Journal of Behavioral Data Science, 2026, *6* (2), 1–67.
DOI: https://doi.org/10.35566/jbds/gomer2026

Mosaic Monte Carlo: A New Method of Simulation Design to Improve the Generalizability of Findings

Brenna Gomer[1 [0000−0002−4370−0256]], Han Byul Lee[2 [0009−0000−1701−9304]], and Young Min Kim[1 [0000−0002−0957−388X]]

[1] University of Utah, Salt Lake City, UT 84112, USA
brenna.gomer@psych.utah.edu, youngmin.kim@utah.edu
[2] University of Southern California, Los Angeles, CA 90089
hanbyull@usc.edu

Abstract. Monte Carlo simulation studies are an essential tool to test the performance of statistical methods. They are often implemented by generating data from a small number of data-generating models for a large number of replications. However, it is not guaranteed that a statistical method tested on a handful of data-generating models will perform well in other scenarios. Simulations are necessarily limited in scope, and so it is all too possible for results to unknowingly fail to generalize to real applications. This issue of generalizability is particularly relevant for methods that are more sensitive to parameter values such as those used in missing data analysis and Bayesian statistics. In this paper, we propose a new type of simulation design called Mosaic Monte Carlo that can help improve the generalizability of Monte Carlo simulation studies to real world applications. This method implements simulations by breaking up replications into smaller subsets, each using a different data-generating model. This approach to simulation design improves the generalizability of results beyond traditional designs.

Keywords: Simulation design · Monte Carlo · Generalizability

1 Introduction

Monte Carlo simulation studies are the bread and butter of testing the performance of statistical methods. They are often implemented by choosing a number of factors to manipulate – such as sample size, magnitude of correlations, and model size – and then randomly generating data under the corresponding data-generating model(s) across a large number of replications. Research on topics such as pharmacology, structural equation modeling, quality control, and many others use Monte Carlo approaches to simulate models, compute statistical precision and accuracy, and evaluate new methods (Curran, Bollen, Paxton, Kirby, &

Chen, 2002; Goutelle et al., 2009; Iranmanesh, Parchami, & Sadeghpour Gildeh, 2022).

Various concerns about the design and implementation of Monte Carlo simulation studies have been raised in recent years. Choosing the appropriate sample size range is one potential challenge (Paxton, Curran, Bollen, Kirby, & Chen, 2001; Skrondal, 2000; Spence, 1983). Other concerns include non-converged results and appropriate selection of software (Paxton et al., 2001; Spence, 1983). Concerns and guidance about best practices regarding the design and implementation of Monte Carlo simulation studies more broadly have been discussed by Morris, White, and Crowther (2019) and Siepe et al. (2024). Morris et al. (2019) propose an ADEMP structure (an acronym for Aims, Data-generating mechanism, Estimands, Methods, and Performance measures) to aid in planning simulation studies, as shown in Table 1. The ADEMP structure is implemented in modern statistical simulation studies to ensure reproducibility. Siepe et al. (2024) expand on this framework tailored to an audience of psychology researchers. Burton, Altman, Royston, and Holder (2006), who defined the essential preparatory procedures for simulations in medical statistics, historically contributed to the shift toward these standardized practices.

Another issue that has garnered attention is the number of replications used in Monte Carlo simulation studies (Schaffer & Kim, 2007). Brooks (2002) states that if not enough replications are utilized in simulations, the results could be misleading. A few studies (Preecha, 2004; Supawan, 2004) suggest guidelines for the number of replications for common statistical analyses such as ANOVA and regression. However, a recent development that has gained traction is the use of so-called Monte Carlo Standard Errors(MCSEs; Koehler, Brown, & Haneuse, 2009). Koehler et al. (2009) propose using these to quantify the uncertainty in Monte Carlo designs. In their article, the authors found that there can be unexpectedly substantial uncertainty in results even with 1000 replications. The authors suggest that several thousand replications may be needed at times in order to obtain good precision of estimates. Of course, the number of replications needed for a particular simulation study will be context-dependent, but MCSEs can be helpful for determining how many replications are adequate. Monte Carlo standard errors are now recommended to be part of the standard output reported in simulation studies (Morris et al., 2019; Siepe et al., 2024).

In contrast, Skrondal (2000) argues that one of the main issues plaguing simulation studies is external validity rather than precision. Boulesteix, Stierle, and Hapfelmeier (2015) state that methodological research remains deeply vulnerable to publication bias, which is the systematic suppression of negative data in favor of over-optimistic conclusions. In addition, standard computational literature frequently suffers from optimization bias, in which authors evaluate their own procedures under non-neutral, overly favorable conditions (Boulesteix, Lauer, & Eugster, 2013). A main contributing factor of this issue is the choice of the data-generating mechanism, which serves as the primary component affecting variation across simulation conditions (Boulesteix et al., 2020).

Table 1: Key steps and decisions in the planning, coding, analysis and reporting of simulation studies reprinted from Morris et al. (2019)

Planning
Aims
– Identify *specific* aims of simulation study.
Data-generating mechanisms
– In relation to the aims, decide whether to use resampling or simulation from some parametric model.
– For simulation from a parametric model, decide how simple or complex the model should be and whether it should be based on real data.
– Determine what factors to vary and the levels of factors to use.
– Decide whether factors should be varied fully factorially, partly factorially or one-at-a-time.
Estimand/target of analysis
– Define estimands and/or other targets of the simulation study.
Methods
– Identify methods to be evaluated and consider whether they are appropriate for estimand/target identified. For method comparison studies, make a careful review of the literature to ensure inclusion of relevant methods.
Performance measures
– List all performance measures to be estimated, justifying their relevance to estimands or other targets.
– For less-used performance measures, give explicit formulae for the avoidance of ambiguity.
– Choose a value of n_{sim} that achieves acceptable Monte Carlo SE for key performance measures.
Coding and Execution
– Separate scripts used to analyze simulated datasets from scripts to analyze estimates datasets.
– Start small and build up code, including plenty of checks.
– Set the random number seed once per simulation repetition.
– Store the random number states at the start of each repetition.
– If running chunks of the simulation in parallel, use separate streams of random numbers.
Analysis
– Conduct exploratory analysis of results, particularly graphical exploration.
– Compute estimates of performance and Monte Carlo SEs for these estimates.
Reporting
– Describe simulation study using ADEMP structure with sufficient rationale for choices.
– Structure graphical and tabular presentations to place performance of competing methods side-by-side.
– Include Monte Carlo SE as an estimate of simulation uncertainty.
– Publish code to execute the simulation study including user-written routines.

However, it is common practice in Monte Carlo simulations to choose few sets of population parameters, especially in cases when they are not the topic of interest and not thought to influence the performance of the statistical method being studied. The value of parameters are often chosen arbitrarily or without justification, as in Collins, Schafer, and Kam (2001); Gomer, Jiang, and Yuan (2019); Hu and Bentler (1999), and many more. Alternatively, they may be based on values from a real dataset, as in Ke and Wang (2015). However, neither of these approaches can guarantee that findings will generalize to other data-generating models. Not only is there potential for misspecification of population parameters according to Paxton et al. (2001), but Kulinskaya, Hoaglin, and Bakbergenuly (2021) show that even modifying subtle assumptions in data generation (i.e., such as shifting between fixed and random intercept models) can also affect the apparent performance and ranking of statistical methods.

Morris et al. (2019) explicitly ask whether a chosen data-generating mechanism may favor some methods over others, and how the choice of data-generating mechanism can be checked and justified. However, even when researchers do justify such choices, some areas of study may be particularly sensitive to the choice of parameter values, such as missing data analysis and statistical approaches that use Bayesian methodology. For example, consider the simulation studies in Gomer and Yuan (2021) involving a multiple regression model. In Study 1, regression coefficients were set to either $(0, 0, 0, 0)'$ or $(0, 1, 1, 1)'$ and predictors were set to be correlated or uncorrelated. When comparing two versions of the missing not at random (MNAR) mechanism (focused MNAR and diffuse MNAR 1a), it would appear that one systematically yields more biased results than the other across these conditions. However, in a small simulation study in the supplemental material, the regression coefficients are arbitrarily set to either $(0, 0, 0, 0)'$ or $(0, 2, 2, 2)'$ and the pattern reverses. The difference in coefficient values between Study 1 and the supplemental material appears trivial. It would be hard to anticipate that such a change would be influential, especially since the relationships between variables are all the same. If this paper had been smaller in scope and focused only on Study 1, it would have been easy to draw an entirely wrong conclusion from the findings.

Critically, this is an example in the literature where the choice of parameter values can play a significant role in the overall conclusion of an article, *even though they may not seem related to the topic being studied*. Methodologists design simulation studies with the belief that simulation factors could have systematic effects on the outcome of interest. However, there may be other scenarios, particularly in missing data analysis and Bayesian contexts, in which small adjustments to parameter values influence simulation results in ways that are *nonsystematic*. We will show an example of this in a later section. If these factors are not varied in a simulation design then generalizability will be threatened and recommendations become questionable. However, interpreting findings that vary nonsystematically is another difficulty.

Franklin, Schneeweiss, Polinski, and Rassen (2014) and Schreck, Slynko, and Saadati (2024) show an alternative methodological approach to addressing the

external validity of simulation studies using plasmode simulations. In a plasmode simulation, real empirical datasets are used to anchor the data-generating mechanism by resampling observed covariate data (Franklin et al., 2014). This preserves the complex, real-world dependence structures that are often difficult to synthesize parametrically (Schreck et al., 2024). An investigator-specified outcome-generating model is then applied to the resampled data to introduce a known "truth," such as a specific treatment effect or effect size. While plasmode simulation achieves realism by grounding the simulation in a complex empirical reality, its generalizability remains bounded by the specific characteristics and representativeness of the chosen source sample (Schreck et al., 2024).

Nevertheless, it is difficult to test multiple data-generating models and parameters with Monte Carlo simulations due to computational burden and the need to summarize results. For example, let us consider a basic regression Monte Carlo simulation with 4 sample sizes, 4 probability distributions, 4 sets of regression coefficients, and 4 sets of predictor correlations. With 1,000 replications, this simulation requires calculations on 256,000 datasets. If the statistical method being studied requires iteration (i.e., multiple imputation, Bayesian estimation) then this design is already computationally intensive. What's more, this design still considers a very limited set of data-generating models and still faces issues of generalizability. There are also 256 distinct simulation factors to sift through and summarize in a meaningful way. In many cases, it is simply not feasible to consider a wider range of data-generating models.

Skrondal (2000) suggests expanding experimental factors and using fewer replications in order to address the issue of computational complexity. He also illustrates how to use fractional factorial designs in order to save computational time. However, issues related to summarizing large sets of results and nonsystematic relationships remain. Leigh and Bryant (2015) focus on a so-called "Bayesian style" approach to selecting parameter values in simulation studies instead of grid-based approaches to address issues of computational demand. In their approach, a posterior distribution of parameters is reported conditional on a particular simulation outcome. This also sidesteps issues related to summarizing results. In their approach, parameters are sampled using an algorithm based on importance sampling which selects parameters adaptively rather than over an entire grid. The suggestions in both articles are aimed at solving the generalizability issue of Monte Carlo simulation studies without placing significant burdens on computational complexity.

In this article, we test a new type of simulation design with the goal of improving the generalizability of Monte Carlo simulation studies. We call this method "Mosaic Monte Carlo." In art, a mosaic is a pattern or an image that is created from many small pieces. In Mosaic Monte Carlo, the idea is to break up the replications of a given simulation condition into smaller subsets, each using a different data-generating model. For example, instead of using the same data-generating model across 1,000 replications, we use 20 data-generating models with 50 replications each. The data-generating models act like the small mosaic pieces in the art metaphor. In the context of a multiple regression model, we

might choose to vary the values of the regression coefficients and the predictor correlations. The performance measures calculated from the data-generating models – for example, bias, variance, and type I error rates – can then be summarized and presented for the simulation condition of interest, which is similar to how small mosaic pieces are used to form an overall image or pattern in art.

The primary purpose of Mosaic Monte Carlo is to account for factors that have a nonsystematic impact on the outcome(s) of interest, thus improving generalizability of simulation studies. While researchers often can generate a list of experimental factors that have systematic impacts on their performance measures before conducting their simulation study, it may be more difficult to predict what aspects of data-generating mechanisms lead to nonsystematic impacts. Thus, we propose using Mosaic Monte Carlo as a two-stage procedure involving a *screening phase* and a *simulation phase*. The screening phase helps researchers to identify which aspects of data-generating mechanisms have an impact on their performance measures and whether these effects are systematic or nonsystematic. The simulation phase incorporates this information, conducts the simulation study of interest, and summarizes the results.

In a similar vein as Skrondal (2000), we propose using fewer replications; however, we do so across factors that are not experimental conditions in order to retain precision across those dimensions. An advantage of Mosaic Monte Carlo over that of Leigh and Bryant (2015) is that our method does not require knowledge of Bayesian statistics. Mosaic Monte Carlo also is designed to account for nonsystematic relationships between parameter values and the outcome(s) of interest, whereas the approach suggested by Leigh and Bryant (2015) assume the presence of a systematic relationship. An advantage of Mosaic Monte Carlo over plasmode simulations (Franklin et al., 2014; Schreck et al., 2024) is that while both plasmode simulation and Mosaic Monte Carlo share the fundamental goal of improving the generalizability of methodological findings beyond narrow, arbitrary parameter specifications, Mosaic Monte Carlo utilizes a broad, synthetic parameter space, rather than anchoring to one empirical dataset. This strategy mitigates the nonsystematic impacts of arbitrary parameter choices without relying on real-world data availability.

The remainder of this article is organized as follows. First, we describe how to implement Mosaic Monte Carlo in practice. Then, we perform a small demonstration that highlights how Mosaic Monte Carlo can be used in a context with missing data. Next, we present the methods and results for a simulation study that tests the baseline performance of Mosaic Monte Carlo. We also provide an empirical example that uses Mosaic Monte Carlo to replicate findings from a simulation study in the context of structural equation modeling. We conclude with a discussion of our main findings and recommendations.

2 How to implement Mosaic Monte Carlo

In this section, we describe how Mosaic Monte Carlo can be implemented in practice as a two-stage procedure. Some example code is also provided in a

github repository which can serve as a reference. As a note, the code is written for clarity rather than computational efficiency. This procedure is applicable across modeling contexts but we have written the guidance under the assumption that at least some of the estimands of interest are parameter estimates and thus that MCSEs can be calculated. However, Mosaic Monte Carlo can still be used for other targets of simulation studies, as we illustrate later in our empirical example.

2.1 Terminology

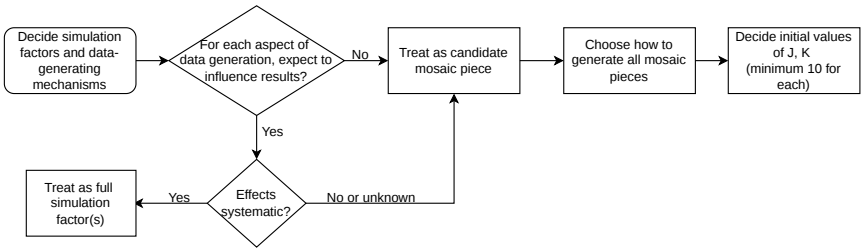
Throughout the remainder of this paper, we will use terminology consistent with that used in the ADEMP structure developed by Morris et al. (2019). Aspects of data generation that are varied in Mosaic Monte Carlo are called *mosaic pieces*. In the screening phase, these are referred to as *candidate mosaic pieces* until the researcher verifies that these aspects of data generation impact results nonsystematically.

2.2 Stage 1: Screening phase

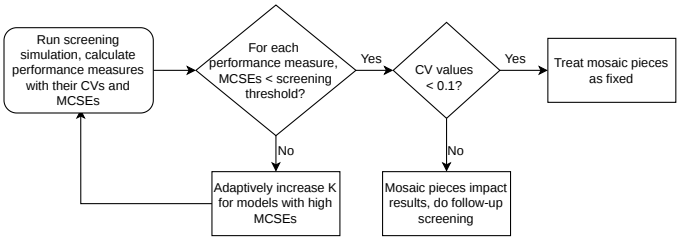
The purpose of the screening phase in Mosaic Monte Carlo is to help researchers distinguish factors that have systematic vs. nonsystematic effects on their simulation results and choose how to allocate replications across data-generating models. The flowchart in Figure 1 breaks this process up into 3 steps. We will discuss each of these steps in the subsections below.

Step 1: Setting up the screening phase In the first step in Figure 1(a), the researcher makes a series of decisions about the design of his simulation study. Researchers following the ADEMP framework for planning simulations (Morris et al., 2019) will complete all steps before conducting the screening phase of Mosaic Monte Carlo. Then, researchers revisit the data-generating mechanisms step and consider whether these factors are expected to influence any targets of the simulation study. Unless a particular factor is expected to systematically impact results, it should be treated as a candidate mosaic piece and investigated further rather than being held as fixed. When choosing population parameters, the following categories are often appropriate candidate mosaic pieces in Mosaic Monte Carlo:

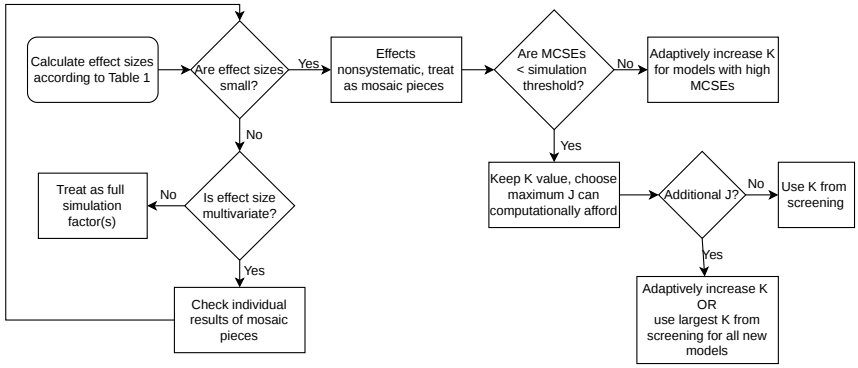
- **Population parameters that are not essential to the research aims**. An example of this could be the measurement error of variables in a simulation study examining the impact of model misspecification in the context of confirmatory factor analysis. A second example could be the difficulty parameter of items in a simulation study seeking to identify the breaking point of estimation for long tests given to small sample sizes.
- **Population parameters that help address the research aims but whose values can be chosen arbitrarily**. An example of this could be



(a) Stage 1: Setting up the screening simulation



(b) Stage 2: Implementing the screening simulation



(c) Stage 3: Distinguishing systematic from nonsystematic effects and preparing for the simulation phase

Figure 1: Flowchart for steps of screening phase of Mosaic Monte Carlo

the value of regression coefficients in a simulation study examining bias of their estimates in the context of missing data. Another example could be the predictor correlations between variables in a simulation study examining the effects of multicollinearity. *In both cases, the specific values of the population parameters are related to the topic of study but the researcher hopes to generalize findings beyond the specific parameter values chosen.* In such cases, selecting multiple parameter values helps strengthen the findings.

Depending on the topic of interest, other aspects of data generation that are not population parameters could be candidate mosaic pieces. For example, a researcher examining the performance of a missing-data method with data that is Missing At Random (MAR) would need to choose how to create missing values. She may choose an interval selection method that removes values on a variable Y if the corresponding value on a second variable X is above a certain threshold. The value on X usually is chosen to obtain a specific percentage of missing values, but the researcher could vary which variable serves as the X variable if there are multiple candidates.

The next decision to be made in step 1 of the screening phase is to choose how to generate the mosaic pieces. In the case of population parameters, these should be generated such that they cover a broad but realistic range of values. A familiar method of doing this would be traditional random sampling. Uniform distributions are a great choice for parameter generation because the resulting parameters will be evenly spread across the range of possible values. In contrast, generating parameters from a normal distribution will result in values that are more tightly clustered towards the center, which is less desirable from a generalizability standpoint. However, other sampling methods such as Latin hypercube sampling and low discrepancy sequences can also be used to generate mosaic pieces which ensure the desired sampling space is adequately represented in its entirety (McKay, Beckman, & Conover, 2000; Tuffin, 1996). Another alternative is to expand plasmode simulation to include many empirical datasets, although such a task may be time-intensive.

If random sampling is used, the corresponding parameters for the distribution itself (analogous to hyperparameters in Bayesian statistics) must be chosen after the generating distribution has been selected. In the case of the uniform distribution, this means specifying meaningful boundaries. For example, correlations should occur on the interval [-1, 1] in order to be valid. If a researcher is interested in the effects of multicollinearity, the interval could be narrowed to [-1, -0.7] and [0.7, 1]. This allows the researcher to specify parameter values according to the topic of interest while still making use of Mosaic Monte Carlo. If a different probability distribution is preferred over the uniform distribution, our general recommendation would be to specify the distribution in such a way that there is as large a variance as possible. This will improve the generalizability of the study. However, care must be taken that this range is still representative of the realistic application space and does not draw in implausible regions. What constitutes a realistic application space will vary by context but could include things such as ensuring that the smallest factor loadings in a structural equation model meet guidelines for interpretability according to the criteria in Stevens (1992).

As mentioned previously, the idea of Mosaic Monte Carlo is not limited to parameter generation, although that is our main focus. When other aspects of simulation studies are used to form the mosaic, choices about how to generate the mosaic pieces may be more systematic. Revisiting our example of the researcher investigating the performance of a missing-data method with MAR data, she

may only have 4 variables that could play the role of X. In that case, she may alternate between all 4 possible candidates.

As a note, multiple types of population parameters and/or aspects of the simulation study can be varied simultaneously to create the mosaic. In other words, each data-generating model is likely to be constructed from several mosaic pieces, such as regression coefficients and correlation values. This will help to reduce the computational costs involved in the screening phase.

The last decision the researcher needs to make in this step is the initial number of data-generating models (J) and replications (K) to use in the screening simulation. At this point, the researcher should choose a large enough K to obtain reliable estimates of his performance measures and a large enough J to reliably calculate a standard deviation and to have sufficient sample size for analyses in step 3. We suggest using a minimum of 10 for both J and K. However, it is important to note that the starting value of K will be updated in the next step of the screening phase and that the value of J used in the screening phase can be increased in the simulation phase.

Step 2: Implementing the screening simulation In this step shown in Figure 1(b), the researcher conducts a small-scale simulation study that follows the same design of the simulation study he intends to run. This step determines which candidate mosaic pieces actually impact results and which can be treated as fixed later on in the simulation phase. Simulation factors and data-generating mechanisms already known not to interact with candidate mosaic pieces (i.e., based on previous studies in the literature) can be held constant in the screening simulation. In contrast, if these could have possible interaction effects or the effects are unknown, researchers can select several or all values of each factor/data-generating mechanism and run the screening simulation in a fully-crossed design. We acknowledge that this may not be computationally feasible in some applications. In such cases, we recommend researchers select values of factors/data-generating mechanisms that are most likely to elicit effects in the candidate mosaic pieces. For example, this could be the smallest sample size or the largest percentage of missingness.

When programming the simulation study, some care must be taken to ensure that the mosaic is implemented correctly. We will give some brief guidance on how this might be achieved using two alternative strategies. These strategies work in both stages of Mosaic Monte Carlo. The code provided in our supplemental materials provide examples under both strategies.

The first strategy for implementing Mosaic Monte Carlo is to create the mosaic pieces and data-generating models first, and then run the simulation study in a second and separate step. However, the specific parameter values used for candidate mosaic pieces should be saved so that they can be referenced later. For example, these values would be used for calculations of bias.

The second strategy is to create the mosaic pieces and data-generating models while running the simulation study simultaneously. For example, if iterating over looped code, candidate mosaic pieces would be randomly generated inside the

loop, incorporated into the data-generating model, and then used to create the datasets. In this strategy, parameter values for candidate mosaic pieces do not need to be saved if all the desired output is also calculated inside the loop.

When conducting the screening simulation, researchers should not only compute the values of performance measures but also calculate their MCSEs for each distinct data-generating model. These will be used to determine if the number of replications is adequate. The MCSEs can be calculated following the guidelines in Koehler et al. (2009) in order to obtain J separate MCSE values for each performance measure. Additionally, the coefficient of variation (CV) for each performance measure across the J distinct data-generating models should be calculated. This will be used to determine whether the candidate mosaic pieces actually impact results or if they can be treated as fixed. The formula for the CV in this context is provided below:

$$CV = SD(PM)/Mean(PM) = \frac{\sqrt{1/(n-1)\sum_{j=1}^{J}(PM_j - \overline{PM}_j)^2}}{\overline{PM}_j} \quad (1)$$

where PM stands for performance measure, $\overline{PM}_j = 1/J\sum_{j=1}^{J} PM_j$, and the values of the performance measures are calculated in the usual way for each distinct data-generating model.

After the screening simulation has been run, the researcher examines the MCSEs for each performance measure and determines if these values fall below the desired screening threshold. We cannot recommend a specific value for this screening threshold as MCSEs will vary by performance measure and context. However, MCSEs should be small enough to have confidence in using the performance measures to calculate CVs across the J data-generating models and effect sizes in step 3 of the screening phase. For data-generating models that yield MCSEs exceeding the screening threshold, the number of replications should be increased until acceptable MCSEs are obtained for each performance measure. This means that some data-generating models will have greater numbers of replications than others. This adaptive approach helps direct computational resources more efficiently.

Once acceptable MCSEs have been obtained in all data-generating models for all performance measures, the researcher next determines which candidate mosaic pieces actually impact results using CVs. A CV value less than 0.1 suggests that the performance measure has low variability across data-generating models and thus is relatively unaffected by the candidate mosaic pieces. If this is true for all performance measures, the candidate mosaic pieces can be treated as fixed in the simulation phase of Mosaic Monte Carlo and the screening phase ends without proceeding to step 3. CV values greater than 0.1 suggest that the candidate mosaic pieces do have an impact on results and these should be examined further in step 3 of the screening phase. In cases when performance measures have a target value of 0 (i.e., relative bias), CV values will be unduly inflated because the means are so small. In such cases, the SD of the performance measure can be used instead of the CV with the same 0.1 cutoff.

Step 3: Distinguishing systematic from nonsystematic effects and preparing for the main simulation study The last step in the screening phase of Mosaic Monte Carlo makes use of effect sizes in order to determine which candidate mosaic pieces have systematic vs. nonsystematic effects on results. As shown in Figure 1(c), this step also helps the researcher decide the numbers of replications and data-generating models to use in the simulation phase of Mosaic Monte Carlo. In this step, the researcher first calculates the effect sizes according to the guidance in Table 2 for each performance measure using the output of the screening simulation. The sample size for such analyses is J, the number of data-generating models. Referring to Table 2, performance measures such as bias and variance can be used as outcome variables in a multiple regression model with candidate mosaic pieces as predictors and R^2_{adj} serving as a multivariate effect size. If the value of R^2_{adj} exceeds the suggested cut-off, this suggests that one or more candidate mosaic pieces have a systematic impact on the performance measure and the values of individual regression coefficients should be examined. Candidate mosaic pieces corresponding to large regression coefficients likely have systematic impacts on the results and should be treated as full simulation factors in the simulation phase. In contrast, candidate mosaic pieces corresponding to small regression coefficients likely have nonsystematic impacts on results and thus should be treated as mosaic pieces in the simulation phase.

Table 2: Effect size criteria for distinguishing between systematic and nonsystematic effects in screening phase

Performance measure	Candidate mosaic piece	Modeling framework	Effect size	Suggested cut-off
continuous	categorical, numeric	multiple regression	R^2_{adj}	0.1
binary	categorical, numeric	logistic regression	pseudo R^2	0.1
categorical	categorical, numeric	multinomial regression	odds ratio?	1.5
proportion	categorical		Cohen's h	0.2

In this example, what constitutes a "large" vs. "small" regression coefficient depends upon the scale of the performance measure and the candidate mosaic pieces. It is thus up to the discretion of the researcher. However, we note that statistical significance should not be considered when making these decisions, as it is directly influenced by the number of data-generating models and thus is not a reliable criterion. As a note, candidate mosaic pieces that have nonsystematic effects on one performance measure but systematic effects on another should be treated as a full simulation factor in the simulation phase.

After determining which candidate mosaic pieces will remain mosaic pieces in the simulation phase and which will become full simulation factors, the researcher re-evaluates the MCSE values from the output of the screening simulation. Now, the question is whether or not these MCSEs are less than the desired simulation threshold. The simulation threshold should be small enough

to obtain adequate precision in the final reporting of simulation results, which is likely to be more strict than the number used for the screening threshold. For data-generating models whose MCSEs already meet this stricter criterion, the same number of replications can be used in the simulation phase. The number of replications should be increased for data-generating models whose MCSEs do not yet meet the simulation threshold until acceptable values are obtained. Once acceptable MCSEs have been obtained for all data-generating models and performance measures, the researcher should decide how many data-generating models can be computationally afforded. In cases when the researcher can afford greater J, MCSEs can be calculated for the new set of data-generating models and compared to the simulation threshold in order to determine appropriate K. Alternatively, the researcher could use the largest K obtained from the previous data-generating models on the new set.

2.3 Stage 2: Simulation phase

Conducting the simulation study in the second stage of Mosaic Monte Carlo likely involves making minor expansions to the screening simulation in the first stage. Actually, the output from step 3 of the screening phase with finalized K values can be re-used. The simulation study will need to be run anew for the set of new data-generating models and for any simulation conditions that were omitted from the screening phase.

The final task in the simulation phase of Mosaic Monte Carlo is to summarize the results across data-generating models. There are many possible approaches that could be used and the best approach will depend on the number of simulation conditions and the nature of the mosaic pieces. Some possible approaches include: 1) distributional summaries, 2) aggregation, and 3) frequencies.

Distributional summaries, such as boxplots, can be especially useful for gauging several aspects of performance simultaneously and provide nuanced information. For example, side-by-side boxplots comparing the relative bias of parameter estimates across several estimation methods can illustrate the accuracy of methods by the center of the boxplots. Reliability of each method can be assessed by the range, interquartile range, and presence of outliers. The overlap in boxplots between methods can indicate how advantageous one method is compared to the others. Aggregation can be useful to conserve space when less nuance is required or when the scope is relatively broad, such as constructing overall summaries or making general recommendations. However, care must be taken to aggregate only over mosaic pieces that impact results nonsystematically according to the screening stage. Frequencies can be a useful summary for binary performance measures such as nonconvergence rates or counting the number of times a method was identified as the best-performing method among several candidates. Since many methodological studies seek to answer multiple interrelated questions, it is likely that several summarizing strategies could be used together in one paper.

When aggregation is used as a summarizing strategy, calculations of bias and variance must be adjusted if the corresponding parameters form part of the

mosaic. The modified formula for calculating aggregated empirical bias of an estimator $\widehat{\theta}$ for θ is given by

$$\frac{1}{J}\sum_{j=1}^{J}\frac{1}{K}\sum_{k=1}^{K}\widehat{\theta}_{jk}-\theta_j \tag{2}$$

where j refers to a single data-generating model with J distinct data-generating models in total (i.e., the number of distinct data-generating models created from the mosaic) and k denotes a single replication with K replications in total. The formula for calculating aggregated empirical relative bias is given by

$$\frac{1}{J}\sum_{j=1}^{J}\frac{1}{K}\sum_{k=1}^{K}\frac{\widehat{\theta}_{jk}-\theta_j}{\theta_j} \tag{3}$$

Supposing that $\overline{\theta}_j = 1/k\sum_{k=1}^{K}\theta_{jk}$, Equations 1 and 2 will be unbiased estimators of the aggregate (relative) bias as long as all $\overline{\theta}_j - \theta_j$ are unbiased estimators of bias.

If standard errors or variance are part of the desired output for the simulation study, the corresponding population values should be fixed to a constant to ensure the data-generating models are comparable. To calculate the empirical variance of an estimator $\widehat{\theta}$ for a θ that is varied in the mosaic, the following modified formula can be used:

$$\frac{1}{J}\sum_{j=1}^{J}\sum_{k=1}^{K}\frac{(\widehat{\theta}_{jk}-\widehat{\theta}_j)^2}{K-1} \tag{4}$$

where $\widehat{\theta}_j$ is the mean of the estimates across all K replications for a given data-generating model j, $\widehat{\theta}_j = \frac{1}{K}\sum_{k=1}^{K}\widehat{\theta}_{jk}$. In other words, the user should calculate the variance of the estimates across all replications for a given data-generating model. Then, these variances should be averaged across all data-generating models J to obtain a single value. Since $1/(K-1)\sum_{k=1}^{K}(\widehat{\theta}_{jk}-\widehat{\theta}_j)^2$ is the formula for the sample variance of $\widehat{\theta}_{jk}$ values, Equation 3 inherits unbiasedness and consistency from the sample variance (Casella & Berger, 2001). However, despite these desirable finite-sample properties, each individual sample variance can be considerably noisy for small values of K. Equation 3 is also a mean so it is sensitive to outliers that may arise under such conditions.

Please note that the user should not simply take the variance of the $\widehat{\theta}_{jk}$ values across all replications and across all data-generating models simultaneously. When θ is part of the mosaic, this practice falsely treats the $\widehat{\theta}_{jk}$ values as though they are comparable across different data-generating models. More formally, and assuming that all data-generating models have the same K for simplicity,

$$\frac{1}{J}\sum_{j=1}^{J}\sum_{k=1}^{K}\frac{(\widehat{\theta}_{jk}-\widehat{\theta}_j)^2}{K-1} \neq \sum_{j=1}^{J}\sum_{k=1}^{K}\frac{(\widehat{\theta}_{jk}-\overline{\theta})^2}{JK-1}$$

where $\overline{\theta}$ is the mean of all $\widehat{\theta}_{jk}$ values, $\overline{\theta} = 1/JK \sum_{j=1}^{J} \sum_{k=1}^{K} \widehat{\theta}_{jk}$. After some mathematical manipulation expanding the sum of squares, the nonequivalency between the two formulas is even more apparent:

$$\frac{1}{J} \frac{\sum_{j=1}^{J} \sum_{k=1}^{K} \widehat{\theta}_{jk}^2 - K \sum_{j=1}^{J} \widehat{\theta}_{j}^2}{K-1} \neq \frac{\sum_{j=1}^{J} \sum_{k=1}^{K} \widehat{\theta}_{jk}^2 - JK\overline{\theta}^2}{JK-1}$$

2.4 Considering computational burden

Implementing Mosaic Monte Carlo designs will add computational burden to simulation studies. For researchers who already adapt the number of replications to obtain acceptable MCSEs, the additional burden comes from the screening phase of Mosaic Monte Carlo. We acknowledge that in some applications, it is simply not feasible to follow the procedure outlined in Figure 1 in its entirety. Compromises may need to be made on the number of factors varied in the mosaic, the number of data-generating models, and the number of replications. Compromises can also be made by reducing the levels of simulation factors and/or holding some simulation factors constant. It is important to note that attempts to implement a screening phase will improve the generalizability of findings even if screening is done on a small scale. We recommend to make the screening phase as exhaustive as can be afforded, while acknowledging that this may deviate from the guidance in the flowchart.

3 Demonstration of Mosaic Monte Carlo

This section contains a brief demonstration of Mosaic Monte Carlo and how it can be used when aspects of data generation have nonsystematic impacts on results. We hope this provides readers with additional clarity on the usefulness of Mosaic Monte Carlo and how it can be implemented in practice. The R code for this demonstration is available in a github repository.

Continuing discussion of Gomer and Yuan (2021) as an example, suppose we want to see if one MNAR subtype consistently leads to more biased regression coefficients than the other in a similar setting. Specifically, our research aims are to determine 1) whether diffuse MNAR leads to more biased regression coefficient estimates and 2) how often this occurs. We investigate the relative bias of the regression coefficient associated with missing values ($\widehat{\beta}_3$ in the original article). We calculate the relative bias of $\widehat{\beta}_3$ under both focused and diffuse MNAR (corresponding to diffuse MNAR 1a in the original article) when maximum likelihood is used as a missing-data method.

3.1 Stage 1: Screening Phase

We use the same simulation design as Study 1 in Gomer and Yuan (2021) but revisit aspects of data generation for candidate mosaic pieces. Our main concern is that the values of regression coefficients and predictor coefficients may have

nonsystematic impacts on the bias of $\widehat{\beta}_3$ so following the flowchart in Figure 1(a), we treat these as candidate mosaic pieces. We choose to generate regression coefficients from a uniform distribution $Unif(-5, 5)$ and predictor correlations from $Unif(-0.6, 0.6)$. To begin with, we will use $K = 10$ replications for each of $J = 60$ data-generating models. This value of J was chosen in order to have sufficient sample size for conducting the multiple regression analysis in step 3 of the screening phase assuming that there are 6 predictors (i.e., 3 correlations and 3 regression coefficient population values and following the rule of thumb that there should be 10 observations per predictor).

In step 2 of the screening phase, we run the screening simulation and calculate the MCSEs for the relative bias of $\widehat{\beta}_3$ for each data-generating model. We also calculate the CV of the relative bias across data-generating models under both focused and diffuse MNAR. Although relative bias has a target of 0, the setting of MNAR subtypes creates levels of relative bias that are noticeably different from 0 and thus will not unduly impact CV values in this case. For the purpose of screening, we will aim to have all MCSEs less than a 0.2 screening threshold.

After adaptively increasing K for data-generating models whose MCSEs exceeded the 0.2 screening threshold, we obtained acceptable MCSEs for all data-generating models with 310 replications. The CV of the relative bias of $\widehat{\beta}_3$ was 0.934 under focused MNAR and 4.2 under diffuse MNAR. According to the flowchart in Figure 1(b), this suggests that the candidate mosaic pieces do impact the results under both MNAR subtypes since the CV values are greater than 0.1. We thus proceed to step 3 of the screening phase.

In step 3, we conduct multiple regression analyses using the relative bias of $\widehat{\beta}_3$ under focused and diffuse MNAR as outcome variables and the parameters $\beta_1, \beta_2, \beta_3, \rho_{12}, \rho_{13}$, and ρ_{23} as predictors. The output for this analysis is shown in Table 3. In this table, we can see that the R^2_{adj} values are quite small and is even negative under diffuse MNAR. With poor model fit, the predictors do little to explain the variance in the bias of $\widehat{\beta}_3$. This suggests that the candidate mosaic pieces impact the bias of $\widehat{\beta}_3$ nonsystematically.

Table 3: Multiple regression results determining which candidate mosaic pieces affect results systematically in step 3 of screening phase

	Focused	Diffuse
Regression Coefficient Values		
(Intercept)	0.0390	0.2036
β_1	0.0009	-0.0205
β_2	-0.0033	-0.0053
β_3	0.0010	-0.0226
ρ_{12}	-0.0079	-0.4006
ρ_{13}	0.0188	-0.4206
ρ_{23}	-0.0260	0.0347
Model Fit		
R^2_{adj}	0.1024	-0.0522

The final task is to refine the number of J and K for the simulation phase of Mosaic Monte Carlo. Suppose we want to use a simulation threshold of 0.1 for our MCSEs. In this case, there are several data-generating models that need additional replications to meet this stricter cutoff. After adaptively increasing K for these models, we obtain acceptable MCSEs for all data-generating models with 1010 replications. Although this number may seem large, there was only one data-generating model that required this value.

3.2 Stage 2: Simulation Phase

The output of step 3 can be used for the simulation phase of Mosaic Monte Carlo. Recall that our research aims are to determine 1) whether diffuse MNAR leads to more biased regression coefficient estimates and 2) how often this occurs. To address the first aim, we used the values of relative bias of $\widehat{\beta}_3$ obtained under focused and diffuse MNAR across 60 data-generating models to construct boxplots and calculate Cohen's d. The boxplots are distributional summaries that provide nuanced information while Cohen's d provides a concise aggregate summary.

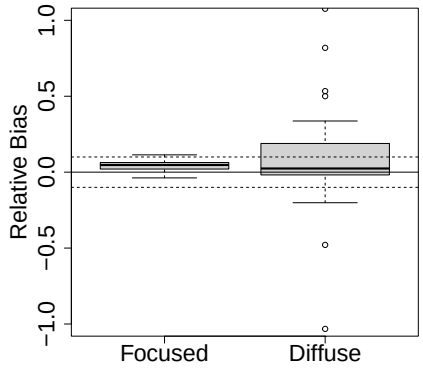


Figure 2: Boxplots of relative bias of $\widehat{\beta}_3$ under focused and diffuse MNAR across 60 data-generating models. Solid reference line drawn at 0 and dashed lines drawn at 10% relative bias.

The side-by-side boxplots in Figure 2 show that the range of relative bias is larger under diffuse MNAR than under focused MNAR. In fact, the relative bias of $\widehat{\beta}_3$ rarely exceeds 10% under focused MNAR whereas this occurs more commonly under diffuse MNAR. However, we obtained $d = 0.2699$. This is a small effect size, suggesting that the bias of $\widehat{\beta}_3$ is not substantially worse under diffuse

MNAR when we examine the trends overall. Taken together, these findings show that the bias of $\widehat{\beta}_3$ has potential to be much larger under diffuse MNAR than focused MNAR but that this does not occur regularly enough to make a blanket statement that diffuse MNAR leads to worse bias.

To address our second aim, we counted the proportion of times that the bias of $\widehat{\beta}_3$ was worse under diffuse MNAR compared to focused MNAR. Worse bias was obtained under diffuse MNAR 55% of the time, which translates to a Cohen's h of 0.2. This is also a small effect size, suggesting that the bias of $\widehat{\beta}_3$ is not worse more frequently under diffuse MNAR.

3.3 Summary

In this demonstration, Mosaic Monte Carlo was helpful for illustrating that bias is not always worse under a specific subtype of MNAR. Based on the analyses we performed, we were also able to identify that correlation and regression coefficient values had some nonsystematic impact on whether bias would be worse under diffuse MNAR compared to focused MNAR. Although diffuse MNAR has potential for leading to large levels of bias compared to focused MNAR, this did not occur often enough to warrant a bigger claim that diffuse MNAR is worse than focused MNAR.

4 Methods

In addition to the small demonstration in the previous section, a larger Monte Carlo simulation study was run in R (R Core Team, 2020) to assess possible detrimental impacts of using Mosaic Monte Carlo compared to traditional simulation design. In particular, we sought to determine whether aggregation can safely summarize information related to bias and variance estimates, especially when there is a large number of data-generating models relative to the number of replications. The code for this simulation study is provided in a github repository. We focus on our method's impact on bias and variance because these are two popular metrics used to evaluate the performance of statistical methods in Monte Carlo simulation studies. Specifically, we sought to address the following research questions:

- RQ1: To what extent does the aggregation strategy of Mosaic Monte Carlo impact the overall bias of statistical estimates in a simulation study?
- RQ2: To what extent does the aggregation strategy of Mosaic Monte Carlo impact the overall variance of statistical estimates in a simulation study?
- RQ3: To what extent does sample size and sampling variability affect the answers to RQ1 and RQ2?

We also evaluate Mosaic Monte Carlo under relatively ideal conditions in which estimates are unbiased. This will establish the baseline performance of Mosaic Monte Carlo.

4.1 Simulation design

The data-generating models used in this study were multiple regression models with 3 predictors of the form:

$$Y_{ij} = \beta_{0j} + \beta_{1j}X_{1j} + \beta_{2j}X_{2j} + \beta_{3j}X_{3j} + \varepsilon_{ij}$$

where the β_j coefficients were treated as candidate mosaic pieces and randomly generated from $Unif(0.25, 10)$. Here, i denotes individual data values with $i = 1, \ldots, n$ and j denotes a specific data-generating model with $j = 1, \ldots J$. The residuals were given by $\varepsilon_{ij} \sim N(0, \sigma_\varepsilon^2)$ where σ_ε^2 is set to 9, 25, and 100. The error variance was treated as a full simulation factor in order to make sure that the randomly generated data-generating models are comparable when analyzing the impact of our method on estimates of variance. The sample size n was also treated as a full simulation factor and set to 50, 75, 100, 200, and 400. The X variables were treated as fixed across replications and initially generated from a standard multivariate normal distribution using the R package `mvtnorm` (Genz et al., 2020), with correlation coefficients treated as candidate mosaic pieces and randomly generated from $Unif(0.1, 0.6)$. Note that this means there is no conceptual difference between the regression coefficients and that they are essentially exchangeable.

The total number of unique data-generating models J was set to be 1, 10, 25, 50, 100, 1000 and the number of replications for each data-generating model was set to be 1000, 100, 40, 20, 10, and 1 respectively as shown in Table 4. That is, the simulation condition that uses 1 distinct data-generating model was conducted with 1000 replications while the simulation condition that uses 100 distinct data-generating models was conducted with 10 replications for each model. Using this scheme, the total number of replications for each set of simulation conditions was always 1000 but consisted of a mixture of data-generating models. Note that the condition with 1000 replications and 1 distinct data-generating model serves as the control condition, as this corresponds to traditional simulation design. In contrast, the condition with 1 replication and 1000 data-generating models serves as a fully random condition. In Table 4, the ratio of data-generating models to replications increases from left to right, with the most extreme condition tested being 1000 data-generating models with 1 replication each.

Table 4: Combinations tested for the number of population models J and the number of replications K for Mosaic Monte Carlo.

Number of population models J	1	10	25	50	100	1000
Number of replications K	1000	100	40	20	10	1

The total number of simulation conditions tested was $n \times J \times \sigma_\varepsilon^2 = 5 \times 6 \times 3 = 90$ and each set of conditions (consisting of 1000 replications each) was run 500 times (we will refer to these as "test runs" to distinguish them from replications),

yielding $500 \times 1000 = 500,000$ replications in total for each set of simulation conditions.

4.2 Evaluation criteria

Bias As discussed previously, calculations of bias in Mosaic Monte Carlo under the aggregation summarizing strategy require modified formulas which were provided in Equations 2 and 3. As a reminder, the formula for relative bias is given by

$$\frac{1}{J}\sum_{j=1}^{J}\frac{1}{K}\sum_{k=1}^{K}\frac{\widehat{\theta}_{jk}-\theta_j}{\theta_j}$$

where k corresponds to a given replication, j denotes a given data-generating model, and θ is the population parameter corresponding to the estimator $\widehat{\theta}$. Because the setting of the simulation study will yield theoretically unbiased results, we evaluated the performance of Mosaic Monte Carlo by examining the average and maximum *absolute* relative bias across the 500 test runs. This differs from how Mosaic Monte Carlo would be implemented in practice but allows us to better investigate deviations in performance.

Variance To examine the impact of Mosaic Monte Carlo via the aggregation summarizing strategy on estimates of variance, we compared the empirical variances of the regression coefficients $\widehat{\boldsymbol{\beta}}$ to their true variances, where the true variance-covariance matrix is given by

$$VAR(\widehat{\boldsymbol{\beta}}) = \sigma^2(X^TX)^{-1} \tag{5}$$

where X is the $n \times p+1$ design matrix (consisting of the dataset and an additional column of 1's) and σ^2 is the true error variance (which is fixed in the data generation process). The empirical variance was computed by calculating the variance of the regression coefficient estimates across all replications and then averaging across the data-generating models, as shown previously in Equation 4.

Monte Carlo standard errors Monte Carlo standard errors (MCSEs) were calculated for the bias and variance estimates of regression coefficients to assess loss in precision. For bias, the MCSE was calculated for each data-generating model separately by calculating the standard deviation of the bias estimates and dividing by the square root of the number of replications according to the guidelines in Koehler et al. (2009). Bootstrap estimates of MCSEs were obtained for the variance of regression coefficients by drawing 200 bootstrap samples of the regression coefficient estimates for a given data-generating model, calculating their variances, and then taking the standard deviations of the 200 bootstrap values (Koehler et al., 2009).

5 Screening phase

As previously discussed, aggregation is only a viable summarizing strategy when it is done across mosaic pieces which impact results nonsystematically. Thus, we first sought to verify that our candidate mosaic pieces met this criteria by conducting the screening phase of Mosaic Monte Carlo before running our main simulation study. Our main simulation study uses a design with J and K as full simulation factors so we will omit parts of step 3 in the screening phase that refine these values for a later simulation phase.

In the screening phase, our candidate mosaic pieces were the values of regression coefficients and predictor correlations. We used an initial $J = 50$ to ensure adequate sample size for conducting multiple regression analyses and an initial K of 200. We adaptively increased K for data-generating models whose MCSEs exceeded 20% of the corresponding true variance values. Thus, the MCSE screening thresholds were customized for each data-generating model to reflect true differences in underlying variability. Because regression coefficient estimates are unbiased in this setting, we used the SD instead of CV for calculations involving relative bias as a performance measure. To reduce computational burden, we conducted the screening phase with a fixed $n = 50$ and error variance of 100.

After adaptively increasing K to 500, all data-generating models obtained MCSEs within the desired screening thresholds. The SD for the bias of estimated regression coefficients are all less than the 0.1 threshold for CV values according to the first column of results in Table 5. To examine the impact of the mosaic pieces on variance estimates, we used the ratio of empirical variance to true variance as a performance measure. This quantity is more comparable across data-generating models with differing true variances. However, the CVs for the ratio of empirical to true variances of regression coefficients are just over the 0.1 threshold in Table 5. This suggests that population values of coefficients and correlations do not impact bias of regression coefficient estimates but perhaps could impact their variance estimates. Therefore, we conducted follow-up screening to determine whether these are systematic vs. nonsystematic effects. We used multiple regression with population coefficient and correlation values as predictors and the bias or variance estimates of each regression coefficient. The values of R^2_{adj} in Table 5 were all less than 0.1, suggesting that the population values of regression coefficients and correlations have nonsystematic impacts on estimates of bias and variance. This implies that aggregating the results of the simulation study is defensible across regression coefficient and correlation mosaic pieces.

Table 5: SD, CV, and R^2_{adj} values for the screening phase of the simulation study examining population coefficient and correlation values as mosaic pieces on the bias and variance of regression coefficient estimates

	Bias		Var Ratio	
Coef	SD	R^2_{adj}	CV	R^2_{adj}
$\widehat{\beta}_0$	0.0735	0.0182	0.1019	-0.0932
$\widehat{\beta}_1$	0.0640	-0.0046	0.1136	0.0084
$\widehat{\beta}_2$	0.0464	-0.0888	0.1003	-0.0086
$\widehat{\beta}_3$	0.0483	-0.0191	0.1076	-0.0448

6 Results

The results of our simulation study are presented in this section which is organized as follows. We begin with results that show the relative bias of regression coefficients with their MCSEs. Then, we present results regarding the variance of the regression coefficients and their MCSEs. Recall that the regression coefficients $\widehat{\beta}_1$, $\widehat{\beta}_2$, and $\widehat{\beta}_3$ are exchangeable. In most of our tables and figures, the results for $\widehat{\beta}_2$ and $\widehat{\beta}_3$ are omitted for brevity. These results can be found in our supplemental materials.

6.1 Bias

The aggregated absolute relative bias was calculated according to Equation 3 for each regression coefficient across the 500 test runs from different random seeds. The results for $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are shown in Table 6. The results for $\widehat{\beta}_2$ and $\widehat{\beta}_3$ are comparable to $\widehat{\beta}_1$ and available in our supplemental material.

When there is only one randomized data-generating model with 1000 replications (i.e., under the control condition), the average absolute relative bias for the regression coefficients is minimal. This is true regardless of sample size, with the largest bias reaching 1.5% in a few conditions where the residual variance was 100. For the conditions with larger numbers of data-generating models and fewer replications, the average relative bias also tends to be small. However, in extreme cases when the sample size is 50 or 75 and the residual variance is 100, there are a few conditions where the absolute relative bias can be as large as 2 or 3%. As the number of data-generating models increases and the number of replications decreases, the absolute relative bias also tends to increase in small amounts. For example, when the sample size is 50 and the error variance is 25, the absolute relative bias of $\widehat{\beta_0}$ for 10 data-generating models with 100 replications each is 0.0096 while the absolute relative bias for 1000 data-generating models with 1 replication each has increased to 0.0109. Compared to the traditional Monte Carlo condition (with an absolute relative bias of 0.0067), the absolute relative bias of $\widehat{\beta}_0$ increases by approximately 0.003 by including 10 data-generating models and increases by approximately 0.004 by including 1000

Table 6: Aggregated absolute relative bias of $\widehat{\beta}_0$ and $\widehat{\beta}_1$ across all 500 test runs.

		Randomized Population Models / Replications							
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10	1000/1
					$\widehat{\beta}_0$				
50	9	0.0041	0.0055	0.0061	0.0058	0.0070	0.0066	0.0061	0.0064
	25	0.0067	0.0096	0.0103	0.0100	0.0108	0.0113	0.0120	0.0109
	100	0.0141	0.0194	0.0213	0.0208	0.0229	0.0229	0.0225	0.0231
75	9	0.0035	0.0043	0.0053	0.0051	0.0051	0.0056	0.0060	0.0051
	25	0.0049	0.0081	0.0082	0.0087	0.0088	0.0090	0.0100	0.0092
	100	0.0101	0.0135	0.0163	0.0174	0.0196	0.0184	0.0187	0.0179
100	9	0.0029	0.0039	0.0041	0.0043	0.0044	0.0045	0.0051	0.0052
	25	0.0046	0.0063	0.0074	0.0072	0.0074	0.0074	0.0081	0.0089
	100	0.0087	0.0136	0.0149	0.0153	0.0160	0.0150	0.0159	0.0172
200	9	0.0020	0.0029	0.0033	0.0033	0.0035	0.0033	0.0033	0.0041
	25	0.0035	0.0048	0.0052	0.0050	0.0055	0.0056	0.0053	0.0063
	100	0.0066	0.0097	0.0102	0.0102	0.0112	0.0114	0.0111	0.0129
400	9	0.0013	0.0019	0.0022	0.0023	0.0022	0.0022	0.0024	0.0025
	25	0.0022	0.0034	0.0033	0.0035	0.0036	0.0036	0.0040	0.0039
	100	0.0042	0.0067	0.0072	0.0075	0.0071	0.0076	0.0078	0.0081
					$\widehat{\beta}_1$				
50	9	0.0045	0.0071	0.0078	0.0081	0.0082	0.0083	0.0080	0.0092
	25	0.0071	0.0118	0.0136	0.0142	0.0134	0.0134	0.0135	0.0151
	100	0.0152	0.0213	0.0276	0.0246	0.0263	0.0262	0.0263	0.0313
75	9	0.0040	0.0054	0.0060	0.0060	0.0063	0.0065	0.0067	0.0062
	25	0.0065	0.0098	0.0106	0.0107	0.0108	0.0103	0.0116	0.0110
	100	0.0140	0.0186	0.0217	0.0209	0.0217	0.0217	0.0243	0.0219
100	9	0.0037	0.0047	0.0051	0.0055	0.0059	0.0053	0.0056	0.0060
	25	0.0062	0.0081	0.0083	0.0094	0.0093	0.0088	0.0091	0.0098
	100	0.0126	0.0173	0.0177	0.0195	0.0187	0.0184	0.0175	0.0198
200	9	0.0025	0.0035	0.0037	0.0038	0.0041	0.0043	0.0039	0.0039
	25	0.0044	0.0060	0.0057	0.0064	0.0071	0.0068	0.0067	0.0066
	100	0.0076	0.0119	0.0117	0.0135	0.0143	0.0138	0.0130	0.0135
400	9	0.0019	0.0025	0.0027	0.0029	0.0028	0.0028	0.0028	0.0034
	25	0.0035	0.0042	0.0046	0.0053	0.0050	0.0048	0.0047	0.0057
	100	0.0055	0.0076	0.0097	0.0095	0.0090	0.0090	0.0093	0.0111

data-generating models. The amount of this increase is also influenced by sampling variability (i.e., residual variance) and sample size, with smaller sample sizes and larger sampling variability leading to larger increases in absolute relative bias as the number of data-generating models increase and the number of replications decrease.

In a worst-case-scenario analysis, we also calculated the maximum absolute relative bias in each condition across all 500 test runs. The results in Table 7

show an interesting pattern – in most conditions, traditional Monte Carlo (1 data-generating model with 1000 replications) has the largest maximum biases compared to Mosaic Monte Carlo designs with larger numbers of data-generating models. In conditions where traditional Monte Carlo does *not* have the largest maximum bias, the largest values tend to occur with smaller numbers of data-generating models (such as with $n = 50$ and $\sigma_\varepsilon^2 = 25$ when the largest maximum bias occurs with 20 data-generating models for $\widehat{\beta}_1$). The maximum biases tend to become smaller with greater numbers of data-generating models, although it is not a uniform pattern. This is a result of sampling variability. Occasionally, randomly generated datasets corresponding to a particular data-generating model may over or under represent the extremes of a distribution and lead to inaccurate parameter estimates. Mosaic Monte Carlo designs dilute the impact of these extreme occurrences, whereas traditional Monte Carlo designs offer no protection against this scenario.

To illustrate these differences in sampling variability across the competing Monte Carlo designs, the distribution of the aggregated bias of $\widehat{\beta}_1$ across designs is shown in the boxplots in Figure 3 when $n = 50$ or 100. Each boxplot is constructed from 500 points which correspond to the aggregated bias of $\widehat{\beta}_1$ calculated from a single test run.

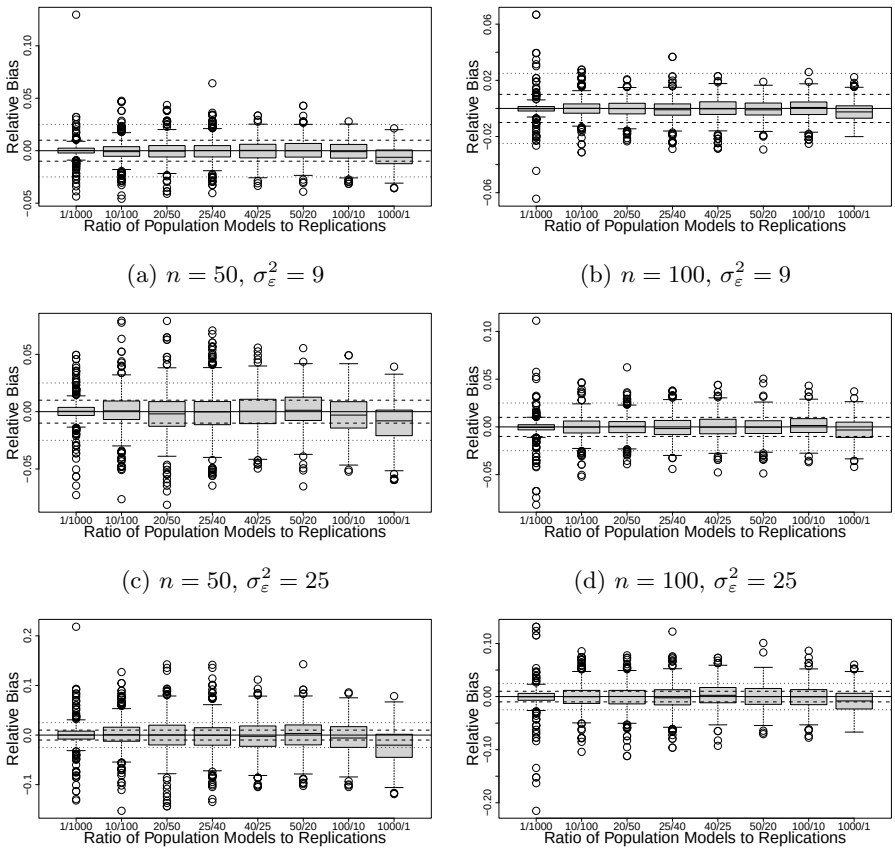
These boxplots demonstrate that across sample sizes and conditions of residual variance, Mosaic Monte Carlo designs lead to smaller ranges of values for the relative bias of $\widehat{\beta}_1$ across the 500 test runs. Larger sample sizes and smaller values of residual variance lead to greater precision. Traditional Monte Carlo designs with 1 data-generating model and 1000 replications have the smallest interquartile range compared to the Mosaic Monte Carlo designs, but also contain the most and largest outliers and therefore the largest overall ranges. As a reminder, implementing either approach in practice would be comparable to a single test run corresponding to a single point in the boxplots of Figure 3.

Figure 4 contains boxplots of the distribution of relative bias for $\widehat{\beta}_1$ at larger sample sizes. The trends in these boxplots are similar to the ones in Figure 3: Mosaic Monte Carlo has smaller ranges of relative bias than traditional Monte Carlo designs but has larger interquartile ranges. Greater precision can be obtained with larger sample sizes and smaller values of residual variance. It is worth noting that the number of outliers in the boxplots are a stable feature that depends on the number of data-generating models and are not affected by conditions of sample size or residual variance. Thus, a researcher could not build in protection against possible outliers by adjusting these features. Figures 3 and 4 together illustrate that the probability of inaccuracy is higher for Mosaic Monte Carlo designs but that the magnitude of these inaccuracies can be much less than under traditional Monte Carlo.

Since MCSEs are specific to data-generating models, we present the MCSEs for the relative bias of the regression coefficients in two ways. First, we present the aggregated average MCSEs across all 500 test runs in Table 8. This quantifies the Monte Carlo error that is generally present across the data-generating models in a given condition. Then, we present the maximum of the maximum MCSEs

Table 7: Maximum absolute relative bias of $\widehat{\beta}_0$ and $\widehat{\beta}_1$ across all 500 test runs.

		Randomized Population Models / Replications							
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10	1000/1
					$\widehat{\beta}_0$				
50	9	0.0662	0.0389	0.0408	0.0379	0.0538	0.0294	0.0234	0.0236
	25	0.0947	0.0648	0.0652	0.0632	0.0576	0.0501	0.0499	0.0337
	100	0.1893	0.1297	0.1304	0.1265	0.1794	0.1301	0.0934	0.0786
75	9	0.0392	0.0343	0.0374	0.0296	0.0342	0.0296	0.0265	0.0171
	25	0.0570	0.0478	0.0623	0.0502	0.0570	0.0355	0.0441	0.0432
	100	0.1305	0.0957	0.0933	0.0687	0.1012	0.0987	0.0882	0.0704
100	9	0.0409	0.0285	0.0225	0.0282	0.0220	0.0193	0.0222	0.0182
	25	0.0647	0.0395	0.0381	0.0384	0.0368	0.0326	0.0371	0.0310
	100	0.1293	0.1000	0.0832	0.0768	0.0737	0.0632	0.0743	0.0620
200	9	0.0298	0.0174	0.0174	0.0178	0.0169	0.0170	0.0172	0.0165
	25	0.0382	0.0240	0.0254	0.0280	0.0282	0.0284	0.0287	0.0275
	100	0.1122	0.0451	0.0564	0.0559	0.0861	0.0524	0.0575	0.0551
400	9	0.0156	0.0164	0.0118	0.0153	0.0134	0.0097	0.0094	0.0088
	25	0.0241	0.0310	0.0197	0.0133	0.0223	0.0196	0.0220	0.0152
	100	0.0492	0.0620	0.0393	0.0353	0.0446	0.0393	0.0329	0.0311
					$\widehat{\beta}_1$				
50	9	0.1297	0.0476	0.0435	0.0643	0.0336	0.0428	0.0316	0.0357
	25	0.0728	0.0793	0.0812	0.0708	0.0558	0.0652	0.0523	0.0595
	100	0.2184	0.1527	0.1436	0.1416	0.1117	0.1428	0.1047	0.1190
75	9	0.0816	0.0455	0.0380	0.0349	0.0415	0.0288	0.0283	0.0270
	25	0.1126	0.0788	0.0460	0.0517	0.0692	0.0454	0.0472	0.0486
	100	0.3324	0.1103	0.1011	0.1165	0.1507	0.1167	0.0868	0.0972
100	9	0.0667	0.0313	0.0234	0.0367	0.0286	0.0293	0.0259	0.0223
	25	0.1112	0.0521	0.0624	0.0441	0.0476	0.0505	0.0432	0.0419
	100	0.2152	0.1042	0.1120	0.1223	0.0929	0.1010	0.0863	0.0669
200	9	0.0302	0.0178	0.0188	0.0179	0.0253	0.0151	0.0184	0.0154
	25	0.0478	0.0432	0.0330	0.0556	0.0298	0.0291	0.0307	0.0214
	100	0.1006	0.0764	0.0626	0.0556	0.0843	0.0706	0.0575	0.0512
400	9	0.0538	0.0163	0.0121	0.0152	0.0140	0.0133	0.0107	0.0110
	25	0.0896	0.0271	0.0344	0.0273	0.0231	0.0241	0.0198	0.0184
	100	0.1392	0.0683	0.0688	0.0521	0.0468	0.0360	0.0462	0.0439



(a) $n = 50$, $\sigma_\varepsilon^2 = 9$

(b) $n = 100$, $\sigma_\varepsilon^2 = 9$

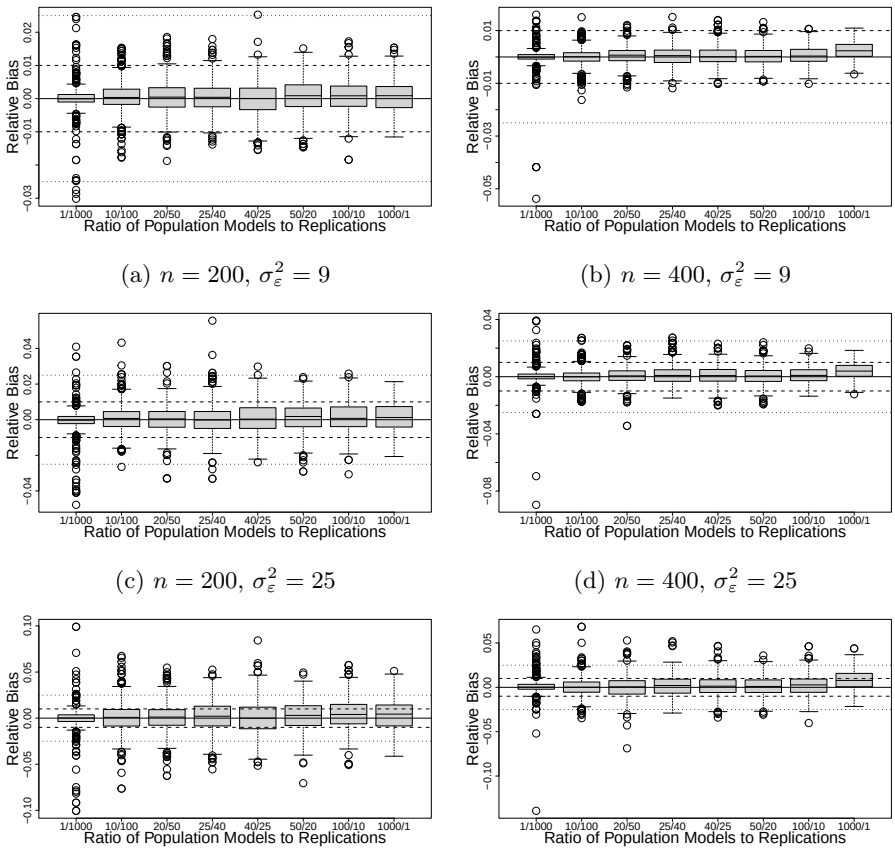
(c) $n = 50$, $\sigma_\varepsilon^2 = 25$

(d) $n = 100$, $\sigma_\varepsilon^2 = 25$

(e) $n = 50$, $\sigma_\varepsilon^2 = 100$

(f) $n = 100$, $\sigma_\varepsilon^2 = 100$

Figure 3: Distribution of aggregated bias of $\widehat{\beta}_1$ across 500 test runs for $n = 50$ and $n = 100$ across conditions of residual variance. Solid line drawn at 0, dashed and dotted lines denote relative bias of 1% and 2.5% respectively.



(a) $n = 200$, $\sigma_\varepsilon^2 = 9$

(b) $n = 400$, $\sigma_\varepsilon^2 = 9$

(c) $n = 200$, $\sigma_\varepsilon^2 = 25$

(d) $n = 400$, $\sigma_\varepsilon^2 = 25$

(e) $n = 200$, $\sigma_\varepsilon^2 = 100$

(f) $n = 400$, $\sigma_\varepsilon^2 = 100$

Figure 4: Distribution of aggregated bias of $\widehat{\beta}_1$ across 500 test runs for $n = 200$ and $n = 400$ across conditions of residual variance. Solid line drawn at 0, dashed and dotted lines denote relative bias of 1% and 2.5%, respectively.

across data-generating models and all 500 test runs in Table 9. That is, the maximum MCSE for a specific data-generating model is recorded and then the maximum of those values across the 500 test runs are shown in the table. This quantifies the maximum Monte Carlo error that is present in a given condition. As a note, the fully random condition with 1000 data-generating models and 1 replication each are not shown because the MCSEs cannot be calculated in conditions with only 1 replication.

Table 8 shows that as the number of data-generating models increase (and the number of replications decrease), there is an increase in the average MCSE values. This is expected since the calculating of MCSEs involves dividing by the square root of the number of replications. In fact, there is about a ten-fold increase in the MCSE from conditions with a single data-generating model to conditions with 100 data-generating models. This means that the differences are mostly in the denominator and that standard deviations of relative bias of the regression coefficients are generally stable across different Mosaic Monte Carlo designs and traditional Monte Carlo. Also as expected, MCSEs tend decrease with larger sample sizes and increase with larger values of residual variance. Table 9 shows that the maximum MCSEs across Mosaic Monte Carlo designs can be quite different from each other. There is larger than a ten-fold increase between conditions with 1 data-generating model and conditions with 100 data-generating models. This happens because nothing in the table has been aggregated and so extreme occurrences will not be diluted by the Mosaic Monte Carlo procedure. For example, there are 50,000 MCSEs in conditions with 100 data-generating models but only 500 MCSEs in conditions with 1 data-generating model. The extremes of distributions will be better captured when there are 50,000 observations compared to 500. Like Table 8, Table 9 also shows that the maximum MCSEs of the relative bias of regression coefficients decreases with increasing sample size and increases with increasing residual variance.

6.2 Variance

Table 10 contains the aggregated ratios of empirical variance estimates to their true variances for $\widehat{\beta}_0$ and $\widehat{\beta}_1$. Values greater than 1 indicate that the empirical variance is larger than the true variance whereas values less than 1 indicate that the empirical variance is smaller than the true variance. Note that the condition with 1000 data-generating models with 1 replication each is omitted from the table; this is because the empirical variance cannot be calculated with only 1 replication.

There are a few trends worth noting in this table. First, the differences between the average empirical and true variances are generally to the third decimal place or less. Most of the exceptions occur regarding $\widehat{\beta}_1$ when the number of data-generating models is 100 and the sample size is 100 or less, when the differences can be up to the second decimal place. The biggest discrepancy occurs for 100 data-generating models with a sample size of 100 and $\sigma_\varepsilon^2 = 100$, when the ratio of empirical to true variance is 0.9798. Second, all conditions varying the number of data-generating models (i.e., both traditional and Mosaic Monte Carlo designs)

Table 8: Aggregated average MCSE of bias of $\widehat{\beta}_0$ and $\widehat{\beta}_1$ across 500 test runs.

		Randomized Population Models / Replications						
n	σ^2_ε	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
					$\widehat{\beta}_0$			
50	9	0.0051	0.0159	0.0225	0.0253	0.0320	0.0359	0.0505
	25	0.0086	0.0274	0.0379	0.0424	0.0536	0.0599	0.0846
	100	0.0174	0.0540	0.0762	0.0850	0.1067	0.1200	0.1687
75	9	0.0044	0.0132	0.0188	0.0210	0.0264	0.0296	0.0408
	25	0.0066	0.0219	0.0306	0.0341	0.0435	0.0493	0.0685
	100	0.0132	0.0440	0.0616	0.0694	0.0877	0.0991	0.1367
100	9	0.0035	0.0113	0.0158	0.0176	0.0223	0.0251	0.0350
	25	0.0056	0.0190	0.0266	0.0299	0.0380	0.0425	0.0593
	100	0.0129	0.0394	0.0541	0.0604	0.0760	0.0855	0.1171
200	9	0.0025	0.0080	0.0113	0.0127	0.0161	0.0181	0.0249
	25	0.0042	0.0133	0.0189	0.0209	0.0265	0.0299	0.0413
	100	0.0086	0.0267	0.0373	0.0419	0.0527	0.0597	0.0835
400	9	0.0018	0.0056	0.0078	0.0088	0.0111	0.0125	0.0175
	25	0.0029	0.0094	0.0130	0.0145	0.0184	0.0208	0.0293
	100	0.0055	0.0183	0.0258	0.0291	0.0369	0.0415	0.0583
					$\widehat{\beta}_1$			
50	9	0.0061	0.0199	0.0278	0.0315	0.0398	0.0443	0.0611
	25	0.0109	0.0331	0.0471	0.0527	0.0663	0.0738	0.1033
	100	0.0188	0.0644	0.0924	0.1029	0.1299	0.1460	0.2036
75	9	0.0052	0.0155	0.0220	0.0246	0.0313	0.0351	0.0490
	25	0.0087	0.0270	0.0371	0.0416	0.0524	0.0583	0.0819
	100	0.0172	0.0511	0.0737	0.0833	0.1041	0.1160	0.1633
100	9	0.0044	0.0134	0.0190	0.0212	0.0269	0.0297	0.0414
	25	0.0070	0.0228	0.0316	0.0353	0.0451	0.0504	0.0692
	100	0.0150	0.0465	0.0641	0.0721	0.0893	0.0993	0.1391
200	9	0.0033	0.0095	0.0133	0.0149	0.0190	0.0213	0.0294
	25	0.0055	0.0165	0.0225	0.0249	0.0316	0.0351	0.0495
	100	0.0098	0.0317	0.0447	0.0501	0.0633	0.0708	0.0979
400	9	0.0022	0.0066	0.0094	0.0105	0.0133	0.0148	0.0208
	25	0.0037	0.0115	0.0158	0.0179	0.0226	0.0250	0.0347
	100	0.0068	0.0223	0.0319	0.0358	0.0450	0.0499	0.0695

Table 9: Maximum of maximum MCSEs of bias of $\widehat{\beta}_0$ and $\widehat{\beta}_1$ across all 500 test runs.

		Randomized Population Models / Replications						
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
					$\widehat{\beta}_0$			
50	9	0.0505	0.2045	0.2954	0.3354	0.4978	0.5123	0.8322
	25	0.0789	0.3408	0.4565	0.5584	0.7244	0.8365	1.3871
	100	0.1684	0.6034	0.9848	1.1168	1.5071	1.7078	2.7722
75	9	0.0420	0.1389	0.2367	0.2689	0.3762	0.4046	0.6476
	25	0.0701	0.2506	0.3945	0.4321	0.5786	0.7500	1.1201
	100	0.1399	0.5012	0.7890	0.8325	1.2542	1.5000	2.1742
100	9	0.0364	0.1329	0.2045	0.2208	0.2937	0.3617	0.6033
	25	0.0603	0.2094	0.3409	0.3710	0.5176	0.6028	0.9394
	100	0.1142	0.4188	0.6393	0.7718	0.9869	1.2551	1.8788
200	9	0.0251	0.0869	0.1366	0.1574	0.2386	0.2563	0.3871
	25	0.0430	0.1656	0.2239	0.2645	0.3976	0.4271	0.6511
	100	0.0859	0.3311	0.4779	0.5290	0.7522	0.8252	1.4023
400	9	0.0184	0.0628	0.1018	0.1251	0.1576	0.1711	0.2824
	25	0.0306	0.1033	0.1697	0.2085	0.2610	0.2862	0.4684
	100	0.0615	0.2094	0.2881	0.4170	0.5255	0.5725	1.0187
					$\widehat{\beta}_1$			
50	9	0.0688	0.2709	0.3977	0.5046	0.5576	0.8135	1.1824
	25	0.1208	0.4515	0.6629	0.7519	1.0611	1.1554	2.0557
	100	0.2464	0.7958	1.3258	1.6821	1.8587	2.7117	3.7231
75	9	0.0558	0.2114	0.2818	0.4518	0.5543	0.5210	1.2056
	25	0.0934	0.3159	0.4696	0.6336	0.9239	0.9034	1.6431
	100	0.1869	0.5971	1.0591	1.5059	1.4457	1.7259	3.3885
100	9	0.0504	0.1743	0.2390	0.3228	0.3728	0.4316	0.7433
	25	0.0769	0.2753	0.3890	0.4862	0.5984	0.6875	1.2281
	100	0.1568	0.5660	0.7870	1.0759	1.1961	1.5578	2.4543
200	9	0.0348	0.1120	0.1651	0.2031	0.2626	0.3157	0.5922
	25	0.0510	0.1992	0.2889	0.3277	0.4604	0.4792	1.0325
	100	0.1160	0.3715	0.5454	0.6447	0.9208	1.0715	2.0650
400	9	0.0219	0.0754	0.1144	0.1251	0.1870	0.2256	0.3359
	25	0.0422	0.1361	0.1888	0.2472	0.3002	0.3884	0.5693
	100	0.0844	0.2723	0.3679	0.5049	0.6233	0.7050	1.0968

Table 10: Aggregated ratio of empirical to true variance of $\widehat{\beta}_0$ and $\widehat{\beta}_1$ across all 500 test runs.

		Randomized Population Models / Replications						
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
					$\widehat{\beta}_0$			
50	9	0.9978	0.9955	0.9974	1.0000	1.0054	1.0032	1.0043
	25	0.9989	0.9989	0.9987	0.9996	1.0020	1.0021	1.0098
	100	0.9974	0.9994	1.0002	1.0002	1.0026	1.0010	1.0050
75	9	1.0007	0.9946	0.9997	0.9978	1.0029	1.0015	0.9955
	25	1.0024	0.9933	0.9974	1.0011	1.0040	1.0036	0.9941
	100	0.9995	0.9997	0.9987	0.9993	1.0036	1.0063	0.9951
100	9	0.9995	0.9980	0.9974	0.9977	0.9974	1.0007	0.9909
	25	0.9960	0.9955	0.9965	0.9966	0.9945	0.9955	0.9923
	100	0.9990	0.9965	0.9961	0.9983	0.9971	0.9989	0.9912
200	9	1.0020	0.9969	0.9952	0.9948	0.9912	0.9958	0.9924
	25	0.9976	0.9989	0.9997	0.9951	0.9913	0.9974	0.9912
	100	1.0027	0.9976	0.9992	0.9952	0.9902	0.9931	0.9942
400	9	0.9984	0.9991	0.9986	0.9976	0.9933	0.9904	0.9910
	25	0.9985	1.0003	0.9982	0.9975	0.9910	0.9930	0.9895
	100	0.9972	0.9993	0.9921	0.9960	0.9898	0.9912	0.9943
					$\widehat{\beta}_1$			
50	9	0.9993	1.0079	1.0026	1.0044	1.0071	1.0143	1.0138
	25	1.0013	1.0057	1.0061	1.0035	1.0068	1.0101	1.0154
	100	0.9957	1.0058	1.0033	1.0005	1.0046	1.0119	1.0138
75	9	0.9982	0.9985	1.0015	1.0009	0.9993	1.0004	1.0121
	25	0.9978	0.9989	1.0057	1.0062	0.9976	0.9963	1.0129
	100	0.9986	1.0023	1.0045	1.0035	0.9963	0.9971	1.0117
100	9	1.0006	1.0037	0.9958	0.9974	0.9994	0.9940	0.9802
	25	0.9986	1.0045	0.9964	0.9962	0.9956	0.9955	0.9818
	100	1.0005	1.0033	1.0001	0.9984	0.9985	0.9947	0.9798
200	9	1.0082	0.9968	0.9954	0.9922	0.9930	1.0007	1.0026
	25	1.0055	0.9941	0.9979	0.9978	0.9946	1.0044	1.0059
	100	1.0019	0.9939	0.9971	0.9970	0.9950	1.0021	0.9996
400	9	0.9968	0.9973	0.9960	0.9947	0.9956	0.9975	1.0003
	25	1.0042	1.0031	0.9949	0.9934	0.9948	0.9987	1.0006
	100	1.0008	1.0006	0.9972	0.9977	1.0006	1.0007	0.9989

tend to underestimate the variance of $\widehat{\beta}_0$ while overestimating the variance of $\widehat{\beta}_1$. Another interesting finding from this table is that there does not appear to be a systematic relationship between the number of data-generating models and the ratios of empirical to true variances. For example, the ratio of empirical to true variance of $\widehat{\beta}_1$ increases from 0.9978 under traditional Monte Carlo (i.e., 1 data-generating model with 1000 replications) to 1.0062 when the number of data-generating models is increased to 25 for $n = 75$ and $\sigma_\varepsilon^2 = 25$. However, the ratio of empirical to true variance then decreases to 0.9963 when the number of data-generating models is increased to 50 under the same set of conditions. The values in Table 10 show that Mosaic Monte Carlo does not lead to systematic or substantial increases or decreases in variance estimates compared to traditional Monte Carlo designs.

Table 11: Average and maximum MCSEs of variance of regression coefficients across 500 test runs for $n = 50$ and $\sigma_\varepsilon^2 = 100$.

		Randomized Population Models / Replications						
Coefficient	MCSE	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
$\widehat{\beta}_0$	Average	0.0445	0.1384	0.1921	0.2146	0.2570	0.2792	0.3612
	Maximum	0.0543	0.1567	0.2127	0.2334	0.2898	0.3047	0.3841
$\widehat{\beta}_1$	Average	0.0449	0.1383	0.1910	0.2141	0.2579	0.2787	0.3634
	Maximum	0.0561	0.1575	0.2103	0.2402	0.2818	0.3011	0.3874

Table 11 shows the MCSEs of the variance estimates of $\widehat{\beta}_0$ and $\widehat{\beta}_1$. Unlike the MCSEs of their biases, there is very little change in MCSE values across conditions of sample size or residual variance and so results are shown only for the condition with $n = 50$ and $\sigma_\varepsilon^2 = 100$ for the sake of brevity. The aggregated average MCSE of variance estimates in Table 11 show variability across the number of data-generating models that is not explained by the number of replications; there is less than a ten-fold increase in MCSE values as the number of data-generating models is increased from 1 to 100. Interestingly, this suggests that there is actually *more* variation in variance estimates due to Monte Carlo error in traditional Monte Carlo designs with 1 data-generating model compared to Mosaic designs involving several data-generating models. This pattern is present for both regression coefficients and for both the aggregated average MCSEs and the maximum of the maximum MCSEs.

Since researchers would only run Mosaic Monte Carlo designs a single time in implementation, we also illustrate the differences between empirical and true variance in a single test run to show what results might look like in practice. We will show results for $\widehat{\beta}_0$ and $\widehat{\beta}_1$ in conditions with $n = 100$ and $\sigma_\varepsilon^2 = 100$, as the patterns are the same for the other conditions of sample size and residual variance.

Figure 5 shows the empirical variances of $\hat{\beta}_0$ for the first test run. The top plot shows the condition for 10 randomized data-generating models with 100

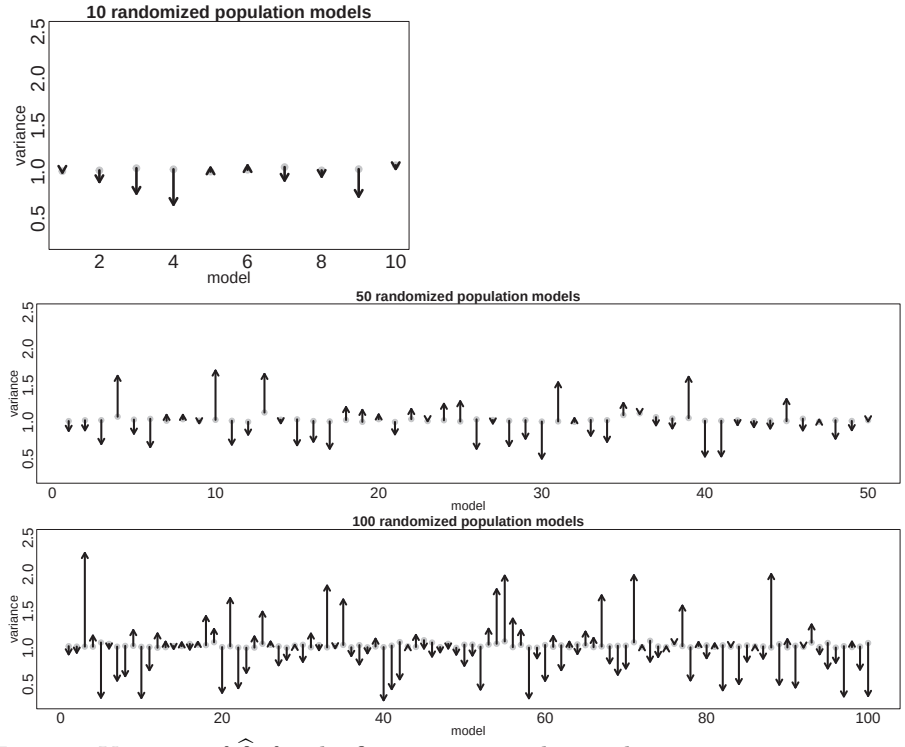


Figure 5: Variance of $\widehat{\beta}_0$ for the first test run with sample size 100, error variance = 100. The top plot corresponds to the condition with 10 randomized population models and 100 replications while the middle plot corresponds to the condition with 50 randomized population models and 20 replications. The bottom plot corresponds to the condition with 100 randomized population models and 10 replications. Gray dots denote the population variance of $\widehat{\beta}_0$ and the arrows denote the empirical variance of $\widehat{\beta}_0$.

replications each. As we might expect, the differences between the true and empirical variance is very small, since the empirical variance is calculated with 100 replications in the denominator. The differences are less than 1% and the directions of the arrows do not suggest a systematic tendency to either underestimate or overestimate the population variance. The middle plot shows larger differences between the true variance and the empirical variance of the 50 data-generating models, although there are several models whose empirical variance estimates are quite close to the true variance. This pattern is intuitive since the empirical variance is calculated using only 20 replications. We can see an even greater degree of differences between the true variance and empirical variance in the bottom plot, which is the condition for 100 data-generating models with 10 replications each. From Figure 5, it is clear that the empirical variance will have greater differences from the true variance as the number of data-generating models increase. However, if the results are analyzed all together and averaged to form the mosaic, we will see small differences between the average empirical variance and the true variance as in Table 10.

Figure 6 shows the difference between empirical variances and the true variance for the regression coefficient $\widehat{\beta}_1$. Recall that there is no meaningful difference between the results for $\widehat{\beta}_1$ and the other two regression coefficients. The true variance fluctuates between data-generating models more than what was seen for the true variance of $\widehat{\beta}_0$. As a reminder, this fluctuation occurs because we held X fixed across replications but allowed it to vary across data-generating models. However, the general trends from this plot are similar to what was seen in Figure 5. That is, the differences between the empirical variances and the true variances fluctuate more widely as the number of data-generating models increase.

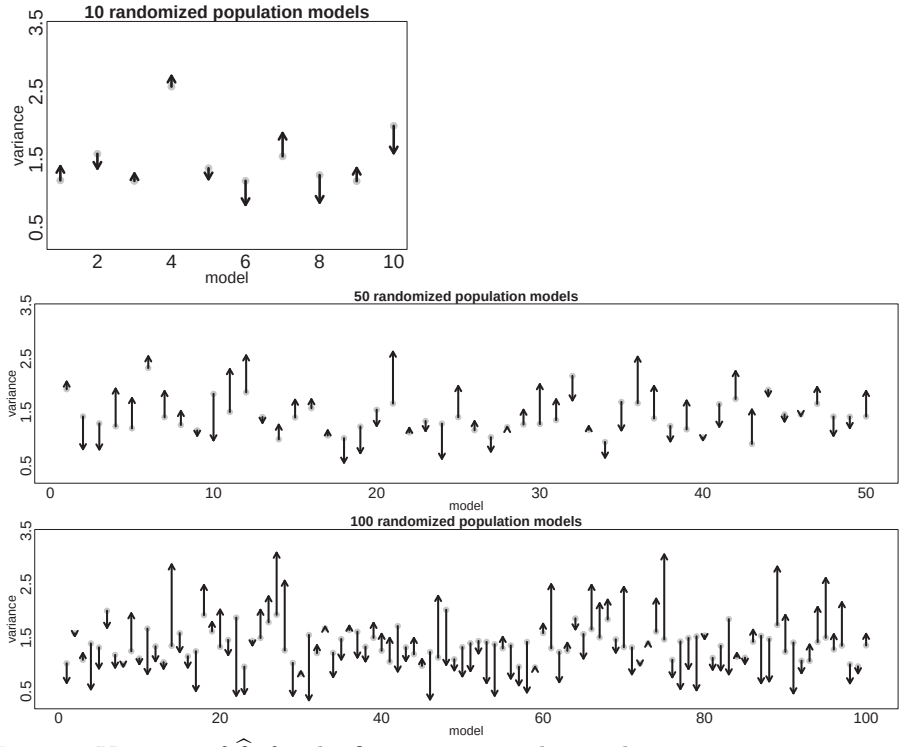


Figure 6: Variance of $\widehat{\beta}_1$ for the first test run with sample size 100, error variance = 100. The top plot corresponds to the condition with 10 randomized population models and 100 replications while the middle plot corresponds to the condition with 50 randomized population models and 20 replications. The bottom plot corresponds to the condition with 100 randomized population models and 10 replications. Gray dots denote the population variance of $\widehat{\beta}_1$ and the arrows denote the empirical variance of $\widehat{\beta}_1$.

7 Empirical Example

To demonstrate the performance of Mosaic Monte Carlo in a "real world context", we used our method to redo part of the simulation study from Gomer et al. (2019). This article developed and compared several effect size measures in the context of structural equation modeling. Our goal is to examine the performance of 6 of these effect size measures under the same conditions of the original simulation study but using Mosaic Monte Carlo with factor loadings and factor correlations as candidate mosaic pieces. We will replicate Tables 4 and 5 from the original article containing results from the hierarchical linear regression and distance of the effect size measures from F_0 (i.e., the population value of the discrepancy function). Since MCSEs are designed for estimators while the original work studied the population values of effect size measures, we will not compute MCSEs in this example. Due to this and and heavy computational burden in the simulation design, we set the number of data-generating models J to be 100 and the number of replications K to be 50. These values are the maximum J and K that could be afforded that would give a large enough sample size in the regression analyses in step 3 of the screening phase. The code for this empirical example is provided in a github repository.

7.1 Data generation

Data were generated according to a 3 factor measurement model with 9, 15, or 30 manifest variables. The factor correlation values were randomly generated from $Unif(0.1, 0.7)$ while the factor loadings were randomly generated from $Unif(0.4, 0.9)$, keeping the first loadings on each factor fixed to 1. These boundaries were chosen in order to obtain interpretable factor loadings according to the criteria in Stevens (1992). The population values for the factor variances and error variances were set to 1. The data distributions were normal, t, elliptical, non-normal, uniform, and mixture distributions according to the same specifications used in Gomer et al. (2019). The sample size was varied from 75, 100, 120, 150, 200, 300, 500, 800, to 1000. In conditions where the model was misspecified, a single cross-loading was omitted whose value ranged from 0.1 to 1 in increments of 0.1. The model was estimated via standard maximum likelihood (ML) or using robust M-estimation. The formulas for the effect sizes we considered are shown in Table 12.

7.2 Evaluation Criteria

Hierarchical linear regression To evaluate the performance of the effect size measures, the original article conducted a hierarchical linear regression to determine the impact of sample size, model size, distribution, and model misspecification on the value of the candidate effect size measures. The reduced model had a single predictor of model misspecification while the full model included predictors for sample size, model size, and distribution. Effect sizes with good performance were considered to be those with large values of R^2_{adj} and when the change in

Table 12: Definitions of the effect size measures in Gomer et al. (2019)

Effect Size	Formula	Assumptions in Construction
$\mathscr{E}_1$	$\frac{2\cdot[E(T_{ML}\|H_1)-E(T_{ML}\|H_0)]}{((N-1)\cdot(VAR(T_{ML}\|H_1)-2df))^{1/2}}$	$Var(T_{ML}\|H_1) = 2df + 4\lambda_n$
$\mathscr{E}_2$	$E\left[\left(\frac{max(T_{ML}-df,0)}{N-1}\right)^{1/2}\right]$	$E(T_{ML}\|H_0) = df$
$\mathscr{E}_3$	$\left(\frac{E(T_{ML}\|H_1)-E(T_{ML}\|H_0)}{N-1}\right)^{1/2}$	—
$\mathscr{E}_4$	$E\left[\left(\frac{T_{ML}-tr(\boldsymbol{\Gamma})}{N-1}\right)^{1/2}\right]$	$E(T_{ML}\|H_0) = tr(\boldsymbol{\Gamma})$
$\mathscr{E}_5$	$E\left[\left(\frac{T_{RML}-df}{N-1}\right)^{1/2}\right]$	$E(T_{ML}\|H_0) = df$
$\mathscr{E}_6$	$\left(\frac{E(T_{RML}\|H_1)-E(T_{RML}\|H_0)}{N-1}\right)^{1/2}$	—

Note. $\mathscr{E}_3$ and $\mathscr{E}_6$ do not make use of any assumptions in their formulas.

R^2_{adj} between the full and reduced models was small; in other words, the values of the effect size measure were mostly determined by the magnitude of model misspecification and not by the extraneous factors of sample size, model size, and distribution.

Distance to F_0 Gomer et al. (2019) also analyzed the performance of the candidate effect size measures by calculating the distance of each effect size measure from F_0, the population value of the discrepancy function. This is a measure of model misspecification. Effect size measures that were closest to F_0 were considered to have good performance.

7.3 Screening phase

Due to computational burden, we performed a limited version of the simulation study in the screening phase. We conducted the screening simulation for models with 9 manifest variables and only considered model misspecification sizes of 0, 0.5, and 1 with a sample size of $n = 200$. However, we retained all 8 data-generating distributions. When the size of model misspecification was 0, the effect size measures had target values close to 0 so the SD was used as criteria in place of the CV for this condition.

In step 2 of the screening phase, we obtained SD and CV values ranging from approximately 0 to 0.406 across conditions of model misspecification size, distribution, and effect size measure. This suggests that the candidate mosaic pieces (i.e., factor loadings and correlations) could impact the values of the effect size measures in some conditions. Thus, we conducted follow-up investigation in step 3 of the screening phase.

We performed multiple regression analyses with the values of effect size measures as outcome variables and the factor loadings and correlations as predictors.

The values of R^2_{adj} from these regressions for each effect size measure are shown in Table 13. As can be seen from the table, the values of R^2_{adj} are low for all effect size measures. This suggests that factor loadings and correlations have nonsystematic impacts on the values of effect size measures and also that aggregating over these mosaic pieces is defensible.

Table 13: $\Delta R^2_{adj.}$ values from the multiple regressions in the screening phase with factor loadings and correlations as candidate mosaic pieces

$\mathscr{E}_1$	$\mathscr{E}_2$	$\mathscr{E}_3$	$\mathscr{E}_4$	$\mathscr{E}_5$	$\mathscr{E}_6$
0.0206	0.0287	0.0195	0.0306	0.0282	0.0186

7.4 Results

The results of the hierarchical regression analysis when implementing the simulation study using Mosaic Monte Carlo are shown in Table 14. Rows with the smallest change in $R^2_{adj.}$ are bolded. Encouragingly, Mosaic Monte Carlo suggests that the three best effect size measures according to this criteria are $\mathscr{E}_1$, $\mathscr{E}_3$, and $\mathscr{E}_6$. This replicates the results of the original article. The specific values in the table also tend to have small differences from Table 4 in the original article. However, there is one exception: the $R^2_{adj.}$ values are quite small for $\mathscr{E}_1$. When looking at our results further, it appears that there were computational difficulties with this effect size measure because the formula involves the square root of a difference. When the difference is negative, the effect size cannot be calculated. This has a higher chance of occuring under Mosaic Monte Carlo methodology because of random variation in the data-generating process. However, it is impossible to know the characteristics of real datasets in advance so Mosaic Monte Carlo was able to capture a real difficulty of using $\mathscr{E}_1$ in practical applications.

Table 14: Hierarchical regression results using Mosaic Monte Carlo.

ES measure	$R^2_{adj.}(full)$	$R^2_{adj.}(reduced)$	$\Delta R^2_{adj.}$
$\mathscr{E}_1$	**0.6024**	**0.4760**	**0.1264**
$\mathscr{E}_2$	0.7678	0.2586	0.5093
$\mathscr{E}_3$	**0.9153**	**0.8492**	**0.0661**
$\mathscr{E}_4$	0.6649	0.2938	0.3711
$\mathscr{E}_5$	0.6999	0.2387	0.4612
$\mathscr{E}_6$	**0.8685**	**0.7121**	**0.1565**

Note. $R^2_{adj.}$ for the full and reduced models of each effect size measure. Bold rows correspond to the three effect size measures with the lowest $\Delta R^2_{adj.}$

To match the results presented in the original article, we re-created Table 5 showing the distances of $\mathscr{E}_1$, $\mathscr{E}_3$, and $\mathscr{E}_6$ from F_0. The results based on our Mosaic Monte Carlo simulation are shown in Table 15. $\mathscr{E}_3$ is closest to F_0 in almost every simulation condition, regardless of whether it is estimated with maximum likelihood or M-estimation. This mirrors the findings in the original article.

Table 15: Distance to F_0 for three effect size measures that were closest to F_0 in Gomer et al. (2019).

		Average Distance from F_0					
		$\mathscr{E}_1$		$\mathscr{E}_3$		$\mathscr{E}_6$	
p	Distribution	ML	$M\text{-}est$	ML	$M\text{-}est$	ML	$M\text{-}est$
9	Normal	0.0858	0.0823	0.0142	0.0144	0.0337	0.0549
	t	0.1147	0.0886	0.0401	0.0326	0.0604	0.1459
	Exponential	0.2307	0.0926	0.0269	0.0157	0.1109	0.1911
	Elliptical	0.0666	0.0692	0.0137	0.0142	0.0146	0.0148
	Non-normal	0.0972	0.0953	0.0409	0.0357	0.0430	0.0755
	Mix 1	0.0561	0.0542	0.0131	0.0125	0.0148	0.0916
	Mix 2	0.2316	0.0992	0.0283	0.0159	0.1120	0.1902
	Mix 3	0.2264	0.0752	0.0233	0.0128	0.0990	0.2317
15	Normal	0.1156	0.1148	0.0072	0.0074	0.0350	0.0671
	t	0.1851	0.1280	0.0595	0.0504	0.0824	0.1580
	Exponential	0.3312	0.1120	0.0227	0.0080	0.1443	0.2480
	Elliptical	0.0758	0.0718	0.0068	0.0067	0.0075	0.0072
	Non-normal	0.1484	0.1415	0.0606	0.0570	0.0620	0.0706
	Mix 1	0.1001	0.0928	0.0118	0.0100	0.0135	0.0925
	Mix 2	0.3329	0.1134	0.0229	0.0081	0.1443	0.2479
	Mix 3	0.3082	0.0907	0.0183	0.0108	0.1318	0.2863
30	Normal	0.1884	0.2151	0.0044	0.0039	0.0369	0.0806
	t	0.3286	0.2369	0.0974	0.0965	0.1264	0.1890
	Exponential	0.4923	0.2071	0.0134	0.0055	0.1660	0.3055
	Elliptical	0.2141	0.2121	0.0039	0.0039	0.0042	0.0041
	Non-normal	0.2570	0.2552	0.1006	0.1004	0.1012	0.1030
	Mix 1	0.1718	0.1369	0.0062	0.0064	0.0056	0.0587
	Mix 2	0.4929	0.2071	0.0139	0.0055	0.1664	0.3055
	Mix 3	0.4871	0.2159	0.0087	0.0280	0.1526	0.3323

Note. ML indicates that the model was fit via maximum likelihood and $M - est$ indicates that the model was fit via M-estimation. p indicates the number of manifest variables in the model. Mix 1, Mix 2, and Mix 3 distributions were mixtures of different distributions as defined by Gomer et al. (2019).

7.5 Discussion

The goal of replicating the simulation study from Gomer et al. (2019) using Mosaic Monte Carlo was to demonstrate that Mosaic Monte Carlo can be a viable simulation strategy and that it can improve the generalizability of findings. Not only was Mosaic Monte Carlo able to replicate the conclusion of the article and recommend $\mathscr{E}_3$ as the best candidate effect size measure, but it was also able to detect problematic features of $\mathscr{E}_1$ that failed to be captured originally.

8 Discussion and conclusion

The goal of this paper was to test an approach to Monte Carlo simulation design that breaks up large numbers of replications across factors that nonsystematically affect outcome(s) of interest. In the screening phase, researchers determine which factors have nonsystematic effects and appropriate numbers of data-generating models and replications. The simulation phase involves carrying out the main study of interest. We call this two-stage approach Mosaic Monte Carlo. This method can be used to improve the generalizability of findings from simulation studies compared to traditional designs that hold such factors as fixed. We showed how Mosaic Monte Carlo can be implemented in practice using an example from the literature on missing data. To test the baseline performance of Mosaic Monte Carlo, we examined how estimates of bias (RQ1) and variance (RQ2) were impacted by this approach when using the aggregation summarizing strategy and when the number of data-generating models is large compared to the number of replications. We conducted our study in the context of regression and also investigated how sample size and sampling variability might play a role in estimates of bias and variance (RQ3).

With respect to bias (RQ1), we found that Mosaic Monte Carlo generally leads to small increases in absolute relative bias for the regression coefficients which was also affected by the sample size and residual variance – small sample sizes with larger residual variances lead to increased absolute relative bias. However, traditional Monte Carlo designs had more outliers in performance and had the largest maximum values of absolute relative bias.

Regarding concerns about variance (RQ2), we found that Mosaic Monte Carlo did not systematically impact estimates of variance. Variance was affected mostly by sample size and by residual variance, and large differences between the true and empirical variance estimates occurred even in the control condition. Thus, we believe our method works well according to this criterion.

Our simulation study also showed that traditional Monte Carlo has more outliers in performance than Mosaic Monte Carlo. This means an entire set of results from an individual simulation study could be close or far from the true values with no way to distinguish between the two. It is almost unheard of to consider that an entire set of simulation results could be inaccurate reflections of the underlying data-generating model and yet this occurred multiple times in our main simulation study. In other words, the findings from a traditional Monte

Carlo study could be substantially wrong with no way to detect it. Mosaic Monte Carlo helps to insure against this scenario.

The results from our study were encouraging and suggest that it is feasible to conduct Monte Carlo simulation studies by using this Mosaic Monte Carlo approach. This may be particularly useful for topics which may be sensitive to model parameter values such as missing data analysis and studies involving Bayesian methods. The results from the empirical example confirm that Mosaic Monte Carlo may be useful in settings where methods are not thought to be sensitive to the data-generating process, since Mosaic Monte Carlo was able to detect an issue with the formula of one of the candidate effect size measures.

8.1 Limitations

We evaluated the performance of Mosaic Monte Carlo by examining how estimates of bias and variance were impacted, although there may be other criteria of interest in simulation studies. Furthermore, we examined the performance of our method in a case where sample statistics are unbiased. We also only conducted our simulation study in the context of regression with a small number of variables which may be idealistic. We did not evaluate Mosaic Monte Carlo in contexts involving missing data or Bayesian statistics, although these areas were a prime motivation for developing this procedure.

A limitation of Mosaic Monte Carlo itself is that the user must be careful not to vary population parameters that would make models incomparable. In the context of our study, this applied to the error variance. The particular population parameters that should be fixed depends on the model used, the topic of interest, and the summarizing strategy. Importantly, Mosaic Monte Carlo improves the generalizability of simulation results but only within the boundaries studied. There are still risks that findings based on simulation results will fail to generalize to real data with different characteristics.

Furthermore, the guidance we developed for implementing the screening phase of Mosaic Monte Carlo may not fit all applications. As illustrated by our empirical example, compromises may need to made to reduce computational burden and/or adjustments made with respect to the criteria itself depending on the nature of the study.

8.2 Recommendations

We recommend that researchers who use Monte Carlo simulations in their work consider implementing their studies using Mosaic Monte Carlo, especially when studying methods that are sensitive to parameter values such as missing data analysis and Bayesian methodology. There may be unexpected cases in which aspects of data-generation can have nonsystematic influences on results.

References

Boulesteix, A.-L., Groenwold, R. H., Abrahamowicz, M., Binder, H., Briel, M., Hornung, R., ... Panel, S. (2020). Introduction to statistical simulations in health research. *BMJ Open*, *10*(12), e039921. doi: https://doi.org/10.1136/bmjopen-2020-039921

Boulesteix, A.-L., Lauer, S., & Eugster, M. J. A. (2013). A plea for neutral comparison studies in computational sciences. *PLOS ONE*, *8*(4), e61562. doi: https://doi.org/10.1371/journal.pone.0061562

Boulesteix, A.-L., Stierle, V., & Hapfelmeier, A. (2015). Publication bias in methodological computational research. *Cancer Informatics*, *14*(S5), 11–19. doi: https://doi.org/10.4137/CIN.S30747

Brooks, C. (2002). *Introductory economics for finance*. Cambridge University Press.

Burton, A., Altman, D. G., Royston, P., & Holder, R. L. (2006). The design of simulation studies in medical statistics. *Statistics in Medicine*, *25*(24), 4279–4292. doi: https://doi.org/10.1002/sim.2673

Casella, G., & Berger, R. L. (2001). *Statistical inference* (2nd ed.). Duxbury.

Collins, L. M., Schafer, J. L., & Kam, C.-M. (2001). A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods*, *6*(4), 330–351. doi: https://doi.org/10.1037/1082-989X.6.4.330

Curran, P. J., Bollen, K. A., Paxton, P., Kirby, J., & Chen, F. (2002). The noncentral chi-square distribution in misspecified structural equation models: Finite sample results from a monte carlo simulation. *Multivariate Behavioral Research*, *37*(1), 1–36. doi: https://doi.org/10.1207/s15327906mbr3701_01

Franklin, J. M., Schneeweiss, S., Polinski, J. M., & Rassen, J. A. (2014). Plasmode simulation for the evaluation of pharmacoepidemiologic methods in complex healthcare databases. *Computational Statistics & Data Analysis*, *72*, 219–226. doi: https://doi.org/10.1016/j.csda.2013.10.018

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., & Hothorn, T. (2020). mvtnorm: Multivariate normal and t distributions [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=mvtnorm` (R package version 1.1-1)

Gomer, B., Jiang, G., & Yuan, K.-H. (2019). New effect size measures for structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, *26*(3), 371–389. doi: https://doi.org/10.1080/10705511.2018.1545231

Gomer, B., & Yuan, K.-H. (2021). Subtypes of the missing not at random missing data mechanism. *Psychological Methods*, *26*(5), 559–598. doi: https://doi.org/10.1037/met0000377

Goutelle, S., Bourguignon, L., Maire, P. H., Van Guilder, M., Conte, J. E., & Jelliffe, R. W. (2009). Population modeling and monte carlo simulation study of the pharmacokinetics and antituberculosis pharmacodynamics of rifampin in lungs. *Antimicrobial Agents and Chemotherapy*, *53*(7), 2974–2981. doi: https://doi.org/10.1128/AAC.01520-08

Hu, L.-T., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journal*, *6*(1), 1–55. doi: https://doi.org/10.1080/10705519909540118

Iranmanesh, H., Parchami, A., & Sadeghpour Gildeh, B. (2022). Statistical testing quality and its monte carlo simulation based on fuzzy specification limits. *Iranian Journal of Fuzzy Systems*, *19*(3). doi: https://doi.org/10.22111/ijfs.2022.6940

Ke, Z., & Wang, L. (2015). Detecting individual differences in change: Methods and comparisons. *Structural Equation Modeling: A Multidisciplinary Journal*, *22*(3), 382–400. doi: https://doi.org/10.1080/10705511.2014.936096

Koehler, E., Brown, E., & Haneuse, S. J.-P. A. (2009). On the assessment of monte carlo error in simulation-based statistical analyses. *The American Statistician*, *63*(2), 155–162. doi: https://doi.org/10.1198/tast.2009.0030

Kulinskaya, E., Hoaglin, D. C., & Bakbergenuly, I. (2021). Exploring consequences of simulation design for apparent performance of methods of meta-analysis. *Statistical Methods in Medical Research*, *30*(7), 1667–1690. doi: https://doi.org/10.1177/09622802211013065

Leigh, J. W., & Bryant, D. (2015). Monte carlo strategies for selecting parameter values in simulation experiments. *Systematic Biology*, *64*(5), 741–751. doi: https://doi.org/10.1093/sysbio/syv030

McKay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *42*(1), 55–61. doi: https://doi.org/10.1080/00401706.2000.10485979

Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, *38*(11), 2074–2102. doi: https://doi.org/10.1002/sim.8086

Paxton, P., Curran, P. J., Bollen, K. A., Kirby, J., & Chen, F. (2001). Monte carlo experiments: Design and implementation. *Structural Equation Modeling: A Multidisciplinary Journal*, *8*(2), 287–312. doi: https://doi.org/10.1207/s15328007sem0802_7

Preecha, C. (2004). *Numbers of replications required in anova simulation studies*. University of Northern Colorado.

R Core Team. (2020). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Schaffer, J. R., & Kim, M.-J. (2007). Number of replications required in control chart monte carlo simulation studies. *Communications in Statistics—Simulation and Computation*, *36*(5), 1075–1087. doi: https://doi.org/10.1080/03610910701539963

Schreck, N., Slynko, A., & Saadati, M. (2024). Statistical plasmode simulations—potentials, challenges and recommendations. *Statistics in Medicine*, *43*(9), 1804–1825. doi: https://doi.org/10.1002/sim.10012

Siepe, B. S., Bartoš, F., Morris, T. P., Boulesteix, A.-L., Heck, D. W., & Pawel,

S. (2024). Simulation studies for methodological research in psychology: A standardized template for planning, preregistration, and reporting. *Psychological Methods*. doi: https://doi.org/10.1037/met0000695

Skrondal, A. (2000). Design and analysis of monte carlo experiments: Attacking the conventional wisdom. *Multivariate Behavioral Research*, *35*(2), 137–167. doi: https://doi.org/10.1207/s15327906mbr3502_1

Spence, I. (1983). Monte carlo simulation studies. *Applied Psychological Measurement*, *7*(4), 405–425. doi: https://doi.org/10.1177/014662168300700403

Stevens, J. P. (1992). *Applied multivariate statistics for the social sciences*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Supawan, P. (2004). *An examination of the number of replications required in regression simulation studies*. University of Northern Colorado.

Tuffin, B. (1996). *On the use of low discrepancy sequences in monte carlo methods* (No. 1060). Rennes, France: IRISA.

Appendix: Supplemental Materials

Table A1: SUPP: Aggregated absolute relative bias of $\widehat{\beta}_2$ and $\widehat{\beta}_3$ across all 500 test runs.

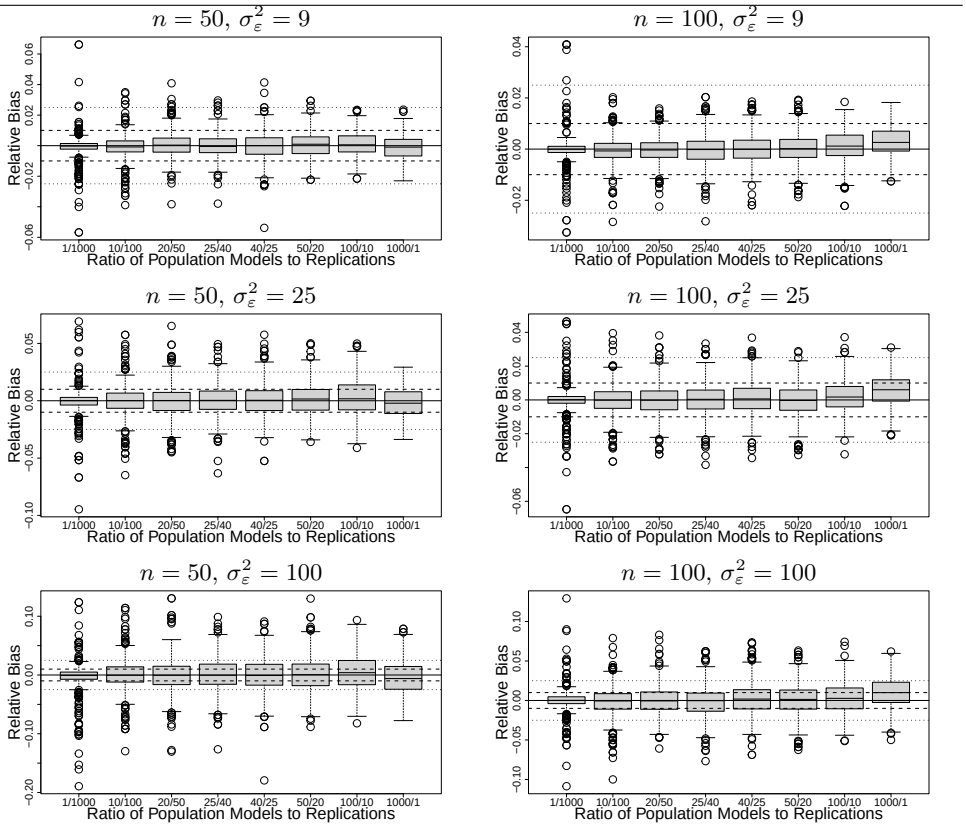
		Randomized Population Models / Replications							
n	σ^2_ε	1/1000	10/100	20/50	25/40	40/25	50/20	100/10	1000/1
					$\widehat{\beta}_2$				
50	9	0.0045	0.0073	0.0073	0.0077	0.0074	0.0077	0.0085	0.0077
	25	0.0077	0.0117	0.0128	0.0127	0.0133	0.0128	0.0137	0.0134
	100	0.0164	0.0226	0.0248	0.0242	0.0262	0.0263	0.0284	0.0263
75	9	0.0036	0.0058	0.0060	0.0064	0.0065	0.0062	0.0066	0.0066
	25	0.0064	0.0093	0.0101	0.0106	0.0104	0.0109	0.0104	0.0103
	100	0.0121	0.0202	0.0195	0.0220	0.0187	0.0212	0.0231	0.0221
100	9	0.0038	0.0052	0.0057	0.0057	0.0051	0.0052	0.0054	0.0054
	25	0.0060	0.0090	0.0087	0.0085	0.0087	0.0090	0.0095	0.0085
	100	0.0099	0.0163	0.0186	0.0181	0.0189	0.0186	0.0202	0.0180
200	9	0.0023	0.0035	0.0037	0.0038	0.0038	0.0039	0.0042	0.0040
	25	0.0037	0.0060	0.0065	0.0058	0.0062	0.0064	0.0066	0.0065
	100	0.0069	0.0119	0.0116	0.0122	0.0126	0.0119	0.0136	0.0131
400	9	0.0019	0.0027	0.0026	0.0028	0.0026	0.0029	0.0032	0.0028
	25	0.0026	0.0045	0.0044	0.0040	0.0043	0.0047	0.0051	0.0042
	100	0.0055	0.0083	0.0086	0.0078	0.0084	0.0088	0.0096	0.0086
					$\widehat{\beta}_3$				
50	9	0.0055	0.0074	0.0087	0.0077	0.0089	0.0087	0.0076	0.0088
	25	0.0088	0.0118	0.0132	0.0138	0.0140	0.0143	0.0128	0.0148
	100	0.0180	0.0236	0.0302	0.0282	0.0287	0.0275	0.0268	0.0296
75	9	0.0045	0.0053	0.0064	0.0068	0.0071	0.0061	0.0068	0.0085
	25	0.0068	0.0100	0.0108	0.0110	0.0118	0.0113	0.0111	0.0136
	100	0.0156	0.0205	0.0228	0.0228	0.0233	0.0229	0.0217	0.0270
100	9	0.0037	0.0051	0.0053	0.0056	0.0061	0.0058	0.0055	0.0053
	25	0.0065	0.0085	0.0086	0.0086	0.0096	0.0091	0.0094	0.0098
	100	0.0134	0.0174	0.0165	0.0184	0.0203	0.0185	0.0199	0.0175
200	9	0.0028	0.0037	0.0041	0.0042	0.0038	0.0039	0.0041	0.0040
	25	0.0044	0.0065	0.0067	0.0072	0.0067	0.0065	0.0069	0.0070
	100	0.0079	0.0128	0.0140	0.0147	0.0128	0.0127	0.0137	0.0138
400	9	0.0015	0.0025	0.0028	0.0030	0.0028	0.0027	0.0026	0.0032
	25	0.0036	0.0042	0.0048	0.0051	0.0047	0.0044	0.0046	0.0052
	100	0.0058	0.0087	0.0084	0.0095	0.0093	0.0089	0.0087	0.0105

Table A2: SUPP: Maximum absolute relative bias of $\widehat{\beta}_{jl}$ across all 500 test runs.

		Randomized Population Models / Replications							
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10	1000/1
					$\widehat{\beta}_2$				
50	9	0.0681	0.0470	0.0577	0.0488	0.0354	0.0407	0.0304	0.0385
	25	0.1714	0.0832	0.0962	0.0828	0.0556	0.0679	0.0442	0.0575
	100	0.3617	0.2013	0.1923	0.1656	0.1180	0.1354	0.0962	0.1284
75	9	0.1069	0.0510	0.0331	0.0367	0.0322	0.0314	0.0286	0.0206
	25	0.1297	0.0773	0.0552	0.0533	0.0478	0.0610	0.0383	0.0502
	100	0.3564	0.1547	0.0933	0.1067	0.0956	0.1221	0.0954	0.1005
100	9	0.0676	0.0451	0.0368	0.0285	0.0247	0.0246	0.0252	0.0219
	25	0.1127	0.0605	0.0710	0.0432	0.0489	0.0421	0.0420	0.0358
	100	0.2061	0.0969	0.0850	0.0949	0.0854	0.0842	0.0779	0.0783
200	9	0.0297	0.0260	0.0230	0.0275	0.0171	0.0237	0.0156	0.0156
	25	0.0544	0.0440	0.0346	0.0311	0.0276	0.0292	0.0291	0.0236
	100	0.0990	0.0973	0.0678	0.0782	0.0552	0.0790	0.0605	0.0471
400	9	0.0251	0.0227	0.0145	0.0118	0.0109	0.0114	0.0115	0.0099
	25	0.0418	0.0551	0.0242	0.0222	0.0181	0.0190	0.0201	0.0149
	100	0.0835	0.0453	0.0484	0.0393	0.0360	0.0392	0.0402	0.0329
					$\widehat{\beta}_3$				
50	9	0.1063	0.0510	0.0504	0.0636	0.0418	0.0400	0.0299	0.0280
	25	0.1772	0.1033	0.0840	0.1060	0.0632	0.0667	0.0529	0.0520
	100	0.4134	0.1421	0.1681	0.1559	0.1263	0.1352	0.1229	0.1040
75	9	0.0744	0.0352	0.0430	0.0343	0.0500	0.0299	0.0277	0.0258
	25	0.0890	0.0569	0.0500	0.0620	0.0509	0.0520	0.0631	0.0469
	100	0.2124	0.1172	0.1433	0.1145	0.1668	0.1118	0.1263	0.0938
100	9	0.0933	0.0383	0.0248	0.0302	0.0437	0.0270	0.0236	0.0186
	25	0.1225	0.0639	0.0407	0.0504	0.0728	0.0422	0.0509	0.0326
	100	0.2269	0.1278	0.0875	0.1007	0.1084	0.1002	0.1018	0.0653
200	9	0.0427	0.0181	0.0212	0.0259	0.0249	0.0193	0.0166	0.0141
	25	0.0699	0.0521	0.0354	0.0432	0.0414	0.0275	0.0281	0.0256
	100	0.1220	0.0676	0.0798	0.0618	0.0641	0.0643	0.0563	0.0511
400	9	0.0201	0.0275	0.0130	0.0148	0.0123	0.0126	0.0105	0.0117
	25	0.0663	0.0459	0.0241	0.0246	0.0205	0.0201	0.0202	0.0212
	100	0.1326	0.0556	0.0622	0.0492	0.0396	0.0402	0.0392	0.0377

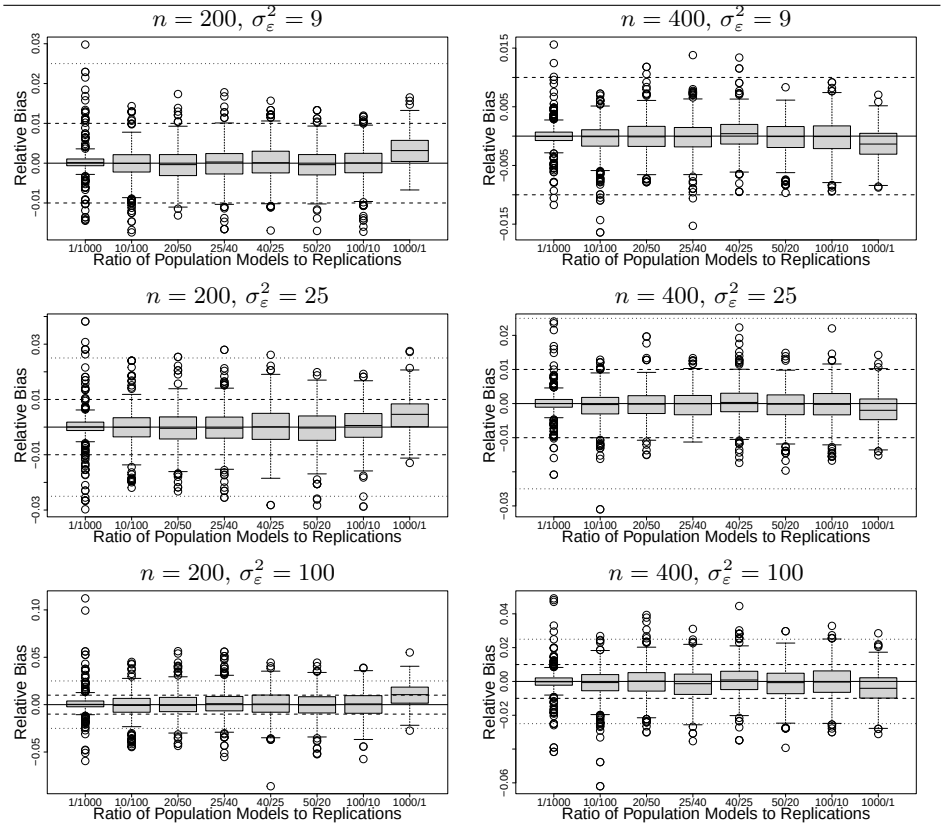
Note. j denotes a particular population model and l denotes a particular regression coefficient.

Table A3: SUPP: Distribution of bias of $\widehat{\beta}_0$ across 500 test runs for $n = 50$ and $n = 100$ across conditions of residual variance



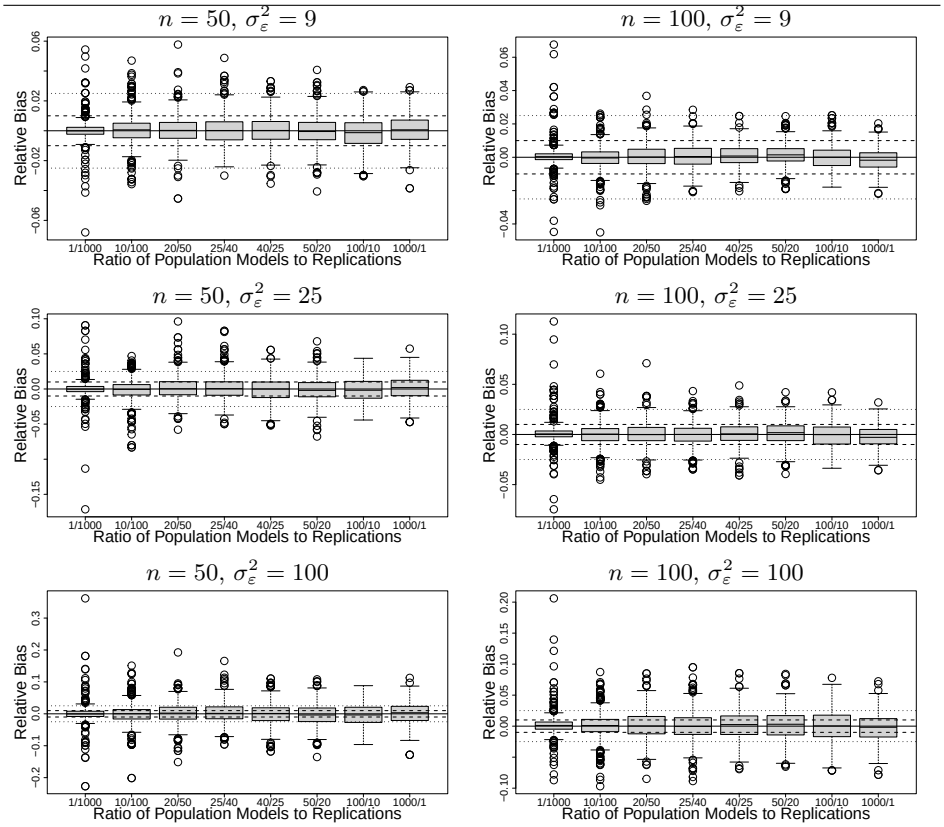
Note. Solid line drawn at 0, dashed and dotted lines denote relative bias of 1% and 2.5% respectively.

Table A4: SUPP: Distribution of bias of $\widehat{\beta}_0$ across 500 test runs for $n = 200$ and $n = 400$ across conditions of residual variance



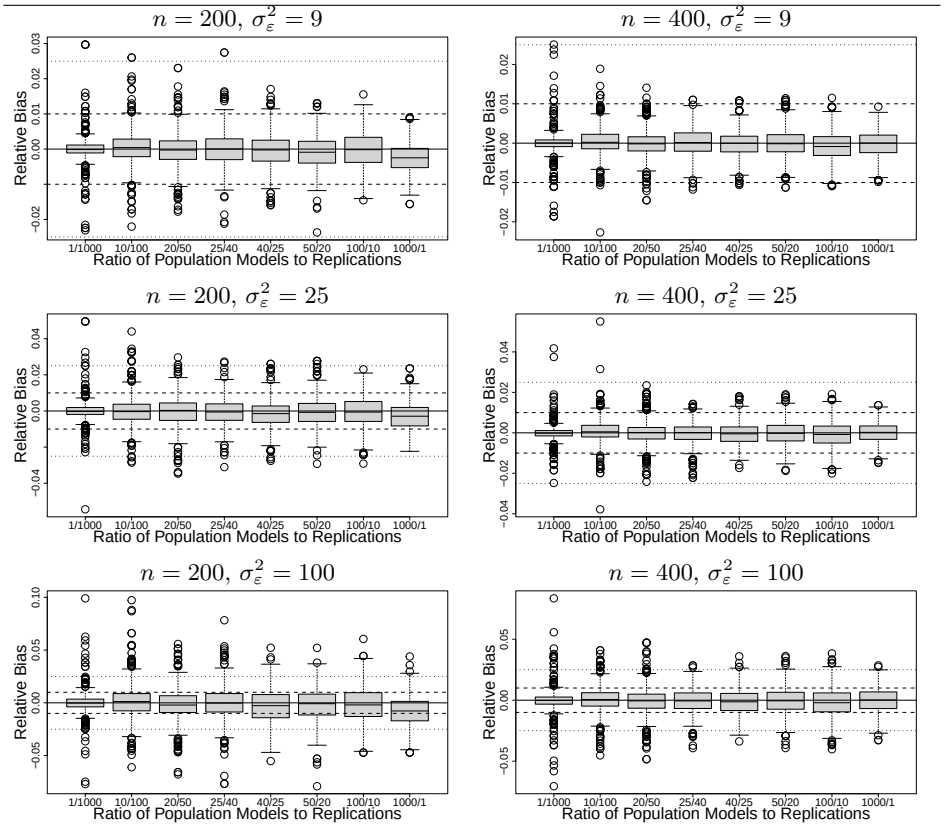
Note. Solid line drawn at 0, dashed and dotted lines denote relative bias of 1% and 2.5% respectively.

Table A5: SUPP: Distribution of bias of $\widehat{\beta}_2$ across 500 test runs for $n = 50$ and $n = 100$ across conditions of residual variance



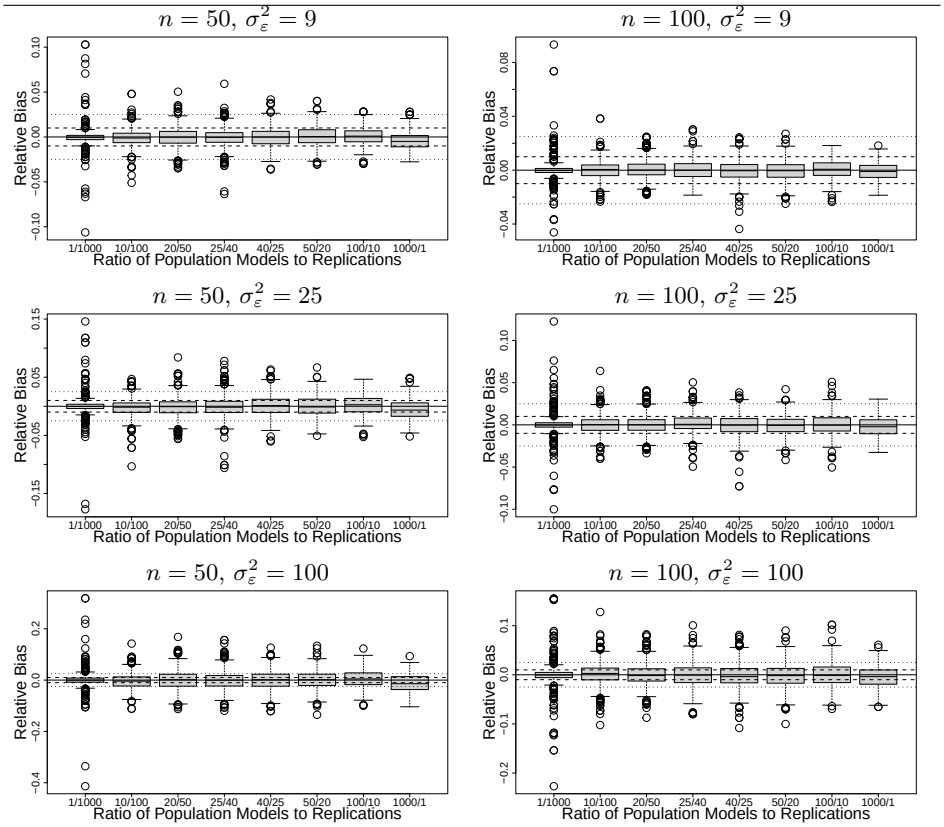
Note. Solid line drawn at 0, dashed and dotted lines denote relative bias of 1% and 2.5% respectively.

Table A6: SUPP: Distribution of bias of $\widehat{\beta}_2$ across 500 test runs for $n = 200$ and $n = 400$ across conditions of residual variance



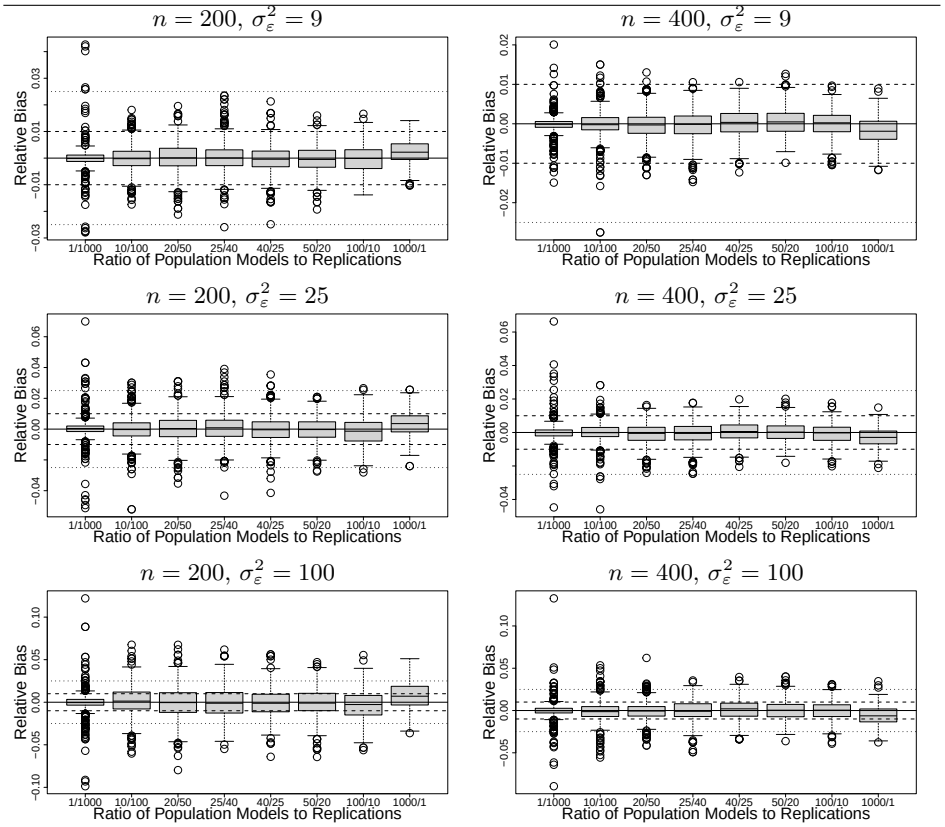
Note. Solid line drawn at 0, dashed and dotted lines denote relative bias of 1% and 2.5% respectively.

Table A7: SUPP: Distribution of bias of $\widehat{\beta}_3$ across 500 test runs for $n = 50$ and $n = 100$ across conditions of residual variance



Note. Solid line drawn at 0, dashed and dotted lines denote relative bias of 1% and 2.5% respectively.

Table A8: SUPP: Distribution of bias of $\widehat{\beta}_3$ across 500 test runs for $n = 200$ and $n = 400$ across conditions of residual variance



Note. Solid line drawn at 0, dashed and dotted lines denote relative bias of 1% and 2.5% respectively.

Table A9: SUPP: Aggregated ratio of empirical to true variance of $\widehat{\beta}_2$ and $\widehat{\beta}_3$ across all 500 test runs.

		Randomized Population Models / Replications						
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
					$\widehat{\beta}_2$			
50	9	0.9991	0.9941	0.9976	0.9976	0.9983	1.0041	0.9975
	25	1.0026	0.9916	0.9996	0.9953	1.0013	1.0012	0.9967
	100	0.9994	0.9943	1.0002	0.9996	0.9963	1.0011	0.9979
75	9	0.9992	1.0000	1.0022	0.9973	1.0044	1.0005	1.0026
	25	1.0003	0.9992	1.0007	1.0019	0.9993	0.9951	1.0039
	100	0.9999	0.9997	1.0014	1.0006	0.9960	0.9972	1.0038
100	9	0.9996	1.0022	0.9974	1.0022	1.0022	1.0032	0.9969
	25	0.9998	1.0037	0.9982	1.0008	0.9979	1.0016	0.9968
	100	1.0000	1.0018	0.9975	1.0037	0.9988	1.0021	0.9939
200	9	1.0021	0.9955	1.0017	0.9989	0.9949	0.9917	1.0058
	25	1.0035	0.9930	1.0020	1.0001	0.9964	0.9969	1.0043
	100	0.9990	0.9914	0.9979	0.9986	0.9954	0.9914	0.9972
400	9	0.9958	0.9897	1.0022	1.0003	1.0040	1.0035	0.9975
	25	1.0020	0.9973	1.0030	0.9987	1.0038	1.0026	1.0037
	100	1.0023	0.9961	0.9980	1.0002	1.0010	1.0009	0.9959
					$\widehat{\beta}_3$			
50	9	0.9986	1.0009	0.9995	1.0063	1.0039	1.0053	1.0147
	25	0.9987	0.9990	1.0010	1.0063	1.0058	1.0055	1.0153
	100	1.0024	0.9978	1.0016	1.0005	1.0094	1.0101	1.0125
75	9	0.9979	1.0012	1.0112	1.0095	1.0031	1.0016	1.0113
	25	1.0008	1.0047	1.0090	1.0064	1.0035	1.0060	1.0111
	100	1.0021	1.0024	1.0113	1.0095	1.0043	1.0115	1.0131
100	9	1.0001	0.9962	0.9954	1.0032	1.0019	1.0003	1.0135
	25	0.9977	0.9990	0.9997	1.0018	0.9971	0.9981	1.0123
	100	0.9974	0.9991	1.0010	1.0070	0.9995	1.0001	1.0121
200	9	0.9997	0.9995	1.0035	1.0060	0.9992	1.0048	1.0051
	25	1.0004	1.0005	1.0021	1.0070	1.0006	1.0032	1.0078
	100	1.0004	1.0008	1.0028	1.0089	1.0032	1.0036	1.0036
400	9	0.9931	0.9968	0.9991	0.9992	0.9953	0.9991	0.9816
	25	0.9995	0.9989	0.9972	0.9967	0.9929	0.9960	0.9880
	100	1.0031	0.9973	1.0002	0.9955	0.9954	0.9979	0.9870

Table A10: SUPP: Aggregated true vs. empirical variance of $\widehat{\beta}_0$ across all 500 test runs.

			Randomized Population Models / Replications						
n	σ_ε^2	Value	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
50	9	True	0.1923	0.1920	0.1916	0.1917	0.1917	0.1918	0.1917
		Empirical	0.1918	0.1911	0.1911	0.1918	0.1927	0.1924	0.1926
	25	True	0.5327	0.5328	0.5323	0.5324	0.5327	0.5330	0.5326
		Empirical	0.5321	0.5322	0.5316	0.5322	0.5337	0.5342	0.5379
	100	True	2.1232	2.1330	2.1281	2.1294	2.1307	2.1307	2.1309
		Empirical	2.1177	2.1318	2.1286	2.1298	2.1361	2.1329	2.1417
75	9	True	0.1250	0.1251	0.1250	0.1250	0.1250	0.1250	0.1250
		Empirical	0.1251	0.1245	0.1249	0.1247	0.1254	0.1252	0.1245
	25	True	0.3476	0.3473	0.3470	0.3470	0.3472	0.3473	0.3472
		Empirical	0.3484	0.3450	0.3461	0.3474	0.3485	0.3486	0.3451
	100	True	1.3838	1.3889	1.3883	1.3888	1.3892	1.3891	1.3894
		Empirical	1.3831	1.3884	1.3864	1.3878	1.3942	1.3978	1.3825
100	9	True	0.0928	0.0928	0.0927	0.0928	0.0928	0.0928	0.0928
		Empirical	0.0928	0.0926	0.0925	0.0925	0.0926	0.0929	0.0919
	25	True	0.2578	0.2578	0.2578	0.2576	0.2578	0.2578	0.2577
		Empirical	0.2568	0.2567	0.2569	0.2567	0.2563	0.2566	0.2558
	100	True	1.0301	1.0311	1.0309	1.0308	1.0310	1.0309	1.0309
		Empirical	1.0291	1.0276	1.0269	1.0290	1.0279	1.0298	1.0218
200	9	True	0.0457	0.0457	0.0457	0.0457	0.0457	0.0457	0.0457
		Empirical	0.0458	0.0455	0.0455	0.0454	0.0453	0.0455	0.0453
	25	True	0.1268	0.1269	0.1269	0.1269	0.1269	0.1269	0.1269
		Empirical	0.1265	0.1268	0.1269	0.1262	0.1258	0.1265	0.1258
	100	True	0.5074	0.5074	0.5074	0.5075	0.5076	0.5076	0.5076
		Empirical	0.5088	0.5062	0.5070	0.5050	0.5026	0.5041	0.5046
400	9	True	0.0227	0.0227	0.0227	0.0227	0.0227	0.0227	0.0227
		Empirical	0.0226	0.0226	0.0226	0.0226	0.0225	0.0225	0.0225
	25	True	0.0630	0.0630	0.0630	0.0630	0.0630	0.0630	0.0630
		Empirical	0.0629	0.0630	0.0629	0.0628	0.0624	0.0625	0.0623
	100	True	0.2519	0.2519	0.2519	0.2519	0.2519	0.2519	0.2519
		Empirical	0.2512	0.2517	0.2499	0.2509	0.2493	0.2497	0.2505

Table A11: SUPP: Aggregated true vs. empirical variance of $\widehat{\beta}_1$ across all 500 test runs.

			Randomized Population Models / Replications						
n	σ_ε^2	Value	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
50	9	True	0.2640	0.2666	0.2658	0.2666	0.2675	0.2671	0.2663
		Empirical	0.2639	0.2687	0.2665	0.2678	0.2694	0.2709	0.2700
	25	True	0.7516	0.7390	0.7391	0.7399	0.7440	0.7429	0.7410
		Empirical	0.7526	0.7432	0.7436	0.7425	0.7491	0.7504	0.7524
	100	True	2.9583	2.9641	2.9599	2.9578	2.9669	2.9615	2.9595
		Empirical	2.9455	2.9812	2.9696	2.9594	2.9807	2.9968	3.0004
75	9	True	0.1728	0.1712	0.1701	0.1702	0.1712	0.1705	0.1709
		Empirical	0.1725	0.1709	0.1703	0.1703	0.1711	0.1705	0.1729
	25	True	0.4845	0.4746	0.4708	0.4721	0.4758	0.4746	0.4739
		Empirical	0.4834	0.4741	0.4735	0.4750	0.4747	0.4729	0.4801
	100	True	1.9202	1.8960	1.8886	1.8922	1.9012	1.9011	1.8969
		Empirical	1.9176	1.9004	1.8971	1.8988	1.8941	1.8955	1.9192
100	9	True	0.1266	0.1255	0.1251	0.1253	0.1259	0.1258	0.1258
		Empirical	0.1267	0.1260	0.1246	0.1250	0.1258	0.1250	0.1233
	25	True	0.3457	0.3511	0.3476	0.3476	0.3493	0.3490	0.3495
		Empirical	0.3452	0.3527	0.3463	0.3463	0.3477	0.3474	0.3432
	100	True	1.4151	1.4017	1.3987	1.3987	1.3983	1.3952	1.3963
		Empirical	1.4158	1.4063	1.3988	1.3965	1.3962	1.3878	1.3682
200	9	True	0.0613	0.0616	0.0612	0.0612	0.0615	0.0614	0.0613
		Empirical	0.0618	0.0614	0.0609	0.0607	0.0610	0.0615	0.0615
	25	True	0.1698	0.1713	0.1702	0.1705	0.1707	0.1700	0.1703
		Empirical	0.1708	0.1703	0.1698	0.1702	0.1698	0.1707	0.1713
	100	True	0.6795	0.6863	0.6832	0.6833	0.6824	0.6817	0.6806
		Empirical	0.6808	0.6821	0.6812	0.6813	0.6790	0.6831	0.6803
400	9	True	0.0302	0.0304	0.0303	0.0302	0.0303	0.0303	0.0303
		Empirical	0.0301	0.0303	0.0302	0.0301	0.0302	0.0302	0.0303
	25	True	0.0848	0.0847	0.0843	0.0843	0.0844	0.0842	0.0843
		Empirical	0.0852	0.0850	0.0838	0.0837	0.0840	0.0841	0.0844
	100	True	0.3404	0.3370	0.3373	0.3365	0.3375	0.3371	0.3374
		Empirical	0.3407	0.3372	0.3363	0.3358	0.3377	0.3373	0.3370

Table A12: SUPP: Aggregated true vs. empirical variance of $\widehat{\beta}_2$ across all 500 test runs.

			Randomized Population Models / Replications						
n	σ_ε^2	Value	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
50	9	True	0.2646	0.2683	0.2676	0.2687	0.2673	0.2668	0.2681
		Empirical	0.2644	0.2667	0.2669	0.2681	0.2668	0.2679	0.2674
	25	True	0.7503	0.7498	0.7420	0.7435	0.7463	0.7466	0.7453
		Empirical	0.7522	0.7436	0.7417	0.7400	0.7473	0.7475	0.7429
	100	True	2.9521	2.9700	2.9817	2.9777	2.9795	2.9676	2.9776
		Empirical	2.9503	2.9530	2.9823	2.9764	2.9685	2.9710	2.9713
75	9	True	0.1739	0.1727	0.1717	0.1726	0.1723	0.1717	0.1723
		Empirical	0.1738	0.1727	0.1721	0.1722	0.1730	0.1717	0.1727
	25	True	0.4807	0.4788	0.4784	0.4784	0.4778	0.4787	0.4771
		Empirical	0.4808	0.4784	0.4787	0.4793	0.4775	0.4763	0.4790
	100	True	1.8823	1.9143	1.9038	1.9110	1.9061	1.9109	1.9119
		Empirical	1.8821	1.9136	1.9065	1.9121	1.8984	1.9055	1.9191
100	9	True	0.1266	0.1264	0.1266	0.1267	0.1263	0.1264	0.1267
		Empirical	0.1266	0.1267	0.1263	0.1270	0.1266	0.1268	0.1263
	25	True	0.3526	0.3520	0.3506	0.3512	0.3526	0.3518	0.3522
		Empirical	0.3526	0.3533	0.3499	0.3515	0.3518	0.3524	0.3511
	100	True	1.4456	1.4069	1.4048	1.4069	1.4090	1.4100	1.4105
		Empirical	1.4456	1.4094	1.4012	1.4122	1.4073	1.4129	1.4019
200	9	True	0.0621	0.0618	0.0617	0.0616	0.0617	0.0617	0.0617
		Empirical	0.0622	0.0615	0.0618	0.0615	0.0614	0.0612	0.0620
	25	True	0.1701	0.1714	0.1711	0.1710	0.1713	0.1710	0.1713
		Empirical	0.1707	0.1702	0.1714	0.1710	0.1706	0.1704	0.1721
	100	True	0.6706	0.6848	0.6863	0.6846	0.6843	0.6842	0.6856
		Empirical	0.6700	0.6789	0.6848	0.6837	0.6812	0.6783	0.6837
400	9	True	0.0303	0.0304	0.0304	0.0304	0.0304	0.0304	0.0304
		Empirical	0.0302	0.0301	0.0305	0.0304	0.0305	0.0305	0.0304
	25	True	0.0853	0.0846	0.0845	0.0845	0.0847	0.0845	0.0846
		Empirical	0.0855	0.0844	0.0848	0.0844	0.0850	0.0848	0.0850
	100	True	0.3424	0.3380	0.3370	0.3376	0.3380	0.3380	0.3381
		Empirical	0.3432	0.3366	0.3363	0.3376	0.3383	0.3383	0.3367

Table A13: SUPP: Aggregated true vs. empirical variance of $\widehat{\beta}_3$ across all 500 test runs.

			Randomized Population Models / Replications						
n	σ_ε^2	Value	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
50	9	True	0.2667	0.2689	0.2690	0.2687	0.2682	0.2682	0.2686
		Empirical	0.2663	0.2691	0.2688	0.2703	0.2692	0.2697	0.2726
	25	True	0.7523	0.7511	0.7467	0.7468	0.7472	0.7483	0.7474
		Empirical	0.7513	0.7503	0.7474	0.7515	0.7515	0.7524	0.7588
	100	True	2.9852	2.9920	2.9720	2.9774	2.9885	2.9894	2.9905
		Empirical	2.9924	2.9852	2.9768	2.9790	3.0167	3.0195	3.0278
75	9	True	0.1761	0.1727	0.1727	0.1728	0.1722	0.1723	0.1726
		Empirical	0.1757	0.1729	0.1746	0.1744	0.1728	0.1726	0.1745
	25	True	0.4760	0.4786	0.4773	0.4777	0.4784	0.4786	0.4782
		Empirical	0.4764	0.4808	0.4816	0.4808	0.4801	0.4815	0.4835
	100	True	1.8835	1.9193	1.9109	1.9082	1.9104	1.9135	1.9133
		Empirical	1.8875	1.9240	1.9325	1.9262	1.9186	1.9355	1.9385
100	9	True	0.1270	0.1260	0.1262	0.1262	0.1265	0.1265	0.1267
		Empirical	0.1270	0.1255	0.1256	0.1266	0.1267	0.1265	0.1284
	25	True	0.3548	0.3514	0.3515	0.3514	0.3515	0.3518	0.3521
		Empirical	0.3540	0.3511	0.3514	0.3520	0.3505	0.3512	0.3564
	100	True	1.4131	1.4099	1.4135	1.4101	1.4060	1.4072	1.4072
		Empirical	1.4095	1.4086	1.4148	1.4200	1.4053	1.4073	1.4242
200	9	True	0.0623	0.0614	0.0614	0.0613	0.0615	0.0616	0.0615
		Empirical	0.0623	0.0614	0.0616	0.0617	0.0614	0.0619	0.0618
	25	True	0.1712	0.1720	0.1705	0.1709	0.1710	0.1705	0.1710
		Empirical	0.1713	0.1721	0.1709	0.1721	0.1711	0.1711	0.1723
	100	True	0.6946	0.6855	0.6834	0.6844	0.6843	0.6842	0.6826
		Empirical	0.6949	0.6861	0.6853	0.6904	0.6865	0.6866	0.6850
400	9	True	0.0305	0.0303	0.0304	0.0303	0.0303	0.0304	0.0304
		Empirical	0.0303	0.0302	0.0303	0.0303	0.0302	0.0303	0.0298
	25	True	0.0859	0.0844	0.0844	0.0845	0.0844	0.0844	0.0846
		Empirical	0.0858	0.0843	0.0842	0.0842	0.0838	0.0840	0.0835
	100	True	0.3400	0.3377	0.3377	0.3373	0.3376	0.3372	0.3378
		Empirical	0.3411	0.3368	0.3378	0.3358	0.3361	0.3365	0.3334

Table A14: SUPP: Maximum of maximum MCSEs of bias of regression coefficients across all 500 test runs

		Randomized Population Models / Replications						
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
					$\widehat{\beta}_2$			
50	9	0.0659	0.2347	0.3511	0.4558	0.5422	0.6435	1.0834
	25	0.1099	0.4113	0.7493	0.7596	0.8867	1.0726	1.8056
	100	0.2198	0.8119	1.1664	1.5193	1.9508	2.3412	3.8010
75	9	0.0498	0.2128	0.2914	0.3497	0.4663	0.5569	0.8689
	25	0.0762	0.3547	0.4856	0.5417	0.7250	0.8467	1.3215
	100	0.1346	0.6683	0.8911	1.1658	1.5543	1.8562	2.7909
100	9	0.0444	0.1588	0.2532	0.2698	0.3909	0.4109	0.8254
	25	0.0575	0.3204	0.4221	0.4936	0.6618	0.8847	1.3756
	100	0.1335	0.6409	0.7214	0.8987	1.3235	1.5343	2.3510
200	9	0.0296	0.1163	0.1812	0.2047	0.2837	0.3554	0.4941
	25	0.0493	0.1858	0.2682	0.3402	0.4218	0.4952	0.8532
	100	0.1004	0.3628	0.6042	0.6823	0.9455	0.9383	1.7189
400	9	0.0168	0.0770	0.1350	0.1442	0.1891	0.2149	0.3488
	25	0.0296	0.1291	0.2145	0.2244	0.3059	0.3478	0.6280
	100	0.0721	0.2419	0.3358	0.4806	0.6304	0.6956	1.2559
					$\widehat{\beta}_3$			
50	9	0.0681	0.2757	0.4511	0.4621	0.6015	0.8532	1.2181
	25	0.0967	0.4034	0.6574	0.9384	1.0634	1.4221	1.7772
	100	0.1935	0.7525	1.2428	1.3177	2.1267	2.4602	5.1888
75	9	0.0574	0.1946	0.3072	0.3436	0.4474	0.5282	0.9045
	25	0.0958	0.3244	0.5232	0.5727	0.7451	0.9289	1.7120
	100	0.1914	0.6816	1.0463	1.0725	1.4182	1.8579	3.4241
100	9	0.0457	0.1854	0.2544	0.2826	0.3928	0.4934	0.6848
	25	0.0818	0.2604	0.4225	0.5314	0.6547	0.8232	1.1981
	100	0.1311	0.5208	0.8411	1.0627	1.3094	1.6447	2.1348
200	9	0.0325	0.1279	0.1956	0.1888	0.2958	0.3393	0.4952
	25	0.0541	0.2132	0.3260	0.3564	0.4930	0.4983	0.8253
	100	0.1075	0.4305	0.5744	0.7216	0.9860	1.1083	1.6311
400	9	0.0225	0.0761	0.1349	0.1418	0.1808	0.2506	0.3465
	25	0.0391	0.1288	0.2249	0.2212	0.3062	0.4177	0.5894
	100	0.0771	0.2727	0.4498	0.4489	0.5962	0.8313	1.1788

Table A15: Aggregated average MCSE of bias of regression coefficients across 500 test runs

		Randomized Population Models / Replications						
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
					$\widehat{\beta}_2$			
50	9	0.0065	0.0201	0.0274	0.0308	0.0383	0.0427	0.0596
	25	0.0104	0.0333	0.0454	0.0505	0.0632	0.0714	0.0990
	100	0.0209	0.0650	0.0899	0.1004	0.1278	0.1423	0.1984
75	9	0.0046	0.0160	0.0214	0.0243	0.0307	0.0341	0.0481
	25	0.0081	0.0255	0.0354	0.0398	0.0506	0.0566	0.0788
	100	0.0152	0.0517	0.0719	0.0811	0.1018	0.1142	0.1611
100	9	0.0045	0.0136	0.0187	0.0209	0.0261	0.0291	0.0409
	25	0.0075	0.0229	0.0310	0.0345	0.0439	0.0490	0.0685
	100	0.0130	0.0451	0.0621	0.0698	0.0884	0.0978	0.1361
200	9	0.0029	0.0095	0.0131	0.0145	0.0184	0.0206	0.0288
	25	0.0049	0.0161	0.0217	0.0243	0.0308	0.0343	0.0480
	100	0.0090	0.0306	0.0427	0.0481	0.0611	0.0685	0.0957
400	9	0.0020	0.0069	0.0095	0.0106	0.0132	0.0147	0.0203
	25	0.0032	0.0110	0.0154	0.0170	0.0218	0.0242	0.0337
	100	0.0066	0.0210	0.0296	0.0335	0.0427	0.0479	0.0664
					$\widehat{\beta}_3$			
50	9	0.0069	0.0205	0.0296	0.0328	0.0407	0.0452	0.0625
	25	0.0112	0.0335	0.0474	0.0536	0.0669	0.0744	0.1033
	100	0.0220	0.0688	0.0985	0.1097	0.1369	0.1501	0.2078
75	9	0.0055	0.0166	0.0237	0.0267	0.0328	0.0364	0.0504
	25	0.0092	0.0274	0.0391	0.0437	0.0552	0.0614	0.0842
	100	0.0185	0.0547	0.0784	0.0875	0.1079	0.1201	0.1668
100	9	0.0045	0.0139	0.0201	0.0224	0.0277	0.0309	0.0427
	25	0.0080	0.0228	0.0331	0.0369	0.0462	0.0512	0.0712
	100	0.0154	0.0465	0.0662	0.0738	0.0925	0.1031	0.1431
200	9	0.0035	0.0102	0.0143	0.0158	0.0194	0.0216	0.0298
	25	0.0059	0.0168	0.0241	0.0267	0.0328	0.0365	0.0500
	100	0.0105	0.0331	0.0467	0.0524	0.0649	0.0722	0.0996
400	9	0.0022	0.0070	0.0101	0.0111	0.0137	0.0152	0.0207
	25	0.0044	0.0117	0.0168	0.0187	0.0229	0.0254	0.0347
	100	0.0072	0.0228	0.0325	0.0363	0.0452	0.0504	0.0694

Table A16: Aggregated average MCSE of variance of regression coefficients across 500 test runs

		Randomized Population Models / Replications						
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
					$\widehat{\beta}_0$			
50	9	0.0445	0.1383	0.1927	0.2152	0.2575	0.2798	0.3604
50	25	0.0443	0.1383	0.1920	0.2144	0.2575	0.2792	0.3615
50	100	0.0445	0.1384	0.1921	0.2146	0.2570	0.2792	0.3612
75	9	0.0445	0.1384	0.1909	0.2145	0.2566	0.2774	0.3615
75	25	0.0447	0.1379	0.1911	0.2149	0.2572	0.2780	0.3614
75	100	0.0447	0.1380	0.1905	0.2148	0.2567	0.2777	0.3616
100	9	0.0445	0.1387	0.1916	0.2140	0.2581	0.2797	0.3636
100	25	0.0444	0.1390	0.1918	0.2140	0.2584	0.2797	0.3647
100	100	0.0445	0.1381	0.1918	0.2146	0.2579	0.2792	0.3647
200	9	0.0445	0.1385	0.1918	0.2144	0.2576	0.2785	0.3611
200	25	0.0446	0.1383	0.1919	0.2147	0.2574	0.2789	0.3604
200	100	0.0443	0.1384	0.1917	0.2149	0.2576	0.2780	0.3607
400	9	0.0447	0.1380	0.1921	0.2144	0.2579	0.2804	0.3644
400	25	0.0446	0.1381	0.1923	0.2147	0.2582	0.2800	0.3636
400	100	0.0445	0.1385	0.1915	0.2142	0.2587	0.2805	0.3642
					$\widehat{\beta}_1$			
50	9	0.0443	0.1388	0.1912	0.2136	0.2583	0.2791	0.3627
50	25	0.0446	0.1386	0.1916	0.2138	0.2582	0.2790	0.3630
50	100	0.0449	0.1383	0.1910	0.2141	0.2579	0.2787	0.3634
75	9	0.0448	0.1384	0.1920	0.2156	0.2574	0.2796	0.3622
75	25	0.0447	0.1378	0.1914	0.2149	0.2576	0.2792	0.3613
75	100	0.0448	0.1387	0.1918	0.2150	0.2580	0.2792	0.3618
100	9	0.0448	0.1385	0.1915	0.2141	0.2552	0.2779	0.3626
100	25	0.0447	0.1385	0.1913	0.2137	0.2550	0.2775	0.3622
100	100	0.0445	0.1384	0.1916	0.2136	0.2554	0.2781	0.3620
200	9	0.0445	0.1378	0.1917	0.2148	0.2583	0.2796	0.3624
200	25	0.0445	0.1375	0.1923	0.2152	0.2588	0.2802	0.3620
200	100	0.0445	0.1376	0.1920	0.2148	0.2583	0.2796	0.3610
400	9	0.0445	0.1382	0.1913	0.2135	0.2571	0.2781	0.3611
400	25	0.0446	0.1382	0.1914	0.2133	0.2564	0.2773	0.3620
400	100	0.0446	0.1384	0.1917	0.2136	0.2558	0.2771	0.3620

Table A17: SUPP: Aggregated average MCSE of variance of regression coefficients across 500 test runs

		Randomized Population Models / Replications						
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
					$\widehat{\beta}_2$			
50	9	0.0445	0.1387	0.1926	0.2146	0.2566	0.2787	0.3599
50	25	0.0442	0.1387	0.1921	0.2141	0.2561	0.2787	0.3599
50	100	0.0445	0.1386	0.1922	0.2142	0.2563	0.2784	0.3606
75	9	0.0446	0.1375	0.1908	0.2139	0.2561	0.2784	0.3612
75	25	0.0444	0.1380	0.1911	0.2134	0.2563	0.2784	0.3612
75	100	0.0445	0.1377	0.1911	0.2137	0.2559	0.2781	0.3617
100	9	0.0442	0.1384	0.1918	0.2135	0.2563	0.2782	0.3600
100	25	0.0444	0.1387	0.1917	0.2141	0.2564	0.2787	0.3595
100	100	0.0446	0.1387	0.1912	0.2139	0.2565	0.2785	0.3595
200	9	0.0444	0.1376	0.1909	0.2136	0.2554	0.2786	0.3634
200	25	0.0442	0.1382	0.1905	0.2136	0.2559	0.2788	0.3632
200	100	0.0444	0.1381	0.1912	0.2142	0.2554	0.2778	0.3635
400	9	0.0445	0.1375	0.1916	0.2149	0.2580	0.2800	0.3659
400	25	0.0442	0.1374	0.1921	0.2147	0.2575	0.2796	0.3664
400	100	0.0447	0.1376	0.1913	0.2143	0.2580	0.2801	0.3660
					$\widehat{\beta}_3$			
50	9	0.0444	0.1381	0.1913	0.2134	0.2569	0.2806	0.3609
50	25	0.0445	0.1382	0.1911	0.2133	0.2569	0.2800	0.3620
50	100	0.0444	0.1379	0.1911	0.2134	0.2571	0.2808	0.3614
75	9	0.0446	0.1383	0.1922	0.2162	0.2576	0.2788	0.3597
75	25	0.0445	0.1380	0.1922	0.2152	0.2573	0.2786	0.3601
75	100	0.0446	0.1381	0.1918	0.2150	0.2573	0.2785	0.3593
100	9	0.0447	0.1386	0.1901	0.2131	0.2567	0.2782	0.3614
100	25	0.0446	0.1382	0.1909	0.2136	0.2559	0.2771	0.3615
100	100	0.0447	0.1380	0.1905	0.2135	0.2562	0.2774	0.3618
200	9	0.0447	0.1386	0.1927	0.2145	0.2568	0.2784	0.3618
200	25	0.0448	0.1383	0.1928	0.2151	0.2568	0.2784	0.3624
200	100	0.0446	0.1386	0.1926	0.2151	0.2569	0.2784	0.3621
400	9	0.0448	0.1381	0.1912	0.2143	0.2565	0.2784	0.3616
400	25	0.0448	0.1379	0.1914	0.2139	0.2561	0.2789	0.3616
400	100	0.0445	0.1377	0.1925	0.2142	0.2565	0.2786	0.3612

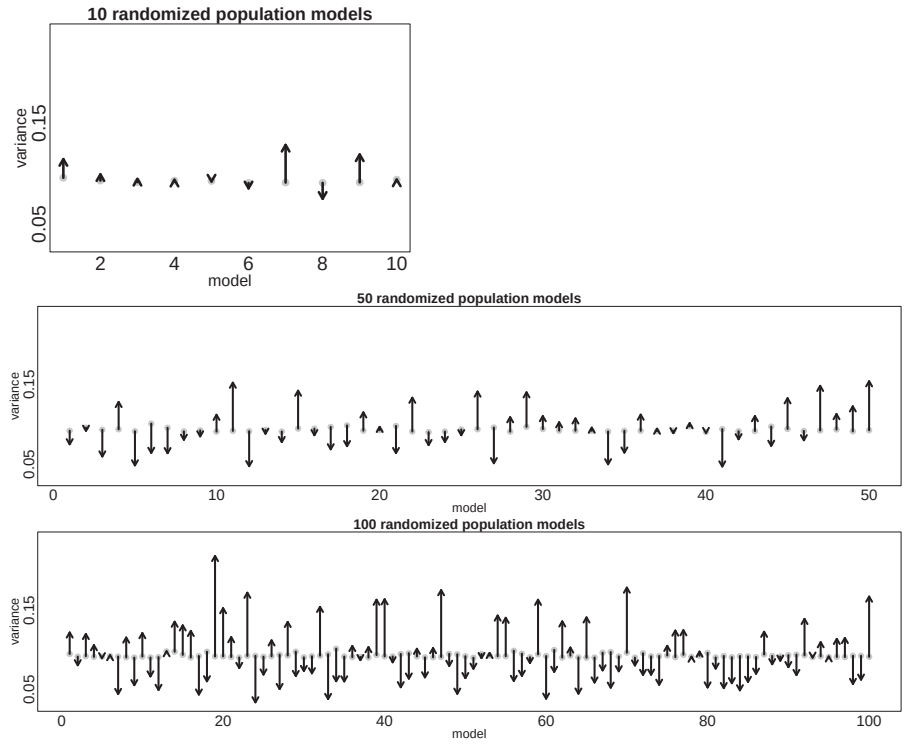
Table A18: SUPP: Maximum of maximum MCSEs of variance of regression coefficients across 500 test runs

		Randomized Population Models / Replications						
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
					$\widehat{\beta}_0$			
50	9	0.0543	0.1567	0.2126	0.2372	0.2898	0.2987	0.3841
50	25	0.0543	0.1532	0.2120	0.2372	0.2898	0.3006	0.3841
50	100	0.0543	0.1567	0.2127	0.2334	0.2898	0.3047	0.3841
75	9	0.0556	0.1562	0.2155	0.2376	0.2887	0.3022	0.3858
75	25	0.0523	0.1530	0.2126	0.2429	0.2887	0.3011	0.3831
75	100	0.0521	0.1562	0.2049	0.2429	0.2778	0.3045	0.3906
100	9	0.0529	0.1565	0.2100	0.2390	0.2824	0.3066	0.3842
100	25	0.0526	0.1567	0.2112	0.2387	0.2824	0.3066	0.3847
100	100	0.0523	0.1567	0.2075	0.2390	0.2794	0.3088	0.3842
200	9	0.0522	0.1534	0.2119	0.2403	0.2820	0.3008	0.3834
200	25	0.0522	0.1534	0.2143	0.2403	0.2959	0.3009	0.3834
200	100	0.0522	0.1558	0.2099	0.2403	0.2959	0.3009	0.3824
400	9	0.0546	0.1535	0.2143	0.2436	0.2844	0.3022	0.3954
400	25	0.0551	0.1550	0.2145	0.2362	0.2828	0.3044	0.3889
400	100	0.0551	0.1550	0.2145	0.2436	0.2812	0.3014	0.3881
					$\widehat{\beta}_1$			
50	9	0.0550	0.1575	0.2162	0.2363	0.2871	0.3060	0.3874
50	25	0.0561	0.1557	0.2127	0.2408	0.2871	0.3060	0.3874
50	100	0.0561	0.1575	0.2103	0.2402	0.2818	0.3011	0.3874
75	9	0.0542	0.1559	0.2134	0.2361	0.2839	0.3107	0.3859
75	25	0.0542	0.1529	0.2117	0.2361	0.2802	0.3017	0.3844
75	100	0.0542	0.1559	0.2117	0.2337	0.2859	0.3027	0.3858
100	9	0.0526	0.1541	0.2110	0.2448	0.2753	0.3034	0.3922
100	25	0.0553	0.1542	0.2145	0.2448	0.2820	0.3004	0.3922
100	100	0.0553	0.1544	0.2091	0.2448	0.2824	0.2985	0.3806
200	9	0.0553	0.1549	0.2148	0.2408	0.2863	0.3033	0.3869
200	25	0.0553	0.1549	0.2104	0.2370	0.2863	0.3045	0.3869
200	100	0.0517	0.1549	0.2148	0.2351	0.2863	0.3038	0.3869
400	9	0.0542	0.1563	0.2161	0.2318	0.2803	0.3129	0.3845
400	25	0.0523	0.1543	0.2161	0.2320	0.2812	0.2974	0.3824
400	100	0.0523	0.1558	0.2106	0.2320	0.2827	0.3027	0.3851

Table A19: SUPP: Maximum of maximum MCSEs of variance of regression coefficients across 500 test runs

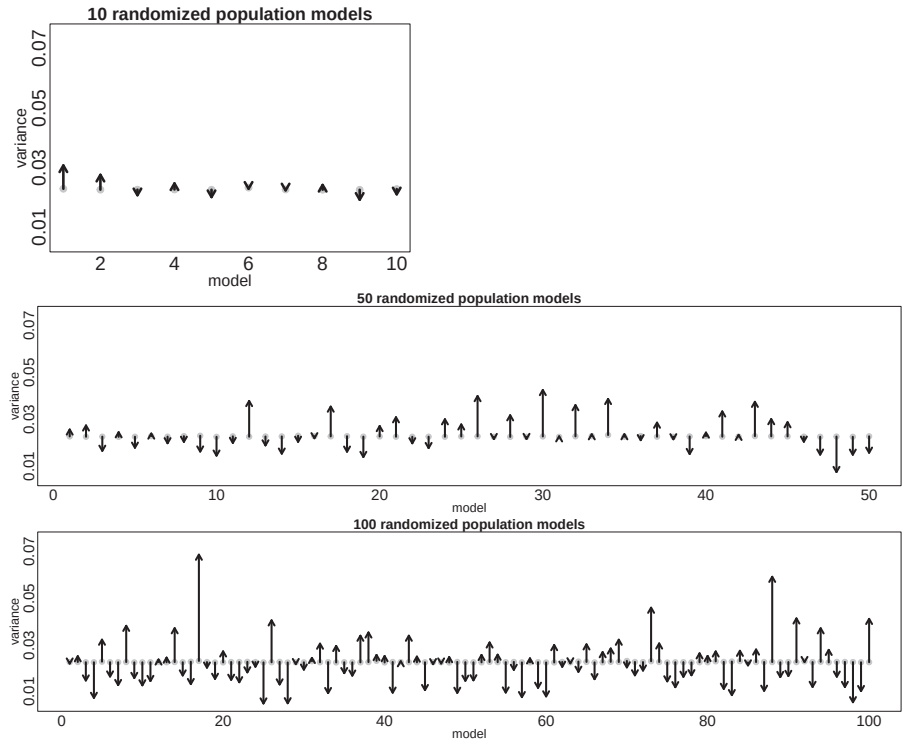
		Randomized Population Models / Replications						
n	σ_ε^2	1/1000	10/100	20/50	25/40	40/25	50/20	100/10
					$\widehat{\beta}_2$			
50	9	0.0516	0.2400	0.3964	0.4591	0.5856	0.5835	0.7150
	25	0.0530	0.2466	0.3812	0.4570	0.5999	0.6061	0.7150
	100	0.0530	0.2312	0.3711	0.4640	0.5838	0.5782	0.7149
75	9	0.0530	0.2343	0.3782	0.4383	0.5772	0.6037	0.7703
	25	0.0530	0.2321	0.3934	0.4493	0.5744	0.6044	0.7710
	100	0.0532	0.2343	0.4079	0.4580	0.5807	0.6044	0.7703
100	9	0.0535	0.2213	0.3938	0.4613	0.5184	0.5967	0.7902
	25	0.0526	0.2234	0.3900	0.4672	0.5175	0.5967	0.7902
	100	0.0553	0.2191	0.3938	0.4672	0.5220	0.5805	0.7902
200	9	0.0552	0.2206	0.3556	0.4236	0.5442	0.5994	0.7647
	25	0.0511	0.2206	0.3386	0.4236	0.5371	0.5994	0.7648
	100	0.0552	0.2105	0.3530	0.4218	0.5287	0.5891	0.7637
400	9	0.0537	0.2208	0.3423	0.4309	0.5045	0.6385	0.7743
	25	0.0517	0.2375	0.3566	0.4309	0.5290	0.6385	0.7743
	100	0.0524	0.2375	0.3527	0.4286	0.5290	0.6355	0.7743
					$\widehat{\beta}_3$			
50	9	0.0551	0.2560	0.3596	0.4100	0.5883	0.6477	0.7268
	25	0.0551	0.2560	0.3442	0.4184	0.5883	0.6297	0.7268
	100	0.0551	0.2560	0.3568	0.3955	0.5516	0.6477	0.7268
75	9	0.0563	0.2433	0.4120	0.4143	0.6030	0.6261	0.7594
	25	0.0537	0.2412	0.4120	0.4103	0.5797	0.6261	0.7638
	100	0.0563	0.2412	0.3498	0.4103	0.5806	0.6009	0.7686
100	9	0.0559	0.2580	0.3845	0.4400	0.5051	0.5974	0.7309
	25	0.0549	0.2271	0.4010	0.4554	0.5061	0.5974	0.7309
	100	0.0559	0.2264	0.4010	0.4477	0.5061	0.5803	0.7307
200	9	0.0530	0.2243	0.3684	0.4264	0.5887	0.6326	0.7024
	25	0.0549	0.2419	0.3336	0.4239	0.5842	0.6332	0.7040
	100	0.0518	0.2334	0.3641	0.4298	0.5929	0.6422	0.7025
400	9	0.0537	0.2155	0.3465	0.4309	0.5455	0.6401	0.7885
	25	0.0542	0.2078	0.3570	0.4279	0.5455	0.6302	0.7914
	100	0.0542	0.2158	0.3465	0.4386	0.5455	0.6447	0.7909

Figure A1: SUPP: Variance of $\widehat{\beta}_0$ for the first test run with sample size 100, error variance = 9.



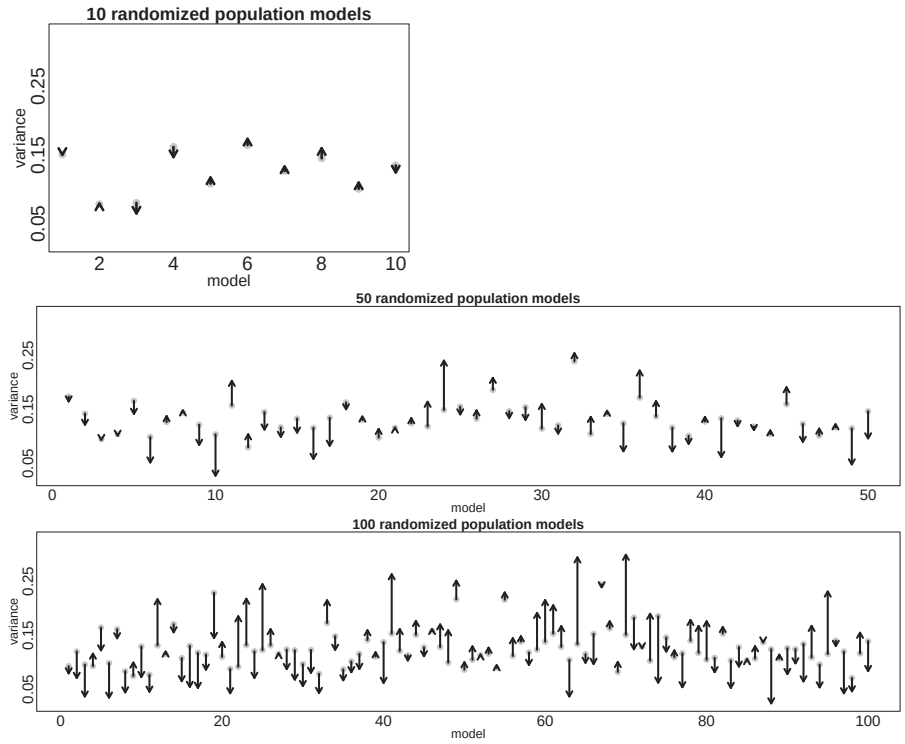
Note. The top plot corresponds to the condition with 10 randomized population models and 100 replications while the middle plot corresponds to the condition with 50 randomized population models and 20 replications. The bottom plot corresponds to the condition with 100 randomized population models and 10 replications. Gray dots denote the population variance of $\widehat{\beta}_0$ and the arrows denote the empirical variance of $\widehat{\beta}_0$.

Figure A2: SUPP: Variance of $\widehat{\beta}_0$ for the first test run with sample size 400, error variance = 9.



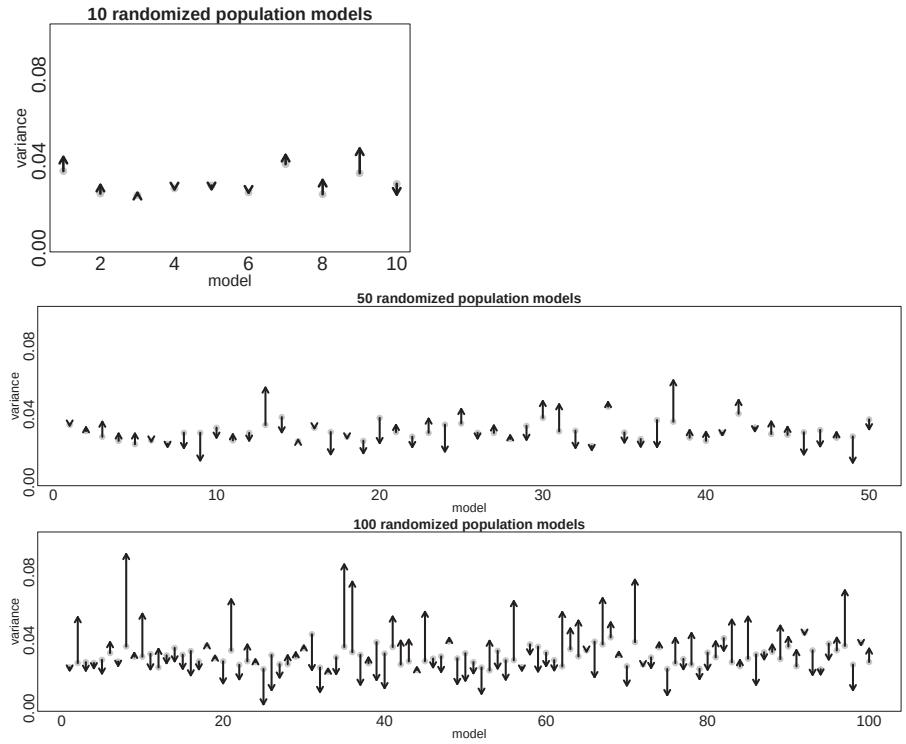
Note. The top plot corresponds to the condition with 10 randomized population models and 100 replications while the middle plot corresponds to the condition with 50 randomized population models and 20 replications. The bottom plot corresponds to the condition with 100 randomized population models and 10 replications. Gray dots denote the population variance of $\widehat{\beta}_0$ and the arrows denote the empirical variance of $\widehat{\beta}_0$.

Figure A3: SUPP: Variance of $\widehat{\beta}_1$ for the first test run with sample size 100, error variance = 9.



Note. The top plot corresponds to the condition with 10 randomized population models and 100 replications while the middle plot corresponds to the condition with 50 randomized population models and 20 replications. The bottom plot corresponds to the condition with 100 randomized population models and 10 replications. Gray dots denote the population variance of $\widehat{\beta}_0$ and the arrows denote the empirical variance of $\widehat{\beta}_0$.

Figure A4: SUPP: Variance of $\widehat{\beta}_1$ for the first test run with sample size 400, error variance = 9.



Note. The top plot corresponds to the condition with 10 randomized population models and 100 replications while the middle plot corresponds to the condition with 50 randomized population models and 20 replications. The bottom plot corresponds to the condition with 100 randomized population models and 10 replications. Gray dots denote the population variance of $\widehat{\beta}_0$ and the arrows denote the empirical variance of $\widehat{\beta}_0$.