is based on only C observations, because estimation

---

[4] The country-level error, η_c, in (3) can be written $\eta_c = u_c + \bar{\varepsilon}_c + \bar{X}_c(\beta - \hat{\beta})$. With large N_c, $\bar{\varepsilon}_c$ can be ignored because its variance ($=\sigma_\varepsilon^2/N_c$) will be negligible compared to that of u_c, the unobserved country-specific effect. The term $\bar{X}_c(\beta - \hat{\beta})$, the sampling error of the estimated country effects, is heteroscedastistic, but with large N_c it is also small. As $N_C \to \infty$ the equation error then converges to u_c, which by assumption is homoscedastistic and normal (Donald and Lang 2007: 225; Wooldridge 2010: 892). Therefore step 2 can be estimated efficiently using OLS, with hypothesis testing of $\boldsymbol{\gamma}$ based on the t-distribution (with C-k-1 degrees of freedom, where k is the number of $\boldsymbol{Z}_c$ variables). In the more general case of a heteroscedastistic error η_c at step 2, GLS would be the efficient estimator. Borjas and Sueyoshi (1994), Hanushek (1974) and Donald and Lang (2007) provide alternative calculations of the weighting matrix for feasible GLS. However, feasible GLS estimates are only consistent (and distributed normally) for large C (because estimates of the weighting matrix are 'unreliable' with small C) . Given the large N_c, small C structure of most cross-country survey data, OLS (relying on a large N_c approximation) appears preferable to GLS (relying on large C approximation).

uses either the coefficients on country-level indicator variables or country means (the dependent variable in (3)). No matter how many individual-level observations (N_c) underlie the calculation of these means, we are effectively using only C observations at the country level (Donald and Lang 2007; Wooldridge 2010, chapter 20). For the same reasons, the variance of the country-specific effect (σ_u^2) is also likely to be estimated imprecisely.

The small number of countries has several implications. First, the country-level parameters, $\boldsymbol{\gamma}$, are estimated much less precisely than would be suggested by OLS estimation of (1) using all individual-level observations. Ignoring the group-level error results in standard errors that are too small (Moulton 1986).

Second, even if cluster-robust standard errors are used, the assumption that u_c is normally distributed is crucial for hypothesis testing because we cannot rely on large sample sizes to provide an asymptotically normal distribution of the parameter estimates. If u_c is not normally distributed, tests of statistical significance will not in general be accurate. Furthermore, even if u_c is normal, hypothesis tests and confidence intervals should be based on the t distribution and not the standard normal (z) distribution.[5] For small C, the t critical values are considerably larger than the corresponding z values, implying that standard z tests will find statistically significant results too often.

Third, a small C places a practical limit on the number of variables that can be included in $\boldsymbol{Z}$. With only a small number of countries, it is impossible to disentangle institutional effects in detail. Even calculating the variance of the country effects is problematic when the number of countries is small. Thus formal statistical inference is difficult. Nonetheless one can always compare the country effects $\hat{v}_c$ derived from the first step of estimation using less formal descriptive methods such as exploratory data analysis including graphs. See Bowers and Drake (2005) and the empirical illustration in Section 8 of this Supplementary Material document for examples.

[5] As noted above, if the second step is estimated by OLS, standard software will produce t-statistics that are correctly referred to the t-distribution with C–k–1 degrees of freedom (where k is the number of country-level variables).

§4. Generation of data for Monte-Carlo simulations

The generated regressors for the Monte-Carlo simulations were based on the empirical distribution of variables among women aged 18–64 years from 26 countries in the 2007 EU-SILC data (using a sample of working women for the hours model and a sample of all women for the participation model). The regressors are *age* (continuous), *age-squared*, *cohab* (whether married or cohabiting; binary), *nownch* (number of own children; integer), *isced* (educational level; four categories with the lowest excluded from the regressions), and *chexp* (country spending on childcare and pre-primary spending as a % of GDP, continuous).

We specified the joint distribution of the regressors by exploiting the fact that each combination of regressor values defines a cell with an associated probability of occurrence.

We derived the cell probabilities from the empirical frequency distributions and then generated data sets reflecting these distributions for each value of *C* (and for each model) using a random number generator. To incorporate age into the cell-based approach, we first fitted either a Singh-Maddala distribution (hours models) or a uniform distribution (participation models) in the EU-SILC data.

The fitted parameters were used to generate values of age between 18 and 64 in the simulated data (values used in the regressions). They were grouped into five categories in order to construct the cells.

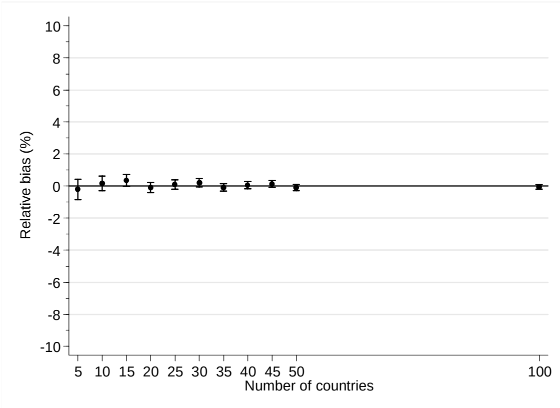
Stata code for running the Monte Carlo simulations is provided in §9.

§5. Basic linear model: simulations repeated using different initial seed

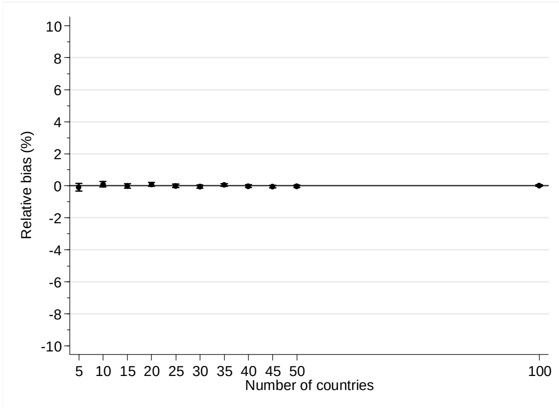
The graph below was derived in exactly the same way as Figure 1 in the main text, with the exception that the simulations used a different initial seed. Observe that the 95% CIs in the graph for *chexp* include zero for all values of *C*. See the main text for further discussion.

Relative parameter bias (%): linear model with random intercept and country-level regressor (basic model for ‘hours’), selected parameters

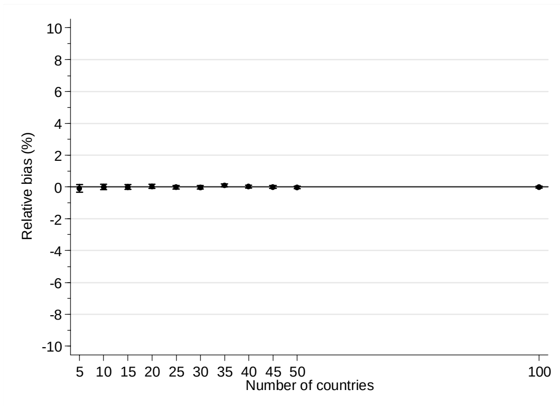
Individual-level fixed effect: *cohab*



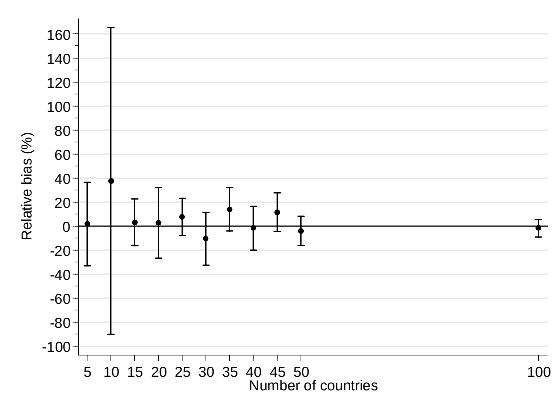
Individual-level fixed effect: *nownch*



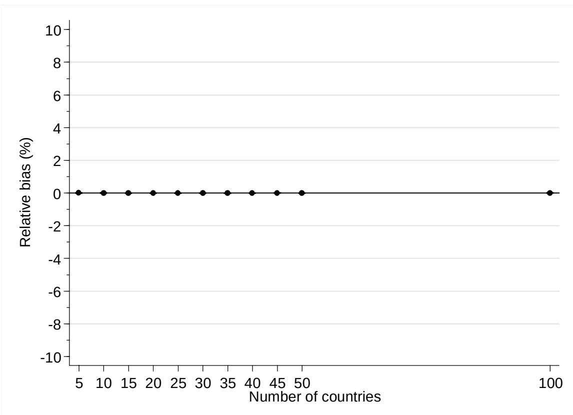
Individual-level fixed effect: *age*



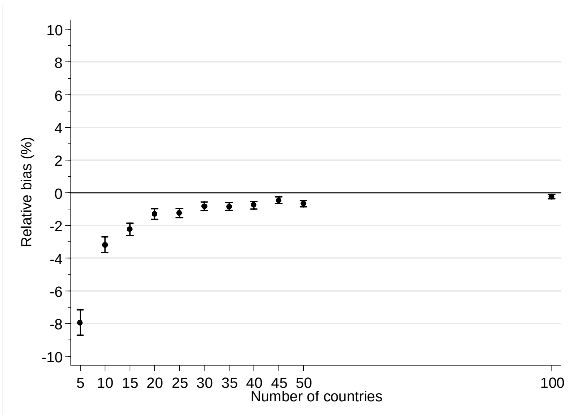
Country-level fixed effect: *chexp*



Individual-level variance: *sig_e*



Country-level variance: *sig_u*



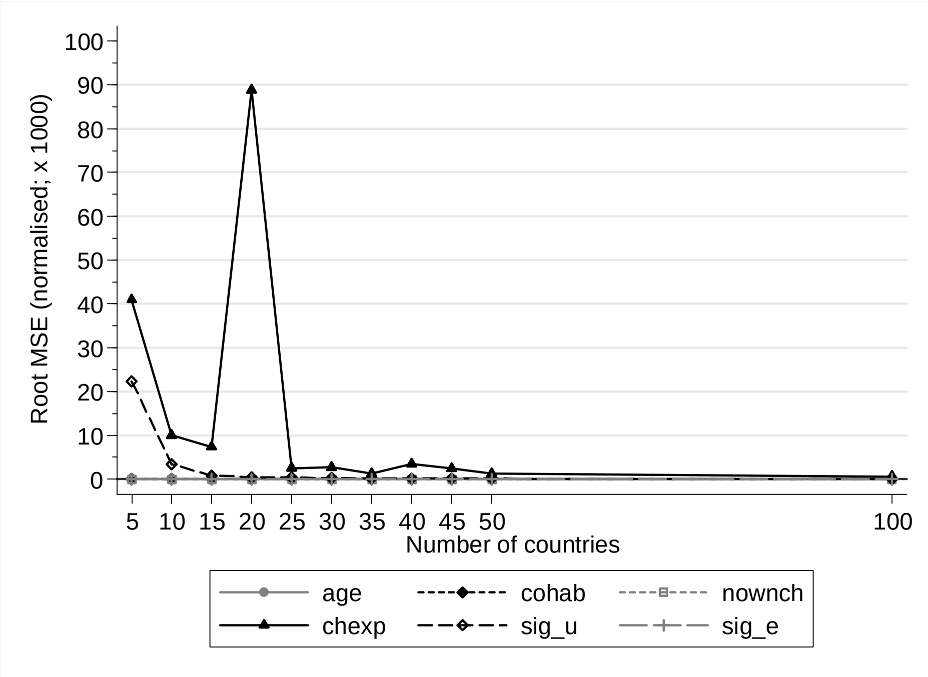
Notes. The filled circles show estimates of relative parameter bias, and the vertical bars show their 95% confidence intervals (see main text for definitions). The parameters and their labels are defined in Table 4 and the main text. Number of Monte-Carlo replications, R = 10,000. Observe the much larger vertical scale for the country-level fixed effect *chexp*.

§6. Assessment of estimator accuracy in terms of RMSE

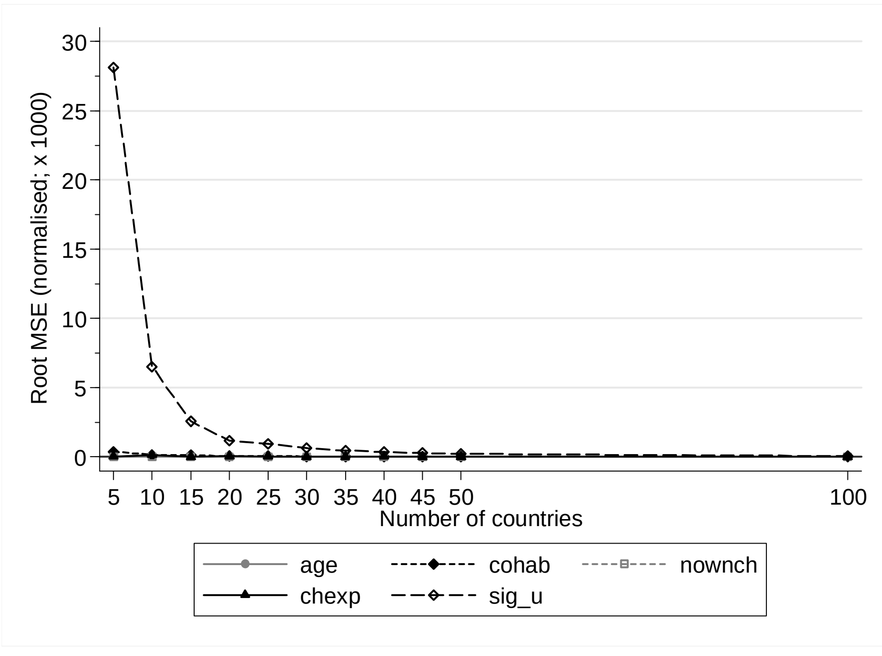
In the main text, two dimensions of estimator accuracy (bias and variability) are summarised separately. Sometimes analysts summarise the two dimensions using a composite statistic, RMSE. RMSE has the same scale as the corresponding parameter and so, in order to make the statistics comparable across parameters, each RMSE statistic has been expressed relative to the absolute value of the corresponding parameter (and multiplied by 1000 for legibility).

The conclusions in the main text are confirmed: there is relatively poor accuracy at small values of *C* of estimates of the country-level fixed effect and the country-level variance parameters.

Basic linear model



Basic logit model



§7. Monte-Carlo simulation estimates for extended linear and logit models

The table below shows the parameters used to specify the extended models of hours and participation that are described in the main text. The extended models differ from the basic models by the inclusion of two random slopes (on *cohab* and *nownch*) and individual-country interactions (of *cohab* and *nownch* with *chexp*).

Figures S.1–S.4 that follow summarise the Monte-Carlo results for these models.

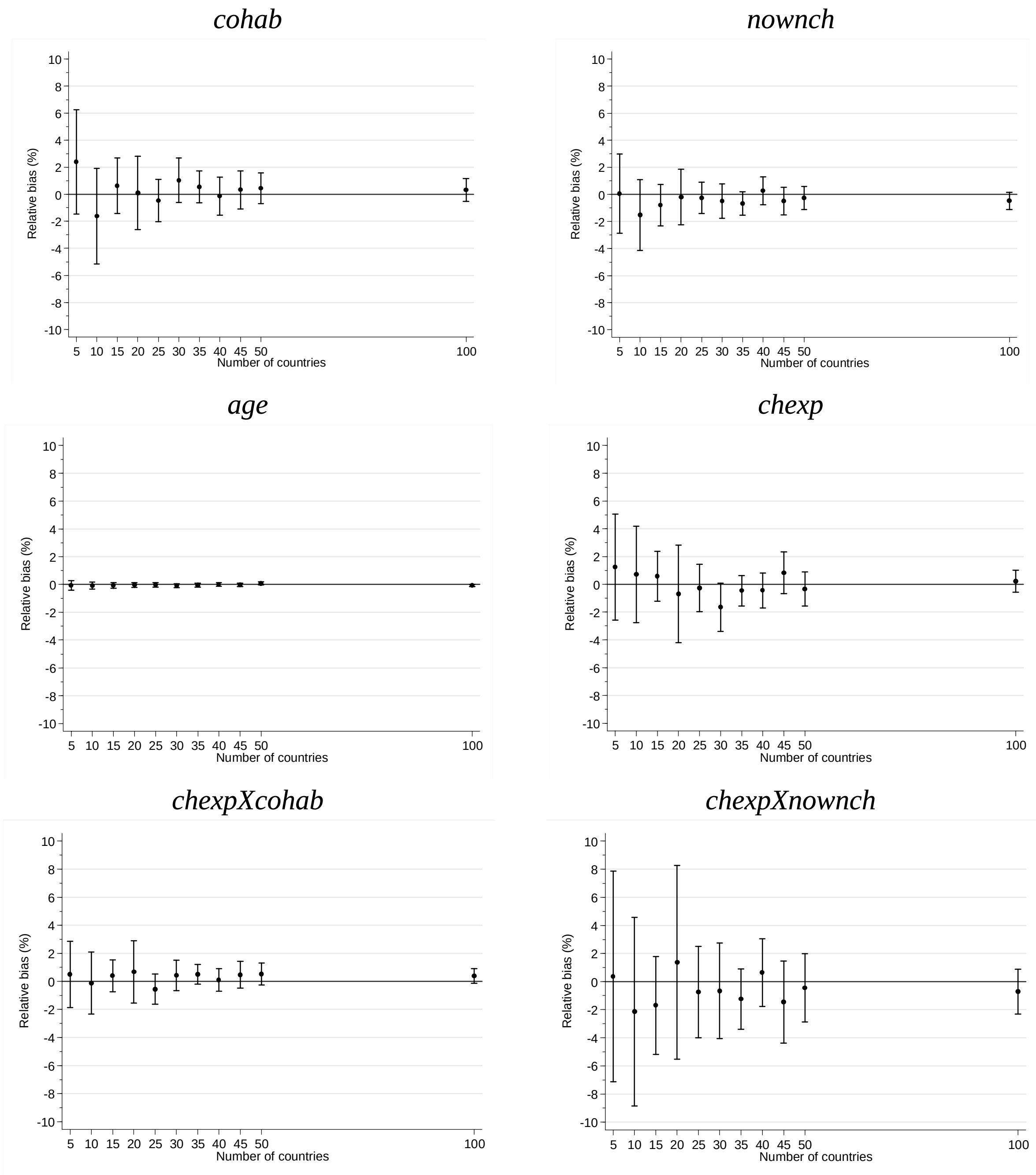
Tables containing the estimates underlying Figures 1–6 in the main body of the paper and Figures S.1–S.4 below, as well as additional estimates, are provided as appendices to Bryan and Jenkins (2013).

Model specifications and parameter values for extended model simulations

Regressor description		Parameter values: Hours	Participation
Fixed effects			
Intercept	*constant*	22	–9.1
Age	age_{ic}	0.8	0.5
Age-squared	$(age_{ic})^2$	–0.01	–0.006
In cohabitating partnership	$cohab_{ic}$	–1	0.02
No. own children present	$nownch_{ic}$	–1.2	–0.27
ISCED category 3	$isced3_{ic}$	0.7	0.7
ISCED category 4	$isced4_{ic}$	1.4	0.9
ISCED categories 5, 6	$isced56_{ic}$	1.6	1.4
Expenditure on children	$chexp_c$	–2.7	0.7
Interaction: cross-level	$chexp_c \times cohab_{ic}$	2.4	0.6
Interaction: cross-level	$chexp_c \times nownch_{ic}$	0.7	–0.1
Random effects variances			
Individual	σ_e (*sig_e*)	9.4	$\pi/\sqrt{3}$
Country	σ_u (*sig_u*)	2.4	0.38
Random slope (*cohab*)	σ_{b3c} (*sig_b3c*)	1.2	0.25
Random slope (*nownch*)	σ_{b4c} (*sig_b4c*)	1.2	0.13
	ICC	0.061	0.042

Notes. See main text for explanation of the models and regressors. The random effects are: an individual-specific error $e_{ic} \sim N(0, \sigma_e^2)$; a random intercept $u_c \sim N(0, \sigma_u^2)$; a random coefficient on $cohab_{ic}$, $b3c \sim N(0, \sigma_{b3c}^2)$; and a random coefficient on $nownch_{ic}$, $b4c \sim N(0, \sigma_{b4c}^2)$. $chexp_c$ is the country-level regressor.

Figure S.1. Relative parameter bias (%): linear model with random intercept, two random slopes, country-level regressor and individual-country interaction (Extended model for hours), selected parameters



Continued overleaf

Figure S.1 (continued). Relative parameter bias (%): linear model with random intercept, two random slopes, country-level regressor and individual-country interaction (Extended model for hours), selected parameters

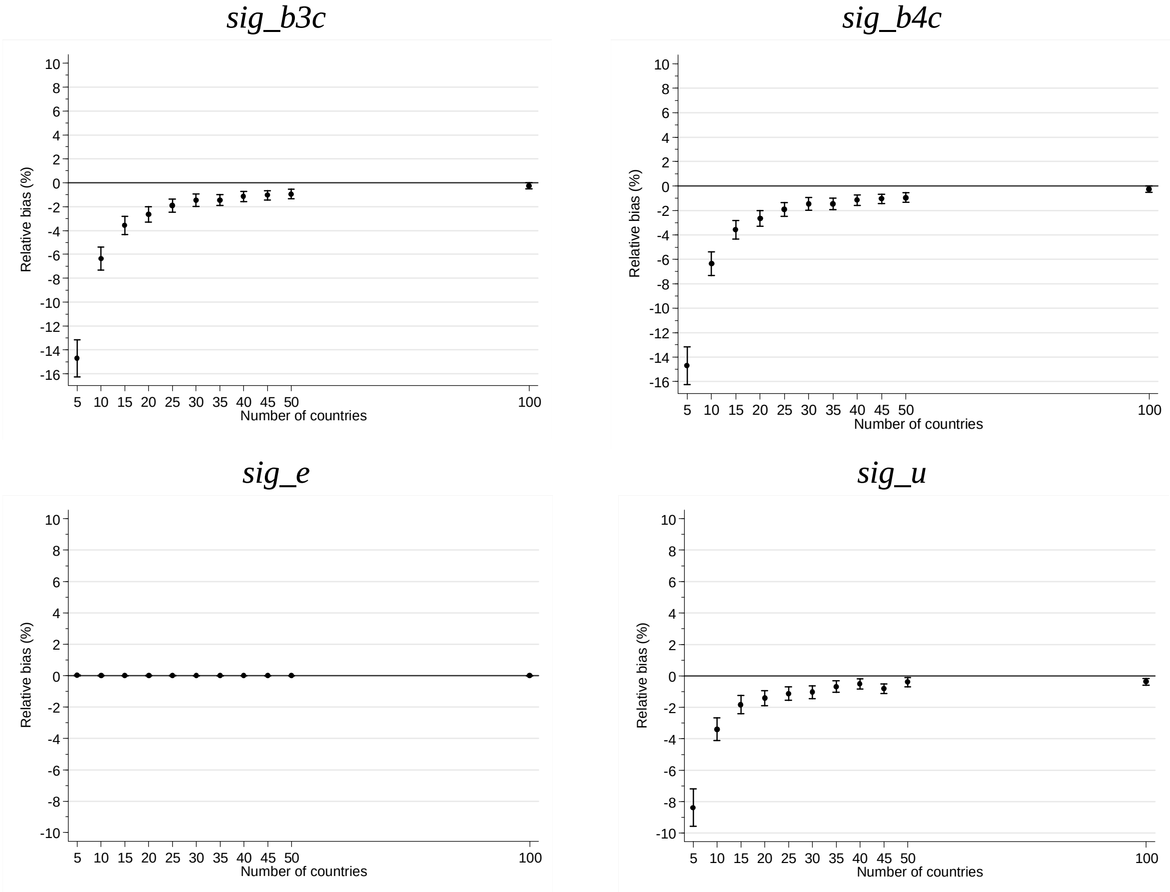
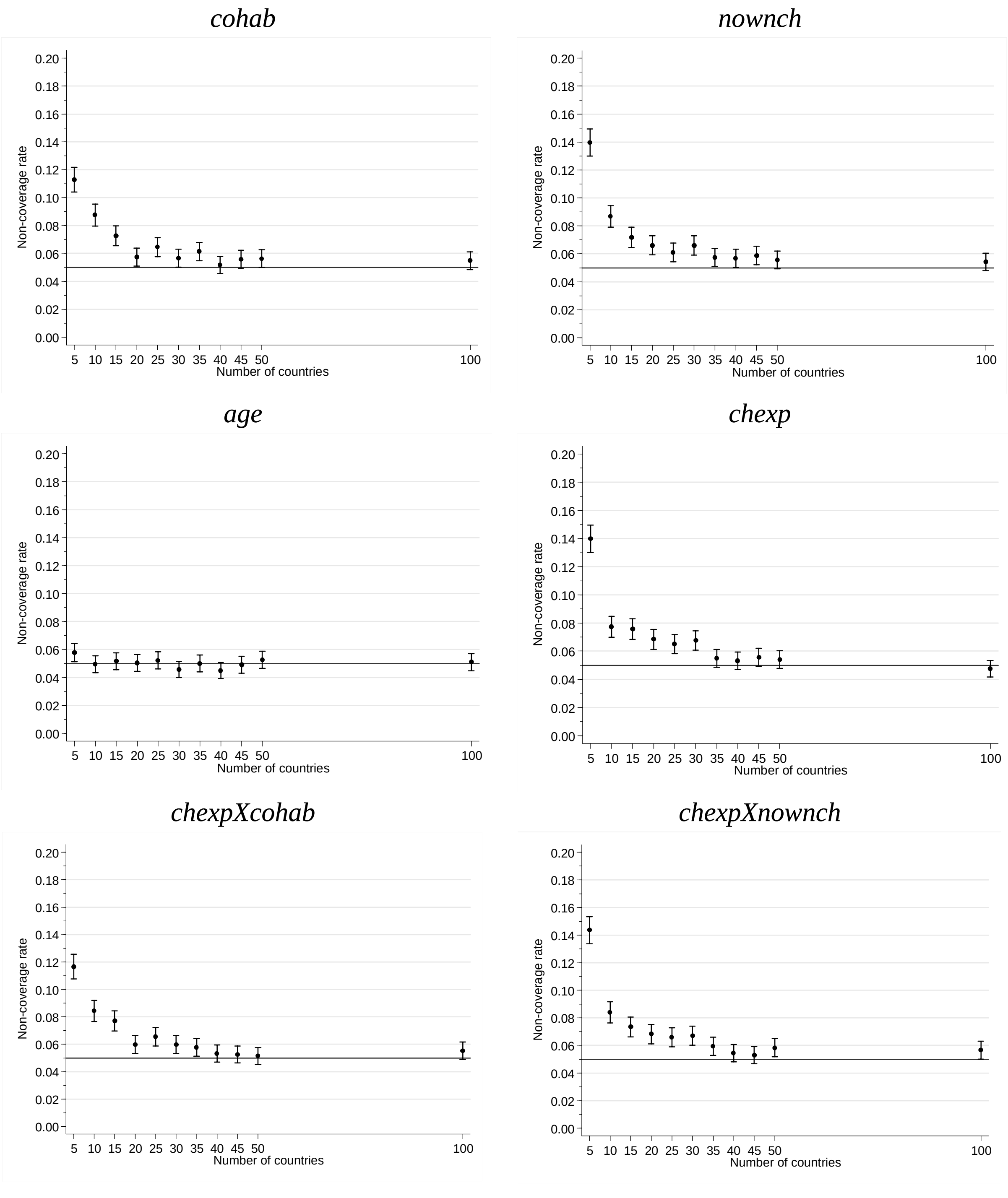


Figure S.2. Non-coverage rate: linear model with random intercept, two random slopes, country-level regressor and individual-country interaction (Extended model for hours), selected parameters



Continued overleaf

Figure S.2 (continued). Non-coverage rate: linear model with random intercept, two random slopes, country-level regressor and individual-country interaction (Extended model for hours), selected parameters

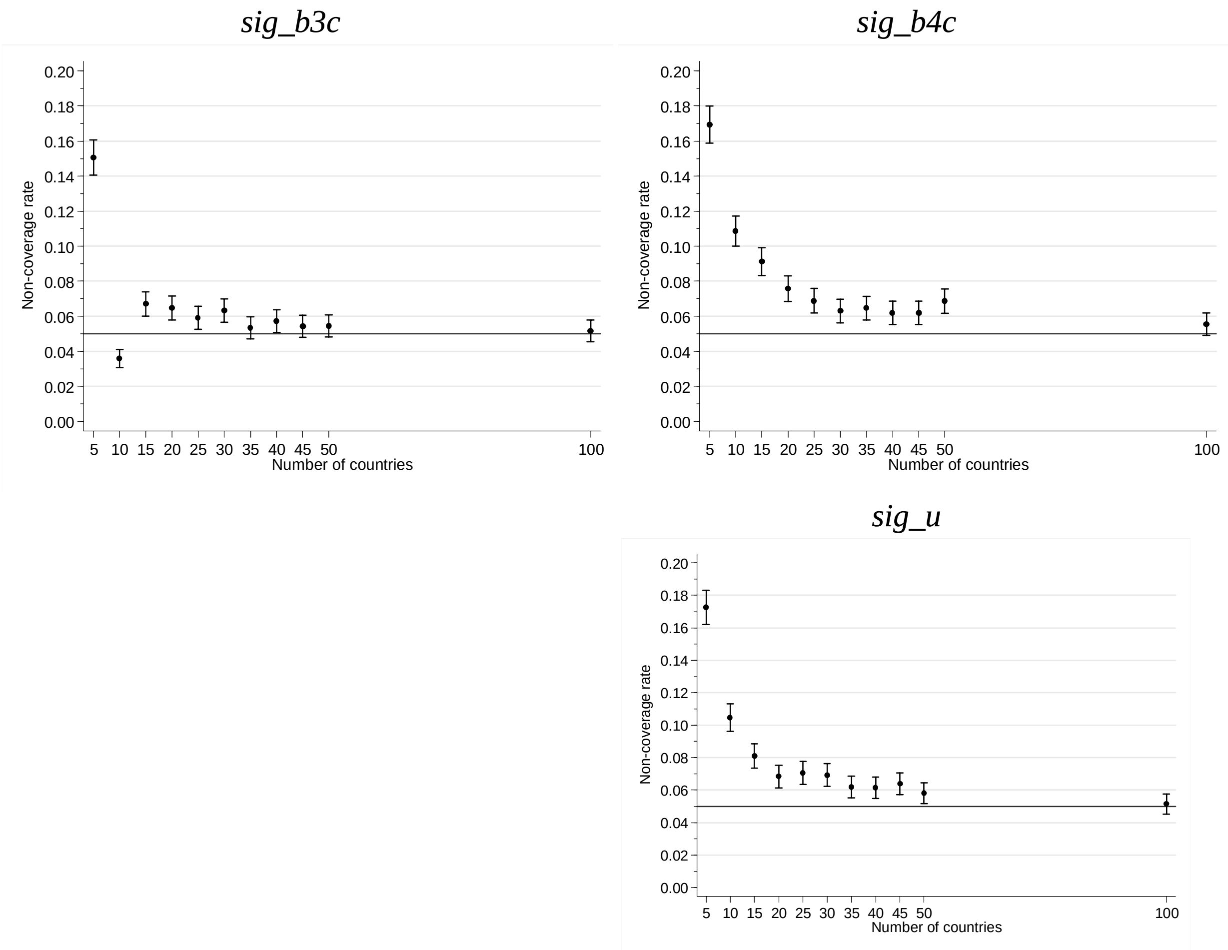
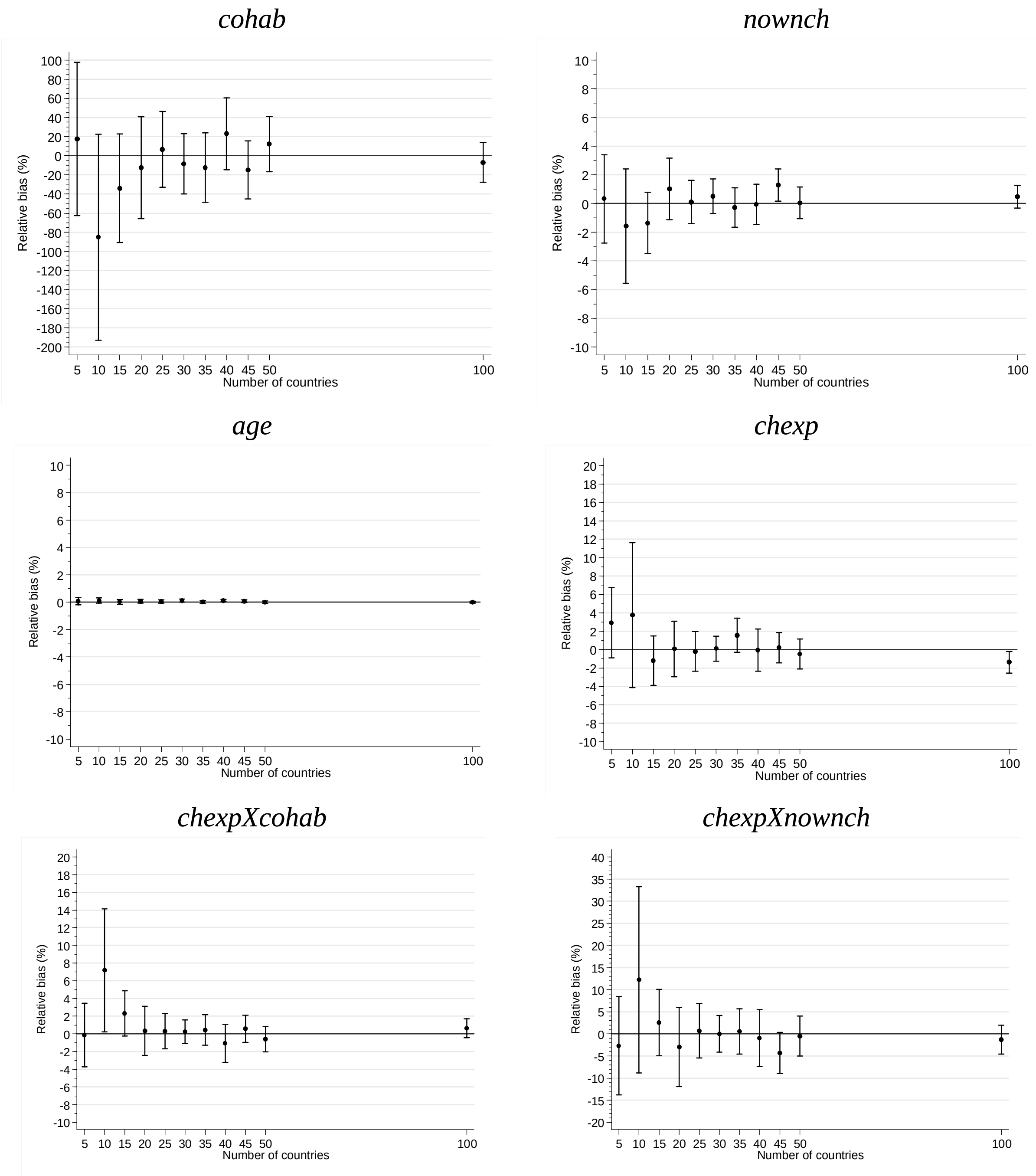


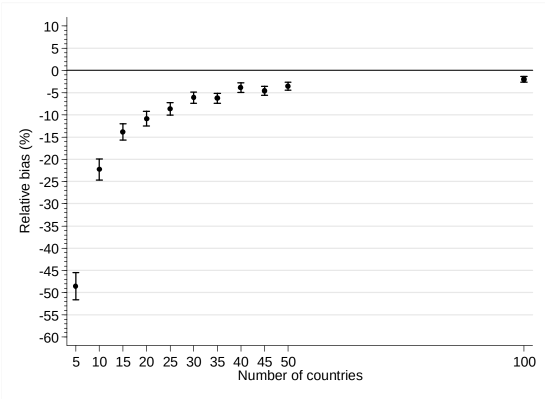
Figure S.3. Relative parameter bias (%): binary logit model with random intercept, two random slopes, country-level regressor and individual-country interaction (Extended model for participation), selected parameters



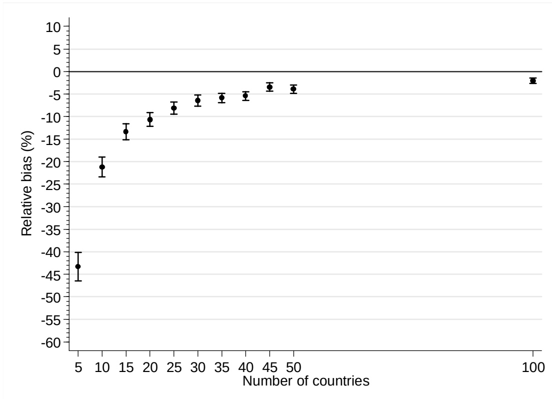
Continued overleaf

Figure S.3 (continued). Relative parameter bias (%): binary logit model with random intercept, two random slopes, country-level regressor and individual-country interaction (Extended model for participation), selected parameters

sig_b3c



sig_b4c



sig_u

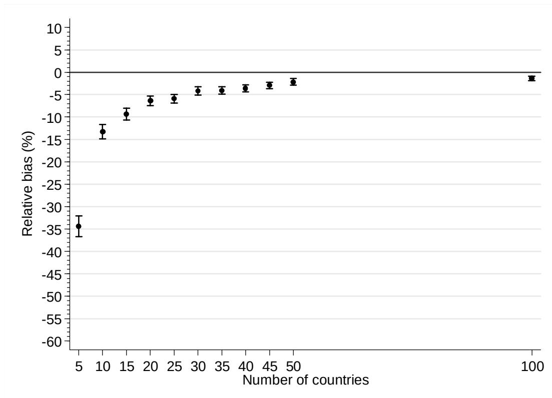
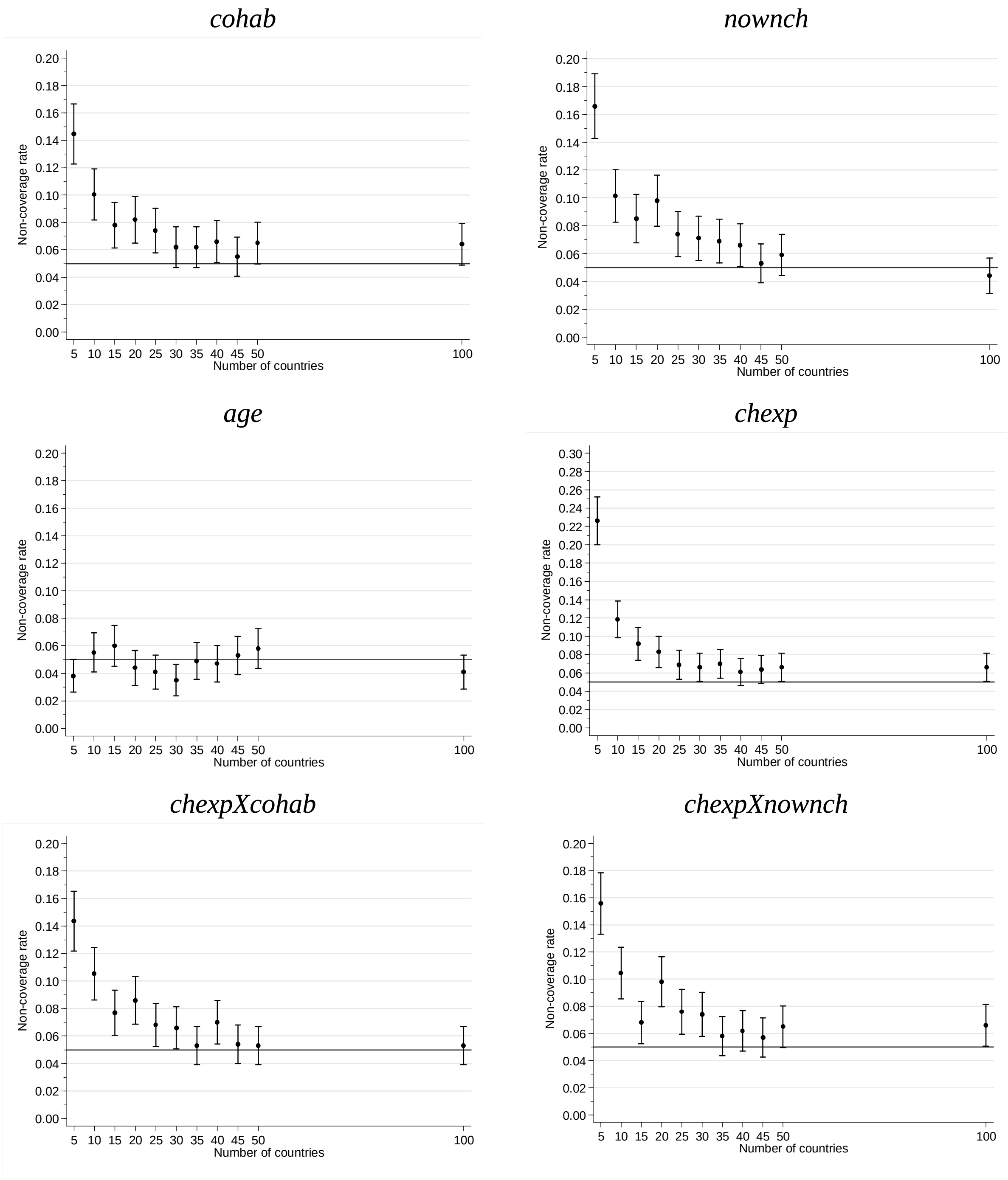


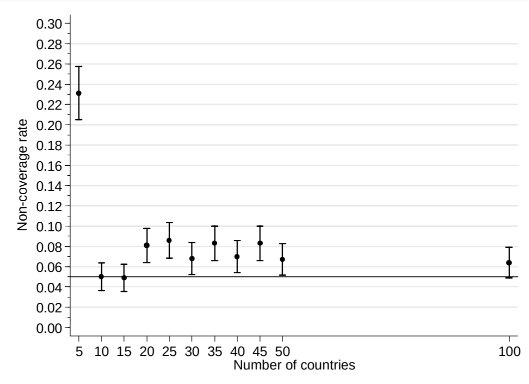
Figure S.4. Non-coverage rate: binary logit model with random intercept, two random slopes, country-level regressor and individual-country interaction (Extended model for participation), selected parameters



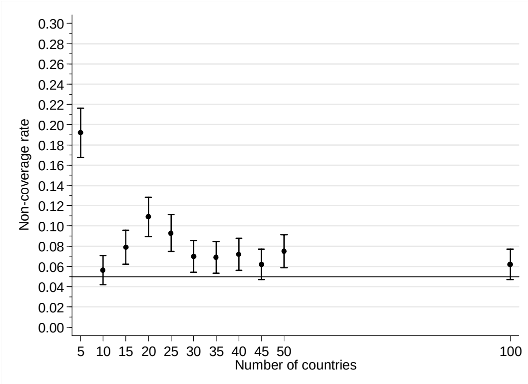
Continued overleaf

Figure S.4 (continued). Non-coverage rate: binary logit model with random intercept, two random slopes, country-level regressor and individual-country interaction (Extended model for participation), selected parameters

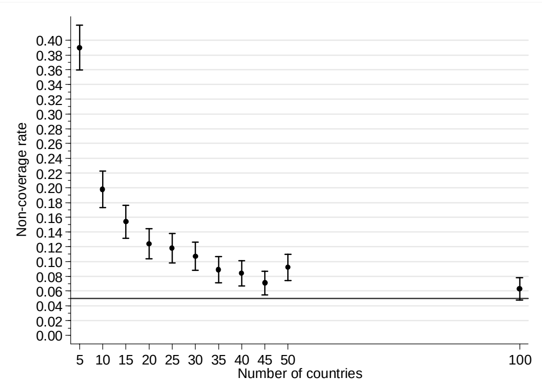
sig_b3c



sig_b4c



sig_u



§8. Illustration of two-step estimation and data visualisation: hours worked and work participation

We present a simple example of two-step estimation using data from EU-SILC.[6] Since the Monte Carlo simulations indicated that we could expect substantial problems with around 10 countries, we randomly selected 10 from the available 26 countries.[7] We estimate models of the form specified in Table 3 of the main text, that is linear models of working hours and non-linear (binary logit) models of work participation. Tables S.9 and S.10 summarise the results and also report estimates of the models produced by standard multilevel (MLM) estimators: REML for hours and ML (with adaptive quadrature) for participation.

As expected, the two-step and MLM estimators produce almost identical results except for the country-level variances and the standard errors of country-level coefficients. The MLM estimates of the country-level random component (*sig_u*) are about 12% smaller than the two-step estimates (and the estimated *ICC* is correspondingly smaller in MLM); and the standard error of the country-level fixed parameter (*chexp*) is about 12% larger using the two-step method than using MLM. The result is that the effect of *chexp* tests as significant at the 5% level in MLM but not in the two-step model.

Figures S.5 and S.6 illustrate the visualisation method of assessing country-level differences. It replaces step 2 of the two-step method with graphs of the estimated country intercepts (from step 1) against childcare spending (these are the data points used in the step 2 regressions reported in Tables S.9 and S.10). As well as showing the overall country-level relationship, visualisation allows us to highlight natural country grouping such as North-West Europe, Southern European, Eastern Europe and Scandinavia (Nordic countries) – which may form clusters worthy of further investigation.

---

[6] We use data from 2007, 4th release, which contains 26 countries: the 27 EU member states excluding Bulgaria, Malta and Romania, plus Iceland and Norway.

[7] The 10 countries are: Denmark, Estonia, Germany, Hungary, Finland, France, Netherlands, Poland, Portugal, United Kingdom.

Table S.9. Model of working hours with country-specific intercepts

	Parameter estimates (standard errors)					
Method	*cohab*$_{ic}$	*nownch*$_{ic}$	*chexp*$_{c}$	*sig_u*	*sig_e*	*ICC*
MLM (REML)	–1.152***	–1.585***	0.424	4.341***	9.880***	0.162
	(0.114)	(0.051)	(4.785)	(0.972)	(0.033)	
Two-step:						
Step 1 (FE)	–1.151***	–1.585***			9.881	
	(0.114)	(0.051)				
Step 2 (OLS)			0.421	4.856		0.195
			(5.349)			

Notes: other explanatory variables are: age, age squared, and highest education level (3 dummy variables); number of observations is 45,464 and number of countries is 10; * significant at 10%; ** significant at 5%; *** significant at 1%; the MLM significance levels are as reported by Stata, and refer to critical values from *z*-distribution; the two-step significance levels refer to critical values from *t*(8)-distribution.

Table S.10. Model of work participation with country-specific intercepts

	Parameter estimates (standard errors)				
Method	*cohab*$_{ic}$	*nownch*$_{ic}$	*chexp*$_{c}$	*sig_u*	*ICC*
MLM logit (FML)	0.072***	-0.288***	0.529**	0.229***	0.016
	(0.021)	(0.010)	(0.255)	(0.052)	
Two-step:					
Step 1	0.071***	-0.288***			
	(0.021)	(0.010)			
Step 2 (OLS)			0.529	0.258	0.020
			(0.284)		

Notes: other explanatory variables are: age, age squared, and highest education level (3 dummy variables); number of observations is 73,169 and number of countries is 10; * significant at 10%; ** significant at 5%; *** significant at 1%; the MLM significance levels are as reported by Stata, and refer to critical values from *z*-distribution; the two-step significance levels refer to critical values from *t*(8)-distribution.

Figure S.5. Model of working hours: country-specific intercepts and childcare/pre-primary spending

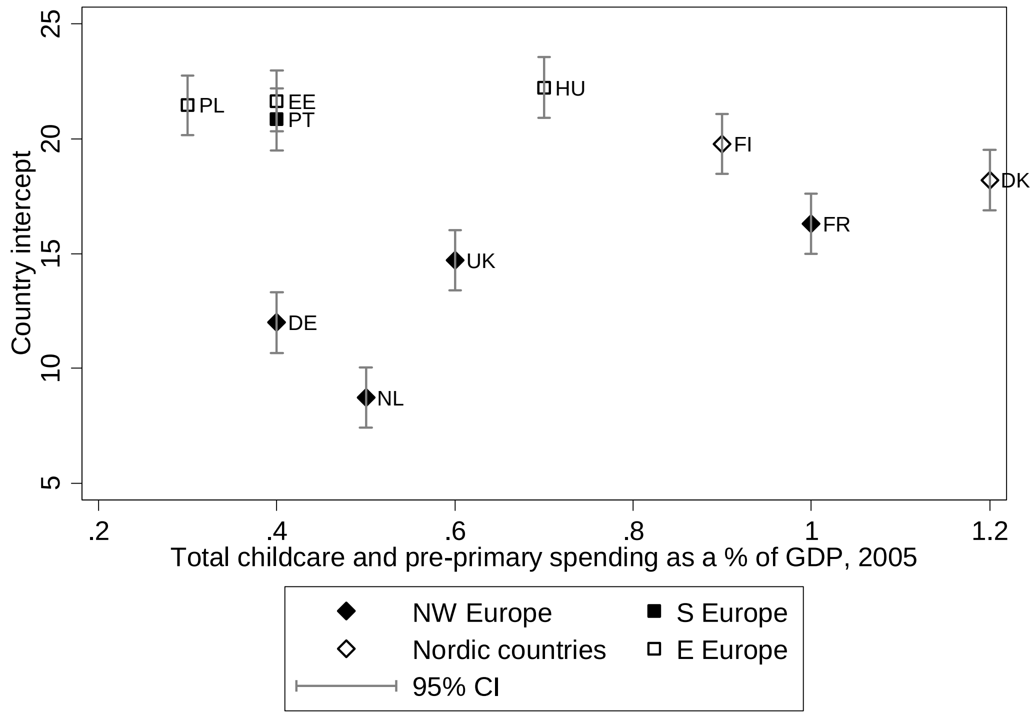
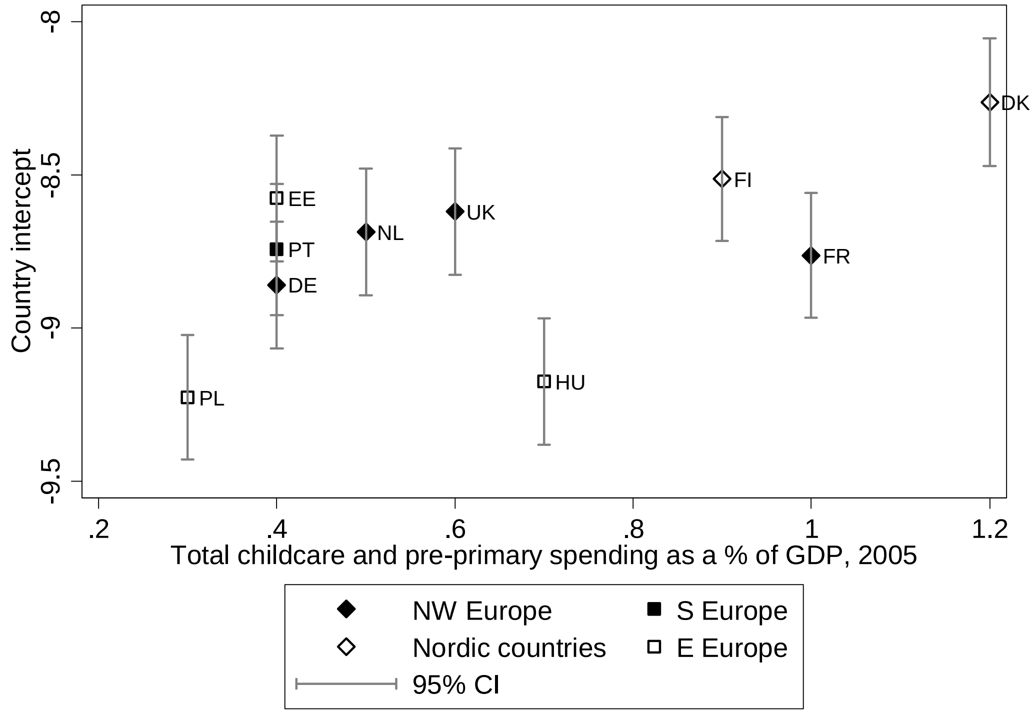


Figure S.6. Model of work participation: country-specific intercepts and childcare/pre-primary spending (10 countries)



§9 Stata do-file code used for running the Monte-Carlo simulations

- **(a) Basic linear model**
- **(b) Extended linear model**
- **(c) Basic logit model**
- (d) **Extended logit model**

(a) Basic linear model (mc_hours_model1_v01b_Nc_C.do)

```
clear all
set more off
version 12

capture log close

local Nc = `1'

local C = `2'

local N = `C' * `Nc'


log using mc_hours_model1_v01b_`Nc'_`C'.log, replace


********** Monte-Carlo exercise **************
*
* Outcome = Number of hours
*
* Model 1: simple random intercept model with country-level predictor
*
* Use arguments to do file to set parameters of truth model: e.g.
*
* do mc_hours_model1_v01b_Nc_C 1000 10
*
*              where "1000" is # persons/country and "10" is # countries
*
* (1) declare model parameter values
* (2) create data set
* (3) set-up MC (10000 reps)
* (4) post-fitting processing of results
*
*********************************************

*********************************************
* (1) declare model parameter values
*********************************************

di "Number of persons per country, N_c = " `Nc'
di "Number of countries, C   = " `C'
di "Total sample size (Nc x C), N   = " `N'

/*

Woman aged 18-64 with positive work hours

Model 3 for hours:

Hours_ic = b0
       + b1 * age_ic
       + b2 * age-squared_ic
       + b3 * cohab_ic
       + b4 * nownch_ic
       + b5 * isced3_ic
       + b6 * isced4_ic
       + b7 * isced56_ic
       + c1 * chexp_c              <-- country=level
       + u_c
       + e_ic
```

```
        u_c  ~ N(0, sig_u^2)
        e_ic ~ N(0, sig_e^2)
        cov(u_c, e_ic) = 0


*/

local b0 = 22
local b1 = 0.8
local b2 = -0.01
local b3 = -1
local b4 = -1.2
local b5 = 0.7
local b6 = 1.4
local b7 = 1.6
local c1 = -0.23

local sig_u = 3.5
local sig_e = 9.5



local icc = (`sig_u')^2 / ( (`sig_u')^2 + (`sig_e')^2  )
di "ICC = " `icc'

************************************************
* (2) create data set (using summary.log stats)
************************************************

set obs `N'

set seed 123456789

* "fixed" part of model (X, Z assumed same/fixed across experiments)

ge id = _n
ge country = autocode(id, `C', 1, `N')
egen country_id = group(country)
bys country_id: ge tag = _n == 1
compress

* age distribution
* Singh-Maddala parameters: a =

local a = 4.35
local b = 250
local q = 1656
ge age = `b' * ( (1/runiform())^(1/`q') - 1 )^(1/`a')
sum age
replace age = . if (age < 18 | age > 64)
sum age
count if missing(age)
local nm = r(N)
while `nm' > 0 {
        replace age = `b' * ( (1/runiform())^(1/`q') - 1 )^(1/`a') if missing(age)
        sum age
        replace age = . if (age < 18 | age > 64)
        count if missing(age)
        local nm = r(N)
}

replace age = round(age)

ge agegp = .
replace agegp = 1 if age >= 18 & age < 25
replace agegp = 2 if age >= 25 & age < 40
replace agegp = 3 if age >= 40 & age < 50
replace agegp = 4 if age >= 50 & age < 60
replace agegp = 5 if age >= 60 & age < .
lab def agegp 1 "18-24" 2 "25-39" 3 "40-49" 4 "50-59" 5 "60-64"
lab val agegp agegp
ta agegp

ge agesq = age^2

ge cohab = .
ge nownch = .
```

```
ge isced012 = 0
ge isced3 = 0
ge isced4 = 0
ge isced56 = 0

lab var cohab "Partnership status
lab def cohab 0 "No partner" 1 "Has parter"
lab val cohab cohab
lab var nownch "# kids"
lab def nownch 0 "0" 1 "1" 2 "2" 3 "3+"
lab val nownch nownch

ge type = .
lab var type "Characteristics combination"
lab def type ///
           1 "Not marr 0 isced0,1,2" ///
           2 "Not marr 0 isced3" ///
           3 "Not marr 0 isced4" ///
           4 "Not marr 0 isced5,6" ///
           5 "Not marr 1 isced0,1,2" ///
           6 "Not marr 1 isced3" ///
           7 "Not marr 1 isced4" ///
           8 "Not marr 1 isced5,6" ///
           9 "Not marr 2 isced0,1,2" ///
          10 "Not marr 2 isced3" ///
          11 "Not marr 2 isced4" ///
          12 "Not marr 2 isced5,6" ///
          13 "Not marr 3+ isced0,1,2" ///
          14 "Not marr 3+ isced3" ///
          15 "Not marr 3+ isced4" ///
          16 "Not marr 3+ isced5,6" ///
          17 "Marr 0 isced0,1,2" ///
          18 "Marr 0 isced3" ///
          19 "Marr 0 isced4" ///
          20 "Marr 0 isced5,6" ///
          21 "Marr 1 isced0,1,2" ///
          22 "Marr 1 isced3" ///
          23 "Marr 1 isced4" ///
          24 "Marr 1 isced5,6" ///
          25 "Marr 2 isced0,1,2" ///
          26 "Marr 2 isced3" ///
          27 "Marr 2 isced4" ///
          28 "Marr 2 isced5,6" ///
          29 "Marr 3+ isced0,1,2" ///
          30 "Marr 3+ isced3" ///
          31 "Marr 3+ isced4" ///
          32 "Marr 3+ isced5,6"
lab val type type

ge double u = 100 * runiform()

replace type =  1 if agegp == 1 & u <= 14.42
replace type =  2 if agegp == 1 & u > 14.42  &  u <= 58.23
replace type =  3 if agegp == 1 & u > 58.23  &  u <= 62.91
replace type =  4 if agegp == 1 & u > 62.91  &  u <= 74.45
replace type =  5 if agegp == 1 & u > 74.45  &  u <= 75.07
replace type =  6 if agegp == 1 & u > 75.07  &  u <= 76.08
replace type =  7 if agegp == 1 & u > 76.08  &  u <= 76.12
replace type =  8 if agegp == 1 & u > 76.12  &  u <= 76.24
replace type =  9 if agegp == 1 & u > 76.24  &  u <= 76.29
replace type = 10 if agegp == 1 & u > 76.29  &  u <= 76.37
replace type = 11 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 12 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 13 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 14 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 15 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 16 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 17 if agegp == 1 & u > 76.37  &  u <= 78.77
replace type = 18 if agegp == 1 & u > 78.77  &  u <= 89.45
replace type = 19 if agegp == 1 & u > 89.45  &  u <= 90.30
replace type = 20 if agegp == 1 & u > 90.30  &  u <= 94.27
replace type = 21 if agegp == 1 & u > 94.27  &  u <= 95.41
replace type = 22 if agegp == 1 & u > 95.41  &  u <= 98.28
replace type = 23 if agegp == 1 & u > 98.28  &  u <= 98.44
replace type = 24 if agegp == 1 & u > 98.44  &  u <= 98.84
replace type = 25 if agegp == 1 & u > 98.84  &  u <= 99.21
replace type = 26 if agegp == 1 & u > 99.21  &  u <= 99.86
```

```
replace type = 27 if agegp == 1 & u > 99.86  &  u <= 99.89
replace type = 28 if agegp == 1 & u > 99.89  &  u <= 99.92
replace type = 29 if agegp == 1 & u > 99.92  &  u <= 99.95
replace type = 30 if agegp == 1 & u > 99.95  &  u <= 100.00
replace type = 31 if agegp == 1 & u > 100.00                      // # cases = 0
replace type = 32 if agegp == 1 & u > 100.00                      // # cases = 0

replace type =  1 if agegp == 2 & u <= 1.96
replace type =  2 if agegp == 2 & u >  1.96  &  u <= 10.21
replace type =  3 if agegp == 2 & u > 10.21  &  u <= 11.72
replace type =  4 if agegp == 2 & u > 11.72  &  u <= 22.25
replace type =  5 if agegp == 2 & u > 22.25  &  u <= 22.92
replace type =  6 if agegp == 2 & u > 22.92  &  u <= 25.31
replace type =  7 if agegp == 2 & u > 25.31  &  u <= 25.64
replace type =  8 if agegp == 2 & u > 25.64  &  u <= 26.93
replace type =  9 if agegp == 2 & u > 26.93  &  u <= 27.41
replace type = 10 if agegp == 2 & u > 27.41  &  u <= 28.77
replace type = 11 if agegp == 2 & u > 28.77  &  u <= 28.91
replace type = 12 if agegp == 2 & u > 28.91  &  u <= 29.44
replace type = 13 if agegp == 2 & u > 29.44  &  u <= 29.67
replace type = 14 if agegp == 2 & u > 29.67  &  u <= 30.00
replace type = 15 if agegp == 2 & u > 30.00  &  u <= 30.01
replace type = 16 if agegp == 2 & u > 30.01  &  u <= 30.12
replace type = 17 if agegp == 2 & u > 30.12  &  u <= 31.43
replace type = 18 if agegp == 2 & u > 31.43  &  u <= 37.24
replace type = 19 if agegp == 2 & u > 37.24  &  u <= 38.09
replace type = 20 if agegp == 2 & u > 38.09  &  u <= 46.16
replace type = 21 if agegp == 2 & u > 46.16  &  u <= 48.43
replace type = 22 if agegp == 2 & u > 48.43  &  u <= 57.34
replace type = 23 if agegp == 2 & u > 57.34  &  u <= 58.49
replace type = 24 if agegp == 2 & u > 58.49  &  u <= 65.70
replace type = 25 if agegp == 2 & u > 65.70  &  u <= 68.83
replace type = 26 if agegp == 2 & u > 68.83  &  u <= 81.70
replace type = 27 if agegp == 2 & u > 81.70  &  u <= 83.15
replace type = 28 if agegp == 2 & u > 83.15  &  u <= 92.10
replace type = 29 if agegp == 2 & u > 92.10  &  u <= 93.43
replace type = 30 if agegp == 2 & u > 93.43  &  u <= 97.19
replace type = 31 if agegp == 2 & u > 97.19  &  u <= 97.52
replace type = 32 if agegp == 2 & u > 97.52

replace type =  1 if agegp == 3 & u <= 1.33
replace type =  2 if agegp == 3 & u >  1.33  &  u <=  4.46
replace type =  3 if agegp == 3 & u >  4.46  &  u <=  5.05
replace type =  4 if agegp == 3 & u >  5.05  &  u <=  7.83
replace type =  5 if agegp == 3 & u >  7.83  &  u <=  8.81
replace type =  6 if agegp == 3 & u >  8.81  &  u <= 11.63
replace type =  7 if agegp == 3 & u > 11.63  &  u <= 12.16
replace type =  8 if agegp == 3 & u > 12.16  &  u <= 14.04
replace type =  9 if agegp == 3 & u > 14.04  &  u <= 14.82
replace type = 10 if agegp == 3 & u > 14.82  &  u <= 17.13
replace type = 11 if agegp == 3 & u > 17.13  &  u <= 17.50
replace type = 12 if agegp == 3 & u > 17.50  &  u <= 19.00
replace type = 13 if agegp == 3 & u > 19.00  &  u <= 19.30
replace type = 14 if agegp == 3 & u > 19.30  &  u <= 20.00
replace type = 15 if agegp == 3 & u > 20.00  &  u <= 20.09
replace type = 16 if agegp == 3 & u > 20.09  &  u <= 20.59
replace type = 17 if agegp == 3 & u > 20.59  &  u <= 22.54
replace type = 18 if agegp == 3 & u > 22.54  &  u <= 27.32
replace type = 19 if agegp == 3 & u > 27.32  &  u <= 27.99
replace type = 20 if agegp == 3 & u > 27.99  &  u <= 30.90
replace type = 21 if agegp == 3 & u > 30.90  &  u <= 34.62
replace type = 22 if agegp == 3 & u > 34.62  &  u <= 44.00
replace type = 23 if agegp == 3 & u > 44.00  &  u <= 45.29
replace type = 24 if agegp == 3 & u > 45.29  &  u <= 50.88
replace type = 25 if agegp == 3 & u > 50.88  &  u <= 56.49
replace type = 26 if agegp == 3 & u > 56.49  &  u <= 72.54
replace type = 27 if agegp == 3 & u > 72.54  &  u <= 74.33
replace type = 28 if agegp == 3 & u > 74.33  &  u <= 85.71
replace type = 29 if agegp == 3 & u > 85.71  &  u <= 87.94
replace type = 30 if agegp == 3 & u > 87.94  &  u <= 94.45
replace type = 31 if agegp == 3 & u > 94.45  &  u <= 95.15
replace type = 32 if agegp == 3 & u > 95.15

replace type =  1 if agegp == 4 & u <= 2.78
replace type =  2 if agegp == 4 & u >  2.78  &  u <=  8.08
replace type =  3 if agegp == 4 & u >  8.08  &  u <=  8.84
replace type =  4 if agegp == 4 & u >  8.84  &  u <= 13.28
```

```
replace type =  5 if agegp == 4 & u > 13.28  &  u <= 14.83
replace type =  6 if agegp == 4 & u > 14.83  &  u <= 17.96
replace type =  7 if agegp == 4 & u > 17.96  &  u <= 18.36
replace type =  8 if agegp == 4 & u > 18.36  &  u <= 20.37
replace type =  9 if agegp == 4 & u > 20.37  &  u <= 20.97
replace type = 10 if agegp == 4 & u > 20.97  &  u <= 22.05
replace type = 11 if agegp == 4 & u > 22.05  &  u <= 22.21
replace type = 12 if agegp == 4 & u > 22.21  &  u <= 22.97
replace type = 13 if agegp == 4 & u > 22.97  &  u <= 23.12
replace type = 14 if agegp == 4 & u > 23.12  &  u <= 23.32
replace type = 15 if agegp == 4 & u > 23.32  &  u <= 23.34
replace type = 16 if agegp == 4 & u > 23.34  &  u <= 23.48
replace type = 17 if agegp == 4 & u > 23.48  &  u <= 32.15
replace type = 18 if agegp == 4 & u > 32.15  &  u <= 48.05
replace type = 19 if agegp == 4 & u > 48.05  &  u <= 49.82
replace type = 20 if agegp == 4 & u > 49.82  &  u <= 59.33
replace type = 21 if agegp == 4 & u > 59.33  &  u <= 64.92
replace type = 22 if agegp == 4 & u > 64.92  &  u <= 75.12
replace type = 23 if agegp == 4 & u > 75.12  &  u <= 76.31
replace type = 24 if agegp == 4 & u > 76.31  &  u <= 82.55
replace type = 25 if agegp == 4 & u > 82.55  &  u <= 85.74
replace type = 26 if agegp == 4 & u > 85.74  &  u <= 91.31
replace type = 27 if agegp == 4 & u > 91.31  &  u <= 91.88
replace type = 28 if agegp == 4 & u > 91.88  &  u <= 96.37
replace type = 29 if agegp == 4 & u > 96.37  &  u <= 97.20
replace type = 30 if agegp == 4 & u > 97.20  &  u <= 98.52
replace type = 31 if agegp == 4 & u > 98.52  &  u <= 98.72
replace type = 32 if agegp == 4 & u > 98.72

replace type =  1 if agegp == 5 & u <= 5.66
replace type =  2 if agegp == 5 & u >  5.66  &  u <= 12.73
replace type =  3 if agegp == 5 & u > 12.73  &  u <= 13.92
replace type =  4 if agegp == 5 & u > 13.92  &  u <= 20.87
replace type =  5 if agegp == 5 & u > 20.87  &  u <= 23.05
replace type =  6 if agegp == 5 & u > 23.05  &  u <= 24.81
replace type =  7 if agegp == 5 & u > 24.81  &  u <= 25.17
replace type =  8 if agegp == 5 & u > 25.17  &  u <= 26.72
replace type =  9 if agegp == 5 & u > 26.72  &  u <= 27.08
replace type = 10 if agegp == 5 & u > 27.08  &  u <= 27.38
replace type = 11 if agegp == 5 & u > 27.38  &  u <= 27.41
replace type = 12 if agegp == 5 & u > 27.41  &  u <= 27.80
replace type = 13 if agegp == 5 & u > 27.80  &  u <= 27.89
replace type = 14 if agegp == 5 & u > 27.89  &  u <= 27.92
replace type = 15 if agegp == 5 & u > 27.92  &  u <= 27.92   // # cases = 0
replace type = 16 if agegp == 5 & u > 27.92  &  u <= 27.98
replace type = 17 if agegp == 5 & u > 27.98  &  u <= 44.53
replace type = 18 if agegp == 5 & u > 44.53  &  u <= 65.64
replace type = 19 if agegp == 5 & u > 65.64  &  u <= 68.11
replace type = 20 if agegp == 5 & u > 68.11  &  u <= 84.09
replace type = 21 if agegp == 5 & u > 84.09  &  u <= 89.78
replace type = 22 if agegp == 5 & u > 89.78  &  u <= 93.48
replace type = 23 if agegp == 5 & u > 93.48  &  u <= 94.02
replace type = 24 if agegp == 5 & u > 94.02  &  u <= 97.12
replace type = 25 if agegp == 5 & u > 97.12  &  u <= 98.10
replace type = 26 if agegp == 5 & u > 98.10  &  u <= 98.82
replace type = 27 if agegp == 5 & u > 98.82  &  u <= 98.82  // # cases = 0
replace type = 28 if agegp == 5 & u > 98.82  &  u <= 99.42
replace type = 29 if agegp == 5 & u > 99.42  &  u <= 99.66
replace type = 30 if agegp == 5 & u > 99.66  &  u <= 99.84
replace type = 31 if agegp == 5 & u > 99.84  &  u <= 99.90
replace type = 32 if agegp == 5 & u > 99.90

ta type agegp, col missing

replace cohab = 0 if inrange(type, 1, 16)
replace cohab = 1 if inrange(type, 17, 32)

replace nownch = 0 if inrange(type, 1, 4) | inrange(type, 17, 20)
replace nownch = 1 if inrange(type, 5, 8) | inrange(type, 21, 24)
replace nownch = 2 if inrange(type, 9, 12) | inrange(type, 25, 28)
replace nownch = 3 if inrange(type, 13, 16) | inrange(type, 29, 32)

replace isced012 = 1 if inlist(type, 1, 5, 9, 13, 17, 21, 25, 29)
replace isced3   = 1 if inlist(type, 2, 6, 10, 14, 18, 22, 26, 30)
replace isced4          = 1 if inlist(type, 3, 7, 11, 15, 19, 23, 27, 31)
replace isced56  = 1 if inlist(type, 4, 8, 12, 16, 20, 24, 28, 32)
```

```
gen edlevel = .
replace edlevel = 1 if isced012 == 1
replace edlevel = 2 if isced3 == 1
replace edlevel = 3 if isced4 == 1
replace edlevel = 4 if isced56 == 1
lab def edlevel 1 "isced0,1,2" 2 "isced3" 3 "isced4" 4 "isced5,6"
lab val edlevel edlevel


* lognormal distribution for child exp. Parameters: m = -0.665, v = 0.535

local m = -0.665
local v = 0.535

ge chexp = exp( `m' + `v' * invnormal( runiform() ) ) if tag
lab var chexp "child care expend, % GDP"
bys country_id: replace chexp = chexp[1]

ge chexpXcohab = chexp * cohab
ge chexpXnownch = chexp * nownch

lab var chexpXcohab "chexp,cohab interaction"
lab var chexpXnownch "chexp,nownch interaction"

compress

ge fixed = `b0'                             ///
       + `b1' * age                         ///
       + `b2' * agesq                       ///
       + `b3' * cohab                       ///
       + `b4' * nownch                      ///
       + `b5' * isced3                      ///
       + `b6' * isced4                      ///
       + `b7' * isced56                     ///
       + `c1' * chexp


ta country_id, sum(age)
ta country_id, sum(isced012)
ta country_id, sum(isced3)
ta country_id, sum(isced4)
ta country_id, sum(isced56)
ta country_id, sum(cohab)
ta country_id, sum(nownch)
ta country_id, sum(chexp)

de, fullnames
su
save mc_hours_model1_v01b_`Nc'_`C'_data, replace


************************************************
* (3) set-up MC
************************************************



cap program drop mc_silc
program define mc_silc

       version 11
       args sig_e sig_u

       capture drop y u_c e_ic

       gen e_ic =  rnormal(0,`sig_e')

       gen u_c = rnormal(0, `sig_u') if tag
       bys country_id : replace u_c = u_c[1]


       gen y = fixed + u_c + e_ic

              // default estimation method is REML, cov structure independent
       xtmixed y age agesq cohab nownch isced3 isced4 isced56 ///
                chexp  || country_id:  , nolog  iter(250)
```

```
end

di "Time is: " c(current_time) " on " c(current_date)
simulate _b _se converged = e(converged) logRLL = e(ll) ///
	, reps(10000) saving(mc_hours_model1_v01b_`Nc'_`C'_output.dta, replace double)  ///
	: mc_silc (`sig_e') (`sig_u')
di "Time is: " c(current_time) " on " c(current_date)


de, fullnames
ds, varwidth(20)
su

*********************************************
* (4) post-fitting processing of results and saving data
*********************************************


* convert back to original metric; calculate summary stats


* quietly {

	rename _eq7_converged  converged
	rename _eq7_logRLL logRLL

	renpfix y_

	ge sig_e = exp(lnsig_e_b_cons)
	ge sig_e_se = exp(lnsig_e_b_cons)*lnsig_e_se_cons

	ge sig_u = exp(lns1_1_1_b_cons)
	ge sig_u_se = sig_u * lns1_1_1_se_cons

* absolute bias
	ge b_cons_abias = b_cons - `b0'
	lab var b_cons_abias "Absolute bias b0 (cons)"

	ge b_age_abias = b_age - `b1'
	lab var b_age_abias "Absolute bias b1 (age)"

	ge b_agesq_abias = b_agesq - `b2'
	lab var b_agesq_abias "Absolute bias b2 (agesq)"

	ge b_cohab_abias = b_cohab - `b3'
	lab var b_cohab_abias "Absolute bias b3 (cohab)"

	ge b_nownch_abias = b_nownch - `b4'
	lab var b_nownch_abias "Absolute bias b4 (nownch)"

	ge b_isced3_abias = b_isced3 - `b5'
	lab var b_isced3_abias "Absolute bias b5 (isced3)"

	ge b_isced4_abias = b_isced4 - `b6'
	lab var b_isced4_abias "Absolute bias b6 (isced4)"

	ge b_isced56_abias = b_isced56 - `b7'
	lab var b_isced56_abias "Absolute bias b7 (isced56)"

	ge c_chexp_abias = b_chexp - `c1'
	lab var c_chexp_abias "Absolute bias c1 (chexp)"

	ge sig_u_abias = sig_u - `sig_u'
	lab var sig_u_abias "Absolute bias sig_u"

	ge sig_e_abias  = sig_e - `sig_e'
	lab var sig_e_abias "Absolute bias sig_e"


* relative bias

	ge b_cons_rbias = 100*(b_cons/`b0' - 1)
	lab var b_cons_rbias "% Relative bias b0 (cons)"

	ge b_age_rbias = 100*(b_age/`b1' - 1)
	lab var b_age_rbias "% Relative bias b1 (age)"
```

```
	ge b_agesq_rbias = 100*(b_agesq/`b2' - 1)
	lab var b_agesq_rbias "% Relative bias b2 (agesq)"

	ge b_cohab_rbias = 100*(b_cohab/`b3' - 1)
	lab var b_cohab_rbias "% Relative bias b3 (cohab)"

	ge b_nownch_rbias = 100*(b_nownch/`b4' - 1)
	lab var b_nownch_rbias "% Relative bias b4 (nownch)"

	ge b_isced3_rbias = 100*(b_isced3/`b5' - 1)
	lab var b_isced3_rbias "% Relative bias b5 (isced3)"

	ge b_isced4_rbias = 100*(b_isced4/`b6' - 1)
	lab var b_isced4_rbias "% Relative bias b6 (isced4)"

	ge b_isced56_rbias = 100*(b_isced56/`b7' - 1)
	lab var b_isced56_rbias "% Relative bias b7 (isced56)"

	ge c_chexp_rbias = 100*(b_chexp/`c1' - 1)
	lab var c_chexp_rbias "% Relative bias c1 (chexp)"

	ge sig_u_rbias = 100*(sig_u / `sig_u' - 1)
	lab var sig_u_rbias "% Relative bias sig_u"

	ge sig_e_rbias = 100*(sig_e / `sig_e' - 1)
	lab var sig_e_rbias "% Relative bias sig_e"



* non-coverage

	ge b0_lb = b_cons + se_cons*invnormal( (1-.95)/2 )
	ge b0_ub = b_cons - se_cons*invnormal( (1-.95)/2 )
	lab var b0_lb "lb 95% CI, b0 (cons)"
	lab var b0_ub "ub 95% CI, b0 (cons)"
	ge b_cons_noncover = 0
	replace b_cons_noncover = 1 if (`b0' < b0_lb | `b0' > b0_ub)
	lab var b_cons_noncover "1: true b0 not in 95%CI"

	ge b1_lb = b_age + se_age*invnormal( (1-.95)/2 )
	ge b1_ub = b_age - se_age*invnormal( (1-.95)/2 )
	lab var b1_lb "lb 95% CI, b1 (age)"
	lab var b1_ub "ub 95% CI, b1 (age)"
	ge b_age_noncover = 0
	replace b_age_noncover = 1 if (`b1' < b1_lb | `b1' > b1_ub)
	lab var b_age_noncover "1: true b1 not in 95%CI"

	ge b2_lb = b_agesq + se_agesq*invnormal( (1-.95)/2 )
	ge b2_ub = b_agesq - se_agesq*invnormal( (1-.95)/2 )
	lab var b2_lb "lb 95% CI, b2 (agesq)"
	lab var b2_ub "ub 95% CI, b2 (agesq)"
	ge b_agesq_noncover = 0
	replace b_agesq_noncover = 1 if (`b2' < b2_lb | `b2' > b2_ub)
	lab var b_agesq_noncover "1: true b2 not in 95%CI"

	ge b3_lb = b_cohab + se_cohab*invnormal( (1-.95)/2 )
	ge b3_ub = b_cohab - se_cohab*invnormal( (1-.95)/2 )
	lab var b3_lb "lb 95% CI, b0 (cohab)"
	lab var b3_ub "ub 95% CI, b0 (cohab)"
	ge b_cohab_noncover = 0
	replace b_cohab_noncover = 1 if (`b3' < b3_lb | `b3' > b3_ub)
	lab var b_cohab_noncover "1: true b3 not in 95%CI"

	ge b4_lb = b_nownch + se_nownch*invnormal( (1-.95)/2 )
	ge b4_ub = b_nownch - se_nownch*invnormal( (1-.95)/2 )
	lab var b4_lb "lb 95% CI, b0 (nownch)"
	lab var b4_ub "ub 95% CI, b0 (nownch)"
	ge b_nownch_noncover = 0
	replace b_nownch_noncover = 1 if (`b4' < b4_lb | `b4' > b4_ub)
	lab var b_nownch_noncover "1: true b4 not in 95%CI"

	ge b5_lb = b_isced3 + se_isced3*invnormal( (1-.95)/2 )
	ge b5_ub = b_isced3 - se_isced3*invnormal( (1-.95)/2 )
	lab var b5_lb "lb 95% CI, b0 (isced3)"
	lab var b5_ub "ub 95% CI, b0 (isced3)"
	ge b_isced3_noncover = 0
	replace b_isced3_noncover = 1 if (`b5' < b5_lb | `b5' > b5_ub)
```

```
        lab var b_isced3_noncover "1: true b5 not in 95%CI"

        ge b6_lb = b_isced4 + se_isced4*invnormal( (1-.95)/2 )
        ge b6_ub = b_isced4 - se_isced4*invnormal( (1-.95)/2 )
        lab var b6_lb "lb 95% CI, b0 (isced4)"
        lab var b6_ub "ub 95% CI, b0 (isced4)"
        ge b_isced4_noncover = 0
        replace b_isced4_noncover = 1 if (`b6' < b6_lb | `b6' > b6_ub)
        lab var b_isced4_noncover "1: true b6 not in 95%CI"

        ge b7_lb = b_isced56 + se_isced56*invnormal( (1-.95)/2 )
        ge b7_ub = b_isced56 - se_isced56*invnormal( (1-.95)/2 )
        lab var b7_lb "lb 95% CI, b0 (isced56)"
        lab var b7_ub "ub 95% CI, b0 (isced56)"
        ge b_isced56_noncover = 0
        replace b_isced56_noncover = 1 if (`b7' < b7_lb | `b7' > b7_ub)
        lab var b_isced56_noncover "1: true b7 not in 95%CI"

        ge c1_lb = b_chexp + se_chexp*invnormal( (1-.95)/2 )
        ge c1_ub = b_chexp - se_chexp*invnormal( (1-.95)/2 )
        lab var c1_lb "lb 95% CI, c1 (chexp)"
        lab var c1_ub "ub 95% CI, c1 (chexp)"
        ge c_chexp_noncover = 0
        replace c_chexp_noncover = 1 if (`c1' < c1_lb | `c1' > c1_ub)
        lab var c_chexp_noncover "1: true c1 not in 95%CI"

        ge sig_u_lb = sig_u + sig_u_se*invnormal( (1-.95)/2 )
        ge sig_u_ub = sig_u - sig_u_se*invnormal( (1-.95)/2 )
        lab var sig_u_lb "lb 95% CI, sig_u"
        lab var sig_u_ub "ub 95% CI, sig_u"
        ge sig_u_noncover = 0
        replace sig_u_noncover = 1 if (`sig_u' < sig_u_lb | `sig_u' > sig_u_ub)
        lab var sig_u_noncover "1: true sig_u not in 95%CI"

        ge sig_e_lb = sig_e + sig_e_se*invnormal( (1-.95)/2 )
        ge sig_e_ub = sig_e - sig_e_se*invnormal( (1-.95)/2 )
        lab var sig_e_lb "lb 95% CI, sig_e"
        lab var sig_e_ub "ub 95% CI, sig_e"
        ge sig_e_noncover = 0
        replace sig_e_noncover = 1 if (`sig_e' < sig_e_lb | `sig_e' > sig_e_ub)
        lab var sig_e_noncover "1: true sig_e not in 95%CI"


* other stats

        ge Nc = `Nc'
        lab var Nc "No. persons per country"

        ge C = `C'
        lab var C "Number of countries"

        ge icc = `icc'
        lab var icc "icc = var_u/(var_u+var_e)"

        drop lns*
        compress

* }


save, replace

de, fullnames


su b_cons b_age b_agesq  b_cohab  b_nownch b_isced3 b_isced4 b_isced56 b_chexp

su se_cons se_age se_agesq  se_cohab  se_nownch se_isced3 se_isced4 se_isced56 se_chexp

su sig_e sig_u

su *abias

su *rbias

su *noncover
```

```
su Nc C icc converged

preserve
keep if converged == 1

su b_cons b_age b_agesq  b_cohab  b_nownch b_isced3 b_isced4 b_isced56 b_chexp

su se_cons se_age se_agesq  se_cohab  se_nownch se_isced3 se_isced4 se_isced56 se_chexp

su sig_e sig_u

su *abias

su *rbias

su *noncover

su Nc C icc

restore

*******************************
log close
```

(b) Extended linear model (mc_hours_model3_v01a_Nc_C.do)

```
clear all
set more off
version 11
set mem 500m
capture log close

local Nc = `1'

local C = `2'

local N = `C' * `Nc'


log using mc_hours_model3_v01a_`Nc'_`C'.log, replace


********** Monte-Carlo exercise **************
*
* Outcome = Number of hours
*
* Model 3: interaction between individual and country-level variables, and
*          country-specific random slopes
*
* Use arguments to do file to set parameters of truth model: e.g.
*
* do mc_hours_model3_v01a_Nc_C 1000 10
*
*              where "1000" is # persons/country and "10" is # countries
*
* (1) declare model parameter values
* (2) create data set
* (3) set-up MC (5000 reps)
* (4) post-fitting processing of results
*
*********************************************

*********************************************
* (1) declare model parameter values
*********************************************

di "Number of persons per country, N_c = " `Nc'
di "Number of countries, C   = " `C'
```

```
di "Total sample size (Nc x C), N   = " `N'

/*

Woman aged 18-64 with positive work hours

Model 3 for hours:

Hours_ic = b0
        + b1 * age_ic
        + b2 * age-squared_ic
        + b3 * cohab_ic    +  b3c * cohab_ic  <--- random slope
        + b4 * nownch_ic   +  b4c * nownch_ic   <--- random slope
        + b5 * isced3_ic
        + b6 * isced4_ic
        + b7 * isced56_ic
        + c1 * chexp_c                <-- country=level
        + c2 * (chexp_c X cohab_ic)
        + c3 * (chexp_c X nownch_ic)
        + u_c
        + e_ic

        u_c  ~ N(0, sig_u^2)
        e_ic ~ N(0, sig_e^2)
        cov(u_c, e_ic) = 0

        b3c ~ N(0, sig_b3c^2)
        b4c ~ N(0, sig_b4c^2)

*/

local b0 = 22
local b1 = 0.8
local b2 = -0.01
local b3 = -1
local b4 = -1.2
local b5 = 0.7
local b6 = 1.4
local b7 = 1.6
local c1 = -2.7
local c2 = 2.4
local c3 = 0.7
local sig_u = 2.4
local sig_e = 9.4

local sig_b3c = 1.2
local sig_b4c = 1.2

local icc = (`sig_u')^2 / ( (`sig_u')^2 + (`sig_e')^2  )
di "ICC = " `icc'

************************************************
* (2) create data set (using summary.log stats)
************************************************

set obs `N'

set seed 123456789

* "fixed" part of model (X, Z assumed same/fixed across experiments)

ge id = _n
ge country = autocode(id, `C', 1, `N')
egen country_id = group(country)
bys country_id: ge tag = _n == 1
compress

* age distribution
* Singh-Maddala parameters: a =

local a = 4.35
local b = 250
local q = 1656
ge age = `b' * ( (1/runiform())^(1/`q') - 1 )^(1/`a')
sum age
replace age = . if (age < 18 | age > 64)
sum age
```

```
count if missing(age)
local nm = r(N)
while `nm' > 0 {
        replace age = `b' * ( (1/runiform())^(1/`q') - 1 )^(1/`a') if missing(age)
        sum age
        replace age = . if (age < 18 | age > 64)
        count if missing(age)
        local nm = r(N)
}

replace age = round(age)

ge agegp = .
replace agegp = 1 if age >= 18 & age < 25
replace agegp = 2 if age >= 25 & age < 40
replace agegp = 3 if age >= 40 & age < 50
replace agegp = 4 if age >= 50 & age < 60
replace agegp = 5 if age >= 60 & age < .
lab def agegp 1 "18-24" 2 "25-39" 3 "40-49" 4 "50-59" 5 "60-64"
lab val agegp agegp
ta agegp

ge agesq = age^2

ge cohab = .
ge nownch = .
ge isced012 = 0
ge isced3 = 0
ge isced4 = 0
ge isced56 = 0

lab var cohab "Partnership status
lab def cohab 0 "No partner" 1 "Has parter"
lab val cohab cohab
lab var nownch "# kids"
lab def nownch 0 "0" 1 "1" 2 "2" 3 "3+"
lab val nownch nownch

ge type = .
lab var type "Characteristics combination"
lab def type ///
           1 "Not marr 0 isced0,1,2" ///
           2 "Not marr 0 isced3" ///
           3 "Not marr 0 isced4" ///
           4 "Not marr 0 isced5,6" ///
           5 "Not marr 1 isced0,1,2" ///
           6 "Not marr 1 isced3" ///
           7 "Not marr 1 isced4" ///
           8 "Not marr 1 isced5,6" ///
           9 "Not marr 2 isced0,1,2" ///
          10 "Not marr 2 isced3" ///
          11 "Not marr 2 isced4" ///
          12 "Not marr 2 isced5,6" ///
          13 "Not marr 3+ isced0,1,2" ///
          14 "Not marr 3+ isced3" ///
          15 "Not marr 3+ isced4" ///
          16 "Not marr 3+ isced5,6" ///
          17 "Marr 0 isced0,1,2" ///
          18 "Marr 0 isced3" ///
          19 "Marr 0 isced4" ///
          20 "Marr 0 isced5,6" ///
          21 "Marr 1 isced0,1,2" ///
          22 "Marr 1 isced3" ///
          23 "Marr 1 isced4" ///
          24 "Marr 1 isced5,6" ///
          25 "Marr 2 isced0,1,2" ///
          26 "Marr 2 isced3" ///
          27 "Marr 2 isced4" ///
          28 "Marr 2 isced5,6" ///
          29 "Marr 3+ isced0,1,2" ///
          30 "Marr 3+ isced3" ///
          31 "Marr 3+ isced4" ///
          32 "Marr 3+ isced5,6"
lab val type type

ge double u = 100 * runiform()
```

```
replace type =  1 if agegp == 1 & u <= 14.42
replace type =  2 if agegp == 1 & u > 14.42  &  u <= 58.23
replace type =  3 if agegp == 1 & u > 58.23  &  u <= 62.91
replace type =  4 if agegp == 1 & u > 62.91  &  u <= 74.45
replace type =  5 if agegp == 1 & u > 74.45  &  u <= 75.07
replace type =  6 if agegp == 1 & u > 75.07  &  u <= 76.08
replace type =  7 if agegp == 1 & u > 76.08  &  u <= 76.12
replace type =  8 if agegp == 1 & u > 76.12  &  u <= 76.24
replace type =  9 if agegp == 1 & u > 76.24  &  u <= 76.29
replace type = 10 if agegp == 1 & u > 76.29  &  u <= 76.37
replace type = 11 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 12 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 13 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 14 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 15 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 16 if agegp == 1 & u > 76.37  &  u <= 76.37   // # cases = 0
replace type = 17 if agegp == 1 & u > 76.37  &  u <= 78.77
replace type = 18 if agegp == 1 & u > 78.77  &  u <= 89.45
replace type = 19 if agegp == 1 & u > 89.45  &  u <= 90.30
replace type = 20 if agegp == 1 & u > 90.30  &  u <= 94.27
replace type = 21 if agegp == 1 & u > 94.27  &  u <= 95.41
replace type = 22 if agegp == 1 & u > 95.41  &  u <= 98.28
replace type = 23 if agegp == 1 & u > 98.28  &  u <= 98.44
replace type = 24 if agegp == 1 & u > 98.44  &  u <= 98.84
replace type = 25 if agegp == 1 & u > 98.84  &  u <= 99.21
replace type = 26 if agegp == 1 & u > 99.21  &  u <= 99.86
replace type = 27 if agegp == 1 & u > 99.86  &  u <= 99.89
replace type = 28 if agegp == 1 & u > 99.89  &  u <= 99.92
replace type = 29 if agegp == 1 & u > 99.92  &  u <= 99.95
replace type = 30 if agegp == 1 & u > 99.95  &  u <= 100.00
replace type = 31 if agegp == 1 & u > 100.00                     // # cases = 0
replace type = 32 if agegp == 1 & u > 100.00                     // # cases = 0

replace type =  1 if agegp == 2 & u <= 1.96
replace type =  2 if agegp == 2 & u >  1.96  &  u <= 10.21
replace type =  3 if agegp == 2 & u > 10.21  &  u <= 11.72
replace type =  4 if agegp == 2 & u > 11.72  &  u <= 22.25
replace type =  5 if agegp == 2 & u > 22.25  &  u <= 22.92
replace type =  6 if agegp == 2 & u > 22.92  &  u <= 25.31
replace type =  7 if agegp == 2 & u > 25.31  &  u <= 25.64
replace type =  8 if agegp == 2 & u > 25.64  &  u <= 26.93
replace type =  9 if agegp == 2 & u > 26.93  &  u <= 27.41
replace type = 10 if agegp == 2 & u > 27.41  &  u <= 28.77
replace type = 11 if agegp == 2 & u > 28.77  &  u <= 28.91
replace type = 12 if agegp == 2 & u > 28.91  &  u <= 29.44
replace type = 13 if agegp == 2 & u > 29.44  &  u <= 29.67
replace type = 14 if agegp == 2 & u > 29.67  &  u <= 30.00
replace type = 15 if agegp == 2 & u > 30.00  &  u <= 30.01
replace type = 16 if agegp == 2 & u > 30.01  &  u <= 30.12
replace type = 17 if agegp == 2 & u > 30.12  &  u <= 31.43
replace type = 18 if agegp == 2 & u > 31.43  &  u <= 37.24
replace type = 19 if agegp == 2 & u > 37.24  &  u <= 38.09
replace type = 20 if agegp == 2 & u > 38.09  &  u <= 46.16
replace type = 21 if agegp == 2 & u > 46.16  &  u <= 48.43
replace type = 22 if agegp == 2 & u > 48.43  &  u <= 57.34
replace type = 23 if agegp == 2 & u > 57.34  &  u <= 58.49
replace type = 24 if agegp == 2 & u > 58.49  &  u <= 65.70
replace type = 25 if agegp == 2 & u > 65.70  &  u <= 68.83
replace type = 26 if agegp == 2 & u > 68.83  &  u <= 81.70
replace type = 27 if agegp == 2 & u > 81.70  &  u <= 83.15
replace type = 28 if agegp == 2 & u > 83.15  &  u <= 92.10
replace type = 29 if agegp == 2 & u > 92.10  &  u <= 93.43
replace type = 30 if agegp == 2 & u > 93.43  &  u <= 97.19
replace type = 31 if agegp == 2 & u > 97.19  &  u <= 97.52
replace type = 32 if agegp == 2 & u > 97.52

replace type =  1 if agegp == 3 & u <= 1.33
replace type =  2 if agegp == 3 & u >  1.33  &  u <=  4.46
replace type =  3 if agegp == 3 & u >  4.46  &  u <=  5.05
replace type =  4 if agegp == 3 & u >  5.05  &  u <=  7.83
replace type =  5 if agegp == 3 & u >  7.83  &  u <=  8.81
replace type =  6 if agegp == 3 & u >  8.81  &  u <= 11.63
replace type =  7 if agegp == 3 & u > 11.63  &  u <= 12.16
replace type =  8 if agegp == 3 & u > 12.16  &  u <= 14.04
replace type =  9 if agegp == 3 & u > 14.04  &  u <= 14.82
replace type = 10 if agegp == 3 & u > 14.82  &  u <= 17.13
replace type = 11 if agegp == 3 & u > 17.13  &  u <= 17.50
```

```
replace type = 12 if agegp == 3 & u > 17.50  &  u <= 19.00
replace type = 13 if agegp == 3 & u > 19.00  &  u <= 19.30
replace type = 14 if agegp == 3 & u > 19.30  &  u <= 20.00
replace type = 15 if agegp == 3 & u > 20.00  &  u <= 20.09
replace type = 16 if agegp == 3 & u > 20.09  &  u <= 20.59
replace type = 17 if agegp == 3 & u > 20.59  &  u <= 22.54
replace type = 18 if agegp == 3 & u > 22.54  &  u <= 27.32
replace type = 19 if agegp == 3 & u > 27.32  &  u <= 27.99
replace type = 20 if agegp == 3 & u > 27.99  &  u <= 30.90
replace type = 21 if agegp == 3 & u > 30.90  &  u <= 34.62
replace type = 22 if agegp == 3 & u > 34.62  &  u <= 44.00
replace type = 23 if agegp == 3 & u > 44.00  &  u <= 45.29
replace type = 24 if agegp == 3 & u > 45.29  &  u <= 50.88
replace type = 25 if agegp == 3 & u > 50.88  &  u <= 56.49
replace type = 26 if agegp == 3 & u > 56.49  &  u <= 72.54
replace type = 27 if agegp == 3 & u > 72.54  &  u <= 74.33
replace type = 28 if agegp == 3 & u > 74.33  &  u <= 85.71
replace type = 29 if agegp == 3 & u > 85.71  &  u <= 87.94
replace type = 30 if agegp == 3 & u > 87.94  &  u <= 94.45
replace type = 31 if agegp == 3 & u > 94.45  &  u <= 95.15
replace type = 32 if agegp == 3 & u > 95.15

replace type =  1 if agegp == 4 & u <= 2.78
replace type =  2 if agegp == 4 & u >  2.78  &  u <=  8.08
replace type =  3 if agegp == 4 & u >  8.08  &  u <=  8.84
replace type =  4 if agegp == 4 & u >  8.84  &  u <= 13.28
replace type =  5 if agegp == 4 & u > 13.28  &  u <= 14.83
replace type =  6 if agegp == 4 & u > 14.83  &  u <= 17.96
replace type =  7 if agegp == 4 & u > 17.96  &  u <= 18.36
replace type =  8 if agegp == 4 & u > 18.36  &  u <= 20.37
replace type =  9 if agegp == 4 & u > 20.37  &  u <= 20.97
replace type = 10 if agegp == 4 & u > 20.97  &  u <= 22.05
replace type = 11 if agegp == 4 & u > 22.05  &  u <= 22.21
replace type = 12 if agegp == 4 & u > 22.21  &  u <= 22.97
replace type = 13 if agegp == 4 & u > 22.97  &  u <= 23.12
replace type = 14 if agegp == 4 & u > 23.12  &  u <= 23.32
replace type = 15 if agegp == 4 & u > 23.32  &  u <= 23.34
replace type = 16 if agegp == 4 & u > 23.34  &  u <= 23.48
replace type = 17 if agegp == 4 & u > 23.48  &  u <= 32.15
replace type = 18 if agegp == 4 & u > 32.15  &  u <= 48.05
replace type = 19 if agegp == 4 & u > 48.05  &  u <= 49.82
replace type = 20 if agegp == 4 & u > 49.82  &  u <= 59.33
replace type = 21 if agegp == 4 & u > 59.33  &  u <= 64.92
replace type = 22 if agegp == 4 & u > 64.92  &  u <= 75.12
replace type = 23 if agegp == 4 & u > 75.12  &  u <= 76.31
replace type = 24 if agegp == 4 & u > 76.31  &  u <= 82.55
replace type = 25 if agegp == 4 & u > 82.55  &  u <= 85.74
replace type = 26 if agegp == 4 & u > 85.74  &  u <= 91.31
replace type = 27 if agegp == 4 & u > 91.31  &  u <= 91.88
replace type = 28 if agegp == 4 & u > 91.88  &  u <= 96.37
replace type = 29 if agegp == 4 & u > 96.37  &  u <= 97.20
replace type = 30 if agegp == 4 & u > 97.20  &  u <= 98.52
replace type = 31 if agegp == 4 & u > 98.52  &  u <= 98.72
replace type = 32 if agegp == 4 & u > 98.72

replace type =  1 if agegp == 5 & u <= 5.66
replace type =  2 if agegp == 5 & u >  5.66  &  u <= 12.73
replace type =  3 if agegp == 5 & u > 12.73  &  u <= 13.92
replace type =  4 if agegp == 5 & u > 13.92  &  u <= 20.87
replace type =  5 if agegp == 5 & u > 20.87  &  u <= 23.05
replace type =  6 if agegp == 5 & u > 23.05  &  u <= 24.81
replace type =  7 if agegp == 5 & u > 24.81  &  u <= 25.17
replace type =  8 if agegp == 5 & u > 25.17  &  u <= 26.72
replace type =  9 if agegp == 5 & u > 26.72  &  u <= 27.08
replace type = 10 if agegp == 5 & u > 27.08  &  u <= 27.38
replace type = 11 if agegp == 5 & u > 27.38  &  u <= 27.41
replace type = 12 if agegp == 5 & u > 27.41  &  u <= 27.80
replace type = 13 if agegp == 5 & u > 27.80  &  u <= 27.89
replace type = 14 if agegp == 5 & u > 27.89  &  u <= 27.92
replace type = 15 if agegp == 5 & u > 27.92  &  u <= 27.92   // # cases = 0
replace type = 16 if agegp == 5 & u > 27.92  &  u <= 27.98
replace type = 17 if agegp == 5 & u > 27.98  &  u <= 44.53
replace type = 18 if agegp == 5 & u > 44.53  &  u <= 65.64
replace type = 19 if agegp == 5 & u > 65.64  &  u <= 68.11
replace type = 20 if agegp == 5 & u > 68.11  &  u <= 84.09
replace type = 21 if agegp == 5 & u > 84.09  &  u <= 89.78
replace type = 22 if agegp == 5 & u > 89.78  &  u <= 93.48
```

```
replace type = 23 if agegp == 5 & u > 93.48  &  u <= 94.02
replace type = 24 if agegp == 5 & u > 94.02  &  u <= 97.12
replace type = 25 if agegp == 5 & u > 97.12  &  u <= 98.10
replace type = 26 if agegp == 5 & u > 98.10  &  u <= 98.82
replace type = 27 if agegp == 5 & u > 98.82  &  u <= 98.82  // # cases = 0
replace type = 28 if agegp == 5 & u > 98.82  &  u <= 99.42
replace type = 29 if agegp == 5 & u > 99.42  &  u <= 99.66
replace type = 30 if agegp == 5 & u > 99.66  &  u <= 99.84
replace type = 31 if agegp == 5 & u > 99.84  &  u <= 99.90
replace type = 32 if agegp == 5 & u > 99.90

ta type agegp, col missing

replace cohab = 0 if inrange(type, 1, 16)
replace cohab = 1 if inrange(type, 17, 32)

replace nownch = 0 if inrange(type, 1, 4) | inrange(type, 17, 20)
replace nownch = 1 if inrange(type, 5, 8) | inrange(type, 21, 24)
replace nownch = 2 if inrange(type, 9, 12) | inrange(type, 25, 28)
replace nownch = 3 if inrange(type, 13, 16) | inrange(type, 29, 32)

replace isced012 = 1 if inlist(type, 1, 5, 9, 13, 17, 21, 25, 29)
replace isced3   = 1 if inlist(type, 2, 6, 10, 14, 18, 22, 26, 30)
replace isced4         = 1 if inlist(type, 3, 7, 11, 15, 19, 23, 27, 31)
replace isced56  = 1 if inlist(type, 4, 8, 12, 16, 20, 24, 28, 32)

gen edlevel = .
replace edlevel = 1 if isced012 == 1
replace edlevel = 2 if isced3 == 1
replace edlevel = 3 if isced4 == 1
replace edlevel = 4 if isced56 == 1
lab def edlevel 1 "isced0,1,2" 2 "isced3" 3 "isced4" 4 "isced5,6"
lab val edlevel edlevel


* lognormal distribution for child exp. Parameters: m = -0.665, v = 0.535

local m = -0.665
local v = 0.535

ge chexp = exp( `m' + `v' * invnormal( runiform() ) ) if tag
lab var chexp "child care expend, % GDP"
bys country_id: replace chexp = chexp[1]

ge chexpXcohab = chexp * cohab
ge chexpXnownch = chexp * nownch

lab var chexpXcohab "chexp,cohab interaction"
lab var chexpXnownch "chexp,nownch interaction"

compress

ge fixed = `b0'                           ///
       + `b1' * age                       ///
       + `b2' * agesq                     ///
       + `b3' * cohab                     ///
       + `b4' * nownch                    ///
       + `b5' * isced3                    ///
       + `b6' * isced4                    ///
       + `b7' * isced56                   ///
       + `c1' * chexp                     ///
       + `c2' * chexpXcohab               ///
       + `c3' * chexpXnownch


ta country_id, sum(age)
ta country_id, sum(isced012)
ta country_id, sum(isced3)
ta country_id, sum(isced4)
ta country_id, sum(isced56)
ta country_id, sum(cohab)
ta country_id, sum(nownch)
ta country_id, sum(chexp)

de, fullnames
su
save mc_hours_model3_v01a_`Nc'_`C'_data, replace
```

```
************************************************
* (3) set-up MC
************************************************



cap program drop mc_silc
program define mc_silc

        version 11
        args sig_e sig_u sig_b3c sig_b4c

        capture drop y u_c b3c b4c e_ic

        gen e_ic =  rnormal(0,`sig_e')

        gen u_c = rnormal(0, `sig_u') if tag
        bys country_id : replace u_c = u_c[1]

        gen b3c = rnormal(0, `sig_b3c') if tag
        bys country_id: replace b3c = b3c[1]

        gen b4c = rnormal(0, `sig_b4c') if tag
        bys country_id: replace b4c = b4c[1]

        gen y = fixed + u_c + b3c*cohab + b4c*nownch + e_ic

                // default estimation method is REML, cov structure independent
        xtmixed y age agesq cohab nownch isced3 isced4 isced56 ///
                  chexp chexpXcohab chexpXnownch || country_id: cohab nownch , nolog  iter(250)

end

di "Time is: " c(current_time) " on " c(current_date)
simulate _b _se converged = e(converged) logRLL = e(ll) ///
        , reps(5000) saving(mc_hours_model3_v01a_`Nc'_`C'_output.dta, replace double)  ///
        : mc_silc (`sig_e') (`sig_u') (`sig_b3c') (`sig_b4c')
di "Time is: " c(current_time) " on " c(current_date)


de, fullnames
ds, varwidth(20)
su

*********************************************
* (4) post-fitting processing of results and saving data
*********************************************


* convert back to original metric; calculate summary stats

/*

lns1_1_1 sd(cohab) = sig_b3c
lns1_1_2 sd(nownch) = sig_b4c
lns1_1_3 sig_u

*/

* quietly {

        rename _eq11_converged  converged
        rename _eq11_logRLL logRLL

        renpfix y_

        ge sig_e = exp(lnsig_e_b_cons)
        ge sig_e_se = exp(lnsig_e_b_cons)*lnsig_e_se_cons

        ge sig_b3c = exp(lns1_1_1_b_cons)
        ge sig_b3c_se = sig_b3c * lns1_1_1_se_cons

        ge sig_b4c = exp(lns1_1_2_b_cons)
        ge sig_b4c_se = sig_b4c * lns1_1_2_se_cons
```

```
        ge sig_u = exp(lns1_1_3_b_cons)
        ge sig_u_se = sig_u * lns1_1_3_se_cons

* absolute bias
        ge b_cons_abias = b_cons - `b0'
        lab var b_cons_abias "Absolute bias b0 (cons)"

        ge b_age_abias = b_age - `b1'
        lab var b_age_abias "Absolute bias b1 (age)"

        ge b_agesq_abias = b_agesq - `b2'
        lab var b_agesq_abias "Absolute bias b2 (agesq)"

        ge b_cohab_abias = b_cohab - `b3'
        lab var b_cohab_abias "Absolute bias b3 (cohab)"

        ge b_nownch_abias = b_nownch - `b4'
        lab var b_nownch_abias "Absolute bias b4 (nownch)"

        ge b_isced3_abias = b_isced3 - `b5'
        lab var b_isced3_abias "Absolute bias b5 (isced3)"

        ge b_isced4_abias = b_isced4 - `b6'
        lab var b_isced4_abias "Absolute bias b6 (isced4)"

        ge b_isced56_abias = b_isced56 - `b7'
        lab var b_isced56_abias "Absolute bias b7 (isced56)"

        ge c_chexp_abias = b_chexp - `c1'
        lab var c_chexp_abias "Absolute bias c1 (chexp)"

        ge c_chexpXcohab_abias = b_chexpXcohab - `c2'
        lab var c_chexpXcohab_abias "Absolute bias c2 (chexpXcohab)"

        ge c_chexpXnownch_abias = b_chexpXnownch - `c3'
        lab var c_chexpXnownch_abias "Absolute bias c3 (chexpXnownch)"

        ge sig_u_abias = sig_u - `sig_u'
        lab var sig_u_abias "Absolute bias sig_u"

        ge sig_e_abias  = sig_e - `sig_e'
        lab var sig_e_abias "Absolute bias sig_e"

        ge sig_b3c_abias = sig_b3c - `sig_b3c'
        lab var sig_b3c_abias "Absolute bias sig_b3c"

        ge sig_b4c_abias = sig_b4c - `sig_b4c'
        lab var sig_b4c_abias "Absolute bias sig_b4c"

* relative bias

        ge b_cons_rbias = 100*(b_cons/`b0' - 1)
        lab var b_cons_rbias "% Relative bias b0 (cons)"

        ge b_age_rbias = 100*(b_age/`b1' - 1)
        lab var b_age_rbias "% Relative bias b1 (age)"

        ge b_agesq_rbias = 100*(b_agesq/`b2' - 1)
        lab var b_agesq_rbias "% Relative bias b2 (agesq)"

        ge b_cohab_rbias = 100*(b_cohab/`b3' - 1)
        lab var b_cohab_rbias "% Relative bias b3 (cohab)"

        ge b_nownch_rbias = 100*(b_nownch/`b4' - 1)
        lab var b_nownch_rbias "% Relative bias b4 (nownch)"

        ge b_isced3_rbias = 100*(b_isced3/`b5' - 1)
        lab var b_isced3_rbias "% Relative bias b5 (isced3)"

        ge b_isced4_rbias = 100*(b_isced4/`b6' - 1)
        lab var b_isced4_rbias "% Relative bias b6 (isced4)"

        ge b_isced56_rbias = 100*(b_isced56/`b7' - 1)
        lab var b_isced56_rbias "% Relative bias b7 (isced56)"

        ge c_chexp_rbias = 100*(b_chexp/`c1' - 1)
        lab var c_chexp_rbias "% Relative bias c1 (chexp)"
```

```
	ge c_chexpXcohab_rbias = 100*(b_chexpXcohab/`c2' - 1)
	lab var c_chexpXcohab_rbias "% Relative bias c2 (chexpXcohab)"

	ge c_chexpXnownch_rbias = 100*(b_chexpXnownch/`c3' - 1)
	lab var c_chexpXnownch_rbias "% Relative bias c3 (chexpXnownch)"

	ge sig_u_rbias = 100*(sig_u / `sig_u' - 1)
	lab var sig_u_rbias "% Relative bias sig_u"

	ge sig_e_rbias = 100*(sig_e / `sig_e' - 1)
	lab var sig_e_rbias "% Relative bias sig_e"

	ge sig_b3c_rbias = 100*(sig_b3c / `sig_b3c' - 1)
	lab var sig_b3c_rbias "% Relative bias sig_b3c"

	ge sig_b4c_rbias = 100*(sig_b3c / `sig_b4c' - 1)
	lab var sig_b4c_rbias "% Relative bias sig_b4c"



* non-coverage

	ge b0_lb = b_cons + se_cons*invnormal( (1-.95)/2 )
	ge b0_ub = b_cons - se_cons*invnormal( (1-.95)/2 )
	lab var b0_lb "lb 95% CI, b0 (cons)"
	lab var b0_ub "ub 95% CI, b0 (cons)"
	ge b_cons_noncover = 0
	replace b_cons_noncover = 1 if (`b0' < b0_lb | `b0' > b0_ub)
	lab var b_cons_noncover "1: true b0 not in 95%CI"

	ge b1_lb = b_age + se_age*invnormal( (1-.95)/2 )
	ge b1_ub = b_age - se_age*invnormal( (1-.95)/2 )
	lab var b1_lb "lb 95% CI, b1 (age)"
	lab var b1_ub "ub 95% CI, b1 (age)"
	ge b_age_noncover = 0
	replace b_age_noncover = 1 if (`b1' < b1_lb | `b1' > b1_ub)
	lab var b_age_noncover "1: true b1 not in 95%CI"

	ge b2_lb = b_agesq + se_agesq*invnormal( (1-.95)/2 )
	ge b2_ub = b_agesq - se_agesq*invnormal( (1-.95)/2 )
	lab var b2_lb "lb 95% CI, b2 (agesq)"
	lab var b2_ub "ub 95% CI, b2 (agesq)"
	ge b_agesq_noncover = 0
	replace b_agesq_noncover = 1 if (`b2' < b2_lb | `b2' > b2_ub)
	lab var b_agesq_noncover "1: true b2 not in 95%CI"

	ge b3_lb = b_cohab + se_cohab*invnormal( (1-.95)/2 )
	ge b3_ub = b_cohab - se_cohab*invnormal( (1-.95)/2 )
	lab var b3_lb "lb 95% CI, b0 (cohab)"
	lab var b3_ub "ub 95% CI, b0 (cohab)"
	ge b_cohab_noncover = 0
	replace b_cohab_noncover = 1 if (`b3' < b3_lb | `b3' > b3_ub)
	lab var b_cohab_noncover "1: true b3 not in 95%CI"

	ge b4_lb = b_nownch + se_nownch*invnormal( (1-.95)/2 )
	ge b4_ub = b_nownch - se_nownch*invnormal( (1-.95)/2 )
	lab var b4_lb "lb 95% CI, b0 (nownch)"
	lab var b4_ub "ub 95% CI, b0 (nownch)"
	ge b_nownch_noncover = 0
	replace b_nownch_noncover = 1 if (`b4' < b4_lb | `b4' > b4_ub)
	lab var b_nownch_noncover "1: true b4 not in 95%CI"

	ge b5_lb = b_isced3 + se_isced3*invnormal( (1-.95)/2 )
	ge b5_ub = b_isced3 - se_isced3*invnormal( (1-.95)/2 )
	lab var b5_lb "lb 95% CI, b0 (isced3)"
	lab var b5_ub "ub 95% CI, b0 (isced3)"
	ge b_isced3_noncover = 0
	replace b_isced3_noncover = 1 if (`b5' < b5_lb | `b5' > b5_ub)
	lab var b_isced3_noncover "1: true b5 not in 95%CI"

	ge b6_lb = b_isced4 + se_isced4*invnormal( (1-.95)/2 )
	ge b6_ub = b_isced4 - se_isced4*invnormal( (1-.95)/2 )
	lab var b6_lb "lb 95% CI, b0 (isced4)"
	lab var b6_ub "ub 95% CI, b0 (isced4)"
	ge b_isced4_noncover = 0
	replace b_isced4_noncover = 1 if (`b6' < b6_lb | `b6' > b6_ub)
```

```
        lab var b_isced4_noncover "1: true b6 not in 95%CI"

        ge b7_lb = b_isced56 + se_isced56*invnormal( (1-.95)/2 )
        ge b7_ub = b_isced56 - se_isced56*invnormal( (1-.95)/2 )
        lab var b7_lb "lb 95% CI, b0 (isced56)"
        lab var b7_ub "ub 95% CI, b0 (isced56)"
        ge b_isced56_noncover = 0
        replace b_isced56_noncover = 1 if (`b7' < b7_lb | `b7' > b7_ub)
        lab var b_isced56_noncover "1: true b7 not in 95%CI"

        ge c1_lb = b_chexp + se_chexp*invnormal( (1-.95)/2 )
        ge c1_ub = b_chexp - se_chexp*invnormal( (1-.95)/2 )
        lab var c1_lb "lb 95% CI, c1 (chexp)"
        lab var c1_ub "ub 95% CI, c1 (chexp)"
        ge c_chexp_noncover = 0
        replace c_chexp_noncover = 1 if (`c1' < c1_lb | `c1' > c1_ub)
        lab var c_chexp_noncover "1: true c1 not in 95%CI"

        ge c2_lb = b_chexpXcohab + se_chexpXcohab*invnormal( (1-.95)/2 )
        ge c2_ub = b_chexpXcohab - se_chexpXcohab*invnormal( (1-.95)/2 )
        lab var c2_lb "lb 95% CI, c2 (chexpXcohab)"
        lab var c2_ub "ub 95% CI, c2 (chexpXcohab)"
        ge c_chexpXcohab_noncover = 0
        replace c_chexpXcohab_noncover = 1 if (`c2' < c2_lb | `c2' > c2_ub)
        lab var c_chexpXcohab_noncover "1: true c2 not in 95%CI"

        ge c3_lb = b_chexpXownch + se_chexpXownch*invnormal( (1-.95)/2 )
        ge c3_ub = b_chexpXownch - se_chexpXownch*invnormal( (1-.95)/2 )
        lab var c3_lb "lb 95% CI, c3 (chexpXownch)"
        lab var c3_ub "ub 95% CI, c3 (chexpXownch)"
        ge c_chexpXownch_noncover = 0
        replace c_chexpXownch_noncover = 1 if (`c3' < c3_lb | `c3' > c3_ub)
        lab var c_chexpXownch_noncover "1: true c3 not in 95%CI"

        ge sig_u_lb = sig_u + sig_u_se*invnormal( (1-.95)/2 )
        ge sig_u_ub = sig_u - sig_u_se*invnormal( (1-.95)/2 )
        lab var sig_u_lb "lb 95% CI, sig_u"
        lab var sig_u_ub "ub 95% CI, sig_u"
        ge sig_u_noncover = 0
        replace sig_u_noncover = 1 if (`sig_u' < sig_u_lb | `sig_u' > sig_u_ub)
        lab var sig_u_noncover "1: true sig_u not in 95%CI"

        ge sig_e_lb = sig_e + sig_e_se*invnormal( (1-.95)/2 )
        ge sig_e_ub = sig_e - sig_e_se*invnormal( (1-.95)/2 )
        lab var sig_e_lb "lb 95% CI, sig_e"
        lab var sig_e_ub "ub 95% CI, sig_e"
        ge sig_e_noncover = 0
        replace sig_e_noncover = 1 if (`sig_e' < sig_e_lb | `sig_e' > sig_e_ub)
        lab var sig_e_noncover "1: true sig_e not in 95%CI"

        ge sig_b3c_lb = sig_b3c + sig_b3c_se*invnormal( (1-.95)/2 )
        ge sig_b3c_ub = sig_b3c - sig_b3c_se*invnormal( (1-.95)/2 )
        lab var sig_b3c_lb "lb 95% CI, sig_b3c"
        lab var sig_b3c_ub "ub 95% CI, sig_b3c"
        ge sig_b3c_noncover = 0
        replace sig_b3c_noncover = 1 if (`sig_b3c' < sig_b3c_lb | `sig_b3c' > sig_b3c_ub)
        lab var sig_b3c_noncover "1: true sig_b3c not in 95%CI"

        ge sig_b4c_lb = sig_b4c + sig_b4c_se*invnormal( (1-.95)/2 )
        ge sig_b4c_ub = sig_b4c - sig_b4c_se*invnormal( (1-.95)/2 )
        lab var sig_b4c_lb "lb 95% CI, sig_b4c"
        lab var sig_b4c_ub "ub 95% CI, sig_b4c"
        ge sig_b4c_noncover = 0
        replace sig_b4c_noncover = 1 if (`sig_b4c' < sig_b4c_lb | `sig_b4c' > sig_b4c_ub)
        lab var sig_b4c_noncover "1: true sig_b4c not in 95%CI"



* other stats

        ge Nc = `Nc'
        lab var Nc "No. persons per country"

        ge C = `C'
        lab var C "Number of countries"

        ge icc = `icc'
```

```
        lab var icc "icc = var_u/(var_u+var_e)"

        drop lns*
        compress

* }


save, replace

de, fullnames


su b_cons b_age b_agesq  b_cohab  b_nownch b_isced3 b_isced4 b_isced56 b_chexp b_chexpXcohab
b_chexpXnownch

su se_cons se_age se_agesq  se_cohab  se_nownch se_isced3 se_isced4 se_isced56 se_chexp
se_chexpXcohab  se_chexpXnownch

su sig_e sig_u sig_b3c sig_b4c

su *abias

su *rbias

su *noncover

su Nc C icc converged

preserve
keep if converged == 1

su b_cons b_age b_agesq  b_cohab  b_nownch b_isced3 b_isced4 b_isced56 b_chexp b_chexpXcohab
b_chexpXnownch

su se_cons se_age se_agesq  se_cohab  se_nownch se_isced3 se_isced4 se_isced56 se_chexp
se_chexpXcohab  se_chexpXnownch

su sig_e sig_u sig_b3c sig_b4c

su *abias

su *rbias

su *noncover

su Nc C icc

restore


*******************************
log close
```

(c) Basic logit model (mc_part_model1_v01a_Nc_C.do)

```
clear all
set more off
version 11
set mem 500m
capture log close

local Nc = `1'

local C = `2'

local N = `C' * `Nc'


log using mc_partic_model1_v01a_`Nc'_`C'.log, replace


********** Monte-Carlo exercise 5000 replications **************
```

```
*
* Outcome = participation
*
* Model 1: no interaction between individual and country-level variables, and no
*          country-specific random slopes
*
* Use arguments to do file to set parameters of truth model: e.g.
*
* do mc_partic_model1_v01a_Nc_C 1000 10
*
*              where "1000" is # persons/country and "10" is # countries
*
* (1) declare model parameter values
* (2) create data set
* (3) set-up MC (1000 reps)
* (4) post-fitting processing of results
*
*********************************************

*********************************************
* (1) declare model parameter values
*********************************************

di "Number of persons per country, N_c = " `Nc'
di "Number of countries, C   = " `C'
di "Total sample size (Nc x C), N   = " `N'

/*

Woman aged 18-64

Model 3 for particpation:

y*_ic = b0
       + b1 * age_ic
       + b2 * age-squared_ic
       + b3 * cohab_ic
       + b4 * nownch_ic
       + b5 * isced3_ic
       + b6 * isced4_ic
       + b7 * isced56_ic
       + c1 * chexp_c                <-- country=level
       + u_c
       + e_ic

       u_c  ~ N(0, sig_u^2)
       e_ic ~ N(0, sig_e^2)
       cov(u_c, e_ic) = 0


*/

local b0 = -9.1
local b1 = 0.5
local b2 = -0.006
local b3 = 0.02
local b4 = -.27
local b5 = 0.7
local b6 = 0.9
local b7 = 1.4
local c1 = 0.98
local sig_u = .275
local sig_e = sqrt( _pi^2 / 3 )

local icc = (`sig_u')^2 / ( (`sig_u')^2 + (`sig_e')^2  )
di "ICC = " `icc'

************************************************
* (2) create data set (using summary.log stats)
************************************************

set obs `N'

set seed 123456789

* "fixed" part of model (X, Z assumed same/fixed across experiments)
```

```
ge id = _n
ge country = autocode(id, `C', 1, `N')
egen country_id = group(country)
bys country_id: ge tag = _n == 1
compress

* age distribution
* uniform distribution of integers between 18 and 64

ge age = 18+int((64 - 18 +1)*runiform())
su age

ge agegp = .
replace agegp = 1 if age >= 18 & age < 25
replace agegp = 2 if age >= 25 & age < 40
replace agegp = 3 if age >= 40 & age < 50
replace agegp = 4 if age >= 50 & age < 60
replace agegp = 5 if age >= 60 & age < .
lab def agegp 1 "18-24" 2 "25-39" 3 "40-49" 4 "50-59" 5 "60-64"
lab val agegp agegp
ta agegp

ge agesq = age^2

ge cohab = .
ge nownch = .
ge isced012 = 0
ge isced3 = 0
ge isced4 = 0
ge isced56 = 0

lab var cohab "Partnership status
lab def cohab 0 "No partner" 1 "Has parter"
lab val cohab cohab
lab var nownch "# kids"
lab def nownch 0 "0" 1 "1" 2 "2" 3 "3+"
lab val nownch nownch

ge type = .
lab var type "Characteristics combination"
lab def type ///
           1 "Not marr 0 isced0,1,2" ///
           2 "Not marr 0 isced3" ///
           3 "Not marr 0 isced4" ///
           4 "Not marr 0 isced5,6" ///
           5 "Not marr 1 isced0,1,2" ///
           6 "Not marr 1 isced3" ///
           7 "Not marr 1 isced4" ///
           8 "Not marr 1 isced5,6" ///
           9 "Not marr 2 isced0,1,2" ///
          10 "Not marr 2 isced3" ///
          11 "Not marr 2 isced4" ///
          12 "Not marr 2 isced5,6" ///
          13 "Not marr 3+ isced0,1,2" ///
          14 "Not marr 3+ isced3" ///
          15 "Not marr 3+ isced4" ///
          16 "Not marr 3+ isced5,6" ///
          17 "Marr 0 isced0,1,2" ///
          18 "Marr 0 isced3" ///
          19 "Marr 0 isced4" ///
          20 "Marr 0 isced5,6" ///
          21 "Marr 1 isced0,1,2" ///
          22 "Marr 1 isced3" ///
          23 "Marr 1 isced4" ///
          24 "Marr 1 isced5,6" ///
          25 "Marr 2 isced0,1,2" ///
          26 "Marr 2 isced3" ///
          27 "Marr 2 isced4" ///
          28 "Marr 2 isced5,6" ///
          29 "Marr 3+ isced0,1,2" ///
          30 "Marr 3+ isced3" ///
          31 "Marr 3+ isced4" ///
          32 "Marr 3+ isced5,6"
lab val type type

ge double u = 100 * runiform()
```

```
replace type =  1 if agegp == 1 & u <= 24.77
replace type =  2 if agegp == 1 & u > 24.77  &  u <= 72.98
replace type =  3 if agegp == 1 & u > 72.98  &  u <= 75.70
replace type =  4 if agegp == 1 & u > 75.70  &  u <= 82.98
replace type =  5 if agegp == 1 & u > 82.98  &  u <= 83.72
replace type =  6 if agegp == 1 & u > 83.72  &  u <= 84.77
replace type =  7 if agegp == 1 & u > 84.77  &  u <= 84.82
replace type =  8 if agegp == 1 & u > 84.82  &  u <= 84.90
replace type =  9 if agegp == 1 & u > 84.90  &  u <= 85.04
replace type = 10 if agegp == 1 & u > 85.04  &  u <= 85.14
replace type = 11 if agegp == 1 & u > 85.14  &  u <= 85.14
replace type = 12 if agegp == 1 & u > 85.15  &  u <= 85.15
replace type = 13 if agegp == 1 & u > 85.18  &  u <= 85.18
replace type = 14 if agegp == 1 & u > 85.19  &  u <= 85.19
replace type = 15 if agegp == 1 & u > 85.19  &  u <= 85.19
replace type = 16 if agegp == 1 & u > 85.19  &  u <= 85.19
replace type = 17 if agegp == 1 & u > 85.19  &  u <= 86.74
replace type = 18 if agegp == 1 & u > 86.74  &  u <= 92.13
replace type = 19 if agegp == 1 & u > 92.13  &  u <= 92.59
replace type = 20 if agegp == 1 & u > 92.59  &  u <= 94.27
replace type = 21 if agegp == 1 & u > 94.27  &  u <= 95.66
replace type = 22 if agegp == 1 & u > 95.66  &  u <= 98.09
replace type = 23 if agegp == 1 & u > 98.09  &  u <= 98.23
replace type = 24 if agegp == 1 & u > 98.23  &  u <= 98.51
replace type = 25 if agegp == 1 & u > 98.51  &  u <= 99.11
replace type = 26 if agegp == 1 & u > 99.11  &  u <= 99.71
replace type = 27 if agegp == 1 & u > 99.71  &  u <= 99.75
replace type = 28 if agegp == 1 & u > 99.75  &  u <= 99.79
replace type = 29 if agegp == 1 & u > 99.79  &  u <= 99.92
replace type = 30 if agegp == 1 & u > 99.92  &  u <= 99.99
replace type = 31 if agegp == 1 & u > 99.99  &  u <= 99.99
replace type = 32 if agegp == 1 & u > 99.99

replace type =  1 if agegp == 2 & u <= 2.46
replace type =  2 if agegp == 2 & u >  2.46 &  u <= 10.89
replace type =  3 if agegp == 2 & u > 10.89  &  u <= 12.29
replace type =  4 if agegp == 2 & u > 12.29  &  u <= 21.40
replace type =  5 if agegp == 2 & u > 21.40  &  u <= 22.21
replace type =  6 if agegp == 2 & u > 22.21  &  u <= 24.56
replace type =  7 if agegp == 2 & u > 24.56  &  u <= 24.86
replace type =  8 if agegp == 2 & u > 24.86  &  u <= 25.93
replace type =  9 if agegp == 2 & u > 25.93  &  u <= 26.60
replace type = 10 if agegp == 2 & u > 26.60  &  u <= 28.07
replace type = 11 if agegp == 2 & u > 28.07  &  u <= 28.20
replace type = 12 if agegp == 2 & u > 28.20  &  u <= 28.67
replace type = 13 if agegp == 2 & u > 28.67  &  u <= 29.09
replace type = 14 if agegp == 2 & u > 29.09  &  u <= 29.55
replace type = 15 if agegp == 2 & u > 29.55  &  u <= 29.58
replace type = 16 if agegp == 2 & u > 29.58  &  u <= 29.69
replace type = 17 if agegp == 2 & u > 29.69  &  u <= 31.11
replace type = 18 if agegp == 2 & u > 31.11  &  u <= 36.06
replace type = 19 if agegp == 2 & u > 36.06  &  u <= 36.82
replace type = 20 if agegp == 2 & u > 36.82  &  u <= 43.19
replace type = 21 if agegp == 2 & u > 43.19  &  u <= 46.13
replace type = 22 if agegp == 2 & u > 46.13  &  u <= 55.06
replace type = 23 if agegp == 2 & u > 55.06  &  u <= 56.24
replace type = 24 if agegp == 2 & u > 56.24  &  u <= 62.69
replace type = 25 if agegp == 2 & u > 62.69  &  u <= 67.21
replace type = 26 if agegp == 2 & u > 67.21  &  u <= 80.43
replace type = 27 if agegp == 2 & u > 80.43  &  u <= 81.90
replace type = 28 if agegp == 2 & u > 81.90  &  u <= 89.78
replace type = 29 if agegp == 2 & u > 89.78  &  u <= 92.30
replace type = 30 if agegp == 2 & u > 92.30  &  u <= 97.11
replace type = 31 if agegp == 2 & u > 97.11  &  u <= 97.53
replace type = 32 if agegp == 2 & u > 97.53

replace type =  1 if agegp == 3 & u <= 1.82
replace type =  2 if agegp == 3 & u >  1.82 &  u <=  4.85
replace type =  3 if agegp == 3 & u >  4.85 &  u <=  5.39
replace type =  4 if agegp == 3 & u >  5.39 &  u <=  7.75
replace type =  5 if agegp == 3 & u >  7.75 &  u <=  8.98
replace type =  6 if agegp == 3 & u >  8.98  &  u <= 11.65
replace type =  7 if agegp == 3 & u > 11.65  &  u <= 12.12
replace type =  8 if agegp == 3 & u > 12.12  &  u <= 13.70
replace type =  9 if agegp == 3 & u > 13.70  &  u <= 14.64
replace type = 10 if agegp == 3 & u > 14.64  &  u <= 16.78
replace type = 11 if agegp == 3 & u > 16.78  &  u <= 17.13
```

```
replace type = 12 if agegp == 3 & u > 17.13  &  u <= 18.36
replace type = 13 if agegp == 3 & u > 18.36  &  u <= 18.82
replace type = 14 if agegp == 3 & u > 18.82  &  u <= 19.55
replace type = 15 if agegp == 3 & u > 19.55  &  u <= 19.65
replace type = 16 if agegp == 3 & u > 19.65  &  u <= 20.09
replace type = 17 if agegp == 3 & u > 20.09  &  u <= 22.59
replace type = 18 if agegp == 3 & u > 22.59  &  u <= 27.12
replace type = 19 if agegp == 3 & u > 27.12  &  u <= 27.73
replace type = 20 if agegp == 3 & u > 27.73  &  u <= 30.20
replace type = 21 if agegp == 3 & u > 30.20  &  u <= 35.00
replace type = 22 if agegp == 3 & u > 35.00  &  u <= 44.06
replace type = 23 if agegp == 3 & u > 44.06  &  u <= 45.24
replace type = 24 if agegp == 3 & u > 45.24  &  u <= 50.04
replace type = 25 if agegp == 3 & u > 50.04  &  u <= 57.54
replace type = 26 if agegp == 3 & u > 57.54  &  u <= 73.05
replace type = 27 if agegp == 3 & u > 73.05  &  u <= 74.78
replace type = 28 if agegp == 3 & u > 74.78  &  u <= 84.41
replace type = 29 if agegp == 3 & u > 84.41  &  u <= 87.95
replace type = 30 if agegp == 3 & u > 87.95  &  u <= 94.83
replace type = 31 if agegp == 3 & u > 94.83  &  u <= 95.57
replace type = 32 if agegp == 3 & u > 95.57

replace type =  1 if agegp == 4 & u <= 3.98
replace type =  2 if agegp == 4 & u >  3.98  &  u <=  9.32
replace type =  3 if agegp == 4 & u >  9.32 &  u <=  9.98
replace type =  4 if agegp == 4 & u >  9.98 &  u <= 13.24
replace type =  5 if agegp == 4 & u > 13.24  &  u <= 15.27
replace type =  6 if agegp == 4 & u > 15.27  &  u <= 18.19
replace type =  7 if agegp == 4 & u > 18.19  &  u <= 18.54
replace type =  8 if agegp == 4 & u > 18.54  &  u <= 19.97
replace type =  9 if agegp == 4 & u > 19.97  &  u <= 20.78
replace type = 10 if agegp == 4 & u > 20.78  &  u <= 21.77
replace type = 11 if agegp == 4 & u > 21.77  &  u <= 21.89
replace type = 12 if agegp == 4 & u > 21.89  &  u <= 22.43
replace type = 13 if agegp == 4 & u > 22.43  &  u <= 22.65
replace type = 14 if agegp == 4 & u > 22.65  &  u <= 22.85
replace type = 15 if agegp == 4 & u > 22.85  &  u <= 22.87
replace type = 16 if agegp == 4 & u > 22.87  &  u <= 22.97
replace type = 17 if agegp == 4 & u > 22.97  &  u <= 34.87
replace type = 18 if agegp == 4 & u > 34.87  &  u <= 50.67
replace type = 19 if agegp == 4 & u > 50.67  &  u <= 52.23
replace type = 20 if agegp == 4 & u > 52.23  &  u <= 59.29
replace type = 21 if agegp == 4 & u > 59.29  &  u <= 67.70
replace type = 22 if agegp == 4 & u > 67.70  &  u <= 77.60
replace type = 23 if agegp == 4 & u > 77.60  &  u <= 78.64
replace type = 24 if agegp == 4 & u > 78.64  &  u <= 83.15
replace type = 25 if agegp == 4 & u > 83.15  &  u <= 87.63
replace type = 26 if agegp == 4 & u > 87.63  &  u <= 92.71
replace type = 27 if agegp == 4 & u > 92.71  &  u <= 93.22
replace type = 28 if agegp == 4 & u > 93.22  &  u <= 96.35
replace type = 29 if agegp == 4 & u > 96.35  &  u <= 97.65
replace type = 30 if agegp == 4 & u > 97.65  &  u <= 98.96
replace type = 30 if agegp == 4 & u > 98.96  &  u <= 99.10
replace type = 32 if agegp == 4 & u > 98.96

replace type =  1 if agegp == 5 & u <= 7.88
replace type =  2 if agegp == 5 & u >  7.88 &  u <= 15.29
replace type =  3 if agegp == 5 & u > 15.29  &  u <= 16.07
replace type =  4 if agegp == 5 & u > 16.07  &  u <= 19.69
replace type =  5 if agegp == 5 & u > 19.69  &  u <= 22.76
replace type =  6 if agegp == 5 & u > 22.76  &  u <= 24.50
replace type =  7 if agegp == 5 & u > 24.50  &  u <= 24.73
replace type =  8 if agegp == 5 & u > 24.73  &  u <= 25.54
replace type =  9 if agegp == 5 & u > 25.54  &  u <= 26.11
replace type = 10 if agegp == 5 & u > 26.11  &  u <= 26.40
replace type = 11 if agegp == 5 & u > 26.40  &  u <= 26.43
replace type = 12 if agegp == 5 & u > 26.43  &  u <= 26.57
replace type = 13 if agegp == 5 & u > 26.57  &  u <= 26.69
replace type = 14 if agegp == 5 & u > 26.69  &  u <= 26.75
replace type = 15 if agegp == 5 & u > 26.75  &  u <= 26.75
replace type = 16 if agegp == 5 & u > 26.75  &  u <= 26.78
replace type = 17 if agegp == 5 & u > 26.78  &  u <= 49.63
replace type = 18 if agegp == 5 & u > 49.63  &  u <= 70.58
replace type = 19 if agegp == 5 & u > 70.58  &  u <= 72.50
replace type = 20 if agegp == 5 & u > 72.50  &  u <= 81.33
replace type = 21 if agegp == 5 & u > 81.33  &  u <= 89.36
replace type = 22 if agegp == 5 & u > 89.36  &  u <= 93.70
```

```
replace type = 23 if agegp == 5 & u > 93.70  &  u <= 94.24
replace type = 24 if agegp == 5 & u > 94.24  &  u <= 96.06
replace type = 25 if agegp == 5 & u > 96.06  &  u <= 98.01
replace type = 26 if agegp == 5 & u > 98.01  &  u <= 98.90
replace type = 27 if agegp == 5 & u > 98.90  &  u <= 98.94
replace type = 28 if agegp == 5 & u > 98.94  &  u <= 99.35
replace type = 29 if agegp == 5 & u > 99.35  &  u <= 99.77
replace type = 30 if agegp == 5 & u > 99.77  &  u <= 99.90
replace type = 31 if agegp == 5 & u > 99.90  &  u <= 99.93
replace type = 32 if agegp == 5 & u > 99.93

ta type agegp, col missing

replace cohab = 0 if inrange(type, 1, 16)
replace cohab = 1 if inrange(type, 17, 32)

replace nownch = 0 if inrange(type, 1, 4) | inrange(type, 17, 20)
replace nownch = 1 if inrange(type, 5, 8) | inrange(type, 21, 24)
replace nownch = 2 if inrange(type, 9, 12) | inrange(type, 25, 28)
replace nownch = 3 if inrange(type, 13, 16) | inrange(type, 29, 32)

replace isced012 = 1 if inlist(type, 1, 5, 9, 13, 17, 21, 25, 29)
replace isced3   = 1 if inlist(type, 2, 6, 10, 14, 18, 22, 26, 30)
replace isced4         = 1 if inlist(type, 3, 7, 11, 15, 19, 23, 27, 31)
replace isced56  = 1 if inlist(type, 4, 8, 12, 16, 20, 24, 28, 32)

gen edlevel = .
replace edlevel = 1 if isced012 == 1
replace edlevel = 2 if isced3 == 1
replace edlevel = 3 if isced4 == 1
replace edlevel = 4 if isced56 == 1
lab def edlevel 1 "isced0,1,2" 2 "isced3" 3 "isced4" 4 "isced5,6"
lab val edlevel edlevel


* lognormal distribution for child exp. Parameters: m = -0.665, v = 0.535

local m = -0.665
local v = 0.535

ge chexp = exp( `m' + `v' * invnormal( runiform() ) ) if tag
lab var chexp "child care expend, % GDP"
bys country_id: replace chexp = chexp[1]



compress

ge fixed = `b0'                             ///
       + `b1' * age                         ///
       + `b2' * agesq                       ///
       + `b3' * cohab                       ///
       + `b4' * nownch                      ///
       + `b5' * isced3                      ///
       + `b6' * isced4                      ///
       + `b7' * isced56                     ///
       + `c1' * chexp


ta country_id, sum(age)
ta country_id, sum(isced012)
ta country_id, sum(isced3)
ta country_id, sum(isced4)
ta country_id, sum(isced56)
ta country_id, sum(cohab)
ta country_id, sum(nownch)
ta country_id, sum(chexp)

de, fullnames
su
save mc_partic_model1_v01a_`Nc'_`C'_data, replace


************************************************
* (3) set-up MC
************************************************
```

```
cap program drop mc_silcp
program define mc_silcp

        version 11

        args sig_u

        capture drop y u_c e_ic

        gen e_ic =  logit( runiform() )

        gen u_c = rnormal(0, `sig_u') if tag
        bys country_id : replace u_c = u_c[1]

        ge y = cond(fixed + u_c + e_ic > 0, 1, 0 )

                // default estimation method is used (adaptive quadrature; 7 points), cov
structure independent
        xtmelogit y age agesq cohab nownch isced3 isced4 isced56 ///
                 chexp  || country_id: , nolog  iter(250)

end

di "Time is: " c(current_time) " on " c(current_date)
simulate _b _se converged = e(converged) logRLL = e(ll) ///
        , reps(5000) saving(mc_partic_model1_v01a_`Nc'_`C'_output.dta, replace double)  ///
        : mc_silcp  (`sig_u')
di "Time is: " c(current_time) " on " c(current_date)


de, fullnames
ds, varwidth(20)
su

*********************************************
* (4) post-fitting processing of results and saving data
*********************************************


* convert back to original metric; calculate summary stats


* quietly {

        rename _eq5_converged  converged
        rename _eq5_logRLL logRLL

        renpfix eq1_

        ge sig_u = exp(lns1_1_1_b_cons)
        ge sig_u_se = sig_u * lns1_1_1_se_cons


* absolute bias
        ge b_cons_abias = b_cons - `b0'
        lab var b_cons_abias "Absolute bias b0 (cons)"

        ge b_age_abias = b_age - `b1'
        lab var b_age_abias "Absolute bias b1 (age)"

        ge b_agesq_abias = b_agesq - `b2'
        lab var b_agesq_abias "Absolute bias b2 (agesq)"

        ge b_cohab_abias = b_cohab - `b3'
        lab var b_cohab_abias "Absolute bias b3 (cohab)"

        ge b_nownch_abias = b_nownch - `b4'
        lab var b_nownch_abias "Absolute bias b4 (nownch)"

        ge b_isced3_abias = b_isced3 - `b5'
        lab var b_isced3_abias "Absolute bias b5 (isced3)"

        ge b_isced4_abias = b_isced4 - `b6'
        lab var b_isced4_abias "Absolute bias b6 (isced4)"
```

```
	ge b_isced56_abias = b_isced56 - `b7'
	lab var b_isced56_abias "Absolute bias b7 (isced56)"

	ge c_chexp_abias = b_chexp - `c1'
	lab var c_chexp_abias "Absolute bias c1 (chexp)"

	ge sig_u_abias = sig_u - `sig_u'
	lab var sig_u_abias "Absolute bias sig_u"


* relative bias

	ge b_cons_rbias = 100*(b_cons/`b0' - 1)
	lab var b_cons_rbias "% Relative bias b0 (cons)"

	ge b_age_rbias = 100*(b_age/`b1' - 1)
	lab var b_age_rbias "% Relative bias b1 (age)"

	ge b_agesq_rbias = 100*(b_agesq/`b2' - 1)
	lab var b_agesq_rbias "% Relative bias b2 (agesq)"

	ge b_cohab_rbias = 100*(b_cohab/`b3' - 1)
	lab var b_cohab_rbias "% Relative bias b3 (cohab)"

	ge b_nownch_rbias = 100*(b_nownch/`b4' - 1)
	lab var b_nownch_rbias "% Relative bias b4 (nownch)"

	ge b_isced3_rbias = 100*(b_isced3/`b5' - 1)
	lab var b_isced3_rbias "% Relative bias b5 (isced3)"

	ge b_isced4_rbias = 100*(b_isced4/`b6' - 1)
	lab var b_isced4_rbias "% Relative bias b6 (isced4)"

	ge b_isced56_rbias = 100*(b_isced56/`b7' - 1)
	lab var b_isced56_rbias "% Relative bias b7 (isced56)"

	ge c_chexp_rbias = 100*(b_chexp/`c1' - 1)
	lab var c_chexp_rbias "% Relative bias c1 (chexp)"

	ge sig_u_rbias = 100*(sig_u / `sig_u' - 1)
	lab var sig_u_rbias "% Relative bias sig_u"


* non-coverage

	ge b0_lb = b_cons + se_cons*invnormal( (1-.95)/2 )
	ge b0_ub = b_cons - se_cons*invnormal( (1-.95)/2 )
	lab var b0_lb "lb 95% CI, b0 (cons)"
	lab var b0_ub "ub 95% CI, b0 (cons)"
	ge b_cons_noncover = 0
	replace b_cons_noncover = 1 if (`b0' < b0_lb | `b0' > b0_ub)
	lab var b_cons_noncover "1: true b0 not in 95%CI"

	ge b1_lb = b_age + se_age*invnormal( (1-.95)/2 )
	ge b1_ub = b_age - se_age*invnormal( (1-.95)/2 )
	lab var b1_lb "lb 95% CI, b1 (age)"
	lab var b1_ub "ub 95% CI, b1 (age)"
	ge b_age_noncover = 0
	replace b_age_noncover = 1 if (`b1' < b1_lb | `b1' > b1_ub)
	lab var b_age_noncover "1: true b1 not in 95%CI"

	ge b2_lb = b_agesq + se_agesq*invnormal( (1-.95)/2 )
	ge b2_ub = b_agesq - se_agesq*invnormal( (1-.95)/2 )
	lab var b2_lb "lb 95% CI, b2 (agesq)"
	lab var b2_ub "ub 95% CI, b2 (agesq)"
	ge b_agesq_noncover = 0
	replace b_agesq_noncover = 1 if (`b2' < b2_lb | `b2' > b2_ub)
	lab var b_agesq_noncover "1: true b2 not in 95%CI"

	ge b3_lb = b_cohab + se_cohab*invnormal( (1-.95)/2 )
	ge b3_ub = b_cohab - se_cohab*invnormal( (1-.95)/2 )
	lab var b3_lb "lb 95% CI, b0 (cohab)"
	lab var b3_ub "ub 95% CI, b0 (cohab)"
	ge b_cohab_noncover = 0
	replace b_cohab_noncover = 1 if (`b3' < b3_lb | `b3' > b3_ub)
	lab var b_cohab_noncover "1: true b3 not in 95%CI"
```

```
        ge b4_lb = b_nownch + se_nownch*invnormal( (1-.95)/2 )
        ge b4_ub = b_nownch - se_nownch*invnormal( (1-.95)/2 )
        lab var b4_lb "lb 95% CI, b0 (nownch)"
        lab var b4_ub "ub 95% CI, b0 (nownch)"
        ge b_nownch_noncover = 0
        replace b_nownch_noncover = 1 if (`b4' < b4_lb | `b4' > b4_ub)
        lab var b_nownch_noncover "1: true b4 not in 95%CI"

        ge b5_lb = b_isced3 + se_isced3*invnormal( (1-.95)/2 )
        ge b5_ub = b_isced3 - se_isced3*invnormal( (1-.95)/2 )
        lab var b5_lb "lb 95% CI, b0 (isced3)"
        lab var b5_ub "ub 95% CI, b0 (isced3)"
        ge b_isced3_noncover = 0
        replace b_isced3_noncover = 1 if (`b5' < b5_lb | `b5' > b5_ub)
        lab var b_isced3_noncover "1: true b5 not in 95%CI"

        ge b6_lb = b_isced4 + se_isced4*invnormal( (1-.95)/2 )
        ge b6_ub = b_isced4 - se_isced4*invnormal( (1-.95)/2 )
        lab var b6_lb "lb 95% CI, b0 (isced4)"
        lab var b6_ub "ub 95% CI, b0 (isced4)"
        ge b_isced4_noncover = 0
        replace b_isced4_noncover = 1 if (`b6' < b6_lb | `b6' > b6_ub)
        lab var b_isced4_noncover "1: true b6 not in 95%CI"

        ge b7_lb = b_isced56 + se_isced56*invnormal( (1-.95)/2 )
        ge b7_ub = b_isced56 - se_isced56*invnormal( (1-.95)/2 )
        lab var b7_lb "lb 95% CI, b0 (isced56)"
        lab var b7_ub "ub 95% CI, b0 (isced56)"
        ge b_isced56_noncover = 0
        replace b_isced56_noncover = 1 if (`b7' < b7_lb | `b7' > b7_ub)
        lab var b_isced56_noncover "1: true b7 not in 95%CI"

        ge c1_lb = b_chexp + se_chexp*invnormal( (1-.95)/2 )
        ge c1_ub = b_chexp - se_chexp*invnormal( (1-.95)/2 )
        lab var c1_lb "lb 95% CI, c1 (chexp)"
        lab var c1_ub "ub 95% CI, c1 (chexp)"
        ge c_chexp_noncover = 0
        replace c_chexp_noncover = 1 if (`c1' < c1_lb | `c1' > c1_ub)
        lab var c_chexp_noncover "1: true c1 not in 95%CI"

        ge sig_u_lb = sig_u + sig_u_se*invnormal( (1-.95)/2 )
        ge sig_u_ub = sig_u - sig_u_se*invnormal( (1-.95)/2 )
        lab var sig_u_lb "lb 95% CI, sig_u"
        lab var sig_u_ub "ub 95% CI, sig_u"
        ge sig_u_noncover = 0
        replace sig_u_noncover = 1 if (`sig_u' < sig_u_lb | `sig_u' > sig_u_ub)
        lab var sig_u_noncover "1: true sig_u not in 95%CI"


* other stats

        ge Nc = `Nc'
        lab var Nc "No. persons per country"

        ge C = `C'
        lab var C "Number of countries"

        ge icc = `icc'
        lab var icc "icc = var_u/(var_u+var_e)"

        drop lns*
        compress

* }


save, replace

de, fullnames


su b_cons b_age b_agesq  b_cohab  b_nownch b_isced3 b_isced4 b_isced56 b_chexp

su se_cons se_age se_agesq  se_cohab  se_nownch se_isced3 se_isced4 se_isced56 se_chexp

su sig_u
```

```
su *abias

su *rbias

su *noncover

su Nc C icc converged

preserve
keep if converged == 1

su b_cons b_age b_agesq  b_cohab  b_nownch b_isced3 b_isced4 b_isced56 b_chexp

su se_cons se_age se_agesq  se_cohab  se_nownch se_isced3 se_isced4 se_isced56 se_chexp

su sig_u

su *abias

su *rbias

su *noncover

su Nc C icc

restore

*******************************
log close
```

(d) Extended logit model (mc_part_model3_v01_Nc_C.do)

```
clear all
set more off
version 11
set mem 500m
capture log close

local Nc = `1'

local C = `2'

local N = `C' * `Nc'


log using mc_partic_model3_v01_`Nc'_`C'.log, replace


********** Monte-Carlo exercise **************
*
* Outcome = participation
*
* Model 3: interaction between individual and country-level variables, and
*          country-specific random slopes
*
* Use arguments to do file to set parameters of truth model: e.g.
*
* do mc_partic_model3_v01_Nc_C 1000 10
*
*              where "1000" is # persons/country and "10" is # countries
*
* (1) declare model parameter values
* (2) create data set
* (3) set-up MC (1000 reps)
* (4) post-fitting processing of results
*
*********************************************

*********************************************
* (1) declare model parameter values
*********************************************

di "Number of persons per country, N_c = " `Nc'
```

```
di "Number of countries, C   = " `C'
di "Total sample size (Nc x C), N   = " `N'

/*

Woman aged 18-64

Model 3 for particpation:

y*_ic = b0
        + b1 * age_ic
        + b2 * age-squared_ic
        + b3 * cohab_ic    +  b3c * cohab_ic  <--- random slope
        + b4 * nownch_ic   +  b4c * nownch_ic   <--- random slope
        + b5 * isced3_ic
        + b6 * isced4_ic
        + b7 * isced56_ic
        + c1 * chexp_c                <-- country=level
        + c2 * (chexp_c X cohab_ic)
        + c3 * (chexp_c X nownch_ic)
        + u_c
        + e_ic

        u_c  ~ N(0, sig_u^2)
        e_ic ~ N(0, sig_e^2)
        cov(u_c, e_ic) = 0

        b3c ~ N(0, sig_b3c^2)
        b4c ~ N(0, sig_b4c^2)

*/



local b0 = -9.1
local b1 = 0.5
local b2 = -0.006
local b3 = 0.02
local b4 = -.27
local b5 = 0.7
local b6 = 0.9
local b7 = 1.4
local c1 = 0.7
local c2 = 0.6
local c3 = -0.1
local sig_u = 0.38
local sig_e = sqrt( _pi^2 / 3 )

local sig_b3c = 0.25
local sig_b4c = 0.13

local icc = (`sig_u')^2 / ( (`sig_u')^2 + (`sig_e')^2  )
di "ICC = " `icc'

************************************************
* (2) create data set (using summary.log stats)
************************************************

set obs `N'

set seed 123456789

* "fixed" part of model (X, Z assumed same/fixed across experiments)

ge id = _n
ge country = autocode(id, `C', 1, `N')
egen country_id = group(country)
bys country_id: ge tag = _n == 1
compress

* age distribution
* uniform distribution of integers between 18 and 64

ge age = 18+int((64 - 18 +1)*runiform())
su age

ge agegp = .
```

```
replace agegp = 1 if age >= 18 & age < 25
replace agegp = 2 if age >= 25 & age < 40
replace agegp = 3 if age >= 40 & age < 50
replace agegp = 4 if age >= 50 & age < 60
replace agegp = 5 if age >= 60 & age < .
lab def agegp 1 "18-24" 2 "25-39" 3 "40-49" 4 "50-59" 5 "60-64"
lab val agegp agegp
ta agegp

ge agesq = age^2

ge cohab = .
ge nownch = .
ge isced012 = 0
ge isced3 = 0
ge isced4 = 0
ge isced56 = 0

lab var cohab "Partnership status
lab def cohab 0 "No partner" 1 "Has parter"
lab val cohab cohab
lab var nownch "# kids"
lab def nownch 0 "0" 1 "1" 2 "2" 3 "3+"
lab val nownch nownch

ge type = .
lab var type "Characteristics combination"
lab def type ///
           1 "Not marr 0 isced0,1,2" ///
           2 "Not marr 0 isced3" ///
           3 "Not marr 0 isced4" ///
           4 "Not marr 0 isced5,6" ///
           5 "Not marr 1 isced0,1,2" ///
           6 "Not marr 1 isced3" ///
           7 "Not marr 1 isced4" ///
           8 "Not marr 1 isced5,6" ///
           9 "Not marr 2 isced0,1,2" ///
          10 "Not marr 2 isced3" ///
          11 "Not marr 2 isced4" ///
          12 "Not marr 2 isced5,6" ///
          13 "Not marr 3+ isced0,1,2" ///
          14 "Not marr 3+ isced3" ///
          15 "Not marr 3+ isced4" ///
          16 "Not marr 3+ isced5,6" ///
          17 "Marr 0 isced0,1,2" ///
          18 "Marr 0 isced3" ///
          19 "Marr 0 isced4" ///
          20 "Marr 0 isced5,6" ///
          21 "Marr 1 isced0,1,2" ///
          22 "Marr 1 isced3" ///
          23 "Marr 1 isced4" ///
          24 "Marr 1 isced5,6" ///
          25 "Marr 2 isced0,1,2" ///
          26 "Marr 2 isced3" ///
          27 "Marr 2 isced4" ///
          28 "Marr 2 isced5,6" ///
          29 "Marr 3+ isced0,1,2" ///
          30 "Marr 3+ isced3" ///
          31 "Marr 3+ isced4" ///
          32 "Marr 3+ isced5,6"
lab val type type

ge double u = 100 * runiform()

replace type =  1 if agegp == 1 & u <= 24.77
replace type =  2 if agegp == 1 & u > 24.77  &  u <= 72.98
replace type =  3 if agegp == 1 & u > 72.98  &  u <= 75.70
replace type =  4 if agegp == 1 & u > 75.70  &  u <= 82.98
replace type =  5 if agegp == 1 & u > 82.98  &  u <= 83.72
replace type =  6 if agegp == 1 & u > 83.72  &  u <= 84.77
replace type =  7 if agegp == 1 & u > 84.77  &  u <= 84.82
replace type =  8 if agegp == 1 & u > 84.82  &  u <= 84.90
replace type =  9 if agegp == 1 & u > 84.90  &  u <= 85.04
replace type = 10 if agegp == 1 & u > 85.04  &  u <= 85.14
replace type = 11 if agegp == 1 & u > 85.14  &  u <= 85.14
replace type = 12 if agegp == 1 & u > 85.15  &  u <= 85.15
replace type = 13 if agegp == 1 & u > 85.18  &  u <= 85.18
```

```
replace type = 14 if agegp == 1 & u > 85.19  &  u <= 85.19
replace type = 15 if agegp == 1 & u > 85.19  &  u <= 85.19
replace type = 16 if agegp == 1 & u > 85.19  &  u <= 85.19
replace type = 17 if agegp == 1 & u > 85.19  &  u <= 86.74
replace type = 18 if agegp == 1 & u > 86.74  &  u <= 92.13
replace type = 19 if agegp == 1 & u > 92.13  &  u <= 92.59
replace type = 20 if agegp == 1 & u > 92.59  &  u <= 94.27
replace type = 21 if agegp == 1 & u > 94.27  &  u <= 95.66
replace type = 22 if agegp == 1 & u > 95.66  &  u <= 98.09
replace type = 23 if agegp == 1 & u > 98.09  &  u <= 98.23
replace type = 24 if agegp == 1 & u > 98.23  &  u <= 98.51
replace type = 25 if agegp == 1 & u > 98.51  &  u <= 99.11
replace type = 26 if agegp == 1 & u > 99.11  &  u <= 99.71
replace type = 27 if agegp == 1 & u > 99.71  &  u <= 99.75
replace type = 28 if agegp == 1 & u > 99.75  &  u <= 99.79
replace type = 29 if agegp == 1 & u > 99.79  &  u <= 99.92
replace type = 30 if agegp == 1 & u > 99.92  &  u <= 99.99
replace type = 31 if agegp == 1 & u > 99.99  &  u <= 99.99
replace type = 32 if agegp == 1 & u > 99.99

replace type =  1 if agegp == 2 & u <= 2.46
replace type =  2 if agegp == 2 & u >  2.46 &  u <= 10.89
replace type =  3 if agegp == 2 & u > 10.89  &  u <= 12.29
replace type =  4 if agegp == 2 & u > 12.29  &  u <= 21.40
replace type =  5 if agegp == 2 & u > 21.40  &  u <= 22.21
replace type =  6 if agegp == 2 & u > 22.21  &  u <= 24.56
replace type =  7 if agegp == 2 & u > 24.56  &  u <= 24.86
replace type =  8 if agegp == 2 & u > 24.86  &  u <= 25.93
replace type =  9 if agegp == 2 & u > 25.93  &  u <= 26.60
replace type = 10 if agegp == 2 & u > 26.60  &  u <= 28.07
replace type = 11 if agegp == 2 & u > 28.07  &  u <= 28.20
replace type = 12 if agegp == 2 & u > 28.20  &  u <= 28.67
replace type = 13 if agegp == 2 & u > 28.67  &  u <= 29.09
replace type = 14 if agegp == 2 & u > 29.09  &  u <= 29.55
replace type = 15 if agegp == 2 & u > 29.55  &  u <= 29.58
replace type = 16 if agegp == 2 & u > 29.58  &  u <= 29.69
replace type = 17 if agegp == 2 & u > 29.69  &  u <= 31.11
replace type = 18 if agegp == 2 & u > 31.11  &  u <= 36.06
replace type = 19 if agegp == 2 & u > 36.06  &  u <= 36.82
replace type = 20 if agegp == 2 & u > 36.82  &  u <= 43.19
replace type = 21 if agegp == 2 & u > 43.19  &  u <= 46.13
replace type = 22 if agegp == 2 & u > 46.13  &  u <= 55.06
replace type = 23 if agegp == 2 & u > 55.06  &  u <= 56.24
replace type = 24 if agegp == 2 & u > 56.24  &  u <= 62.69
replace type = 25 if agegp == 2 & u > 62.69  &  u <= 67.21
replace type = 26 if agegp == 2 & u > 67.21  &  u <= 80.43
replace type = 27 if agegp == 2 & u > 80.43  &  u <= 81.90
replace type = 28 if agegp == 2 & u > 81.90  &  u <= 89.78
replace type = 29 if agegp == 2 & u > 89.78  &  u <= 92.30
replace type = 30 if agegp == 2 & u > 92.30  &  u <= 97.11
replace type = 31 if agegp == 2 & u > 97.11  &  u <= 97.53
replace type = 32 if agegp == 2 & u > 97.53

replace type =  1 if agegp == 3 & u <= 1.82
replace type =  2 if agegp == 3 & u >  1.82 &  u <=  4.85
replace type =  3 if agegp == 3 & u >  4.85 &  u <=  5.39
replace type =  4 if agegp == 3 & u >  5.39 &  u <=  7.75
replace type =  5 if agegp == 3 & u >  7.75 &  u <=  8.98
replace type =  6 if agegp == 3 & u >  8.98  &  u <= 11.65
replace type =  7 if agegp == 3 & u > 11.65  &  u <= 12.12
replace type =  8 if agegp == 3 & u > 12.12  &  u <= 13.70
replace type =  9 if agegp == 3 & u > 13.70  &  u <= 14.64
replace type = 10 if agegp == 3 & u > 14.64  &  u <= 16.78
replace type = 11 if agegp == 3 & u > 16.78  &  u <= 17.13
replace type = 12 if agegp == 3 & u > 17.13  &  u <= 18.36
replace type = 13 if agegp == 3 & u > 18.36  &  u <= 18.82
replace type = 14 if agegp == 3 & u > 18.82  &  u <= 19.55
replace type = 15 if agegp == 3 & u > 19.55  &  u <= 19.65
replace type = 16 if agegp == 3 & u > 19.65  &  u <= 20.09
replace type = 17 if agegp == 3 & u > 20.09  &  u <= 22.59
replace type = 18 if agegp == 3 & u > 22.59  &  u <= 27.12
replace type = 19 if agegp == 3 & u > 27.12  &  u <= 27.73
replace type = 20 if agegp == 3 & u > 27.73  &  u <= 30.20
replace type = 21 if agegp == 3 & u > 30.20  &  u <= 35.00
replace type = 22 if agegp == 3 & u > 35.00  &  u <= 44.06
replace type = 23 if agegp == 3 & u > 44.06  &  u <= 45.24
replace type = 24 if agegp == 3 & u > 45.24  &  u <= 50.04
```

```
replace type = 25 if agegp == 3 & u > 50.04  &  u <= 57.54
replace type = 26 if agegp == 3 & u > 57.54  &  u <= 73.05
replace type = 27 if agegp == 3 & u > 73.05  &  u <= 74.78
replace type = 28 if agegp == 3 & u > 74.78  &  u <= 84.41
replace type = 29 if agegp == 3 & u > 84.41  &  u <= 87.95
replace type = 30 if agegp == 3 & u > 87.95  &  u <= 94.83
replace type = 31 if agegp == 3 & u > 94.83  &  u <= 95.57
replace type = 32 if agegp == 3 & u > 95.57

replace type =  1 if agegp == 4 & u <= 3.98
replace type =  2 if agegp == 4 & u >  3.98  &  u <=  9.32
replace type =  3 if agegp == 4 & u >  9.32 &  u <=  9.98
replace type =  4 if agegp == 4 & u >  9.98 &  u <= 13.24
replace type =  5 if agegp == 4 & u > 13.24  &  u <= 15.27
replace type =  6 if agegp == 4 & u > 15.27  &  u <= 18.19
replace type =  7 if agegp == 4 & u > 18.19  &  u <= 18.54
replace type =  8 if agegp == 4 & u > 18.54  &  u <= 19.97
replace type =  9 if agegp == 4 & u > 19.97  &  u <= 20.78
replace type = 10 if agegp == 4 & u > 20.78  &  u <= 21.77
replace type = 11 if agegp == 4 & u > 21.77  &  u <= 21.89
replace type = 12 if agegp == 4 & u > 21.89  &  u <= 22.43
replace type = 13 if agegp == 4 & u > 22.43  &  u <= 22.65
replace type = 14 if agegp == 4 & u > 22.65  &  u <= 22.85
replace type = 15 if agegp == 4 & u > 22.85  &  u <= 22.87
replace type = 16 if agegp == 4 & u > 22.87  &  u <= 22.97
replace type = 17 if agegp == 4 & u > 22.97  &  u <= 34.87
replace type = 18 if agegp == 4 & u > 34.87  &  u <= 50.67
replace type = 19 if agegp == 4 & u > 50.67  &  u <= 52.23
replace type = 20 if agegp == 4 & u > 52.23  &  u <= 59.29
replace type = 21 if agegp == 4 & u > 59.29  &  u <= 67.70
replace type = 22 if agegp == 4 & u > 67.70  &  u <= 77.60
replace type = 23 if agegp == 4 & u > 77.60  &  u <= 78.64
replace type = 24 if agegp == 4 & u > 78.64  &  u <= 83.15
replace type = 25 if agegp == 4 & u > 83.15  &  u <= 87.63
replace type = 26 if agegp == 4 & u > 87.63  &  u <= 92.71
replace type = 27 if agegp == 4 & u > 92.71  &  u <= 93.22
replace type = 28 if agegp == 4 & u > 93.22  &  u <= 96.35
replace type = 29 if agegp == 4 & u > 96.35  &  u <= 97.65
replace type = 30 if agegp == 4 & u > 97.65  &  u <= 98.96
replace type = 30 if agegp == 4 & u > 98.96  &  u <= 99.10
replace type = 32 if agegp == 4 & u > 98.96

replace type =  1 if agegp == 5 & u <= 7.88
replace type =  2 if agegp == 5 & u >  7.88 &  u <= 15.29
replace type =  3 if agegp == 5 & u > 15.29  &  u <= 16.07
replace type =  4 if agegp == 5 & u > 16.07  &  u <= 19.69
replace type =  5 if agegp == 5 & u > 19.69  &  u <= 22.76
replace type =  6 if agegp == 5 & u > 22.76  &  u <= 24.50
replace type =  7 if agegp == 5 & u > 24.50  &  u <= 24.73
replace type =  8 if agegp == 5 & u > 24.73  &  u <= 25.54
replace type =  9 if agegp == 5 & u > 25.54  &  u <= 26.11
replace type = 10 if agegp == 5 & u > 26.11  &  u <= 26.40
replace type = 11 if agegp == 5 & u > 26.40  &  u <= 26.43
replace type = 12 if agegp == 5 & u > 26.43  &  u <= 26.57
replace type = 13 if agegp == 5 & u > 26.57  &  u <= 26.69
replace type = 14 if agegp == 5 & u > 26.69  &  u <= 26.75
replace type = 15 if agegp == 5 & u > 26.75  &  u <= 26.75
replace type = 16 if agegp == 5 & u > 26.75  &  u <= 26.78
replace type = 17 if agegp == 5 & u > 26.78  &  u <= 49.63
replace type = 18 if agegp == 5 & u > 49.63  &  u <= 70.58
replace type = 19 if agegp == 5 & u > 70.58  &  u <= 72.50
replace type = 20 if agegp == 5 & u > 72.50  &  u <= 81.33
replace type = 21 if agegp == 5 & u > 81.33  &  u <= 89.36
replace type = 22 if agegp == 5 & u > 89.36  &  u <= 93.70
replace type = 23 if agegp == 5 & u > 93.70  &  u <= 94.24
replace type = 24 if agegp == 5 & u > 94.24  &  u <= 96.06
replace type = 25 if agegp == 5 & u > 96.06  &  u <= 98.01
replace type = 26 if agegp == 5 & u > 98.01  &  u <= 98.90
replace type = 27 if agegp == 5 & u > 98.90  &  u <= 98.94
replace type = 28 if agegp == 5 & u > 98.94  &  u <= 99.35
replace type = 29 if agegp == 5 & u > 99.35  &  u <= 99.77
replace type = 30 if agegp == 5 & u > 99.77  &  u <= 99.90
replace type = 31 if agegp == 5 & u > 99.90  &  u <= 99.93
replace type = 32 if agegp == 5 & u > 99.93

ta type agegp, col missing
```

```
replace cohab = 0 if inrange(type, 1, 16)
replace cohab = 1 if inrange(type, 17, 32)

replace nownch = 0 if inrange(type, 1, 4) | inrange(type, 17, 20)
replace nownch = 1 if inrange(type, 5, 8) | inrange(type, 21, 24)
replace nownch = 2 if inrange(type, 9, 12) | inrange(type, 25, 28)
replace nownch = 3 if inrange(type, 13, 16) | inrange(type, 29, 32)

replace isced012 = 1 if inlist(type, 1, 5, 9, 13, 17, 21, 25, 29)
replace isced3   = 1 if inlist(type, 2, 6, 10, 14, 18, 22, 26, 30)
replace isced4  = 1 if inlist(type, 3, 7, 11, 15, 19, 23, 27, 31)
replace isced56  = 1 if inlist(type, 4, 8, 12, 16, 20, 24, 28, 32)

gen edlevel = .
replace edlevel = 1 if isced012 == 1
replace edlevel = 2 if isced3 == 1
replace edlevel = 3 if isced4 == 1
replace edlevel = 4 if isced56 == 1
lab def edlevel 1 "isced0,1,2" 2 "isced3" 3 "isced4" 4 "isced5,6"
lab val edlevel edlevel


* lognormal distribution for child exp. Parameters: m = -0.665, v = 0.535

local m = -0.665
local v = 0.535

ge chexp = exp( `m' + `v' * invnormal( runiform() ) ) if tag
lab var chexp "child care expend, % GDP"
bys country_id: replace chexp = chexp[1]

ge chexpXcohab = chexp * cohab
ge chexpXnownch = chexp * nownch

lab var chexpXcohab "chexp,cohab interaction"
lab var chexpXnownch "chexp,nownch interaction"

compress

ge fixed = `b0'                              ///
       + `b1' * age                          ///
       + `b2' * agesq                        ///
       + `b3' * cohab                        ///
       + `b4' * nownch                       ///
       + `b5' * isced3                       ///
       + `b6' * isced4                       ///
       + `b7' * isced56                      ///
       + `c1' * chexp                        ///
       + `c2' * chexpXcohab                  ///
       + `c3' * chexpXnownch


ta country_id, sum(age)
ta country_id, sum(isced012)
ta country_id, sum(isced3)
ta country_id, sum(isced4)
ta country_id, sum(isced56)
ta country_id, sum(cohab)
ta country_id, sum(nownch)
ta country_id, sum(chexp)

de, fullnames
su
save mc_partic_model3_v01_`Nc'_`C'_data, replace


************************************************
* (3) set-up MC
************************************************



cap program drop mc_silcp
program define mc_silcp

       version 11
```

```
        args sig_u sig_b3c sig_b4c

        capture drop y u_c b3c b4c e_ic

        gen e_ic =  logit( runiform() )

        gen u_c = rnormal(0, `sig_u') if tag
        bys country_id : replace u_c = u_c[1]

        gen b3c = rnormal(0, `sig_b3c') if tag
        bys country_id: replace b3c = b3c[1]

        gen b4c = rnormal(0, `sig_b4c') if tag
        bys country_id: replace b4c = b4c[1]

        ge y = cond(fixed + u_c + b3c*cohab + b4c*nownch + e_ic > 0, 1, 0 )

// default estimation method is used (adaptive quadrature; 7 points), cov structure
independent
        xtmelogit y age agesq cohab nownch isced3 isced4 isced56 ///
                chexp chexpXcohab chexpXnownch || country_id: cohab nownch , nolog  iter(250)

end

di "Time is: " c(current_time) " on " c(current_date)
simulate _b _se converged = e(converged) logRLL = e(ll) ///
        , reps(1000) saving(mc_partic_model3_v01_`Nc'_`C'_output.dta, replace double)  ///
        : mc_silcp (`sig_u') (`sig_b3c') (`sig_b4c')
di "Time is: " c(current_time) " on " c(current_date)


de, fullnames
ds, varwidth(20)
su

*********************************************
* (4) post-fitting processing of results and saving data
*********************************************


* convert back to original metric; calculate summary stats

/*

lns1_1_1 sd(cohab) = sig_b3c
lns1_1_2 sd(nownch) = sig_b4c
lns1_1_3 sig_u

*/

* quietly {

        rename _eq9_converged  converged
        rename _eq9_logRLL logRLL

        renpfix eq1_

        ge sig_b3c = exp(lns1_1_1_b_cons)
        ge sig_b3c_se = sig_b3c * lns1_1_1_se_cons

        ge sig_b4c = exp(lns1_1_2_b_cons)
        ge sig_b4c_se = sig_b4c * lns1_1_2_se_cons

        ge sig_u = exp(lns1_1_3_b_cons)
        ge sig_u_se = sig_u * lns1_1_3_se_cons

* absolute bias
        ge b_cons_abias = b_cons - `b0'
        lab var b_cons_abias "Absolute bias b0 (cons)"

        ge b_age_abias = b_age - `b1'
        lab var b_age_abias "Absolute bias b1 (age)"

        ge b_agesq_abias = b_agesq - `b2'
        lab var b_agesq_abias "Absolute bias b2 (agesq)"

        ge b_cohab_abias = b_cohab - `b3'
```

```
        lab var b_cohab_abias "Absolute bias b3 (cohab)"

        ge b_nownch_abias = b_nownch - `b4'
        lab var b_nownch_abias "Absolute bias b4 (nownch)"

        ge b_isced3_abias = b_isced3 - `b5'
        lab var b_isced3_abias "Absolute bias b5 (isced3)"

        ge b_isced4_abias = b_isced4 - `b6'
        lab var b_isced4_abias "Absolute bias b6 (isced4)"

        ge b_isced56_abias = b_isced56 - `b7'
        lab var b_isced56_abias "Absolute bias b7 (isced56)"

        ge c_chexp_abias = b_chexp - `c1'
        lab var c_chexp_abias "Absolute bias c1 (chexp)"

        ge c_chexpXcohab_abias = b_chexpXcohab - `c2'
        lab var c_chexpXcohab_abias "Absolute bias c2 (chexpXcohab)"

        ge c_chexpXnownch_abias = b_chexpXnownch - `c3'
        lab var c_chexpXnownch_abias "Absolute bias c3 (chexpXnownch)"

        ge sig_u_abias = sig_u - `sig_u'
        lab var sig_u_abias "Absolute bias sig_u"

        ge sig_b3c_abias = sig_b3c - `sig_b3c'
        lab var sig_b3c_abias "Absolute bias sig_b3c"

        ge sig_b4c_abias = sig_b4c - `sig_b4c'
        lab var sig_b4c_abias "Absolute bias sig_b4c"

* relative bias

        ge b_cons_rbias = 100*(b_cons/`b0' - 1)
        lab var b_cons_rbias "% Relative bias b0 (cons)"

        ge b_age_rbias = 100*(b_age/`b1' - 1)
        lab var b_age_rbias "% Relative bias b1 (age)"

        ge b_agesq_rbias = 100*(b_agesq/`b2' - 1)
        lab var b_agesq_rbias "% Relative bias b2 (agesq)"

        ge b_cohab_rbias = 100*(b_cohab/`b3' - 1)
        lab var b_cohab_rbias "% Relative bias b3 (cohab)"

        ge b_nownch_rbias = 100*(b_nownch/`b4' - 1)
        lab var b_nownch_rbias "% Relative bias b4 (nownch)"

        ge b_isced3_rbias = 100*(b_isced3/`b5' - 1)
        lab var b_isced3_rbias "% Relative bias b5 (isced3)"

        ge b_isced4_rbias = 100*(b_isced4/`b6' - 1)
        lab var b_isced4_rbias "% Relative bias b6 (isced4)"

        ge b_isced56_rbias = 100*(b_isced56/`b7' - 1)
        lab var b_isced56_rbias "% Relative bias b7 (isced56)"

        ge c_chexp_rbias = 100*(b_chexp/`c1' - 1)
        lab var c_chexp_rbias "% Relative bias c1 (chexp)"

        ge c_chexpXcohab_rbias = 100*(b_chexpXcohab/`c2' - 1)
        lab var c_chexpXcohab_rbias "% Relative bias c2 (chexpXcohab)"

        ge c_chexpXnownch_rbias = 100*(b_chexpXnownch/`c3' - 1)
        lab var c_chexpXnownch_rbias "% Relative bias c3 (chexpXnownch)"

        ge sig_u_rbias = 100*(sig_u / `sig_u' - 1)
        lab var sig_u_rbias "% Relative bias sig_u"

        ge sig_b3c_rbias = 100*(sig_b3c / `sig_b3c' - 1)
        lab var sig_b3c_rbias "% Relative bias sig_b3c"

        ge sig_b4c_rbias = 100*(sig_b3c / `sig_b4c' - 1)
        lab var sig_b4c_rbias "% Relative bias sig_b4c"
```

```
* non-coverage

        ge b0_lb = b_cons + se_cons*invnormal( (1-.95)/2 )
        ge b0_ub = b_cons - se_cons*invnormal( (1-.95)/2 )
        lab var b0_lb "lb 95% CI, b0 (cons)"
        lab var b0_ub "ub 95% CI, b0 (cons)"
        ge b_cons_noncover = 0
        replace b_cons_noncover = 1 if (`b0' < b0_lb | `b0' > b0_ub)
        lab var b_cons_noncover "1: true b0 not in 95%CI"

        ge b1_lb = b_age + se_age*invnormal( (1-.95)/2 )
        ge b1_ub = b_age - se_age*invnormal( (1-.95)/2 )
        lab var b1_lb "lb 95% CI, b1 (age)"
        lab var b1_ub "ub 95% CI, b1 (age)"
        ge b_age_noncover = 0
        replace b_age_noncover = 1 if (`b1' < b1_lb | `b1' > b1_ub)
        lab var b_age_noncover "1: true b1 not in 95%CI"

        ge b2_lb = b_agesq + se_agesq*invnormal( (1-.95)/2 )
        ge b2_ub = b_agesq - se_agesq*invnormal( (1-.95)/2 )
        lab var b2_lb "lb 95% CI, b2 (agesq)"
        lab var b2_ub "ub 95% CI, b2 (agesq)"
        ge b_agesq_noncover = 0
        replace b_agesq_noncover = 1 if (`b2' < b2_lb | `b2' > b2_ub)
        lab var b_agesq_noncover "1: true b2 not in 95%CI"

        ge b3_lb = b_cohab + se_cohab*invnormal( (1-.95)/2 )
        ge b3_ub = b_cohab - se_cohab*invnormal( (1-.95)/2 )
        lab var b3_lb "lb 95% CI, b0 (cohab)"
        lab var b3_ub "ub 95% CI, b0 (cohab)"
        ge b_cohab_noncover = 0
        replace b_cohab_noncover = 1 if (`b3' < b3_lb | `b3' > b3_ub)
        lab var b_cohab_noncover "1: true b3 not in 95%CI"

        ge b4_lb = b_nownch + se_nownch*invnormal( (1-.95)/2 )
        ge b4_ub = b_nownch - se_nownch*invnormal( (1-.95)/2 )
        lab var b4_lb "lb 95% CI, b0 (nownch)"
        lab var b4_ub "ub 95% CI, b0 (nownch)"
        ge b_nownch_noncover = 0
        replace b_nownch_noncover = 1 if (`b4' < b4_lb | `b4' > b4_ub)
        lab var b_nownch_noncover "1: true b4 not in 95%CI"

        ge b5_lb = b_isced3 + se_isced3*invnormal( (1-.95)/2 )
        ge b5_ub = b_isced3 - se_isced3*invnormal( (1-.95)/2 )
        lab var b5_lb "lb 95% CI, b0 (isced3)"
        lab var b5_ub "ub 95% CI, b0 (isced3)"
        ge b_isced3_noncover = 0
        replace b_isced3_noncover = 1 if (`b5' < b5_lb | `b5' > b5_ub)
        lab var b_isced3_noncover "1: true b5 not in 95%CI"

        ge b6_lb = b_isced4 + se_isced4*invnormal( (1-.95)/2 )
        ge b6_ub = b_isced4 - se_isced4*invnormal( (1-.95)/2 )
        lab var b6_lb "lb 95% CI, b0 (isced4)"
        lab var b6_ub "ub 95% CI, b0 (isced4)"
        ge b_isced4_noncover = 0
        replace b_isced4_noncover = 1 if (`b6' < b6_lb | `b6' > b6_ub)
        lab var b_isced4_noncover "1: true b6 not in 95%CI"

        ge b7_lb = b_isced56 + se_isced56*invnormal( (1-.95)/2 )
        ge b7_ub = b_isced56 - se_isced56*invnormal( (1-.95)/2 )
        lab var b7_lb "lb 95% CI, b0 (isced56)"
        lab var b7_ub "ub 95% CI, b0 (isced56)"
        ge b_isced56_noncover = 0
        replace b_isced56_noncover = 1 if (`b7' < b7_lb | `b7' > b7_ub)
        lab var b_isced56_noncover "1: true b7 not in 95%CI"

        ge c1_lb = b_chexp + se_chexp*invnormal( (1-.95)/2 )
        ge c1_ub = b_chexp - se_chexp*invnormal( (1-.95)/2 )
        lab var c1_lb "lb 95% CI, c1 (chexp)"
        lab var c1_ub "ub 95% CI, c1 (chexp)"
        ge c_chexp_noncover = 0
        replace c_chexp_noncover = 1 if (`c1' < c1_lb | `c1' > c1_ub)
        lab var c_chexp_noncover "1: true c1 not in 95%CI"

        ge c2_lb = b_chexpXcohab + se_chexpXcohab*invnormal( (1-.95)/2 )
        ge c2_ub = b_chexpXcohab - se_chexpXcohab*invnormal( (1-.95)/2 )
```

```
        lab var c2_lb "lb 95% CI, c2 (chexpXcohab)"
        lab var c2_ub "ub 95% CI, c2 (chexpXcohab)"
        ge c_chexpXcohab_noncover = 0
        replace c_chexpXcohab_noncover = 1 if (`c2' < c2_lb | `c2' > c2_ub)
        lab var c_chexpXcohab_noncover "1: true c2 not in 95%CI"

        ge c3_lb = b_chexpXnownch + se_chexpXnownch*invnormal( (1-.95)/2 )
        ge c3_ub = b_chexpXnownch - se_chexpXnownch*invnormal( (1-.95)/2 )
        lab var c3_lb "lb 95% CI, c3 (chexpXnownch)"
        lab var c3_ub "ub 95% CI, c3 (chexpXnownch)"
        ge c_chexpXnownch_noncover = 0
        replace c_chexpXnownch_noncover = 1 if (`c3' < c3_lb | `c3' > c3_ub)
        lab var c_chexpXnownch_noncover "1: true c3 not in 95%CI"

        ge sig_u_lb = sig_u + sig_u_se*invnormal( (1-.95)/2 )
        ge sig_u_ub = sig_u - sig_u_se*invnormal( (1-.95)/2 )
        lab var sig_u_lb "lb 95% CI, sig_u"
        lab var sig_u_ub "ub 95% CI, sig_u"
        ge sig_u_noncover = 0
        replace sig_u_noncover = 1 if (`sig_u' < sig_u_lb | `sig_u' > sig_u_ub)
        lab var sig_u_noncover "1: true sig_u not in 95%CI"

        ge sig_b3c_lb = sig_b3c + sig_b3c_se*invnormal( (1-.95)/2 )
        ge sig_b3c_ub = sig_b3c - sig_b3c_se*invnormal( (1-.95)/2 )
        lab var sig_b3c_lb "lb 95% CI, sig_b3c"
        lab var sig_b3c_ub "ub 95% CI, sig_b3c"
        ge sig_b3c_noncover = 0
        replace sig_b3c_noncover = 1 if (`sig_b3c' < sig_b3c_lb | `sig_b3c' > sig_b3c_ub)
        lab var sig_b3c_noncover "1: true sig_b3c not in 95%CI"

        ge sig_b4c_lb = sig_b4c + sig_b4c_se*invnormal( (1-.95)/2 )
        ge sig_b4c_ub = sig_b4c - sig_b4c_se*invnormal( (1-.95)/2 )
        lab var sig_b4c_lb "lb 95% CI, sig_b4c"
        lab var sig_b4c_ub "ub 95% CI, sig_b4c"
        ge sig_b4c_noncover = 0
        replace sig_b4c_noncover = 1 if (`sig_b4c' < sig_b4c_lb | `sig_b4c' > sig_b4c_ub)
        lab var sig_b4c_noncover "1: true sig_b4c not in 95%CI"


* other stats

        ge Nc = `Nc'
        lab var Nc "No. persons per country"

        ge C = `C'
        lab var C "Number of countries"

        ge icc = `icc'
        lab var icc "icc = var_u/(var_u+var_e)"

        drop lns*
        compress

* }


save, replace

de, fullnames


su b_cons b_age b_agesq  b_cohab  b_nownch b_isced3 b_isced4 b_isced56 b_chexp b_chexpXcohab
b_chexpXnownch

su se_cons se_age se_agesq  se_cohab  se_nownch se_isced3 se_isced4 se_isced56 se_chexp
se_chexpXcohab  se_chexpXnownch

su sig_u sig_b3c sig_b4c

su *abias

su *rbias

su *noncover

su Nc C icc converged
```

```
preserve
keep if converged == 1

su b_cons b_age b_agesq  b_cohab  b_nownch b_isced3 b_isced4 b_isced56 b_chexp b_chexpXcohab
b_chexpXnownch

su se_cons se_age se_agesq  se_cohab  se_nownch se_isced3 se_isced4 se_isced56 se_chexp
se_chexpXcohab  se_chexpXnownch

su sig_u sig_b3c sig_b4c

su *abias

su *rbias

su *noncover

su Nc C icc

restore


*******************************
log close
```

[End of Supplementary Material]