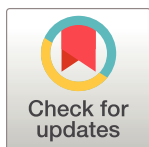RESEARCH ARTICLE

Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in estimating the mean squared error of the least squares estimator in linear regression

Marieke Stolte [1]*, **Nicholas Schreck**[2], **Alla Slynko**[3], **Maral Saadati**[2], **Axel Benner**[2], **Jörg Rahnenführer**[1], **Andrea Bommert**[1]

1 Department of Statistics, TU Dortmund University, Dortmund, North Rhine-Westphalia, Germany, **2** Division of Biostatistics, German Cancer Research Center, Heidelberg, Baden-Wuerttemberg, Germany, **3** Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario, Canada

* stolte@statistik.tu-dortmund.de

Abstract

Simulation is a crucial tool for the evaluation and comparison of statistical methods. How to design fair and neutral simulation studies is therefore of great interest for both researchers developing new methods and practitioners confronted with the choice of the most suitable method. The term simulation usually refers to parametric simulation, that is, computer experiments using artificial data made up of pseudo-random numbers. Plasmode simulation, that is, computer experiments using the combination of resampling feature data from a real-life dataset and generating the target variable with a known user-selected outcome-generating model, is an alternative that is often claimed to produce more realistic data. We compare parametric and Plasmode simulation for the example of estimating the mean squared error (MSE) of the least squares estimator (LSE) in linear regression. If the true underlying data-generating process (DGP) and the outcome-generating model (OGM) were known, parametric simulation would obviously be the best choice in terms of estimating the MSE well. However, in reality, both are usually unknown, so researchers have to make assumptions: in Plasmode simulation studies for the OGM, in parametric simulation for both DGP and OGM. Most likely, these assumptions do not exactly reflect the truth. Here, we aim to find out how assumptions deviating from the true DGP and the true OGM affect the performance of parametric and Plasmode simulations in the context of MSE estimation for the LSE and in which situations which simulation type is preferable. Our results suggest that the preferable simulation method depends on many factors, including the number of features, and on how and to what extent the assumptions of a parametric simulation differ from the true DGP. Also, the resampling strategy used for Plasmode influences the results. In particular, subsampling with a small sampling proportion can be recommended.

Introduction

Simulation studies are usually defined as computer experiments using artificial data generated by a pseudo-random number generator for which some truth about the data-generating process (DGP) and the outcome-generating model (OGM) is known, e.g., the true parameter values of the OGM or the distribution of the features. Well-designed, fair simulation studies are needed both for the evaluation of newly introduced methods and, in particular, for the neutral comparison of existing methods [1]. The DGP and OGM are usually chosen to either reflect realistic scenarios or edge cases for the application of the method of interest. We investigate the first case here.

We call the kind of simulation with artificial data, where the DGP and OGM are fully known, "parametric" simulations. Non-parametric simulation, where all data are real-life data, is not part of our analysis. Plasmode simulation is a special case of semi-parametric simulation, which is characterized by parts of the data being real-life data and parts of the DGP or OGM being specified. [2] give a technical introduction to (parametric) simulation studies and focus on guidance for best practices in performing and reporting simulation studies ("ADEMP" criteria). [3] give a general and more applied introduction to (parametric) simulation studies.

Parametric simulation studies are a crucial tool in the performance evaluation and comparison of statistical methods since they can offer insights beyond analytical results [2, 3] and can be used to evaluate criteria that cannot be assessed on real data where the DGP and OGM are unknown [3]. An example is the bias of an estimator, which can only be evaluated if the true parameter value can be controlled within the simulation. Therefore, one of the main advantages of parametric simulation studies is the full knowledge of the parameters of the DGP and OGM within the study. Another advantage is the possibility of investigating large numbers of different scenarios which permits analyzing how the performance of methods depends on the choice of the DGP and OGM. Moreover, it is possible to generate very large numbers of datasets. One of the main disadvantages is the simplification of real-life DGPs. Often very simple DGPs and OGMs are chosen arbitrarily which then do not reflect the often complex real-life processes. This may lead to wrong conclusions [3]. The over-simplification gets even worse for high-dimensional data, as it gets harder, for example, to specify realistic distributions and correlation structures for an increasing number of variables [4].

A different approach are so-called statistical Plasmodes as first introduced by [5]. [4] distinguish between statistical and biological Plasmodes depending on the procedure used for generating data. The motivation of statistical Plasmodes is to preserve a realistic data structure by resampling feature data from real-life datasets instead of using pseudo-random numbers as usually done in parametric simulation. At the same time, some control over the generated data is given by generating outcome variables for the given resampled feature data according to a known outcome-generating model like in parametric simulation. So, parametric and Plasmode simulations differ in the generation of features, while outcomes are generated in the same manner. For the feature resampling, different resampling approaches can be utilized [4]. Biological Plasmodes are generated by natural biological processes, for example in a wet lab by manipulating biological samples. In this paper, only statistical Plasmodes are considered.

The main advantage of Plasmode simulations is that the DGP does not have to be specified. The resampling is claimed to ensure the generation of realistic feature data. At the same time, quantities depending on the parameters of the OGM can still be assessed in contrast to fully non-parametric simulations. However, the resampling requires suitable datasets from the true DGP of interest with not too few observations. Depending on the application, this might be a major limitation. A more detailed discussion of the advantages and disadvantages of parametric vs. Plasmode simulation is given in [4]. The authors especially point out that evidence for

the often-made claim of Plasmode simulations producing more realistic data, i.e. data that is closer to the true DGP, is missing. Also, the authors emphasize the lack of studies on the effect of the often arbitrary choice of OGMs, which might affect both Plasmode and parametric simulation studies.

We aim to compare the ability of Plasmode and parametric simulation to assess the performance of statistical methods, especially concerning how misspecifications affect the results in both cases. We do this in a controlled simulation scenario so that we know both the true DGP and the true OGM, for evaluation purposes. In general, if we knew the truth, parametric simulation using this truth would be best. Since the truth is usually unknown in real-life applications, for parametric simulation researchers instead have to make assumptions about the DGP. These assumptions might deviate from the truth. Without deviations, the parametric simulation will always perform best since it accurately reflects the truth. On the other hand, when the parametric assumptions about the DGP are far from the truth, we expect Plasmode to be superior since the resampling is expected to give results that are rarely very far from the true DGP. Our goal is to determine the extent of deviation for which the parametric simulation gets worse than Plasmode. Therefore, we aim to find out

1. How much the DGP chosen in the parametric simulation can deviate from the truth before the parametric simulation becomes worse than Plasmode.

2. How deviations of the chosen OGM from the true OGM affect both parametric and Plasmode simulations.

3. How the choice of the resampling type affects the Plasmode simulation.

Based on the results, we are able to give guidance in which situations to choose parametric or Plasmode simulation and how to perform it.

We restrict our analysis to a simple scenario and focus on the estimation of the mean squared error (MSE) of the least squares estimator (LSE) in a linear regression model. Therefore, we focus on explanatory performance of the linear model and do not consider predictive performance. Moreover, we restrict to the low-dimensional setting, i.e., at most $p = 50$ features. We compare how well both parametric simulations with different assumptions about the DGP and the OGM and Plasmode simulations using different resampling strategies estimate the true MSE. So here, we check how well parametric and Plasmode simulation perform for one particular example of application. We investigate this for different true DGPs and OGMs. To compare different methods via simulation, this approach ensures that the simulation studies approximate well the performance of the methods for the true DGP and OGM.

The article is structured as follows. First, we describe parametric and Plasmode simulation in general and our specific simulation setup. Afterwards, we present the results of our simulations and provide recommendations for performing simulations based on our results. Finally, the results are summarized and discussed.

Methods

In the following, we briefly explain parametric and Plasmode simulation in general, pointing out, in particular, the different options for the resampling strategy in Plasmode simulations.

Parametric simulation

In parametric simulation studies, the whole data-generating process (DGP) and the outcome-generating model (OGM) have to be specified and are therefore known within the study. They are usually set up to either be as close as possible to a certain kind of data that the researcher is

interested in (e.g. gene expression data) or to cover as many situations as possible, possibly including extreme situations. We focus on the first case. Given the specified DGP, a large number of feature datasets is generated using pseudo-random number generators. In all cases that we are investigating, a target variable is then generated from these features by applying the OGM. This yields a large number of datasets which are then used for applying the methods of interest. This procedure allows the researcher to evaluate the performance of the methods with respect to a metric of interest. The process of generating the datasets can be seen as mimicking the repeated collection of samples from a large population. The results can provide insights into how the methods under study perform on average for datasets that are similar to the chosen DGPs and OGMs [3]. For more details on how to design, perform, analyze, and report parametric simulation studies, refer to [2].

Plasmode simulation

In Plasmode simulation studies, no assumptions on the DGP for the feature data are made. Instead, it is required to have a representative real-life dataset at hand that resulted from the true DGP [4]. If only real data was used, we would have no control over the DGP and OGM in our simulation. This means that we could not estimate certain quantities (e.g. the bias of an estimator) that directly depend on the true unknown parameters [3]. To enable us to estimate the quantities that directly depend on the true parameters of the OGM (which are most quantities of interest for performance evaluation of models), Plasmode simulation combines the use of real feature data with a known OGM. A Plasmode simulation study then works as follows. In each iteration, a Plasmode dataset is drawn from the real-life dataset at hand. The researcher has to decide on the resampling method. Possible methods include

- n out of n Bootstrap [6], i.e. drawing with replacement a dataset of the same size as the original dataset,

- m out of n Bootstrap [7–9], i.e. drawing with replacement $m < n$ observations of the original dataset,

- subsampling, i.e. drawing without replacement $m < n$ observations of the original dataset,

  or other adaptations of Bootstrap like

- smoothed Bootstrap [10–13], i.e. applying kernel-estimation to the empirical distribution of the original dataset and resampling from this smoothed empirical distribution,

- wild Bootstrap [14], i.e. adding the standardized values of each variable scaled by a random number to the original variable, or

- no resampling, i.e. using the whole dataset as it is.

A discussion of the first three options in the context of Plasmode simulation can be found in [4]. In the case of m out of n Bootstrap and subsampling, the researcher also has to decide on the number of observations to draw. For m out of n Bootstrap, there exists a data-dependent algorithm to find the optimal value of m [15]. After resampling a number of Plasmode datasets, the OGM is applied to each of the datasets to generate the outcomes. The resulting datasets can then be used for computing the performance metrics of interest like in parametric simulation. For more details, see [4].

Setup of the comparison study

In the following, we first describe the general approach and then the detailed setup of our comparison study.

General approach

We conduct our comparisons with respect to different true DGPs and OGMs. For each scenario, we calculate the true MSE of the least squares estimator (LSE). We then perform a parametric and a Plasmode simulation for estimating the MSE. For these simulations, we choose different DGPs and OGMs. The estimated MSEs resulting from these simulations are then compared to the true MSEs to assess how well the parametric and Plasmode simulation approximate the true values.

The outcomes are in all cases generated according to a linear model

$$y = X\beta + \varepsilon \tag{1}$$

for the true scenarios as well as for the parametric and Plasmode simulations. Note that the intercept is included in this model. The true MSE of the LSE $\hat{\beta}$ depends on the number of observations n, the residual variance of ε, and the distribution of the features X. For fixed X, the LSE is unbiased and thus the MSE reduces to the variance, which is given by $\text{Var}(\hat{\beta}|X) = \sigma^2(X^TX)^{-1}$.

To define the true DGP and OGM, we have to determine

- the true distribution of the features,
- the true parameter vector β, and
- the true distribution of the error term ε.

In the context of the parametric simulation, both DGP and OGM have to be chosen, so the distribution of the features, the coefficient vector, and the error distribution have to be specified. Inside the Plasmode simulation, only the OGM has to be chosen, so the coefficient vector and the error distribution have to be specified.

Simulation setup

In this section, we describe the true scenarios used in the simulation as well as the deviations from these true scenarios that are assumed for the parametric or Plasmode simulation.

True scenarios. We use different scenarios as our truth for the comparison. Table 1 gives an overview of these scenarios. The scenarios differ in the number of features (p) and observations (n) as well as the true correlation structure. In all scenarios, we assume that our features come from a multivariate normal distribution with mean zero and variances of one, that the true vector of coefficients (β) consists of all ones, and that the true error distribution is $N(0, 0.3^2)$.

We start with simple scenarios with only two features ($p = 2$), which are sampled from a bivariate Gaussian distribution with mean zero, variances of one, and a pairwise correlation of 0.2 or 0.5, and 50 or 100 observations. We use the same parameter settings for $p = 10$ except that we only look at pairwise correlations of 0.2. For $p = 50$, we always use 100 observations for identifiability reasons. Once again, we set all pairwise correlations to 0.2. Additionally, we use block diagonal correlation matrices with five blocks of ten features each. Within each block, the correlations are once set to $0.2^{|i-j|}$ and once to $0.5^{|i-j|}$ for all $i \neq j$. Features from different blocks are assigned a correlation of 0.

Table 1. Parameters for true data generating processes (DGP) and outcome generating models (OGM). In all scenarios, the true vector of coefficients is equal to $(1, \ldots, 1)^T \in \mathbb{R}^{p+1}$ and the error distribution is set to $\varepsilon \sim N(0, 0.3^2)$. $\mathbf{0}_p$ denotes the p-dimensional vector of zeros.

Name	p	n	Distribution of features
$(p2n100\rho0.2)$	2	100	$(X_1, X_2)^T \sim N_2(\mathbf{0}_2, \Sigma)$ with $\Sigma_{i,j} = 0.2 \; \forall i \neq j$, $\Sigma_{ii} = 1$
$(p2n50\rho0.2)$	2	50	$(X_1, X_2)^T \sim N_2(\mathbf{0}_2, \Sigma)$ with $\Sigma_{i,j} = 0.2 \; \forall i \neq j$, $\Sigma_{ii} = 1$
$(p2n100\rho0.5)$	2	100	$(X_1, X_2)^T \sim N_2(\mathbf{0}_2, \Sigma)$ with $\Sigma_{i,j} = 0.5 \; \forall i \neq j$, $\Sigma_{ii} = 1$
$(p10n100\rho0.2)$	10	100	$(X_1, \ldots, X_{10})^T \sim N_{10}(\mathbf{0}_{10}, \Sigma)$ with $\Sigma_{i,j} = 0.2 \; \forall i \neq j$, $\Sigma_{ii} = 1$
$(p10n50\rho0.2)$	10	50	$(X_1, \ldots, X_{10})^T \sim N_{10}(\mathbf{0}_{10}, \Sigma)$ with $\Sigma_{i,j} = 0.2 \; \forall i \neq j$, $\Sigma_{ii} = 1$
$(p50n100\rho0.2)$	50	100	$(X_1, \ldots, X_{50})^T \sim N_{50}(\mathbf{0}_{50}, \Sigma)$ with $\Sigma_{i,j} = 0.2 \; \forall i \neq j$, $\Sigma_{ii} = 1$
$(p50n100\rho0.2^{\lvert i-j\rvert})$	50	100	$(X_1, \ldots, X_{50})^T \sim N_{50}(\mathbf{0}_{50}, \Sigma)$ with covariance matrix Σ with blockdiagonal structure where within each of 5 blocks of 10 features the pairwise covariance/ correlation between the ith and jth feature of the block is given as $0.2^{\lvert i-j\rvert}$ and all variances are equal to 1
$(p50n100\rho0.5^{\lvert i-j\rvert})$	50	100	$(X_1, \ldots, X_{50})^T \sim N_{50}(\mathbf{0}_{50}, \Sigma)$ with covariance matrix Σ with block diagonal structure where within each of 5 blocks of 10 features the pairwise covariance/ correlation between the ith and jth feature of the block is given as $0.5^{\lvert i-j\rvert}$ and all variances are equal to 1
(quake)	3	100	$(X_1, \ldots, X_3)^T \sim N_3(\mathbf{0}_3, \Sigma)$ with covariance matrix Σ estimated from real dataset quake [16]
(wine_quality)	11	100	$(X_1, \ldots, X_{11})^T \sim N_{11}(\mathbf{0}_{11}, \Sigma)$ with covariance matrix Σ estimated from real dataset wine_quality [17]
(pol)	26	100	$(X_1, \ldots, X_{26})^T \sim N_{26}(\mathbf{0}_{26}, \Sigma)$ with covariance matrix Σ estimated from real dataset pol [18]
(Yolanda)	100	200	$(X_1, \ldots, X_{100})^T \sim N_{100}(\mathbf{0}_{100}, \Sigma)$ with covariance matrix Σ estimated from real dataset Yolanda [19]

https://doi.org/10.1371/journal.pone.0299989.t001

The scenarios are chosen to represent a low, a moderate, and a higher number of features for which the estimation process is still stable for 100 observations.

Moreover, we use covariance matrices estimated from real datasets to see how the simulations behave with more complicated correlation structures. We chose regression datasets that were available on OpenML [20], were used in the benchmark in [21], had at most 100 features, no constant features, no missing values and pairwise correlations with absolute values of at most 0.95. With these criteria, we ended up with four datasets: quake [16], wine_quality [17], pol [18], and Yolanda [19].

Deviations from true scenarios. We choose DGPs and OGMs for parametric and Plasmode simulation that present different kinds of deviations from the truth described in the previous section. The general structures of these deviations are listed in Table 2. A complete list of the specific parameter values that were chosen can be found in S1 Table. As a baseline, we assume the true scenario, which reflects the case that we—by chance—correctly specify all parameters in the simulations. Then we consider choices for each part of the DGP and OGM that reflect increasing deviations from the truth. For the coefficients, we use different values that are either wrong, but of the same order, or that even differ a large factor. We also included the case of assuming no effect ($\beta = 0$) which is an important special case that might be of interest in many studies. For the distribution of ε, we either only misspecify its standard deviation or misspecify the distribution as either more heavy-tailed (scaled t-distribution) or skewed (scaled and shifted χ^2-distribution). As deviations from the true feature distribution, we first still assume multivariate normal distribution but with wrong correlations, expectations, or variances. We then look at entirely wrong distributions, namely Gaussian mixture, log-normal, and Bernoulli distribution. The true correlation structure is preserved in those cases. We

Table 2. Deviations from true DGP and OGM for parametric and Plasmode simulation.

Scenario name	Description
True model	Assumptions coincide with truth
Coefficients misspecified I	Assumed β vector $(0, 1/p, 2/p, \ldots, 1)^T \in \mathbb{R}^{p+1}$ instead of $\mathbf{1}_{p+1}$
Coefficients misspecified II	Assumed β vector $\mathbf{0.05}_{p+1}$ instead of $\mathbf{1}_{p+1}$
Coefficients misspecified III	Assumed β vector $\mathbf{10}_{p+1}$ instead of $\mathbf{1}_{p+1}$
Coefficients misspecified IV	Assumed β vector $\mathbf{0}_{p+1}$ instead of $\mathbf{1}_{p+1}$
Error sd misspecified c	Assumed $\sigma = c$ instead of $\sigma = 0.3$ for $\varepsilon \sim N(0, \sigma^2)$
Correlation misspecified ρ	Assumed fixed pairwise correlation of ρ
Correlation misspecified $\rho^{\|i-j\|}$	Assumed pairwise correlation of $\rho^{\|i-j\|}$ for ith and jth feature for $p = 10$, or ith and jth feature within each of 5 blocks of 10 features for $p = 50$, respectively
Coefficients (I) and correlation (ρ) misspec.	$\mathbf{0.05}_{p+1}$ instead of $\mathbf{1}_{p+1}$ and fixed pairwise correlation of ρ instead of ρ_{true}
Coefficients (II) and correlation (ρ) misspec.	Assumed β vector $\mathbf{10}_{p+1}$ instead of $\mathbf{1}_{p+1}$ and fixed pairwise correlation of ρ instead of ρ_{true}
Error sd (0.4) and correlation (ρ) misspec.	Assumed $\sigma = 0.4$ instead of $\sigma = 0.3$ for $\varepsilon \sim N(0, \sigma^2)$ and fixed pairwise correlation of ρ instead of ρ_{true}
Feature distribution misspecified N(0,1), N(μ,1)	Assumed expectation of μ for second half of features
Feature distribution misspecified N(μ,1)	Assumed expectation of μ for all features
Feature distribution misspecified N(0,1), N(0,σ^2)	Assumed variance of σ^2 for second half of features
Feature distribution misspecified N(0, σ^2)	Assumed variance of σ^2 for all features
Feature distribution misspecified N(0,1), $(1 - \alpha)$N(0,1)+αN(0,10)	Assumed marginal distribution of second half of features as Gaussian mixture with 100α% outliers sampled from $N(0, 10)$ and marginal distribution of first half of features misspecified as normal with mean 0 and variance σ^2 that matches the variance σ^2 of the second half of features, $Cor(X_i, X_j) = \rho_{true}$, $i \neq j$ still holds
Feature distribution misspecified N(μ, σ^2), $(1 - \alpha)$N(0,1)+αN(3,1)	Assumed marginal distribution of second half of features as Gaussian mixture with 100α% of the observations sampled from $N(3, 1)$ and marginal distribution of first half of features misspecified as normal with mean μ and variance σ^2 chosen such that they match mean and variance of the second half of features, $Cor(X_i, X_j) = \rho_{true}$, $i \neq j$ still holds
Feature distribution misspecified N(1.65,2.83), logN(0,1)	Assumed marginal distribution of second half of features misspecified as log-normal with parameters 0 and 1 and marginal distribution of first half of features misspecified as normal with matching mean and variance, $Cor(X_i, X_j) = \rho_{true}$, $i \neq j$ still holds
Feature distribution misspecified Bin(π)	Assumed marginal distribution of second feature misspecified as Bernoulli with a success probability of π, $Cor(X_i, X_j) = \rho_{true}$, $i \neq j$ still holds
Error distribution misspecified t(df) scaled	Assumed $\varepsilon \sim t_{\text{df}}$ and scaled ε to still have sd 0.3
Error distribution misspecified chisq (df) scaled	Assumed $\varepsilon \sim \chi^2_{\text{df}}$ and shifted and scaled ε to still have mean 0 and sd 0.3

achieve this by generating Gaussian mixture, log-normal, and Bernoulli variables from multivariate normals and setting the covariance matrix of the underlying normals in a way such that the corresponding variables have the desired variances and covariances. For log-normals and Bernoulli variables with variables of the same distribution, the calculation can be found in [22, 23]. The calculation for log-normal and Bernoulli variables in combination with normal variables as well as all calculations for Gaussian mixture variables can be found in S1 Appendix.

Simulation procedure

The overall simulation structure is described in Algorithm 1. For each true scenario, we first approximate the true MSE by drawing 25 000 000 datasets of size n from the true distribution of X with the first column being a vector of ones, corresponding to the intercept of the model. We then calculate $X\beta$ and add random noise ε according to the true distribution of ε and define this as our outcome vector y belonging to the respective dataset. For each pair of data X and corresponding target y, we estimate $\hat{\beta}$ using least squares estimation. We then calculate the component-wise means over the replications of the simulation of $(\hat{\beta}_j - \beta_j)^2, j = 0, \ldots, p$, with p denoting the number of features, as estimates of the true component-wise MSEs. We refer to these quantities as the "true" component-wise MSEs. In each true scenario, we then perform parametric and Plasmode simulations for estimating the component-wise MSEs under the assumption that we do not know the respective true scenario.

Algorithm 1 Structure of simulation process

Require: $n > 0$ (number of observations), $0 < p < n$ (number of features), $n.mse > 0$ (number of MSE estimations), $n.mod > 0$ (number of LSEs, i.e. model estimates, used for estimation of one estimated MSE), true DGP (distribution of features), true OGM (β, distribution of ε), assumed DGP (assumed distribution of features), assumed OGM (β_a, assumed distribution of ε), *type* of Bootstrap, proportion π for resampling (= 1 for n out of n Bootstrap, Wild Bootstrap, and Smoothed Bootstrap)

Ensure: Error in estimated MSE for parametric simulation

1: $\text{MSE}_{true;j} \leftarrow \mathbb{E}[(\hat{\beta}_j - \beta_j)^2], j = 0, \ldots, p$, for the LSE $\hat{\beta}$ in the true model
2: **for** $k = 1, \ldots, n.mse$ **do**
3:   $X^{(k,i)} \leftarrow$ design matrix generated with Algorithm 2 or 3 for $i = 1, \ldots, n.mod$
4:   **for** $i = 1, \ldots, n.mod$ **do**
5:     $\varepsilon^{(k,i)} \leftarrow$ noise sampled from assumed distribution of ε
6:     $y^{(k,i)} \leftarrow X^{(k,i)} \beta_a + \varepsilon^{(k,i)}$
7:     $\hat{\beta}^{(k,i)} \leftarrow ((X^{(k,i)})^T X^{(k,i)})^{-1} (X^{(k,i)})^T y^{(k,i)}$ ▷ LSE
8:   **end for**
9:   $\text{MSE}_j^{(k)} \leftarrow \frac{1}{n.mod} \sum_{i=1}^{n.mod} (\hat{\beta}_j^{(k,i)} - \beta_{a;j})^2, j = 0, \ldots, p$
10:   $Err_j^{(k)} \leftarrow \text{MSE}_j^{(k)} - \text{MSE}_{true;j}, j = 0, \ldots, p$
11: **end for**

The process for data generation for parametric simulation is described in Algorithm 2. We make different assumptions on the distribution of the features (X), the values of the coefficients β, and the distribution of ε. We then generate $n.mod = 1000$ datasets according to these assumptions using pseudo-random numbers.

Algorithm 2 Structure of feature data generation for parametric simulation

Require: $n > 0$, $0 < p < n$, assumed DGP, k (iteration number of Algorithm 1)

Ensure: Generated datasets

1: **for** $i = 1, \ldots, n.mod$ **do**   ▷ Inner Simulation
2:   $X^{(k,i)} \leftarrow$ design matrix drawn from assumed data generating process using a pseudo-random number generator
3: **end for**

In some scenarios, we use parametric simulation with estimation of mean and covariance. For this, at the beginning of each simulation, one dataset of size $n = 1000$ is sampled from the true DGP and the mean and covariance are estimated from this dataset and used as the assumed mean and covariance of the assumed DGP. This corresponds to the case that researchers might have data at hand from which they estimate some characteristics of the DGP

to incorporate them into a parametric simulation in order to perform a more realistic simulation.

The procedure for data generation for Plasmode simulation is described in Algorithm 3. Here, we have to specify the resampling method to use. As a first step for each Plasmode simulation, one dataset is drawn from the true DGP. Note that in each case, the number of observations *after* resampling has to match the number of observations used for parametric simulation and for the true scenario to ensure a fair comparison of methods. For a more detailed discussion of this issue, see below. We then draw $n.mod = 1000$ resampled datasets from our dataset according to the chosen resampling method.

Algorithm 3 Structure of feature data generation for Plasmode simulation

Require: $n > 0$, $0 < p < n$, true DGP, *type* of Bootstrap, proportion π for resampling (= 1 for n out of n Bootstrap, Wild Bootstrap and Smoothed Bootstrap), k (iteration number of Algorithm 1)
Ensure: Plasmode datasets
1: $X_{Plasm}^{(k)} \leftarrow$ design matrix $\in \mathbb{R}^{\lceil n/\pi \rceil \times (p+1)}$ drawn from true DGP
2: **for** $i = 1, \ldots, n.mod$ **do** $\triangleright$ Inner Simulation
3:   **if** *type* == "m out of n Bootstrap" **or** *type* == "n out of n Bootstrap" **then**
4:      $X^{(k,i)} \leftarrow n$ rows sampled from $X_{Plasm}^{(k)}$ with replacement
5:   **else**
6:     **if** *type* == "Subsampling" **then**
7:        $X^{(k,i)} \leftarrow n$ rows sampled from $X_{Plasm}^{(k)}$ without replacement
8:     **else**
9:       **if** *type* == "Wild Bootstrap" **then**
10:          $a \leftarrow$ vector of p numbers sampled from $N(0, 1)$
11:          $X_1^{(k,i)} \leftarrow \mathbf{1}_n$
12:          $X_j^{(k,i)} \leftarrow X_j^{(k,i)} + a_j \cdot (X_j^{(k,i)} - \bar{X}_j^{(k,i)})/SD(X_j^{(k,i)})$, $j = 2, \ldots, p + 1$
13:        **else**
14:          **if** *type* == "Smoothed Bootstrap" **then**
15:             $X^{(k,i)} \leftarrow n$ rows sampled from $X_{Plasm}^{(k)}$ with replacement + random noise from a multivariate normal distribution centered at the data points and parameterized by corresponding bandwidth matrix estimated by Silverman's rule [24]);
16:          **end if**
17:        **end if**
18:     **end if**
19:   **end if**
20: **end for**

We utilize the following Bootstrap versions:

- *m* out of *n* Bootstrap [7–9] with resampling proportion $\pi \in \{0.01, 0.1, 0.5, 0.632, 0.8, 0.9\}$, i.e. drawing with replacement n observations out of $\lceil n/\pi \rceil$ observations,

- *n* out of *n* Bootstrap [6], i.e. drawing with replacement n observations out of n (special case of *m* out of *n* Bootstrap for $\pi = 1$),

- Smoothed Bootstrap [10–13], i.e. drawing with replacement n observations out of the smoothed empirical distribution of n observations,

- Wild Bootstrap [14], i.e. adding the standardized version of each observed feature vector scaled with a noise factor sampled from $N(0, 1)$ to the observed feature vectors, and

- subsampling with resampling proportion $\pi \in \{0.01, 0.1, 0.5, 0.632, 0.8, 0.9\}$, i.e. drawing without replacement n observations out of $\lceil n/\pi \rceil$ observations,

- no resampling, equivalent to subsampling with resampling proportion $\pi = 1$.

We do not determine an optimal resampling proportion, e.g. with the algorithm introduced in [15], since it takes too much time to repeat this for every dataset in the simulation. Instead, we try a range of resampling proportions.

On each dataset generated either according to the parametric or the Plasmode approach, the linear model (1) using the chosen parameters for the OGM is then applied to generate the outcome variable. From these, $\hat{\beta}$ is estimated for each dataset. The MSE is estimated as the average component-wise squared difference of the estimated and assumed coefficient vectors. One estimated MSE value corresponds to the result of one parametric or Plasmode simulation study. The whole process is repeated 100 times so we can see how much variation exists in the MSE estimation when repeating the parametric or Plasmode simulation study.

Performance evaluation

To compare the performance of parametric and Plasmode simulation, we look at their errors in MSE estimation. For each type of simulation (parametric, parametric with estimation of mean and variance, Plasmode with different resampling methods and proportions) we obtain 100 estimated MSEs for each deviation from the true DGP and OGM. We calculate the component-wise absolute errors as the differences between estimated MSEs and corresponding true MSEs in each case. Additionally, we calculate the relative errors by dividing the absolute errors by the corresponding true MSEs. We aggregate the absolute and relative errors per simulation over the coefficients by taking the arithmetic mean over the absolute component-wise values. We aggregate over the repetitions of the simulation studies by taking the median of the aggregated values. With this strategy, runs with large errors in single coefficients obtain large aggregated values, while the overall aggregated value across simulation repetitions is robust against single simulations with large aggregated errors.

We examine the errors graphically using boxplots. An example with a corresponding explanation will be shown later. Additionally, we analyze how much the assumptions in parametric simulation can deviate from the truth until the results are worse than with Plasmode simulation. Therefore, we sort the parametric deviations within each subgroup (e.g. deviation from the variance of the multivariate normal) in increasing order of the magnitude of the deviation (e.g. if the true variance is 1 and the tested values are 0.1, . . ., 0.99, these are ordered decreasingly) and identify the first value in this order for which the fully aggregated error for parametric simulation is larger than that for the considered type of Plasmode simulation. In this way, we can quantitatively compare parametric simulation to the different Plasmode variants. If the first value where parametric is worse than Plasmode is close to the true value, it follows that for deviations of this type, the parametric simulation is very sensitive to small deviations and we have to be very confident in our parameter settings for the DGP if we want to use parametric simulation. These are the cases where Plasmode might be superior to parametric simulation.

Software

All analyses are performed using R 4.2.2 [25]. We use the mvtnorm package [26, 27] to simulate data from multivariate normal distributions. For smoothed Bootstrap, the R package kernelboot [28] is used. For visualization of the results, we use the ggplot2 package [29] and ggh4x [30]. The R code and results for the simulation are available on Zenodo (https://doi.org/10.5281/zenodo.10567144, https://doi.org/10.5281/zenodo.10567059).
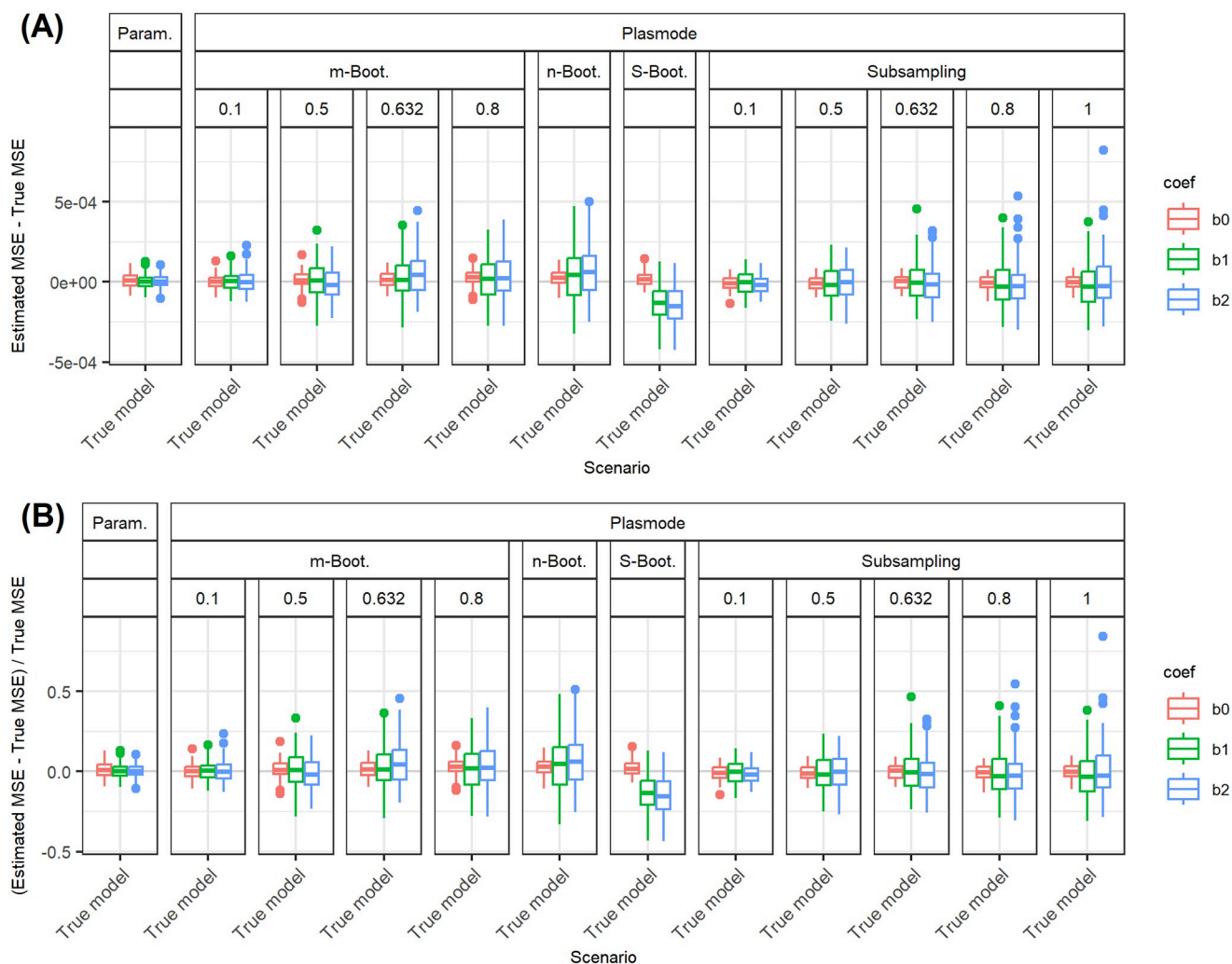
Fig 1. Relative error in MSE estimation for individual coefficients for different types of Plasmode simulation compared to parametric simulation under assumption of true DGP and OGM.

https://doi.org/10.1371/journal.pone.0299989.g001

Results

In this Section, we evaluate the results of the simulations. First, we explain the plots for one simple scenario and type of deviation. Then, the different resampling strategies for Plasmode are compared. Afterward, we discuss the results for the different types of deviations. Last, we consider the results for correlation structures estimated from real data as well as the effect of the size of the resampled dataset.

Example

In the following, we explain the displays that we use in the subsequent sections using one concrete example. We again consider the two scenarios with $p = 2$, $n = 100$, pairwise correlation of 0.2, $\beta = \mathbf{1}_3$, and $\varepsilon \sim N(0, 0.3^2)$ or $\varepsilon \sim N(0, 3^2)$. We calculated the errors in the MSE estimation using parametric and Plasmode simulation as described in the previous section. We display the errors in different ways using boxplots. We display the absolute or relative errors for each coefficient individually like in Figs 1 and 3, and S1 Fig, or aggregated over the coefficients like
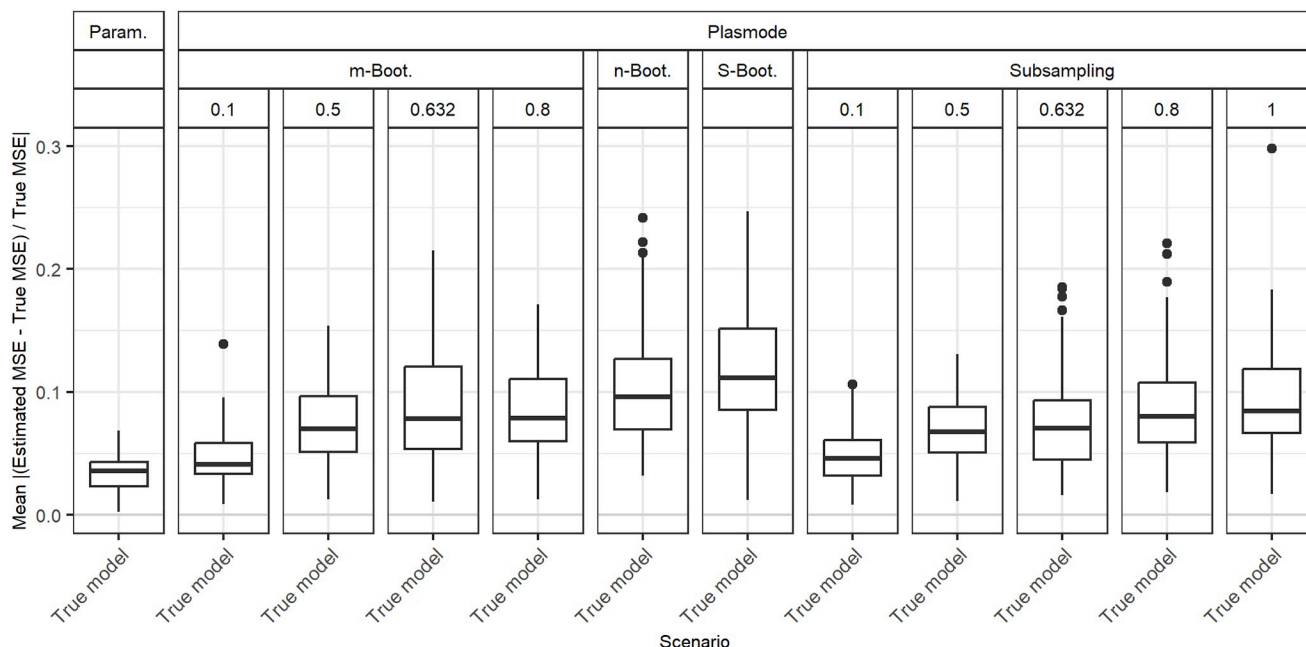
Fig 2. Absolute value of the relative error in MSE estimation averaged over individual coefficients, for different types of Plasmode simulation compared to parametric simulation under the assumption of the true DGP and OGM, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \, \forall i \neq j$.

https://doi.org/10.1371/journal.pone.0299989.g002

in Fig 2. We use the unaggregated version in cases where the error for different coefficients might behave differently. This is for example the case for deviations in the feature distribution of the second half of features. Otherwise, if all coefficients behave similarly, we use the aggregated version.

For the individual coefficients, the absolute or relative errors of the 100 repetitions of each type of simulation (parametric, different types of Plasmode) are displayed in one box per coefficient. This is done separately for the true model and each deviation. The deviations are described on the *x*-axis and coefficients are distinguished by differently colored boxes. The headers give information about the type of simulation used. The first row is the distinction between parametric and Plasmode simulation. The second row gives the type of Plasmode simulation. The third row gives the resampling proportion. For example in Fig 1 in the third facet, the relative errors per coefficient for Plasmode using *m* out of *n* Bootstrap with a resampling proportion of 0.5 are displayed. This corresponds to sampling with replacement 100 observations from a dataset of 200 observations for each simulation. For parametric simulation, *n* out of *n* Bootstrap and Smoothed Bootstrap, there is no subsampling proportion so this field is left empty. We leave out Wild Bootstrap in the following analyses since it produces very large outliers and is consistently outperformed by all other Bootstrap types (see e.g. Fig 4). We abbreviate *m* out of *n* Bootstrap as *m*-Bootstrap, *n* out of *n* Bootstrap as *n*-Bootstrap and Smoothed Bootstrap as *S*-Bootstrap. If necessary we further abbreviate Bootstrap as Boot. or B., Parametric as Param. or Prm. and Subsampling as Sub.

For the aggregated errors, we display the mean over the absolute values of the errors of the individual coefficients per deviation, i.e. the mean error per coefficient of one simulation, for the 100 repetitions of each simulation type in one box. Apart from the aggregation, the figures are constructed in the same way as for the individual coefficients.
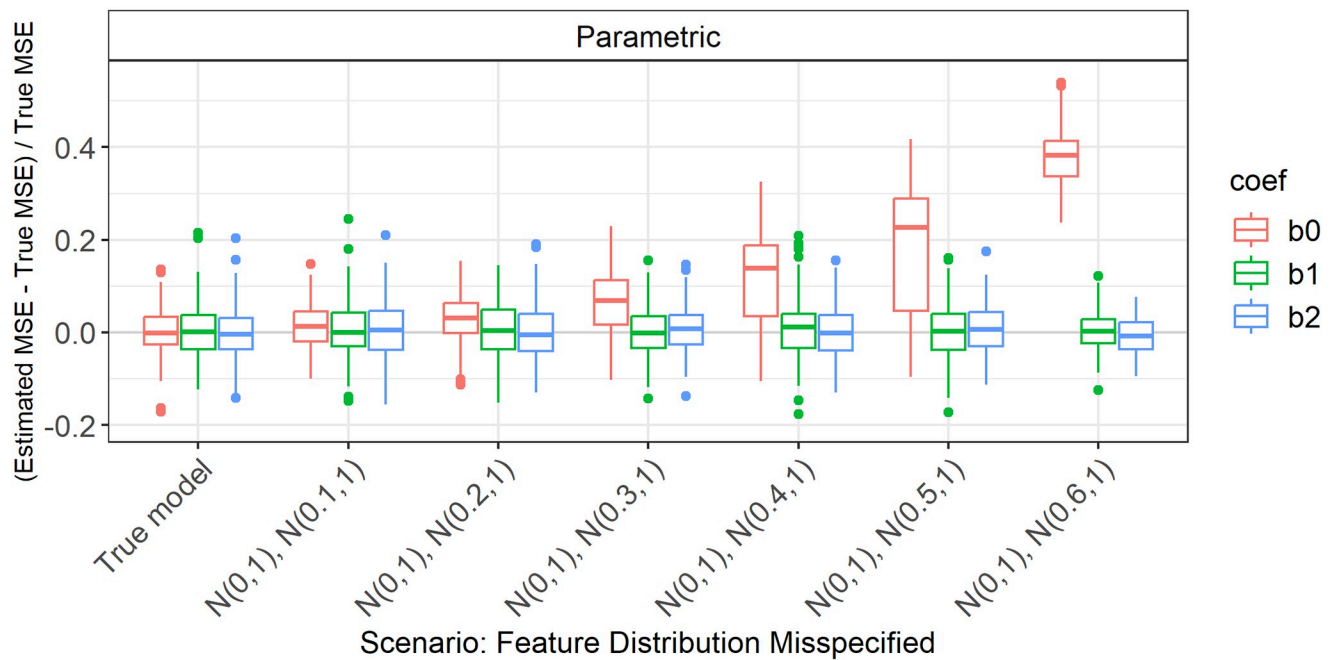
Fig 3. Absolute value of relative error in MSE estimation for individual coefficients when the assumed feature distribution in parametric simulation deviates from the true distribution, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$.

There are two types of comparisons: we can compare the performance of different types of simulation for the true model like in Figs 1 and 2, and S1 Fig to see how well each simulation type would perform if we knew the truth. Or we can compare the performance for differently strong deviations like in Fig 3. This allows us to assess the impact of different deviations on the performance. We can also combine both displays and show the performance for one kind of deviation for all those types of simulation that are affected by it and the performance for all other simulation types for the true model only. For example, in the case of deviation from the mean of the second feature distribution, we can show the errors for parametric simulation for different amounts of deviation as in Fig 3 along with the performance of the Plasmode types under the true model as in Fig 1 or S1 Fig. We cannot misspecify the feature distribution in Plasmode simulation, so only the true model is shown. This combined version is the display that we will use for the rest of our analysis.

In general, we might be interested in both absolute and relative errors. As can be seen in S1 Fig, the absolute errors for our specific problem are directly dependent on the chosen parameter for the error standard deviation: if the standard deviation changes by a factor of 10, e.g. here from $\sigma = 0.3$ to $\sigma = 3$, the errors change by a factor of approximately $10^2 = 100$, which can easily be checked by the theoretical relation $\text{Var}(\hat{\beta}|X) = \sigma^2 X^T X$, using that the LSE is unbiased for fixed X. The relative errors, on the other hand, are independent of σ since the factor affects both the absolute error and the true MSE, by which the absolute error is divided, in the same way. This is for example demonstrated in Fig 1. Therefore, we will only display the relative version for the rest of this analysis since the absolute values could be scaled to be arbitrarily small or large by choosing the error variance accordingly. We will also restrict our analysis to the case $\sigma = 0.3$, since this leads to more stable simulations than $\sigma = 3$, as the latter corresponds to an extremely low signal-to-noise ratio.
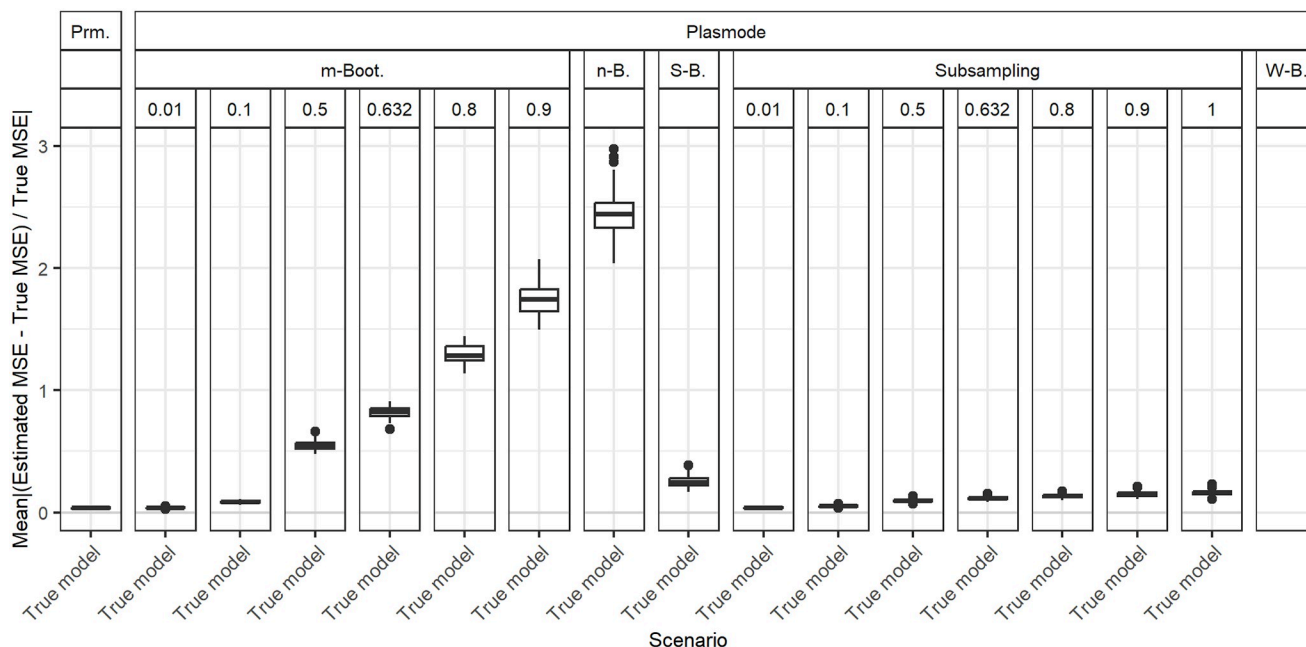
Fig 4. Absolute value of relative error in the MSE estimation averaged over individual coefficients for different types of Plasmode simulation compared to parametric simulation, under the assumption of the true data generating process and outcome generating model, for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$.

Comparison of different Plasmode types and resampling proportions

Fig 4 shows the aggregated relative errors for $p = 50$ with fixed pairwise correlations of 0.2 for the true model for all types of simulation. This example confirms that overall, Plasmode using Wild Bootstrap performs worst. All values for its relative errors lie outside the range of all other resampling types. This is similar for other scenarios, such that we do not show the results for Wild Bootstrap in any other plot. Within the other simulation types, Plasmode using the n out of n Bootstrap performs worst with relative mean errors of around 2.5 and also relatively high variation. m out of n Bootstrap and subsampling perform better both in terms of the median aggregated error and in terms of smaller variability with decreasing resampling proportion, i.e. the larger the dataset from which the 100 observations are sampled, the lower the variability. m out of n Bootstrap converges towards n out of n Bootstrap for increasing subsampling proportions. Except for very low subsampling proportions (0.1 and 0.01), Bootstrap performs worse than subsampling both with regard to median aggregated error and variability. It is interesting to note, that no resampling (i.e. subsampling with a subsampling rate of one), which means using the same feature data for the whole simulation and only sampling new observations of the target, still outperforms m out of n Bootstrap with subsampling proportions from 0.5 on as well as the smoothed, n out of n, and wild bootstrap. Smoothed Bootstrap performs worse than all subsampling versions, but better than the m out of n Bootstrap for subsampling proportions from 0.5 on. With Smoothed Bootstrap, the true MSE of the slope coefficients gets consistently underestimated under the true model (see e.g. Fig 1). Subsampling and m out of n Bootstrap are indistinguishable for very low subsampling proportions since the impact of duplicate observations decreases with increasing size of the dataset from which we resample. For a proportion of 0.01, both these approaches perform as well as the parametric simulation. These results reflect what we also have seen in all other scenarios,
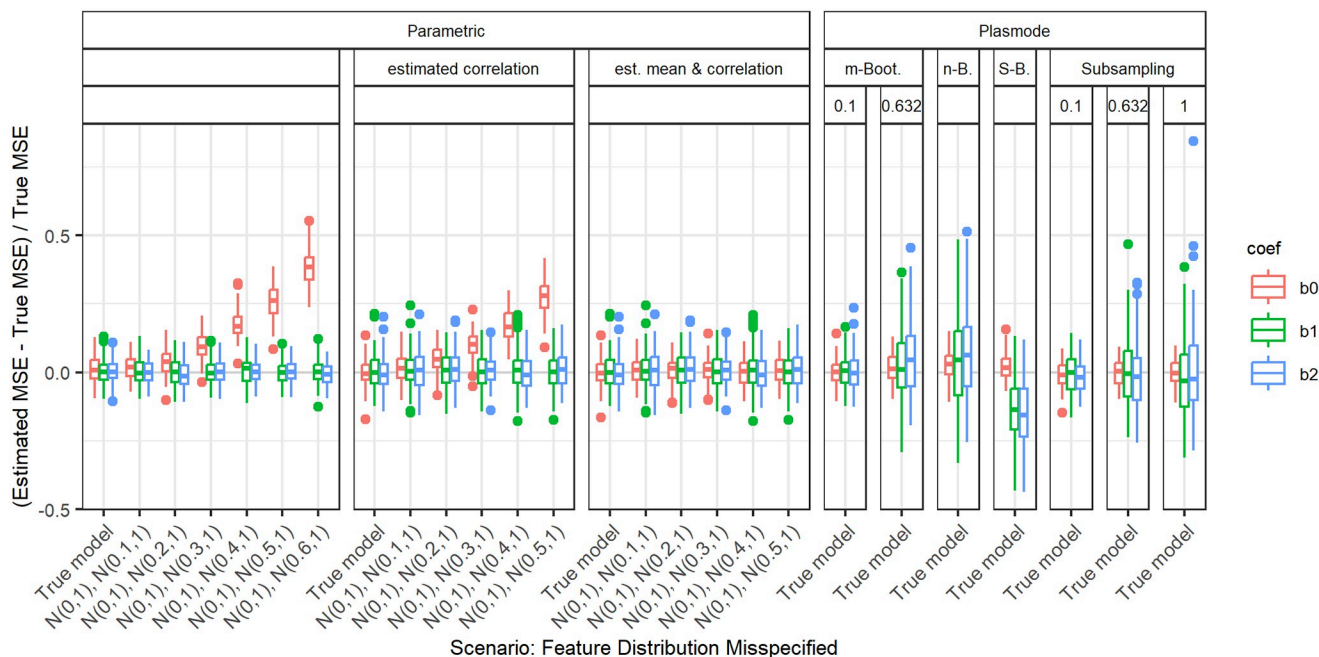
Fig 5. Relative error in MSE estimation for individual coefficients when the assumed mean of the marginal distribution of the second feature in parametric simulation deviates from the true mean, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$. N(0,1), N($\mu$,1) denotes that the first feature is generated from a standard normal (truth), and the second feature is generated from a normal distribution with mean μ instead (deviation).

although for lower p, the differences between the simulation types become very small. It should be noted that in these simulations, subsampling and m out of n Bootstrap require a larger data-set to resample from for lower resampling proportions. This might give them an advantage. Due to its very poor performance, we will exclude the wild Bootstrap from now on. We will also reduce the values of resampling proportions to 0.1 and 0.632 for m out of n Bootstrap and to 0.1, 0.632, and 1 for subsampling for more clarity. The numbers were chosen to represent a relatively low and a relatively high resampling proportion. Moreover, 0.632 has been used in Plasmode simulations, motivated by the expected proportion of non-duplicated observations for n out of n Bootstrap [31].

Deviations from true feature distribution

We will now take a look at the different deviations from the true feature distribution. These only affect the parametric simulation. Since in all cases, different coefficients are affected differently, we always show the individual errors per coefficient. We focus on the case $p = 2$ and $n = 100$ since for this we can still display the errors for individual coefficients in a clear manner. The results can be transferred to higher numbers of features or lower numbers of observations. As expected, the absolute values of the errors are larger for higher values of p or smaller values of n, but the qualitative results are the same. In all cases, we only display the range of deviations that is relevant to the comparison of parametric and Plasmode simulation.

Gaussian with wrong expectation. Fig 5 shows the relative errors in case of deviations from the expectation of the second feature (Feature distribution misspecified N(0,1), N(μ,1), cf. Table 2). We can observe that the second coefficient stays unaffected while the errors for the intercept increase with increasing deviations from the true mean. This result is to be expected, as can be seen by reparametrization. If the truth is $X_2 \sim N(0, 1)$ and we assume

$X_2^a \sim N(\mu, 1)$, $\mu > 0$, we can rewrite the resulting linear model using X_2^a instead of X_2 as

$$
\begin{aligned}
Y &= \beta_0 + X_1\beta_1 + X_2^a\beta_2 + \varepsilon \\
&= \beta_0 + X_1\beta_1 + (X_2 + \mu)\beta_2 + \varepsilon \\
&= \underbrace{\beta_0 + \mu\beta_2}_{=:\beta_0^{new}} + X_1\beta_1 + X_2\beta_2 + \varepsilon.
\end{aligned}
$$

As $\mu > 0$ and $\beta_2 > 0$ in our case, it holds $\beta_0^{new} > \beta_0$.

The errors in the intercept can be prevented by estimating the mean using a dataset sampled from the true DGP. This leads to slightly increased variance in the errors of the parametric simulation, but stable median errors that are close to zero for all coefficients.

Gaussian with wrong variance. Fig 6 shows the relative errors for deviations from the true variance of the second feature. We see that both slope coefficients are affected. The true MSE is underestimated by the simulation and this underestimation gets worse for increasing (misspecified) variance of the second feature.

Again, this behavior can be prevented by estimating the covariance matrix from a dataset from the true DGP at the cost of slightly increased variation. For estimation of the covariance matrix, the dataset from the true DGP must have sufficiently many observations. Here, we used 1000 observations which is sufficient for $p = 2$, as well as for $p = 50$. Smaller numbers of observations are insufficient for $p = 50$ as can be seen in S2 Fig. When increasing the variance of the second feature, the error in MSE converges to an upper bound corresponding to the true MSE, since the estimated MSE converges to zero for increasing variances. This can lead to problems later on, when we look for the first deviation where the aggregated error for
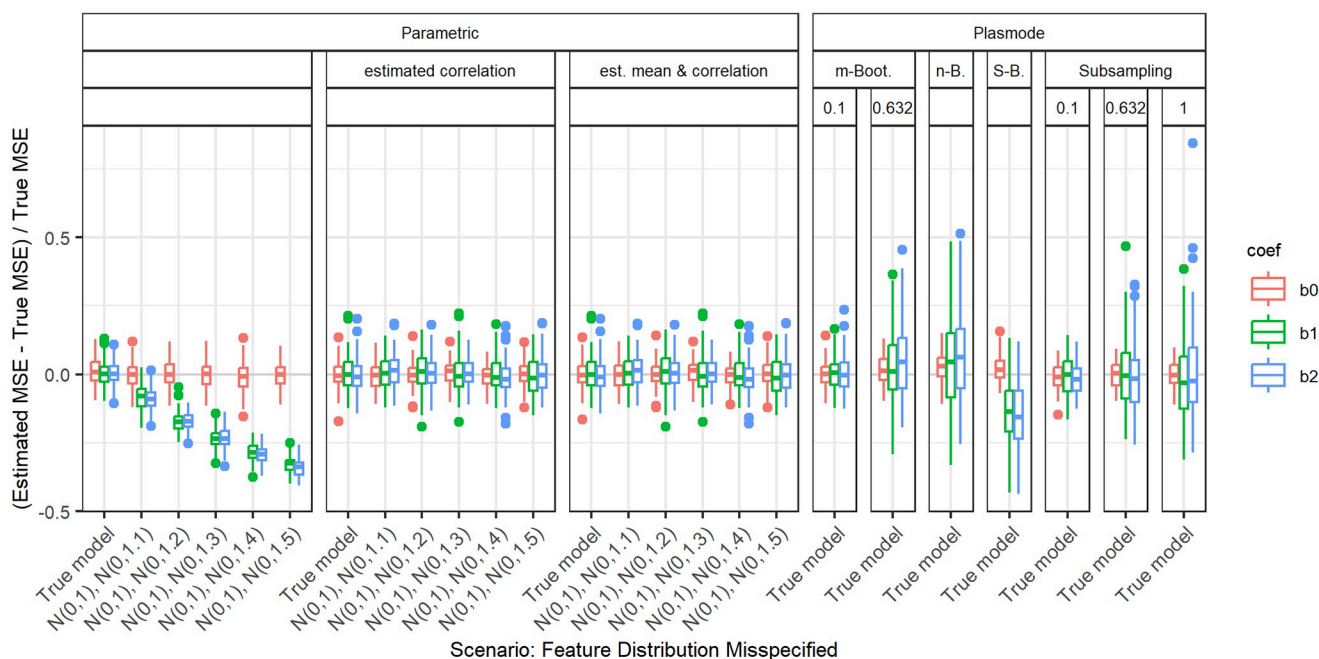

Fig 6. Relative error in MSE estimation for individual coefficients when the assumed variance of the marginal distribution of the second feature in parametric simulation deviates from the true variance, for $p = 2$, $n = 100$, $\boldsymbol{\beta} = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$. $N(0,1)$, $N(0,\sigma^2)$ denotes that the first feature is generated from a standard normal (truth), and the second feature is generated from a normal distribution with variance σ^2 instead (deviation).

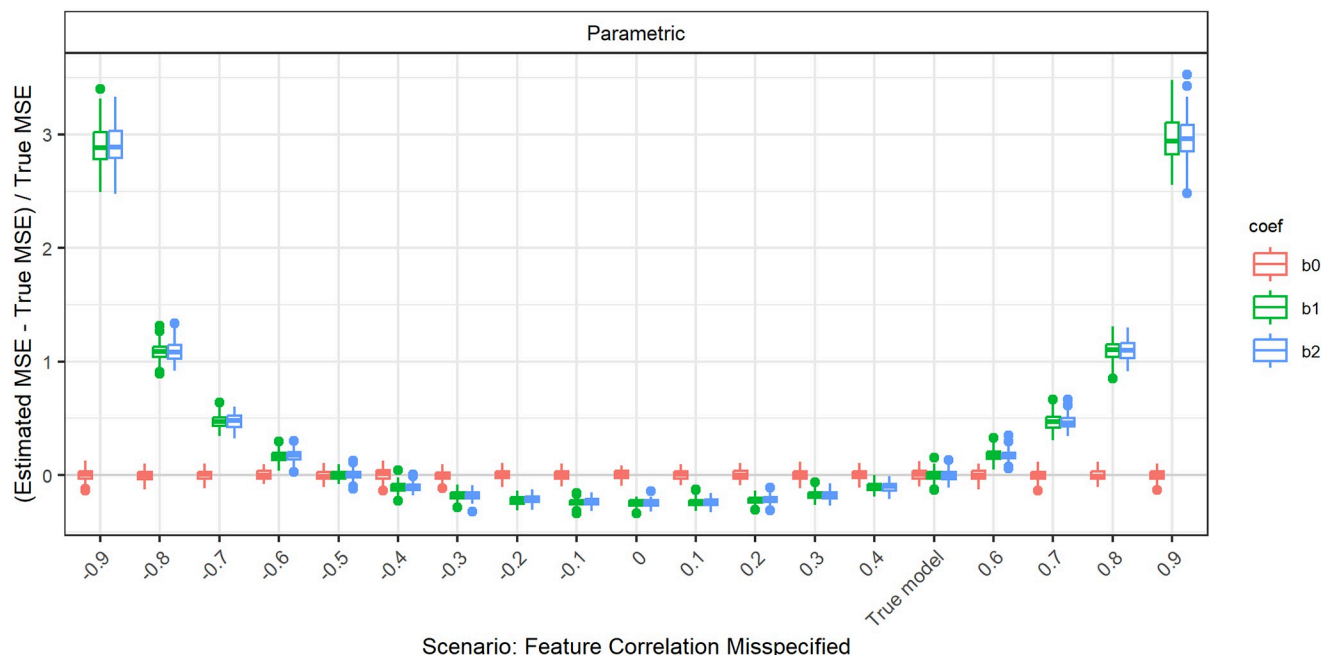https://doi.org/10.1371/journal.pone.0299989.g006

Fig 7. Relative error in MSE estimation for individual coefficients when the assumed correlation of the features in parametric simulation deviates from true correlation, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.5 \; \forall i \neq j$.

parametric simulation exceeds the error for Plasmode simulation. For low p, the upper bound of the error for increasing the feature variance in parametric simulation is still larger than the errors obtained with Plasmode simulation. However, for large p, where Plasmode performs worse, the error reached even with very high values for the variance of the second half of features is smaller than that of some Plasmode types. This is demonstrated in S3 Fig for the case of $p = 50$. There, we show the mean of the relative errors of the coefficients per simulation run. In that case, we do not take the absolute values before averaging, to demonstrate the direction of the errors. In the present case, this is no problem since either the MSEs for all coefficients are overestimated or all are underestimated, so there is no risk of the errors of different coefficients cancelling out in the mean. Decreasing instead of increasing the variance of the second half of features leads to an overestimation of the true MSE and this overestimation is unbounded. Therefore, in settings where the upper bound does not exceed the errors of all Plasmode types, we use decreasing instead of increasing variances, see e.g. S4 Fig.

Gaussian with wrong correlations. The overall influence of misspecifying the pairwise correlations of the features is more easily demonstrated, when the true pairwise correlations are 0.5 instead of 0.2. The relative errors in this case for parametric simulation are shown in Fig 7.

The intercept is unaffected when misspecifying the correlation. For the errors in the slopes, we observe a parabolic shape that intersects with zero at the true correlation of 0.5 and at −0.5. For the MSE estimation, the sign of the correlation does not seem to have any influence, only the absolute value, as the parabolic shape is symmetrical around zero. When overestimating the absolute value of the true correlation, the true MSE is overestimated. For underestimating the absolute value of the true correlation, the true MSE is underestimated.

This pattern is also observed for a true correlation of 0.2 (Fig 8). For the comparison of parametric and Plasmode, we concentrate on assuming a correlation that is higher than the
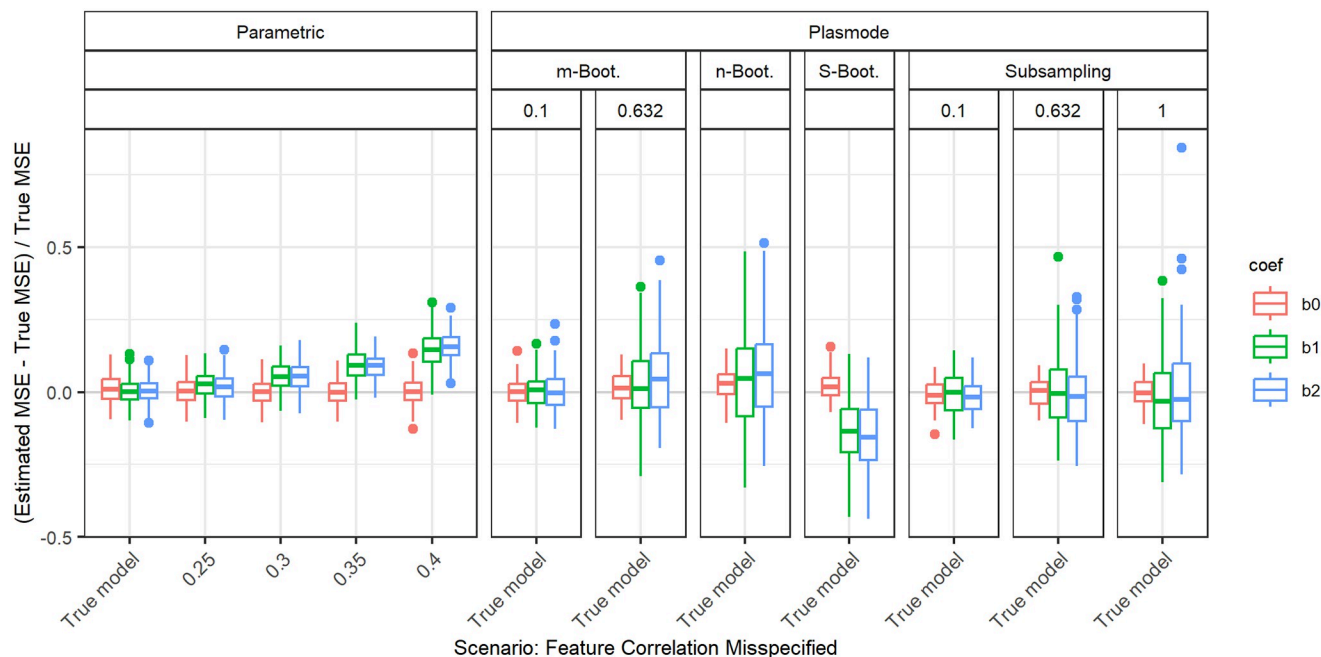
Fig 8. Relative error in MSE estimation for individual coefficients when the assumed correlation of the features in parametric simulation deviates from true correlation, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$.

true correlation, since for these deviations, the errors are monotonously increasing. This can for example be seen in the comparison for true fixed pairwise correlations of 0.2 and $p = 2$, $n = 100$ as shown in Fig 8.

The observed shape is plausible from a theoretical point of view. The MSE of the LSE given X is equal to its variance, as it is unbiased. This variance is given as the diagonal of $\sigma^2 (X^T X)^{-1}$. For X drawn from a multivariate normal distribution, i.e. ignoring the intercept term, $(X^T X)^{-1}$ follows an inverse Wishart distribution. Its expectation is given by the inverse covariance matrix Σ^{-1} of this multivariate normal. When explicitly calculating the diagonal values of Σ^{-1} in case of pairwise fixed correlations of ρ, we can see that this expectation depends quadratically on ρ, which matches the observed form.

When the true correlation matrix has a block structure, we observe lower errors for the coefficients at the margins of the blocks if the value of the correlations but not their structure is misspecified (Fig 9). Again, this can be derived theoretically for the very simple case described above when inserting the block diagonal structure for Σ.

Gaussian mixture. Next, we use two different versions of Gaussian mixtures as feature distributions for the second half of the features. With this, not only the parameter but the whole shape of the distribution is altered. For the first type of Gaussian mixture, a proportion of α of the observations stems from a normal distribution with mean 3 and variance 1. This yields a bimodal distribution. For the second type of Gaussian mixture, a proportion of α of the observations stems from a normal distribution with mean 0 and variance 10. This represents a contamination model with outliers. In both cases, the remaining proportion of $1 - \alpha$ stems from the standard normal, in agreement with the true distribution. We always set the marginal distribution of the first feature to a normal that has the same mean and variance as the Gaussian mixture for the second marginal distribution and successively increase the
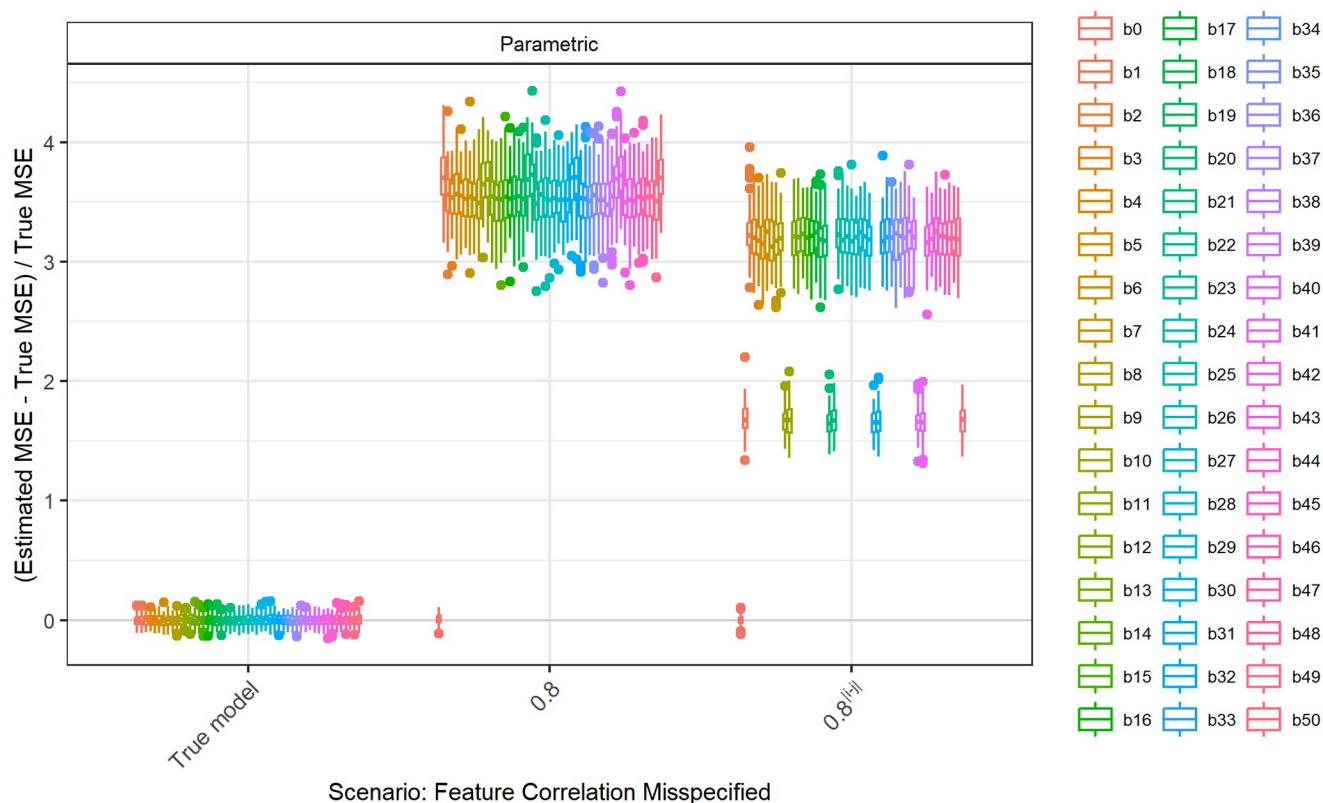
Fig 9. Relative error in MSE estimation for individual coefficients when the assumed correlation of the features in parametric simulation deviates from true correlation, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2^{|i-j|}$ for ith and jth feature within each of the 5 blocks.

proportion in the mixing distribution. This enables us to separate the influence of the change in expectation and variance of the distribution from the effect of the bimodality and outliers.

In the bimodal case (Fig 10) we see that with an increasing proportion of observations from the $N(3, 1)$ distribution, the underestimation of the MSE for the corresponding second coefficient also increases. It is still less pronounced than for the first coefficient which corresponds to the normal with wrong expectation and variance. This might be due to the fact that most of the observations in the mixture distribution belong to the true distribution. In the case of a normal with wrong expectation and variance, all observations come from a distribution that differs from the true one.

For the contamination model (Fig 11) we observe the same behavior, but the differences between the coefficients are smaller there.

Log-normal. Fig 12 shows the relative errors for the individual coefficients when the distribution of the second feature is misspecified as log-normal and the distribution of the first feature is misspecified as a normal with matching mean and variance. There is a large overestimation of the MSE for the intercept, while the MSEs for the other coefficients are underestimated. The underestimation is slightly worse for the second coefficient than for the first, so the additional skewness of the log-normal leads to worse MSE estimation compared to a normal with the same mean and variance. The errors in all coefficients for this deviation are considerably higher than the ones of any Plasmode variant that is compared here.

Bernoulli. Fig 13 shows the relative errors for the individual coefficients when the distribution of the second feature is misspecified as Bernoulli and the distribution of the first feature
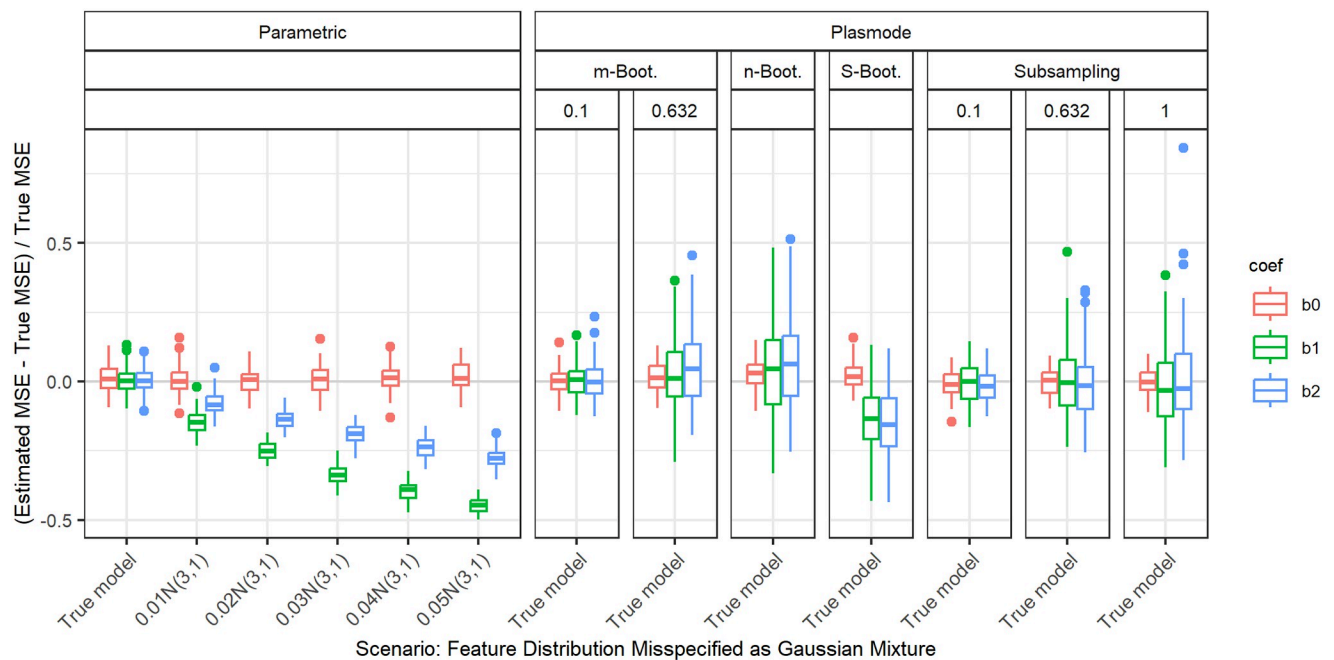
Fig 10. Relative error in MSE estimation for individual coefficients when the assumed marginal distribution of the second feature in parametric simulation is misspecified as Gaussian mixture with increasing proportion of data drawn from Gaussian with different expectations (bimodal distribution). The mean and the variance of the marginal normal distribution of the first feature are set to match those of the second. The mixing proportion is given on the x-axis.

https://doi.org/10.1371/journal.pone.0299989.g010

Fig 11. Relative error in MSE estimation for individual coefficients when the assumed marginal distribution of the second feature in parametric simulation is misspecified as Gaussian mixture with increasing proportion of data drawn from Gaussian with different variance (contaminated distribution), for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2$ $\forall i \neq j$. The mean and the variance of the marginal normal distribution of the first feature are set to match those of the second. The mixing proportion is given on the x-axis.
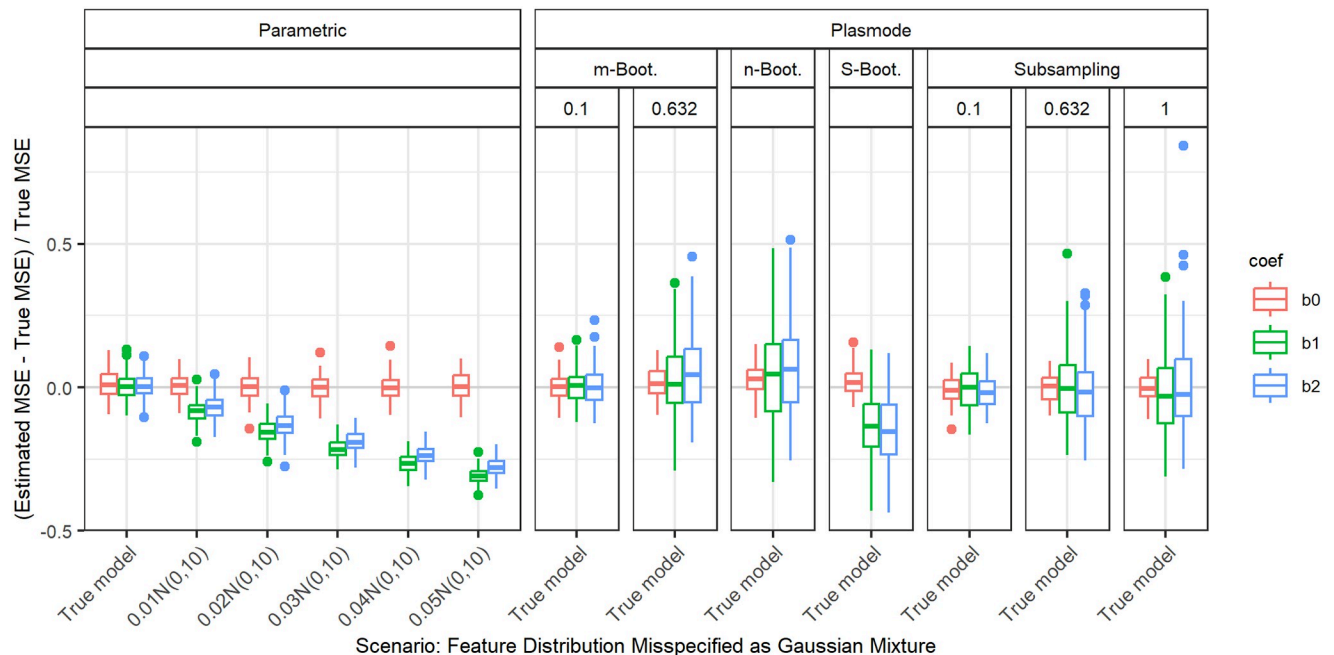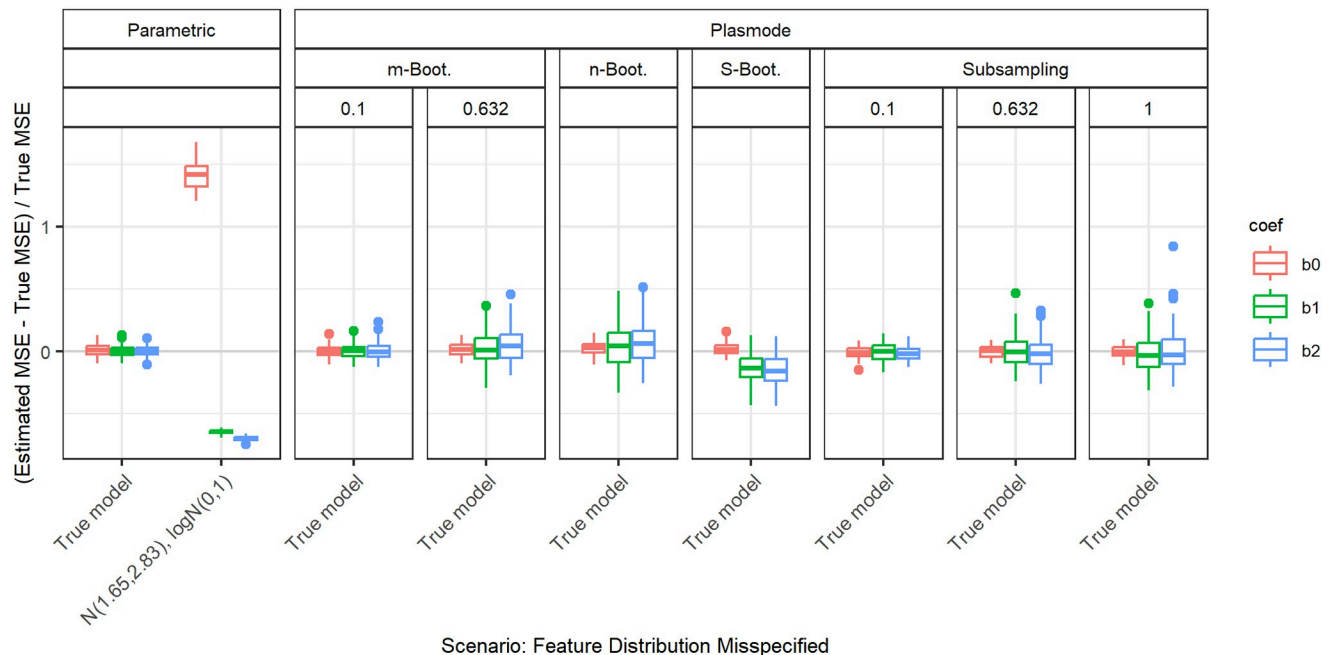
https://doi.org/10.1371/journal.pone.0299989.g011

Fig 12. Relative error in MSE estimation for individual coefficients when the assumed marginal distribution of the second feature in parametric simulation is misspecified as log-normal, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$. The mean and the variance of the marginal normal distribution of the first feature are set to match those of the second.

https://doi.org/10.1371/journal.pone.0299989.g012
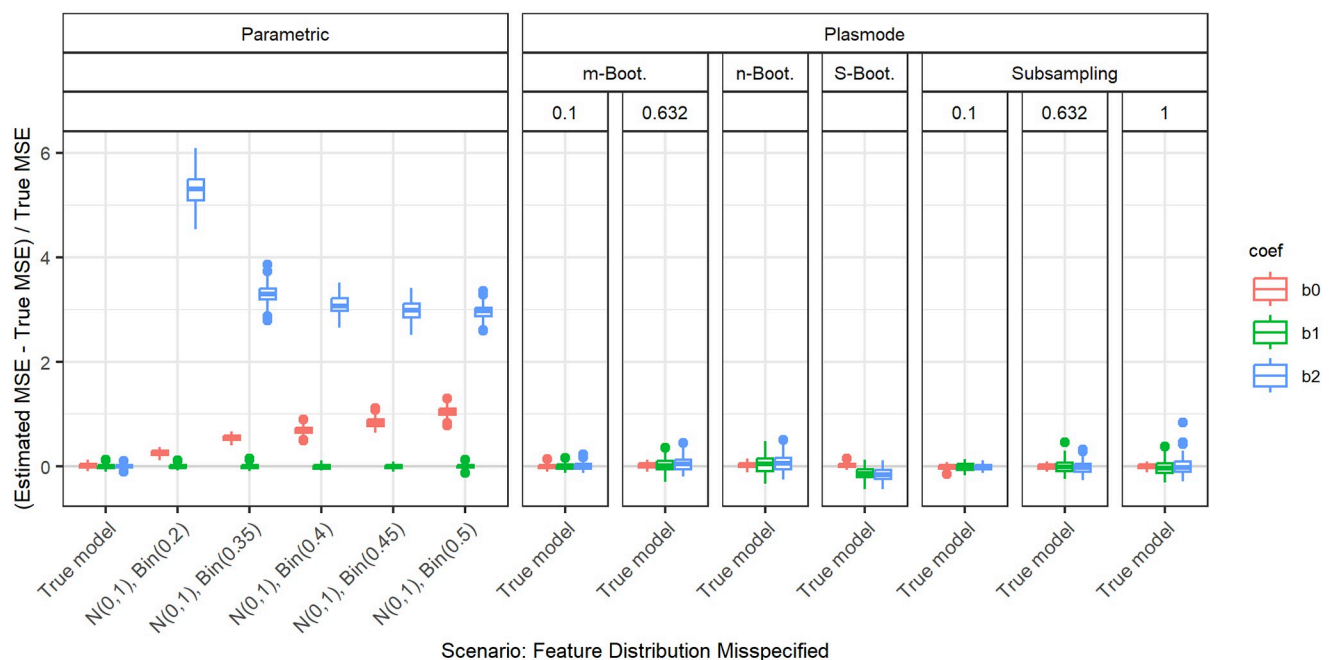
Fig 13. Relative error in MSE estimation for individual coefficients when the assumed marginal distribution of the second feature in parametric simulation is misspecified as Bernoulli with different success probabilities, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$.

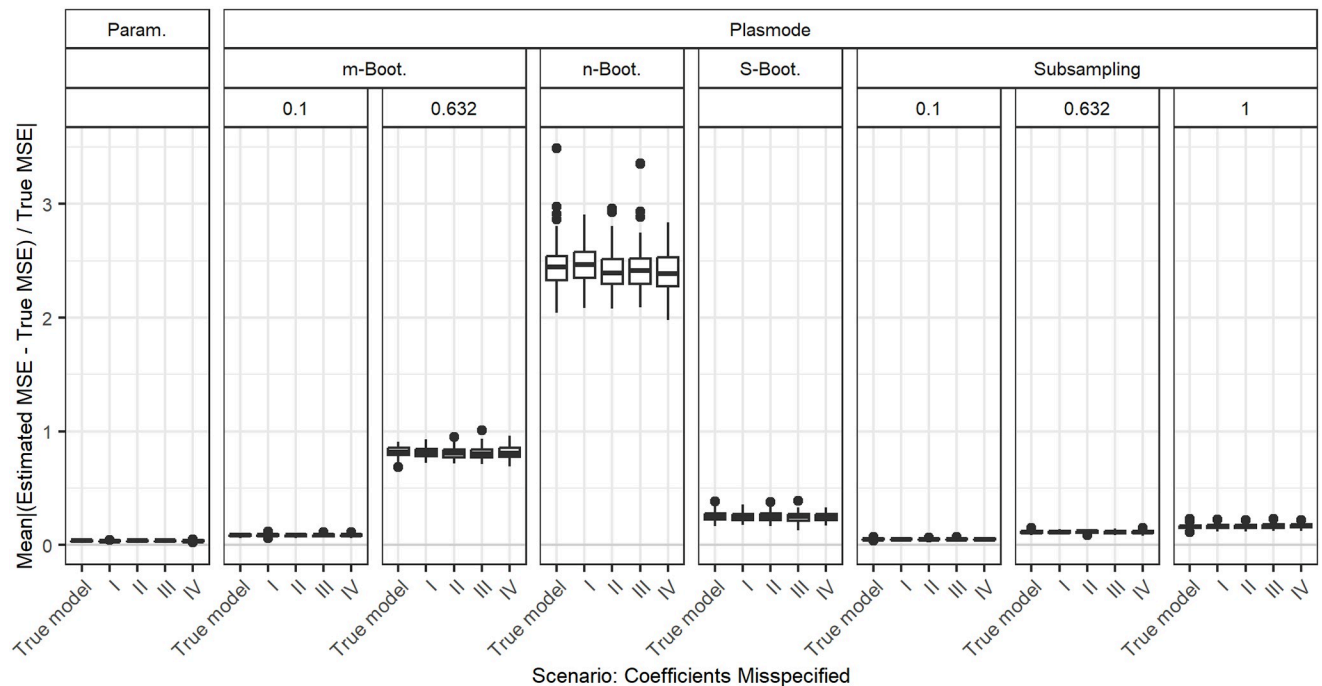https://doi.org/10.1371/journal.pone.0299989.g013

Fig 14. Absolute value of relative error in MSE estimation averaged over individual coefficients when the assumed coefficients in parametric and Plasmode simulation are misspecified, for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \, \forall i \neq j$, $\beta_I = (0, 0.02, \ldots, 1)^T$, $\beta_{II} = 0.05_{51}$, $\beta_{III} = 10_{51}$, $\beta_{IV} = 0_{51}$. Large outliers for n out of n Bootstrap are not displayed.

is correctly specified as a standard normal. We observe an increasing overestimation of the MSE for the intercept with increasing success probabilities. The MSE for the first coefficient is unaffected. The MSE for the coefficient belonging to the binary feature is also clearly overestimated where the overestimation decreases towards success probabilities of 0.5. The errors for the intercept and the second coefficient for this deviation are considerably higher than the ones of any Plasmode variant that is compared here.

Deviations from true coefficients

Fig 14 shows the aggregated relative errors in MSE estimation for $p = 50$ and fixed correlations of 0.2 for misspecifications of the coefficient vector β. Since the specification of the coefficient vector is part of the OGM, this concerns all types of simulations. For each simulation type, the errors for the misspecified coefficients do not differ from the errors for the true model. Therefore, we conclude that the assumed values for the coefficients do not affect the simulation results. The theoretical MSE formula for given X is also only dependent on σ and X, so independent of β.

Deviations from true error variance

In Fig 15, the aggregated relative errors in MSE estimation for $p = 50$ and fixed correlations of 0.2 for misspecifications of the standard deviation of the error term ε are shown. Here, we use the relative errors directly without taking the absolute value to demonstrate under- and overestimation. This again concerns all types of simulation. In general, for too small error standard deviations, the true MSE is underestimated, and for too large error standard deviations, the true MSE is overestimated. This pattern is visible for nearly all types of simulations. For m out
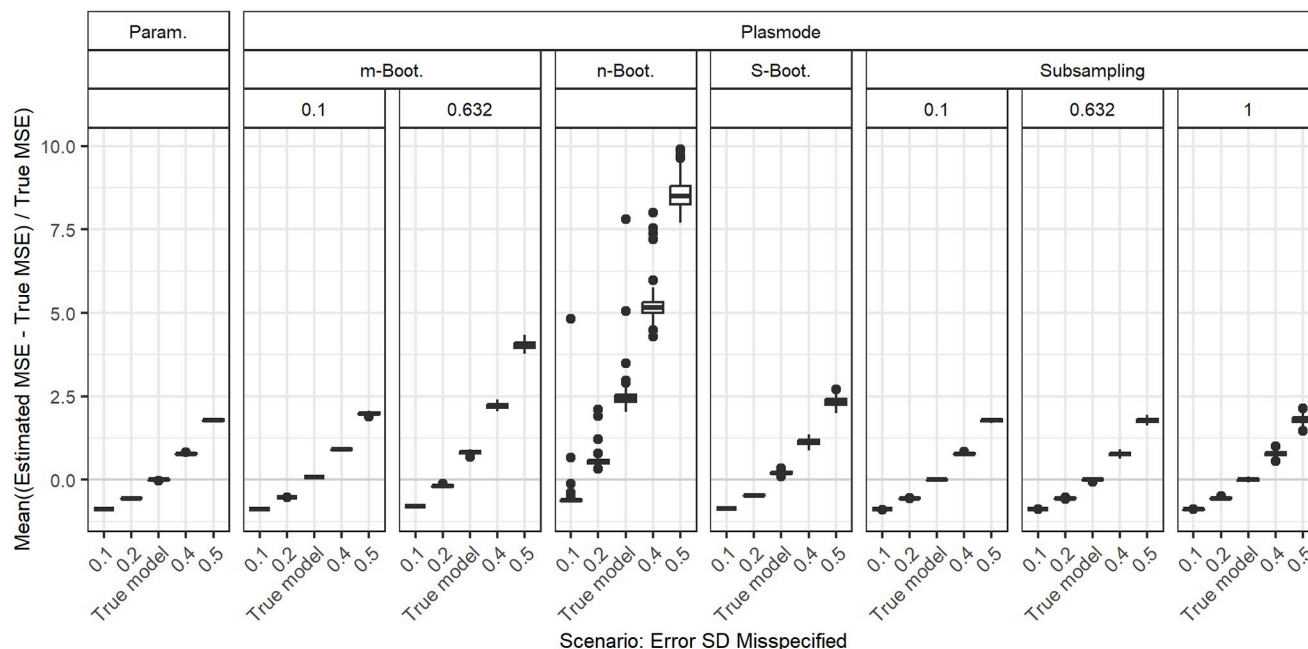
Fig 15. Absolute value of relative error in MSE estimation averaged over individual coefficients when the assumed error variance in parametric and Plasmode simulation are misspecified for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \ \forall i \neq j$. Large outliers for n out of n Bootstrap are not displayed.

of n Bootstrap with large resampling proportions as well as for n out of n Bootstrap, the MSE is overestimated even for the true model, and the errors for other values of the error standard deviation are shifted up accordingly. This leads to values closest to zero for too small error standard deviations. In all cases, the variability of the errors increases with increasing error standard deviation. We observe the same ordering that has already resulted for the true model (see Fig 4) when comparing the errors from different simulation types for misspecified error standard deviations.

Deviations from true error distribution

In Fig 16, the aggregated relative errors in MSE estimation for $p = 50$ and fixed correlations of 0.2 for misspecifications of the distribution of the error term are shown. There are two types of misspecifications that we compare. We use t-distributed errors as an example of a heavier-tailed distribution and χ^2-distributed errors as an example of a skewed distribution. Both are scaled and shifted in a way that the errors still have zero expectation and a standard deviation of 0.3. Overall, the distribution of the errors does not seem to have any influence on the error in MSE estimation as long as the error standard deviation and zero mean are preserved.

True DGP: Correlation estimated from real data

We now analyze the results for the scenarios where the true correlation matrix is estimated from a real dataset. In the following, we only discuss the results that differ from those for the more simple correlation structures we looked at before. These are all deviations that do not alter the correlation matrix. For deviations from the true correlations, it gets more complicated. In the case of small correlations which differ little, the results are still similar to those that we saw before. For example, Fig 17 shows the results for the correlation estimated from

Fig 16. Absolute value of relative error in MSE estimation averaged over individual coefficients when the assumed error distributions in parametric and Plasmode simulation are misspecified, for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \, \forall i \neq j$. Large outliers for n out of n Bootstrap are not displayed.
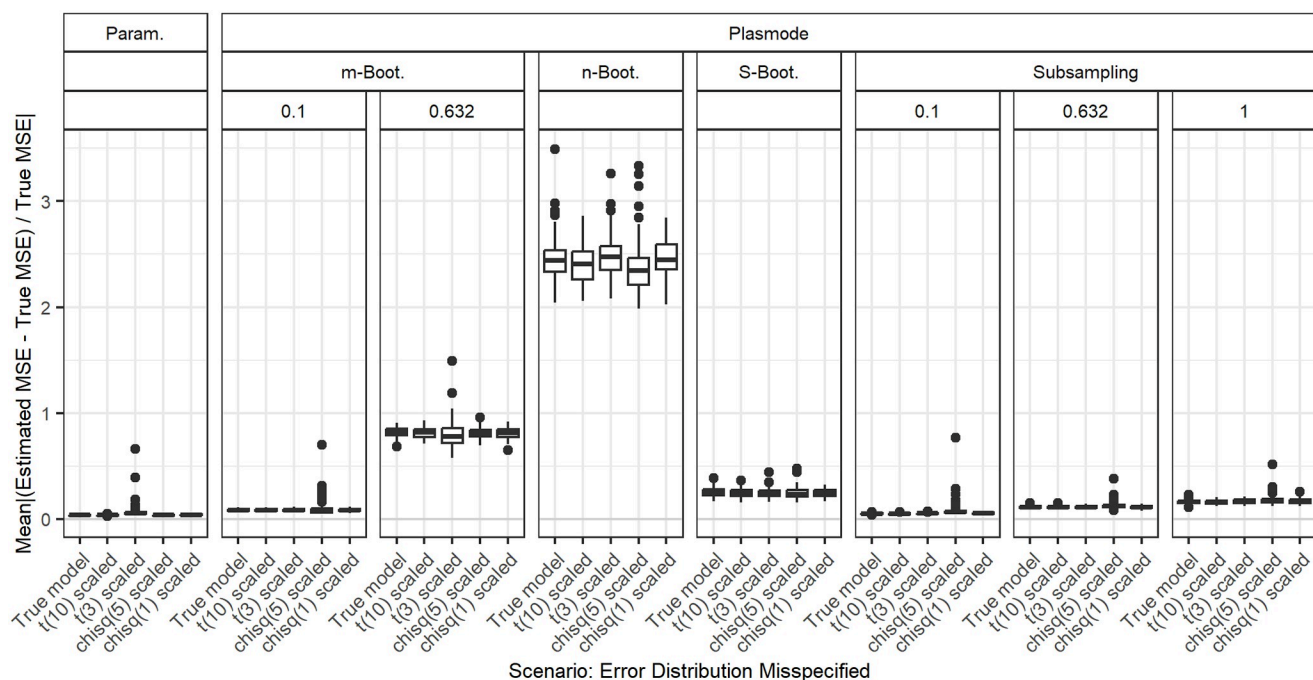
Fig 17. Absolute value of relative error in MSE estimation for individual coefficients when the assumed feature correlation matrix in parametric simulation is misspecified. True correlation matrix is estimated from the benchmark dataset quake ($p = 3$, $n = 100$, $\beta = 1_4$, $\sigma = 0.3$).

the dataset quake. The true pairwise correlations are $\text{Cor}(X_1, X_2) = -0.1286$, $\text{Cor}(X_1, X_3) = -0.0151$, and $\text{Cor}(X_2, X_3) = 0.1353$. The results look similar to those we saw before for fixed correlations of 0.2. On the other hand, for the other datasets, the estimated pairwise correlations show higher variation, which means that no fixed value can be used to approximate all correlations simultaneously in a good way. This is for example clearly visible in Fig 18 for the correlation matrix estimated from the dataset wine_quality. For each choice of fixed pairwise correlation, there are some coefficients with very large relative errors. This can also lead to errors showing a pattern that differs from the parabolic shape we observed before (Fig 7), as can be seen in Fig 19 for the dataset Yolanda. For those cases where no constant correlation approximates all real correlations well, many of the Plasmode variants outperform parametric simulation for all assumed oversimplified correlation structures. A possible cure for parametric simulation would be to estimate the correlation structure from real data which—in this case—corresponds to the true model. Overall, assuming some simple correlation structure, like often done in parametric simulations, might lead to high errors in the estimation of the MSE in cases where the true correlation structure is more complicated. To correctly guess this correlation structure is highly unlikely, and it might even be impossible to specify complicated correlation structures in high-dimensional settings.

Size of resampled datasets

Until now, we have always compared simulations that use the same number of observations, which leads to differently sized datasets from which the Plasmode data is resampled. This might seem unintuitive, but is necessary to ensure a fairer comparison of the simulation methods since the true MSE that the estimations are compared to, is monotonously decreasing in the number of observations in the dataset. Therefore, if we set the size of the dataset that we
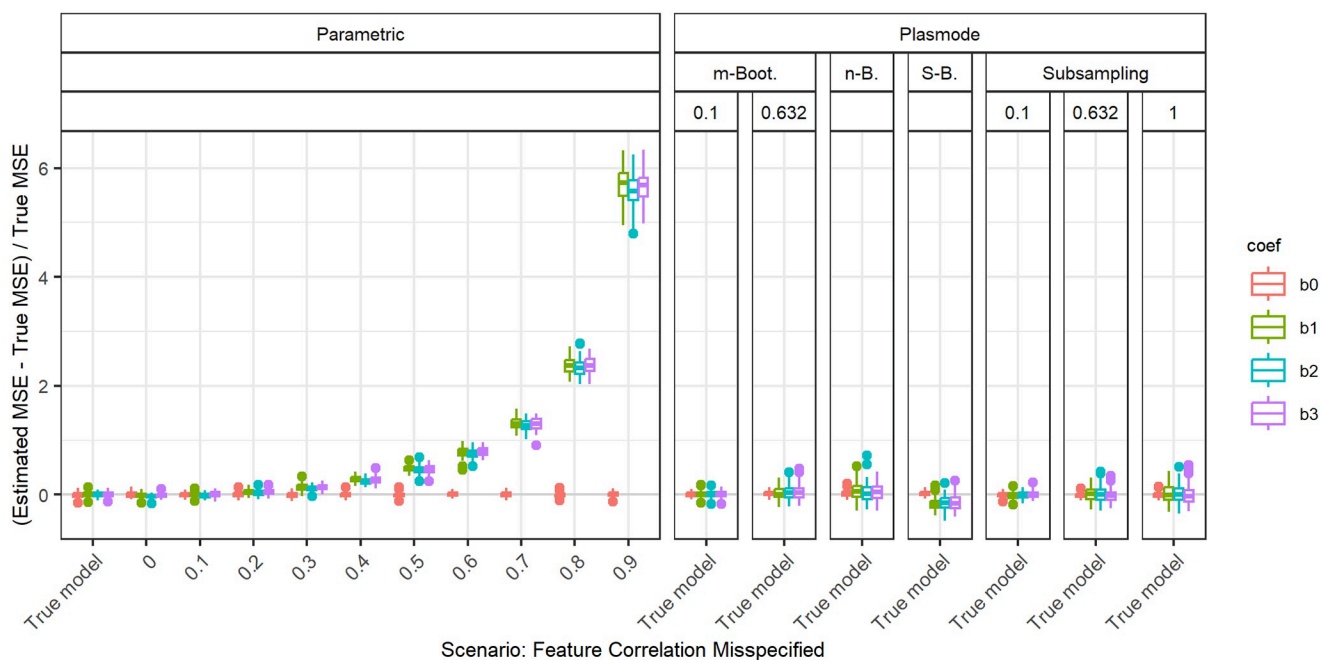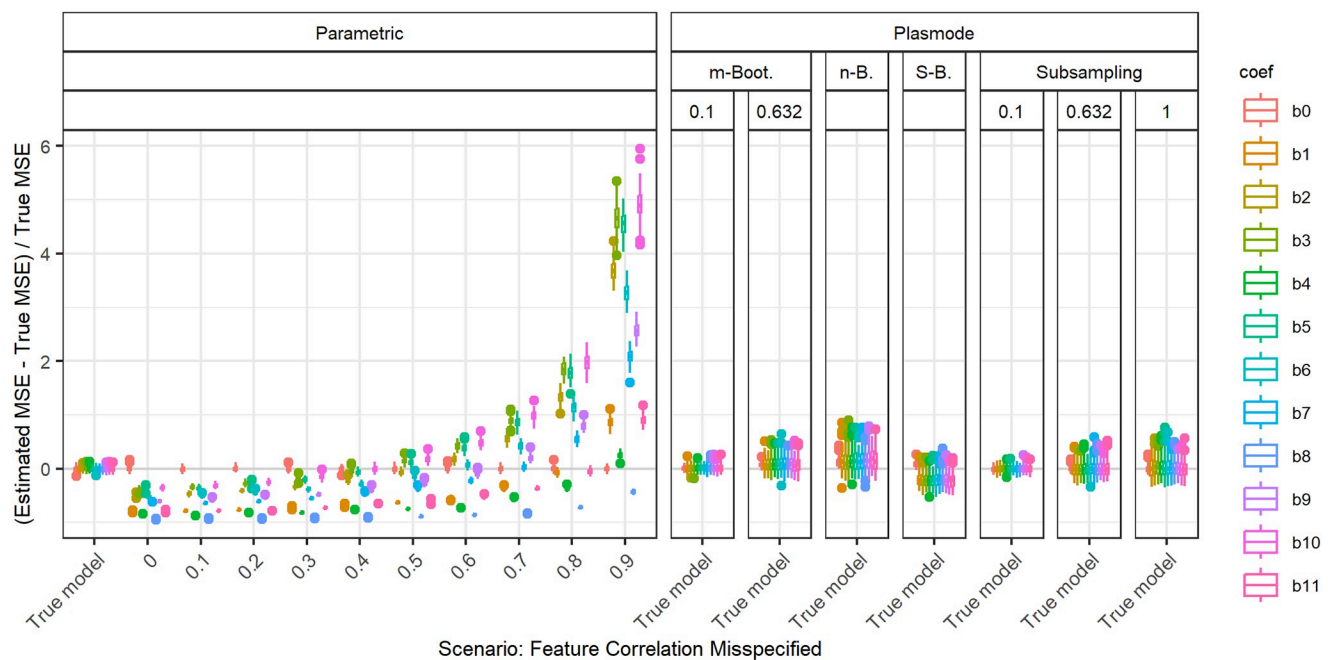

Fig 18. Absolute value of relative error in MSE estimation for individual coefficients when the assumed feature correlation matrix in parametric simulation is misspecified. True correlation matrix is estimated from benchmark dataset wine_quality ($p = 11$, $n = 100$, $\beta = \mathbf{1}_{12}$, $\sigma = 0.3$).
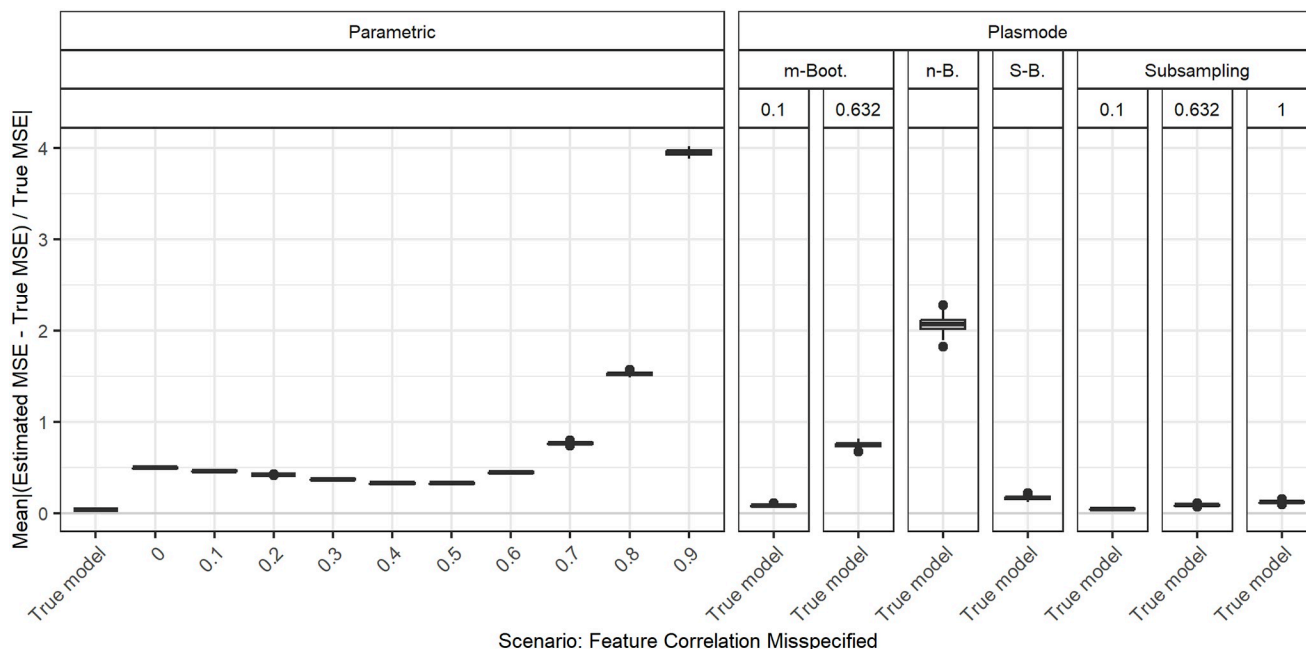
Fig 19. Absolute value of relative error in MSE estimation averaged over individual coefficients when the assumed feature correlation matrix in parametric simulation is misspecified. True correlation matrix is estimated from benchmark dataset Yolanda ($p = 100$, $n = 200$, $\beta = \mathbf{1}_{101}$, $\sigma = 0.3$).

are resampling from to 100 and resample smaller datasets from this, the MSE will always be overestimated, even for the true model. This means that if we want to estimate the MSE for datasets of a certain size n, we have to use datasets of that exact size in our simulations. However, it might be unrealistic that we have a dataset of the correct size at hand to resample from for our simulation. For example, if we use simulation to estimate a quantity that cannot be estimated directly from the data since it depends on unknown parameters (e.g. the bias of an estimator), we might have a concrete dataset at hand for which we want to estimate this quantity. In this case, Plasmode would be a natural choice and since the number of observations is limited, we might use resampled datasets of smaller size to estimate the quantity for the whole dataset. We now discuss the results for this case for $p = 10$ for the true model. For $p = 2$, differences between the resampling methods are very small anyway. For $p = 50$, it will be hard to differentiate between the errors occurring due to the differently sized datasets and the errors caused by approaching the boundary of identifiability. Fig 20 shows the results for the different Bootstrap methods compared to parametric simulation for differing sizes of datasets resampled from a dataset of size 100. For comparison, the case of resampling 100 out of 158 observations that has been used in the analysis so far for a resampling proportion of 0.632 is also included. The estimated MSEs are compared to the true MSE for $n = 100$ in all cases. Higher errors are observed for smaller sizes of the resampled dataset. The smallest errors are observed for subsampling with the subsampling proportion approaching the number of observations in the dataset. So in the case where the number of observations is limited to the number of observations that we are interested in, it might even be the best choice to do no resampling at all and just generate different responses for the MSE estimation. It should be noted that when fixing the size of the dataset to resample from, the n out of n Bootstrap performs comparably well. A reason for this might be that it uses a dataset of size 100 for estimating the MSE. Therefore, no errors occur due to the dependency of the MSE on n. Moreover, the n out of n Bootstrap can
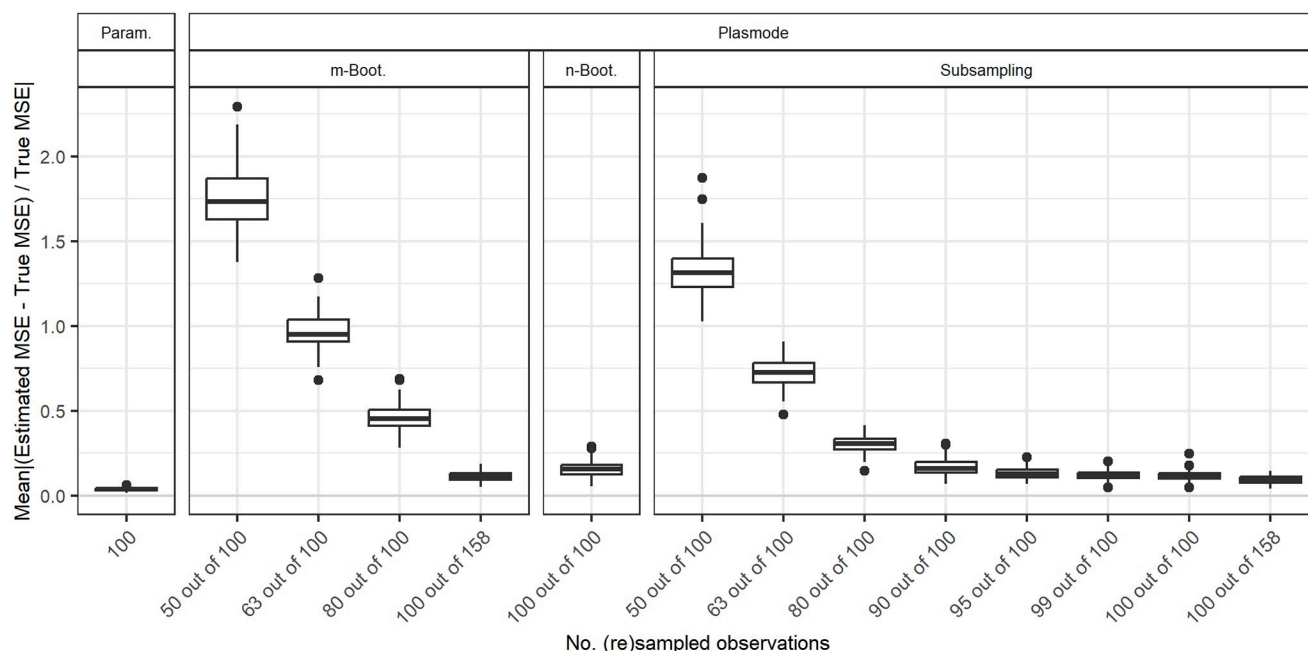
Fig 20. Comparison of different resampling types for different numbers of observations resampled from a dataset with 100 observations. Absolute value of relative error in MSE estimation averaged over individual coefficients when the true model is assumed in parametric and Plasmode simulation, for $p = 10$, $n = 100$, $\beta = \mathbf{1}_{11}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$.

use the dataset more efficiently since it uses more samples for the MSE estimation than sub-sampling or the *m* out of *n* Bootstrap with lower resampling proportions.

Conclusions and recommendations

In the following, we summarize what we have learned from the comparisons that we performed. First, we provide some general insights. Then, we present detailed comparisons, for which type of deviations from the data-generating process Plasmode was superior to parametric simulation in our analyses.

General insights

We looked at different true data-generating processes (DGP) and deviations from those for the estimation of the MSE of the least squares estimator (LSE) in linear regression to compare how well different simulation types perform in this case. Overall, we saw that if there is no deviation from the true scenario, parametric simulation outperforms all Plasmode simulations. The same holds for deviations that affect parametric as well as Plasmode, i.e. deviations from the outcome generating model (OGM), given that the DGP used for parametric simulation is close to the truth. We saw that the misspecification of the coefficients and of the error distribution (as long as expectation and variance are kept) does not have any effect on the quality of the MSE estimation while the misspecification of the error standard deviation does have an effect.

Misspecifications of the DGP only affect parametric simulation. For all kinds of misspecifications of the DGP in parametric simulation (misspecification of expectation, variance, correlation, whole distribution), parametric simulation can become worse than Plasmode. The degree of misspecification needed for Plasmode to be superior depends on the type of

misspecification, the resampling method used in the context of Plasmode that we compare with, and on the number of observations n and the number of features p. A detailed analysis of the degree of misspecification needed for Plasmode to be superior is given in Subsection 2.

Within the different resampling strategies for Plasmode simulations we observed that in general, Wild Bootstrap performed worst, followed by n out of n Bootstrap. m out of n Bootstrap performed better than n out of n and subsampling usually performed best. For both m out of n Bootstrap and subsampling, smaller resampling proportions are favorable. This means that for a fixed number of subsampled observations n of interest, larger datasets to resample from are required. Smoothed Bootstrap usually performs worse than subsampling and even than no resampling (subsampling proportion of one), but better than m out of n Bootstrap with moderate resampling rates, i.e. rates larger than 0.5. When the number of observations for resampling is limited to the number of observations that we are interested in, we are restricted to n out of n Bootstrap, Smoothed Bootstrap, Wild Bootstrap, no resampling at all (i.e. subsampling with the proportion of one), or resampling a dataset of smaller size for Plasmode. Our analyses suggest that no resampling at all or subsampling with a subsampling proportion very close to one might be the best choice in this case. This is due to the dependence of the MSE on the number of observations, which leads to biased estimates of the MSE if the number of observations used for the simulation differs from the number of observations of interest.

Detailed comparisons

Table 3 presents the values for each scenario and deviation at which certain types of Plasmode simulation are superior to parametric simulation. As discussed before, this is only applicable to deviations regarding the data-generating process (DGP). The numbers given in the Plasmode columns are calculated as follows. For the given scenario, deviation and Plasmode type, the deviations are ordered increasingly. Then, the first deviation for which the median aggregated relative error of parametric is higher than that for the Plasmode type is identified. These values correspond to the medians in the aggregated boxplots. For example in the first row, the case of $p = 2$, $n = 100$ and fixed pairwise correlations of 0.2 is analyzed for deviations of the assumed expected value for the second feature. The true expectation is 0. Plasmode with m out of n Bootstrap or subsampling with a resampling proportion of 0.1 is superior to parametric simulation for assumed expectations of 0.25 and higher. Plasmode with m out of n Bootstrap or subsampling with a resampling proportion of 0.632 is only superior for assumed expectations of 0.4 and higher, n out of n Bootstrap for values of 0.5 and higher, Smoothed Bootstrap for values of 0.55 and higher, and Plasmode without resampling (subsampling with proportion of 1) for values of 0.45 and higher.

When using correlation matrices estimated from real datasets, the order for the deviations in the correlations is unclear, as discussed before. Therefore, they are excluded from the comparison. Also, in all cases, assuming log-normal or binary data instead of normal data is worse than all Plasmode variants and therefore also excluded.

For these analyses, in the parametric simulations, the expectations and high variances were increased in steps of 0.05, and the low variances were decreased in steps of 0.1. The mixing proportion for Gaussian mixtures and the pairwise correlations were increased in steps of 0.01.

For $p = 50$ and assuming Gaussian mixtures, in some cases even a proportion of 100% data for the second half of features coming from the wrong distribution is not sufficient for Plasmode to be superior, as can be concluded from the values found for deviating expectations and variances. The corresponding entries in Table 3 are left empty in these cases.

Table 3. Smallest deviations in parametric simulations for which Plasmode simulation is superior to parametric simulation. p denotes the number of features, n the number of observations. True ρ gives the true correlation structure, scenario type the type of deviation and true value the true parameter value that the deviation refers to.

p	n	True ρ	Scenario type	True value	m-Bootstrap 0.1	m-Bootstrap 0.632	n-Bootstrap	Smoothed Bootstrap	Subsampling 0.1	Subsampling 0.632	No resampling
2	100	0.2	Expectation of 2nd feature misspecified	0	0.25	0.4	0.5	0.55	0.25	0.4	0.45
2	100	0.2	Variance of 2nd feature misspecified	1	1.05	1.15	1.15	1.2	1.1	1.1	1.15
2	100	0.2	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.02	0.02	0.03	0.01	0.02	0.02
2	100	0.2	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.01	0.02	0.02	0.01	0.01	0.01
2	100	0.2	Feature correlation misspecified	N(0,1)	0.28	0.35	0.39	0.4	0.29	0.35	0.36
2	50	0.2	Expectation of 2nd feature misspecified	0	0.3	0.5	0.55	0.6	0.3	0.5	0.55
2	50	0.2	Variance of 2nd feature misspecified	1	1.1	1.2	1.25	1.3	1.1	1.2	1.2
2	50	0.2	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.03	0.03	0.04	0.01	0.03	0.03
2	50	0.2	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.02	0.02	0.03	0.01	0.02	0.02
2	50	0.2	Feature correlation misspecified	N(0,1)	0.33	0.41	0.41	0.41	0.29	0.39	0.41
2	50	0.5	Expectation of 2nd feature misspecified	0	0.25	0.35	0.4	0.5	0.25	0.35	0.4
2	50	0.5	Variance of 2nd feature misspecified	1	1.05	1.15	1.15	1.25	1.05	1.1	1.15
2	50	0.5	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.02	0.02	0.03	0.01	0.02	0.02
2	50	0.5	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.02	0.02	0.03	0.01	0.01	0.02
2	50	0.5	Feature correlation misspecified	N(0,1)	0.54	0.57	0.57	0.61	0.54	0.56	0.57
10	100	0.2	Expectation of 2nd half of features misspecified	0	0.25	0.45	0.6	0.7	0.25	0.4	0.5
10	100	0.2	Variance of 2nd half of features misspecified	1	1.1	1.25	1.4	1.55	1.1	1.2	1.25
10	100	0.2	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.02	0.03	0.04	0.01	0.02	0.02
10	100	0.2	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.02	0.02	0.03	0.01	0.01	0.02
10	100	0.2	Feature correlation misspecified	N(0,1)	0.24	0.29	0.33	0.36	0.24	0.28	0.3
10	100	0.2	Feature correlation misspecified $\rho^{|i-j|}$	N(0,1)	0.24	0.38	0.41	0.44	0.24	0.36	0.39
10	50	0.2	Expectation of 2nd half of features misspecified	0	0.3	0.7	0.9	0.7	0.3	0.55	0.65
10	50	0.2	Variance of 2nd half of features misspecified	1	1.1	1.55	2.45	1.55	1.1	1.3	1.45
10	50	0.2	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.05	0.08	0.04	0.01	0.03	0.04
10	50	0.2	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.03	0.06	0.03	0.01	0.02	0.03
10	50	0.2	Feature correlation misspecified	N(0,1)	0.26	0.36	0.43	0.36	0.25	0.31	0.34
10	50	0.2	Feature correlation misspecified $\rho^{|i-j|}$	N(0,1)	0.33	0.44	0.5	0.44	0.22	0.39	0.42
50	100	0.2	Expectation of 2nd half of features misspecified	0	0.4	1.55	2.7	0.8	0.25	0.5	0.65
50	100	0.2	Variance of 2nd half of features misspecified (too small)	1	0.88	0.38	0.17	0.69	0.94	0.84	0.77
50	100	0.2	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.02	0.57		0.05	0.01	0.02	0.03
50	100	0.2	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.98		0.04	0.01	0.02	0.02
50	100	0.2	Feature correlation misspecified	N(0,1)	0.27	0.57	0.78	0.37	0.24	0.29	0.32
50	100	0.2	Feature correlation misspecified $\rho^{|i-j|}$	N(0,1)	0.25	0.62	0.79	0.46	0.34	0.21	0.42
50	100	$0.2^{|i-j|}$ in 5 blocks	Expectation of 2nd half of features misspecified	0	0.4	2.05	2.6	0.8	0.25	0.5	0.6
50	100	$0.2^{|i-j|}$ in 5 blocks	Variance of 2nd half of features misspecified (too small)	1	0.88	0.39	0.17	0.68	0.94	0.84	0.77
50	100	$0.2^{|i-j|}$ in 5 blocks	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.02	0.51		0.05	0.01	0.02	0.03
50	100	$0.2^{|i-j|}$ in 5 blocks	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.26		0.03	0.01	0.02	0.02
50	100	$0.2^{|i-j|}$ in 5 blocks	Feature correlation misspecified	N(0,1)	0.2	0.5	0.74	0.28	0.2	0.2	0.22
50	100	$0.2^{|i-j|}$ in 5 blocks	Feature correlation misspecified $\rho^{|i-j|}$	$0.2^{|i-j|}$	0.3	0.59	0.78	0.41	0.25	0.32	0.36
50	100	$0.5^{|i-j|}$ in 5 blocks	Expectation of 2nd half of features misspecified	0	0.5	2.05	3.4	2.05	0.3	0.6	0.8
50	100	$0.5^{|i-j|}$ in 5 blocks	Variance of 2nd half of features misspecified (too small)	1	0.88	0.39	0.17	0.68	0.94	0.84	0.78
50	100	$0.5^{|i-j|}$ in 5 blocks	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.02	0.43		0.05	0.01	0.02	0.04
50	100	$0.5^{|i-j|}$ in 5 blocks	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.26		0.02	0.01	0.01	0.01
50	100	$0.5^{|i-j|}$ in 5 blocks	Feature correlation misspecified	N(0,1)	0.5	0.67	0.83	0.51	0.5	0.5	0.5
50	100	$0.5^{|i-j|}$ in 5 blocks	Feature correlation misspecified $\rho^{|i-j|}$	$0.5^{|i-j|}$	0.54	0.72	0.85	0.6	0.53	0.55	0.58
3	100	quake	Expectation of 2nd half of features misspecified	0	0.3	0.5	1	1	0.3	0.45	1
3	100	quake	Variance of 2nd half of features misspecified (too small)	1	0.99	0.99	0.99	0.99	0.99	0.99	0.99
3	100	quake	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.02	0.02	0.03	0.01	0.02	0.02
3	100	quake	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.01	0.02	0.02	0.01	0.01	0.01

Summary and discussion

We performed a simulation study to compare the performance of parametric and Plasmode simulation in the context of MSE estimation for the least squares estimator (LSE) in the linear regression model. For parametric simulation, artificial data is generated according to a fully user-specified data-generating process (DGP) for generating the feature data and an outcome-generating model (OGM) for generating the outcome variable. In contrast to that, in Plasmode simulation the feature data is generated by resampling from a real-life dataset and only the OGM has to be specified. For comparing the two approaches, we need control of the true underlying DGP and OGM. We used different true DGPs and OGMs. Since the true DGP and OGM are unknown in practice, they must be specified when conducting a simulation study. For Plasmode simulation the DGP is implicitly given by the chosen dataset. This specification is likely a deviation from the truth. Therefore, we examined the influence of different deviations on both types of simulation studies. Note that for Plasmode, there is no explicit deviation from the DGP. When resampling from a dataset, one samples from the empirical DGP which ideally converges to the true DGP.

Within Plasmode simulations, we compared different resampling strategies, namely n out of n Bootstrap, m out of n Bootstrap, subsampling, smoothed Bootstrap, and wild Bootstrap, and where applicable also different resampling proportions. Each simulation strategy was evaluated based on the differences between the MSEs estimated using the respective method and the true MSEs. If the true DGP and OGM are known, it is obvious that parametric simulation is the optimal choice as long as drawing from the true DGP and OGM is feasible. However, in reality, the true DGP and OGM are unknown and can at best be approximated using expert knowledge. In Plasmode simulations, as long as a dataset from the DGP of interest is given, only the OGM has to be specified. Therefore, our aim was to find out

1. How much the DGP chosen in the parametric simulation can deviate from the truth before the parametric simulation becomes worse than Plasmode simulation.

2. How deviations of the chosen OGM from the true OGM affect both parametric and Plasmode simulations.

3. How the choice of the resampling type affects the Plasmode simulation.

In general, we observed that parametric simulation is superior to Plasmode in all situations where the DGP is specified correctly, i.e. for the true situation or deviations from the true OGM only. For deviations from the DGP in parametric simulation, it depends on the kind of deviation, the degree of deviation, the number of observations, and especially on the number of features in the dataset and the type of resampling used for the Plasmode simulation. For very small deviations, parametric simulations usually remain superior. For low numbers of observations, or especially for higher numbers of features, the performance of Plasmode simulation decreases more drastically than that of parametric simulation both in terms of the median difference between the estimated and true MSE and the variation of the estimated MSE. This means that the deviations from the true DGP in the assumptions of parametric simulation have to be larger for Plasmode to be superior. The effect is more pronounced when using resampling strategies with replacement and a high resampling proportion. A reason for this might be that in these cases, the number of unique observations is lower. Therefore less information is contained in the data, so the variance and consequently the MSE of the estimator is inflated. On the other hand, there are certain settings where Plasmode was always superior to parametric simulation in our study, such as when the DGP was severely misspecified, e.g. when using binary instead of standard normal features.

The effect that Plasmode notably overestimates the true MSE for increasing p, especially for resampling with replacement and high resampling proportions, might be a property of the chosen simulation setup. It is known that using Bootstrap to estimate the variance of the LSE in linear regression models for $p/n \to \kappa \in (0, 1)$ can lead to severe overestimation of the true variance. For n out of n Bootstrap and features sampled from a multivariate standard Gaussian distribution, this property was formally shown and additionally demonstrated via simulation in [32]. Overestimation of the variance implies overestimation of the MSE, so the arguments made in [32] might in part explain the bad performance of Plasmode simulations that we observed. The authors also derived an overestimation of the variance by Jackknifing which is similar to subsampling with resampling proportions very close to one.

If the distribution class of the features was misspecified as log-normal or even Bernoulli instead of normal or if the true correlation matrix of the features is more complex and parametric simulation uses an oversimplified approximation for it, all types of resampling used for Plasmode simulations were superior.

Regarding the resampling strategy used for Plasmode simulations, we observed that wild Bootstrap performed by far the worst with respect to MSE estimation. For the remaining types, n out of n Bootstrap was usually inferior to the other resampling strategies. The performance of m out of n Bootstrap and subsampling depends on the chosen subsampling proportion. Generally, smaller proportions are beneficial. Note that the size of the resulting dataset after resampling has to be fixed, so a smaller proportion corresponds to a larger dataset from which to sample. For small resampling proportions, m out of n Bootstrap and subsampling behave very similarly, for larger proportions, subsampling performs better. Smoothed Bootstrap performs similarly to m out of n Bootstrap and subsampling with moderate resampling proportions. For the resampling proportion approaching one, m out of n Bootstrap converges to n out of n Bootstrap. No resampling (subsampling with a resampling proportion of one), i.e. using the whole dataset for the features and only generating new responses in each iteration of the simulation, performed better than n out of n Bootstrap. The differences between the resampling types except for wild Bootstrap are negligible for small numbers of features ($p = 2$). In that case, Plasmode using any resampling strategy might be a good option since even very small deviations from the DGP lead to parametric simulation being inferior. In general, we suggest using subsampling with a small resampling proportion if feasible.

For larger numbers of features, the performance of Plasmode simulations gets worse in general. Nevertheless, there might still be good reasons for Plasmode simulations in this case. For example, the specification of the DGP gets more and more complicated with an increasing number of features. Especially the specification of the correlation structure is non-trivial as the number of pairwise correlations increases quadratically with the number of features. This might lead to the choice of oversimplified correlation structures for which we observed a clearly inferior performance of parametric compared to Plasmode simulation. A remedy could be to at least estimate key parameters like mean and covariance matrix from a real dataset for the parametric simulation. We observed good results for that strategy at least as long as the dataset from which the parameters are estimated is big enough.

The availability of datasets might be a major limitation for the application of Plasmode simulations. In general, at least one suitable dataset from the DGP of interest is required, see Section 3.2 in [4] for a discussion. Ideally, this dataset is considerably larger than the sample size n of interest, to allow for a low resampling proportion. In practice, this might often not be given. If the dataset size is limited to the sample size one is interested in, our comparison suggests that no resampling (subsampling with a resampling proportion of one) might be a reasonable variant since the error made by using a dataset of the wrong sample size might outweigh the advantage of lower resampling proportions.

Overall, the choice of the simulation type should be carefully considered for each application. A combination of parametric and Plasmode simulation within a simulation study might be a solution to use both the flexibility of parametric simulation and the ability of Plasmode to preserve characteristics of real-life data.

So far, the comparison of parametric and Plasmode simulation is limited to the specific example of estimating the MSE of the LSE. Therefore, further studies on other endpoints as well as a comparison for high-dimensional data which brings additional challenges might be interesting extensions of the analysis at hand.

Supporting information

S1 Fig. Absolute error in MSE estimation for individual coefficients for different types of Plasmode simulation compared to parametric simulation under assumption of true data generating process and outcome generating model. (A) $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$. (B) $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$.
(TIF)

S2 Fig. Relative error in MSE estimation averaged over individual coefficients when the variance of the second half of features is misspecified for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$. The first facet displays the errors in case the misspecified variances are used in the simulation. The remaining facets display the errors for using a variance that is estimated using datasets of different sizes from the true DGP for parametric simulation instead.
(TIF)

S3 Fig. Relative error in MSE estimation averaged over individual coefficients when the assumed variance of the second half of features exceeds the true variance for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$. Large outliers for n out of n Bootstrap not displayed.
(TIF)

S4 Fig. Relative error in MSE estimation averaged over individual coefficients when the assumed variance of the second half of features underestimates the true variance for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2 \; \forall i \neq j$. Large outliers for n out of n Bootstrap not displayed.
(TIF)

S1 Table. Complete list of deviations from true scenarios.
(PDF)

S1 Appendix. Simulation of Bernoulli, log-normal, and Gaussian mixture variables with fixed correlations.
(PDF)

Acknowledgments

Author Contributions

Conceptualization: Marieke Stolte, Nicholas Schreck, Alla Slynko, Maral Saadati, Axel Benner, Jörg Rahnenführer, Andrea Bommert.

Formal analysis: Marieke Stolte.

Methodology: Marieke Stolte, Jörg Rahnenführer, Andrea Bommert.

Software: Marieke Stolte.

Supervision: Jörg Rahnenführer, Andrea Bommert.

Visualization: Marieke Stolte.

Writing – original draft: Marieke Stolte.

Writing – review & editing: Nicholas Schreck, Alla Slynko, Maral Saadati, Axel Benner, Jörg Rahnenführer, Andrea Bommert.

References

1. Boulesteix AL, Lauer S, Eugster MJA. A Plea for Neutral Comparison Studies in Computational Sciences. PLOS ONE. 2013; 8(4):e61562. https://doi.org/10.1371/journal.pone.0061562 PMID: 23637855

2. Morris TP, White IR, Crowther MJ. Using simulation studies to evaluate statistical methods. Statistics in Medicine. 2019; 38(11):2074–2102. https://doi.org/10.1002/sim.8086 PMID: 30652356

3. Boulesteix AL, Groenwold RH, Abrahamowicz M, Binder H, Briel M, Hornung R, et al. Introduction to statistical simulations in health research. BMJ Open. 2020; 10(12):e039921. https://doi.org/10.1136/bmjopen-2020-039921 PMID: 33318113

4. Schreck N, Slynko A, Saadati M, Benner A. Statistical Plasmode Simulations—Potentials, Challenges and Recommendations; 2023. Available from: http://arxiv.org/abs/2305.06028.

5. Cattell RB. A General Plasmode (No. 30-10-5-2) for Factor Analytic Exercises and Research: By Raymond B. Cattell and Joseph Jaspers. Society of Multivariate Experimental Psychology; 1967.

6. Efron B. Bootstrap Methods: Another Look at the Jackknife. The Annals of Statistics. 1979; 7(1):1–26. https://doi.org/10.1214/aos/1176344552

7. Götze F. Asymptotic approximation and the bootstrap. IMS Bulletin. 1993; p. 305.

8. Bickel PJ, Götze F, van Zwet WR. Resampling Fewer Than *n* Observations: Gains, Losses, and Remedies for Losses. Statistica Sinica. 1997; 7(1):1–31.

9. Politis DN, Romano JP, Wolf M. Subsampling. Springer Science & Business Media; 1999.

10. Efron B. Nonparametric Estimates of Standard Error: The Jackknife, the Bootstrap and Other Methods. Biometrika. 1981; 68(3):589–599. https://doi.org/10.1093/biomet/68.3.589

11. Silverman BW, Young GA. The bootstrap: To smooth or not to smooth? Biometrika. 1987; 74(3):469–479. https://doi.org/10.1093/biomet/74.3.469

12. Hall P, DiCiccio TJ, Romano JP. On Smoothing and the Bootstrap. The Annals of Statistics. 1989; 17(2):692–704. https://doi.org/10.1214/aos/1176347135

13. Wang S. Optimizing the smoothed bootstrap. Annals of the Institute of Statistical Mathematics. 1995; 47(1):65–80. https://doi.org/10.1007/BF00773412

14. Wu CFJ. Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis. The Annals of Statistics. 1986; 14(4):1261–1295. https://doi.org/10.1214/aos/1176350161

15. Bickel P, Sakov A. On the Choice of m in the m Out of n Bootstrap and Confidence Bounds for Extrema. Statistica Sinica. 2008; 18.

16. Simonoff JS. Smoothing Methods in Statistics. Springer Science & Business Media; 1996.

17. Cortez P, Cerdeira A, Almeida F, Matos T, Reis J. Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems. 2009; 47(4):547–553. https://doi.org/10.1016/j.dss.2009.05.016

18. Weiss SM, Indurkhya N. Rule-based Machine Learning Methods for Functional Prediction. Journal of Artificial Intelligence Research. 1995; 3:383–403. https://doi.org/10.1613/jair.199

19. Guyon I, Sun-Hosoya L, Boullé M, Escalante H, Escalera S, Liu Z, et al. In: Analysis of the AutoML Challenge series 2015-2018; 2017. Available from: https://www.automl.org/book/.

20. Vanschoren J, van Rijn JN, Bischl B, Torgo L. OpenML: networked science in machine learning. SIGKDD Explorations. 2013; 15(2):49–60. https://doi.org/10.1145/2641190.2641198

21. Gijsbers P, LeDell E, Thomas J, Poirier S, Bischl B, Vanschoren J. An Open Source AutoML Benchmark. CoRR. 2019;abs/1907.00909.

22. Astivia OLO, Zumbo BD. Population models and simulation methods: The case of the Spearman rank correlation. British Journal of Mathematical and Statistical Psychology. 2017; 70(3):347–367. https://doi.org/10.1111/bmsp.12085 PMID: 28140458

23. Emrich LJ, Piedmonte MR. A Method for Generating High-Dimensional Multivariate Binary Variates. The American Statistician. 1991; 45(4):302–304. https://doi.org/10.1080/00031305.1991.10475828

24. Silverman BW. Density estimation for statistics and data analysis. Chapman & Hall/CRC; 1986.

25. R Core Team. R: A Language and Environment for Statistical Computing; 2023. Available from: https://www.R-project.org/.

26. Genz A, Bretz F, Miwa T, Mi X, Leisch F, Scheipl F, et al. mvtnorm: Multivariate Normal and t Distributions; 2021. Available from: https://CRAN.R-project.org/package=mvtnorm.

27. Genz A, Bretz F. Computation of Multivariate Normal and t Probabilities. Lecture Notes in Statistics. Heidelberg: Springer-Verlag; 2009.

28. Wolodzko T. kernelboot: Smoothed Bootstrap and Random Generation from Kernel Densities; 2023. Available from: https://CRAN.R-project.org/package=kernelboot.

29. Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York; 2016. Available from: https://ggplot2.tidyverse.org.

30. van den Brand T. ggh4x: Hacks for'ggplot2'; 2023. Available from: https://CRAN.R-project.org/package=ggh4x.

31. De Bin R, Janitza S, Sauerbrei W, Boulesteix AL. Subsampling versus Bootstrapping in Resampling-Based Model Selection for Multivariable Regression. Biometrics. 2016; 72(1):272–280. https://doi.org/10.1111/biom.12381 PMID: 26288150

32. Karoui NE, Purdom E. Can we trust the bootstrap in high-dimension?; 2016.