JDSSV

Journal of Data Science, Statistics, and Visualisation

December 2021, Volume I, Issue IV. *doi: 10.52933/jdssv.v1i4.9*

INTEREST: INteractive Tool for Exploring REsults from Simulation sTudies

Alessandro Gasparini
University of Leicester

Tim P. Morris
MRC Clinical Trials Unit at UCL

Michael J. Crowther
University of Leicester

Abstract

Simulation studies allow us to explore the properties of statistical methods. They provide a powerful tool with a multiplicity of aims; among others: evaluating and comparing new or existing statistical methods, assessing violations of modelling assumptions, helping with the understanding of statistical concepts, and supporting the design of clinical trials. The increased availability of powerful computational tools and usable software has contributed to the rise of simulation studies in the current literature. However, simulation studies involve increasingly complex designs, making it difficult to provide all relevant results clearly. Dissemination of results plays a focal role in simulation studies: it can drive applied analysts to use methods that have been shown to perform well in their settings, guide researchers to develop new methods in a promising direction, and provide insights into less established methods. It is crucial that we can digest relevant results of simulation studies. Therefore, we developed **INTEREST**: an *INteractive Tool for Exploring REsults from Simulation sTudies*. The tool has been developed using the **Shiny** framework in R and is available as a web app or as a standalone package. It requires uploading a tidy format dataset with the results of a simulation study in R, Stata, SAS, SPSS, or comma-separated format. A variety of performance measures are estimated automatically along with Monte Carlo standard errors; results and performance summaries are displayed both in tabular and graphical fashion, with a wide variety of available plots. Consequently, the reader can focus on simulation parameters and estimands of most interest. In conclusion, **INTEREST** can facilitate the investigation of results from simulation studies and supplement the reporting of results, allowing researchers to share detailed results from their simulations, readers to explore them freely.

Keywords: simulation study, Monte Carlo, visualisation, reporting, R, **Shiny**, replicability.

1. Background

Monte Carlo simulation studies are computer experiments based on generating pseudo-random observations from a known truth. Statisticians usually mean *Monte Carlo simulation study* when they say *Simulation study*; throughout this article, we will just use *simulation study* but this encapsulates Monte Carlo simulation studies. Simulation studies have several applications and represent an invaluable tool for statistical research nowadays: in statistics, establishing properties of current methods is key to allow them to be used – or not – with confidence. Sometimes it is not possible to derive exact analytical properties; for example, a large sample approximation may be possible, but evaluating the approximation in finite samples is required. Approximations often require assumptions as well: what are the consequences of violating such assumptions? Monte Carlo simulation studies come to the rescue and can help to answer these questions. They also can help answer questions such as: Is an estimator biased in a finite sample? What are the consequences of model misspecification? Do confidence intervals for a given parameter achieve the advertised/nominal level of coverage? How does a newly developed method compare to an established one? What is the power to detect a desired effect size under complex experimental settings and analysis methods?

Simulation studies are being used increasingly in a wide variety of settings. For instance, searching on the database of peer-reviewed research literature Scopus (`https://www.scopus.com`) with the query string `TITLE-ABS-KEY ("simulation study") AND SUBJAREA (math)` yields more than 30000 results with a 20-fold increase during the last 30 years, from 148 documents in 1989 to 3185 in 2019 (Figure 1). The increased availability of powerful computational tools and ready-to-use software to researchers has surely contributed to the rise of simulation studies in the current literature.

Despite the popularity of simulation studies, they are often poorly designed, analysed, and reported. Morris et al. (2019) reviewed 100 research articles published in Volume 34 of *Statistics in Medicine* (2015) with at least one simulation study and found that information on data-generating mechanisms (DGMs), number of repetitions, software, and estimands were often lacking or poorly reported, making critical appraise and replication of published studies a difficult task. Another aspect of simulation studies that is often poorly reported or not reported at all is the Monte Carlo error of estimated performance measures, defined as the standard error of estimated performance, owing to the fact that a finite number of repetitions are used and so performance is estimated with uncertainty. Monte Carlo errors play an important role in understanding the role of chance in the results of simulation studies and have been showed to be severely under reported (Koehler et al. 2009).

The possibility of independently verifying results from scientific studies is a fundamental aspect of science (Laine et al. 2007); as a consequence, several reporting guidelines have emerged under the banner of the EQUATOR Network (`http://www.equator-`

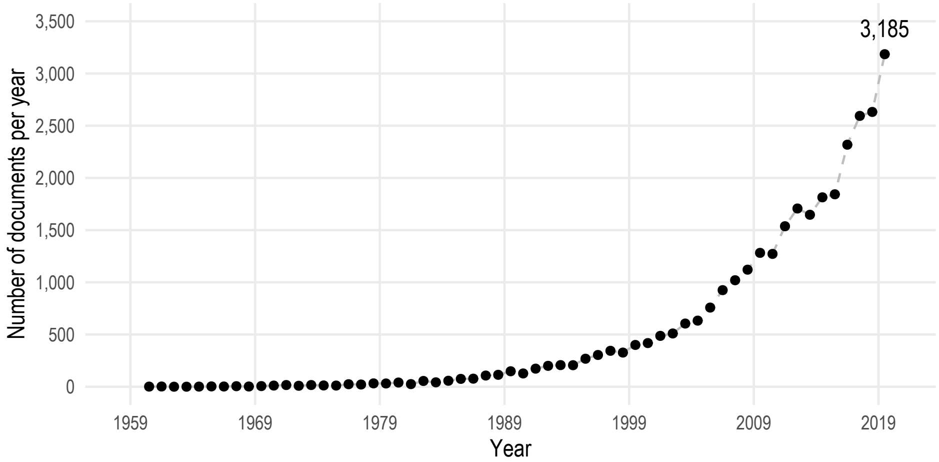


Figure 1: Trend in published documents on simulation studies from 1960 onwards. The number of documents was identified on Scopus via the search key `TITLE-ABS-KEY ("simulation study") AND SUBJAREA (math)`, and the number of documents identified in 2019 is labelled on the plot.

`network.org`) (Schulz et al. 2010; von Elm et al. 2007). Despite similar calls for harmonised reporting to allow for greater reproducibility in the area of computation science (Peng 2011) and several articles advocating for more rigour in specific aspects of simulation studies (Hoaglin and Andrews 1975; Hauck and Anderson 1984; Díaz-Emparanza 2002; Burton et al. 2006; White 2010; Smith and Marshall 2011), design and reporting guidelines for simulation studies are generally lacking in the statistical literature, with a few examples in the area of structural equation modelling (Bandalos and Gagné 2012; Boomsma 2013). Morris et al. (2019) introduced the ADEMP framework (Aims, Data-generating mechanisms, Estimands, Methods, Performance measures) aiming to fill precisely that gap. In the *Reporting* section they compared the several ways of reporting results that they observed in their reviews, including results in text for small simulation studies, tabulating and plotting results, and even the nested-loop plot proposed by Rücker and Schwarzer for fully-factorial simulation studies with many data-generating mechanisms (Rücker and Schwarzer 2014). They concluded by arguing that *there is no correct way to present results, but we encourage careful thought to facilitate readability, considering the comparisons that need to be made.*

As outlined in Spiegelhalter et al. (2011), there is little experimental evidence on how different types of visualisations are perceived; despite that, they highlight the ease of improving understanding via interactive visualisations that can be adjusted by the user to best fit specific requirements. The recent advent of tools such as Data-Driven Documents (**D**$^{\mathbf{3}}$, or **D3.js**) (Bostock et al. 2011) and **Shiny** (Chang et al. 2019) has further facilitated the development of interactive visualisations.

The increased availability of powerful computational tools has not only contributed to a rise in the popularity of simulation studies, it has also allowed researchers to simulate an ever-growing number of data-generating mechanisms and include several estimands and methods to compare: up to 4.2×10^{10}, 32, and 33, respectively, in the aforementioned review (Morris et al. 2019). With a large number of data-generating mechanisms, estimands, or methods, analysing and reporting the results of a simulation study becomes cumbersome: What results shall we focus on so as not to bewilder readers? Which estimands and methods should we include in our tables and plots? How should we plot or tabulate several data-generating mechanisms at once?

In an attempt to address these questions, we developed **INTEREST**, an *INteractive Tool for Exploring REsults from Simulation sTudies*. **INTEREST** is a browser-based interactive tool, and it requires first uploading a dataset with results from a simulation study; then, it estimates performance measures and it displays a variety of tables and plots automatically. The user can focus on specific data-generating mechanisms, estimands, and methods: tables and plots are updated automatically. This article will introduce the implementation details of **INTEREST** in the *Implementation* section and the main features in the *Results and discussion* section, where we will further discuss its relevance. We also present a case study to motivate the use of INTEREST and illustrate its use in practice. Finally, we conclude the manuscript with some remarks in the *Conclusions* section.

2. Implementation

INTEREST was developed using the free statistical software R (R Core Team 2020) and the R package **Shiny** (Chang et al. 2019). **Shiny** is an R package (and framework) that allows building interactive web apps straight from within R: the resulting applications can be hosted online, embedded in reports and dashboards, or just run as standalone apps.

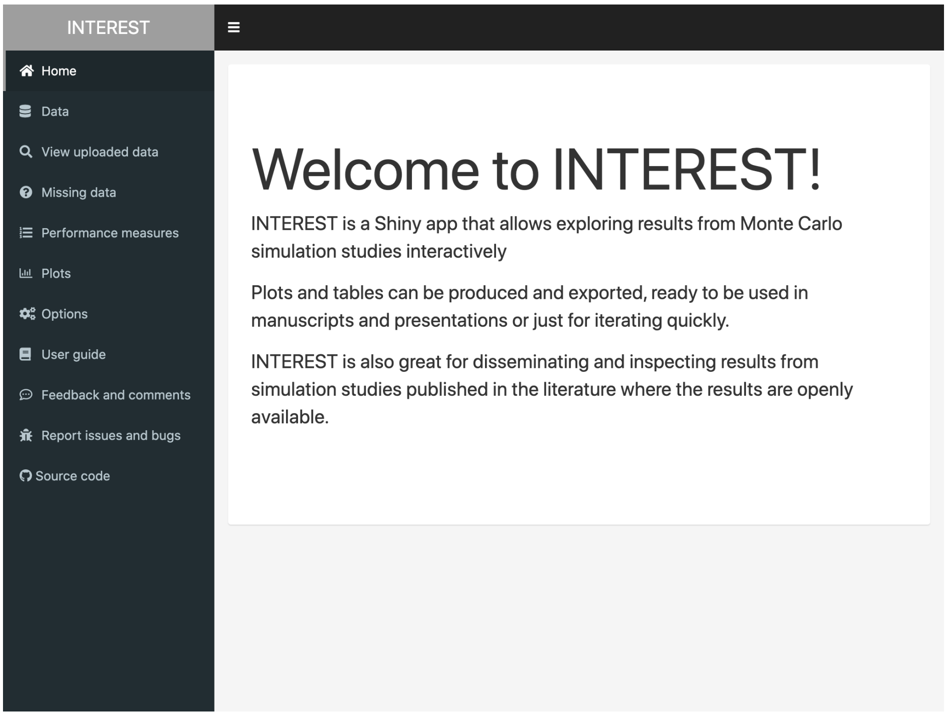
The front-end of **INTEREST** has been built using the **shinydashboard** package (Chang and Borges Ribeiro 2018); **shinydashboard** is based upon **AdminLTE** (`https://adminlte.io/`), an open-source admin control panel built on top of the Bootstrap framework (Version 3.x) and released under the MIT license.

The back-end functionality of **INTEREST** is published as a standalone R package named **rsimsum** for easier long-term maintainability (Gasparini 2018); **rsimsum** is freely available on the Comprehensive R Archive Network (CRAN) under the GNU General Public License Version 3 (`https://www.gnu.org/licenses/gpl-3.0`).

INTEREST is available as an online application and as a standalone version for offline use. The online version is hosted at `https://interest.shinyapps.io/interest/`, and can be accessed via any web browser on any device (desktop computers, laptops, tablets, smartphones, etc.). The standalone offline version can be obtained from GitHub (`https://github.com/ellessenne/interest`) and can be run on any desktop computer and laptop with a local instance of R; if required, R can be downloaded for free from the website of the R project (R Core Team 2020). INTEREST (as **rsimsum**) is published under the GNU General Public License Version 3.

3. Results and discussion

The main interface of **INTEREST** is presented in Figure 2. The interface is composed of a main area on the right and a navigation bar on the left; the navigation bar includes sub-menus for customising plots or modifying the default behaviour of **INTEREST**. We now introduce and describe the functionality of the application.



Welcome to INTEREST!

INTEREST is a Shiny app that allows exploring results from Monte Carlo simulation studies interactively

Plots and tables can be produced and exported, ready to be used in manuscripts and presentations or just for iterating quickly.

INTEREST is also great for disseminating and inspecting results from simulation studies published in the literature where the results are openly available.

Figure 2: Homepage of **INTEREST**. On the left, the navigation bar with sub-menus useful to tune the default behaviour of the app. On the right, the main window of **INTEREST**.

3.1. Data

The use of **INTEREST** starts by providing a tidy dataset (also known as long format, with variables in columns and observations in rows (Wickham 2014); an example of tidy data is included in Table 1) with results from a simulation study via the *Data* tab from the side menu. A dataset can be provided to **INTEREST** in three different ways:

1. The user can upload a dataset. The uploaded file can be a comma-separated file (`.csv`), a Stata dataset (version 8-15, `.dta`), an SPSS dataset (`.sav`), a SAS dataset (`.sas7bdat`), or an R serialised object (`.rds`); the format will be inferred automatically from the extension of the uploaded file, and the auto-detection is case-insensitive. It is also possible to upload compressed files (ending in `.gz`, `.bz2`, `.xz`, or `.zip`) that are automatically decompressed. The maximum supported file size is 100MB.

2. The user can provide a URL link to a dataset hosted elsewhere. All considerations relative to the file format from point (1) are also valid here.

3. Finally, the user can paste a dataset (e.g., from Microsoft Excel) in a text box. The pasted data is assumed to be tab-separated.

If users stored the results of their simulation study in a different format, we recommend using one of the readily available tools (e.g., the `pivot_*` functions from the **tidyr** package in R or the `reshape` command in Stata) to reshape the data before uploading it to **INTEREST**.

Table 1: Example of dataset in tidy format, with each row identifying a repetition for each combination of data-generating mechanism and analytical method.

Repetition	DGM	Method	Estimate
1	1	1	$\hat{\theta}_{1,1,1}$
2	1	1	$\hat{\theta}_{2,1,1}$
3	1	1	$\hat{\theta}_{3,1,1}$
1	2	1	$\hat{\theta}_{1,2,1}$
2	2	1	$\hat{\theta}_{2,2,1}$
3	2	1	$\hat{\theta}_{3,2,1}$
1	1	2	$\hat{\theta}_{1,1,2}$
2	1	2	$\hat{\theta}_{2,1,2}$
3	1	2	$\hat{\theta}_{3,1,2}$
1	2	2	$\hat{\theta}_{1,2,2}$
2	2	2	$\hat{\theta}_{2,2,2}$
3	2	2	$\hat{\theta}_{3,2,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Once a dataset has been uploaded via one of the three methods outlined, the user will have to define the variables required by **INTEREST** and some optional variables, depending on the structure of the input dataset. The names of each column (i.e., variable) from the uploaded dataset automatically populate a set of select-list inputs to assist the user.

The only variable required by **INTEREST** is a variable defining a point estimate from the simulation study; users can also pass standard errors of such estimates, and the true value of the estimand. If neither of these values is provided, only performance measures that can actually be calculated with the available information are returned. In order to provide additional flexibility, the user can define a column in the dataset that defines the true values of the estimand: this is especially useful for instance in settings where the true value can vary between repetitions. Further to that, a user can provide repetition-specific confidence bounds or even use t-distributed critical values rather than normal theory (by specifying a column that contains degrees of freedom per each repetition); once again, this can all be set via the *Data* tab, and will affect relevant performance measures. Finally, a user can define a variable representing methods being compared with the current simulation study (and choose the comparator), and one or more variables defining data-generating mechanisms (DGMs, e.g., sample size, true correlation, true baseline hazard function for survival models, etc.). We denote with *methods* the levels of the factor of primary comparative interest in a simulation study, and not necessarily an analytical method (strictly speaking). Other factors (e.g., characteristics of the data-generating mechanism) can be used as well, if representing the primary comparative interest of a study.

In its current form, **INTEREST** can only accept a single column as a *method* variable; when the primary focus of a simulation study is on several factors at once, we suggest pre-processing the dataset by creating a single column with all possible combinations from the factors of interest (e.g., using the `interaction` function in R).

The *View uploaded data* side tab in **INTEREST** displays the dataset uploaded by the user using the R package **DT**, an R interface to the **DataTables** plug-in for jQuery (Xie et al. 2020). The resulting table is interactive and can be sorted and filtered by the user. It is good practice to verify that the uploaded dataset is as expected before continuing with the analysis and any visual exploration.

3.2. Missing data

INTEREST includes a section for exploring missingness of estimates and/or standard errors from each repetition of a simulation study, which may occur, for example, due to non-convergence of some repetitions. Missing values need to be carefully explored and handled at the initial stage of any analysis. Missingness may originate as a consequence of software failures: if so, the code could (or should) be made more robust to ensure fewer or no failures. Conversely, missing data may arise as a consequence of characteristics of the simulated data, yielding to non-convergence of the estimation procedures. In other words, missing values may not be missing completely at random. A discussion on the interpretation of missing values can be found elsewhere (White et al. 2011; Morris et al. 2019).

The missing data functionality is based on the R package **naniar** (Tierney et al. 2020), and can be accessed via the *Missing data* tab. It comprises visual and tabular summaries; missing data visualisations available in **INTEREST** are the following:

- Bar plots of number (or proportion) of missing values by method and data-generating mechanism (if defined). Number and proportion of missing values are produced for each variable included in the data uploaded to **INTEREST**.
- A plot to visualise the amount of missing data in the whole dataset.
- A scatter plot with missing status depicted with different colours; to be able to plot missing values, they are replaced with values 10% lower than the minimum value in that variable. This plot allows identifying trends and patterns between variables in missing values (e.g., all estimates with a very large standard error have a missing point estimate).
- A heat plot with methods on the horizontal axis and the data-generating mechanisms on the vertical axis, with the colour fill representing the percentage of missingness in each tile.

Each plot can be further customised and exported (e.g., for use in slides and reports): see more details in the *Plots* section below. Finally, **INTEREST** computes and outputs a table with the number, proportion, and the cumulative number of missing values per variable, stratifying by method and data-generating mechanisms; the table can be easily exported to LaTeX format for further use (via the `kable` function from the R package **knitr** (Xie 2020)).

3.3. Performance measures

INTEREST estimates performance measures automatically as soon as the user defines the required variables via the *Data* tab. Supported performance measures are presented

Table 2: Overview of performance measures estimated by **INTEREST**.

Performance measure	Description
Bias	Deviation between estimate and the true value
Empirical standard error	Log-run standard deviation of the estimator
Relative precision against a reference	Precision of a Method B compared to a reference Method A
Mean squared error	The sum of squared bias and variance of the estimator
Model standard error	Average estimated standard error
Coverage	Probability that a confidence interval contains the true value
Bias-eliminated coverage	Coverage after removing bias, i.e., by computing the probability that a confidence interval contains the average point estimate across repetitions instead of the true value
Power	Power of a significance test

in Table 2, and discussed in more detail elsewhere (Burton et al. 2006; White 2010; Morris et al. 2019). In addition to that, **INTEREST** returns the mean and median estimates, and the mean and median squared error of the estimates. Finally, **INTEREST** computes and returns Monte Carlo standard errors by default. The list of performance measures estimated by **INTEREST** can be customised via the *Options* tab: by default, all are included.

3.4. Tables

Estimated performance measures are presented in tabular form in the *Performance measures* side tab, once again using the R package **DT**. The table of estimated performance measures is relative to a given data-generating mechanism, which can be modified using a select list input on the side. It is also possible to customise the number of significant digits and to select whether Monte Carlo standard errors should be excluded in each table or not, via the *Options* tab.

Finally, it is possible to export the tables in two ways:

1. Export the table in LaTeX format, e.g., for use in reports, articles, or presentations, via the *Export table* tab and the `kable` function from the R package **knitr** (Xie 2020). The caption of the table can be directly customised.

2. Export estimated performance measures as a dataset, e.g., to be used with a different software package of choice. The table of estimated performance measures can be exported as displayed by **INTEREST** or in tidy format, and in a variety of formats: comma-separated (`.csv`), tab-separated (`.tsv`), R (`.rds`), Stata (version 8-15, `.dta`), SPSS (`.sav`), and SAS (`.sas7bdat`).

3.5. Plots

INTEREST can produce a variety of plots to automatically visualise results from simulation studies. Plots produced by **INTEREST** can be categorised into two broad groups: plots of estimates (and their estimated standard errors) and plots of performance, following analysis. Plots for method-wise comparisons of estimated values and standard errors are:

- Scatter plots.
- Bland-Altman plots (Altman and Bland 1983; Bland and Altman 1999).
- Ridgeline plots (Wilke 2018).
- Contour and hexbin plots (as implemented in **ggplot2**'s `geom_density_2d` and `geom_hex` geometric objects).

Each plot will include all data-generating mechanisms by default and allows comparing serial trends and the relative performance of methods included in the simulation study; contour and hexbin plots are especially useful to deal with overplotting.

Conversely, the following plots are supported for estimated performance:

- Plots of performance measures with confidence intervals based on Monte Carlo standard errors. There are two variations of this plot: forest plots and lolly plots. Both methods display the estimated performance measure alongside confidence intervals based on Monte Carlo standard errors; different methods are arranged side by side, either on the horizontal or on the vertical axis.
- Heat plots of performance measures: these plots are mosaic plots where the several methods being compared (if defined) are on the horizontal axis and the data-generating mechanisms are on the vertical axis. Then, each tile of the mosaic plot is coloured according to the value of a given performance measure. To the best of our knowledge, this is a novel way of visualising results from simulation studies, with an application in practice that can be found elsewhere (Gasparini et al. 2019).
- Zip plots to visually explain coverage probabilities by plotting the confidence intervals directly. More information on zip plots is presented elsewhere (Morris et al. 2019).
- Nested loop plots, useful to compare performance measures from studies with several DGMs at once. This visualisation is described in more detail elsewhere (Rücker and Schwarzer 2014).

Finally, all plots can be exported for use in manuscript, reports, or presentations by simply clicking the *Save plot* button underneath a plot; all plots are exported by default in `.png` format, but other options are available via the *Options* tab. For instance, to suit a wide variety of possible use cases, **INTEREST** supports several alternative image formats such as `pdf`, `svg`, and `eps`. Through the *Options* tab it is also possible to customise the resolution of the plot for non-vectorial format (in dots per inch, `dpi`) and the physical size (height and width) of the plots to be exported. The *Options* tab allows further customisations: for instance, it is possible to (1) define a custom label for the x-axis and the y-axis and (2) change the overall appearance of the plot by applying one of the predefined themes (which are described in more detail in the *User guide* tab).

3.6. Interactive apps for exploring results

INTEREST allows researchers to upload a dataset with the results of their Monte Carlo simulation study obtaining estimates of performance in a quick and straightforward way. This is very appealing, especially with simulation studies with several data-generating mechanisms where it could be confusing to investigate all scenarios at once. Using the app, it is possible to vary data-generating mechanisms and obtain updated tables and plots in real-time, therefore allowing one to quickly iterate and take into consideration all possible scenarios.

3.7. Interactive apps for disseminating results

One of the intended usage scenarios for **INTEREST** consists of supplementing reporting of simulation studies. This is especially useful with large simulation studies, where it is most cumbersome to summarise all results in a manuscript: it is common to include in the main manuscript only a subset of results for conciseness. The remaining results are then relegated to supplementary material, web appendices, or not published at all - undermining dissemination and replicability of a study.

Furthermore, given that it is becoming increasingly common to publish the code of simulation study, one could publish the dataset with the results alongside the code used to obtain it. That dataset could then be uploaded to **INTEREST** by readers, who could then explore the full results of the study as they wish. Given the ubiquity of web services like GitHub (`https://github.com`) and data-sharing repositories such as Zenodo (`https://zenodo.org/`), we encourage **INTEREST** users to publish the full results of their simulation studies online for other users to download and experiment with.

4. Future developments

Although **INTEREST** is fully functional in its current state, several future developments are being planned. For instance, we aim to include support for multiple estimands at once as currently supported by **rsimsum** via the `multisimsum` function. We also aim to improve the flexibility of **INTEREST** in terms of customisation (of tables and plots), e.g., by displaying the raw R code used to generate the plots behind the scenes. Finally, we are considering adding additional interactive features to the app via HTML widgets, **D^3**, or other approaches; there are several R packages that allow incorporating interactive graphs into **Shiny** apps such as **htmlwidgets** (Vaidyanathan et al. 2019), **plotly** (Sievert 2018), and **r2d3** (Luraschi and Allaire 2018).

5. Case study

The case study included in this section illustrates the use of **INTEREST** to analyse publicly available results of a simulation study. In particular, we will be using the results from the worked illustrative example included in Morris et al. (2019).

The study dataset contains the results of a simulation study comparing three different methods for estimating the hazard ratio in a randomised trial with a time to event outcome. In particular, the methods being compared are proportional hazards survival models of the kind:

$$h_i(t) = h_0(t) \exp(X_i \theta),$$

where θ is the log hazard ratio for the effect of a binary exposure (e.g., treatment). This class of models requires an assumption regarding the shape of the baseline hazard function $h_0(t)$: it can be assumed to follow a given parametric distribution, or it can be left unspecified (yielding therefore a Cox model). The *aim* of this simulation study consists of assessing the impact of such an assumption on the estimation of the log hazard ratio.

Morris et al. (2019) consider two distinct *data-generating mechanisms*, varying the baseline hazard function:

1. An exponential baseline hazard with $\lambda = 0.1$ (DGM $= 1$).

2. A Weibull baseline hazard with $\lambda = 0.1, \gamma = 1.5$ (DGM $= 2$).

In both settings, data are simulated on 300 patients with a binary covariate (e.g., treatment) simulated using $X_i \sim \text{Bern}(0.5)$ - simple randomisation with an equal allocation ratio. The log hazard ratio is set to be $\theta = -0.50$; this is the true value of the *estimand* of interest.

Three distinct *methods* are fitted to each simulated scenario: a parametric survival model that assumes an exponential baseline hazard, a parametric survival model that assumes a Weibull baseline hazard, and a Cox semi-parametric survival model.

Finally, the *performance measures* of interest are bias, coverage, empirical and model-based standard errors. Assuming that $\text{Var}(\hat{\theta}) \leq 0.04$, 1600 repetitions are run to ensure that the Monte Carlo standard error of bias (the key performance measure of interest) is lower than 0.005.

The dataset with the results of this simulation study is publicly available in Stata format, and can be downloaded from a GitHub repository at the following URL:

`https://github.com/tpmorris/simtutorial/raw/master/Stata/estimates.dta`

Within the dataset published on GitHub, the exponential, Weibull, and Cox models are coded as Model 1, 2, and 3, respectively.

The workflow of **INTEREST** starts by providing the dataset with the results of the simulation study. Given that the dataset is already available online, we can directly pass the URL above to **INTEREST** and then define the required variables (as illustrated in Figure 3); the uploaded dataset can then be verified via the *View uploaded data* tab (Figure 4).

Table 3: Example of LaTeX table directly exported from **INTEREST**, case study DGM 2: true Weibull baseline hazard function.

Performance Measure	1	2	3
Bias in point estimate	0.0494 (0.0035)	0.0048 (0.0038)	0.0062 (0.0038)
Empirical standard error	0.1381 (0.0024)	0.1516 (0.0027)	0.1511 (0.0027)
Model-based standard error	0.1539 (0.0001)	0.1541 (0.0001)	0.1542 (0.0001)
Coverage of nominal 95% confidence interval	0.9600 (0.0049)	0.9556 (0.0051)	0.9575 (0.0050)

We can also customise the performance measures reported by **INTEREST** via the *Options* tab (Figure 5), e.g., focussing on those outlined above as key performance measures (bias, coverage probability, empirical standard errors, model-based standard errors).

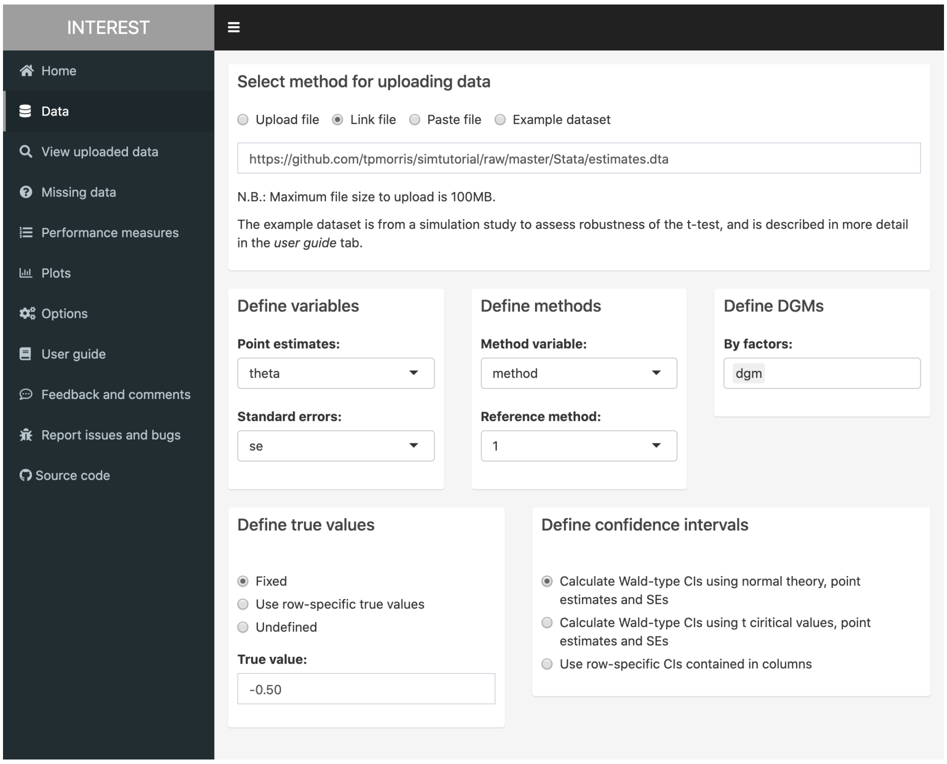
The next step of the workflow consists of investigating missing values: this can be achieved via the *Missing data* tab. In particular, there is no missing data in the study dataset (Figure 6). We can, therefore, continue the analysis knowing that there is no pattern of serial missingness or non-convergence issues in our data.

The performance measures of interest are tabulated in the *Performance measures* tab, e.g., for DGM = 2 (Figure 7). We can see that bias for the exponential model is much larger than the Weibull and Cox models: approximately 10% of the true value (in absolute terms) compared to less than 1%. Empirical and model-based standard errors are quite similar for the Weibull and Cox models; conversely, the exponential model seemed to overestimate the model-based standard error. Coverage was as advertised for all methods, at approximately 95%. By comparison, all models performed equally in the other scenario (DGM = 1); these results are omitted from the manuscript for brevity, but we encourage readers to replicate this analysis and verify our statement.

The *Performance measures* tab provides a LaTeX table ready to be pasted (e.g., in a manuscript): the resulting table is included as Table 3. A dataset with all the estimated performance measures tabulated here can also be exported to be used elsewhere (Figure 8).

We can also visualise the results of this simulation study. First, we can produce a method-wise comparison of point estimates from each method using e.g., scatter plots (Figure 9) or Bland-Altman plots (Figure 10). With both plots, it is possible to appreciate that for the DGM with $\gamma = 1.5$, the exponential model yields point estimates that are quite different compared to the Weibull and Cox models. Analogous plots can be obtained for estimated standard errors.

The performance measures tabulated in the *Performance measures* tab can also be plotted via the *Plots* tab. For instance, it is straightforward to obtain a forest plot for bias (as illustrated in Figure 11) which can be exported by clicking the *Save plot* button. The plots' appearance can also be customised via the *Options* tab, e.g., by modifying the axes' labels and the overall theme of the plot (Figure 12); the resulting forest plot, exported in `.pdf` format, is included as Figure 13. Several other data visualisations are supported by **INTEREST**, as described in the previous sections: lolly plots, zip plots, and so on.



Select method for uploading data

○ Upload file ● Link file ○ Paste file ○ Example dataset

https://github.com/tpmorris/simtutorial/raw/master/Stata/estimates.dta

N.B.: Maximum file size to upload is 100MB.

The example dataset is from a simulation study to assess robustness of the t-test, and is described in more detail in the *user guide* tab.

Define variables

Point estimates: theta

Standard errors: se

Define methods

Method variable: method

Reference method: 1

Define DGMs

By factors: dgm

Define true values

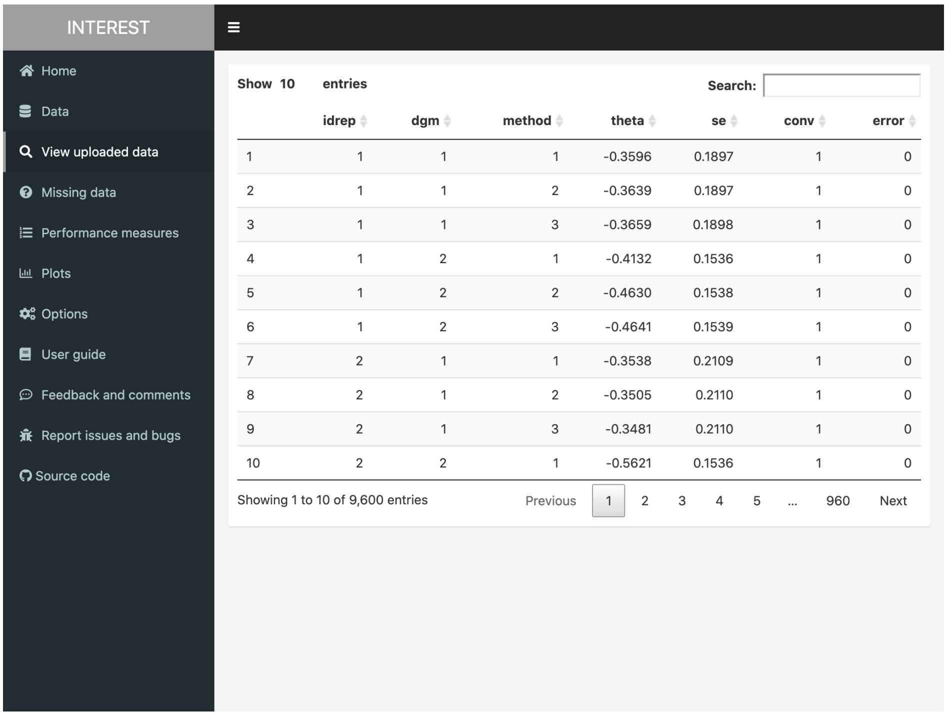
● Fixed
○ Use row-specific true values
○ Undefined

True value: -0.50

Define confidence intervals

● Calculate Wald-type CIs using normal theory, point estimates and SEs
○ Calculate Wald-type CIs using t ciritical values, point estimates and SEs
○ Use row-specific CIs contained in columns

Figure 3: App interface to load the dataset for the case study. **INTEREST** can import datasets that are available online by simply pasting a link to it; then, the required variables can be defined via a list of pre-populated select inputs.

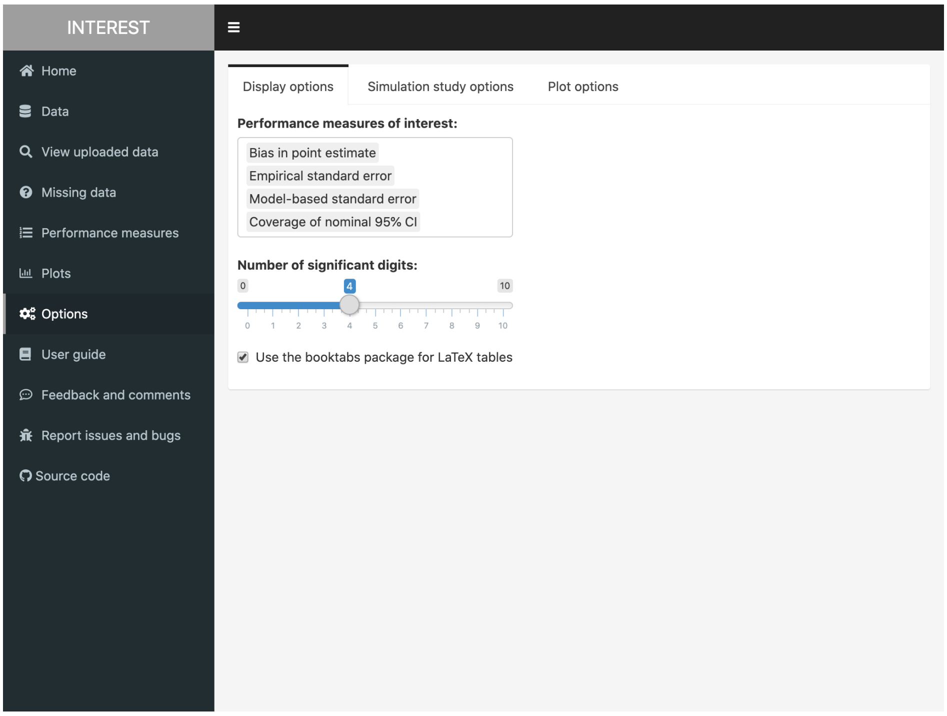


Show 10 entries

	idrep	dgm	method	theta	se	conv	error
1	1	1	1	-0.3596	0.1897	1	0
2	1	1	2	-0.3639	0.1897	1	0
3	1	1	3	-0.3659	0.1898	1	0
4	1	2	1	-0.4132	0.1536	1	0
5	1	2	2	-0.4630	0.1538	1	0
6	1	2	3	-0.4641	0.1539	1	0
7	2	1	1	-0.3538	0.2109	1	0
8	2	1	2	-0.3505	0.2110	1	0
9	2	1	3	-0.3481	0.2110	1	0
10	2	2	1	-0.5621	0.1536	1	0

Showing 1 to 10 of 9,600 entries

Figure 4: Verifying the dataset for the case study. After importing the study dataset, it is recommended to verify that the uploaded data is correct.



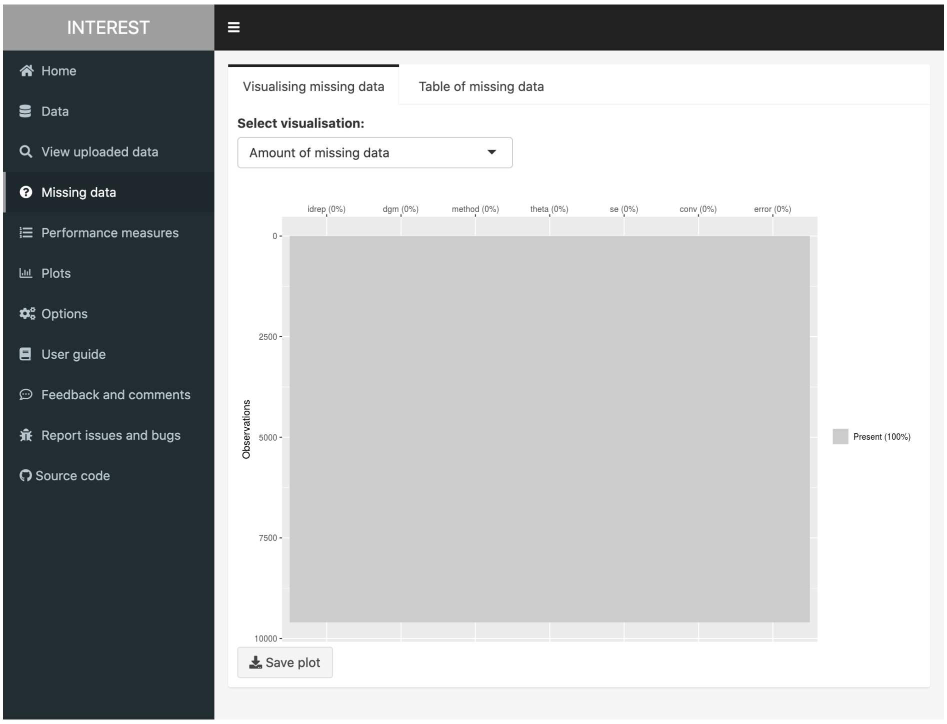
Performance measures of interest:

Bias in point estimate
Empirical standard error
Model-based standard error
Coverage of nominal 95% CI

Number of significant digits: 4

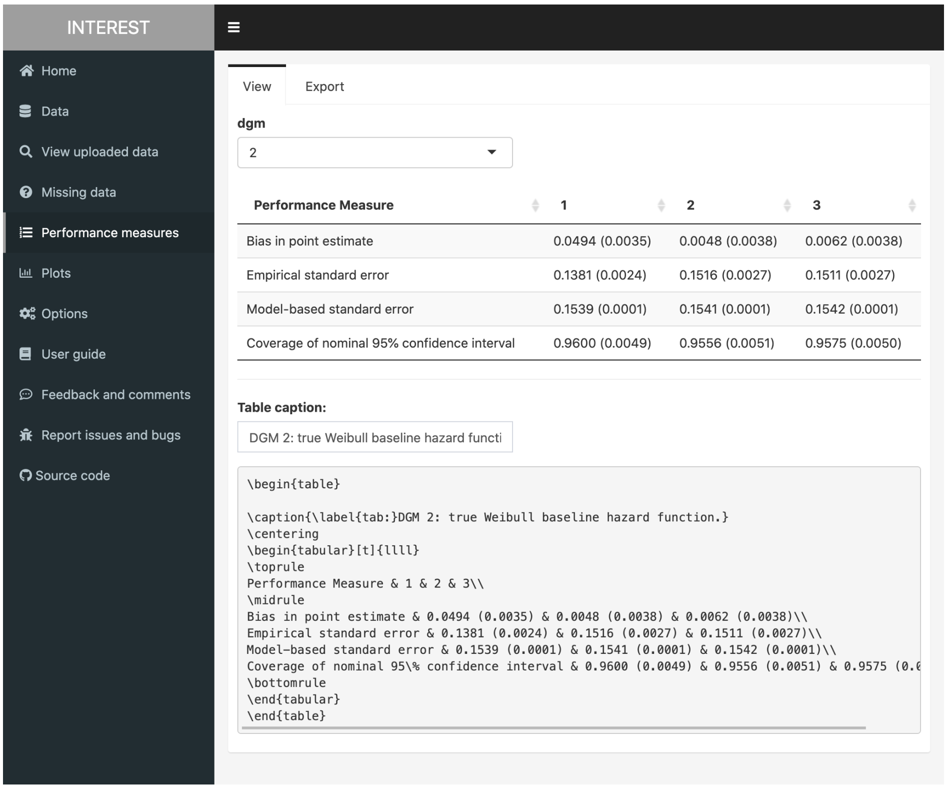
☑ Use the booktabs package for LaTeX tables

Figure 5: Customising the performance measures reported by **INTEREST**. It is possible to focus on a subset of key performance measures by selecting them via the *Options* tab.



Select visualisation: Amount of missing data

Figure 6: Investigating missing data. Missingness patterns in the study dataset need to be assessed before continuing with the analysis. Several visualisations and tabular displays are available from the *Missing data* tab.



dgm: 2

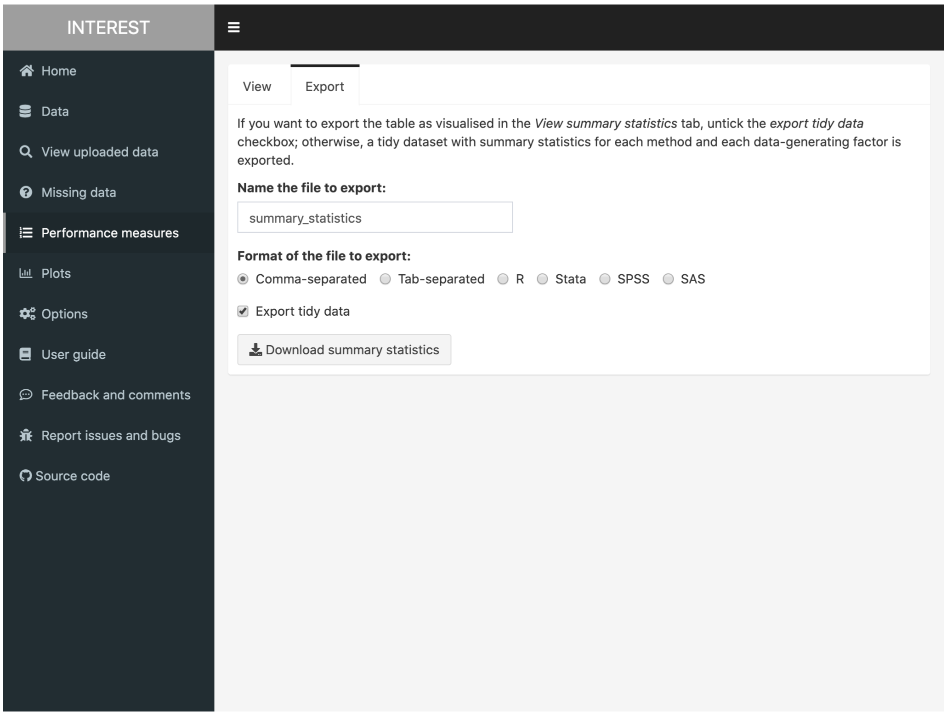
Performance Measure	1	2	3
Bias in point estimate	0.0494 (0.0035)	0.0048 (0.0038)	0.0062 (0.0038)
Empirical standard error	0.1381 (0.0024)	0.1516 (0.0027)	0.1511 (0.0027)
Model-based standard error	0.1539 (0.0001)	0.1541 (0.0001)	0.1542 (0.0001)
Coverage of nominal 95% confidence interval	0.9600 (0.0049)	0.9556 (0.0051)	0.9575 (0.0050)

Table caption: DGM 2: true Weibull baseline hazard functi

```
\begin{table}

\caption{\label{tab:}DGM 2: true Weibull baseline hazard function.}
\centering
\begin{tabular}[t]{llll}
\toprule
Performance Measure & 1 & 2 & 3\\
\midrule
Bias in point estimate & 0.0494 (0.0035) & 0.0048 (0.0038) & 0.0062 (0.0038)\\
Empirical standard error & 0.1381 (0.0024) & 0.1516 (0.0027) & 0.1511 (0.0027)\\
Model-based standard error & 0.1539 (0.0001) & 0.1541 (0.0001) & 0.1542 (0.0001)\\
Coverage of nominal 95\% confidence interval & 0.9600 (0.0049) & 0.9556 (0.0051) & 0.9575 (0.0
\bottomrule
\end{tabular}
\end{table}
```

Figure 7: Table of performance measures for a given DGM. Performance measures of interest are tabulated in the *Performance measures* tab, e.g., for the 2$^{\text{nd}}$ DGM (with a Weibull baseline hazard function).



If you want to export the table as visualised in the *View summary statistics* tab, untick the *export tidy data* checkbox; otherwise, a tidy dataset with summary statistics for each method and each data-generating factor is exported.

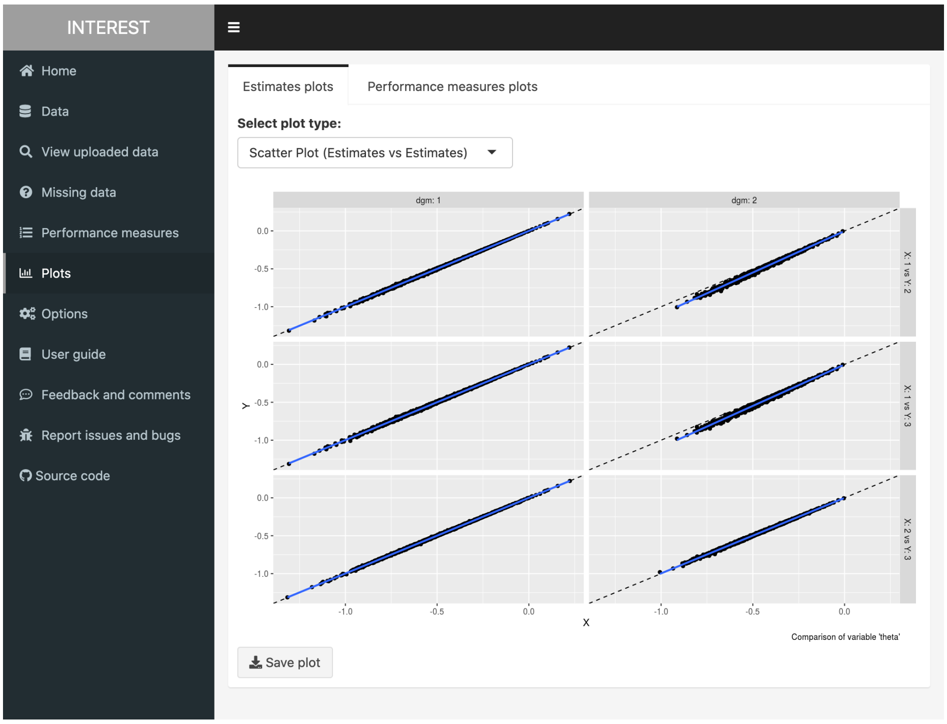
Name the file to export: summary_statistics

Format of the file to export:
● Comma-separated ○ Tab-separated ○ R ○ Stata ○ SPSS ○ SAS

☑ Export tidy data

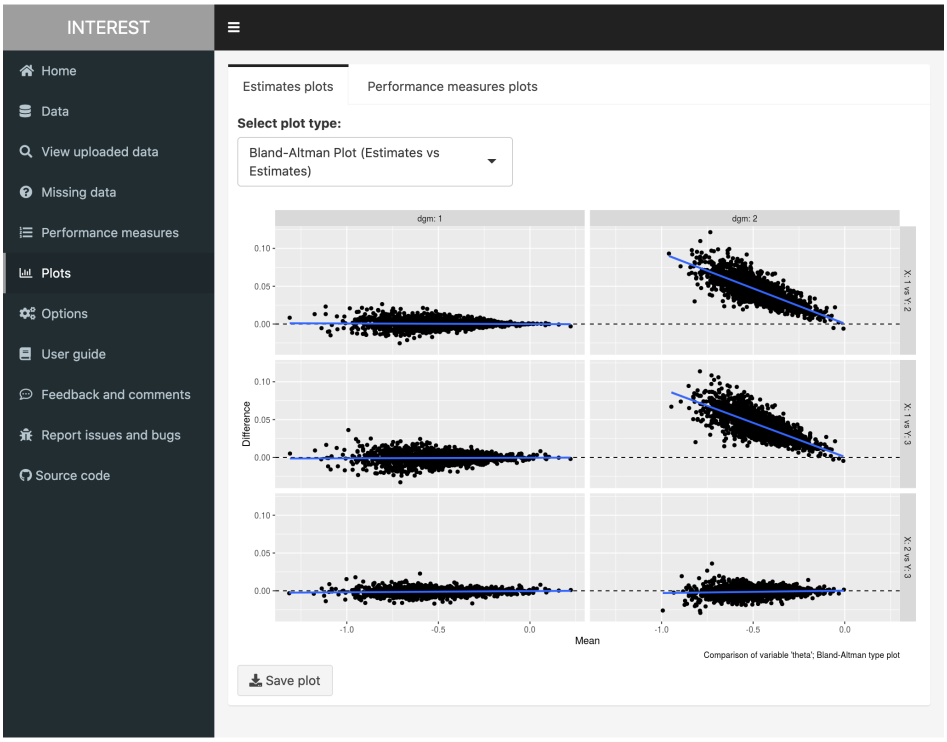
Download summary statistics

Figure 8: Exporting options for estimated performance measures. Performance measures of interest can be exported in a variety of formats ready to be used elsewhere (e.g., for dissemination purposes or to develop ad-hoc visualisations).



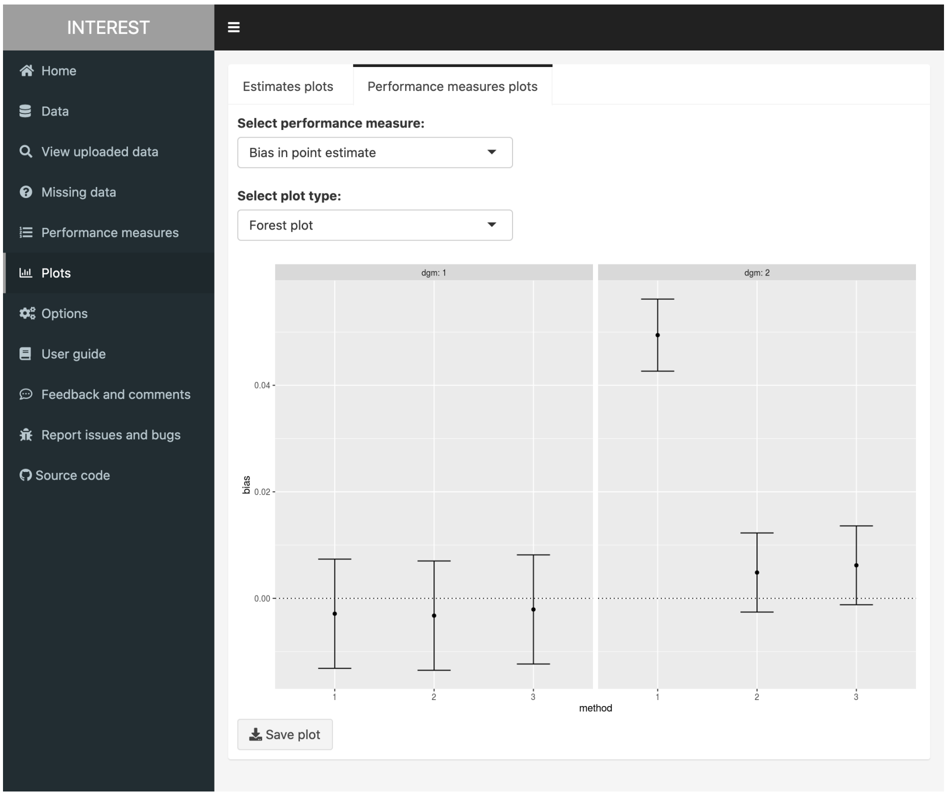
Select plot type: Scatter Plot (Estimates vs Estimates)

Figure 9: Visual comparison of point estimates via scatter plots. Points estimates for each method-DGM combination can be produced automatically using **INTEREST**.



Select plot type: Bland-Altman Plot (Estimates vs Estimates)

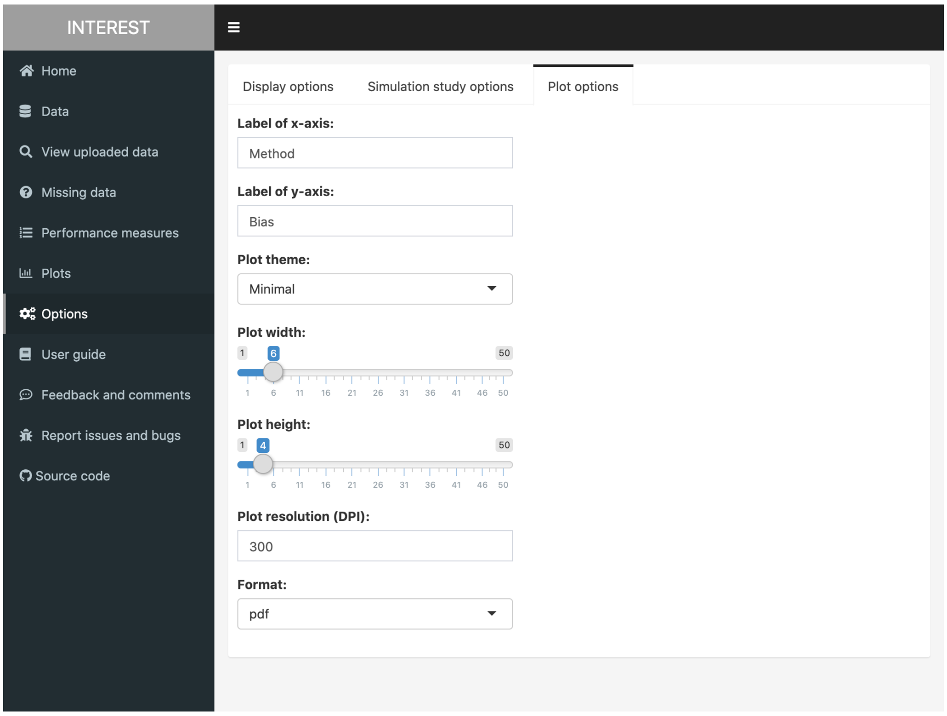
Figure 10: Visual comparison of point estimates via Bland-Altman plots. Points estimates for each method-DGM combination can be produced automatically using **INTEREST**.



Select performance measure: Bias in point estimate

Select plot type: Forest plot

Figure 11: Visual comparison of performance measures via forest plots. Estimated performance measures such as bias can be easily plotted via the *Plots* tab.



Label of x-axis: Method

Label of y-axis: Bias

Plot theme: Minimal

Plot width: 6

Plot height: 4

Plot resolution (DPI): 300

Format: pdf

Figure 12: Customising the visual appearance of plots. **INTEREST** allows customising the appearance of plots produced by the app via the *Options* tab, e.g., by modifying the axes' labels and/or the overall theme.

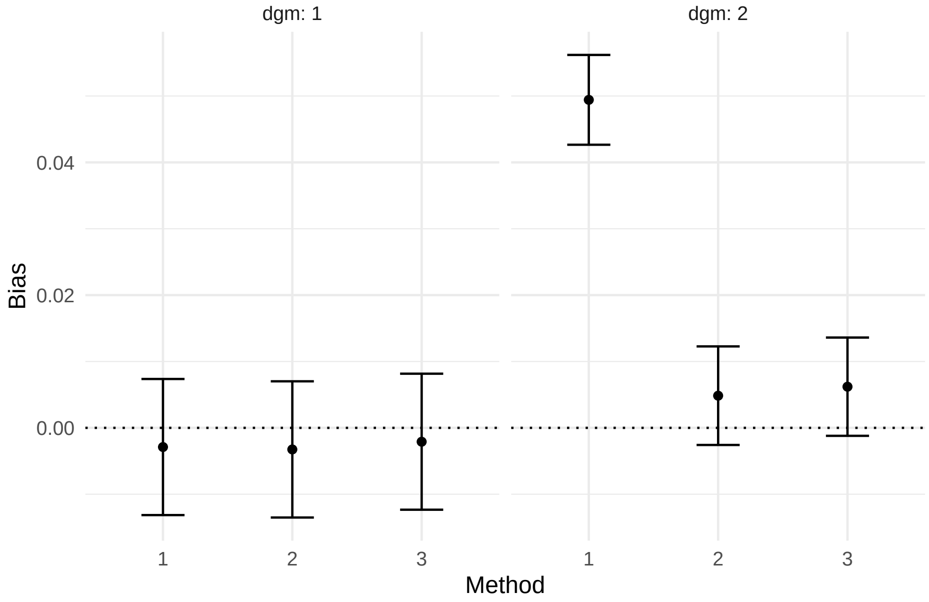


Figure 13: Forest plot for bias, case study on survival regression modelling. This forest plot produced by **INTEREST** and further customised via the *Options* tab can be directly exported from the app.

6. Conclusions

As outlined in the introduction, Monte Carlo simulation studies are too often poorly analysed and reported (Morris et al. 2019). Given the increased use in methodological statistical research, we hope that **INTEREST** could improve reporting and disseminating results from simulation studies to a large extent. As illustrated in the case study, the exploration and analysis of the Monte Carlo simulation study of Morris et al. (2019) can be fully reproduced by using **INTEREST**. Estimated performance measures are tabulated automatically, and plots can be used to visualise the performance measures of interest. Moreover, the user is not constrained to a given set of plots and can fully explore the results with ease e.g., by varying DGMs to focus on or by choosing different data visualisations. Most interestingly, the only requirement to reproduce the simulation study described in the case study is a device with a web browser and connection to the Internet. To the best of our knowledge, there is no similar application readily available to be used by researchers and readers of published Monte Carlo simulation studies alike.

Acknowledgements

TPM is supported by the Medical Research Council (grant numbers MC_UU_12023/21 and MC_UU_12023/29). MJC is partially funded by the MRC-NIHR Methodology Research Panel (MR/P015433/1).

We thank Ian R. White for discussions that lead to the inception and development of **INTEREST**.

References

Altman, D. G. and Bland, J. M. (1983). Measurement in medicine: The analysis of method comparison studies. *The Statistician*, 32(3):307, DOI: 10.2307/2987937, https://doi.org/10.2307%2F2987937.

Bandalos, D. L. and Gagné, P. (2012). Simulation methods in structural equation modeling. In *Handbook of structural equation modeling*, pages 92–108. The Guilford Press.

Bland, J. M. and Altman, D. G. (1999). Measuring agreement in method comparison studies. *Statistical Methods in Medical Research*, 8(2):135–160, DOI: 10.1177/096228029900800204, https://doi.org/10.1177%2F096228029900800204.

Boomsma, A. (2013). Reporting Monte Carlo studies in structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, 20(3):518–540, DOI: 10.1080/10705511.2013.797839, https://doi.org/10.1080%2F10705511.2013.797839.

Bostock, M., Ogievetsky, V., and Heer, J. (2011). **D^3**: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, DOI: 10.1109/tvcg.2011.185.

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

Chang, W. and Borges Ribeiro, B. (2018). ***shinydashboard**: Create Dashboards with **shiny***, https://CRAN.R-project.org/package=shinydashboard. R package version 0.7.1.

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2019). ***shiny**: Web Application Framework for R*, https://CRAN.R-project.org/package=shiny. R package version 1.4.0.

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

Gasparini, A. (2018). **rsimsum**: Summarise results from Monte Carlo simulation studies. *Journal of Open Source Software*, 3(26):739, DOI: 10.21105/joss.00739, https://doi.org/10.21105/joss.00739.

Gasparini, A., Clements, M. S., Abrams, K. R., and Crowther, M. J. (2019). Impact of model misspecification in shared frailty survival models. *Statistics in Medicine*, DOI: 10.1002/sim.8309, https://doi.org/10.1002/sim.8309.

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

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

Koehler, E., Brown, E., and Haneuse, S. J. (2009). On the assessment of Monte Carlo error in simulation-based statistical analyses. *The American Statistician*, 63(2):155–162, DOI: 10.1198/tast.2009.0030.

Laine, C., Goodman, S. N., Griswold, M. E., and Sox, H. C. (2007). Reproducible research: Moving toward research the public can really trust. *Annals of Internal Medicine*, 146(6):450–453, DOI: 10.7326/0003-4819-146-6-200703200-00154.

Luraschi, J. and Allaire, J. (2018). ***r2d3**: Interface to **D³** Visualizations*, https://CRAN.R-project.org/package=r2d3. R package version 0.2.3.

Morris, T. P., White, I., and Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, pages 1–29, DOI: 10.1002/sim.8086.

Peng, R. D. (2011). Reproducible research in computational science. 334(6060):1226–1227, DOI: 10.1126/science.1213847.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Rücker, G. and Schwarzer, G. (2014). Presenting simulation results in a nested loop plot. *BMC Medical Research Methodology*, 14(1), DOI: 10.1186/1471-2288-14-129.

Schulz, K. F., Altman, D. G., Moher, D., and for the CONSORT Group (2010). CONSORT 2010 Statement: Updated guidelines for reporting parallel group randomised trials. *PLOS Medicine*, 7(3):1–7, DOI: 10.1371/journal.pmed.1000251.

Sievert, C. (2018). ***plotly** for R*, https://plotly-r.com.

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

Spiegelhalter, D., Pearson, M., and Short, I. (2011). Visualizing uncertainty about the future. *Science*, 333(6048):1393–1400, DOI: 10.1126/science.1191181.

Tierney, N., Cook, D., McBain, M., and Fay, C. (2020). ***naniar**: Data Structures, Summaries, and Visualisations for Missing Data*, https://CRAN.R-project.org/package=naniar. R package version 0.5.0.

Vaidyanathan, R., Xie, Y., Allaire, J., Cheng, J., and Russell, K. (2019). ***htmlwidgets**:*

HTML Widgets for R, https://CRAN.R-project.org/package=htmlwidgets. R package version 1.5.1.

von Elm, E., Altman, D. G., Egger, M., Pocock, S. J., Gøtzsch, P., Vandenbroucke, J. P., and for the STROBE Initiative (2007). The Strengthening the Reporting of Observational Studies in Epidemiology (Strobe) Statement: Guidelines for reporting observational studies. *PLOS Medicine*, 4(10):1–5, DOI: 10.1371/journal.pmed.0040296.

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

White, I. R., Royston, P., and Wood, A. M. (2011). Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine*, 30(4):377–399, DOI: 10.1002/sim.4067.

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, 59(10), DOI: 10.18637/jss.v059.i10.

Wilke, C. O. (2018). ***ggridges**: Ridgeline Plots in **ggplot2***, https://CRAN.R-project.org/package=ggridges. R package version 0.5.1.

Xie, Y. (2020). ***knitr**: A General-Purpose Package for Dynamic Report Generation in R*, https://yihui.org/knitr/. R package version 1.28.

Xie, Y., Cheng, J., and Tan, X. (2020). ***DT**: A Wrapper of the JavaScript Library **DataTables***, https://CRAN.R-project.org/package=DT. R package version 0.12.

Affiliation:

Alessandro Gasparini
Biostatistics Research Group
Department of Health Sciences
University of Leicester
George Davies Centre
University Road
Leicester
LE1 7RH
United Kingdom
E-mail: alessandro.gasparini@ki.se

Tim P. Morris
MRC Clinical Trials Unit at UCL
90 High Holborn
London
WC1V 6LJ
United Kingdom

Michael J. Crowther
Biostatistics Research Group
Department of Health Sciences
University of Leicester
George Davies Centre
University Road
Leicester
LE1 7RH
United Kingdom

Journal of Data Science, Statistics, and Visualisation https://jdssv.org/
published by the International Association for Statistical Computing
http://iasc-isi.org/

ISSN 2773-0689
December 2021, Volume I, Issue IV *Submitted:* 2019-09-09
doi:10.52933/jdssv.v1i4.9 *Accepted:* 2020-06-26