Received <day> <Month>, <year>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>

DOI: xxx/xxxx

TUTORIAL IN BIOSTATISTICS

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, UK

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

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

Present Address
90 High Holborn, London WC1V 6LJ, UK

arXiv:1712.03198v1 [stat.ME] 8 Dec 2017

Abstract

Simulation studies are computer experiments which involve creating data by pseudo-random sampling. The key strength of simulation studies is the ability to understand the behaviour of statistical methods because some 'truth' 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, analysed and reported. This article outlines the rationale for using simulation studies and offers guidance for design, execution, analysis, reporting and presentation. In particular, we provide: a structured approach for planning and reporting simulation studies; coherent terminology for simulation studies; guidance on coding simulation studies; a critical discussion of key performance measures and their computation; ideas on structuring tabular and graphical presentation of results; and new graphical presentations. With a view to describing current practice and identifying areas for improvement, we review 100 articles taken from Volume 34 of *Statistics in Medicine* which included at least one simulation study.

KEYWORDS:
Simulation studies; Monte Carlo; Experimental design; Reporting

1 | INTRODUCTION

Simulation studies are computer experiments which 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 the comparison of competing methods. Simulation studies are much used in the pages of *Statistics in Medicine*, but our experience is that many statisticians lack the necessary understanding to execute a simulation study with confidence. Proper understanding of simulation studies would enable such people to run simulation studies themselves and to critically appraise published simulation studies. Issues with design and reporting, which we will demonstrate later, lead to uncritical use or appraisal of simulation studies. In this context, better understanding of the rationale, design, execution, analysis and reporting of simulation studies is necessary to improve what researchers can learn from them.

Simulation studies are used to obtain empirical results about the behaviour of statistical methods in certain scenarios, as opposed to analytic results, which may cover many settings. It is not always possible to obtain analytic results, or may be extremely difficult. Simulation studies come into their own when methods make wrong assumptions or data are messy; that is, they can assess the resilience of methods. This is not always possible with analytic results, which may assume that data arise from a specific model.

Monte Carlo simulation has several uses that are not simulation *studies*. It is used in any analysis with a stochastic element, for example multiple imputation and Markov chain Monte Carlo methods. The remainder of this paper does not consider these uses of Monte Carlo simulation, except where 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:

- Where a new statistical method has been derived mathematically, to check algebra (and code), e.g. approximations, or to provide reassurance that no large error has been made.
- Absolute evaluation of a new or existing statistical method. Often a new method is checked using simulation to ensure it works in the scenarios it was designed for.
- Comparative evaluation of two or more statistical methods.
- Calculation of sample size / power, when designing a study under certain assumptions about the data-generating process(1).

This article is focused primarily on the evaluation of methods, but many of the principles outlined are applicable more generally. Simulations are typically motivated by frequentist theory and used to evaluate frequentist properties of methods (even if the methods are Bayesian).

Simulation studies are empirical experiments, and statisticians – particularly those doing working in applications such as clinical trials – should be familiar with good practice regarding design, analysis, presentation and reporting. It seems that as a profession we fail to follow this practice in our methodological work, as previously lamented by Hoaglin & Andrews(2), Hauck & Anderson(3), Ripley(4), Burton *et al.*(5), and Koehler, Brown & Haneuse(6). 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; failing to report measures of uncertainty would be unacceptable in medical research. There are some wonderful books on simulation methods in general (4, 7, 8) and several excellent articles encouraging rigour in specific aspects of simulation studies (for example (1, 2, 3, 6, 9, 10, 11, 12)) but no unified practical guidance on simulation studies.

This article outlines the rationale for using simulation studies and offers practical guidance for design, execution, analysis, reporting and presentation. Namely we: introduce a structured approach for planning and reporting simulation studies; introduce coherent terminology for simulation studies; provide guidance on coding simulation studies; critically discuss key performance measures and their computation; make suggestions for structuring tabular and graphical presentation of results; and introduce some new graphical presentations. This guidance is designed to enable practitioners to execute a simulation study for the first time and also contains much for more experience practitioners. For reference, the main steps involved, decisions to be made 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 'ADMEP' structure (which we define there). Section 4 gives generic guidance on coding simulation studies. In section 5, we discuss the uses of various performance measures and their computation, stressing the importance of estimating and reporting measures of Monte Carlo uncertainty. Section 6 outlines how to report simulation studies, again using the ADMEP structure, and offers guidance on tabular and graphical presentation of results. Section 7 works through a simple simulation to illustrate in practice the approaches we advocate. Section 8.1 considers some future directions, and section 8 offers some concluding remarks. Examples are drawn from the review and from the authors' areas of interest (which relate mainly to modelling survival data, missing data, meta-analysis and randomised trial design).

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

We conducted a review of practice in 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. Instead, results appear in the paper at relevant points.

We restricted attention to *research articles*, which is the type of article in which simulation studies are typically reported. This excludes the article types: *tutorial in biostatistics*, *commentary*, *book review*, *correction*, *letter to the editor* and *authors' response*. In total, this returned 264 research articles. Of these, 199 (75%) included at least one simulation study.

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

	Section
Planning	3
· Identify the aims, data-generating mechanisms, methods to evaluate, estimands or other 'targets', and performance measures	3.2–3.6
Aims	3.2
· Identify specific aims of simulation study, particularly in terms of the 'phase' (e.g. proof-of-concept)	
Data-generating mechanisms	3.3
· In relation to the aims, decide whether to use resampling or simulation from some parametric model	
· For parametric simulation, decide how simple or complex the model should be and whether it should be based on real data	
· Determine factors to vary and levels of factors to use	3.3
· Decide whether factors should be varied fully factorially, partly factorially or one-at-a-time	
Methods	3.4
· Identify methods to be evaluated. For method comparison studies, make a careful review of the literature to ensure inclusion of relevant methods.	
Estimand / target of analysis	3.5
· Define estimands and/or other targets of the simulation study, relating to the applied contexts for which method is intended to be used.	
Performance measures	3.6, 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} which achieves acceptable MCSE for key performance measures.	5.4
Coding	4
· Separate scripts used to analyse simulated datasets from scripts to analyse 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 (13).	
Analysis	5
· Conduct exploratory analysis of results, particularly graphical	
· Along with estimates of performance, compute Monte Carlo SE	
Reporting	6
· Structure graphical and tabular presentations to place performance measures for competing methods side-by-side.	
· Include estimates of simulation uncertainty.	
· Publish code to execute simulation study including user-written routines.	

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 which involved a simulation study, before randomly assigning articles to a reviewer. This meant that TPM reviewed 35 simulation studies, IRW reviewed 34 and MJC reviewed 31. In case the reviewer was an author or co-author 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.

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
$\hat{\theta}_i$	the estimate of θ from the ith repetition
$\widehat{\text{Var}}(\hat{\theta}_i)$	the estimate of $\text{Var}(\hat{\theta})$ from the ith repetition
$\text{Var}(\hat{\theta})$	is the empirical (long-run) variance of $\hat{\theta}$
α	the nominal significance level
p_i	the p-value returned by the ith repetition

3 | PLANNING SIMULATION STUDIES USING ADMEP

3.1 | Notation

For clarity about the concepts that will follow, we introduce some notation in table 2 .

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

3.2 | Aims

In considering the aims of a simulation study it is instructive to first consider desirable properties of an estimator of θ 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: $\text{E}(\hat{\theta}) = \theta$. Some estimators may be consistent but exhibit small-sample bias (logistic regression for example). Kahan reports a procedure that appears to be unbiased but inconsistent(14).
2. The sample estimate $\widehat{\text{Var}}(\hat{\theta})$ should be a consistent estimate of the long-run variance of $\hat{\theta}$ (see for example (15)).
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 may hope for, but these generally involve combinations of the above.

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 (e.g. White, 1997 (16); Precision, particularly relative to other available methods (e.g. White and Thompson, 2005 (17)); Variance estimation (e.g. Hughes, 2014 (18)); or Robustness to misspecification (e.g. Morris, 2014 (19)).

There is a distinction between simulation studies that offer a proof-of-concept and those aiming to stretch or break methods. 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 methods break; see for example (19).

Alternatively, it may be of interest to compare methods where some or all have been shown to work in principle but the methods under scrutiny deal with 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 favour certain methods over others? How can this be justified?* One common justification is by reference to motivating data. However, this carries a risk of failing to convince others that methods should be used more widely.

3.3 | Data-generating mechanisms

We use the term 'data-generating mechanism' to denote how random numbers are used to produce a simulated dataset. This is in preference to the term '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 data should be generated. See Ripley (4) or Morgan (7) for methods to simulate data from specific distributions.

Data may be generated by producing parametric draws from a known model (once or many times), or resampling with replacement many times from a specific dataset where the model from which data are generated is unknown.

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

It was noted in the introduction that simulation studies provide us with approximate 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.

Much can be controlled in a simulation study, and so statistical principles for designing experiments can and should be called on. In particular, there is often more than one factor that will vary across specific data-generating mechanisms. Varying them factorially is likely to be more informative than one-by-one as doing so permits exploration of interactions between factors. There are however practical implications which 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. If the 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, and this is inadmissible, then it may be necessary for the simulation study to follow a non-factorial structure. Two checks for interaction are outlined 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 given that performance would have been poor for other choices of data-generating mechanism.

A second and more careful approach is based on making and checking predictions beyond the data-generating mechanisms initially considered; an idea similar to external validation. Say we have two factors, A and B, where $A \in \{1, \dots, 8\}$ and $B \in \{1, \dots, 5\}$ in the data-generating mechanism. If the non-factorial portion of the design varies A from 1 to 8 holding $B = 1$, and varies B from 1 to 5 holding $A = 1$, the performance measures could be *predicted* for $A = 8, B = 5$ based on the marginal effects from the non-factorial simulation. 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 (as a result of fitting a main-effects model to existing results to produce explicit predictions at unexplored values of A and B). The simulation study can then be re-run for the single DGM, say $A = 8, B = 5$. Departures from the predicted results imply interaction, with a responsibility to explore further.

In our review, 97 simulation studies used some form of parametric model to generate data while three use resampling methods. Of those which simulated from a parametric model, 27 based parameters on data, one based parameters partly on data, and the remaining 69 on no data. Of the 97 simulation studies using a parametric data-generating model, 91 (94%) provided the parameters used. One simulation study (20) 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 DGMs per simulation study ranged from 1 to 6×10^{11}; figure A1 (in the appendix) summarises aspects of the data-generating mechanisms.

3.4 | Methods

The term 'method' is intended to be generic. It most often refers to a model for analysis, but might refer to a design or some procedure (such as a decision rule). For example, (14) and (21) both evaluated procedures which 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, the aim will be to identify the best. It is therefore 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 their only purpose is to repeat previous research and bloat the results of the current work. An exception might be if a flawed method is used lots 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 (22) and (23)), implemented poorly, or implemented in unfamiliar software. Although R and Stata tend 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 this issue as it impinges on simulation studies. Note that one important role of simulation is to verify that code is correct.

The number of methods evaluated in our review of Volume 34 ranged from 1 to 33 (see figure A2), reflecting the broad range of simulation studies in terms of aims.

Non-convergence may be an issue for certain models. In such a situation, there is a conceptual issue with defining the 'method'. A pure method evaluation would simply assess performance of the model when it converges. However, in practice an analyst whose model does not convergence would then use some other model until one converges. Thus, evaluation of this procedure may be of interest in simulation studies. Crowther, Look and Riley evaluate such a procedure: if a model failed to converge, they applied a model with more quadrature points.(24) We commment further on this issue in section 5.2.

3.5 | Estimands and other targets

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

The choice of estimand is rarely made explicit but is important. For example, in case–control studies logistic regression is used to estimate odds ratios. The estimated odds ratio for this model is asymptotically unbiased, but the estimate of odds (the intercept) is biased. This bias should not concern us because case–control studies are not used to estimate odds (without knowledge of the sampling fraction). Thus the estimand here should be the (log) odds ratio and not the intercept.

Identifying whether different methods target the same estimand can be subtle. For example, in a randomised trial with binary outcome, one might compare two logistic regression analyses, one unadjusted and one adjusted for a baseline 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.(25)

One way to tackle the problem of multiple estimands is to ensure that both methods estimate the same estimand: in the logistic regression case, this would involve post-processing the adjusted regression results to estimate the 'marginal adjusted odds ratio' (26). 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 a nuisance part of standard adjustment. An alternative but less satisfactory way to tackle the problem is to target the null hypothesis that the odds ratio equals 1, because this specific null hypothesis is the same whether the odds ratio is conditional or marginal.

Not all simulation studies evaluate or compare methods which 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

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

Statistical task	Target	Examples of performance measures	Example
Analysis			
Estimation	Estimand	Bias, mean-squared error, coverage	Kuss compares a number of existing methods in terms of bias, power and coverage (20)
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)

(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 shows which performance measures (described more fully below) may be relevant for each target.

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 simulation studies 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 which targeted an estimand is given in the appendix, figure **??**

3.6 | Performance measures

The term 'performance measure' describes a numerical quantity used to assess the performance of a method. Statistical methods for estimation may output for example an estimate $\hat{\theta}$, its variance $\widehat{\text{Var}}(\hat{\theta})$ or $\widehat{\text{SE}}(\hat{\theta})$, degrees of freedom, confidence intervals, test statistics and more (such as an estimate of prognostic ability).

The performance measures required in a simulation study depends on what the study targets (see section 3.5). 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. Precision and coverage of $(1-\alpha)$ confidence intervals may 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

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
Coverage	**42/79 (53%)**	39/63 (62%)	1/9 (11%)	0/2	1/3	1/2
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
Conf. int. width	**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

TABLE 5 The different datasets involved in a simulation study

Dataset	Description and notes
Simulated	A dataset of size n_{obs} produced with some element of random-number generation and 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†	Dataset containing n_{sim} summaries of information from repetitions (e.g. $\hat{\theta}$ or indication of hypothesis rejection) for each combination of DGM, method and target (e.g. each estimand).
States	Datasets containing the start- and/or end-of-repetition random-number-generator states for each simulated dataset (see section 4.1).
Performance measures	Dataset containing estimated performance measures and Monte Carlo standard errors for each DGM, method and target.

†or corresponding summaries for non-estimand targets

some, a performance measure was given a name which its formula demonstrated to be misleading, emphasising the importance of clear terminology in simulation studies.

Description and computation 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 estimates and are subject to error. This fundamental feature of simulation studies does not seem to be widely appreciated, as previously noted(4). This means that first, as with any other empirical experiment, we must not act as if performance are known but present estimates of uncertainty (described in section 5); second, there are implications for choosing the number of repetitions.

4 | COMPUTATIONAL AND PROGRAMMING ISSUES IN SIMULATION STUDIES

In this section we discuss how to code 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

In what follows, 'seed' refers to input planted by the user; 'state' refers to the specific state of the random-number generator at a particular point in time.

All statistical packages capable of Monte Carlo simulation use a random-number generator. The numbers produced are in fact pseudo-random and follow a completely deterministic path, given the initial state. For example, in R and Stata, each time a number is 'drawn', the state of the random-number generator moves forward by one. So if a vector of n random numbers is generated, the state of the random number generator changes n times.

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 are, to all intents and purposes, truly random. 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 take off again from its landing state. Second, if a certain repetition results in some failure, such as non-convergence, the starting state 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 non-convergence 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 utilising 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). This is important, and the following chunk of pseudocode demonstrates the concept:

```
SET randomseed to #

FOR repetition 1 to n_sim

 STORE repetition and randomstate in StatesData[repetition]

 GENERATE simulated dataset

 ...

END FOR
```

When 'stream' random number generation is used (which invokes a 'jump-ahead' to allow simulations to be run in parallel), this practical advice applies *within* a stream(13). There are several user-written R packages allowing independent streams of random numbers. In Stata (from version 15), it is achieved with

```
. set rngstream #
```

prior to setting the seed.

The reason for this advice is to avoid unintended dependence between simulated datasets. We illustrate our caution with an anecdote: one method of recording the states for r repetitions is to set a seed and generate a vector of n_{sim} integers by generating a single random number, recording the state, generating another random number, recording the state, and so on. For the simulation itself, the seed for the ith repetition is set to the ith of the stored states. To clarify the problem, let $n_{\text{obs}} = n_{\text{sim}} = 5$ 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 five times) and proceeds. The second repetition then simulates x_2, which is made up of observations 2 to 5 from $i = 1$ plus one new value. The resulting draws of x for the five repetitions are then:

$$x_1 = (0.3488717, 0.2668857, 0.1366463, 0.0285569, \mathbf{0.8689333})$$
$$x_2 = (0.2668857, 0.1366463, 0.0285569, \mathbf{0.8689333}, 0.3508549)$$
$$x_3 = (0.1366463, 0.0285569, \mathbf{0.8689333}, 0.3508549, 0.0711051)$$
$$x_4 = (0.0285569, \mathbf{0.8689333}, 0.3508549, 0.0711051, 0.3233680)$$
$$x_5 = (\mathbf{0.8689333}, 0.3508549, 0.0711051, 0.3233680, 0.5551032)$$

Note the bold element on each line: The fifth element of x_1 is the first element of x_5 and appears in all repetitions. Only when $i > n$ is the draw of x independent of the first repetition. Such dependency in simulated data has implications for both performance measures and Monte Carlo SE's and is best avoided. (Note that this structure is not certain because the way input seeds map to states and vice versa is not 1:1.)

If the advice to set the seed once only is followed, there are implications for parallelisation, namely that it is inadvisable to parallelise *within* a run (unless streams are being used (13)). Say we wish to produce the estimates dataset for a given DGM via runs on two parallel processors. To be sure to avoid correlation between simulated datasets caused by one run 'jumping in' on the chain of random numbers used by the other, the landing state of the first run must be *known* for the starting seed of the second to be set. Stream random-number generators ensure that the starting seed of the second run is (a long way) ahead of the final state of the first. In the absence of streams, it is sensible to simply run all repetitions in one go.

When a simulation study uses different data-generating mechanisms, these may be run in parallel. Because performance measures will be computed separately for different data-generating mechanisms, jumping in is less of a problem.

Many programs execute methods involving some stochastic element. Examples include multiple imputation, the bootstrap, the g-computation formula, multistate models and Bayesian methods which 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. The implication is that many of the n_{sim} results will be highly correlated, if not identical, and results should not then be trusted. With the number of user-written packages available in R and Stata which may or may not rely on simulation somewhere, checking for such behaviour is worthwhile. A diagnostic for whether any random-number generation is used is to display the current state, 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, this is a cause for concern because it indicates that the seed is being set deterministically by the program.

4.2 | Start small and build up code

It is all too easy to obtain misleading results in a simulation study through very minor coding errors; see for example the comments in reference (31). Such errors are often detected when results are unexpected, for example, when bias appears much greater than theory or instincts suggest. One design implication is that methods with known properties should be included where possible as a check that these properties are exhibited. Mistakes in code are difficult to avoid. One straightforward and intuitive approach for minimising 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, the starting point could be to create one repetition under a large sample size. 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, in Stata, the `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*. Anecdotally, we have seen this syntax trip up good programmers. By checking the variance of a variable simulated by `rnormal` in a large simulated dataset, it will be obvious if it does not behave in the expected fashion. The simulation file can be built to include different data-generating mechanisms, methods or estimands, again

checking that behaviour is as expected. Using the above example again, if the basic data-generating mechanism used N(μ, 1), the issue with specifying standard deviations *vs.* variances would not be detected, but it would for other data-generating mechanisms.

Once satisfied that one large run is behaving sensibly, it is worth setting the sample size required 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 without the full set of simulations.

After thoroughly checking through and generalising 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. There are two reasons for this. First, programs may fail. If this happens in repetition 4,120 of 5,000, we will want to understand why. In this case, a record of the 4,120th state means we can reproduce the problematic dataset instantly.

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

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.

It is often 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 for two reasons. First, 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. Second, 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.

5 | ANALYSIS OF ESTIMATES DATA

5.1 | Checking the estimates data and preliminaries

This section describes the computation of performance measures and Monte Carlo standard errors. We advocate two preliminaries: checking for missing estimates and plots of the estimates data.

The number of missing estimates and missing SE's (for example due to non-convergence) are the first performance measures to assess. Ideally, a method returning missing values should be explored under failed runs (see section 4) and the code made more robust to ensure fewer or no failures.

Missing values in the *estimates* dataset pose a missing data problem regarding the analysis of other performance measures. It is extremely implausible that values are missing completely at random(32); estimates will usually be missing due to non-convergence so will likely depend on some characteristic/s of a given realisation of the data. When the 'method' being evaluated involves an 'analyst's procedure', where the model changes if the first-choice model does not converge, this reduces or removes missing values from the estimates data (see section 3.4).

If more than two methods are evaluated and one method 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 non-trivial 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.

Before undertaking a formal analysis of the estimates dataset it is sensible to undertake some exploratory analysis. In some cases this may be sufficient for analysis. For example, Kahan and Morris (33) aimed to show that balancing treatments in randomised trials induces a correlation between the mean outcomes in two groups. A simple simulation was used and individual estimates presented in a scatterplot of the mean in one group against the other. This was sufficient to demonstrate the issue. A second example can be found in (14), which assessed the performance of a two-stage procedure for analysing factorial trials. Although the procedure was technically unbiased, a histogram of $\hat{\theta}_i$ exhibited a bimodal distribution with modes equally spaced at either side of the truth θ but with almost no values of $\hat{\theta}_i$ close to θ!

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 DGM, 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 DGM, estimand and method, with the aim of identifying bivariate outliers.
3. Bivariate plots of $\hat{\theta}_i$ for one method *vs.* another for each DGM and estimand. The purpose here is to look for correlations between methods and any systematic differences. Where there are more than two methods being compared a graph of every method *vs.* every other or *vs.* the comparator can be useful.
4. A plot of confidence intervals ranked by the significance of the interval's test (as in figure 5). This is a means of understanding any issues with coverage.

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

We now describe common performance measures: properties they are designed to assess, computation and Monte Carlo SE's.

5.2 | Computing common performance measures and their standard errors

This section outlines some common performance measures, pros and cons, how they are computed and how Monte Carlo standard errors (MCSEs) are computed. We suppress the hat notation for performance measures, but emphasise that they are estimates.

The formulas for Monte Carlo SEs are particularly needed. In our review of simulation studies in *Statistics in Medicine* Volume 34, 93 did not mention Monte Carlo SEs for estimated performance measures.

The first performance measure of interest is often bias, which quantifies whether the estimator targets θ on average. Frequentist theory holds unbiasedness to be a key property. The calculation is

$$\text{Bias} = \frac{1}{n_{\text{sim}}} \sum_{i=1}^{n_{\text{sim}}} \hat{\theta}_i - \theta$$

$$\text{MCSE} = \sqrt{\frac{1}{n_{\text{sim}} - 1} \sum_{i=1}^{n_{\text{sim}}} (\hat{\theta}_i - \theta)^2}$$

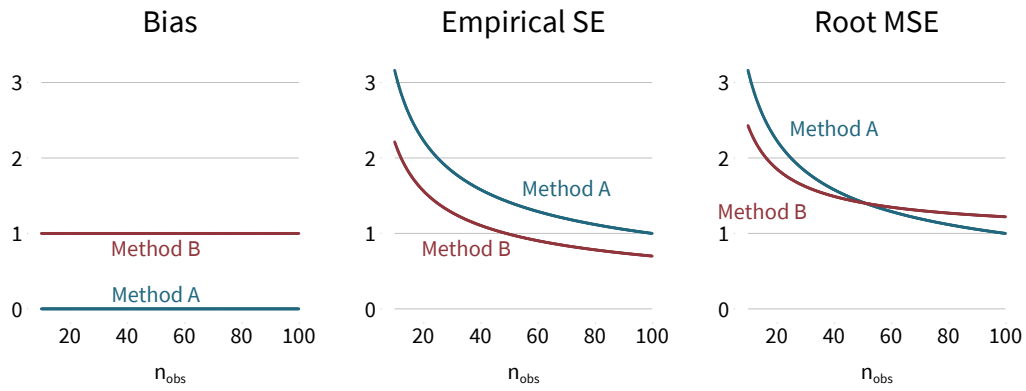
The mean of $\hat{\theta}_i$, $\bar{\theta}$ is often reported instead. This is calculated in the same way but without subtracting the constant θ, and so has the same MCSE. It is sometimes preferable to report the relative per cent bias, rather than absolute bias. If different values of θ are used for different data-generating mechanisms then per cent bias permits a more straightforward comparison across values. However, relative bias can be used only for $|\theta| > 0$. Lack of bias is only 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 depends only on $\hat{\theta}$ and does not require knowledge of θ. It estimates the long-run standard deviation of $\hat{\theta}$ over the n_{sim} repetitions.

$$\text{EmpSE} = \sqrt{\frac{1}{n_{\text{sim}} - 1} \sum_{i=1}^{n_{\text{sim}}} (\hat{\theta}_i - \bar{\theta})^2}$$

$$\text{MCSE} = \frac{\text{EmpSE}}{\sqrt{2(n_{\text{sim}} - 1)}}$$

FIGURE 1 Bias, variance and MSE for two methods. Method 1 is unbiased but imprecise; Method 2 is biased but more precise. The method with lower MSE depends on n.



The empirical SE is a measure of the precision or efficiency of the estimator of θ. Several other designations in common use; in our review we saw the terms 'empirical standard deviation', 'Monte Carlo standard deviation', 'observed SE', the 'sampling SE' and more.

When comparing methods, it may be of interest to investigate the relative precision of methods against a comparator. The precision of method B relative to method A is

$$\text{Relative \% increase in precision} = \frac{\text{Var}(\hat{\theta}_A)}{\text{Var}(\hat{\theta}_B)} = \left(\frac{\text{EmpSE}_A}{\text{EmpSE}_B}\right)^2$$

$$\text{MCSE} \simeq 2\frac{\text{Var}(\hat{\theta}_A)}{\text{Var}(\hat{\theta}_B)}\sqrt{\frac{1-\rho_{AB}^2}{n_{\text{sim}}-1}}$$

where ρ_{AB}^2 is the correlation of $\hat{\theta}_A$ and $\hat{\theta}_B$ (10). Note that if either method is biased, the relative precision should be interpreted with caution because an estimator which is biased towards the null can have a small empirical SE as a result of the bias: $\hat{\theta}/2$ necessarily has smaller empirical SE than $\hat{\theta}$.

A related measure, which also takes the true value of θ into account, is the mean squared error.

$$\text{MSE} = \frac{1}{n_{\text{sim}}}\sum_{i=1}^{n_{\text{sim}}}(\hat{\theta}_i - \theta)^2$$

$$\text{MCSE} = \sqrt{\frac{\sum_{i=1}^{n_{\text{sim}}}\left[(\hat{\theta}_i - \theta)^2 - \text{MSE}\right]^2}{n_{\text{sim}}(n_{\text{sim}}-1)}}$$

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, but the relative influence of bias and of variance on the MSE varies with n_{obs}, making generalisation of results difficult.

The problem is summarised in figure 1 , which depicts the bias, empirical standard error and root MSE (RMSE, favoured here because it is on the same scale as EmpSE) for two hypothetical methods. Method A is unbiased but imprecise (and so RMSE is simply the empirical SE), while method B is biased but more precise (as is often the case with biased methods, see for example (34)). For $n_{\text{obs}} > 50$, RMSE is lower for method B, but for $n_{\text{obs}} > 50$, RMSE is lower for method A. The lesson is that comparisons of MSE are more sensitive to sample size than comparisons of bias or empirical SE alone.

The average estimated SE, which we term the 'model SE', is

$$\text{ModSE} = \sqrt{\frac{1}{n_{\text{sim}}}\sum_{i=1}^{n_{\text{sim}}}\widehat{\text{Var}}(\hat{\theta}_i)}$$

$$\text{MCSE} \simeq \sqrt{\frac{\text{Var}[\widehat{\text{Var}}(\hat{\theta}_i)]}{4n_{\text{sim}} \times \widehat{\text{ModSE}}^2}}$$

This is computed on the variance, rather than standard error, scale because it is on this scale that standard theory yields unbiased estimates. Because the model SE targets the empirical SE, relative error in the model SE is an informative performance measure.

$$\text{Relative \% error in ModSE} = 100\left(\frac{\text{ModSE}}{\text{EmpSE}} - 1\right)$$

$$\text{MCSE} = 100\left(\frac{\text{ModSE}}{\text{EmpSE}}\right)\sqrt{\frac{\text{Var}[\widehat{\text{Var}}(\hat{\theta}_i)]}{4n_{\text{sim}} \times \widehat{\text{ModSE}}^4} + \frac{1}{2(n-1)}}$$

Coverage is a key property for the long-run frequentist behaviour of an estimator. It is defined as the probability that a confidence interval contains θ. For a two-sided interval, the coverage proportion is estimated by

$$\text{Coverage} = \frac{1}{n_{\text{sim}}}\sum_{i=1}^{n_{\text{sim}}} 1(\hat{\theta}_{\text{low},i} \leq \theta \leq \hat{\theta}_{\text{upp},i}) \quad (1)$$

$$\text{MCSE} = \sqrt{\frac{\text{Coverage} \times (1 - \text{Coverage})}{n_{\text{sim}}}} \quad (2)$$

Under-coverage 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) $\widehat{\text{Var}}(\hat{\theta}_i)$ is too variable. Over-coverage occurs as a result of ModSE $>$ EmpSE. This may occur either in the absence or presence of issues (i) and (iii).

Note that Neyman's original description of confidence intervals defined the property of *randomisation validity* as exactly $100(1-\alpha)\%$ of intervals containing θ (see (35, 36, 37)). *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 over- and under-coverage are regarded as similarly bad(38). 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 (see for example (36, 37)).

As noted previously, under-coverage may be due to bias, while under- or over-coverage may be due to ModSE $\neq$ EmpSE. We propose decomposing poor coverage into its causes with a new performance measure: 'bias-corrected coverage'. In essence, the bias of a method is eliminated from confidence intervals by studying coverage of confidence intervals for $\hat{\theta}$ rather than θ. Bias-corrected coverage is computed as follows:

$$\text{Bias-corrected coverage} = \frac{1}{n_{\text{sim}}}\sum_{i=1}^{n_{\text{sim}}} 1(\hat{\theta}_{\text{low},i} \leq \hat{\theta} \leq \hat{\theta}_{\text{upp},i})$$

with MCSE as for coverage.

Power is often of principal interest in simulation studies which target a null hypothesis, or when sample size requirements are being estimated 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 using θ and ModSE, or directly output, for example by a likelihood-ratio test or score test. The computation is as with other performance measures based on binary estimates data:

$$\text{Power} = \frac{1}{n_{\text{sim}}}\sum_{i=1}^{n_{\text{sim}}} 1(p_i \leq \alpha)$$

with MCSE as for coverage.

TABLE 6 Coverage conditional on size of ModSE

Approach	n_{sim} analysed	Coverage (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%)

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 in dose–response methods) and others that we have not yet thought of.

5.3 | Estimation of conditional performance

Note that the relevance of different performance measures will depend on the aims and targets of the simulation study. For example, with linear models, the issue of missing estimates will not be relevant.

Conditional performance will sometimes, but not always, be of interest. Consider the following scenario: $n_{obs} = 30$ 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, first for all repetitions, and then according to tertiles of the model SE. The results are given in table 6 . Note that coverage is seen to be below 95% for the lowest third of standard errors, above 95% for the highest third, and slightly above for the middle third. If coverage were correct regardless of any value of the SE, this may be viewed as positive, but poor conditional performance should be no cause for concern. Methods rarely, if ever, claim to provide good performance in this sense.

One issue with the example described is that, in analysing data, one would never know if the model SE lay in the lower, middle or higher third of all possible model SEs.

In fact, conditional evalutation of performance may sometimes be of central interest. This is particularly true of simulation studies which aim to evaluate study designs, for example where design decisions are made based on the early data. Two simulation studies in our sample explored two-stage procedures in randomised trials, where the estimand is selected after the first stage: the estimand is the treatment effect in a selected subpopulation (39) or the effect of a selected treatment (40). In both cases, estimators were designed to be conditionally unbiased. Kimani reported bias conditional on each possible selection of estimand (39), while Carreras averaged the bias across estimands (40). The former method is stricter and arguably more appropriate since, having selected an estimand, the observer is not interested in the other case(41).

Note that performance conditional on *true* (rather than sample-estimated) parameters which vary across data-generating mechanisms is of course where methods *should* be expected to provide good performance, and we do not recommend averaging over these.

We do not make any general recommendation other than to carefully consider whether to evaluate performance measures conditional on sample statistics.

5.4 | 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 A1 . Four simulation studies did not report n_{sim}. Common sample sizes are $n_{sim} = 500$ and $n_{sim} = 1,000$, as previously reported by Burton *et al.*(5). Of the 87 studies reporting n_{sim}, four provided some justification of the choice. The justifications were:

- 'To evaluate the asymptotic biases' (42)
- 'errors can be reduced by the large number of simulation replicates' (43)
- '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 (e.g., for the empirical coverage) is guaranteed to be smaller than 0.5.' (20)
- Marozzi gives an explicit derivation of Monte Carlo SE (44)

Clearly this is a sub-optimal state of affairs. For some more concrete justifications, see the worked illustrative example in section 7, Keogh and Morris(45), or Morris *et al.*(46)

There exist situations where only one repetition is necessary, particularly when investigating large-sample bias; see for example (16). 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 is given in section 5.2. Plugging in the expected coverage (for example 95%) and rearranging, we get

$$n_{\text{sim}} = \frac{\text{E(Coverage)} \times (1 - \text{E(Coverage)})}{(\text{MCSE}_{\text{req}})^2} \tag{3}$$

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%,

$$n_{\text{sim}} = \frac{95 \times 5}{0.5^2} = 1,900 \text{ repetitions.}$$

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 MCSE of 0.5% , (3) says that $n_{\text{sim}} = 10,000$ repetitions will achieve this MCSE.

A convenient feature of simulation studies is that Monte Carlo SE can be assessed and n_{sim} increased much more cheaply than with other empirical studies. The cost is computational time. For any such scenario it is important to have stored the landing state of the random-number generator (which can be set to take back off without fear of a collision with a previously visited state) or to use a different stream.

5.5 | 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 $\hat{\theta}$ for method A and $\hat{\theta}$ for the comparator; rather the bias for a method stands alone, being computed against θ. There are benchmarks for other performance measures as well, such as coverage (the nominal %) and precision (the Cramér–Rao lower bound(47, 48)).

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(49). They thus estimated θ for each of these estimands from a large simulated dataset.

In our review, nine of 74 studies which included some θ 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 $\hat{\theta}$, 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.

6 | REPORTING

6.1 | The 'methods' section

Our rationale for the ordering of elements in ADMEP is that this is usually the best order to report them in a methods section. If the simulation study has been planned and written out before it is executed then the methods section is largely written. This is a particularly helpful ordering for other researchers who might wish to replicate it.

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.* (50)). 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.*(51)), though if there are more methods than data-generating mechanisms it may be better to swap these (as in Hsu, Taylor and Hu (52)). 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.* (53)).

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 different methods (for example) should be arranged in adjacent rows or columns. But it may also be useful to compare across any of the other dimensions.

The second consideration regards presentation of Monte Carlo SEs, and this tends to confound the first. By presenting them next to performance measures, for examples in parentheses, the table becomes cluttered and hard to digest, obscuring interesting comparisons. For this reason, some authors report the maximum Monte Carle SE in the caption of tables (for example (54, 34)). Results should not be presented to a greater accuracy than is justified by the MCSE. In our review of Volume 34, seven articles presented Monte Carlo SEs for estimated performance measures: three in the text, two in a table, one in a graph, and one in a float caption.

The main advantage of displaying performance measures graphically is that it is easier to quickly spot patterns, particularly over dimensions which are not compared side-by-side. A second advantage is that it becomes possible to present raw data estimates (for example the $\hat{\theta}_i$) rather than performance measures summarising them (see for example figure 3 of (55)). In our experience, these plots are popular and intuitive ways to summarise the $\hat{\theta}_i$ and model SE's. Another example of a plot of estimates data is a histogram given in Kahan(14). This is particularly important as Bias $\simeq 0$, but the procedure he studies is inconsistent, and almost no $\hat{\theta}_i$ is close to θ. This is easily detected by plotting the estimates data. Thus, even if plots of estimates are not planned to be included in publications we urge their use in exploration of simulation results. The main disadvantages of graphical displays of results is that plots can be less space-efficient than tables, it is not possible to read off the exact numbers, and separate plots will usually be required for different performance measures.

Compared with tables, it is easier to accommodate display of Monte Carlo SE's directly in plots of performance measures, and this should be done, for example as 95% confidence intervals. However, the considerations about design for the most relevant comparisons apply similarly. Methods often have names that are hard to arrange side by side in a legible manner. It may thus be preferable to arrange methods in horizontal rows and performance measures moving horizontally in a style similar to a forest plot.

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. Rücker and Schwarzer present an approach to presenting 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(56). Their suggestion is to use 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(56). This is a useful method, but will not suit all designs. There are also challenges: how should the nesting of factors be determined? Should this be planned at the design stage or should it be decided on seeing results (e.g. where the top level of nesting is the factor with the largest influence on results)?

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

7 | WORKED ILLUSTRATIVE EXAMPLE

To make clear the ideas described in this article and demonstrate how they should 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, we include the files required to run the simulation in Stata.

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 for the ith patient

$$h_i(t) = h_0(t)\exp(X_i\theta), \tag{4}$$

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 of 1) 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 4, 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: $\lambda = 0.1, \gamma = 1$ ← both an exponential and a Weibull model
2: $\lambda = 0.1, \gamma = 1.5$ ← 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'.

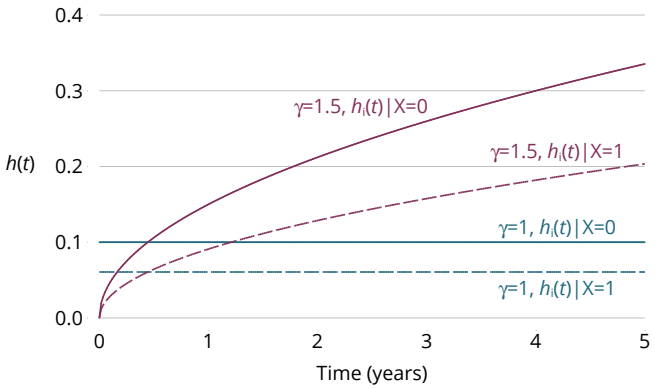
Methods: Each simulated dataset is analysed in three ways, using:

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.

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

FIGURE 2 Visualisation of the true hazard rate over follow-up time in the two DGMs. Blue (flat) lines are for the first data-generating mechanism where $\gamma = 1$; red curves are for the second, where $\gamma = 1.5$



randomised trial.

Performance measures: We will assess 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 $\text{Var}(\hat{\theta}) \leq 0.04$. We decide that we require MCSE lower than 0.005 on the estimate of bias. Given that

$$\text{MCSE} = \sqrt{\text{Var}(\hat{\theta})/n_{\text{sim}}},$$

this implies that we need 1,600 repetitions. If coverage of all methods is 95%, the implication of using $n_{\text{sim}} = 1,600$ is

$$\text{MCSE} = \sqrt{\frac{95 \times 5}{1,600}} = 0.54.$$

With 50% coverage, the MCSE is maximised at 1.25. We find this satisfactory and so proceed with $n_{\text{sim}} = 1,600$ (to be revised if, for example, $\text{SD}(\hat{\theta}) > 0.2$).

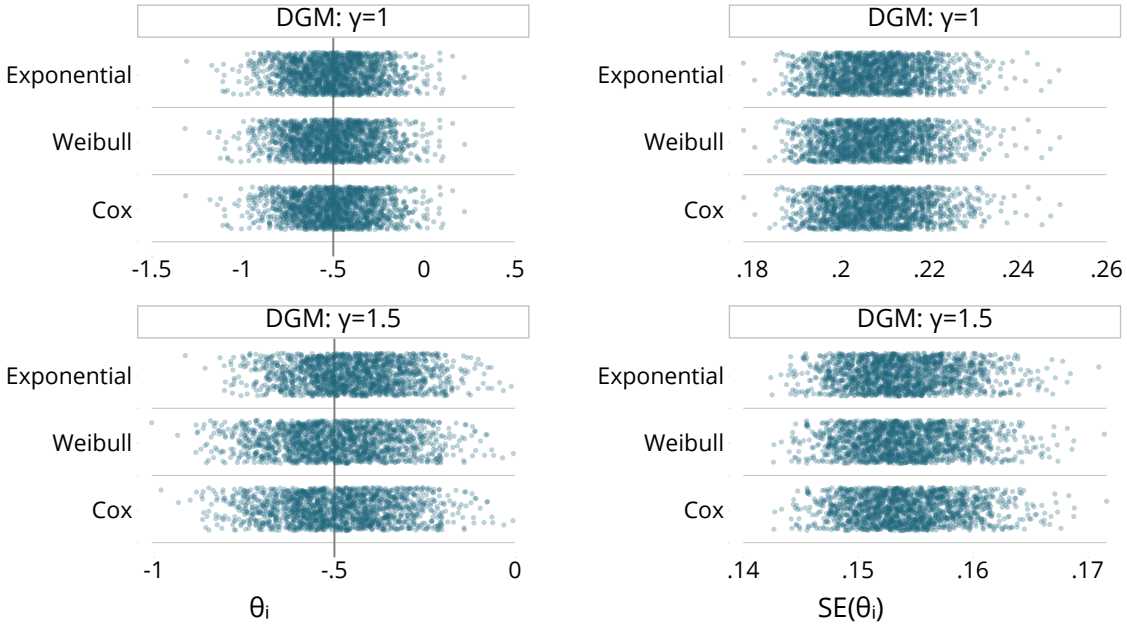
7.2 | Exploration and visualisation of results

Before computing performance measures 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. 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 variance of $\hat{\theta}_i$ 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 lower for the upper panel ($\gamma = 1$) than the lower ($\gamma = 1.5$) but there is very little to choose between the methods.

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 (*i.e.* they are estimated in the same simulated dataset) and are compared to a line of $y = x$. This is done in figure 4 , for the second data-generating mechanism only ($\gamma = 1.5$), which is interesting 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, they are very slightly larger for the exponential model.

Figure 5 is a new visualisation, the 'zip plot', which helps to understand coverage by viewing the confidence intervals directly. For each data-generating mechanism and method, the confidence intervals are centile-ranked according to their significance against the null that $\theta = 0.5$. This ranking is used for the vertical axis and is plotted against the intervals themselves. Intervals

FIGURE 3 Plot of the 1,600 $\hat{\theta}_i$ (left panels) and $\widehat{\mathrm{SE}}(\hat{\theta})_i$ (right panels) by data-generating mechanisms, for the three analysis methods. The vertical axis is repetition number, to provide some separation between points.



for which a two-sided test yields $p < 0.05$ are coloured purple (towards the top), while the remainder are in blue (at the bottom). When a method has 95% coverage, the colour of the intervals switches at 95 on the vertical axis. Finally, the yellow horizontal lines are Monte Carlo 95% confidence intervals for per cent coverage.

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 which do not cover θ. This indicates that the model SEs must overestimate the empirical SE, because coverage is adequate despite bias. Figure 5 helps to make such a feature clear.

7.3 | Analysis of example

The previous section suggested some useful exploratory analyses. Next, we compute the performance measures of interest and present them in a table in which (we hope) the ADMEP 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 MCSEs in parantheses at a smaller point size than the estimate); there is only one estimand.

The results in table 7 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 non-negligible. 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.

FIGURE 4 Comparison of estimates for methods when $\gamma = 1.5$, where each point represents one repetition. Upper triangle displays $\hat{\theta}_i$; lower triangle displays $\widehat{\text{SE}}(\hat{\theta}_i)$.

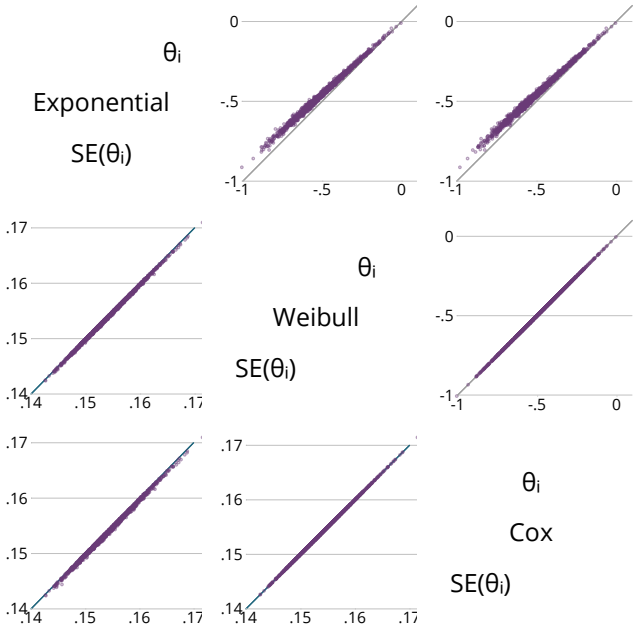
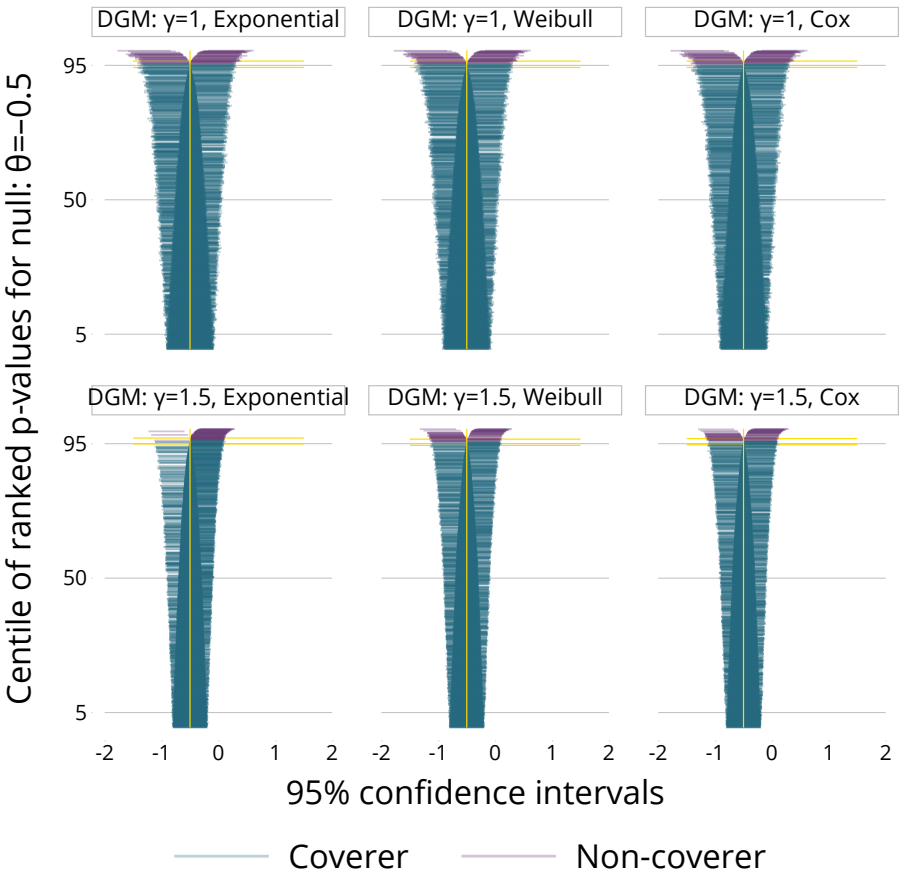


TABLE 7 Estimates of performance measures of interest (MCSEs in parentheses)

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-corrected coverage	$\gamma = 1$	95.6%	(0.5)	95.3%	(0.5)	95.4%	(0.5)
	$\gamma = 1.5$	**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 gain *vs.* Weibull	$\gamma = 1$	0.2%	(0.1)	0	(–)	0.3%	(0.1)
	$\gamma = 1.5$	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 Model SE	$\gamma = 1$	−0.7%	(1.8)	−0.7%	(1.8)	−0.5%	(1.8)
	$\gamma = 1.5$	**11.5%**	(2.0)	1.7%	(1.8)	2.1%	(1.8)

Looking at table 7 , the MCSEs of the performance measures are all acceptable and so we would be happy to draw conclusions about the methods based on the 1,600 repetitions.

FIGURE 5 'Zip plot' of the 1,600 confidence intervals for each data-generating mechanism and analysis method. The vertical axis is the centile of the two-sided p-value against $H_0 : \theta = -0.5$ associated with the confidence interval.



7.4 | Conclusions of example

When an exponential model is misspecified the hazard ratio can be biased. Probably not by much. Further research is needed (using different values of n_{obs} and of γ).

8 | CONCLUDING REMARKS

Simulation studies are an invaluable tool for research into statistical methods, evidenced by the large proportion of Volume 34 *Statistics in Medicine* articles whose conclusions relied in part on simulation studies. Because methods promoted may be used in medical research, 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 ADMEP 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 which 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(57). Reports

of simulation studies are now beginning to explicitly use the ADMEP structure; see Thompson *et al.*(58), Sayers *et al.*(59) and Morris *et al*(46).

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` automates this process for commonly used performance measures(10).

8.1 | Simulation studies evaluating methods of widespread and general interest

One issue that arises with choosing data-generating mechanisms is that different research groups may reach different conclusions. Methods are created by researchers who are most concerned with handling the specific problems they have seen. From those researchers' perspectives, their simulation scenarios will be appropriate (this does not imply 'cheating').

It is worth noting two areas where different research groups have been working on similar problems, and very different strategies have been taken. The first concerns heterogeneity variance estimators in meta-analysis, and the second, methods for handling incomplete data with a multilevel structure.

Langan, Higgins and Simmonds provide a comprehensive review of simulation studies comparing heterogeneity variance estimators for meta-analysis(60). A bewildering array of methods – and of simulation studies – exist for this problem. The authors noted that in general, 'results were based on data that do not represent meta-analyses observed in practice,' and described 'conflict(s) of interest in these non-independent studies'(60). Their article ends by saying that they, not having invented any of the methods, are conducting further simulation studies to address these limitations(60). Petropolou and Mavridis then recently published an article attempting this(61).

Several research groups have been developing methods for handling incomplete data in datasets with a multilevel structure since around 2010. In summer 2013, a meeting was held at LSHTM to come exchange ideas. There were several points of confusion among presenters about the design of simulation studies, particularly regarding data-generating mechanisms. It was not obvious why certain methods had fared well in some simulation studies but not others. As a result, the day stimulated collaboration, leading to simulation studies involving representatives from all the groups(62). This can act as a base from which future simulation studies can be designed.

Both situations highlight the potential value of open repositories of data and code, of clear description of the applied settings for which methods are intended, and a description of situations in which new methods should be tested.

8.2 | Final remark

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

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 one or more person benefits financially.

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 an MRC New Investigator Research Grant (grant number MR/P015433/1).

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

References

[1] Feiveson A. H.. Power by simulation. *The Stata Journal.* 2002;2(2):107–124.

[2] Hoaglin D. C., Andrews D. F.. The reporting of computation-based results in statistics. *The American Statistician.* 1975;29(3):122–126.

[3] Hauck W. W., Anderson S.. A survey regarding the reporting of simulation studies. *The American Statistician.* 1984;38(3):214–216.

[4] Ripley B. D.. *Stochastic Simulation*. New York: Wiley; 1987.

[5] Burton A., Altman D. G., Royston P., Holder R. L.. The design of simulation studies in medical statistics. *Statistics in Medicine.* 2006;25(24):4279–4292.

[6] Koehler E., Brown E., Haneuse . On the assessment of Monte Carlo error in simulation-based statistical analyses. *The American Statistician.* 2009;63:155–162.

[7] Morgan B. J. T.. *Elements of simulation*. Boca Raton: Chapman & Hall/CRC; 1995.

[8] Chang M.. *Monte Carlo simulation for the pharmaceutical industry : concepts, algorithms, and case studies*. CRC Press; 2011.

[9] Díaz-Emparanza I.. Is a small Monte Carlo analysis a good analysis?. *Statistical Papers.* 2002;43(4):567–577.

[10] White I. R.. simsum: Analyses of simulation studies including Monte Carlo error. *Stata Journal.* 2010;10(3):369–385.

[11] Smith M. K., Marshall A.. Importance of protocols for simulation studies in clinical drug development. *Statistical Methods in Medical Research.* 2011;20(6):613–622.

[12] Crowther M. J., Lambert P. C.. Simulating biologically plausible complex survival data. *Statistics in Medicine.* 2013;32(23):4118–4134.

[13] Haramoto H., Matsumoto M., Nishimura T., Panneton F., L'Ecuyer P.. Efficient jump ahead for F2-linear random number generators. *INFORMS Journal on Computing.* 2008;20(3):385–390.

[14] Kahan B. C.. Bias in randomised factorial trials. *Statistics in Medicine.* 2013;32(26):4540–4549.

[15] Kenward M. G., Roger J. H.. Small sample inference for fixed effects from restricted maximum likelihood. *Biometrics.* 1997;53:983–997.

[16] White I. R.. 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). *Statistics in Medicine.* 1997;16(22):2619–2620.

[17] White I. R., Thompson S. G.. Adjusting for partially missing baseline measurements in randomised trials. *Statistics in Medicine.* 2005;24(7):993–1007.

[18] Hughes R. A., Sterne J. A. C., Tilling K.. Comparison of imputation variance estimators. *Statistical Methods in Medical Research.* 2014;:n/a+.

[19] Morris T. P., White I. R., Royston P.. Tuning multiple imputation by predictive mean matching and local residual draws. *BMC Medical Research Methodology.* 2014;14(1):75+.

[20] Kuss O.. Statistical methods for meta-analyses including information from studies without any events—add nothing to nothing and succeed nevertheless. *Statistics in Medicine.* 2015;34(7):1097–1116.

[21] Campbell H., Dean C. B.. The consequences of proportional hazards based model selection. *Statistics in Medicine.* 2014;:1042–1056.

[22] Robins J. M., Wang N.. Inference for imputation estimators. *Biometrika.* 2000;87(1):113–124.

[23] Reiter J. P.. Multiple imputation when records used for imputation are not used or disseminated for analysis. *Biometrika.* 2008;95(4):933–946.

[24] Crowther M. J., Look M. P., Riley R. D.. Multilevel mixed effects parametric survival models using adaptive Gauss–Hermite quadrature with application to recurrent events and individual participant data meta-analysis. *Statistics in Medicine.* 2014;33(22):3844–3858.

[25] Hauck W. W., Anderson S., Marcus S. M.. Should we adjust for covariates in nonlinear regression analyses of randomized trials?. *Controlled Clinical Trials.* 1998;19:249–256.

[26] Zhang Z.. Estimating a marginal causal odds ratio subject to confounding. *Communications in Statistics - Theory and Methods.* 2008;38(3):309–321.

[27] Chaurasia A., Harel O.. Partial F-tests with multiply imputed data in the linear regression framework via coefficient of determination. *Statistics in Medicine.* 2015;34(3):432–443.

[28] Wu C., Shi X., Cui Y., Ma S.. A penalized robust semiparametric approach for geneâĂŞenvironment interactions. *Statistics in Medicine.* 2015;34(30):4016–4030.

[29] Ferrante L., Skrami E., Gesuita R., Cameriere R.. Bayesian calibration for forensic age estimation. *Statistics in Medicine.* 2015;34(10):1779–1790.

[30] Zhang Z., Wang C., Troendle J. F.. Optimizing the order of hypotheses in serial testing of multiple endpoints in clinical trials. *Statistics in Medicine.* 2015;34(9):1467–1482.

[31] Bartlett J.. *Combining bootstrapping with multiple imputation.* 2016.

[32] Rubin D. B.. Inference and missing data. *Biometrika.* 1976;63:581–592.

[33] Kahan B. C., Morris T. P.. Improper analysis of trials randomised using stratified blocks or minimisation. *Statistics in Medicine.* 2012;31(4):328–340.

[34] White I. R., Royston P.. Imputing missing covariate values for the Cox model.. *Statistics in Medicine.* 2009;28(15):1982–1998.

[35] Neyman J.. On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Selection. *Journal of the Royal Statistical Society, Series A.* 1934;97(4):558–625.

[36] Meng X. L.. Multiple-imputation inferences with uncongenial sources of input. *Statistical Science.* 1994;9:538–558.

[37] Rubin D. B.. Multiple imputation after 18+ years. *Journal of the American Statistical Association.* 1996;91(434):473–489.

[38] Morris T. P.. Rank minimization with a two-step analysis should not replace randomization in clinical trials. *Journal of Clinical Epidemiology.* 2012;65(7):810–811.

[39] Kimani P. K., Todd S., Stallard N.. Estimation after subpopulation selection in adaptive seamless trials. *Statistics in Medicine.* 2015;34(18):2581–2601.

[40] Carreras M., Gutjahr G., Brannath W.. Adaptive seamless designs with interim treatment selection: a case study in oncology. *Statistics in Medicine.* 2015;34(8):1317–1333.

[41] Efron B., Hastie T.. *Computer age statistical inference.* Cambridge University Press; 2016.

[42] Taguri M., Chiba Y.. A principal stratification approach for evaluating natural direct and indirect effects in the presence of treatment-induced intermediate confounding. *Statistics in Medicine.* 2015;34(1):131–144.

[43] Li P., Redden D. T.. Small sample performance of bias-corrected sandwich estimators for cluster-randomized trials with binary outcomes. *Statistics in Medicine.* 2015;34(2):281–296.

[44] Marozzi M.. Multivariate multidistance tests for high-dimensional low sample size case-control studies. *Statistics in Medicine.* 2015;34(9):1511–1526.

[45] Keogh R. H., Morris T. P.. *Multiple imputation in Cox regression when there are time-varying effects of exposures.* 2017.

[46] Morris T. P., Fisher D. J., Kenward M. G., Carpenter J. R.. Meta-analysis of Gaussian individual patient data: two stage or not two stage?. *Statistics in Medicine.* in-press 2017;.

[47] Cramér H. C.. *Mathematical Methods of Statistics*. Princeton, NJ: Princeton University Press; 1946.

[48] Rao C. R.. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society.* 1945;37:81–91.

[49] Williamson E. J., Forbes A., White I. R.. Variance reduction in randomised trials by inverse probability weighting using the propensity score. *Statistics in Medicine.* 2014;33(5):721–737.

[50] He X., Whitmore G. A., Loo G. Y., Hochberg M. C., Lee M-L T.. A model for time to fracture with a shock stream superimposed on progressive degradation: the Study of Osteoporotic Fractures. *Statistics in Medicine.* 2015;34(4):652–663.

[51] Chen Y., Hong C., Riley R. D.. An alternative pseudolikelihood method for multivariate random-effects meta-analysis. *Statistics in Medicine.* 2015;34(3):361–380.

[52] Hsu C. H., Taylor J. M. G., Hu C.. Analysis of accelerated failure time data with dependent censoring using auxiliary variables via nonparametric multiple imputation. *Statistics in Medicine.* 2015;34(19):2768–2780.

[53] Alonso A., Milanzi E., Molenberghs G., Buyck C., Bijnens L.. A new modeling approach for quantifying expert opinion in the drug discovery process. *Statistics in Medicine.* 2015;34(9):1590–1604.

[54] Seaman S. R., Bartlett J. W., White I. R.. Multiple imputation of missing covariates with non-linear effects and interactions: an evaluation of statistical methods.. *BMC Medical Research Methodology.* 2012;12(1):46+.

[55] Lambert P. C., Dickman P. W., Rutherford M. J.. Comparison of different approaches to estimating age standardized net survival.. *BMC Medical Research Methodology.* 2015;15(1):64+.

[56] Rücker G., Schwarzer G.. Presenting simulation results in a nested loop plot. *BMC Medical Research Methodology.* 2014;14(1):129+.

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

[57] Austin P. C., Stuart E. A.. Optimal full matching for survival outcomes: a method that merits more widespread use. *Statistics in Medicine.* 2015;34(30):3949–3967.

[58] Thompson J. A., Fielding K. L., Davey C., Aiken A. M., Hargreaves J. R., Hayes R. J.. Bias and inference from misspecified mixed-effect models in stepped wedge trial analysis. *Statistics in Medicine.* 2017;36(23):3670–3682.

[59] Sayers A., Crowther M. J., Judge A., Whitehouse M. R., Blom A. W.. Determining the sample size required to establish whether a medical device is non-inferior to an external benchmark. *BMJ Open.* 2017;7(8):e015397+.

[60] Langan D., Higgins J. P. T., Simmonds M.. Comparative performance of heterogeneity variance estimators in meta-analysis: a review of simulation studies. *Res. Syn. Meth.*. 2017;8(2):181–198.

[61] Petropoulou M., Mavridis D.. A comparison of 20 heterogeneity variance estimators in statistical synthesis of results from studies: a simulation study. *Statistics in Medicine.* 2017;36(27):4266–4280. Assess 'mean absolute error' and claim it is bias.

[62] Audigier V., White I. R., Jolani S., et al. *Multiple imputation for multilevel data with continuous and binary variables.* 2017.

APPENDIX A: REVIEW: SUMMARY OF INFORMATION AROUND DATA-GENERATION.

Figure A1 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, but the largest number being 600 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: $1,000$, 500 and $10,000$.

Figure A2 shows the number of methods evaluated by the simulation studies included in the review. The majority evaluated few methods (with four 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.

Figure **??** shows the number of estimands evaluated by the simulation studies included in the review. In general, there were few, with a single estimand the most common.

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 Results of *Statistics in Medicine* Volume 34 review for data-generating mechanisms: (a) number of data-generating mechanisms used; (b) number of factors varied across data-generating mechanisms; (c) how factors were varied (if >1 factor); (d) number of repetitions n_{sim}.

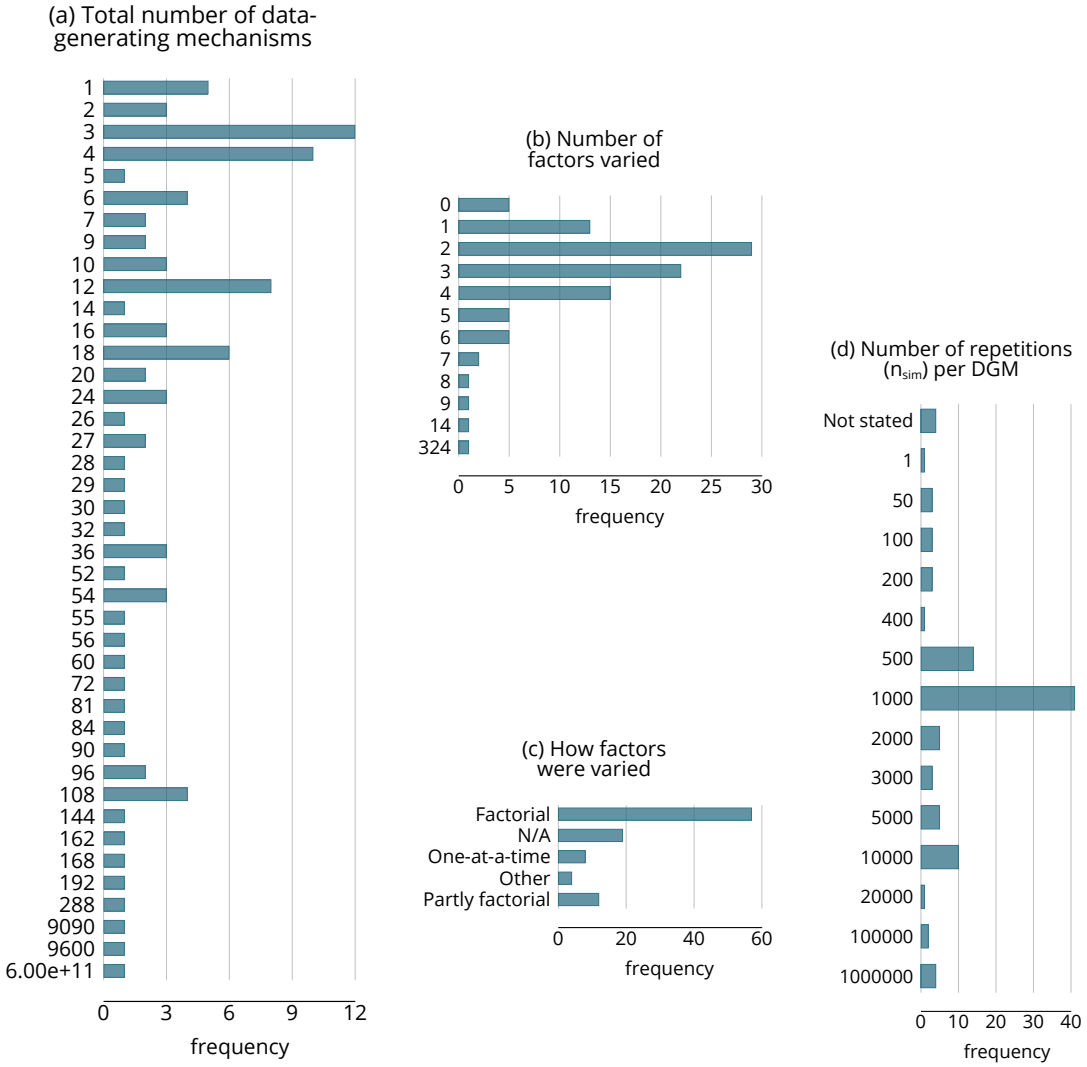


FIGURE A2 Results of *Statistics in Medicine* Volume 34 review for number of methods evaluated.

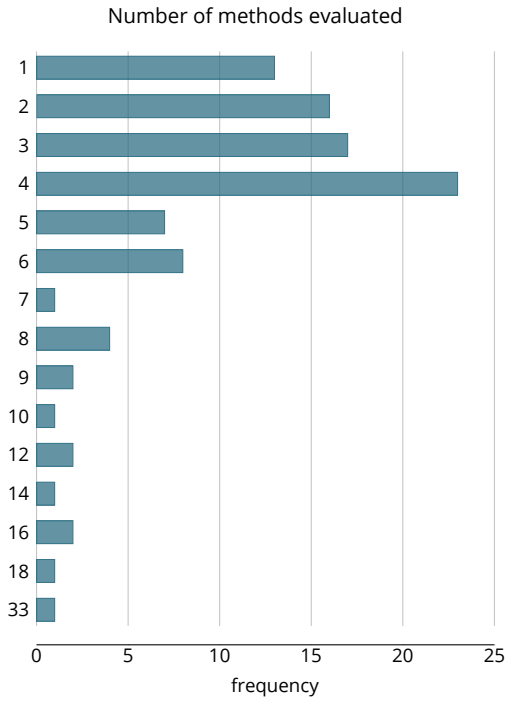


FIGURE A3 Results of *Statistics in Medicine* Volume 34 review for number of estimands evaluated in a simulation study.