Heisig, Jan Paul; Schaeffer, Merlin; Giesecke, Johannes

The Costs of Simplicity: Why Multilevel Models May Benefit from Accounting for Cross-Cluster Differences in the Effects of Controls

American Sociological Review

AMERICAN SOCIOLOGICAL REVIEW

OFFICIAL JOURNAL OF THE AMERICAN SOCIOLOGICAL ASSOCIATION

ONLINE SUPPLEMENT
to article in
AMERICAN SOCIOLOGICAL REVIEW, 2017, VOL. 82

The Costs of Simplicity: Why Multilevel Models May Benefit from Accounting for Cross-Cluster Differences in the Effects of Controls

Jan Paul Heisig
WZB Berlin Social Science Center

Merlin Schaeffer
University of Cologne

Johannes Giesecke
Humboldt University Berlin

CONTENT

PART A: DATA AND MEASURES USED IN THE ILLUSTRATIVE ANALYSES

Our illustrative analyses examine how five individual-level outcome variables relate to the Human Development Index (HDI; United Nations Development Programme 2015) as a broad indicator of a country's modernization. We chose five variables that are representative of the diverse phenomena studied in applied work: generalized trust, xenophobia, occupational status, homophobia, and fear of crime. In addition to the direct effect of the HDI, we also explore if educational differences in these outcomes vary with the level of human development, that is, we estimate cross-level interactions between high education and the HDI.

The illustrative analyses are based on the European Social Survey (ESS Round 6 2012), one of the most widely used datasets in country-comparative multilevel analyses (for detailed documentation, see ESS Round 6 2016). We include all available countries except Kosovo where we detected problems with one individual-level variable (marital status). Specifically, we use the following 28 countries: Albania, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Hungary, Iceland, Ireland, Israel, Italy, Lithuania, Netherlands, Norway, Poland, Portugal, Russian Federation, Slovakia, Slovenia, Spain, Sweden, Switzerland, Ukraine, and the United Kingdom. This sample encompasses a fairly heterogeneous set of countries, with the HDI ranging from .740 in Ukraine to .955 in Norway (mean = .865; standard deviation = .055). National sample sizes vary between 752 respondents in Iceland and 2,958 respondents in Germany.

For simplicity and in keeping with the Monte Carlo simulations, we treat all outcomes as continuous. It might be more appropriate to treat fear of crime and homophobia as ordered, but we have no substantive interest in the results. Our goal is to explore whether models that allow for cross-country heterogeneity in the coefficients of lower-level control variables provide more precise estimates of context effects, and for this purpose linear models are fully sufficient. The measures of generalized trust and xenophobia are based on several survey items. To combine them into a single scale, we conducted a principal component factor analysis using the full country sample and predicted the factor scores. Table A.1 provides details on the variables and underlying survey items, including

their original range. We *z*-standardized all outcome variables to have a mean of 0 and a standard deviation of 1.

Table A.1. Description of Outcome Variables

Outcome	**Operationalization**	**Factor Loading**	**Range**
Generalized Trust	Index of three items		
	"Generally speaking, would you say that most people can be trusted, or that you can't be too careful in dealing with people?"	.846	0–10
	"Do you think that most people would try to take advantage of you if they got the chance, or would they try to be fair?"	.846	0–10
	"Would you say that most of the time people try to be helpful or that they are mostly looking out for themselves?"	.811	0–10
Xenophobia	Index of six items		
	"To what extent do you think [country] should allow people of the same race or ethnic group as most [country]'s people to come and live here?"	.705	1–4
	"How about people of a different race or ethnic group from most [country] people?"	.835	1–4
	"How about people from the poorer countries outside Europe?	.774	1–4
	"Would you say it is generally bad or good for [country]'s economy that people come to live here from other countries?"	–.736	0–10
	"Would you say that [country]'s cultural life is generally undermined or enriched by people coming to live here from other countries?"	–.732	0–10
	"Is [country] made a worse or a better place to live by people coming to live here from other countries?"	–.746	0–10
ISEI	Recoding of ISCO-08 based on: www.harryganzeboom.nl/isco08/index.htm. Among the unemployed, the ISCO-08 occupation refers to the last job held by the respondent.		11.01–88.96
Homophobia	"Gay men and lesbians should be free to live their own life as they wish."		1–5
Fear of crime	"How safe do you – or would you – feel walking alone in this area after dark?"		1–4

We regress these outcome variables on the HDI, several lower-level predictors, and—in some specifications—a cross-level interaction between the HDI and the high education indicator. The HDI values are part of the ESS distribution. We standardized them to have a mean of 0 and standard deviation of 1 at the country (rather than the individual) level. The individual-level variables are gender, age, marital status, being unemployed, and level of education. Age is the only continuous predictor and we standardize it to have a mean of 0 and a standard deviation of 1. The remaining predictors are categorical and we include them using weighted effect coding. Weighted effect coding is similar to grand mean centering of continuous variables (Grotenhuis et al. 2016). It ensures that the intercept corresponds to the predicted outcome for the average individual. This eases interpretation and safeguards against problems that can arise in the estimation of mixed-effects models when the intercept corresponds to a highly idiosyncratic value near or even beyond the boundaries of the observed covariate distribution (Enders and Tofighi 2007; Raudenbush and Bryk 2002). It further limits problems of scale dependence when correlations between random effects are suppressed during the process of model optimization (see Part G). Education is the only categorical predictor with more than two levels (low, intermediate, and high education based on a respondent's highest educational degree). We omit the low education category and include indicators for intermediate and high education. The specifications with cross-level interaction terms include an interaction between the HDI and the high education indicator. Table A.2 provides further details on the independent variables.

Table A.2. Description of Independent Variables

Variable	Operationalization	Mean/%	SD
Human Development Index (HDI)	HDI values as provided with the standard ESS distribution; *z*-standardized on the country level	0	1
Gender	Weighted effect coded variable that indicates women as 1 (value for men: –1.094)	0	
Age	*z*-standardized age in years	0	1
Marital Status	Weighted effect coded variable that indicates legally married respondents as 1(all others: –1.189)	0	
Education	Weighted effect coded variables based on the European Survey version of the International Standard Classification of Education (ES-ISCED)		
	Omitted category: low education (ES-ISCED values 0 to 2)		
	Intermediate Education: ES-ISCED values 3 to 5; indicates intermediate educated as 1, high educated as 0, and low educated as –2.084	0	
	High Education: ES-ISCED values 6 and 7; indicates high educated as 1, intermediate educated as 0, and low educated as –.976	0	
Unemployment	Weighted effect coded variable that indicates unemployed respondents as 1 (all others: –.061); a respondent is classified as unemployed if she did not work *and* actively looked for a job during the seven days before the interview	0	

PART B: DERIVATIONS

We first derive Equation 8, which gives the standard result for omitted variable bias in a two-variable setting. Recall that the true model is $y_{ig} = \beta_{1g} x_{1ig} + \beta_{2g} x_{2ig} + \varepsilon_{ig}$. If we omit x_{2ig}, we instead estimate the following regression for cluster g: $y_{ig} = \tilde{\beta}_{1g} x_{1ig} + \tilde{\varepsilon}_{ig}$. Then:

$$\begin{aligned} \mathrm{E}(\tilde{\beta}_{1g}) &= \frac{\sigma(x_{1ig}, y_{ig})}{\sigma^2(x_{1ig})} \\ &= \frac{\sigma(x_{1ig}, \beta_{1g} x_{1ig} + \beta_{2g} x_{2ig} + \varepsilon_{ig})}{\sigma^2(x_{1ig})} \\ &= \frac{\beta_{1g}\sigma^2(x_{1ig}) + \beta_{2g}\sigma(x_{1ig}, x_{2ig}) + \sigma(x_{1ig}, \varepsilon_{ig})}{\sigma^2(x_{1ig})} \\ &= \beta_{1g} + \beta_{2g} \frac{\sigma(x_{1ig}, x_{2ig})}{\sigma^2(x_{1ig})}. \end{aligned}$$

The last equation uses the fact that $\sigma(x_{1ig}, \varepsilon_{ig}) = 0$, which follows from the standard (exogeneity) assumption $\mathrm{E}(\varepsilon_{ig} \mid X_{ig}) = 0$.

To understand the problems that arise from incorrectly assuming invariant control slopes (Equation 9), note that assuming the slope of x_{2ig} to be invariant is similar to fitting cluster-specific regressions with β_{2g} constrained to equal $\beta_{2\bullet}$, the (weighted) average effect of x_{2ig}. With such a constraint one effectively estimates the effect of x_{1ig} on

$y^*_{ig} = \beta_{1g} x_{1ig} + \beta_{2g} x_{2ig} - \beta_{2\bullet} x_{2ig} + \varepsilon_{ig} = \beta_{1g} x_{1ig} + (\beta_{2g} x_{2ig} - \beta_{2\bullet} x_{2ig}) x_{2ig} + \varepsilon_{ig}$ rather than on $y_{ig} = \beta_{1g} x_{1ig} + \beta_{2g} x_{2ig} + \varepsilon_{ig}$. In this situation, the expectation of the coefficient on x_{1ig} therefore is:

$$
\begin{aligned}
\mathrm{E}(\beta_{1g}^{*}) &= \frac{\sigma(x_{1ig}, y_{ig}^{*})}{\sigma^{2}(x_{1ig})} \\
&= \frac{\sigma(x_{1ig}, \beta_{1g} x_{1ig} + (\beta_{2g} - \beta_{2\bullet}) x_{2ig} + \varepsilon_{ig})}{\sigma^{2}(x_{1ig})} \\
&= \frac{\beta_{1g}\sigma^{2}(x_{1ig}) + (\beta_{2g} - \beta_{2\bullet})\sigma(x_{1ig}, x_{2ig}) + \sigma(x_{1ig}, \varepsilon_{ig})}{\sigma^{2}(x_{1ig})} \\
&= \beta_{1g} + (\beta_{2g} - \beta_{2\bullet}) \frac{\sigma(x_{1ig}, x_{2ig})}{\sigma^{2}(x_{1ig})}.
\end{aligned}
$$

PART C: FULL DGPS OF THE MONTE CARLO SIMULATIONS

The full DGP-DCE has the form:

$$\begin{aligned} y_{ig} &= (\gamma_{00} + \gamma_{01} z_{1g} + \nu_{0g}) + \beta_{10} x_{1ig} \\ &\quad + (\beta_{20} + \nu_{2g}) x_{2ig} + (\beta_{30} + \nu_{3g}) x_{3ig} + (\beta_{40} + \nu_{4g}) x_{4ig} \\ &\quad + (\beta_{50} + \nu_{5g}) x_{5ig} + (\beta_{60} + \nu_{6g}) x_{6ig} + \varepsilon_{ig}. \end{aligned}$$

The full DGP-CLI has the form:

$$\begin{aligned} y_{ig} &= (\gamma_{00} + \gamma_{01} z_{1g} + \nu_{0g}) + (\gamma_{10} + \gamma_{11} z_{1g} + \nu_{1g}) x_{1ig} \\ &\quad + (\beta_{20} + \nu_{2g}) x_{2ig} + (\beta_{30} + \nu_{3g}) x_{3ig} + (\beta_{40} + \nu_{4g}) x_{4ig} \\ &\quad + (\beta_{50} + \nu_{5g}) x_{5ig} + (\beta_{60} + \nu_{6g}) x_{6ig} + \varepsilon_{ig}. \end{aligned}$$

As noted in the main article, we manipulate the extent of heterogeneity in the effects of controls by setting the standard deviation of the random effects on x_{2ig} to x_{6ig} to values greater than zero. For example, in the experimental condition with varying coefficients on three control variables, we set $\sigma(\nu_{2g})$, $\sigma(\nu_{3g})$, and $\sigma(\nu_{4g})$ to either .2 or 1. $\sigma(\nu_{0g})$ and $\sigma(\nu_{1g})$, the standard deviations of the random effects related to the contextual predictor of interest, are always .6.

PART D: IMPLEMENTATION OF MONTE CARLO SIMULATIONS AND SOFTWARE

We conducted all simulations in R (R Core Team 2015). To minimize Monte Carlo error, we obtained 10,000 datasets for each experimental condition and applied all estimators to each of them (i.e., we applied OLS-Cluster, ME-Invariant, and ME-Correct in the direct-context-effect conditions and additionally applied two-step-FGLS in the cross-level interaction conditions). We estimated the mixed-effects models by restricted maximum likelihood using the `lmer` function from the `lme4` package, using the default optimizer `bobyqa` (Bates, Maechler, et al. 2015). To obtain cluster-robust OLS estimates, we used R's built-in `lm` function and the `clx` function by Arai (2015). We also used the `lm` function to run the cluster-specific regressions required for two-step estimation. For the FGLS implementation described in Lewis and Linzer (2005), we borrowed R code posted on the first author's webpage at http://www.sscnet.ucla.edu/polisci/faculty/lewis/software/edvreg.R (last accessed September 1, 2015). We provide the full R Code for the simulations with the online supplements.

Estimation of mixed-effects models can run into convergence problems, that is, the optimizer may fail to identify the maximum of the likelihood function. To alert users to potential convergence problems, the `lmer` function issues warnings when the gradient of the likelihood function is not sufficiently close to zero at the solution or when the Hessian is not positive definite. In concrete applications, one would take various steps in such a situation (e.g., try alternative optimizers, double-check the model and data), but in a simulation study this is obviously infeasible. In our simulations, convergence was hardly an issue for ME-Invariant, but estimation of ME-Correct quite often triggered convergence warnings. This is not surprising because ME-Correct estimates a substantial number of random-effect variances and covariances in experimental conditions where the coefficients of several lower-level variables vary across clusters. Fortunately, separate analysis of replications with and without convergence warnings does not suggest that our main conclusions are sensitive to the convergence status of ME-Correct. In Part E, Section E.5, we further discuss this issue and show simulation results disaggregated by whether convergence warnings occurred.

PART E: ADDITIONAL RESULTS

E.1 Low correlations among random effects

ME-Correct can exploit information about systematic correlations among the random slopes to arrive at somewhat better estimates of the (fixed-effect) parameter of interest. In the Monte Carlo simulations presented in the main article, cross-cluster differences in the slopes were created by drawing multivariate normal effects with a random correlation matrix with an average absolute correlation of .33 (see note 5 in the main article). Further simulations, which we present here, show that the RMSE no longer declines with the number of random slopes when we use a DGP with lower correlations among the random effects (average absolute correlation of .20). However, ME-Correct still yields more precise estimates of context effects than do ME-Invariant and OLS-Cluster in experimental conditions with substantial cross-cluster heterogeneity in the effects of lower-level controls.

Figure E.1.1. Precision of estimated direct context effect by cross-cluster compositional differences and variation in the coefficients of lower-level controls; 25 countries and lower correlations among random slopes than in the main analysis

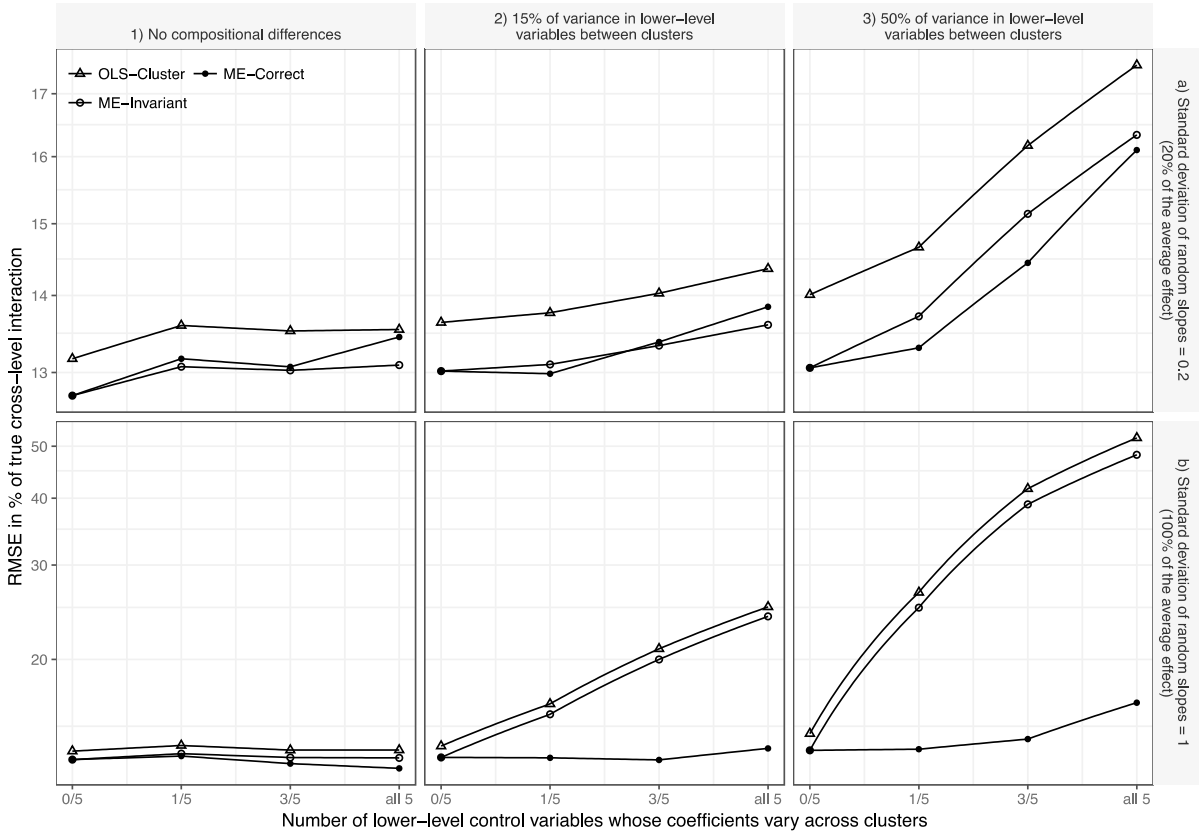
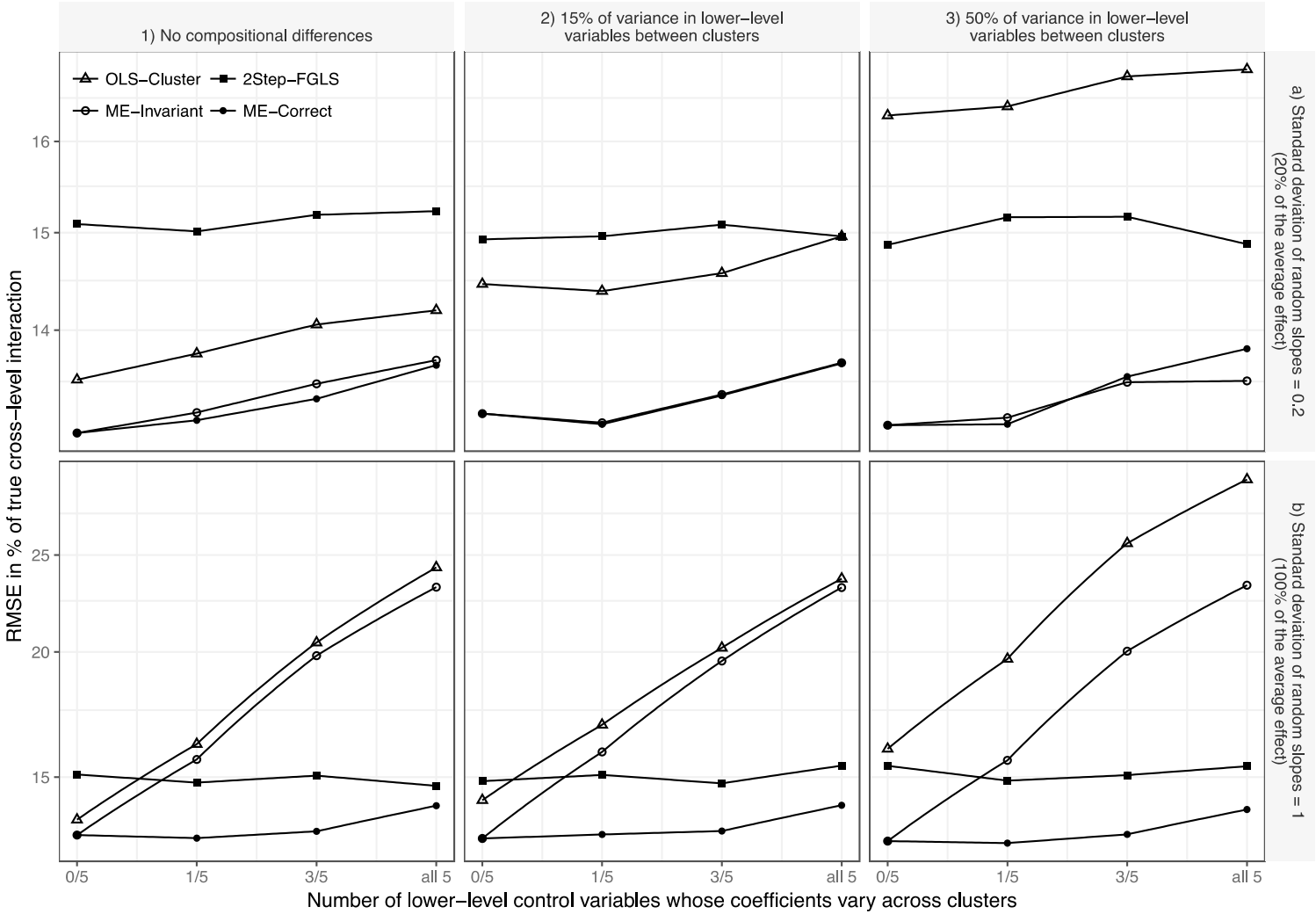


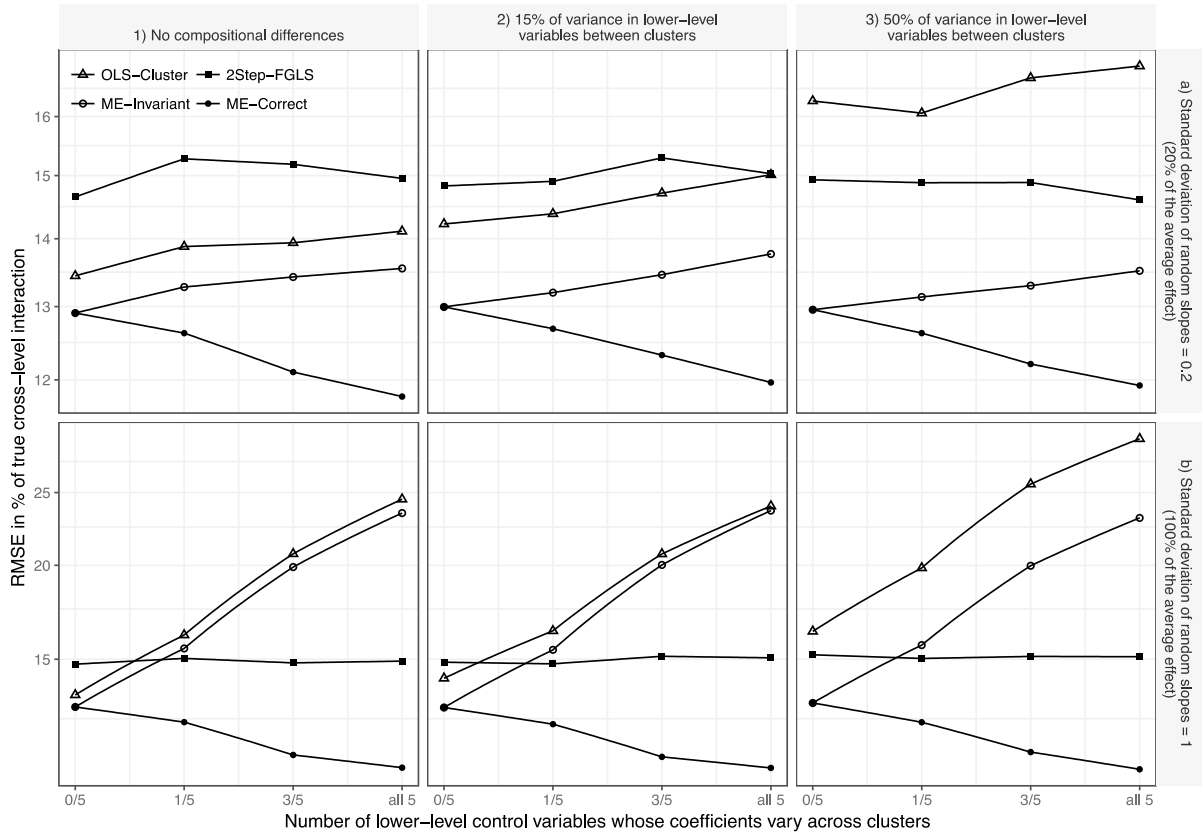
Figure E.1.2. Precision of estimated cross-level interaction by cross-cluster compositional differences and variation in the coefficients of lower-level controls; 25 countries and lower correlations among random slopes than in the main analysis



E.2 Precision of cross-level interaction estimates by extent of compositional differences

For the cross-level interaction case, we only present simulation results based on moderate compositional differences among clusters (15 percent of the variance in lower-level variables between clusters) in the main article. Figure E.2.1 shows differences by extent of compositional differences for the 25 countries scenario. Two findings are noteworthy. First, and in contrast to the direct-context-effect case, neglecting cross-cluster heterogeneity in the effects of lower-level variables leads to efficiency losses even when there are no compositional differences among clusters. Second, unlike in the direct-context-effects case, the efficiency disadvantage of ME-Invariant relative to ME-Correct (and two-step-FGLS) is not related to the extent of compositional differences. This stands in sharp contrast to cluster-robust OLS, whose performance does suffer from increased compositional differences.

Figure E.2.1. Precision of estimated cross-level interaction by cross-cluster compositional differences and variation in the coefficients of lower-level controls; 25 countries



E.3 Coverage rates by extent of compositional differences

Figures E.3.1 and E.3.2 show how the extent of compositional differences affects statistical inference (actual coverage rates of nominal 95 percent confidence intervals) for the direct-context-effect and cross-level-interaction cases, respectively. Results are for the 25 countries scenario. The most important finding is that greater compositional differences exacerbate the undercoverage of analytic confidence intervals for ME-Correct and OLS-Cluster. The accuracy of ME-Invariant is not affected by the degree of compositional differences across clusters.

Figure E.3.1. Statistical inference for direct context effects by extent of compositional differences and cross-cluster variation in the coefficients of lower-level controls; 25 countries

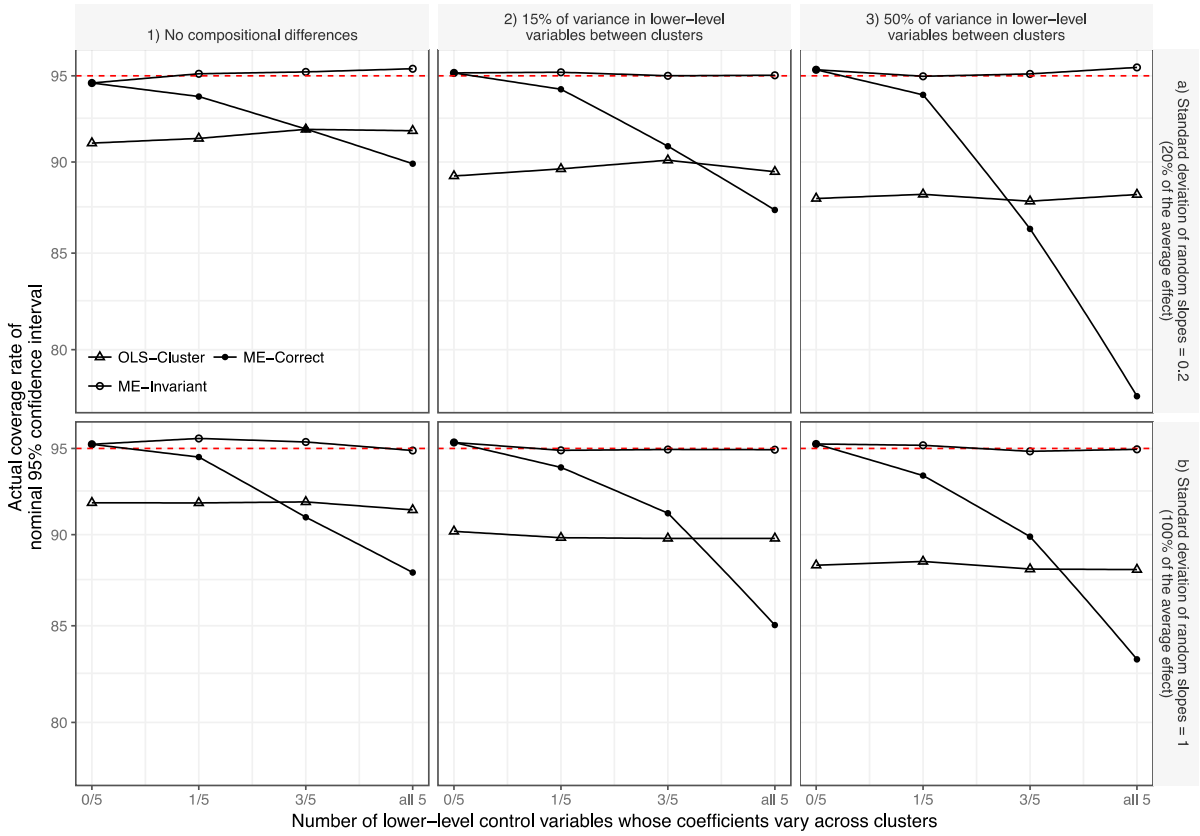
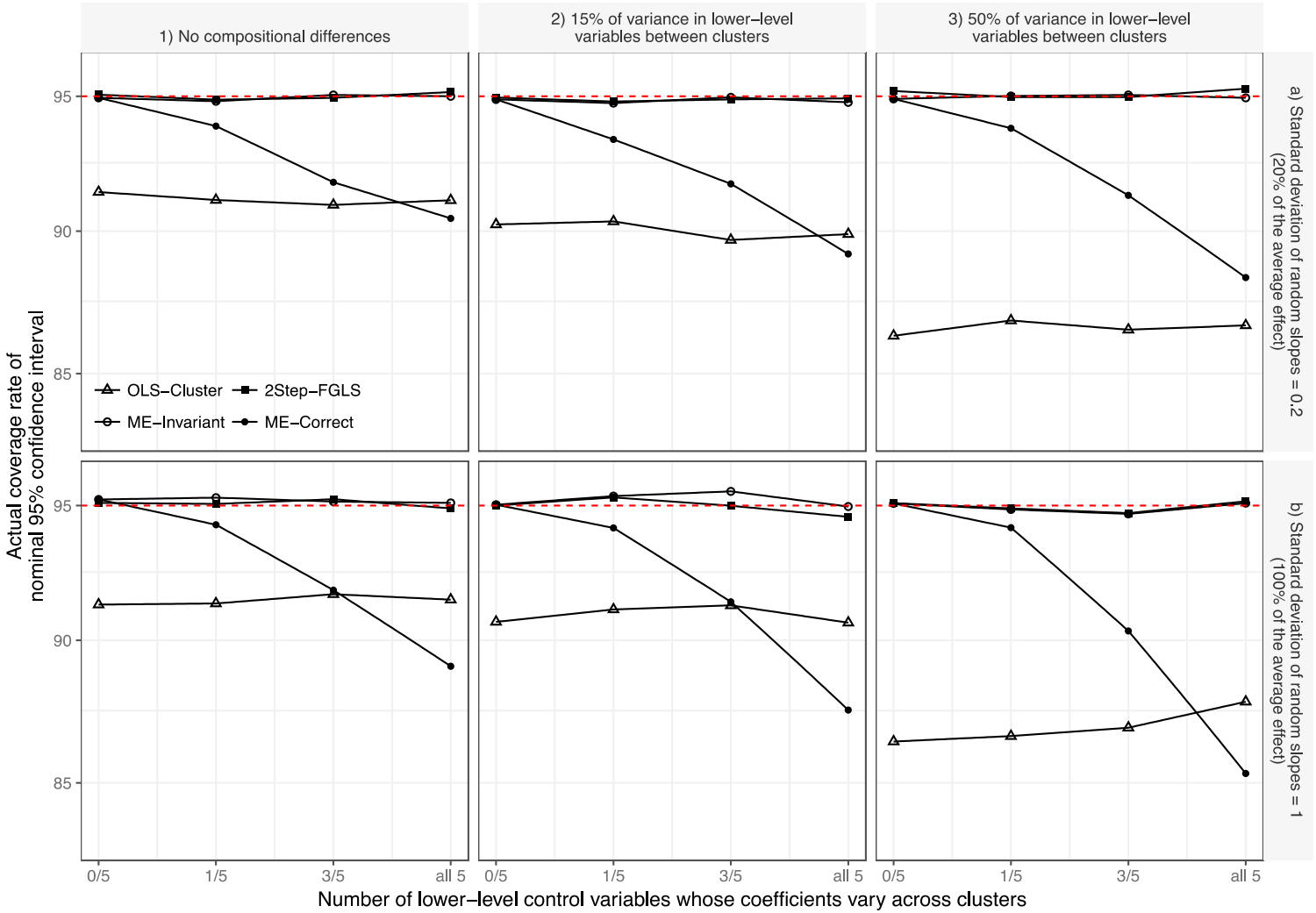


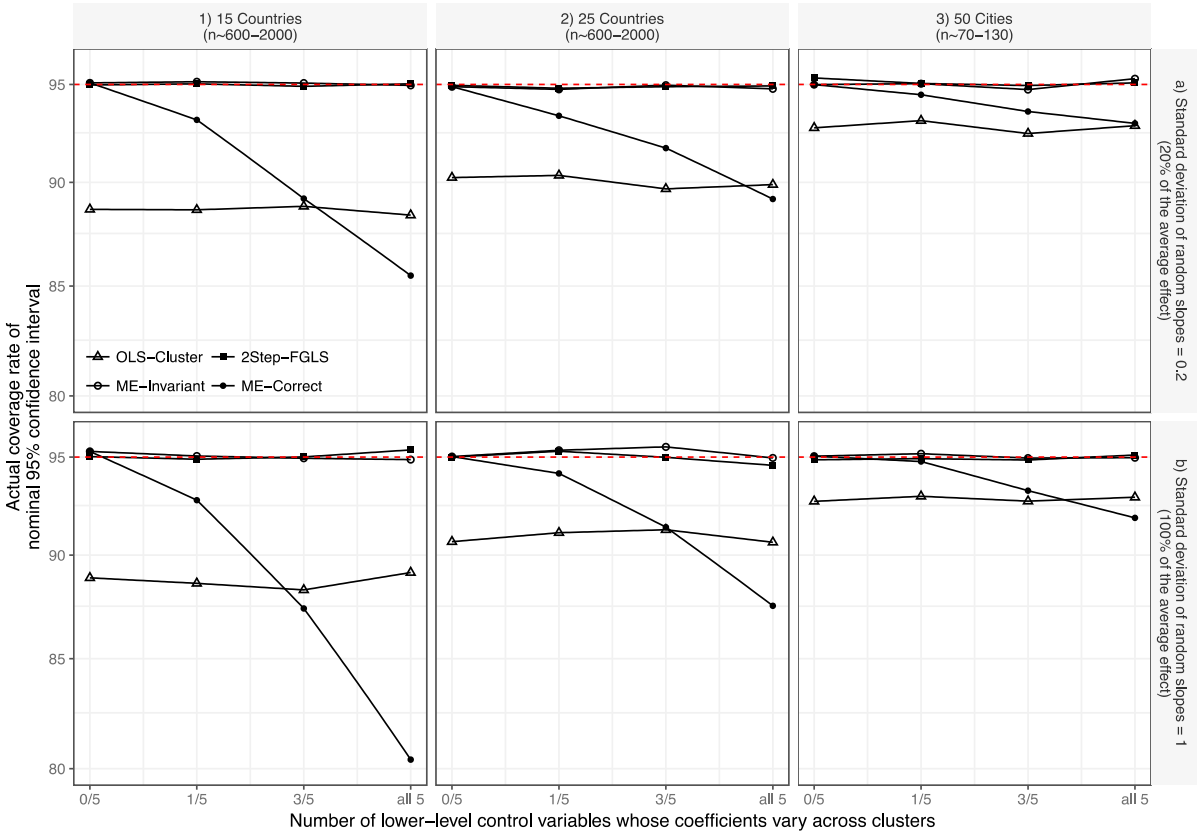
Figure E.3.2. Statistical inference of cross-level interactions by extent of compositional differences and cross-cluster variation in the coefficients of lower-level controls; 25 countries



E.4 Coverage rates by number and size of clusters, cross-level interaction case

Figure E.4.1 shows how actual coverage rates of confidence intervals for the cross-level interaction case differ by number of clusters and cluster size (and the extent of cross-cluster differences in the effects of controls). The most important result is that two-step-FGLS provides accurate coverage in all experimental conditions. Results for the other estimators resemble those for the direct-context-effect case (see Figure 3 in the main article).

Figure E.4.1. Statistical inference for cross-level interaction effects by number and size of clusters and cross-cluster variation in the coefficients of lower-level controls; intermediate compositional differences (15 percent of variance in lower-level variables between clusters)



E.5 Convergence

Estimation of mixed-effects models can run into convergence problems, particularly when the random-effects specification is complex, as it is for ME-Correct in the experimental conditions where the slopes of several controls vary across clusters. Here we discuss how common convergence problems are in the Monte Carlo simulations, and whether they influenced our results. We also provide some recommendations for dealing with convergence problems in actual applications. For simplicity, we label a case as non-convergent whenever the `lmer` function issued a warning (usually

because the scaled gradient of the deviance function exceeded the predefined tolerance threshold; for details see Bates, Maechler, et al. 2015). Many of these cases may be false positives in the sense that the optimizer actually reached the optimal solution; but in a Monte Carlo study this is difficult to verify for individual replications.

With regard to direct context effects, ME-Correct converged successfully (i.e., without warning) in 86 percent of the cases with three random slopes and in 69 percent of cases with five random slopes on controls. These numbers are somewhat smaller—78 and 51 percent, respectively—for the cross-level interaction case (where the model includes an additional random slope on the lower-level variable that is part of the cross-level interaction). These numbers are averaged over all experimental conditions other than the number of controls with varying coefficients.

In Figure E.5.1 we investigate if there are systematic differences in terms of RMSE and coverage between mixed-effects models that did converge (left column) and models that did not (right column). We present results for the case of 50 cities, intermediate compositional differences (15 percent of variance in control variables between clusters), and varying coefficients with a standard deviation of 1. Figure E.5.1 thus disaggregates the results for ME-Invariant and ME-Correct in panel 3b in Figures 1, 2, 3, and 4 by whether the model converged or not. Figures E.5.2 and E.5.3 depict results for 15 and 25 countries (results for the other experimental conditions are available upon request). The size of the markers represents the percentage of replications for a given experimental condition that did or did not converge. It is evident that ME-Invariant rarely failed to converge, in contrast to ME-Correct.

The key conclusion to draw from Figures E.5.1 to E.5.3 is that our main findings hold irrespective of whether ME-Correct converged or not. In particular, we find that ME-Correct models that did not converge successfully have an RMSE similar to that of ME-Correct models that did converge, and generally lower than that of ME-Invariance. With respect to coverage, Figures E.5.1 to E.5.3 show that undercoverage in the presence of many random slopes is not restricted to degenerate cases where ME-Correct did not converge. Ensuring that the model converged properly thus is no solution to the inferential problems noted earlier. That said, there is some indication in the direct-context-effects case that undercoverage is worse for the non-converged models (panels 1c and 2c in Figures E.5.1 to E.5.3). Moreover, the pattern that additional random slopes exacerbate undercoverage does not apply

to the non-converged models in panel 2c of Figure E.5.1. This explains why the aggregate results in panel 3b of Figure 4 in the main article deviate from this pattern.

Figure E.5.1. Precision and statistical inference by convergence of mixed-effects models (50 cities; intermediate compositional differences; coefficients of control variables vary across clusters with a standard deviation of 1)

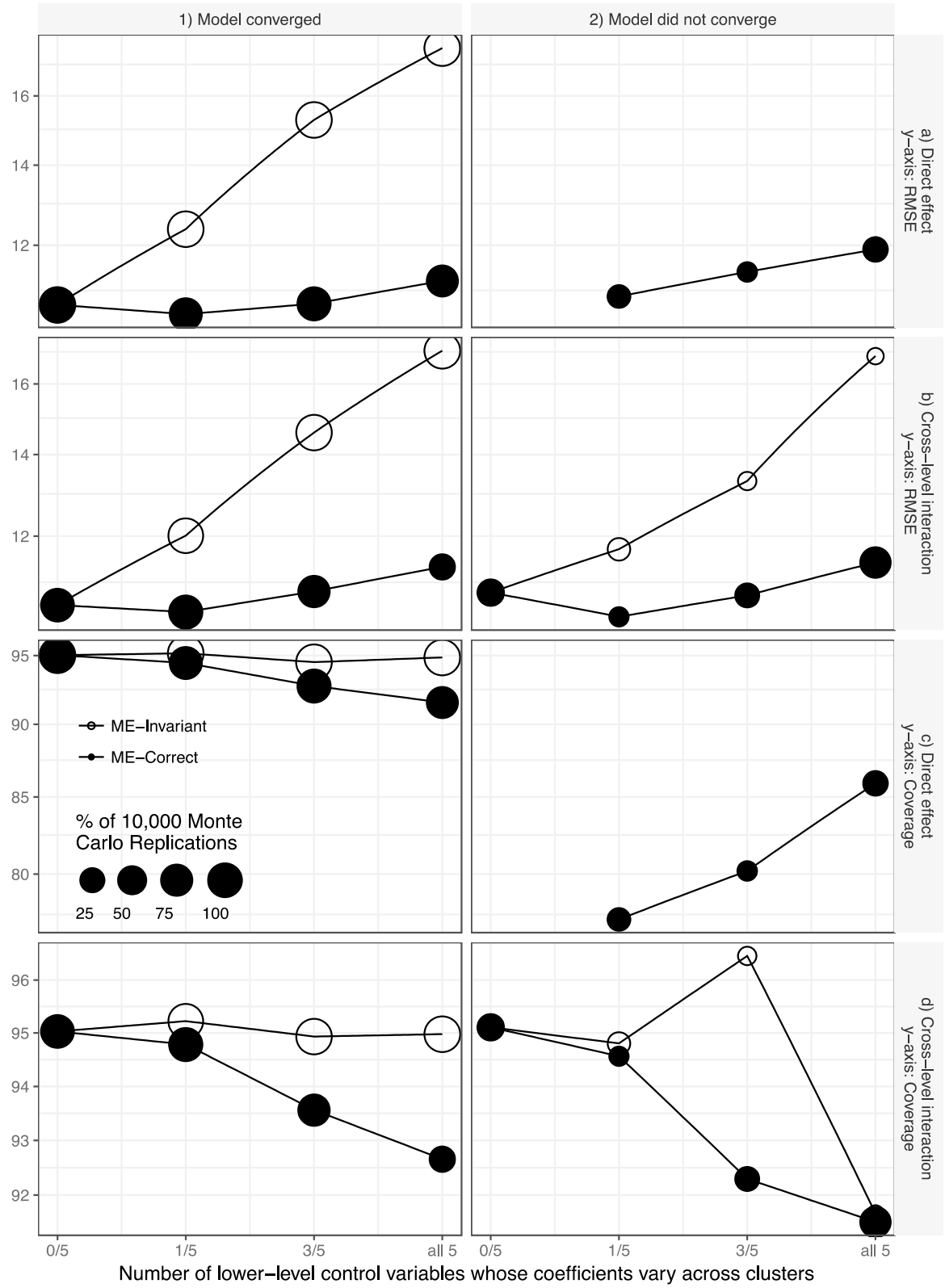


Figure E.5.2. Precision and statistical inference by convergence of mixed-effects models (15 countries; intermediate compositional differences; coefficients of control variables vary across clusters with a standard deviation of 1)

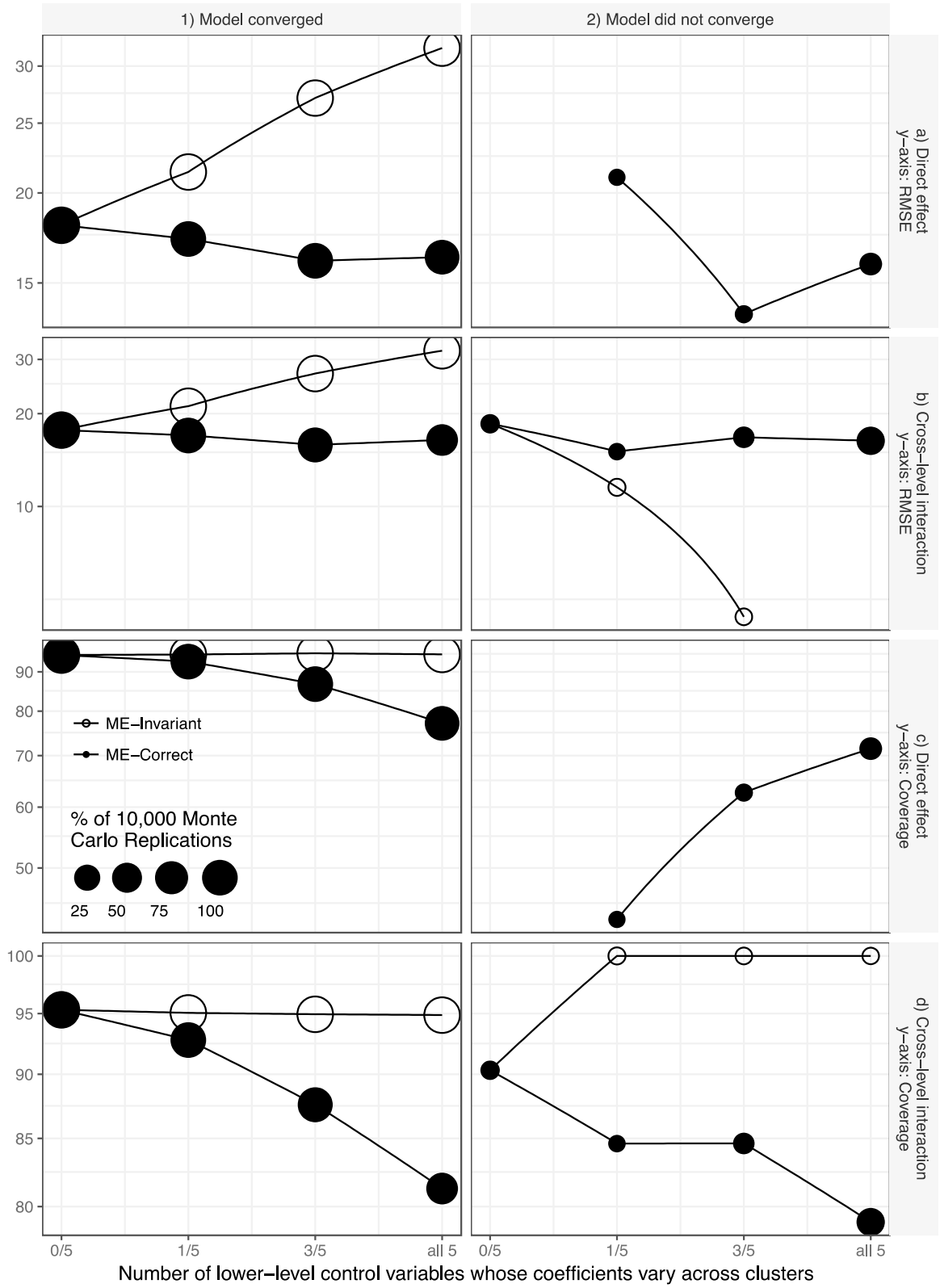
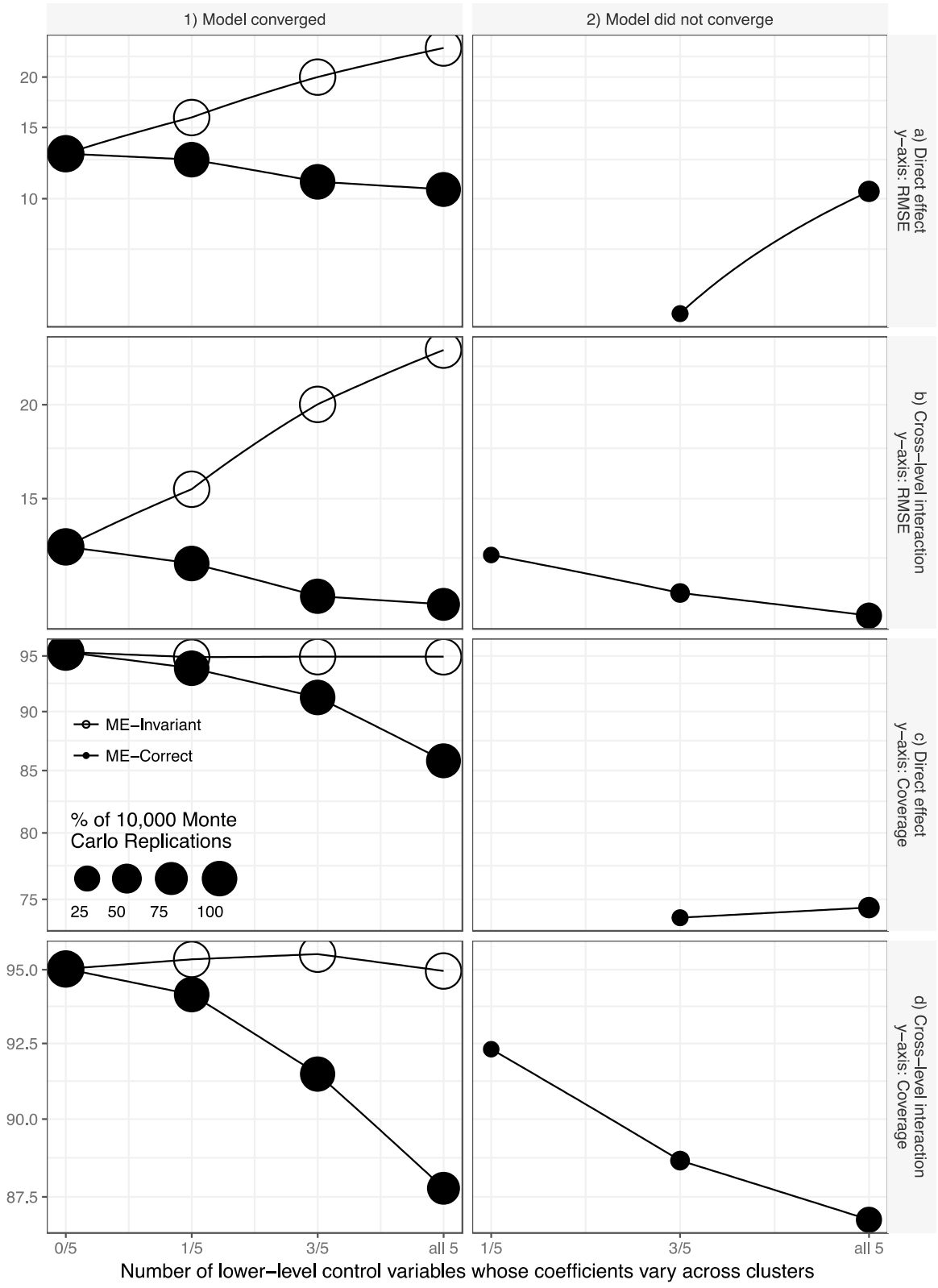


Figure E.5.3. Precision and statistical inference by convergence of mixed-effects models (15 countries; intermediate compositional differences; coefficients of control variables vary across clusters with a standard deviation of 1)



Our main conclusions do not depend on the convergence of mixed-effects models, but we do not mean to imply that non-convergence can be taken lightly in practice. Researchers who encounter convergence problems should exercise great care. Among other things, one may want to increase the number of iterations of the optimization algorithm and try alternative optimizers. In many cases, however, the primary reason for convergence issues will be that the model is too complex given the data at hand. To resolve convergence problems, it will then usually be sufficient to simplify the random-effects structure using the optimization strategy outlined in the main article (see the Flexible Multilevel Modeling in Practice section in the main article) and in Part G of this supplement.[1]

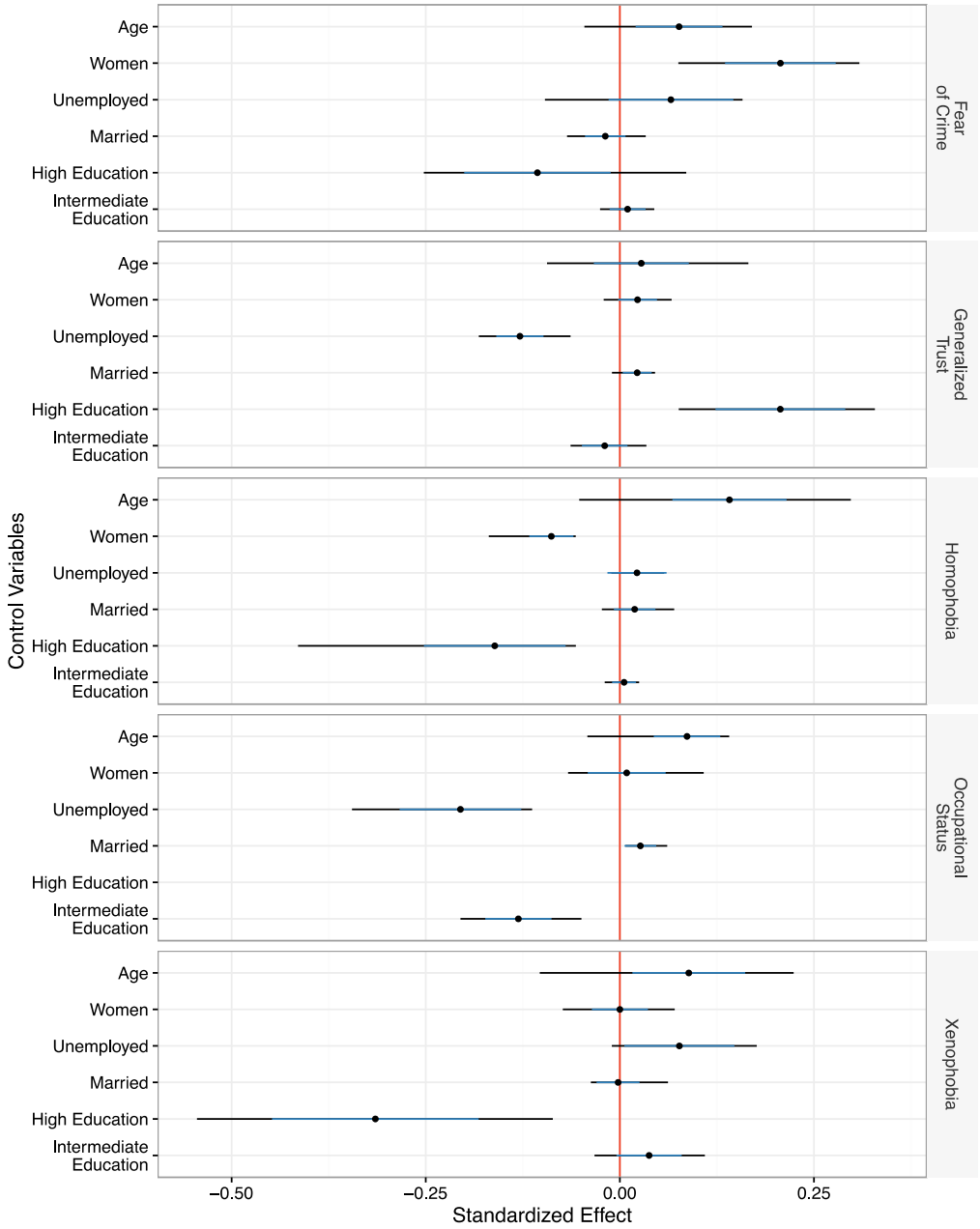
E.6 Varying coefficients of lower-level variables are a real concern

In the main article (see Figure 6), we show—for five different outcome variables—that the coefficients of standard individual-level control variables vary substantially across our sample of 28 countries from the European Social Survey, suggesting that such variation should be an important concern in many (cross-national) multilevel studies. As in our discussion of the Monte Carlo simulations, we expressed the extent of coefficient variation in percent of the respective average coefficient. One might be concerned that the finding of marked variability is driven by small average coefficients in the denominator of that statistic. To address this issue, Figure E.6.1 depicts the average coefficient as well as its standard deviation and range (where the standard deviation and range are based on best linear unbiased predictions, or BLUPs) across the 28 countries. We show results for 29 of the 30 combinations (five dependent and six independent variables). The figure omits the association between high education and occupational status, because that coefficient (beta = 1.023, SD = .089; Min. = .878; Max. = 1.197) is so large that it would distort the scale. In some cases, the average coefficient is indeed quite small. For example, the effect of having intermediate education is close to zero for all dependent variables except occupational status. Recall, however, that all categorical predictors, including intermediate education, are weighted effect coded so that they

[1] In particular, a frequent reason for non-convergence of models with many random slopes is that there is little cross-cluster variability in some of the coefficients that are specified as varying. This can create problems for optimization because the possible parameter space for random-effects variances has a lower bound at zero. In our simulations, this shows in the fact that convergence problems occur more frequently when we set the standard deviation of the coefficients of lower-level controls to .2 rather than 1 (detailed results available upon request). The optimization strategy will be helpful in identifying and removing such random effects.

estimate the difference to the average European respondent (average coefficients on dummy-coded measures would be substantially larger). More importantly, the overall impression emerging from Figure E.6.1 confirms the result from the main article: the coefficients of standard individual-level controls mostly differ markedly across countries.

Figure E.6.1. Fixed effects of lower-level variables and their variation across 28 ESS countries



Note: Thick lines depict +/– one standard deviation and thin lines the range of country-specific coefficients (country-specific coefficients estimated using best linear unbiased predictions from a mixed-effects model with random slopes on all predictors). The figure omits the association between high education and occupational status (see text).

E.7 Absolute width of confidence intervals in the illustrative analyses

Tables 3 and 4 in the main article compare the precision of estimated context effects between the invariant mixed-effects specification and three more flexible alternatives: two-step estimation, the maximally flexible mixed-effects model, and an optimized mixed-effects model. For each of the five outcome variables, the tables report relative differences in the width of 95 percent confidence intervals for the coefficients of the HDI and its cross-level interaction with the high education indicator. In Table E.7.1, we report the absolute width of the (analytic and bootstrapped) confidence intervals that underlie the relative differences in Tables 3 and 4 in the main article.

As noted in the main text, the bootstrapped intervals for the maximally flexible mixed-effects model are consistently (and mostly also substantially) larger than their analytic counterparts. The sole exception is the main effect of the HDI in the model for generalized trust. This confirms the result of the Monte Carlo simulations that analytic inference for complex mixed-effects specifications tends to be anticonservative. For all three other estimators (two-step and the invariant and optimized mixed-effects models), bootstrapped and analytic confidence intervals tend to differ, but the direction is inconsistent, with the former sometimes being larger than the other and sometimes vice versa. Moreover, the direction of the difference tends to be in the same direction for all three estimators (i.e., if the bootstrapped interval is smaller than the analytic for one of the estimators, this also tends to hold for the other two). This suggests that the difference reflects aspects of the data rather than the individual estimators (e.g., a violation of homoscedasticity assumptions). The fact that bootstrapped intervals are not systematically larger than analytic ones for the optimized model indicates that the optimization procedure quite effectively combats overparameterization, which is a likely explanation for the bad performance of analytic intervals in the maximally flexible case. Nevertheless, we recommend that researchers obtain bootstrapped confidence intervals for optimized specifications until this issue has been investigated more systematically. As discussed in the main article, additional Monte Carlo simulations show that analytic intervals for the optimized models have better coverage rates, but still tend to fall short of the nominal 95 percent level.

Table E.7.1. Absolute width of analytic and bootstrapped confidence intervals

		Invariant mixed-effects model		Maximally flexible mixed-effects model		Two-step model		Optimized mixed-effects model	
Outcome	**Context Effect**	**Bootstrap CI**	**Analytic CI**	**Bootstrap CI**	**Analytic CI**	**Bootstrap CI**	**Analytic CI**	**Bootstrap CI**	**Analytic CI**
Direct context effect (DCE)									
Generalized Trust	Direct HDI effect	.2111	.2433	.1967	.1613			.2063	.2128
Homophobia	Direct HDI effect	.2588	.2655	.2588	.2032			.2594	.2638
Xenophobia	Direct HDI effect	.2885	.2894	.3858	.2430			.2485	.2555
Fear of crime	Direct HDI effect	.1790	.1762	.2922	.1451			.1520	.1580
ISEI	Direct HDI effect	.1051	.0932	.1238	.0747			.0931	.0843
Cross-level interaction (CLI)									
Generalized trust	Interaction effect	.0645	.0638	.0615	.0590	.0692	.0742	.0620	.0650
Homophobia	Interaction effect	.0569	.0783	.0668	.0653	.0603	.0816	.0569	.0675
Xenophobia	Interaction effect	.0765	.0859	.0883	.0700	.0771	.0895	.0746	.0726
Fear of crime	Interaction effect	.0588	.0574	.0716	.0535	.0669	.0597	.0613	.0551
ISEI	Interaction effect	.0567	.0531	.0513	.0459	.0749	.0668	.0554	.0490
Generalized trust	Main HDI effect	.2166	.2470	.1674	.1833			.2177	.2474
Homophobia	Main HDI effect	.2593	.2674	.2499	.1997			.2555	.2502
Xenophobia	Main HDI effect	.2789	.2860	.3141	.2448			.2498	.2553
Fear of crime	Main HDI effect	.1705	.1734	.1952	.1512			.1509	.1578
ISEI	Main HDI effect	.1102	.0974	.0888	.0751			.0812	.0760

PART F: OBTAINING BOOTSTRAP CONFIDENCE INTERVALS IN R AND STATA

The Monte Carlo simulation results indicate that analytic inference (i.e., inference based on analytic standard error estimates) is anticonservative for complex mixed-effects specifications. Further analysis suggests that a non-parametric cluster bootstrap effectively addresses these limitations and provides accurate inference (see Figure 5 in the main article). Here, we describe the bootstrap procedure in greater detail and provide a detailed explanation of how to implement it in R and Stata (interested readers might also want to consult the replication files available with the online supplements). We recommend that researchers use bootstrap-based inference when estimating complex mixed-effects specifications. Although the illustrative analyses presented in the Flexible Multilevel Modeling in Practice section in the main article suggest that analytic inference for carefully optimized specifications may be relatively accurate, our Monte Carlo simulations suggest they may fall short of providing accurate standard errors too.

Efron and Tibshirani (1993) and Davison and Hinkley (1997) provide introductions to bootstrap methods. The basic idea is to generate a large number of bootstrap samples (or replicates) by drawing (with replacement) from the original sample. One then applies the estimator of interest to each of these bootstrap samples. Under certain conditions, the distribution of regression coefficients (or other quantities of interest) across the bootstrap replications then can be used to reliably approximate their sampling distributions, even when analytic solutions are unavailable or biased.

One can make a general distinction between parametric and non-parametric bootstrap methods. Each of these two broad classes comprises several variants, and semi-parametric approaches are also possible (Goldstein 2011). Broadly speaking, parametric bootstrap methods do not involve full resampling of observations. Instead, each bootstrap replication consists of estimating the model using the original data, but with a new pseudo outcome variable rather than the original observed outcome variable as the dependent variable. The pseudo outcome variable is created by (1) resampling errors based on the model fitted using the original data (including the original outcome variable) and (2) adding them to the predicted values from that model. In the case of mixed-effects estimation, creation of the pseudo outcome requires sampling of several error components (i.e., cluster-level random

intercept and slopes as well as the lower-level residual error). Non-parametric (or cases) bootstrapping resamples complete observations, resulting in bootstrap samples that typically contain some of the original observations multiple times, while other observations are not included at all. In the case of clustered/hierarchical data, a key question is whether to resample cases at all or only at a subset of the different levels (see below).

Bootstrapping is computationally intensive, especially if combined with a Monte Carlo analysis where a large number of bootstrap replicates have to be created for each Monte Carlo replicate (note that in the Monte Carlo setting, the term "original data" refers to an artificial dataset created by sampling from the DGP of interest). Because of this large computational burden, we did not evaluate the accuracy of bootstrap-based inference for all experimental conditions, but rather focused on one where analytic confidence intervals suffer from severe undercoverage: the case of 15 countries, moderate compositional differences (15 percent of variance between clusters), and three controls with varying slopes, each with a standard deviation of 1. In this condition, analytic 95 percent confidence intervals have an actual coverage rate of 86.56 percent (cf. panel 1b in Figure 3 in the main article). Moreover, we only investigated a non-parametric cases bootstrap, because an exploratory investigation based on fewer Monte Carlo replications suggested that several alternative parametric bootstrap methods improved coverage only to roughly 89 percent.

For 5,000 simulated Monte Carlo datasets, we thus explored the performance of a non-parametric bootstrap procedure. That is, we ran 2,000 bootstrap replications per Monte Carlo dataset, so we had to obtain ten million [= 5,000 × 2,000] estimates. To form the 95 percent bootstrap confidence interval for the *m*th Monte Carlo dataset, we used the "basic" method (Davison and Hinkley 1997: Chapter 5).

For the non-parametric bootstrap, we resampled clusters with replacement using the `lmeresampler` function for multilevel bootstrapping (Loy and Steele 2016). Following recommendations in the literature, we did not resample lower-level units within clusters (Goldstein 2011; Ren et al. 2010). Thus, in the 15 countries case, we would create a bootstrap sample by sampling 15 clusters from the original (simulated) dataset. As noted earlier, a non-parametric bootstrap sample typically includes some of the original clusters several times, whereas others are not

included at all. When estimating the model of interest based on the bootstrap sample, different draws of the same cluster are treated as independent clusters (i.e., technically they are assigned different cluster IDs). Accordingly, the size of the cluster-level sample remains the same and equals the sample size of the original data.[2]

Confidence intervals based on the non-parametric bootstrap show good performance with an actual coverage rate of 95.62 percent, suggesting it provides quite accurate and perhaps even slightly overconservative inference. We therefore recommend that researchers use the non-parametric bootstrap when fitting complex mixed-effects models with small cluster-level samples. We now show how to implement the method in R and Stata.

Implementation in R

The primary package for bootstrapping in R, `boot`, does not support resampling of clusters at this time. Fortunately, a package implementing the non-parametric bootstrap for `lme4`'s `lmer` function has recently become available. The package `lmeresampler` (Loy and Steele 2016) can be installed in the usual way from the Comprehensive R Archive Network (CRAN). Further information, including development versions, is available at https://github.com/aloy/lmeresampler.

The first step toward obtaining bootstrap confidence intervals with `lmeresampler` is to fit the model of interest using the original data. A typical call (fitting a maximally flexible model with dependent variable `y`, six lower-level predictors `x1` to `x6`, and one cluster-level predictor `z` and saving results in the object `orgdat_fit`) might look as follows:

```
orgdat_fit <- lmer(y ~ x1 + x2 + x3 + x4 + x5 + x6 + z + (1 + x1 +
  x2 + x3 + x4 + x5 + x6 | cl_id), data = orgdat, REML = TRUE)
```

The next step is to create the bootstrap replicates using the function `bootstrap.lmerMod` from the `lmeresampler` package. A typical call would look as follows:

[2] The lower-level sample size will typically differ from the original sample.

```
bstraps <- bootstrap.lmerMod(model = orgdat_fit, type = "case", fn
  = extractor, B = 2000, resample = c(TRUE, FALSE))
```

This call requests 2,000 bootstrap replications (`B = 2000`) based on the (original) data and model formula used in `orgdat_fit` (`model = orgdat_fit`). The argument `type = "case"` requests a non-parametric (aka "cases") bootstrap and the argument `resample = c(TRUE, FALSE)` specifies that resampling should occur at the upper but not lower level. Finally, the argument `fn = extractor` tells `bootstrap.lmerMod` to use the function `extractor` for obtaining quantities of interest from the `lmer` fits for the individual replicates. Typically, this function will extract fixed-effects coefficients and variance components and potentially a few other quantities of interest.

The following lines define a rudimentary function that recovers the fixed-effects coefficients and standard deviations of the random effects:

```
extractor <- function(.) {
 extracts <- c(fixef(.), diag(sqrt(VarCorr(.)[[1]])),
  attr(VarCorr(.), "sc"))
 names(extracts) <- c(paste("fecoef.", names(fixef(.)), sep = ""),
 paste("resd.",
   colnames(VarCorr(.)[[1]]), sep = ""), "resd.Residual")
 extracts
}
```

A convenient feature of `lmeresampler` is that it returns an object of class "boot" that can be used with `boot` (Canty and Ripley 2016), the standard package for bootstrap-based inference in R. In particular, the object can be fed into the `boot.ci` function for obtaining confidence intervals. To obtain two-sided 95 percent confidence intervals using the basic method, one would use the following command:

```
boot.ci(bstraps, index = i, conf = .95, type = "basic")
```

where `index = i` requests that the interval be constructed for the *i*th quantity extracted by the `extractor` function.

The bootstrapping process is computationally intensive and it usually pays off to parallelize it. An easy way to do this is to use the package `doParallel`. As the resampling has a random component, it is important to ensure that the parallel worker processes use different random number seeds (otherwise one and the same bootstrap sample will be created multiple times). The `doRNG` package makes this easy and also ensures that the random number streams are truly independent. To ensure reproducibility, one needs to initialize the random number seed using the `set.seed` function before the `foreach` loop. A parallelized version of the bootstrapping command above might then look roughly as follows:

```
  set.seed(455363)

  bstraps <- foreach(i=1:Ncores, .packages = package.list, .export =
export.objects) %dorng% {

    bootstrap.lmerMod(model = orgdat_fit, type = "case", fn =
extractor, B
    = RepsPerCore, resample = c(TRUE, FALSE))
  }
```

The number of cores to use should have been previously declared using `registerDoParallel(cores = Ncores)` and `RepsPerCore` specifies the number of replications per core (which should be equal to the total number of replications divided by the number of cores). The objects `package.list` and `export.objects` are lists of packages/objects that are used inside the foreach loop (e.g., `lmeresampler` and `orgdat_fit`).

Implementation in Stata

In Stata, one can use the `bootstrap` command to perform the non-parametric cluster bootstrap. `bootstrap` is a prefix command followed by a colon and the command to be applied to the bootstrap samples. In the case of interest, this will be a command for mixed-effects estimation such as `mixed`. The syntax is relatively straightforward. The number of bootstrap replicates is set using the `reps` option. Two crucial options are `cluster` and `idcluster`. The `cluster` option specifies the variable that identifies the clusters in the original data. The `idcluster` specifies a new cluster

variable that will be generated in the bootstrap samples. The mixed-effects command following the bootstrap will look very much like the one that one would apply to the original data. The crucial difference is that one needs to use the new rather than the original cluster variable (if the original cluster variable were used, repeated draws of the same cluster would effectively be merged into one large cluster). In addition, it is generally a good idea to use the `nostderr` option, which suppresses the estimation of standard errors for the variance components and speeds up estimation. A typical call (fitting a maximally flexible model with dependent variable `y`, six lower-level predictors `x1` to `x6`, and one cluster-level predictor `z`) might look as follows:

```
set seed 455363
bootstrap, reps(2000) cluster(cl_id) idcluster(bs_cl_id):  ///
   mixed y x1 x2 x3 x4 x5 x6 z || bs_cl_id: x1 x2 x3 x4 x5 x6 ///
   , reml cov(un) nostderr
```

By default, `bootstrap` will report normal-based 95 percent confidence intervals (i.e., intervals based on the bootstrap estimate of the standard error and the 2.5th and 97.5th percentiles of the standard normal distribution). Other types of confidence intervals can be obtained using `estat bootstrap`.

PART G: OPTIMIZATION OF FLEXIBLE MIXED-EFFECTS MODELS IN R AND STATA

Here, we provide a detailed example, including syntax for R and Stata, of the optimization routine outlined in the main article and originally developed by Bates, Kliegl, Vasisth, and Baayen (2015)—herafter BKVB. The example focuses on the question of whether fear of crime declines with modernization, as captured by the HDI, that is, we consider the direct context effect of HDI on fear of crime. The replication files (available with the online supplements) provide annotated step-by-step code for all further illustrative analyses reported in the main article; that is, they document the optimization steps that led to the specifications referred to as optimized models in Table 4 in the main article.

Before we detail the optimization routine, we show how to fit invariant and maximally flexible mixed-effects models in R and Stata. The most widely used commands for estimating linear mixed-effects models in R and Stata are, respectively, the `lmer()` function of the `lme4` package (Bates, Maechler, et al. 2015) and the command `mixed`. The following examples present R and Stata code side by side, displaying R code and output in green and Stata code and output in blue text. The classical invariant random-intercept model that estimates the direct context effect of the HDI on fear of crime can be fit as follows:

```
model1 <- lmer(z_crime ~ womenWec + z_agea + maritalbWec +
  educ2Wec + educ3Wec + uemplaWec + z_hdi + (1 | cntry), data =
  ESS, REML = TRUE)
```

```
mixed z_crime womenWec z_agea maritalbWec educ2Wec educ3Wec
  uemplaWec z_hdi || cntry:, reml
```

These commands tell the respective programs to fit random-intercept models using restricted maximum likelihood (REML). REML is preferable to full maximum likelihood (ML) estimation when the number of clusters is small (Elff et al. 2016; Raudenbush and Bryk 2002), as it is in the present example with a sample of 28 countries. With the `lmer()` function in R, REML is the default, so we only request it for clarity. In Stata, one has to explicitly request REML estimation (because full maximum likelihood estimation is the default). The first variant of the command further tells R to store the results in the object `invariant` and to fit the model using the dataset ESS. In Stata, results

can be stored after fitting the model (using `estimates store`) and the data need to be loaded into memory.

In both cases, the first part of the input specifies the fixed part of the model, that is, the lower- and upper-level variables used to predict the outcome. We assume that the researcher has settled on the specification of the fixed part of the model based on theoretical considerations, previous research, exploratory analyses, and so forth. The routine described here is solely concerned with finding the optimal specification for the random part of the mixed-effects model.

The code for specifying the fixed part follows the conventions for OLS and other regression models in `R` and `Stata`, respectively; in our example, fear of crime is regressed on six lower-level and one country-level independent variables. The more generic parts of the mixed-effects commands are the terms `(1 | cntry)` in the `R` and `|| cntry:` in the `Stata` code. These terms define the random effects or variance components. Here they specify that the observations are clustered by `cntry` (the country of residence) and that only the intercept varies across countries (the `R` code indicates the intercept via the `1`, while `Stata` presupposes it). The results (not shown here in detail) tell us that the overall level of fear of crime declines with the HDI across the sample of 28 European countries (beta = –.259, se = .041).

To allow the slope coefficients to vary across countries, just like the intercept, we need to add the respective variables to the random effects part of the formula. Importantly, we should also allow the random intercept and the additional random slopes to correlate, unless we have good reasons to assume that the random effects vary independently (an essential part of the optimization routine described below is to explore if there are such reasons). With six lower-level predictors, the maximally complex model allows for six random slopes. It can be estimated as follows:

```
model2 <- lmer(z_crime ~ womenWec + z_agea + maritalbWec +
  educ2Wec + educ3Wec + uemplaWec + z_hdi + (1 + womenWec + z_agea
  + maritalbWec + educ2Wec + educ3Wec + uemplaWec | cntry), data =
  ESS, REML = TRUE)
```

```
mixed z_crime womenWec z_agea maritalbWec educ2Wec educ3Wec
  uemplaWec z_hdi || cntry: womenWec z_agea maritalbWec educ2Wec
  educ3Wec uemplaWec, covariance(unstructured) reml
```

Importantly, R by default estimates correlations between the intercept and all six random slopes, while we explicitly need to allow for these correlations when we use Stata. We achieve this via the option covariance(unstructured). Using R, this model converges with reasonable parameter estimates and no convergence warnings. In particular, there are no random effects with zero or near-zero variance, nor any near-to-perfect correlations among different random effects. As before, the model suggests that fear of crime declines with the HDI across the sample of 28 European countries (beta = –.277, se = .034), but the analytical standard error should be interpreted with caution, as the simulation results reported in the main article indicate that it likely is downward biased. Stata issues convergence warnings, and as discussed in the main article there is good reason to believe that the maximally complex model is indeed too demanding for a sample of just 28 countries (a total of 28 random-effects variances and covariances and eight fixed effects need to be estimated). We therefore continue by describing our adaptation of BKVB's optimization procedure, which seeks to strike a balance between complexity and parsimony.

G.1 Model Optimization for the Sample of 28 European Countries

G.1.1 Step 1: Deletion of Random Slopes

The optimization procedure starts by estimating the maximally complex model. Following BKVB, we estimate all models using maximum likelihood (ML) instead of restricted maximum likelihood (REML) during the optimization phase.[3]

To speed up estimation, we also follow BKVB and remove any correlations between the random effects (i.e., assume the random effects to be uncorrelated) initially. This "zero-correlation model" is the baseline for identifying random effects with little explanatory power that can be removed with very little cost in terms of model fit. R users can request the zero-correlation model by specifying a

[3] ML is generally preferred for purposes of model comparison because it remains valid when one compares models with different fixed-effects specifications. As the fixed-effects specification does not change during the optimization procedure, REML would be defensible, but we nevertheless follow BKVB and use ML.

second vertical bar || in the random part of the model formula. In Stata, the zero-correlation model is actually the default.[4] To explicitly request it, one uses the option covariance(independent):

```
model3 <- lmer(z_crime ~ womenWec + z_agea + maritalbWec +
  educ2Wec + educ3Wec + uemplaWec + z_hdi + (1 + womenwec + z_agea
  + maritalbWec + educ2Wec + educ3Wec + uemplawec || cntry), data
  = ESS, REML = FALSE)
```

```
mixed z_crime womenWec z_agea maritalbWec educ2Wec educ3Wec
  uemplaWec z_hdi || cntry: womenWec z_agea maritalbWec educ2Wec
  educ3Wec uemplaWec, covariance(independent) mle
```

To decide whether to remove a random slope (and if so, which one) we compare the baseline model with several candidate models, each of which drops one of the random slopes included in the baseline model (as noted in the main article, we do not consider dropping random slopes on predictors that are part of a cross-level interaction, but this rule does not constrain us in the present case where we are interested in a DCE). At the beginning of the simplification process, the baseline model is the zero-correlation model with all possible random slopes. At later stages, it is the model chosen in the previous simplification step. To identify the random slope whose omission results in the biggest BIC improvement, we simply fit all candidate models (i.e., if the last model included five random slopes, we fit five zero-correlation models, each of which drops exactly one of the five random slopes).[5] We then compare the candidate models in terms of BIC. In R, BIC values can be obtained via the BIC() command (from the R package stats, which is part of the core distribution). In Stata, one uses the post-estimation command estat ic. If the best candidate model (i.e., the one with the lowest BIC value) has a better (i.e., lower) BIC value than the baseline specification, we prefer it to the latter. It becomes the new baseline model and the next iteration of the present step of the optimization procedure begins. If none of the candidate models improve BIC compared to the baseline model, we conclude this step of the optimization process and continue with Step 2, unless a principal component analysis (PCA) of the variance-covariance matrix of the baseline model indicates a need for further simplification. As noted in the main article, BKVB argue that the number of principal components that

[4] That is, unless the R. notation is used. See the Stata documentation of mixed for further details.

[5] Our replication files contain a simple R function searcher for convenient estimation of the candidate models. A similar loop is easy to implement in Stata.

cumulatively account for 100 percent of the variance of the random effects can be thought of as the maximum number of random effects supported by the data. Hence, if the number of random effects in the baseline model is greater than the number of independent principal components, further simplification is warranted even if it involves an increase in BIC. In such a case, we choose the candidate specification that yields the best BIC value as the new baseline model and begin with the next iteration of the present step of the optimization procedure In `R`, we can use the `rePCA()` function from the `RePsychLing` package accompanying BKVB's paper.[6] `Stata` users can use our postestimation command `repca.ado`, which is part of the replication files with the online supplements.

This iterative deletion of random effects closely follows BKVB's routine, but we deviate from their algorithm in two respects. First, as noted earlier, we do not consider deletion of the random intercept or a random slope for a variable that is part of a cross-level interaction (in our case, the random slope on high education when we fit the model with cross-level interaction). A primary reason for this decision is that these random effects are typically important for achieving accurate statistical inference on the effects of contextual variables. Second, whereas BKVB primarily rely on Likelihood Ratio tests for deciding whether to drop a random effect, we found it more useful to focus on changes in BIC. The reason is that we are dealing with much larger datasets than BKVB, who mainly use experimental data with the number of lower-level observations typically falling into the hundreds or lower thousands. In the larger datasets we deal with, using likelihood ratio tests as the criterion typically leads to no model simplification at all. AIC and especially BIC tend to penalize additional parameters more harshly.[7]

Returning to our example, we now examine which of the simpler candidate models achieves the largest reduction in BIC compared to the baseline model (for full details, please see the replication code). We find that dropping the random slope on the indicator variable for intermediate education yields the largest improvement (the BIC is 102103.2 for the baseline model and 102096.6 for the one

[6] At the time of writing, the `RePsychLing` package was available only as a development version. To use it, one first needs to install and load the `devtools` package and then install `RePsychLing` using `install_github("dmbates/RePsychLing")`.

[7] AIC penalizes additional parameters with a factor of 2, whereas BIC uses a factor of log(n) (Müller, Scealy, and Welsh 2013).

without the random slope on intermediate education). In R and Stata, one can estimate the zero-correlation model without the random slope term on intermediate education and obtain the BIC values as follows:

```
model4 <- lmer(z_crime ~ womenWec + z_agea + maritalbWec +
  educ2Wec + educ3Wec + uemplaWec + z_hdi + (1 + womenWec + z_agea
  + maritalbWec + educ3Wec+ uemplaWec  || cntry), data = ESS, REML
  = FALSE)
BIC(model3, model4)
```

```
mixed z_crime womenWec z_agea maritalbWec educ2Wec educ3Wec
  uemplaWec z_hdi || cntry: womenWec z_agea maritalbWec educ2Wec
  educ3Wec uemplaWec, mle
estat ic
mixed religiosity womenWec z_agea maritalbWec educ2Wec educ3Wec
  uemplaWec z_hdi || cntry: womenWec z_agea maritalbWec educ3Wec
  uemplaWec, cov(ind) mle
estat ic
```

The next step is to consider a second round of simplification, with the zero-correlation model without a random slope on intermediate education becoming the new baseline model. For this model, we again drop the (remaining five) random slopes one at a time, and see if doing so leads to further improvements in BIC. Iterating this exercise indicates that further improvements are possible by consecutively dropping the random slopes on unemployment and marital status. We finally derive the following simplified zero-correlation model, which cannot be further improved (in terms of BIC) by dropping one of the remaining random slopes:

```
model5 <- lmer(z_crime ~ womenWec + z_agea + maritalbWec +
  educ2Wec + educ3Wec + uemplaWec + z_hdi + (1 + womenWec + z_agea
  + educ3Wec || cntry), data = ESS, REML = FALSE)
```

```
mixed z_crime womenWec z_agea maritalbWec educ2Wec educ3Wec
  uemplaWec z_hdi || cntry: womenWec z_agea educ3Wec, mle
```

We now subject this model's random-effects covariance matrix to a principal component analysis (PCA) to see whether all remaining random effects are supported. The R command and output look as follows

```
summary(rePCA(model5))
$cntry
Importance of components:
                          [,1]    [,2]    [,3]    [,4]
Standard deviation     0.2321 0.09945 0.07812 0.05920
Proportion of Variance 0.7342 0.13484 0.08319 0.04778
Cumulative Proportion  0.7342 0.86903 0.95222 1.00000
```

In Stata, we would use the following two commands (output omitted):

```
mixed z_crime womenWec z_agea maritalbWec educ2Wec educ3Wec
  uemplaWec z_hdi || cntry: womenWec z_agea educ3Wec, mle
repca
```

The PCA suggests that the data support all four remaining random effects. Thus, the first step of the optimization procedure is complete and we turn to the second step, which reintroduces and then aims to simplify the correlation structure.

G.1.2 Step 2: Re-introduction and simplification of correlations among random effects

In this step, we investigate whether we can improve upon the (reduced) zero-correlation model by (re)introducing correlations between the remaining random effects. To do so, we first estimate a model that allows for correlations between all random effects. If no random slopes were pruned in the previous step, this model will be the maximally flexible model—otherwise it will be simpler. We initially test again if the PCA supports all random effects (output not shown). In the present case, the answer is yes. Occasionally, the model with correlations may not be supported by the data (i.e., the number of principal components that fully account for the variance of the random effects will be smaller than the number of random effects). In this case, an obvious solution is to return to step one and remove another random slope. Sometimes, however, the problem may also disappear when the covariance structure is simplified in the course of the present step of the optimization procedure.

To reintroduce the full set of correlations among the remaining random effects (the random intercept and three random slopes), we simply use the single pipe | rather than the double pipe || operator in R. In Stata, we specify the covariance(unstructured) rather than the covariance(independent) option.

Even with only three random slopes (and the random intercept) remaining, the unconstrained covariance matrix still contains 10 parameters (four variances and six covariances/correlations). The aim of the present step of the optimization procedure therefore is to simplify the model by deleting weak correlations between random effects (i.e., to constrain them to be zero). To identify promising options for simplification, we examine the variance-covariance matrix of random effects. The R output looks as follows:

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 cntry    (Intercept) 0.044569 0.21111
          womenWec    0.004947 0.07033  -0.12
          z_agea      0.002969 0.05449   0.46 -0.07
          educ3Wec    0.008012 0.08951  -0.17 -0.43  0.06
 Residual             0.818636 0.90479
```

A close look reveals that the random slopes for both the women and high education coefficients correlate only weakly with the intercept (r = –.12 and r = –.17) and the random slope of age (r = –.07 and r = .06). Moreover, the random slope of age and the random intercept are quite strongly interrelated (r = –.46). This suggests it might be possible to improve the model by separating the two blocks of interrelated random effects. Fitting such a model, albeit uncommon in applied sociological work, is easily possible with both lmer() and mixed by specifying several groups of random effects that refer to the same level of clustering. The random effects from different groups are always assumed to be independent.

To estimate the candidate model with two blocks of potentially correlated random effects (one consisting of the intercept and age, whereas the other consists of the high education indicator and the indicator for being female), one would use the following commands:

```
model6 <- lmer(z_crime ~ womenWec + z_agea + maritalbWec +
  educ2Wec + educ3Wec + uemplaWec + z_hdi + (1 + z_agea | cntry) +
  (0 + womenWec + educ3Wec | cntry), data = ESS, REML = FALSE)

mixed z_crime womenWec z_agea maritalbWec educ2Wec educ3Wec
  uemplaWec z_hdi || cntry: z_agea, cov(un) || cntry: womenWec
  educ3Wec, cov(un) nocons
```

These syntaxes specify a mixed-effects model with four random effects (a random intercept and three random slopes), grouped into two blocks. The model allows correlations among random effects within, but not across, these blocks. To restrict the correlation between the random intercept and the random slopes for being female and high education to zero, we need to specify the corresponding random-effects block with the `noconstant` option in `Stata`. In `R`, we add a leading `0` (rather than a `1)`. Separating blocks of random effects in the above manner simplifies the model considerably, reducing the number of random-effects parameters that need to be estimated from ten to six (compared to the model allowing for correlations between all random effects).

The identification of separate independent blocks of random effects is the primary means of model simplification in the current step of the optimization procedure. Each block can consist of one or several random effects. If one wants to specify several random effects as completely independent of all other random effects, the `||` operator and `covariance(independent)` options are convenient shorthands. For example, `(0 + z_agea + womenWec + educ3Wec || cntry)` is a shorthand for `(0 + z_agea | cntry) + (0 + womenWec | cntry) + (0 + educ3Wec | cntry)` in `R`. In `Stata`, `|| cntry: z_agea womenWec educ3Wec, covariance(independent) nocons` is equivalent to specifying `|| cntry: z_agea, nocons || cntry: womenWec, nocons || cntry: educ3Wec, nocons`. As illustrated by the above example, separating independent blocks of random effects can greatly reduce model complexity. As before, BIC can be used to judge between models.

Returning to our concrete example, the R output for the random-effects variance-covariance matrix of Model 6 looks as follows:

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 cntry    (Intercept) 0.044171 0.21017
          z_agea      0.002958 0.05439  0.47
 cntry.1  womenWec    0.004939 0.07028
          educ3Wec    0.008012 0.08951  -0.44
 Residual             0.818634 0.90478
```

According to the BIC, this model is better than the model with no constraints on the correlations (BIC = 102101.0, compared to BIC= 102141.5). To further improve the correlation structure between the random effects, we take another look at the updated variance-covariance matrix of Model 6. The two remaining correlations are relatively strong. But among them, the random slopes for being female and high education show the weakest association, and both of these random slopes are already specified as independent of the random intercept, which matters most for the estimation of the direct context effect of HDI. Perhaps we can further improve the model by specifying these random slopes as independent from all others:

```
model7 <- lmer(z_crime ~ womenWec + z_agea + maritalbWec +
  educ2Wec + educ3Wec + uemplaWec + z_hdi + (1 + z_agea | cntry) +
  (0 + womenWec + educ3Wec || cntry), data = ESS, REML = FALSE)

mixed z_crime womenWec z_agea maritalbWec educ2Wec educ3Wec
  uemplaWec z_hdi || cntry: z_agea, cov(un) || cntry: womenWec
  educ3Wec, cov(independent) nocons
```

This second reduction of the random-effects variance-covariance matrix indeed further improves the BIC to 102094.6. With just one strong correlation between the random intercept and the random slope of age remaining, further simplification is hardly possible, and the second step of the optimization procedure concludes.

A final issue to consider when specifying independent blocks of random effects is that removing correlations between random slopes and the random intercept can render the model, and in particular

the estimated context effects, sensitive to the scaling of the lower-level independent variables. Grand mean centering (of continuous predictors) and weighted effect coding (of categorical predictors) safeguard against this issue but do not resolve it entirely. We therefore want to ensure that the findings concerning the effect of the contextual variable(s) of interest—in our case, the HDI—do not change dramatically when correlations between random effects are constrained to zero.[8] Although this possibility may appear to be a serious threat, it is important to note that the procedure for removing correlation parameters outlined in the previous step already reduces the risk considerably (removing crucial correlations should depress performance in terms of BIC or other criteria). As a simple test, we suggest to check whether the deletion of correlations between random slopes strongly alters the context effect in question (and to conduct similar checks repeatedly during the second step of the optimization procedure, comparing estimates before and after deletion of one or several correlations). Some differences are to be expected due to chance (and differences in precision), but if removing correlations results in exceptionally large changes in the size of contextual effects, one should stick with the more complex model that includes the correlations or resort to a simpler model that drops the random slope altogether.

Returning to our concrete example, the model that allows for all correlations among the four random effects remaining after step 1 estimates a HDI context effect of beta = –.277, whereas the optimized Model 7 suggests beta = –.252. We conclude that there is no reason to be concerned about scale dependence in the present case and that Model 7 is the optimal specification.

[8] This complication typically occurs when two conditions hold. First, the contextual variable interacts with a lower-level variable (i.e., there is a cross-level interaction) and this interaction is not included among the predictors (i.e., the fixed part of the model). Second, the model includes a random slope on the lower-level variable, but does not allow the random slope to correlate with the intercept. Intuitively, the explanation is that the unspecified interaction results in a situation where, for any given level of the contextual variable, the fixed part of the mixed-effects model systematically overpredicts the outcome for certain values of lower-level predictor, while systematically underpredicting for other values. Moreover, the extent of over-/under-prediction is systematically related to the value of the contextual variable. When the random slope on the lower-level predictor is allowed to correlate with the random intercept, the random effects can absorb this systematic pattern, and the coefficient estimate on the contextual variable reliably estimates its average (= direct context) effect. When the correlation is restricted to zero, this is no longer possible, and the coefficient on the context variable effectively estimates a potentially conditional effect for the case that the lower-level predictor equals zero. In such a case, one might want to model (and theorize) the so far unmodeled cross-level interaction explicitly. In other words, an unanticipated benefit of "failing the unaltered context-effect test" may be that we detect cross-level interactions that are worthy of further investigation and that otherwise might have been hidden within the covariances of random effects.

G.1.3 Step 3: Re-estimate optimized models via REML and Bootstrap

As a final step, we re-estimate the optimized model using REML and obtain non-parametric cluster bootstrap confidence intervals (see Part F above).

REFERENCES

Arai, Mahmood. 2015. "Cluster-Robust Standard Errors Using R." Stockholm University (www.ne.su.se/polopoly_fs/1.216115.1426234213!/menu/standard/file/clustering1.pdf).

Bates, Douglas, Reinhold Kliegl, Shravan Vasishth, and Harald Baayen. 2015. "Parsimonious Mixed Models." *ArXiv* Preprint ArXiv:1506.04967. Retrieved April 22, 2016 (http://arxiv.org/abs/1506.04967).

Bates, Douglas, Martin Maechler, Ben Bolker, and Steven Walker. 2015. *Lme4: Linear Mixed-Effects Models Using Eigen and S4.* R Package Version 1.1-9 (https://CRAN.R-project.org/package=lme4).

Canty, Angelo, and Brian Ripley. 2016. *Boot: R Package Version 1.3-18* (https://cran.r-project.org/web/packages/boot/).

Davison, A. C., and D. V. Hinkley. 1997. *Bootstrap Methods and Their Application*. Cambridge, UK: Cambridge University Press.

Efron, Bradley, and Robert John Tibshirani. 1993. *An Introduction to the Bootstrap*. Boca Raton, FL: Chapman & Hall.

Elff, Martin, Jan P. Heisig, Merlin Schaeffer, and Susumu Shikano. 2016. "No Need to Turn Bayesian in Multilevel Analysis with Few Clusters: How Frequentist Methods Provide Unbiased Estimates and Accurate Inference." *SocArXiv/Open Science Framework* (Version 2, December 10, 2016; https://osf.io/preprints/socarxiv/z65s4/).

Enders, Craig K., and Davood Tofighi. 2007. "Centering Predictor Variables in Cross-Sectional Multilevel Models: A New Look at an Old Issue." *Psychological Methods* 12(2):121–38.

European Social Survey (ESS) Round 6. 2016. *ESS-6 2012 Documentation Report,* ed. 2.2. Bergen: European Social Survey Data Archive, NSD - Norwegian Centre for Research Data for ESS ERIC.

European Social Survey (ESS) Round 6. 2012. *Data File Edition 2.2*. NSD - Norwegian Centre for Research Data, Norway – Data Archive and distributor of ESS data for ESS ERIC.

Goldstein, Harvey. 2011. "Bootstrapping in Multilevel Models." Pp. 163–71 in *Handbook of Advanced Multilevel Analysis*, edited by J. J. Hox and J. K. Roberts. New York: Routledge.

Grotenhuis, Manfred, Ben Pelzer, Rob Eisinga, Rense Nieuwenhuis, Alexander Schmidt-Catran, and Ruben Konig. 2016. "When Size Matters: Advantages of Weighted Effect Coding in Observational Studies." *International Journal of Public Health* 62(1):163–67.

Lewis, Jeffrey B., and Drew A. Linzer. 2005. "Estimating Regression Models in Which the Dependent Variable Is Based on Estimates." *Political Analysis* 13(4):345–64.

Loy, Adam, and Martin Steele. 2016. *Lmeresampler: Bootstrap Methods for Nested Linear Mixed-Effects Models.* R Package Version 0.1.0 (https://cran.r-project.org/web/packages/lmeresampler/).

Müller, Samuel, J. L. Scealy, and A. H. Welsh. 2013. “Model Selection in Linear Mixed Models.” *Statistical Science* 28(2):135–67.

R Core Team. 2015. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing (http://www.R-project.org/).

Raudenbush, Stephen W., and Anthony S. Bryk. 2002. *Hierarchical Linear Models: Applications and Data Analysis Methods*. Second. Thousand Oaks, CA: Sage Publications.

Ren, Shiquan, Hong Lai, Wenjing Tong, Mostafa Aminzadeh, Xuezhang Hou, and Shenghan Lai. 2010. “Nonparametric Bootstrapping for Hierarchical Data.” *Journal of Applied Statistics* 37(9):1487–98.

United Nations Development Programme, ed. 2015. *Human Development Report 2015*. New York: United Nations Development Programme.