Received: 29 November 2017 | Revised: 23 August 2018 | Accepted: 2 November 2018

DOI: 10.1002/sim.8086

TUTORIAL IN BIOSTATISTICS

WILEY Statistics in Medicine

Using simulation studies to evaluate statistical methods

Tim P. Morris[1] | Ian R. White[1] | Michael J. Crowther[2]

[1]London Hub for Trials Methodology Research, MRC Clinical Trials Unit at UCL, London, United Kingdom

[2]Biostatistics Research Group, Department of Health Sciences, University of Leicester, Leicester, United Kingdom

Correspondence
Tim P. Morris, MRC Clinical Trials Unit at UCL, London, United Kingdom.
Email: tim.morris@ucl.ac.uk

Present Address
Tim P. Morris, 90 High Holborn, London WC1V 6LJ, United Kingdom.

Funding information
Medical Research Council, Grant/Award Number: MC_UU_12023/21, MC_UU_12023/29, and MR/P015433/1

Simulation studies are computer experiments that involve creating data by pseudo-random sampling. A key strength of simulation studies is the ability to understand the behavior of statistical methods because some "truth" (usually some parameter/s of interest) is known from the process of generating the data. This allows us to consider properties of methods, such as bias. While widely used, simulation studies are often poorly designed, analyzed, and reported. This tutorial outlines the rationale for using simulation studies and offers guidance for design, execution, analysis, reporting, and presentation. In particular, this tutorial provides a structured approach for planning and reporting simulation studies, which involves defining aims, data-generating mechanisms, estimands, methods, and performance measures ("ADEMP"); coherent terminology for simulation studies; guidance on coding simulation studies; a critical discussion of key performance measures and their estimation; guidance on structuring tabular and graphical presentation of results; and new graphical presentations. With a view to describing recent practice, we review 100 articles taken from Volume 34 of *Statistics in Medicine*, which included at least one simulation study and identify areas for improvement.

KEYWORDS

graphics for simulation, Monte Carlo, simulation design, simulation reporting, simulation studies

1 | INTRODUCTION

Simulation studies are computer experiments that involve creating data by pseudo-random sampling from known probability distributions. They are an invaluable tool for statistical research, particularly for the evaluation of new methods and for the comparison of alternative methods. Simulation studies are much used in the pages of *Statistics in Medicine*, but our experience is that some statisticians lack the necessary understanding to execute a simulation study with confidence, while others are overconfident and so fail to think carefully about design and report results poorly. Proper understanding of simulation studies would enable the former to both run and critically appraise published simulation studies themselves and the latter to conduct simulation studies with greater care and report with transparency. Simulation studies are empirical experiments, and statisticians should therefore use knowledge of experimental design and analysis in running them. As we shall see, inadequacies with design, analysis, and reporting lead to uncritical use and interpretation of simulation studies. In this context, better understanding of the rationale, design, execution, analysis, and reporting of simulation studies is necessary to improve understanding and interpretation of the findings.

Simulation studies are used to obtain empirical results about the performance of statistical methods in certain scenarios, as opposed to more general analytic (algebraic) results, which may cover many scenarios. It is not always possible, or may

be difficult, to obtain analytic results. Simulation studies come into their own when methods make wrong assumptions or data are messy because they can assess the resilience of methods in such situations. This is not always possible with analytic results, where results may apply only when data arise from a specific model.

"Monte Carlo simulation" means statistical techniques that use pseudo-random sampling, and has many uses that are not simulation *studies*. For example, it is required to implement multiple imputation and Markov Chain Monte Carlo methods. The remainder of this paper does not consider these uses, unless the properties of some such method are being evaluated by a simulation study.

There are many ways to use simulation studies in medical statistics. Some examples are:

- To check algebra (and code), or to provide reassurance that no large error has been made, where a new statistical method has been derived mathematically.
- To assess the relevance of large-sample theory approximations (eg, considering the sampling distribution of an estimator) in finite samples.
- For the absolute evaluation of a new or existing statistical method. Often a new method is checked using simulation to ensure it works in the scenarios for which it was designed.
- For comparative evaluation of two or more statistical methods.
- For calculation of sample size or power when designing a study under certain assumptions.[1]

This article is focused primarily on using simulation studies for the evaluation of methods. Simulation studies for this purpose are typically motivated by frequentist theory and used to evaluate the frequentist properties of methods, even if the methods are Bayesian.[2,3]

It seems that as a profession we fail to follow good practice regarding design, analysis, presentation and reporting in our simulation studies, as lamented previously by Hoaglin and Andrews,[4] Hauck and Anderson,[5] Ripley,[6] Burton et al,[7] and Koehler et al.[8] For example, few reports of simulation studies acknowledge that Monte Carlo procedures will give different results when based on a different set of random numbers and hence are subject to uncertainty, yet failing to report measures of uncertainty would be unacceptable in medical research.

There exist some wonderful books on simulation methods in general[6,9,10] and several excellent articles encouraging rigor in specific aspects of simulation studies,[1,4,5,8,11-16] but until now, no unified practical guidance on simulation studies. This tutorial provides such guidance. More specifically, we: introduce a structured approach for planning and reporting simulation studies; provide coherent terminology for simulation studies; offer guidance on coding simulation studies; critically discuss key performance measures and their estimation; make suggestions for structuring tabular and graphical presentation of results; and introduce several new graphical presentations. This guidance should enable practitioners to execute a simulation study for the first time and contains much for more experience practitioners. For reference, the main steps involved, key decisions and recommendations are summarised in Table 1.

The structure of this tutorial is as follows. We describe a review of a sample of the simulation studies reported in *Statistics in Medicine* Volume 34 (Section 2). In Section 3, we outline a systematic approach to planning simulation studies, using the new "ADEMP" structure (which we define there). Section 4 gives guidance on computational considerations for coding simulation studies. In Section 5, we discuss the purposes of various performance measures and their estimation, stressing the importance of estimating and reporting the Monte Carlo standard error (SE) as a measure of uncertainty due to using a finite number of simulation repetitions. Section 6 outlines how to report simulation studies, again using the ADEMP structure, and offers guidance on tabular and graphical presentation of results. Section 7 works through a simple simulation to illustrate in practice the approaches that we are advocating. Section 8 offers some concluding remarks, with a short Section 8.1 that considers some future directions. Examples are drawn from the review and from the authors' areas of interest (which relate mainly to modeling survival data, missing data, meta-analysis, and randomised trial design).

2 | SIMULATION IN PRACTICE: A REVIEW OF *STATISTICS IN MEDICINE*, VOLUME 34

We undertook a review of practice based on articles published in Volume 34 of *Statistics in Medicine* (2015). This review recorded information relevant to the ideas in this article. In this section, we briefly outline the review but do not give results, which instead are provided at relevant points. The raw data on which results are based are provided as a Stata file in the supplementary materials (see file "volume34reviewdata.dta"; pared down, without comments).

TABLE 1 Key steps and decisions in the planning, coding, analysis and reporting of simulation studies

	Section
PLANNING	3
Aims	3.1
· Identify *specific* aims of simulation study.	
Data-generating mechanisms	3.2
· 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	3.3
· Define estimands and/or other targets of the simulation study.	
Methods	3.4
· 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	3.5, 5.2
· 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.	5.2
· Choose a value of n_{sim} that achieves acceptable Monte Carlo SE for key performance measures.	5.2, 5.3
CODING AND EXECUTION	4
· 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.[17]	
ANALYSIS	5
· Conduct exploratory analysis of results, particularly graphical exploration.	
· Compute estimates of performance and Monte Carlo SEs for these estimates.	5.2
REPORTING	6
· 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.	5.2
· Publish code to execute the simulation study including user-written routines.	8

We restricted attention to research articles, excluding tutorials in biostatistics, commentaries, book reviews, corrections, letters to the editor and authors' responses. In the volume, there were a total of 264 research articles of which 199 (75%) included at least one simulation study.

In planning the review, we needed to select a sample size. Most of the questions of interest involved binary answers. For such questions, to estimate proportions with maximum standard error of 0.05 (occurring when the proportion is 0.5), we randomly selected 100 articles that involved a simulation study, before randomly assigning articles to a reviewer. TPM reviewed 35 simulation studies, IRW reviewed 34 and MJC reviewed 31. In case the reviewer was an author or coauthor of the article, the simulation study was swapped with another reviewer. TPM also reviewed five of the simulation studies allocated to each of the other reviewers to check agreement on key information (results on agreement are given in the Appendix and Figure A1).

3 | PLANNING SIMULATION STUDIES USING ADEMP

For clarity about the concepts that will follow, we introduce some notation in Table 2. Note that θ is used to represent a connceptual estimand and its true value.

TABLE 2 Description of notation

θ	An estimand (conceptually); also true value of the estimand
n_{obs}	Sample size of a simulated dataset
n_{sim}	Number of repetitions used; the simulation sample size
$i = 1, \ldots, n_{\text{sim}}$	Indexes the repetitions of the simulation
$\hat{\theta}$	the estimator of θ
$\hat{\theta}_i$	the estimate of θ from the ith repetition
$\bar{\theta}$	the mean of $\hat{\theta}_i$ across repetitions
$\text{Var}(\hat{\theta})$	the true variance of $\hat{\theta}$, which can be estimated with large n_{sim}
$\widehat{\text{Var}}(\hat{\theta}_i)$	an estimate of $\text{Var}(\hat{\theta})$ from the ith repetition
α	the nominal significance level
p_i	the p-value returned by the ith repetition

In the following sections, we outline the ADEMP structured approach to planning simulation studies. This acronym comes from: *Aims*, *Data-generating mechanisms*, *Methods*, *Estimands*, *Performance measures*.

3.1 | Aims

In considering the aims of a simulation study, it is instructive to first consider desirable properties of an estimator $\hat{\theta}$ from a frequentist perspective.

1. $\hat{\theta}$ should be consistent: as $n \to \infty$, $\hat{\theta} \to \theta$. It is also desirable that $\hat{\theta}$ be unbiased for θ in finite samples: $\text{E}(\hat{\theta}) = \theta$ (though arguably less important since unbiasedness is not an invariant property). Some estimators may be consistent but exhibit small-sample bias (logistic regression for example).
2. The sample estimate $\widehat{\text{Var}}(\hat{\theta})$ should be a consistent estimate of the sampling variance of $\hat{\theta}$ (see, for example, the work of Kenward and Roger.[18])
3. Confidence intervals should have the property that at least $100(1 - \alpha)\%$ of intervals contain θ (see Section 5.2).
4. It is desirable that $\text{Var}(\hat{\theta})$ be as small as possible: that $\hat{\theta}$ be an efficient estimator of θ.

There are other properties we might desire, but these tend to involve combinations of the above. For example, short average confidence interval length may be desirable; this relates to (4) and its validity depends on (1), (2), and (3). Mean squared error is a combination of (1) and (4). Further, properties may be traded off; small bias may be accepted if there is a substantial reduction in $\text{Var}(\hat{\theta})$.

The aims of a simulation study will typically be set out in relation to the above properties, depending on what specifically we wish to learn. A simulation study might primarily investigate: large- or small-sample bias (eg, see the work of White[19]); precision, particularly relative to other available methods (eg, see the work of White[20]); Variance estimation (eg, see the work of Hughes et al[21]); or robustness to misspecification (eg, see the work of Morris et al[22]).

There is a distinction between simulation studies that offer a proof-of-concept, ie, showing that a method is viable (or fallible) in some settings, and those that aim to stretch or break methods, ie, identifying settings where the method may fail. Both are useful and important in statistical research. For example, one may be faced with two competing methods of analysis, both of which are equally easy to implement. Even if the choice is unlikely to materially affect the results, it may be useful to have unrealistically extreme data-generating mechanisms to understand when and how each method fails.[22]

Alternatively, it may be of interest to compare methods where some or all methods have been shown to work in principle but the methods under scrutiny were designed to address slightly different problems. They may be put head-to-head in realistic scenarios. This could be to investigate properties when one method is correct – *How badly do others fail?* – or when all are incorrect in some way – *Which is most robust?* No method will be perfect, and it is useful to understand how methods are likely to perform in the sort of scenarios that might be expected in practice. However, such an approach poses tough questions in terms of generating data: *Does the data-generating mechanism favor certain methods over others? How can this be checked and justified?* One common justification is by reference to motivating data. However, in the absence of a broad spectrum of such motivating data, there is a risk of failing to convince readers that a method is fit for general use.

3.2 | Data-generating mechanisms

We use the term "data-generating mechanism" to denote how random numbers are used to generate a dataset. This is in preference to "data-generating *model*," which implies parametric models and so is a specific class of data-generating mechanism. It is not the purpose of this article to explain how specific types of data should be generated. See Ripley[6] or Morgan[9] for methods to simulate data from specific distributions. In planning a simulation study, it is usual to spend more time deciding on data-generating mechanisms than any other element of ADEMP. There are many subtleties and potential pitfalls, some of which we will mention below.

Data may be generated by producing parametric draws from a known model (once or many times), or by repeated resampling with replacement from a specific dataset (where the true data-generating model is unknown). For resampling studies, the true data-generating mechanism is unknown and resamples are used to study the sampling distribution. While parametric simulation can explore many different data-generating mechanisms (which may be completely unrealistic), resampling typically explores only one mechanism (which will be relevant for at least the study at hand).

The choice of data-generating mechanism(s) will depend on the aims. As noted above, we might investigate a method under a simple data-generating mechanism, a realistic mechanism, or a completely unrealistic mechanism designed to stretch a method to breaking point.

Simulation studies provide us with empirical results for specific scenarios. For this reason, simulation studies will often involve more than one data-generating mechanism to ensure coverage of different scenarios. For example, it is very common to vary the sample size of simulated datasets because performance often varies over n_{obs} (see Section 5.2).

Much can be controlled in a simulation study and statistical principles for designing experiments therefore can and should be called on. In particular, there is often more than one factor that will vary across specific data-generating mechanisms. Factors that are frequently varied are sample size (several values) and true parameter values (for example, setting one or more parameters to be zero or nonzero). Varying these factorially is likely to be more informative than one-by-one away from a "base-case" data-generating mechanism, as doing so permits the exploration of interactions between factors. There are however practical implications that might make this infeasible. The first regards presentation of results (covered in Section 6) and the second computational time. If the issue is simply around presentation, it may be preferable to define a "base case" but perform a factorial simulation study anyway, and if results are consistent with no interaction, presentation can vary factors away from the base case one-by-one.

If the main issue with executing a fully factorial design is computational time, it may be necessary for the simulation study to follow a nonfactorial structure. Three approaches are noted below.

A first pragmatic check may be to consider interactions only where main effects exist. If performance seems acceptable and does not vary according to factor A, it would seem unlikely to have chosen a data-generating mechanism that happened to exhibit this property when performance would have been poor for other choices of data-generating mechanism.

A more careful approach could be taken based on making and checking predictions beyond the data-generating mechanisms initially used; an idea similar to external validation. Suppose we have two factors, A and B, where $A \in \{1, \dots, 8\}$ and $B \in \{1, \dots, 5\}$ in the data-generating mechanism. The base-case is $A = 1, B = 1$. If the nonfactorial portion of the design varies A from 1 to 8 holding $B = 1$, and varies B from 1 to 5 holding $A = 1$, this portion of the simulation study could be used to predict performance when $A = 8, B = 5$. Predictions may be purely qualitative ("bias increases as A increases and as B increases, so when we increase both together, we would expect even larger bias"), or quantitative (based on the marginal effects after fitting a model to existing results, thereby producing explicit predictions at unexplored values of A and B). The simulation study can then be re-run for that single data-generating mechanism, say $A = 8, B = 5$ and predictions compared with the empirical results (with a responsibility to expore further when predictions are poor or incorrect).

Finally, a more satisfactory solution is of course to use a fractional factorial design for the data-generating mechanisms.[3,23]

We now issue some specific pitfalls to help readers in choosing data-generating mechanisms (specifically acknowledging Stephen Senn's input).

1. Resampling with replacement from a dataset but failing to appreciate that results are relevant to an infinite population with the exact characteristics of that dataset. For example, if a trial had a nonsignificant result, the treatment effect is nonzero in the implicit population.[24]

2. Missing the distinction between the logical flow of Bayesian and frequentist simulation. Repeated simulation with a single parameter value is explicitly frequentist. The fact that $\hat{\theta}$ is on average equal to θ does not imply that θ is on average equal to $\hat{\theta}$.
3. Failing to distinguish between what the simulator can know and what the estimator can know.[25]
4. Employing tricks in data-generation without appreciating that the resulting data are not what was desired. As an example, suppose one wishes to simulate bivariate data with a desired R^2, say 0.3. For any given repetition, the observed R^2 will not equal 0.3, but this could be fixed by scaling the residuals. This would produce unintended side effects for other statistics.

In our review, 97 simulation studies used some form of parametric model to generate data while three used resampling methods. Of the 97 that simulated from a parametric model, 27 based parameter values on data, one based parameter values partly on data, and the remaining 69 on no data. Of these 97, 91 (94%) provided the parameters used. The most careful example[26] explored analysis of meta-analysis data and drew the design factors from empirical data on 14,886 performed meta-analyses from 1,991 Cochrane Reviews. The total number of data-generating mechanisms per simulation study ranged from 1 to 4.2×10^{10}; Figure A2 (in the Appendix) summarises aspects of the data-generating mechanisms. Where more than one factor was varied, fully factorial designs were the most frequent, while some used partially factorial designs. None used any of the alternative approaches we have described.

3.3 | Estimands and other targets

The majority of simulation studies evaluate or compare methods for estimating one or more population quantities, which we term *estimands* and denote by θ. An estimand is usually a parameter of the data generating model, but is occasionally some other quantity. For example, when fitting regression models with parameter $\boldsymbol{\beta} = (\beta_0 \ \ldots \ \beta_c)$, the estimand may be a specific β, a measure of prognostic ability, the fitted outcome mean, or something else. In order to choose a relevant estimand, it is important to understand the aims of analysis in practice.

The choice of estimand is sometimes a simple matter of stating a parameter of interest. At other times, it is more subtle. For example, a logistic regression model unadjusted for covariates implies a marginal estimand; a model adjusted for covariates implied a conditional estimand with a different true value (this example is expanded on in Section 3.4).

Not all simulation studies evaluate or compare methods that concern an estimand. Other simulation studies evaluate methods for testing a null hypothesis, for selecting a model, or for prediction. We refer to these as *targets* of the simulation study. The same statistical method could be evaluated against multiple targets. For example, the best method to select a regression model to estimate the coefficient of an exposure (targeting an estimand) may differ from the best model for prediction of outcomes (targeting prediction). Where a simulation study evaluates methods for design, rather than analysis, of a biomedical study, the design is the target.

Table 3 summarises different possible targets of a simulation study and suggests some performance measures (described more fully in Section 3.5) that may be relevant for each target, with examples taken from Volume 34.

In our review, 64 simulation studies targeted an estimand, 21 targeted a null hypothesis, eight targeted a selected model, three targeted predictive performance, and four had some other target. Of the 64 targeting an estimand, 51 stated what the estimand was (either in the description of the simulation study or elsewhere in the article). A figure detailing the number of estimands in simulation studies that targeted an estimand is given in the Appendix, Figure A3.

3.4 | Methods

The term "method" is generic. Most often it refers to a model for analysis, but might refer to a design or some procedure (such as a decision rule). For example, Kahan[31] and Campbell and Dean[32] evaluated procedures that involved choosing an analysis based on the result of a preliminary test in the same data.

In some simulation studies, there will be only one method with no comparators. In this case, selecting the method to be evaluated is very simple. When we aim to compare several methods in order to identify the best, it is important to include serious contenders. There are two issues.

First, it is necessary to have knowledge of previous work in the area to understand which methods are and are not serious contenders. Some methods may be legitimately excluded if they have already been shown to be flawed, and it may be unnecessary to include such methods if the only consequences are repetition of previous research and bloating

TABLE 3 Possible targets of a simulation study and relevant performance measures

Statistical Task	Target	Examples of Performance Measures	Example
Analysis			
Estimation	Estimand	Bias, empirical SE, mean-squared error, coverage	Kuss compares a number of existing methods in terms of bias, power, and coverage.[26]
Testing	Null hypothesis	Type I error rate, power	Chaurasia and Harel compare new methods inn terms of type I and II error rates.[27]
Model selection	Model	Correct model rate, sensitivity or specificity for covariate selection	Wu et al compare four new methods in terms of "true positive" and "false positive" rates of covariate selection[28]
Prediction	Prediction/s	Measures of predictive accuracy, calibration, discrimination	Ferrante compares four methods in terms of mean absolute prediction error, etc.[29]
Design			
Design a study	Selected design	Sample size, expected sample size, power/precision	Zhang compares designs across multiple data-generating mechanisms in terms of number of significant test results (described as "gain") and frequency of achieving the (near) optimal design.[30]

of results. An exception might be if a flawed method is used frequently in practice (for example, *last observation carried forward* with incomplete longitudinal data, or the "3 + 3" design for dose-escalation).

Second, code. New methods are sometimes published but not implemented in any software (for example, Robins and Wang[33] and Reiter[34]), implemented poorly, or implemented in unfamiliar software. For methods that have not been implemented, it is hard to argue that they should be included in simulation studies. Although R and Stata appear to dominate for user-written methods, it is not uncommon to see methods implemented in other packages. See Section 4.3 for a discussion of the situation where the methods under consideration are not all implemented in one same package. Note that one important role of simulation is to verify that code is correct.

Methods may also be excluded if they do not target the specified estimand/s. Understanding whether methods target an estimand or not can be subtle. Returning to the example of randomised trial with a binary outcome, one might compare two logistic regression analyses, one unadjusted and one adjusted for a covariate, where the estimand is the log odds ratio for randomised group. In a simulation study, one would be likely to find that the two methods give different mean estimates, and it would be tempting to conclude that at least one of the methods is biased. However, the two methods target different estimands, that is, the true unadjusted and adjusted log odds ratios differ.[35]

One way to tackle the problem of different estimands is to ensure that both methods estimate the same estimand: in the example of the randomised trial using logistic regression, this would involve postprocessing the adjusted regression results to estimate the adjusted marginal odds ratio, which is the same estimand as the unadjusted analysis.[36] This of course implies that the adjustment vs non adjustment is the method comparison we are interested in (it may not be), and that the conditional estimand is simply a nuisance part of standard adjustment. An alternative (but coarser) way to tackle the problem is to target the null hypothesis, if the two methods test the same null. In the logistic regression example described above, because the setting is a randomised trial, the null hypothesis that the odds ratio equals 1 is the same whether the odds ratio is conditional or marginal.

The number of methods evaluated in our review of Volume 34 ranged from 1 to 33 (see Figure A3).

Nonconvergence and other related issues such as perfect prediction ("separation")[37] can blight some simulation studies. In such situations, there is a conceptual issue with defining a method. A "pure" method evaluation would simply assess performance of a model when it converges. However, in practice, an analyst whose model fails to converge would not give up but explore other models until one converges. Thus, evaluation of such a *procedure* may be of interest in simulation studies. Crowther et al evaluated such a procedure[38]: if a model failed to converge, they applied a model with more quadrature points. We will commment further on this issue in Section 5.2.

3.5 | Performance measures

The term "performance measure" describes a numerical quantity used to assess the performance of a method. The equivalent term "operating characteristic" is sometimes used, particularly in the context of study designs (see, for example, the

TABLE 4 Performance measures evaluated in review of Volume 34 (frequency (and %))

Performance Measure	Overall	By Primary Target			
		Estimand ($n = 64$)	Null Hypothesis ($n = 21$)	Selected Model ($n = 8$)	Predictive Performance ($n = 3$)	Other ($n = 4$)
Convergence	**12/85 (14%)**	10/61 (16%)	1/15 (7%)	1/6 (17%)	0/2	0/1
Bias	**63/80 (79%)**	59/64 (92%)	1/9 (11%)	0/2	2/3	1/2
Empirical SE	**31/78 (40%)**	31/62 (50%)	0/9	0/2	0/3	0/2
Mean squared error	**26/78 (33%)**	22/62 (35%)	2/9 (22%)	0/2	1/3	1/2
Model SE	**22/77 (29%)**	21/62 (34%)	1/9 (11%)	0/2	0/3	0/1
Type I error	**31/95 (33%)**	8/62 (13%)	18/21 (86%)	4/6	0/3	1/3
Power	**28/95 (29%)**	8/63 (13%)	14/20 (17%)	4/6	0/3	2/3
Coverage	**42/79 (53%)**	39/63 (62%)	1/9 (11%)	0/2	1/3	1/2
Conf. int. length	**11/80 (14%)**	9/63 (14%)	0/10	0/2	1/3	1/2

Note: Denominator changes across performance measures because not all are applicable in all simulation studies.

work of Royston et al[39]). Statistical methods for estimation may output for example an estimate $\hat{\theta}_i$, an estimate of variance $\widehat{\text{Var}}(\hat{\theta})_i$ (or standard error $\widehat{\text{SE}}(\hat{\theta})_i$), degrees of freedom, confidence intervals, test statistics, and more (such as an estimate of prognostic performance).

The performance measures required in a simulation study depend on the aims and what the study targets (see Section 3.3). When the target is an estimand, the most obvious performance measure to consider is bias: the amount by which $\hat{\theta}$ exceeds θ on average (this can be positive or negative). Precision and coverage of $(1 - \alpha)$ confidence intervals will also be of interest. Meanwhile, if the target is a null hypothesis, power and type I error rates will be of primary interest. A simulation study targeting an estimand may of course also assess power and type I error.

The performance measures seen in our review are summarised in Table 4. The denominator changes according across performance measures because some are not applicable for some simulation studies. Further, sometimes simulation studies had secondary targets. For example, nine simulation studies primarily targeted a null hypothesis but secondarily targeted an estimand and could have assessed bias, and one of these did so. For eight articles, some performance measures were unclear. In some, a performance measure was given a name that its formula demonstrated to be misleading (an example is the term "mean error," which is bias, when the formula is for mean *absolute* error), emphasizing the importance of clear terminology in simulation studies.

Description and estimation of common performance measures of interest are given in Section 5. An important point to appreciate in design and analysis is that simulation studies are empirical experiments, meaning performance measures are themselves estimated, and estimates of performance are thus subject to error. This fundamental feature of simulation studies does not seem to be widely appreciated, as previously noted.[6] The implications are two-fold. First, we should present estimates of uncertainty (quantified as the Monte Carlo standard error; see Section 5.2). Second, we need to consider the number of repetitions n_{sim} and how this can be chosen (see Section 5.2).

4 | COMPUTATIONAL AND PROGRAMMING ISSUES IN SIMULATION STUDIES

In this section, we discuss consideration when coding a simulation study. It is useful to understand what sort of data are involved. There may be up to four classes of dataset, listed and described in Table 5.

4.1 | Random numbers: setting seeds and storing states

All statistical packages capable of Monte Carlo simulation use a pseudo-random-number generator. Each random number is a deterministic function of the current "*state*" of the random-number generator. After a random number is produced,

TABLE 5 The different datasets that may be involved in a simulation study

Dataset	Description and Notes
Simulated	A dataset of size n_{obs} produced with some element of random-number generation, to which one or more methods are applied to produce some quantity relating to the *target* of the study, such as an estimate of θ.
Estimates[a]	Dataset containing n_{sim} summaries of information from repetitions (eg, $\hat{\theta}$, $\widehat{SE}(\hat{\theta})$, indication of hypothesis rejection) for each combination of data-generating mechanism, method, and target (eg, each estimand).
States	Dataset of containing $n_{sim} + 1$ random-number-generator states: the start state for each simulated dataset and the final state (see Section 4.1).
Performance measures	Dataset containing estimated performance and Monte Carlo standard errors for each data-generating mechanism, method and target.

[a]or corresponding summaries for nonestimand targets

the state changes, ready to produce the next random number. Because the function is deterministic, the state can be set. Typically, the state is set using a "*seed*." Seeds do not necessarily map 1:1 to states and provide doors onto the path of possible states. After enough random-number draws (a very large number in software using modern pseudo-random-number generators), the state will eventually repeat: the path is circular.

The "pseudo" element to random-number generators is sometimes characterised as negative. This is perhaps an artefact of the fact that some early algorithms provided very poor imitations of random numbers. However, modern-era algorithms such as the Mersenne Twister do not suffer from these problems and can, for simulation purposes, be regarded as truly random when used correctly. The toss of a coin or roll of a die may be regarded as equally deterministic, albeit the result of a complex set of unknown factors that act in an uncontrollable fashion. These are not denigrated with the term "pseudo-random": in statistical teaching, they are often given as the ultimate example of randomness. However, many stage magicians can control the flip of a coin! If a computer pseudo-random number generator is sufficiently unpredictable and passes the various tests for randomness, it is churlish to regard the "pseudo" aspect as a weakness.

There are several *positive* implications of using a deterministic and reproducible process for generating random numbers. First, if the number of repetitions is regarded as insufficient, the simulation study can continue from its end state. Second and more importantly, if a certain repetition results in some failure such as nonconvergence, the starting state for that repetition can be noted and the repetition re-run under that state, enabling better understanding of when the method does not work so that issues leading to nonconvergence can be tackled. Finally, the whole simulation study can be independently run by other researchers, giving the potential for exact (rather than approximate) reproduction of results and the scope for additional methods to be included.

Our practical advice for utilizing the deterministic nature of random-number generators is simple but strong: (1) *set the seed at the beginning, once and only once*; (2) *store the state of the random-number generator often* (ideally once at the beginning of each repetition and once *following* repetition $i = n_{sim}$). This is important; the following chunk of pseudocode demonstrates the concept:

```
SET RandomSeed to #
FOR Repetition 1 to n_sim by 1
 STORE Repetition and RandomNumberState in StatesData[Repetition]
 GENERATE simulated dataset
...
END FOR
STORE n_sim+1 and RandomNumberState in StatesData[n_sim+1]
```

The reason for this advice is to avoid unintended dependence between simulated datasets. We will illustrate our caution: one undesirable method of knowing the states for n_{sim} repetitions is to set an initial seed and generate a single vector of length n_{sim} by recording the starting state, generating a single random number, recording the new state, and so on. For the simulation itself, the seed for the ith repetition is then set to the ith element. To clarify the problem, let $n_{obs} = n_{sim} = 4$ and let the first simulation step be generation of vector x from a Uniform(0,1) distribution. The first repetition simulates x_1 (which changes the random number state four times) and proceeds. The second repetition then simulates x_2, which is

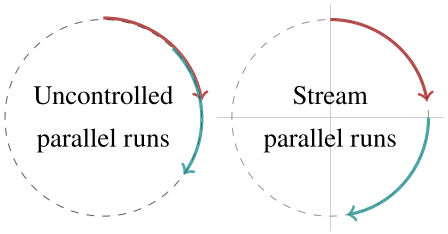
made up of observations 2 to 4 from repetition $i = 1$ and just one new value. Run in Stata 15 (see supplementary material, ie, file "corrstates.do"), the resulting draws of x for the four repetitions are:

$$x_1 = (0.1338766, 0.1364070, 0.4512149, \mathbf{0.0210242})$$
$$x_2 = (0.1364070, 0.4512149, \mathbf{0.0210242}, 0.3508981)$$
$$x_3 = (0.4512149, \mathbf{0.0210242}, 0.3508981, 0.9113581)$$
$$x_4 = (\mathbf{0.0210242}, 0.3508981, 0.9113581, 0.4707521).$$

Note that elements with the same shading contain the same values across rows. The fourth element of x_1 is the first element of x_4 and appears in all repetitions. Only when $i > n_{obs}$ is the draw of x actually independent of the first repetition. Such dependency in simulated data can compromise both performance estimates and Monte Carlo SEs and must be avoided.

4.1.1 | "Stream" random numbers

It is common for parts of simulation studies – fractions of all the repetitions, for example – to be run in parallel on different cores of high-performance computers (which this article will not mention further). If the advice to set the seed once only is followed, the implication for parallelisation is that, while runs for *different* data-generating mechanisms may be parallelised, it is inadvisable to parallelise repetitions *within* a specific data-generating mechanism.



Suppose we wish to parallelise two sets of $n_{sim}/2$ repetitions. Any simulation study will use random numbers (in order) from a section of the circle. Here, each set of repetitions is represented by a clockwise arrow, and uses 80° of the total 360° of random numbers available in the full circle (a caricature for illustrative purposes; in practice, a much smaller fraction would be used). The seed dictates the position on the circle at which an arrow begins (and thus ends). The random numbers used up by the first $n_{sim}/2$ repetitions are represented by the red arrow and for the second n_{sim} by the blue arrow. The left circle depicts two chunks run in parallel with two different, arbitrarily-chosen starting seeds. By chance, they may overlap as seen. This would be a cause for concern. The right circle uses separate streams of random numbers. This breaks the circle into quadrants, and setting the same value of a seed within a stream means that the separate chunks will start at the equivalent point on the quadrants and here there is no chance that one stream will enter another. In the absence of streams, repetitions should not be parallelised for the same data-generating mechanism.

In Stata (version 15 or newer), the stream is set with

```
. set rngstream #
```

prior to setting the seed. In SAS, it is achieved within a `data` step with

```
. call stream(#);
```

In R, this can be achieved with the `rstream` package. Regardless of the package, the same seed must be used within different values of #.

When a simulation study uses multiple data-generating mechanisms, these may be run in parallel. Because performance is typically estimated separately for different data-generating mechanisms, using the same seeds is less of a problem (and may in fact be advantageous, as described in Section 5.4).

Many programs execute methods involving some stochastic element. Examples include multiple imputation, the bootstrap, the g-computation formula, multistate models, and Bayesian methods that use Markov Chain Monte Carlo. Commands to implement these methods involve some random-number generation. It is important to check that such programs do not manipulate the seed. Some packages do have a default seed if not input by the user. If they do set the seed internally, many of the n_{sim} results will be highly correlated, if not identical, and results should not then be trusted. Checking for such behavior is worthwhile. One simple technique is to display the current state of the random-number generator, twice issue the command, and display the state after each run. If the first and second states are the same, then

the program probably does not use random numbers. If the first and second states differ but the second and third do not, the seed is being reset by the program.

4.2 | Start small and build up code

As with any coding task it is all-too-easy to obtain misleading results in a simulation study through very minor coding errors; see, for example, the comments section of Bartlett,[40] where fixing an error in a single line of code completely changed the results. A function may be sloppily written as `a-b*c` such that it is unclear if `(a-b)*c` or `a-(b*c)` was intended; a machine will interpret this code but will not discern the intention.

Errors are often detected when results are unexpected: for example, when bias appears much greater than theory suggests. One design implication is that methods with known properties should be included where possible as a check that these properties are exhibited. One straightforward and intuitive approach for minimizing errors is to start small and specific for one repetition, then build and generalise, including plenty of built-in checks.

In a simulation study with $n_{\text{sim}} > 1$ and several simulated variables, a good starting point is to generate one simulated dataset with large n_{obs}. If variables are being generated separately then the code for each should be added one by one and the generated data explored to (1) check that the code behaves as expected and (2) ensure the data have the desired characteristics. For example, Stata's `rnormal(m,s)` function simulates normal variates with mean`m`and standard deviation`s`. The usual notation for a normal distribution uses a mean and *variance*. We have seen this syntax trip up several good programmers. By checking the standard deviation of a variable simulated by `rnormal()` in a single large simulated dataset, it should be obvious if it does not behave in the expected fashion. The simulation file should be built to include different data-generating mechanisms, methods, or estimands, again checking that behavior is as expected. Using the above example again, if the basic data-generating mechanism used $\text{N}(\mu, 1)$, the issue with specifying standard deviations vs variances would not be detected, but it would for data-generating mechanisms with $\sigma^2 \neq 1$. When satisfied with the large dataset being generated, we apply each method.

Once satisfied that one large run is behaving sensibly, it is worth setting the required n_{obs} for the simulation study and exploring the simulated datasets produced under a handful of different seeds. When satisfied that the program still behaves sensibly, it may be worth running a few (say tens of) repetitions. If, for example, convergence problems are anticipated, or bias is expected to be 0, this can be checked informally without the full set of simulations.

After thoroughly checking through and generalizing code, the full set of n_{sim} repetitions may be run. However, recall the precaution in Section 4.1 to store the states of the random-number generator and the reasons. If failure occurs in repetition 4120 of 5000, we will want to understand why. In this case, a record of the 4120th start state means we can reproduce the problematic dataset quickly.

While the ability to reproduce specific errors is useful, it is also practically helpful to be able to continue even when an error occurs. For this purpose, we direct readers to the `capture` command in Stata and the `try` command in R. The failed analysis must be recorded as a missing value in the Estimates dataset, together with reasons if possible.

4.3 | Using different software packages for different methods

It is frequently the case that competing methods are implemented in different software packages, and it would be more burdensome to try and code them all in one package than to implement them in different packages. There are two possible solutions. The first is to simulate data separately in the different packages and then use the methods on those data. The second is to simulate data in one package and export simulated data so that different methods are based on the same simulated datasets.

Both approaches are valid in principle, but we advocate the latter. First, if data are generated independently for different methods, there will be different (random) Monte Carlo error affecting each repetition. By using the same simulated data for both comparisons, this Monte Carlo error will affect methods' performance in the same way because methods are matched on the same generated data. Second, it is cumbersome to do a job twice, and because different software packages have different quirks, it will not be easy to ensure data really are being generated identically. Third, it is important to understand that our aim is to compare methods, and while the software implementation may be important to evaluate, the way the software package simulates data is not of interest: using a method in practice would involve a software implementation, but not simulating data using that package. Whatever data an analyst was faced with would be the same regardless of the software being used.

Sixty-two simulation studies in our review mentioned software. Table A1 (in the Appendix) describes the specific statistical software mentioned. Seven simulation studies mentioned using more than one statistical package.

5 | ANALYSIS OF ESTIMATES DATA

This section describes estimation for various performance measures along with Monte Carlo SEs. We advocate two preliminaries: checking for missing estimates and plots of the estimates data.

5.1 | Checking the estimates data and preliminaries

The number of missing values, eg, of $\hat{\theta}_i$ and $\widehat{\text{SE}}(\hat{\theta}_i)$ (for example due to nonconvergence), is the first performance measure to assess. The data produced under repetitions for which missing values were returned should be explored to understand how a method failed (see Section 4) and, ideally, the code made more robust to reduce the frequency of failures.

Missing values in the *estimates* dataset pose a missing data problem regarding the analysis of other performance measures. It seems implausible that values would be missing completely at random[41]; estimates will usually be missing due to nonconvergence so will likely depend on some characteristic/s of a given simulated dataset. When the "method" being evaluated involves an analyst's procedure (as described in Section 3.4), for example, the model changes if the first-choice model does not converge, this can reduce or remove missing values from the estimates data (though it changes the nature of the method being evaluated; see Section 3.4).

If more than two methods are evaluated, and one always returns an estimate $\hat{\theta}_i$, then missing values for another method may be related to the returned values for the first method. In the presence of a nontrivial proportion of missing estimates data, analysis of further performance measures should be tentative, particularly when comparing methods with different numbers of $\hat{\theta}_i$ missing. "Nontrivial" means any proportion that could meaningfully alter estimated performance. If we are interested in detecting tiny biases, even 1% may be nontrivial.

Before undertaking a formal analysis of the estimates dataset, it is sensible to undertake some exploratory analysis. Plots are often helpful here. For example, Kahan[31] assessed the performance of a two-stage procedure for the analysis of factorial trials. The procedure was unbiased (both conditionally and unconditionally), yet a histogram of $\hat{\theta}_i$ exhibited a bimodal distribution with modes equally spaced at either side of θ, with almost no values of $\hat{\theta}_i$ close to θ. This may cause concern and would have been missed had the analysis proceeded straight to the estimation of performance.

For simulation studies targeting an estimand, the following plots are often informative:

1. A univariate plot of the distribution of $\hat{\theta}_i$ and $\widehat{\text{SE}}(\hat{\theta}_i)$ for each data-generating mechanism, estimand and method, to inspect the distribution and, in particular, to look for outliers.
2. A bivariate plot of $\widehat{\text{SE}}(\hat{\theta}_i)$ vs $\hat{\theta}_i$ for each data-generating mechanism, estimand and method, with the aim of identifying bivariate outliers.
3. Bivariate plots of $\hat{\theta}_i$ (and possibly $\widehat{\text{SE}}(\hat{\theta}_i)$) for one method vs another for each data-generating mechanism and estimand. The purpose here is to look for correlations between methods and any systematic differences. Where more than two methods are compared, a graph of every method vs every other or vs the comparator can be useful.
4. Limits-of-agreement for $\hat{\theta}_i$ (and possibly $\widehat{\text{SE}}(\hat{\theta}_i)$) compared with a reference method. That is, a plot of the difference vs the mean of each method compared with a comparator.
5. A plot of confidence intervals fractionally ranked by $|z_i|$, where $z_i = (\hat{\theta}_i - \theta)/\text{ModSE}_i$ (as in Figure 5). This is called a zip plot and is a means of understanding any issues with coverage.

These plots will be demonstrated and interpreted in the worked example (Section 7).

5.2 | Estimation of performance and monte carlo standard errors for some common performance measures

This section outlines some common performance measures, properties they are designed to assess, how they are estimated and how Monte Carlo standard errors are computed. We suppress the "hat" notation for performance measures, but emphasise that these are estimates.

For interpretation of results, performance measures should usually be considered jointly (one could prefer a method with zero variance by conveniently ignoring bias).

TABLE 6 Performance measures: definitions, estimates and Monte Carlo standard errors

Performance Measure	Definition	Estimate	Monte Carlo SE of Estimate
Bias	$E[\hat{\theta}] - \theta$	$\frac{1}{n_{sim}}\sum_{i=1}^{n_{sim}}\hat{\theta}_i - \theta$	$\sqrt{\frac{1}{n_{sim}(n_{sim}-1)}\sum_{i=1}^{n_{sim}}(\hat{\theta}_i - \bar{\theta})^2}$
EmpSE	$\sqrt{Var(\hat{\theta})}$	$\sqrt{\frac{1}{n_{sim}-1}\sum_{i=1}^{n_{sim}}(\hat{\theta}_i - \bar{\theta})^2}$	$\frac{\widehat{EmpSE}}{\sqrt{2(n_{sim}-1)}}$
Relative % increase in precision (B vs A) [a]	$100\left(\frac{Var(\hat{\theta}_A)}{Var(\hat{\theta}_B)} - 1\right)$	$100\left(\left(\frac{\widehat{EmpSE}_A}{\widehat{EmpSE}_B}\right)^2 - 1\right)$	$200\left(\frac{\widehat{EmpSE}_A}{\widehat{EmpSE}_B}\right)^2\sqrt{\frac{1-Corr(\hat{\theta}_A,\hat{\theta}_B)^2}{n_{sim}-1}}$
MSE	$E[(\hat{\theta} - \theta)^2]$	$\frac{1}{n_{sim}}\sum_{i=1}^{n_{sim}}(\hat{\theta}_i - \theta)^2$	$\sqrt{\frac{\sum_{i=1}^{n_{sim}}\left[(\hat{\theta}_i-\theta)^2-\widehat{MSE}\right]^2}{n_{sim}(n_{sim}-1)}}$
Average ModSE[a]	$\sqrt{E[\widehat{Var}(\hat{\theta})]}$	$\sqrt{\frac{1}{n_{sim}}\sum_{i=1}^{n_{sim}}\widehat{Var}(\hat{\theta}_i)}$	$\sqrt{\frac{\widehat{Var}[\widehat{Var}(\hat{\theta})]}{4n_{sim}\times\widehat{ModSE}^2}}$ [b]
Relative % error in ModSE[a]	$100\left(\frac{ModSE}{EmpSE} - 1\right)$	$100\left(\frac{\widehat{ModSE}}{\widehat{EmpSE}} - 1\right)$	$100\left(\frac{\widehat{ModSE}}{\widehat{EmpSE}}\right)\sqrt{\frac{\widehat{Var}[\widehat{Var}(\hat{\theta})]}{4n_{sim}\times\widehat{ModSE}^4} + \frac{1}{2(n-1)}}$ [b]
Coverage	$Pr(\hat{\theta}_{low} \le \theta \le \hat{\theta}_{upp})$	$\frac{1}{n_{sim}}\sum_{i=1}^{n_{sim}}1(\hat{\theta}_{low,i} \le \theta \le \hat{\theta}_{upp,i})$	$\sqrt{\frac{\widehat{Cover.}\times(1-\widehat{Cover.})}{n_{sim}}}$
Bias-eliminated coverage	$Pr(\hat{\theta}_{low} \le \bar{\theta} \le \hat{\theta}_{upp})$	$\frac{1}{n_{sim}}\sum_{i=1}^{n_{sim}}1(\hat{\theta}_{low,i} \le \bar{\theta} \le \hat{\theta}_{upp,i})$	$\sqrt{\frac{\widehat{B\text{-}E\ Cover.}\times(1-\widehat{B\text{-}E\ Cover.})}{n_{sim}}}$
Rejection % (power or type I error)	$Pr(p_i \le \alpha)$	$\frac{1}{n_{sim}}\sum_{i=1}^{n_{sim}}1(p_i \le \alpha)$	$\sqrt{\frac{\widehat{Power}\times(1-\widehat{Power})}{n_{sim}}}$

[a] Monte Carlo SEs are approximate for *Relative % increase in precision*, *Average ModSE*, and *Relative % error in ModSE*.

[b] $\widehat{Var}[\widehat{Var}(\hat{\theta})] = \frac{1}{n_{sim}-1}\sum_{i=1}^{n_{sim}}\{\widehat{Var}(\hat{\theta}_i) - \frac{1}{n_{sim}}\sum_{j=1}^{n_{sim}}\widehat{Var}(\hat{\theta}_j)\}^2$.

Monte Carlo standard errors quantify simulation uncertainty: they provide an estimate of the SE of (estimated) performance due to using finite n_{sim}. The Monte Carlo SE targets the sampling distribution of repeatedly running the same simulation study (with n_{sim} repetitions) under different random-number seeds.

In our review of simulation studies in *Statistics in Medicine* Volume 34, 93 did not mention Monte Carlo SEs for estimated performance. The formulas for computing Monte Carlo SEs given in Table 6 with description and comments in the text. For empirical SE, relative % increase in precision, and relative error, the Monte Carlo SE formulas assume normally distributed $\hat{\theta}$; for non-normal $\hat{\theta}$, robust SEs exist; see White and Carlin.[42]

Bias is frequently of central interest, and quantifies whether a method targets θ on average. Frequentist theory holds unbiasedness to be a key property.

The mean of $\hat{\theta}_i$, $\bar{\theta}$, is often reported instead. This is estimated in the same way but without subtracting the constant θ, and so has the same Monte Carlo SE. It is sometimes preferable to report the relative bias, rather than absolute. If different values of θ are used for different data-generating mechanisms then relative bias permits a more straightforward comparison across values. However, relative bias can be used only for $|\theta| > 0$. The absence of bias is one property of an estimator; while it is often of central interest, we may sometimes accept small biases because of other good properties.

The empirical SE is a measure of the precision or efficiency of the estimator of θ. It depends only on $\hat{\theta}_i$ and does not require knowledge of θ. The empirical SE estimates the long-run standard deviation of $\hat{\theta}_i$ over the n_{sim} repetitions. Several other designations are in common use; in our review, the terms used included "empirical standard deviation," "Monte Carlo standard deviation," "observed SE," and "sampling SE."

The empirical standard error can be hard to interpret for a single method (unless compared to a lower bound), and the relative precision is often of interest when comparing methods.

Note that, if either method is biased, relative precision should be interpreted with caution because an estimator that is biased towards the null can have small empirical SE as a result of the bias: $\hat{\theta}_i/2$ has smaller empirical SE than $\hat{\theta}_i$.

A related measure, which also takes the true value of θ into account, is the mean squared error (MSE). The MSE is the sum of the squared bias and variance of $\hat{\theta}$. This appears a natural way to integrate both measures into one summary performance measure (low variance is penalised for bias), but we caution that, for method comparisons, the relative influence of

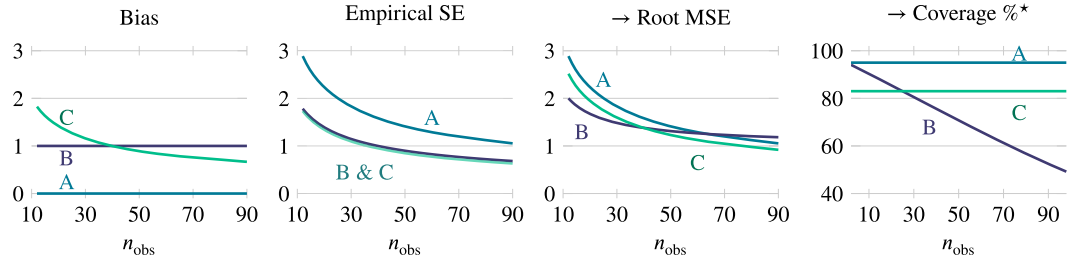


FIGURE 1 The impacts of bias and empirical SE on root MSE and coverage of nominal 95% confidence intervals, compared for three methods: Method A is unbiased but imprecise; Method B is biased (independent of n_{obs}) and more precise; Method C is biased (with bias $\propto \sqrt{1/n_{obs}}$) and the same precision as method B. The comparison of root MSE and coverage depends on the choice of n_{obs}; the constant bias of method B dominates its increasingly poor MSE and coverage as n_{obs} increases [Colour figure can be viewed at wileyonlinelibrary.com]

bias and of variance on the MSE tends to vary with n_{obs} (except when all methods are unbiased), making generalisation of results difficult. This issue is illustrated in the first three panels of Figure 1, which depict the bias, empirical standard error and root MSE (favored here because it is on the same scale as Empirical SE) for three hypothetical methods. Method A is unbiased but imprecise (and so root MSE is simply the empirical SE); method B is biased, with bias constant over n_{obs}, but more precise (as is often the case with biased methods, see, for example, White and Royston[43]); and method C, which is biased in a different way (bias $\propto \sqrt{1/n_{obs}}$) and with precision the same as method B. For $n_{obs} < 60$, root MSE is lower for method B than A, but for $n_{obs} > 60$, root MSE is lower for method A. The lesson is that comparisons of (root) MSE are more sensitive to choice of n_{obs} than comparisons of bias or empirical SE alone. MSE is nonetheless an important performance measure, particularly when the aims of a method relate to prediction rather than estimation, but the implication is that, when MSE is a performance measure, data-generating mechanisms should include a range of values of n_{obs}.

We term the root-mean of the squared model SEs the "average model SE." The aim of the model SE is that $E(ModSE^2) = EmpSE^2$. The model SE explicitly targets the empirical SE. If it systematically misses, this represents a bias in the estimation of model SE. The relative error in average model SE is then an informative performance measure (some prefer the ratio of average model SE to empirical SE).

Coverage of confidence intervals is a key property for the long-run frequentist behavior of an estimator. It is defined as the probability that a confidence interval contains θ.

Note that Neyman's original description of confidence intervals defined the property of *randomisation validity* as exactly $100(1-\alpha)\%$ of intervals containing θ.[44-46] *Confidence validity* is the property that the true percentage is at least $100(1-\alpha)$. This latter definition is less well known than the former, with the result that overcoverage and undercoverage are sometimes regarded as similarly bad.[47] Of course, randomisation validity would usually be preferred over confidence validity because it implies shorter intervals, but this is not always the case. There are examples of procedures that return both shorter intervals and higher coverage.[45,46]

Undercoverage is to be expected if, for example, (i) Bias $\neq 0$, (ii) ModSE $<$ EmpSE, (iii) the distribution of $\hat{\theta}$ is not normal and intervals have been constructed assuming normality, or (iv) $\widehat{Var}(\hat{\theta}_i)$ is too variable. Over-coverage tends to occur as a result of $ModSE^2 > EmpSE^2$. This may occur either in the absence or presence of issues (i) and (iii).

Undercoverage due to bias will tend to deteriorate as n_{obs} increases (unless bias reduces at or faster than a rate of $\sqrt{1/n_{obs}}$). Intuitively, as n_{obs} increases, confidence intervals zero-in on the wrong value. The situation is illustrated in the fourth panel of Figure 1 (with some loss of generality). Coverage was calculated for normal-based confidence intervals with correct interval width such that bias is the only source of under-coverage. Method B, for which bias is independent of n_{obs}, has deteriorating coverage as n_{obs} increases. Method C, for which bias $\propto \sqrt{1/n_{obs}}$, has constant undercoverage. As for MSE, bias dominates as n_{obs} increases. The implication is that, if coverage is being assessed in the presence of bias, the data-generating mechanisms should include a range of values of n_{obs}.

As noted previously, there are two sources of poor coverage: bias leads to undercoverage, while incorrect interval width (for example, because average ModSE $\neq$ EmpSE) may produce undercoverage or overcoverage. We propose a decomposition of poor coverage with a new performance measure: "bias-eliminated coverage." By studying confidence interval coverage for $\bar{\theta}$ rather than for θ, the bias of a method is eliminated from the calculation of coverage. We emphasise that bias-eliminated coverage should not be regarded as a performance measure in its own right. It is to be used for understanding how coverage performance is influenced by bias vs width of confidence intervals. It is obvious that, for the methods in Figure 1, bias-eliminated coverage will be equal for all methods.

TABLE 7 Coverage conditional on size of ModSE

Approach	n_{sim} Analyzed	Coverage (Monte Carlo SE)
All observations	30,000	95.0%(0.1%)
Conditional: ModSE in highest third	10,000	98.0%(0.1%)
Conditional: ModSE in middle third	10,000	95.5% (0.2%)
Conditional: ModSE in lowest third	10,000	91.5% (0.3%)

Rejection rates – power and type I error – are often of principal interest in simulation studies that target a null hypothesis, and power is of particular interest when competing designs are being compared by simulation. Assume we have p-values p_i in the estimates data and are considering nominal significance level α. The p-values may be derived from a Wald statistic $\frac{\hat{\theta}_i}{\widehat{\text{SE}}(\hat{\theta}_i)}$ or output directly, for example by a likelihood-ratio test. An appropriate test would reject a proportion α of the n_{sim} repetitions when the null is true and as often as possible when it is false. The obvious warning is that if the test does not control type I error at level α, power should be interpreted with caution.

It is sometimes (not always) of interest to estimate *conditional* performance. This is particularly true for simulation studies that aim to evaluate alternative study designs, for example, where design decisions are made based on the early data. Two simulation studies in our review sample explored two-stage procedures in randomised trials, where the estimand is selected after the first stage: the estimand was the treatment effect in a selected subpopulation[48] or the effect of a selected treatment.[49] In both cases, estimators were designed to be conditionally unbiased. Kimani et al reported bias conditional on each possible selection of estimand,[48] while Carreras et al considered the bias averaged across estimands.[49] The former method is stricter and arguably more appropriate since, having selected an estimand, the observer is not interested in the other case.[50]

Now, consider the following form of conditional performance: $n_{obs} = 30$ observations are simulated from $y \sim N(\mu, \sigma^2)$, with $\mu = 0$, $\sigma^2 = 1$. For each repetition, 95% confidence intervals for μ are constructed using the t-distribution. The process is repeated $n_{sim} = 30{,}000$ times, and we study the coverage, (1) for all repetitions and (2) according to tertiles of the model SE. The results, given in Table 7, show that coverage is below 95% for the lowest third of standard errors, above 95% for the highest third, and slightly above for the middle third. Poor conditional performance in this sense should not cause concern. Further, it is unhelpful for an analyst faced with a dataset: one would not in practice know in which tertile of possible model SEs a particular model SE lies.

Estimating performance conditional on *true* (rather than sample-estimated) parameters that vary across data-generating mechanisms is where methods *should* be expected to provide good performance, and we do not recommend averaging over these, as is done informally in Gutman and Rubin.[51]

We have described the most commonly reported and generally applicable performance measures, particularly when a simulation study targets an estimand. There are others that are sometimes used (such as the proportion of times the correct dose is selected by dose-finding designs) and others that we have not yet thought of.

5.3 | Sample size for simulation studies

In choosing n_{sim}, the central issue is Monte Carlo error: key performance measures need to be estimated to an acceptable degree of precision.

The values of n_{sim} reported in our review are shown in Figure A2, panel (D). Four simulation studies did not report n_{sim}. Common sample sizes are $n_{sim} = 500$ and $n_{sim} = 1000$, as previously reported by Burton et al.[7] Of the 87 studies reporting n_{sim}, four provided any justification of the choice. These were:

- "*To evaluate the asymptotic biases*"[52]
- "*errors can be reduced by the large number of simulation replicates*"[53]
- "*number was determined mainly to keep computing time within a reasonable limit. A reviewer pointed out that, as an additional justification, by using 10,000 meta-analyses the standard error of an estimated percentage (eg, for the empirical coverage) is guaranteed to be smaller than 0.5.*"[26]
- Most positively, Marozzi gave an explicit derivation of Monte Carlo SE.[54]

Clearly, this is a suboptimal state of affairs. For some more concrete justifications, see the worked illustrative example in Section 7 by Keogh and Morris[55] or Morris et al.[56]

There exist situations where only one repetition is necessary, particularly when investigating large-sample bias.[19] Here, the aim was to demonstrate large-sample bias of an estimator and the single estimate of $\hat{\theta}$ was many model standard errors from its true value.

Where the key performance measure is coverage, n_{sim} can be defined as follows. The Monte Carlo SE of coverage is given in Section 5.2.

Plugging in the expected coverage (for example 95%) and rearranging, we get

$$n_{\text{sim}} = \frac{\text{E(Coverage)} \times (1 - \text{E(Coverage)})}{(\text{Monte Carlo SE}_{\text{req}})^2} \tag{1}$$

with a similar expression if n_{sim} is to be determined based on power. For example, if the SE required for a coverage of 95% is 0.5%,

$$n_{\text{sim}} = \frac{95 \times 5}{0.5^2} = 1900 \text{ repetitions.}$$

Coverage is estimated from n_{sim} binary summaries of the repetitions, so the worst-case SE occurs when coverage is 50%. In this scenario, to keep the required Monte Carlo SE below 0.5%, (1) says that n_{sim} = 10,000 repetitions will achieve this Monte Carlo SE.

A convenient feature of simulation studies is that the Monte Carlo SE can be assessed and n_{sim} increased much more cheaply than with other empirical studies. The cost is computational time. To continue, rather than start again, it is important to have a record the end state of the random-number generator (which can be used as the seed if further repetitions are added) or to use a different stream.

5.4 | Remarks on analysis

We have emphasised repeatedly that simulation studies are empirical experiments. In many biomedical experiments, "controls" are used as a benchmark and the estimated effects of other conditions are estimated as a contrast vs control. However, simulation studies often benefit from having a known "truth," meaning that the contrast vs a control is not often of interest (hence the term "comparator" in Section 3.4). That is, bias need not be estimated as the *difference* between $\bar{\theta} - \theta$ for method A and $\bar{\theta}$ for the control; rather the bias for a method stands alone, being computed against θ, the comparator of interest. There are benchmarks for other performance measures as well, such as coverage (the nominal %) and precision (the Cramér-Rao lower bound[57,58]).

In some cases, the true value of θ is unknown: it may not appear in the data-generating mechanism. If performance measures involving θ are not of interest, this poses no problem. Otherwise, one solution is to *estimate* θ by simulation. Williamson et al simulated data from a logistic model, but θ was not the conditional odds ratio used to generate data; θ was the marginal odds ratio, risk ratio and risk difference.[59] They thus estimated θ for each of these estimands from a large simulated dataset.

In our review of Volume 34, of 74 studies that included some θ, nine estimated it, 57 used a known θ and 8 were unclear. Estimating θ is in our view a sensible and pragmatic approach. However, such an approach must simulate a dataset so large that it is fair to assume that the variance of "θ" is negligible, particularly compared to that of $\bar{\theta}$, and ensure that the states of the random-number generators used in the simulation study do not overlap with the states used for the purpose of estimating θ. In practice, the way to do this is either to use a separate stream for the random numbers, or to run the θ-estimation simulation immediately before the main run.

The estimation of performance measures in Section 5.2 is described for estimating performance once per data-generating-mechanism. This is most suited to simulation studies with few data-generating mechanisms, but many simulation studies are considerably more complex. In such cases, it is natural to fit a model (termed "meta-model" by Skrondal[23]) for performance in terms of the data-generating mechanisms.

An advantage of modeling performance across data-generating mechanisms is that we are able to match repetitions. This reduces the Monte Carlo SE for the comparison of methods. For example, suppose that we have two data-generating mechanisms with θ equal to 1 and 2. We could use the same starting seed so that results are correlated within *i*.

6 | REPORTING

6.1 | The "methods" section

The rationale for the ordering of elements in ADEMP is that this is usually the appropriate order to report them in a methods section. If the simulation study has been planned and written out before it is executed, the methods section is largely written. This is a particularly helpful ordering for other researchers who might wish to replicate the study. Details should be included to allow reproduction as far as possible, such as the value of n_{sim} and how this was decided on, dependence among simulated datasets. Another important element to report is a justification of the chosen targets for particular applied contexts.

6.2 | Presentation of results

Some simulation studies can be very small, for example, exploring one or two performance measures under a single data-generating mechanism. These can be reported in text (as in He et al[60]). In other cases, there are enough results that it becomes necessary to report them in tabular or graphical form. For any tabulation or plot of results, there are four potential dimensions: data generating mechanisms, methods, estimands, and performance measures. This section provides some considerations for presenting these results.

In tabular displays, it is common to divide rows according to data-generating mechanisms and methods as columns (as in Chen et al[61]), though if there are more methods than data-generating mechanisms it is better to swap these (as in Hsu et al[62]). Performance measures and estimands may vary across columns or across rows depending on what makes the table easier to digest (see, for example, Alonso et al[63]).

There are two key considerations in the design of tables. The first is how to place the important comparisons side-by-side. The most important comparisons will typically be of methods, so bias (for example) for different methods should be arranged in adjacent rows or columns.

The second consideration regards presentation of Monte Carlo SEs, and this tends to confound the first. By presenting them next to performance results, for example in parentheses, the table becomes cluttered and hard to digest, obscuring interesting comparisons. For this reason, some authors will report the maximum Monte Carle SE in the caption of tables.[43,64] Results should not be presented to a greater accuracy than is justified by the Monte Carlo SE (eg, 3dp for coverage). In our review of Volume 34, seven articles presented Monte Carlo SEs for estimated performance: three in the text, two in a table, one in a graph, and one in a float caption.

The primary advantage of graphical displays of performance is that it is easier to quickly spot patterns, particularly over dimensions that are not compared side-by-side. A second advantage is that it becomes possible to present raw data estimates (for example the $\hat{\theta}_i$) as well as performance results summarizing them (see, for example, figure 3 of Lambert et al[65]). In our experience, these plots are popular and intuitive ways to summarise the $\hat{\theta}_i$ and model SEs. Another example of a plot of estimates data is a histogram given in Kahan[31] (this was particularly important as Bias $\simeq 0$, but almost no $\hat{\theta}_i$ was close to θ). Even if plots of estimates are not planned to be included in publications, we urge their use in exploration of simulation results.

The main disadvantage of graphical displays of results is that plots can be less space-efficient than tables, it is not possible to read the exact numbers, and separate plots will frequently be required for different performance measures.

Compared with tables, it is easier for plots of performance results to accommodate display of Monte Carlo SEs directly, and this should be done, for example as 95% confidence intervals. The considerations about design of plots to facilitate the most relevant comparisons apply as with tables. Methods often have names that are hard to arrange side by side in a legible manner; it is usually preferable to arrange methods in horizontal rows and performance measures across columns.

As noted previously, full factorial designs can pose problems for presentation of results. One option for presentation is to present data assuming no interaction unless one is obviously present. An alternative approach taken by Rücker and Schwarzer is to present all results of a full factorial simulation study with $4 \times 4 \times 4 \times 4 \times 3 = 768$ data-generating mechanisms, and comparison of six methods.[66] Their proposal is a "nested-loop plot," which loops through nested factors – usually data-generating mechanisms – for an estimand, and plots results for different methods on top of each other.[66] This is a useful graphic but will not suit all designs (and makes depiction of Monte Carlo SE difficult).

There is no one correct way to present results, but we encourage careful thought to facilitate readability and clarity, considering the comparisons that need to be made by readers.

7 | WORKED ILLUSTRATIVE EXAMPLE

To make clear the ideas described in this article and demonstrate how they may be put into practice, we conduct one example simulation study. We hope that the aims and methods are simple enough to be understood by all readers. Further, the files required to run the simulation in Stata are available at https://github.com/tpmorris/simtutorial (with the addition of code for other software planned).

7.1 | Design of example

The example is a comparison of three different methods for estimating the hazard ratio in a randomised trial with a survival outcome.

Consider the proportional hazards model, where we have the hazard rate (event rate at time t conditional on survival until at least time t) for the ith patient

$$h_i(t) = h_0(t)\exp(X_i\theta), \tag{2}$$

with $h_0(t)$ the baseline hazard function, X_i a binary treatment indicator variable coded 0 for control and 1 for the research arm, and θ the log hazard ratio for the effect of treatment. There are various ways to estimate this hazard ratio, with common approaches being the Cox model and standard parametric survival models, such as the exponential and Weibull. The parametric approaches make assumptions about the form of the baseline hazard function $h_0(t)$ whereas the Cox model makes no such assumption. We now describe a simulation study to evaluate the three methods in this simple setting.

Aims: To evaluate the impacts (1) of misspecifying the baseline hazard function on the estimate of the treatment effect θ; (2) of fitting too complex a model when an exponential is sufficient; and (3) of avoiding the issue by using a semiparametric model.

Data-generating mechanisms: We consider two data-generating mechanisms. For both, data are simulated on $n_{\text{obs}} = 500$ patients, representing a possible phase III trial with survival outcome. Let $X_i \in (0, 1)$ be an indicator denoting assignment to treatment, where assignment is generated using $X_i \sim \text{Bern}(0.5)$ – simple randomisation with an equal allocation ratio. We simulate survival times from the model in Equation 2, assuming that $\theta = -0.5$, corresponding to a hazard ratio of 0.607 (3dp). We let $h_0(t) = \lambda\gamma t^{\gamma-1}$. The two data-generating mechanisms differ only in the values of γ:

1. $\lambda = 0.1, \gamma = 1 \leftarrow$ both an exponential and a Weibull model
2. $\lambda = 0.1, \gamma = 1.5 \leftarrow$ a Weibull but not an exponential model.

A plot of the hazard rate $h_i(t)$ for the two data-generating mechanisms is given in Figure 2.

Data are simulated using Stata 15 using the 64-bit Mersenne twister for random number generation. The input seed is "72789."

Estimands: Our estimand θ is the log-hazard ratio for $X = 1$ vs $X = 0$, which would represent a treatment effect in a randomised trial.

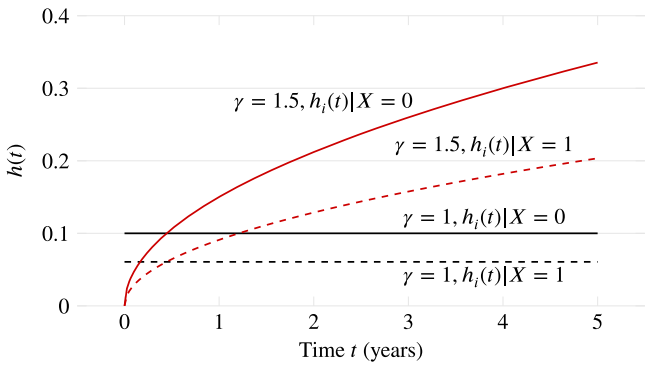


FIGURE 2 Visualisation of the true hazard rate over follow-up time in the two data-generating mechanisms. Black (flat) lines are for the first data-generating mechanism, where $\gamma = 1$; Red curves are for the second, where $\gamma = 1.5$ [Colour figure can be viewed at wileyonlinelibrary.com]

Methods: Each simulated dataset is analyzed in three ways, using the following:

1. An exponential proportional-hazards model;
2. A Weibull proportional-hazards model;
3. A Cox proportional-hazards model.

Note that the exponential model is correctly specified for the first data-generating mechanism but misspecified for the second; the Weibull model is correctly specified for both mechanisms; and the Cox model does not make any assumption about the baseline hazard so is not misspecified for either mechanism.

Performance measures: We will assess convergence, bias, coverage, empirical, and model-based standard errors for $\hat{\theta}$.

Bias is our key performance measure of interest, and we will assume that $\text{SD}(\hat{\theta}) \leq 0.2$, meaning that $\text{Var}(\hat{\theta}) \leq 0.04$. (A conservative estimate based on an initial small simulation run.) We decide that we require Monte Carlo SE of bias to be lower than 0.005. Given that

$$\text{Monte Carlo SE(Bias)} = \sqrt{\text{Var}(\hat{\theta})/n_{\text{sim}}},$$

this implies that we need 1600 repetitions. If coverage of all methods is 95%, the implication of using $n_{\text{sim}} = 1600$ is

$$\text{Monte Carlo SE(Coverage)} = \sqrt{\frac{95 \times 5}{1600}} = 0.54.$$

With 50% coverage, the Monte Carlo SE is maximised at 1.25. We consider this satisfactory and so proceed with $n_{\text{sim}} = 1600$ (to be revised if, for example, $\text{SD}(\hat{\theta}) > 0.2$).

7.2 | Exploration and visualisation of results

The first result to note is that there were no missing $\hat{\theta}_i$ or $\widehat{\text{SE}}(\hat{\theta}_i)$, and "separation" was not an issue.

We first explore the raw results. Figure 3 plots the estimates $\hat{\theta}_i$ and $\widehat{\text{SE}}(\hat{\theta})_i$ for the two data-generating mechanisms and three methods, with means displayed as yellow pipes. The left panels plot $\hat{\theta}_i$. It is clear that, when $\gamma = 1$, the mean and variance of $\hat{\theta}_i$ is very similar for the three methods. The mean is close to the true value of $\theta = -0.5$ for all methods. When in truth $\gamma = 1.5$, the empirical SE is slightly higher for all methods (because there are fewer events among the 500 observations under this data-generating mechanism). The exponential proportional-hazards model is now misspecified and we observe a shift of the mean of $\hat{\theta}_i$ towards the null, indicating some bias. The right panels of Figure 3 plot the estimated standard errors $\widehat{\text{SE}}(\hat{\theta})_i$. These are smaller for the upper panel ($\gamma = 1$) than the lower panel ($\gamma = 1.5$), but there is little to choose between the methods.

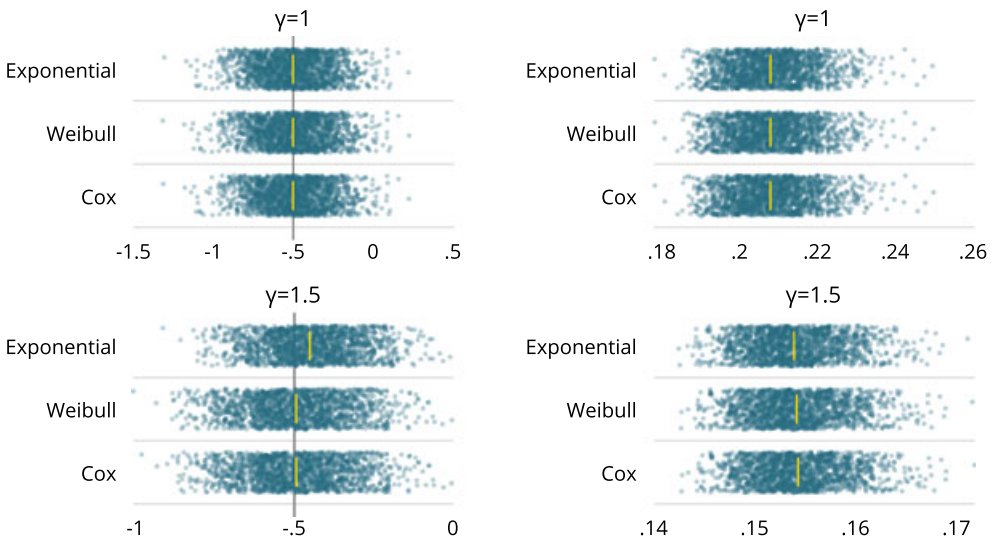


FIGURE 3 Plot of the 1600 $\hat{\theta}_i$ (left panels) and $\widehat{\text{SE}}(\hat{\theta})_i$ (right panels) by data-generating mechanisms, for the three analysis methods. The vertical axis is repetition number, to provide some separation between points. The yellow pipes are sample means [Colour figure can be viewed at wileyonlinelibrary.com]

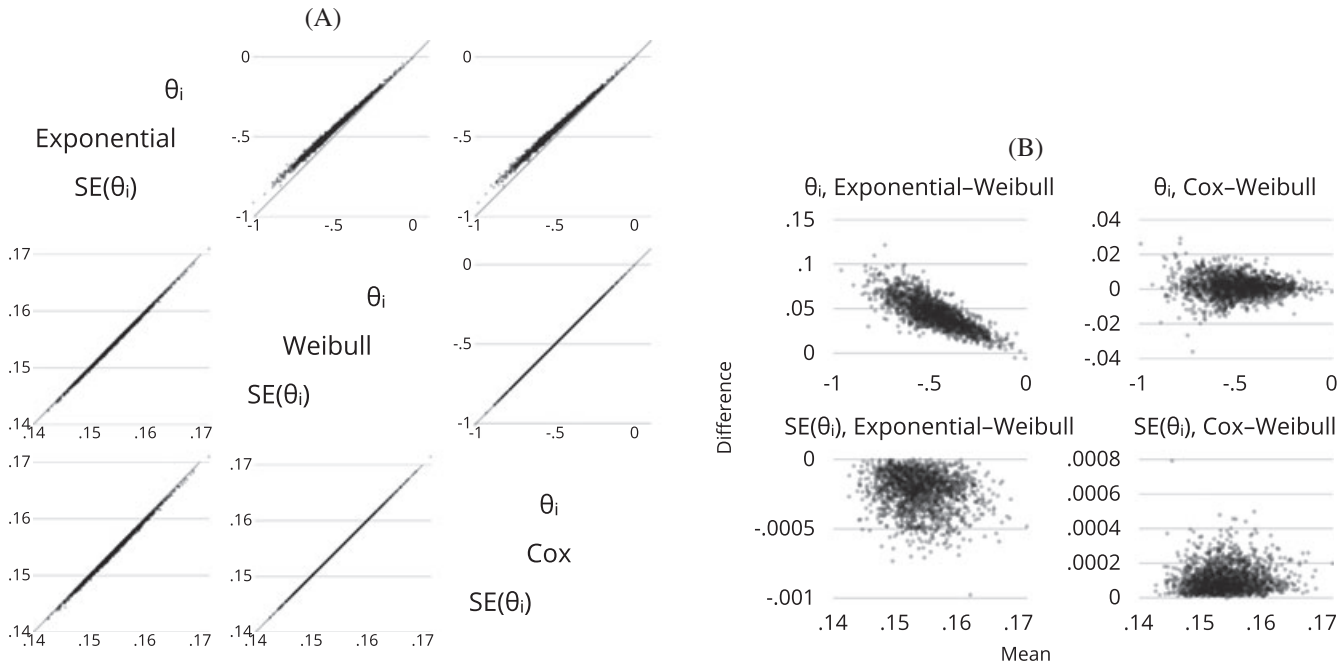


FIGURE 4 Comparison of estimates for methods when $\gamma = 1.5$, where each point represents one repetition. A, Upper triangle displays $\hat{\theta}_i$; lower triangle displays $\widehat{\mathrm{SE}}(\hat{\theta}_i)$; B, Plot of difference vs mean for $\hat{\theta}_i$ and $\widehat{\mathrm{SE}}(\hat{\theta}_i)$, with Weibull as the comparator

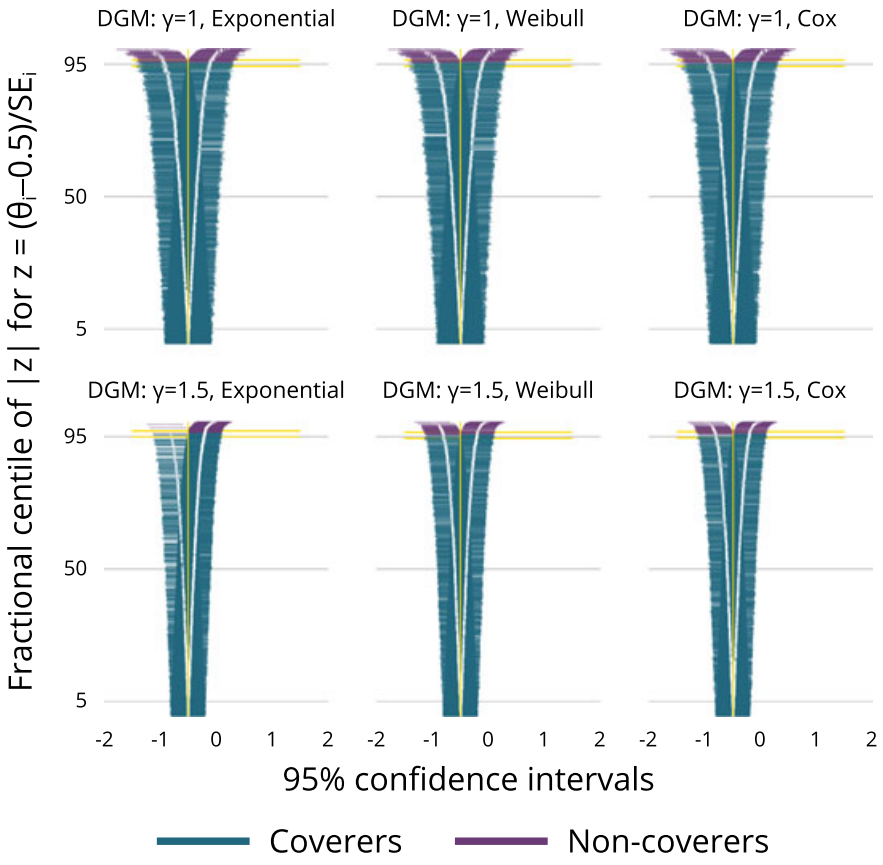


FIGURE 5 "Zip plot" of the 1600 confidence intervals for each data-generating mechanism and analysis method. The vertical axis is the fractional centile of $|z|$ with $z = (\hat{\theta}_i - \theta)/\mathrm{ModSE}$ associated with the confidence interval [Colour figure can be viewed at wileyonlinelibrary.com]

We next compare these estimates by plotting $\hat{\theta}_i$ for each method vs every other method, and the same for $\widehat{\text{SE}}(\hat{\theta})_i$. The data pairs come from the same repetition (ie, they are estimated in the same simulated dataset) and are compared to the line of equality. This is done in Figure 4A, for the second data-generating mechanism only ($\gamma = 1.5$), of interest because the exponential model is misspecified. We can see that the estimates of both $\hat{\theta}_i$ and $\widehat{\text{SE}}(\hat{\theta}_i)$ are highly correlated across all methods. The upper triangle of plots in Figure 4 shows that, while $\hat{\theta}_i$ is almost identical for the Weibull and Cox models, it tends to be systematically closer to 0 for the exponential model. The estimates of $\widehat{\text{SE}}(\hat{\theta}_i)$ show that again, the estimates are extremely similar for the Weibull and Cox models, and are very slightly larger for the exponential model. Figure 4(B) gives the corresponding plots of the difference vs mean (*Weibull* is the comparator method here as it is correctly specified).

Figure 5 is a new visualisation, the "zip plot," which helps to understand coverage by viewing the confidence intervals directly (implemented in Stata; for implementations in R and SAS, see the works of Gasparini and White[67] and Wicklin,[68]

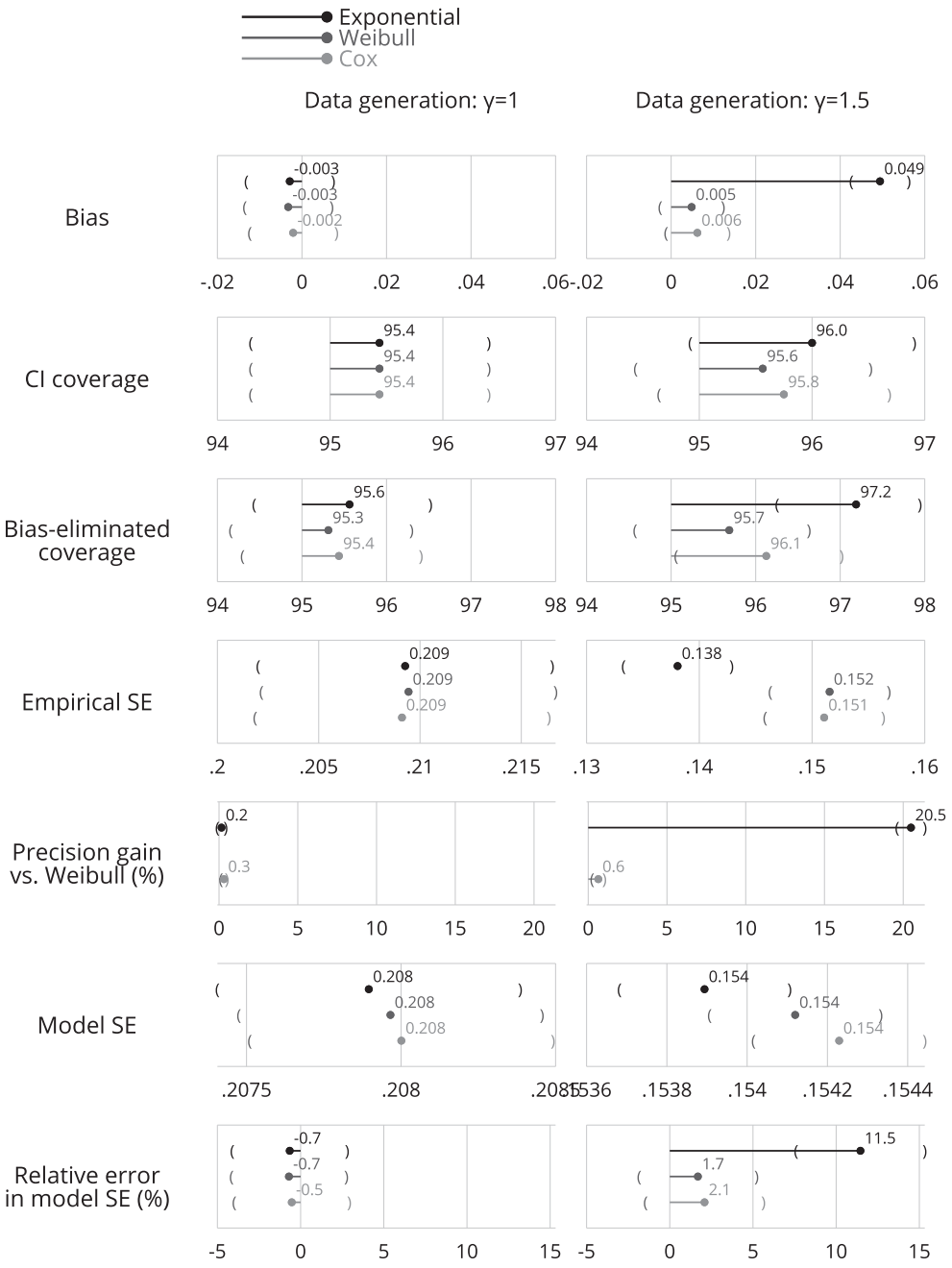


FIGURE 6 Lollipop plot of performance for measures of interest (Monte Carlo 95% confidence intervals in parentheses). Concerning features need not be highlighted since they are readily visible. See, also, Table 8

respectively). For each data-generating mechanism and method, the confidence intervals are fractional-centile-ranked according to $|z_i|$, where $z_i = (\hat{\theta}_i - \theta)/\text{ModSE}$. This ranking is used for the vertical axis and is plotted against the intervals themselves. Intervals which cover θ are colored blue (bottom); those which do not cover are colored purple (top). When a method has 95% coverage, the color of the intervals switches at 95 on the vertical axis. The yellow horizontal lines are Monte Carlo 95% confidence intervals for per cent coverage. (As a general comment, note that the interesting area in many zip plots will be near to the top and so it will often be more informative to "zoom in" on the action, as suggested by Wicklin.[68])

In Figure 5, the upper panel again displays the results when $\gamma = 1$ and the lower panel when $\gamma = 1.5$. Despite coverage being approximately 95% as advertised, there are more intervals to the right of $\theta = -0.5$ than to the left, particularly for those that do not cover θ. This indicates that the model SEs must overestimate the empirical SE, because coverage is adequate despite bias. A zip plot helps to make such a feature clear.

7.3 | Analysis of example

The previous section demonstrated some exploratory analyses that may be of value. Next, we estimate performance for the measures of interest and present them in a table for which (we hope) the ADEMP structure is clear: different performance measures are stacked vertically; for each performance measure, the results for the two data-generating mechanisms each occupy one row; results for different methods are arranged across three columns (with Monte Carlo SEs in parentheses at a smaller point size than the estimate); there is only one estimand.

Also given in Figure 6 is an alternative graphical presentation of estimated performance called a *lollipop plot*. The ADEMP structure is slightly different to the table but again clear: different performance measures are stacked vertically; for each performance measure, the results for the three methods now occupy one row each; results for different methods are arranged across the two columns. Monte Carlo 95% confidence intervals are now represented via parentheses (a visual cue due to the usual presentation of intervals as two numbers within parentheses).

The results confirm more formally some of the features we saw in our exploration of the estimates data. The interesting features concern the exponential model when $\gamma = 1.5$ since the Weibull and Cox models behave well in all cases. We see that the exponential model suffers some bias towards the null, which is approximately 10% of the true value. This is nonnegligible. Next, we see that coverage is still over the nominal 95%, which is surprising in the presence of bias. The empirical SE is the same for all models when $\gamma = 1$ and lowest for the exponential model when $\gamma = 1.5$, while the Weibull and Cox models are very similar; recall however that in the presence of different biases, the empirical SE is not comparable across methods. For relative precision (vs the Weibull model) a very similar pattern is seen as for empirical SE. The model SE is the same for all methods and data-generating mechanisms. This explains why the exponential model has acceptable coverage when $\gamma = 1.5$: the bias is cancelled out by the fact that the model SE is overestimated. This is confirmed by the relative error in Model SE.

TABLE 8 Estimates of performance for measures of interest (Monte Carlo SEs in parentheses). Concerning results are highlighted in bold. See, also, Figure 6

Performance Measure	Data-generating Mechanism	Method: Exponential		Weibull		Cox	
Bias	$\gamma = 1$	−0.003	(0.005)	−0.003	(0.005)	−0.002	(0.005)
	$\gamma = 1.5$	**0.049**	(0.003)	0.005	(0.004)	0.006	(0.004)
Coverage	$\gamma = 1$	95.4%	(0.5)	95.4%	(0.5)	95.4%	(0.5)
	$\gamma = 1.5$	96.0%	(0.5)	95.6%	(0.5)	95.8%	(0.5)
Bias-eliminated	$\gamma = 1$	95.6%	(0.5)	95.3%	(0.5)	95.4%	(0.5)
coverage	$\gamma = 1.5$	**97.2%**	(0.4)	95.7%	(0.5)	96.1%	(0.5)
Empirical SE	$\gamma = 1$	0.209	(0.004)	0.209	(0.004)	0.209	(0.004)
	$\gamma = 1.5$	0.138	(0.002)	0.152	(0.003)	0.151	(0.003)
Relative precision	$\gamma = 1$	0.2%	(0.1)	0	(–)	0.3%	(0.1)
gain vs Weibull	$\gamma = 1.5$	20.5%	(0.4)	0	(–)	0.6%	(0.2)
Model SE	$\gamma = 1$	0.208	(<0.001)	0.208	(<0.001)	0.208	(<0.001)
	$\gamma = 1.5$	0.154	(<0.001)	0.154	(<0.001)	0.154	(<0.001)
Relative error in	$\gamma = 1$	−0.7%	(1.8)	−0.7%	(1.8)	−0.5%	(1.8)
Model SE	$\gamma = 1.5$	**11.5%**	(2.0)	1.7%	(1.8)	2.1%	(1.8)

Looking at Table 8, the Monte Carlo SEs of performance estimates are all acceptable and so we would be happy to draw conclusions about the methods based on the 1600 repetitions.

7.4 | Conclusions of example

When an exponential model is misspecified, the hazard ratio can be biased but, probably, not by much. Further research is needed.

More seriously, note that the data-generating mechanisms we used do not cover any real breadth of scenarios. For example, we might have explored varying n_{obs}, λ, and θ over a range of values to explore when issues are present.

8 | CONCLUDING REMARKS

Simulation studies are an invaluable tool for research into statistical methods, evidenced by the large proportion of Volume 34 of *Statistics in Medicine* articles whose conclusions relied in part on simulation studies. Because methods promoted may be used in medical research (or many other scientific areas), transparent reporting of the design and execution of simulation studies is critical.

While simulation studies are widely used, they tend to be poorly reported by those who publish their results.

There are many areas to be improved in the reporting of simulation studies. Our view is that the two main shortcomings are (i) lack of clarity over the design, which ADEMP aims to deal with, and (ii) failure to report estimates of Monte Carlo uncertainty.

We have described and advocate a structured approach to the planning of simulation studies that involves identifying *aims*, *data-generating mechanisms*, *methods*, *estimands* and *performance measures*. All of these and the rationale for decisions should be included in reporting. For an excellent example of a clearly described design, see Austin and Stuart.[69] Reports of simulation studies are now beginning to explicitly use the ADEMP structure; see Thompson et al,[70] Sayers et al,[71] Morris et al,[56] Keogh and Morris,[55] and Pham et al.[72]

We have given formulas for computing the Monte Carlo standard error for the most common performance measures, and made some suggestions about reporting. Note that the Stata package `simsum`[12] and R package `rsimsum`[67] automate this process for commonly used performance measures. See Boos and Osborne for more general assessment of Monte Carlo SEs for complex performance measures.[73]

8.1 | Future directions

Three areas that we regard as of increasing future importance are simulation protocols, release of code, and consortia of authors. We discuss these as a step towards resolving occurrences of simulation studies with contradictory results.

A charitable view of such contradictory results, which we tend to hold, is that methods are developed by researchers who concerned with handling the specific problems they have seen in practice. Given such a background, they are running the relevant simulation studies (that may not be relevant to others). A less charitable possibility might be selective reporting of only the most favorable (or unfavorable) configurations of data-generating mechanisms, running the simulations many times under different seeds and selecting the most favorable,[3] and more (left to the reader's imagination).

A starting point to addressing this issue is to write detailed simulation protocols before writing code. This would ideally protect against authors choosing to report favorable results and force one to be clear about "ADEMP" in advance. Of course, the weak point is that simulation studies do not to need approval before "data collection," so protocols could be written after-the-fact and this cannot be clear even with published protocols. The counterargument is that protocols must justify the rationale for choices: as well as describing *what* is planned, there is a burden to explain *why*.

Due to the prejudices introduced by experiences of data, it is constructive for authors who have produced contradictory simulation results to work together on "late-phase" simulation studies (using an analogy from clinical trials in drug development). This allows robust discussion of the design and exploration of disparities among previous work. One exemplar of such an approach is in methods for handling incomplete data where the analysis has a multilevel structure. Three groups of researchers had developed methods and worked together on understanding how the methods differ and on simulation studies to evaluate their performance, resulting in the paper by Audigier et al.[74] This approach is in our view more satisfactory than a group of researchers executing a large, late-phase simulation study without the input of authors of previous work, though this strategy is sometimes adopted.[75]

Boulestix, Wilson and Hapfelmeier raise an important consideration[76] for benchmarking studies that is also relevant to late-phase simulation studies: it is easy to assume that the performance of methods is due entirely to the methods themselves, but this ignores the fact that the implementation of a method involves the knowledge and skill of the individuals involved. A poor implementation may make a superior method appear weak. This suggests again that including consortia of authors, as in Audigier et al, is advisable.

No simulation study is definitive and new methods or refinements of methods are inevitable. For researchers wishing to replicate or extend the results of earlier simulation studies, the design of earlier work must have been written out fully and unambiguously. This can be a difficult task and, to ensure this, authors should release simulation code publicly (a policy that is now encouraged by some journals, notably *Biometrical Journal*, and required by others). The happy corollary is that code may be checked more thoroughly by its authors if it is subject to external scrutiny. There is generally no excuse for withholding code. One caveat is for resampling studies, where permissions to release the original data may be lacking (note that code for running the simulation can still be made available even if it cannot be run on the same data).

8.2 | Final remark

Simulation studies are a powerful tool. However, it is important to be aware that, because a simulation study took hard work and thought, we are liable to believe it tells us more than it truly does. To quote Patrick Royston, "Simulation studies reveal points of light on a landscape, but can not illuminate the entire landscape." We can hope and plan to illuminate important points and build up a picture of the landscape, particularly where terrain may be particularly rocky or particularly fertile.

We hope that the guidance in this tutorial will improve researchers' understanding, planning, execution, and future reporting of simulation studies.

ACKNOWLEDGEMENTS

Tim Morris and Ian White are supported by the Medical Research Council (grant numbers MC_UU_12023/21 and MC_UU_12023/29). Michael Crowther is partly supported by a Medical Research Council New Investigator Research Grant (grant number MR/P015433/1).

For thought-provoking discussions and input to this work, we thank Alessandro Gasparini, Tra Pham, Brennan Kahan, Ruth Keogh, Clémence Leyrat, Kristian Brock, Christian Hennig and Patrick Royston. We also thank the many participants who have attended our courses and whose questions and feedback provided the motivation for this article.

The initial manuscript and revised version of this article were released as pre-prints, and we are grateful to the people who contributed reviews and comments on these. In particular, those who made important substantive comments: Maarten van Smeden, Rolf Groenwold, Bas Penning-de Vries, Kim Luijken, Martina McMenamin, Paula Dhiman, Leane McCabe, Alessandro Gasparini, Stephen Senn, Adrian Sayers, Richard Torkar, David Mannheim, Daniel Oberski, Teague Henry, and Rick Wicklin.

CONFLICTS OF INTEREST

All authors declare that they developed and regularly deliver a short course on simulation studies, from which this work grew and from which they have benefited financially.

ORCID

Tim P. Morris https://orcid.org/0000-0001-5850-3610
Ian R. White https://orcid.org/0000-0002-6718-7661
Michael J. Crowther https://orcid.org/0000-0001-8378-8259

REFERENCES

1. Feiveson AH. Power by simulation. *Stata J*. 2002;2(2):107-124.
2. Rubin DB. Bayesianly justifiable and relevant frequency calculations for the applies statistician. *Ann Stat*. 1984;12(4):1151-1172.
3. Grieve AP. Idle thoughts of a 'well-calibrated' Bayesian in clinical drug development. *Pharm Stat*. 2016;15(2):96-108.
4. Hoaglin DC, Andrews DF. The reporting of computation-based results in statistics. *Am Stat*. 1975;29(3):122-126.

5. Hauck WW, Anderson S. A survey regarding the reporting of simulation studies. *Am Stat*. 1984;38(3):214-216.
6. Ripley BD. *Stochastic Simulation*. Hoboken, NJ: John Wiley & Sons; 1987.
7. Burton A, Altman DG, Royston P, Holder RL. The design of simulation studies in medical statistics. *Statist Med*. 2006;25(24):4279-4292.
8. Koehler E, Brown E, Haneuse SJPA. On the assessment of Monte Carlo error in simulation-based statistical analyses. *Am Stat*. 2009;63(2):155-162.
9. Morgan BJT. *Elements of Simulation*. Boca Raton, FL: Chapman & Hall/CRC; 1995.
10. Chang M. *Monte Carlo Simulation for the Pharmaceutical Industry: Concepts, Algorithms, and Case Studies*. Boca Raton, FL: CRC Press; 2011.
11. Díaz-Emparanza I. Is a small Monte Carlo analysis a good analysis? *Stat Pap*. 2002;43(4):567-577.
12. White IR. simsum: analyses of simulation studies including Monte Carlo error. *Stata J*. 2010;10(3):369-385.
13. Smith MK, Marshall A. Importance of protocols for simulation studies in clinical drug development. *Stat Methods Med Res*. 2011;20(6):613-622.
14. Crowther MJ, Lambert PC. Simulating biologically plausible complex survival data. *Statist Med*. 2013;32(23):4118-4134.
15. Holford NHG, Hale M, Ko HC, et al. Simulation in drug development: good practices. Draft Publication of the Center for Drug Development Science (CDDS). 1999.
16. O'Kelly M, Anisimov V, Campbell C, Hamilton S. Proposed best practice for projects that involve modelling and simulation. *Pharm Stat*. 2017;16(2):107-113.
17. Haramoto H, Matsumoto M, Nishimura T, Panneton F, L'Ecuyer P. Efficient jump ahead for F2-linear random number generators. *INFORMS J Comput*. 2008;20(3):385-390.
18. Kenward MG, Roger JH. Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics*. 1997;53(3):983-997.
19. White IR. Letter to the editor: survival analysis of randomized clinical trials adjusted for patients who switch treatments by M. G. Law and J. M. Kaldor, Statistics in Medicine, 15, 2069–2076 (1996). *Statist Med*. 1997;16(22):2619-2620.
20. White IR, Thompson SG. Adjusting for partially missing baseline measurements in randomized trials. *Statist Med*. 2005;24(7):993-1007.
21. Hughes RA, Sterne JAC, Tilling K. Comparison of imputation variance estimators. *Stat Methods Med Res*. 2016;25(6):2541-2557.
22. Morris TP, White IR, Royston P. Tuning multiple imputation by predictive mean matching and local residual draws. *BMC Med Res Methodol*. 2014;14(1):75.
23. Skrondal A. Design and analysis of Monte Carlo experiments: attacking the conventional wisdom. *Multivar Behav Res*. 2000;35(2):137-167.
24. Graves N, Barnett AG, Burn E, Cook D. Smaller clinical trials for dec,on making; using p-values could be costly. London, UK: F1000Research; 2018;7.
25. Senn S, Chambless LE, Roeback JR. Methods for assessing difference between groups in change when initial measurement is subject to intra-individual variation. *Statist Med*. 1994;13(21):2280-2285.
26. Kuss O. Statistical methods for meta-analyses including information from studies without any events—add nothing to nothing and succeed nevertheless. *Statist Med*. 2015;34(7):1097-1116.
27. Chaurasia A, Harel O. Partial F-tests with multiply imputed data in the linear regression framework via coefficient of determination. *Statist Med*. 2015;34(3):432-443.
28. Wu C, Shi X, Cui Y, Ma S. A penalized robust semiparametric approach for gene–environment interactions. *Statist Med*. 2015;34(30):4016-4030.
29. Ferrante L, Skrami E, Gesuita R, Cameriere R. Bayesian calibration for forensic age estimation. *Statist Med*. 2015;34(10):1779-1790.
30. Zhang Z, Wang C, Troendle JF. Optimizing the order of hypotheses in serial testing of multiple endpoints in clinical trials. *Statist Med*. 2015;34(9):1467-1482.
31. Kahan BC. Bias in randomised factorial trials. *Statist Med*. 2013;32(26):4540-4549.
32. Campbell H, Dean CB. The consequences of proportional hazards based model selection. *Statist Med*. 2014;33(6):1042-1056.
33. Robins JM, Wang N. Inference for imputation estimators. *Biometrika*. 2000;87(1):113-124.
34. Reiter JP. Multiple imputation when records used for imputation are not used or disseminated for analysis. *Biometrika*. 2008;95(4):933-946.
35. Hauck WW, Anderson S, Marcus SM. Should we adjust for covariates in nonlinear regression analyses of randomized trials? *Control Clin Trials*. 1998;19(3):249-256.
36. Zhang Z. Estimating a marginal causal odds ratio subject to confounding. *Commun Stat Theory Method*. 2008;38(3):309-321.
37. van Smeden M, de Groot JAH, Moons KGM, et al. No rationale for 1 variable per 10 events criterion for binary logistic regression analysis. *BMC Med Res Methodol*. 2016;16(1).
38. Crowther MJ, Look MP, Riley RD. Multilevel mixed effects parametric survival models using adaptive Gauss–Hermite quadrature with application to recurrent events and individual participant data meta-analysis. *Statist Med*. 2014;33(22):3844-3858.
39. Royston P, Barthel FMS, Parmar MKB, Choodari-Oskooei B, Isham V. Designs for clinical trials with time-to-event outcomes based on stopping guidelines for lack of benefit. *Trials*. 2011;12(1):81.
40. Bartlett JW. Combining bootstrapping with multiple imputation. 2016. http://thestatsgeek.com/2016/03/12/combining-bootstrapping-with-multiple-imputation/
41. Rubin DB. Inference and missing data. *Biometrika*. 1976;63(3):581-592.

42. White IR, Carlin JB. Bias and efficiency of multiple imputation compared with complete-case analysis for missing covariate values. *Statist Med*. 2010;29(28):2920-2931.
43. White IR, Royston P. Imputing missing covariate values for the Cox model. *Statist Med*. 2009;28(15):1982-1998.
44. Neyman J. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *J R Stat Soc*. 1934;97(4):558-625.
45. Meng XL. Multiple-imputation inferences with uncongenial sources of input. *Stat Sci*. 1994;9(4):538-558.
46. Rubin DB. Multiple imputation after 18+ years. *J Am Stat Assoc*. 1996;91(434):473-489.
47. Morris TP. Rank minimization with a two-step analysis should not replace randomization in clinical trials. *J Clin Epidemiol*. 2012;65(7):810-811.
48. Kimani PK, Todd S, Stallard N. Estimation after subpopulation selection in adaptive seamless trials. *Statist Med*. 2015;34(18):2581-2601.
49. Carreras M, Gutjahr G, Brannath W. Adaptive seamless designs with interim treatment selection: a case study in oncology. *Statist Med*. 2015;34(8):1317-1333.
50. Efron B, Hastie T. *Computer Age Statistical Inference*. Cambridge, UK: Cambridge University Press; 2016.
51. Gutman R, Rubin DB. Robust estimation of causal effects of binary treatments in unconfounded studies with dichotomous outcomes. *Statist Med*. 2013;32(11):1795-1814.
52. Taguri M, Chiba Y. A principal stratification approach for evaluating natural direct and indirect effects in the presence of treatment-induced intermediate confounding. *Statist Med*. 2015;34(1):131-144.
53. Li P, Redden DT. Small sample performance of bias-corrected sandwich estimators for cluster-randomized trials with binary outcomes. *Statist Med*. 2015;34(2):281-296.
54. Marozzi M. Multivariate multidistance tests for high-dimensional low sample size case-control studies. *Statist Med*. 2015;34(9):1511-1526.
55. Keogh RH, Morris TP. Multiple imputation in Cox regression when there are time-varying effects of covariates. *Statist Med*. 2018;37(25):3661-3678.
56. Morris TP, Fisher DJ, Kenward MG, Carpenter JR. Meta-analysis of Gaussian individual patient data: two stage or not two stage? *Statist Med*. 2018;37(9):1419-1438.
57. Cramér H. *Mathematical Methods of Statistics*. Princeton, NJ: Princeton University Press; 1946.
58. Rao CR. Information and accuracy attainable in the estimation of statistical parameters. *Bull Calcutta Math Soc*. 1945;37:81-91.
59. Williamson EJ, Forbes A, White IR. Variance reduction in randomised trials by inverse probability weighting using the propensity score. *Statist Med*. 2014;33(5):721-737.
60. He X, Whitmore GA, Loo GY, Hochberg MC, Lee MLT. A model for time to fracture with a shock stream superimposed on progressive degradation: the study of osteoporotic fractures. *Statist Med*. 2015;34(4):652-663.
61. Chen Y, Hong C, Riley RD. An alternative pseudolikelihood method for multivariate random-effects meta-analysis. *Statist Med*. 2015;34(3):361-380.
62. Hsu CH, Taylor JMG, Hu C. Analysis of accelerated failure time data with dependent censoring using auxiliary variables via nonparametric multiple imputation. *Statist Med*. 2015;34(19):2768-2780.
63. Alonso A, Milanzi E, Molenberghs G, Buyck C, Bijnens L. A new modeling approach for quantifying expert opinion in the drug discovery process. *Statist Med*. 2015;34(9):1590-1604.
64. Seaman SR, Bartlett JW, White IR. Multiple imputation of missing covariates with non-linear effects and interactions: an evaluation of statistical methods. *BMC Med Res Methodol*. 2012;12(1):46.
65. Lambert PC, Dickman PW, Rutherford MJ. Comparison of different approaches to estimating age standardized net survival. *BMC Med Res Methodol*. 2015;15(1):64.
66. Rücker G, Schwarzer G. Presenting simulation results in a nested loop plot. *BMC Med Res Methodol*. 2014;14(1):129.
67. Gasparini A, White IR. rsimsum: analysis of simulation studies including Monte Carlo error. R package version 0.3.1. 2018.
68. Wicklin R. A zipper plot for visualizing coverage probability in simulation studies. Cary, NC: SAS Institute Inc; 2018.
69. Austin PC, Stuart EA. Optimal full matching for survival outcomes: a method that merits more widespread use. *Statist Med*. 2015;34(30):3949-3967.
70. Thompson JA, Fielding KL, Davey C, Aiken AM, Hargreaves JR, Hayes RJ. Bias and inference from misspecified mixed-effect models in stepped wedge trial analysis. *Statist Med*. 2017;36(23):3670-3682.
71. Sayers A, Crowther MJ, Judge A, Whitehouse MR, Blom AW. Determining the sample size required to establish whether a medical device is non-inferior to an external benchmark. *BMJ Open*. 2017;7(8):e015397.
72. Pham TM, Carpenter JR, Morris TP, Wood AM, Petersen I. Population-calibrated multiple imputation for a binary/categorical covariate in categorical regression models. *Statist Med*. 2018.
73. Boos DD, Osborne JA. Assessing variability of complex descriptive statistics in Monte Carlo studies using resampling methods. *Int Stat Rev*. 2015;83(2):228-238.
74. Audigier V, White IR, Jolani S, et al. Multiple imputation for multilevel data with continuous and binary variables. *Stat Sci*. 2018;33(2):160-183.
75. Langan D, Higgins JPT, Simmonds M. Comparative performance of heterogeneity variance estimators in meta-analysis: a review of simulation studies. *Res Synth Methods*. 2017;8(2):181-198.
76. Boulesteix AL, Wilson R, Hapfelmeier A. Towards evidence-based computational statistics: lessons from clinical research on the role and design of real-data benchmark studies. *BMC Med Res Methodol*. 2017;17(1).

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Morris TP, White IR, Crowther MJ. Using simulation studies to evaluate statistical methods. *Statistics in Medicine*. 2019;38:2074–2102. https://doi.org/10.1002/sim.8086

APPENDIX

REVIEW: FURTHER INFORMATION

Agreement between TPM and each of the other authors was measured on 14 key variables for the review (IRW and MJC did not review any of the same articles). The variable-by-variable results are given in Figure A1. Note that, for some variables, there were two possible choices (eg, "Results given in a figure Y/N"), while for others, there were many (eg, "Number of repetitions n_{sim}"). Overall, TPM agreed with the other reviewers in 132 of 140 answers (94%). IRW and TPM agreed on 65 of 70 answers (93%); MJC and TPM agreed on 67 or 70 answers (96%).

Figure A2 gives summary information about how data were generated. Panel (A) shows that there was great variation in the total number of data-generating mechanisms, with the majority of simulation studies using under 20, and the largest number being ~ 41 billion. Panel (B) shows that simulation studies tended to vary few factors (with one exception). For the simulation studies varying more than one factor, the most common way to do this was in a fully factorial manner (panel (C)). However, some studies varied the factors one-at-a-time and others mixed the two together. Unfortunately, not all simulation studies noted the number of repetitions (panel (D)). The most common choices of n_{sim} were in descending order: 1000, 500, and 10,000.

Figure A3 a shows the number of estimands evaluated by the simulation studies included in the review. In general, there were a few, with a single estimand the most common. Figure A3 panel (B) gives the number of methods evaluated by the simulation studies included in the review (right panel). The majority evaluated few methods (with four as the most common number). This suggests that simulation studies provide a proof-of-concept, or that the methods are designed for new problems for which there are few alternatives available.

Table A1 lists the software packages mentioned and the number of mentions in simulation studies included in the review. This was based on a lenient judgement: for example, many articles mentioned a software package in which a method was implemented but did not mention what software was used to run the simulation study.

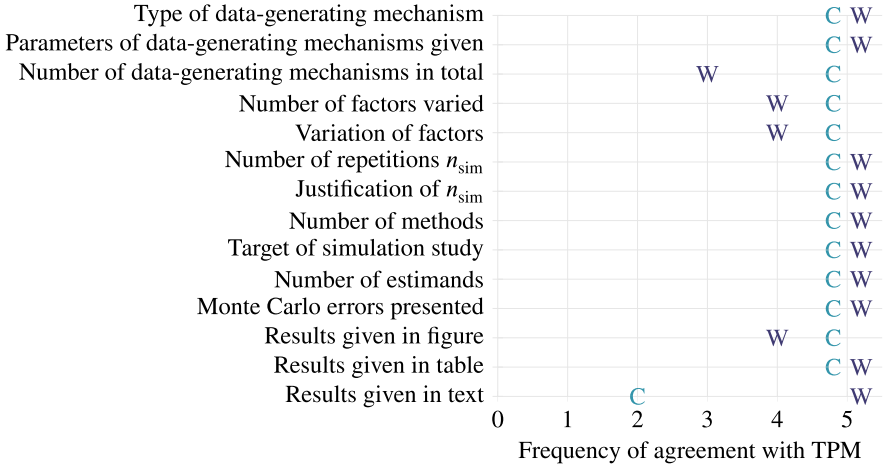


FIGURE A1 Reviewer agreement on key variables for *Statistics in Medicine* Volume 34 review. Frequency of agreement of TPM with IRW (marker W) and MJC (marker C). For the same frequency, C is nudged left and W right to avoid visual clash [Colour figure can be viewed at wileyonlinelibrary.com]

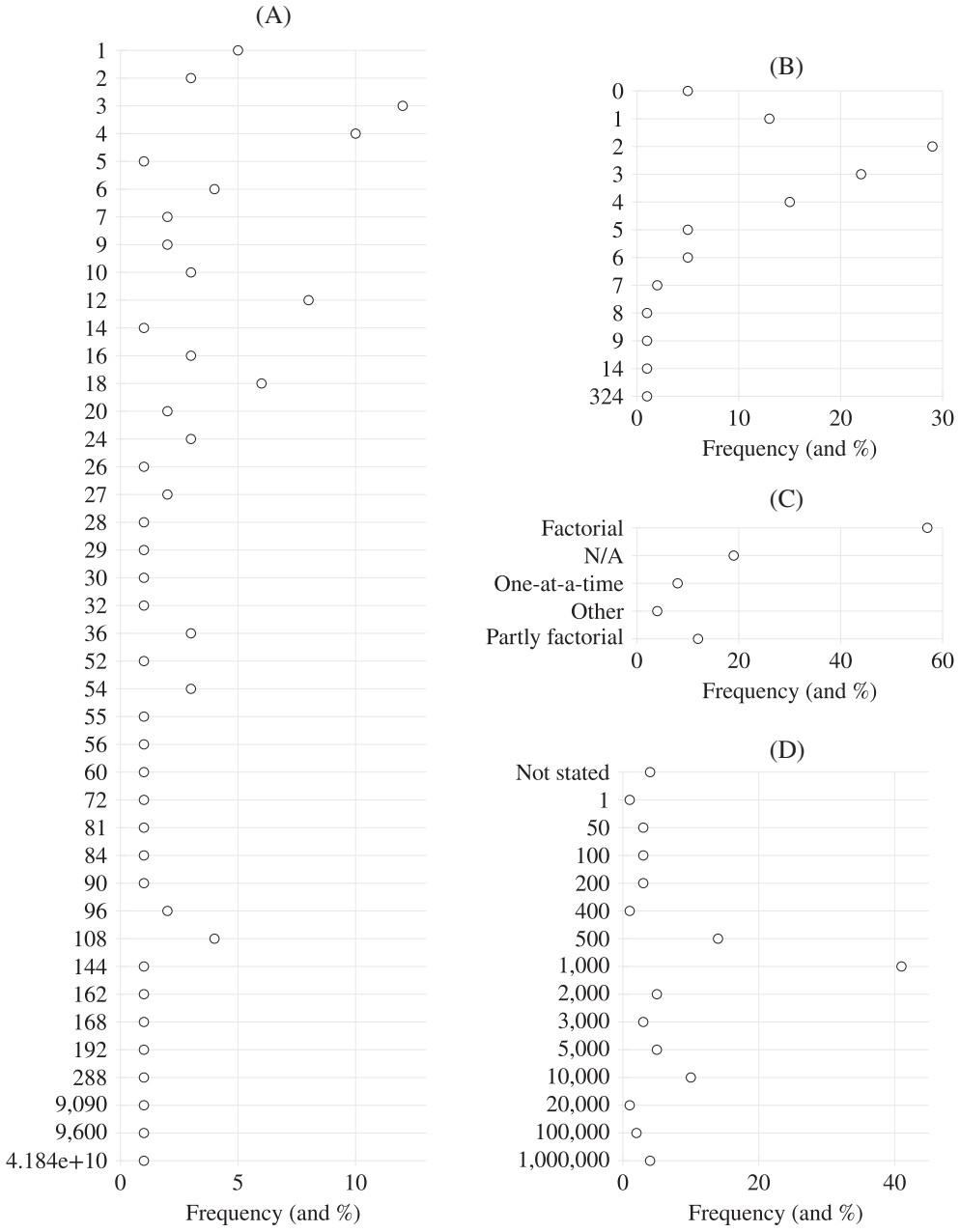


FIGURE A2 Results of *Statistics in Medicine* Volume 34 review for data-generating mechanisms. Values are both frequency and %

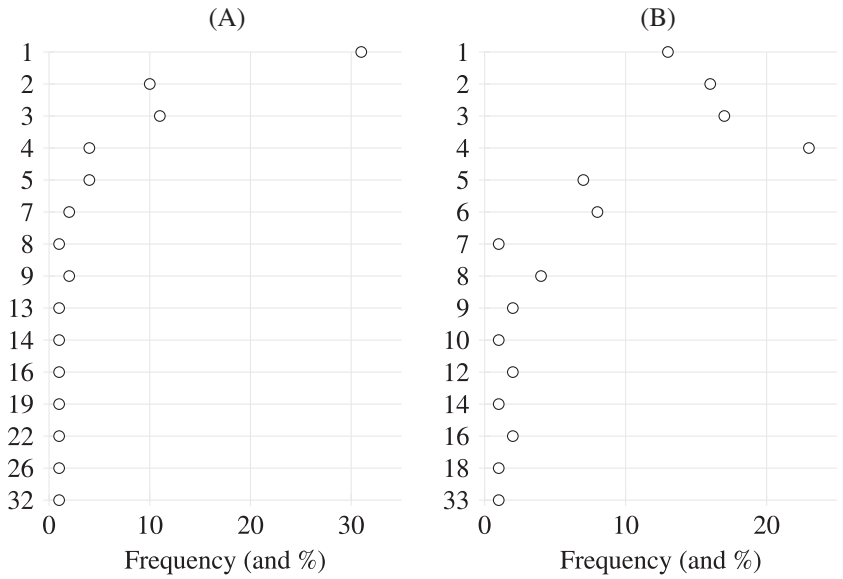


FIGURE A3 Results of *Statistics in Medicine* Volume 34 review for estimands (A) and methods (B) evaluated

TABLE A1 Software mentioned in simulation reports, review of *Statistics in Medicine* Volume 34. Note that there are more than 100 entries as some articles reported more than one package

Software	Freq.
None mentioned	38
C	1
JAGS	1
MATLAB	1
R	41
SAS	17
SaTScan	1
Stata	4
StatXact	1
WinBUGS	3