to appear in Philosophical Transactions of the Royal Society A

Statistical Workflow

Andrew Gelman* Aki Vehtari† Richard McElreath‡

5 Dec 2025

Abstract

We consider several aspects of data analysis that are underemphasized in most presentations of statistical theory and practice. We illustrate some of these with a simple example of Bayesian workflow and conclude by emphasizing shared aspects of Bayesian and non-Bayesian data analysis workflows. The audience for this paper includes statisticians and applied researchers who might not be aware of these commonalities across apparently opposing statistical philosophies.

We organized this special issue on statistical and scientific workflow because we believe that there are shared aspects of quantitative research that are obscured by the varieties of models, methods, and even philosophical frameworks that are successfully employed in statistics and machine learning, and also because it seems that many of the most important aspects of statistical practice, in whatever form, do not make their way into the textbooks. So we solicited a range of articles from leading statisticians and scientists who work in a variety of applied fields, with the hope of obtaining a kaleidoscopic view of workflow from many perspectives.

As anticipated, the different approaches shared by these researchers cannot be easily combined into a master workflow, and we have not tried to collate them. Instead, in this introduction to the special issue we go over some often underestimated aspects of statistical workflow, and we discuss how they are relevant from both Bayesian and non-Bayesian perspectives.

Presentations of statistics and machine learning typically focus on the scenario in which a single model or procedure is fit to a single dataset. Sometimes there is also discussion of testing a model, diagnosing problems with fit, and selecting or averaging among a set of models. In contrast, real-world data analysis is typically an iterative process involving multiple models and integration of different sorts of data. The general problem of integrating data, domain knowledge, mathematical modeling, and computation has been labeled as veridical data science (Yu and Kumbier 2020; Yu and Barter 2024), and integrating these practical steps into a general theoretical framework has been an ongoing challenge.

We can think of statistical and machine learning methodology as a process of increasing codification, from examples to case studies to workflows to methods to theories. In the present article we do not attempt to lay out a comprehensive statistical workflow; rather, we highlight several aspects of data analysis that are underemphasized in most presentations of statistical theory and practice.

We are coming from the perspective of our own Bayesian workflow (Gelman, Vehtari, et al. 2020), but we think these principles apply more generally, and so in each section we consider Bayesian and non-Bayesian frameworks.

1. Some often neglected aspects of data statistical workflow

1.1. Measurement

A crucial but often neglected aspect of any quantitative analysis is measurement—not just a probability model or likelihood function linking data to parameters, but the choice of what to

*Department of Statistics and Department of Political Sciences, Columbia University, New York.
†Department of Computer Science, Aalto University, Finland.
‡Max Planck Institute for Evolutionary Anthropology, Leipzig, Germany.

measure in the first place. Often we are restricted to analysis of available data, in which case it can be helpful to introduce latent variables representing the ideal observations we would like to have. Familiar examples arise in educational testing (with test scores representing measurements of some hypothesized underlying abilities), psychiatric diagnoses, and biometric data that vary during the day but end up being measured at different times for different people.

In the Bayesian framework it is easy to add an additional level of modeling to bridge between measured data and underlying quantities, although some care needs to be exercised in setting up a prior distribution for bias, variance, and correlations of measurement errors. In the setting where the number of latent variables increases linearly with sample size, they are typically not well identified from data, hence the prior cannot be neglected.

In non-Bayesian inference, latent data can be included into a model and then integrated out; alternatively, analytical or simulation work can be done to estimate the bias incurred from ignoring the latent process (for example, attenuation bias in measurement-error regression), and then a correction can be made to construct approximately unbiased estimates under the assumed model.

1.2. Use of prior information

Prior information can enter a statistical analysis in many places. In Bayesian inference, the specification of the data model represents prior information or assumptions, whether this be linearity, a particular functional form such as the logistic, or the structure of a neural network or Gaussian process. Further assumptions arise when specifying dependence in time series, spatial data, and other structures, and families of probability distributions. Whether chosen with relevance to the problem at hand or as conventional or default choices, these correspond to prior information or, equivalently, some partial specification of the class of problems to which a particular model would be used.

Prior knowledge also enters into non-Bayesian methods. Design and sample size of a study will be chosen based on some balance of cost and prior assessment of possible effect sizes and the scale of variation. Prior knowledge guides the choice of transformations, regression predictors, the error distribution, and other information included in the model, as well as what is considered as data to be analyzed, along with structural aspects of a model such as the depth and number of nodes of a network and tuning parameters for regularization. Tuning parameters can sometimes be set using cross validation, but there typically will be aspects of a model or method that will be set ahead of time.

1.3. Combining information from multiple sources

A virtue of Bayesian inference is that it allows integration of different sources of information: different sorts of data can be modeled with different data distributions conditional on a shared set of parameters, which can in turn be allowed to vary. This sort of hierarchical modeling allows inferences to be partially pooled, so that the data analysis does not need to choose between no pooling (analyzing each source of data separately) or complete pooling (denying the difference between the sources). Bayesian inference can thus allow combination of data from different experiments, different scenarios, and different places and times. For example, in Weber et al. (2018), we fit a pharmacokinetic model using informative priors to partially pool the parameter inferences for two different, but related, drugs.

Non-Bayesian methods can also be used to combine information, which is sometimes done using discrete rules (for example, combining datasets when their estimated parameters are not statistically significantly different from each other) and sometimes using hierarchical models fit non-Bayesianly

(for example, using marginal maximum likelihood estimates of the hyperparameters). Data that are not on the same scale or with different measurement biases can be combined using methods such as factor analysis.

Our point here is not to try to match each non-Bayesian method to a Bayesian model but rather to note that, whatever inferential framework is being used, the problem of combining different sources of information is important, and ideas such as leave-one-out predictive evaluation and simulation-based calibration checking should apply.

1.4. Regularization

When data are sparse, or for complicated models in which data are locally sparse—that is, not highly informative about particular parameters—it is important to *regularize* inferences, that is, to constrain them in some way to ensure their stability.

In the Bayesian framework, regularization is performed by the prior distribution, which serves as a soft constraint keeping parameter inferences close to their priors. Ideally this can all go smoothly, but there can be prior-likelihood conflict. In non-Bayesian inference, regularization can be implemented in a similar way using a penalty function—for example, penalized maximum likelihood is equivalent to posterior mode estimation. Other non-Bayesian regularization methods that can be given Bayesian interpretations include wavelet shrinkage (Donoho and Johnstone 1994; Chipman, Kolaczyk, and McCulloch 1997) and dropout in deep learning (Srivastava et al. 2014; Gal and Ghahramani 2016).

Three alternatives to regularization are: (1) to use unregularized estimates which are too noisy to be useful, (2) to simplify the model (which can be considered as an extreme form of regularization in which some parameters are constrained to be exactly zero), or (3) to include more data in the analysis (which can be considered as a form of regularization in which any systematic differences between old and new data are assumed to be exactly zero).

1.5. Using simulation to capture uncertainty

An attractive feature of Bayesian simulation is that uncertainties are propagated automatically. A defining characteristic of Bayesian inference is that all uncertainties are modeled probabilistically; from that perspective, our own applied workflow is not fully Bayesian. We usually operate within a framework in which we do not assume that we have enumerated all possible models of the data. And, for reasons discussed in detail in Chapter 7 of Gelman, Carlin, et al. (2013), we do not usually think it makes sense to try to compute posterior probabilities of different candidate models. Instead we prefer to express our uncertainty about model choice using predictive model averaging or Bayesian stacking, as discussed in Yao et al. (2018). We can then still summarize inferences by simulations, which represent a sort of pseudo-posterior distribution that we can propagate to get simulations, and thus inferences with uncertainty, about any summaries of parameters or predictive quantities.

To continue our theme: non-Bayesian inference can do this too, with the simplest version being to take the point estimates of the parameter vector, along with an estimate of the Fisher information (for many models this will be the negative second derivative matrix of the log likelihood) to represent uncertainty. You can then draw from the corresponding multivariate normal distribution, which serves as a pseudo-posterior distribution for the parameter vector. Under certain conditions, this will be asymptotically Bayesian. But in many settings the posterior is not approximately normal, and uncertainty quantification based on a point estimate and information matrix will fail. In that case there are Bayesian or non-Bayesian methods based on approximating the marginal likelihood (Pinheiro and Bates 2000; Rue, Martino, and Chopin 2009). To put it another way, in non-Bayesian

workflow, simulation or bootstrapping can be viewed not as an expression of uncertainty but as a computational tool to construct approximately calibrated inferences (for example, Krinsky and Robb 1991; DiCiccio and Efron 1996).

1.6. Prediction, generalization, and causal inference

The aims of our inferences are not always the same as the parameters in our models. We are often interested in predictions for new cases, generalization to new population, and causal inferences, which can be expressed in terms of predictions and generalizations. Consider a regression model, $p(y|x,\theta)$, where we have inferences for the parameter vector θ. A prediction for a new value $\tilde{x}$ would have the predictive distribution $p(\tilde{y}|\tilde{x},\theta,y)$, with uncertainty driven by the posterior uncertainty of θ, that is, for each draw $\theta^s,\ s=1,\ldots,S$, we could sample one draw $\tilde{y}^s \sim p(\tilde{y}|\tilde{x},\theta^s)$ and then collect these to obtain S simulations of the predictive quantity $\tilde{y}$. Generalizing to a new population is done in the same way except that $\tilde{x}$ is now a vector of length $\tilde{N}$ representing the values of the predictors in the population of interest, and the result is an $S \times \tilde{N}$ matrix of simulations.

Causal inference follows a similar structure except that we are now interested in differences of counterfactuals: if there is a binary treatment z, then the causal effect for a new item with predictors $\tilde{x}$ is $\tilde{y}^1 - \tilde{y}^0$. Typically we are interested in a population average treatment effect (PATE), which, for each posterior simulation draw θ^s, can be written as,

$$\text{PATE}^s = \frac{1}{\widetilde{N}} \sum_{i=1}^{\widetilde{N}} \left(\text{E}(y|x=\tilde{x}_i, z=1, \theta^s) - \text{E}(y|x=\tilde{x}_i, z=0, \theta^s) \right), \tag{1}$$

where $\tilde{x}_i,\ i=1,\ldots,\widetilde{N}$, represent the values of the predictors in a population that is assumed large enough that we only care about the expected values of the outcome. The S posterior simulation draws θ^s propagate to S values of PATE^s in (1), which we can take as draws representing the posterior distribution of the treatment effect of interest. This is all assuming the measurement and data collection process allows for causal identification; here we are focusing on the challenges of generalization within that context.

For some simple models such as linear regression with no interactions, the population average treatment effect is a single coefficient in the model, but more generally (1) will have to be calculated from the fitted model.

Prediction, generalization, and causal inference are important in non-Bayesian inference too: the larger principle is that we often want to make inferences not just for parameters in the model but for functions of parameters, observed data, and latent data, and this will require some propagation of uncertainty, especially in settings with nonlinearity and dependence of predictive quantities so that one cannot simply plug in point estimates, nor can we combine standard errors of parameters and predictions to get uncertainties for derived quantities. So even in a non-Bayesian context it will make sense to use predictive simulations using some sort of proxy for a posterior distribution, and some assumptions will be required to extrapolate to populations of interest, perhaps using a weighted bootstrap or some other approach that is not explicitly model-based.

1.7. Visualizing model checking and model fit

Some modeling is typically required when combining data from multiple sources. Statistical inference is traditionally framed unidirectionally: you are given a model or procedure which can then be fit to data. In practice, though, our models are at best approximations and at worst the product of unthinking habit or convention, so it is important to check their fit to data and also to check that inferences and predictions make sense.

The goal here, whether in a Bayesian or non-Bayesian context is not to "reject" null hypotheses—we know ahead of time that just about all of our models are wrong—but rather to explore their problems, and our most useful diagnostic tools will not be p-values or other numerical summaries but rather graphs that display data in comparison to fitted models, following the general principles of exploration and model checking laid out by Tukey (1977), Box (1980), and Rubin (1984). Sometimes these comparisons are implicit—for example, a graph of residuals will ideally be patternless with zero average and no trend—while other times it will help to directly juxtapose graphs of observed and simulated data.

In Bayesian inference we use prior, posterior, and cross validation predictive checks, but these principles apply more generally. Non-Bayesian statistical methods typically do not employ fully generative models: if there is a sampling distribution, $p(y|\theta)$ but no prior distribution, $p(\theta)$, then we can perform predictive simulations but they will have to be conditional on some point estimate of θ or a distribution representing uncertainty about the parameters. Again, Bayesian methods of generative simulation and predictive comparison can be used for non-Bayesian purposes, in this case the goal of finding problems with model fit. Indeed, exploratory data analysis more generally can be enhanced by explicit generative models (Gelman 2003, 2004; Hullman and Gelman 2021). Predictive model evaluation should be part of any statistical workflow.

The flip side of model checking is the concern of overfitting: if we have a general procedure to respond to misfit to data by expanding a model until it fits, then we are effectively excluding certain legitimate but unlikely data patterns, and our models are no longer even theoretically correct. This problem of post-selection inference arises with Bayesian as well as classical statistical workflow (Berk et al. 2013; Taylor and Tibshirani 2015). In practice, we would like our final model to account for as much information as possible, and when we might be selecting among a large set of possible models, we prefer to embed these in a larger model, perform predictive model averaging, or use all of the models simultaneously. As discussed by Gelman, Hill, and Yajima (2012), we expect that would work better than trying to formally model the process of model checking and expansion.

1.8. Fitting a sequence of models rather than focusing on just one

Statistical and machine learning theory and methods focus on fitting, and perhaps checking, one model at a time. When multiple models are fit, the problem is often framed as choosing or averaging among them, but here we want to emphasize the idea that models exist in relation to each other. Examples include regressions in which predictions are added one at a time; models in pharmacology with one, two, three, or more compartments; mixture models with increasing numbers of components; multilevel models in which more and more parameters are allowed to vary; and adding dimensions to a factor analysis.

In all these cases, there are several motivations to start simple and add complexity one step at a time. To start with, simpler models are typically easier to understand. Not always—for example, when predicting an outcome that is constrained to fall between 0 and 1, a nonlinear S curve with asymptotes can be easier to parse than a linear regression where we have to worry about boundary issues—but typically a model with fewer parameters and simpler functional forms will be easier to interpret, a useful starting point even if further complexity is unavoidable.

The next point is that complicated models can often be best understood in relation to simpler special cases. If we start with a comparison of averages among an treatment and control group, and then account for pre-treatment variables one at a time, we can see what each additional adjustment does to the estimated causal effect. A related point is that this sort of comparison can be valuable in itself: for understanding an analysis it is useful to know, for example, that the treatment group in a medical was older than the control group and that the benefit of the treatment appeared larger after

adjusting for age. Even something as simple as age adjustment can be tricky (Gelman and Auerbach 2016); our point is that these adjustments, however complicated, can often be best understood in steps.

Another benefit of building up models one step at a time is that sometimes we can reach the pleasant state of having a model that fits well and does the job. At this point it can be helpful to try adding a bit more complexity, just to show that this additional step is not necessary. Again, even setting aside questions of model choice, this extra model can help our understanding.

The final benefit of fitting a sequence of models is computational. Simpler models are often easier to fit—although not always, as adding hierarchical structure to a model can enhance computational stability and improve posterior geometry—and, once we have successfully fit one model, we can use its inferences as a starting point when moving forward. This is related to the practice of using modeling ideas to address computing problems.

Again, all these ideas apply with non-Bayesian inference as well. So even if you are not thinking of your model or method in terms of priors and posteriors, we think it should be valuable for your data-analytic workflow to include bridges between models of varying complexity.

1.9. Initial values and tuning parameters

When performing inference for a parameter vector θ, the *initial values* are the values of θ used to start the computation, and they can often be chosen based on inferences from simpler versions of the model. The *tuning parameters* are the settings on the algorithm (for example, in Hamiltonian Monte Carlo these are the step size, mass matrix, and number of steps per iteration), which themselves must be initialized and are then tuned during an adaptation or warmup stage.

It is appealing for fitting algorithms to run entirely on their own. Unfortunately, initial values can matter, even in simple optimization problems. For problems where default initialization does not work, it is important for reproducibility purposes to include starting points in publicly-available code. Similarly, as algorithms become more complicated they can require more care in tuning parameters; again, it can make sense to start with values that have already worked with simpler versions of a model.

Initial values and tuning parameters can be important for non-Bayesian computation as well, and our point here is that choosing these is part of any computational workflow.

1.10. Simulation-based calibration checking

There are lots of ways that computation can fail—including programming errors, nonidentified models, and difficult geometry that make it challenging for variational or simulation algorithms to traverse the target distribution—and so it is important to check that our computations are doing what they are supposed to be doing. This is not just a problem of an algorithm working in general—converging to the target distribution in a reasonable time and with computational stability—but also working for the particular model and data at hand.

Bayesian computations can be checked using simulation from the prior predictive distribution: the idea is to draw a "true value" of the parameter vector θ^{true} from the prior distribution, $p(\theta)$, then simulate hypothetical data y from the data model, $p(y|\theta^{\text{true}})$, then draw posterior simulations θ^{post} from the posterior distribution, $p(\theta|y)$. It is that last step that is typically the most challenging. When this simulation procedure is repeated many times, the joint distribution, $p(\theta^{\text{true}}, y, \theta^{\text{post}})$ should be symmetric in $(\theta^{\text{true}}, \theta^{\text{post}})$: that is, the joint distribution should be identical whether computed forward or backward (Cook, Gelman, and Rubin 2006; Modrák et al. 2025).

Similar ideas can be applied for non-Bayesian methods but with the challenge that, with no prior distribution, there is also no joint distribution of data and parameters, and thus no extended distribution, $p(\theta^{\text{true}}, y, \theta^{\text{post}})$. Instead, computation can be checked by setting θ^{true} to a fixed value—not drawing it from a nonexistent prior—then simulating $y|\theta^{\text{true}}$, then performing inference on θ, and finally repeating this procedure many times and evaluating the statistical properties of the inferences, for example checking that unbiased estimates actually have the correct expected value under simulation and checking that confidence intervals have their nominal coverage. Except asymptotically or with very simple models, these statistical properties will be only approximate and will depend on the true parameter values, and so it will make sense to perform these simulation checks for reasonable choices of θ^{true} (Savitsky and Gershunskaya 2024).

Even for this sort of approximate simulation checking, it should still be useful—even necessary—when we want to build trust in complex fitting algorithms in statistics and machine learning. Being able to recover true parameter values under ideal conditions is a minimum requirement of any inferential computation, and simulation is a general way to check this.

1.11. Understanding methods by applying them to multiple problems

The practice of fitting multiple models to data is an example of the more general idea that applied data analysis should follow the principles of scientific investigation, including hypothesizing, data collection, and evaluation. In this case, the hypotheses are not statistical assumptions such as "the data are normally distributed," "the relation is linear," or "the effect is zero," but workflow conjectures such as "a linear regression model will be good enough for this problem," "Stan will fit this model in reasonable time," or "Existing data will give us sufficient accuracy for our current inferential goals." Meanwhile, data collection within workflow refers not to collection of actual measurements but rather simulations and inferences, and evaluation refers to various checks that our fitting procedures are working as they are supposed to. For all but the simplest problems, data analysis is an iterative procedure involving many tries of constructing statistical models to approximate the underlying processes of interest, and a fair amount of experimenting to see what will work with the data at hand and where new information is needed.

But there is another dimension to consider, which is that any statistical method will be applied many times—or, if not, we should at least consider the possibility that this could happen. We say this not just as textbook writers and software developers, for whom some choices of defaults and some expectation of repeated use is to be assumed, but also as applied researchers: we want to use methods with good statistical properties, which means that they would give reasonable answers when applied to a series of problems.

Indeed, the very concept of frequency properties of a statistical method implies some "reference set" of problems for which the procedure will be used, and the Bayesian counterpart to this is the prior distribution, which represents an ideal population of true parameter values (Gelman 2018). Either way, there is an implicit population and implicit replications. We can make use of this idea, again, using simulations, this time by simulating replicated data under various assumptions and re-fitting models, and also more generally by thinking of any statistical procedures as not just producing a one-time estimate but also as a mapping from data to inferences. One motivation for wanting to understand the influence of data and priors (or, in a non-Bayesian context, external constraints) is that we should want to know how our methods will work in other settings. The best way to get something to work once is to frame it as a more general problem. Also, working on more problems helps each of us develop our own statistical workflow.

Dose, x_j (log g/ml)	# of rats, n_j	# of deaths, y_j
−0.86	5	0
−0.30	5	1
−0.05	5	3
0.73	5	5

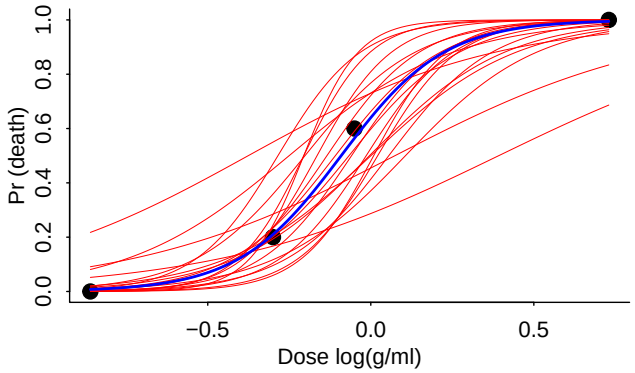

Figure 1: *(a) Data from an experiment on 20 rats exposed to four doses of a toxin, (b) Graph with data, 20 draws from the posterior distribution of the logistic regression,* $\Pr(y_j = 1) = \text{logit}^{-1}(a+bx_j)$*, fit with a weakly informative prior on* a, b*, and the curve plotted using the posterior mean of the parameters.*

1.12. Awareness of goals

Smullyan (1979) wrote, "To know the past, one must first know the future." The application of this principle to statistics is that design and data collection should be aligned with how you plan to analyze your data (Gelman 2024). For a simple example, if you collect a random sample of data (x, y) from a population where x is uniformly distributed between 0 and 1, you should be able to get a good estimate of the average linear regression of y on x in that interval and, with a large enough sample, some sense of departures of $\text{E}(y|x)$ from linearity. Extrapolating beyond this range requires more assumptions, whether they be expressed as priors or variation bounds or restrictions on functional forms. Similar issues arise when estimating interactions: the goals of the analysis will determine where more modeling and data analysis effort is appropriate. Textbook presentations of statistical problems as parameter estimation or hypothesis testing often do not take into account the relevance of inferential and decision goals in data analysis workflow.

2. Example

We demonstrate some aspects of statistical workflow by going through the steps of Bayesian analysis and computation for a small problem. This example is not intended to cover all of workflow or even all the topics covered in this article; rather, we are trying to give a sense of the way that even a simple data analysis is a bit of a research problem involving investigations and unanticipated challenges.

2.1. Setting up and fitting a model

We work through an example from Racine-Poon et al. (1986), also included in Section 2.8 of Gelman, Carlin, et al. (2013), of a logistic regression fit to a bioassay experiment. Figure 1a shows the results of giving specified doses of a toxin to 20 rats. We fit a model assuming independent binomial data with the logistic probability of death being a linear function of dose,

$$y_j \sim \text{binomial}(n_j, \text{logit}^{-1}(a + bx_j)),\ j = 1, \ldots, J = 4.$$

There are no difficult measurement issues for this simple experiment in which life or death is recorded after a fixed time of observation, but in a study of metabolism, for example, we would need to consider accuracy of measurement in the context of variation over time.

To return to the problem at hand, we start by assuming a uniform prior, which here is improper because the parameters a and b are not bounded. With the given data, the posterior in this case is proper, but an improper prior can lead to an improper posterior density and computational problems.

We fit the model in Stan. To visualize the posterior, we plot in Figure 1b the observed proportion of deaths, y_n/n_j, along with the curves, $\text{logit}^{-1}(a + bx_j)$, corresponding to 20 random draws from the posterior distribution of (a, b), and the curve using the posterior mean of the parameters as shown in Figure 1. As discussed in Section 1.7, this sort of graph facilitates the comparison of fitted model to data.

2.2. Checking the computation by fitting to simulated data

A good way to understand a model is to simulate data from it. In our workflow, there are two ways of doing this. We can alter our existing Stan program so that, instead of reading in data, it inputs assumed true values for the parameters in the model, then simulates new data and fits the model to these data, with the goal of seeing if this fit recovers the true parameter values (within the stated range of uncertainty).

Usually, though, when we perform this sort of simulated-data experimentation we simulate the data in our home programming environment (typically R or Python), fit the model by calling Stan, and then go back to R to compare inferences to assumed or simulated true values. This workflow has the advantage that the pre-processing and postprocessing can be done interactively, and we can perform new comparisons on the fly without having to add them to the model.

We start by simulating data from the model, using as true parameter values the posterior median estimate ($\hat{a} = 1.2, \hat{b} = 10.5$) obtained earlier from the model fit to the real data. We then fit the model to the simulated data, which yields posterior mean and standard deviation of 2.7 ± 1.6 for a and 11.7 ± 5.7 for b. The posterior simulations can be used to construct central 90% intervals, which come to $(0.7, 5.7)$ for a and $(4.3, 22.3)$ for b, which comfortably contain the assumed true values.

At this point we can loop the simulation and assess the coverage of the posterior inferences. If we simulate 100 replicated datasets and then, for each, fit the model, obtain posterior inferences, how will they compare to the assumed true values. Should we expect the posterior means of a and b, on average, to equal the true values, a_{true} and b_{true}? Should we expect that the posterior 90% intervals should include the true parameters 90% of the time?

The short answer to the above questions, is that no, we would not expect this sort of exact calibration (see Cai et al. 2024), even on average, unless the true parameters were drawn from their prior distribution that is being used in the fitted model, and this is not even possible given that we are using an improper prior. However, we would like to check that there is some rough calibration, some sense that we can approximately recover the true parameters, if the data really are draw from the model.

So we loop the above simulation 100 times—the dataset is small so the computation will not take long. If the problem were larger, we could distribute these computations in parallel to a cluster, but here we can just run a loop on our local computer.

But a problem comes up. Many of the simulations look fine, as with the example above, with good mixing of the computation and posterior distributions for a and b being close to their assumed true values. But occasionally the computation seems to blow up; for example, from the model fit to one of our simulated datasets:

```
variable  mean    median       sd      mad       q5       q95 rhat ess_bulk
  lp__    -3.90    -3.60     0.75     0.32    -5.36     -3.37 1.00     1917
  a     4771.92  4423.61  2987.22  3303.79   630.97   9361.48 1.48        7
```

```
b   95428.72 88470.35 59744.33 66092.68 12623.81 187230.70 1.48        7
```

The problem here is indicated by a high value of the convergence diagnostic `rhat` and low effective sample size `ess_bulk`. This indicates that the program failed to properly explore the posterior distribution.

2.3. Problem of identification from sparse data

What happened? Maybe it would help to look at the simulated data. We re-run the model (starting everything from scratch from the beginning so as to be using the same random seed), and the simulated data vector `y` is printed out during the Stan run:

```
[0,0,3,5]
```

Looking carefully, we see that these data convey very little information! With those counts of 0, the probability of death at low doses could be arbitrarily close to 0, and with that count of 5 at the other end, the probability of death at the high dose could be arbitrarily close to 1. The logit probabilities of death at low and high doses are thus potentially unbounded negative and positive numbers, respectively, and thus there is nothing in the data constraining the slope b. This is the problem of separation, which has been considered in Bayesian and non-Bayesian frameworks (Firth 1993; Gelman, Jakulin, et al. 2008).

By contrast, the original data contained two intermediate points—the counts of 1 and 3 (see Figure 1)—which served to rule out extremely large values of b. The simulated data contained only one point that was informative in this way, which was not enough to constrain b on the high end. This leads to an improper posterior distribution, and manifests in Stan as poor mixing as there is nothing in the likelihood stopping the chains from drifting toward the limit $b \to \infty$.

This simulated-data exercise tells us that with these values of x and n, the model cannot be reliably fit from data—even if the model being fit is the true data-generating process! The parameters in the model cannot be identified given these data.

There are two ways of fixing this identification problem: we can be lucky enough to have enough intermediate cases or we can fix this identification problem with a proper prior. In this scenario, a proper prior makes sense: infinite slopes would only be plausible if the process were completely deterministic (in which case, why are we doing this?) so we are comfortable encoding the prior knowledge that the slope should be finite.

2.4. Adding prior information to get stable inferences

We assign weak normal$(0, 5)$ priors to the coefficients a and b. Why do we characterize these as weak priors? Start by considering the dose x, which is on a logarithmic scale. A slope $b = 1$ would then imply that a difference of 1 in the log dose would correspond to a difference of 1 in the logistic probability of death; that is, comparing two doses that differ by a factor of 2.7, the probability of death could shift from $\text{logit}^{-1}(-0.5)$ to $\text{logit}^{-1}(0.5)$—that is, from 0.38 to 0.62. Given that this is a toxin, such a large slope seems plausible.

What about a slope of 10? This would imply that, comparing two doses that differ by a factor of 2.7, the probability of death could shift from $\text{logit}^{-1}(-5)$ to $\text{logit}^{-1}(5)$—that is, from 0.01 to 0.99. Without any specific knowledge of the toxin involved, we would judge this to be on the edge of plausibility: it seems doubtful that increasing the dose by a factor of 2.7 would increase the probability of death by so much, but it could be possible. If $b = 10$ is on the edge of plausibility, then a normal$(0, 5)$ is a soft constraint mostly constraining b to be less than 10 in absolute value. This weakly informative prior cannot be making use of all the potential prior information in the problem

at hand—for one thing, we set it up in general terms without reference to the particular toxin being studied; also, the very fact that it is a toxin suggests that we would expect the probability of death to increase with dose so that b would be positive. We further constrain the slope b to be positive, representing our knowledge that we are measuring the effects of a toxin.

It makes sense to set up a weakly informative prior for a as well. Here we can take advantage of the fact that the log-doses x are in a range that includes 0. Given that the experiment is performed with only 20 rats, it seems reasonable to expect that probability of deaths at $x = 0$, which in this case is near the middle of the data, will not be too close to 0 or 1. We feel comfortable assigning a normal$(0, 5)$ prior distribution to a as well, implying that we are pretty sure that $\mathrm{E}(y|x = 0)$ is between $\mathrm{logit}^{-1}(-10) = 4.5 \cdot 10^{-5}$ and $\mathrm{logit}^{-1}(10) = 1 - 4.5 \cdot 10^{-5}$. This seems like a very weak prior, and on the off chance that it is applied to a toxin where the probability of death there really is closer to 0 or 1 than that, we recognize that our inference will mostly rule out such possibilities, not really such a concern given the resolution available given data from only 20 rats.

Fitting to the original data, our new posterior mean estimate is $(\hat{a} = 0.6, \hat{b} = 6.4)$. As expected, the inferences are pulled toward zero compared to the posterior estimate of $(1.2, 10.5)$ from the model fit with default uniform prior). This is an example of regularization, as discussed in Section 1.4.

It makes sense that the inference for the slope b has shifted so much, but it might be surprising that the inference for a changed so much, given the weakness of the prior. It turns out that the change in the inference for a is almost entirely due to a combination of two factors: (1) the prior for b, and (2) posterior correlation between a and b. Supplying information about b has indirectly constrained a.

We can see this by re-fitting the model with an absurdly weak normal$(0, 5000)$ prior for a. In this case, the details of the weakly informative prior for a did not matter and the posterior mean was still $(\hat{a} = 0.6, \hat{b} = 6.4)$. The prior for b was relevant because the data in this case supply so little information about the slope.

Among other things, this is an illustration of how we can perform mini-experiments to understand the fitting process. We had no interest in the normal$(0, 5000)$ prior for its own sake; it was just a device we used to confirm our reasoning about the effect of the original prior on the joint inference for (a, b). And once we have a reasonable prior for these parameters, we could also check our computations using simulation-based calibration, as discussed in Section 1.10.

2.5. Inference for a quantity of interest

We can easily obtain posterior draws from derived quantities. In the bioassay analysis, for regulatory purposes there is interest in the lethal dose 50% (LD50), the dose which dose has 50% probability of death. Solving $\mathrm{logit}^{-1}(\alpha + \beta x) = 0.5$, we get $x_{\mathrm{LD50}} = -\alpha/\beta$. Racine-Poon et al. (1986) mention the 1983 Swiss poison regulation, which defines LD50 based hazardousness categories for chemicals orally given to rats (mg/ml). For comparing to the category ranges, we transform LD50 to this scale.

Figure 2 shows a quantile dot plot of the posterior distribution of LD50 with the category boundaries. From the fitted model, we can confidently classify the tested toxin to Category 4. If there were much uncertainty to which category a toxin would belong, it would be possible to design a future experiment to maximize the expected information gain.

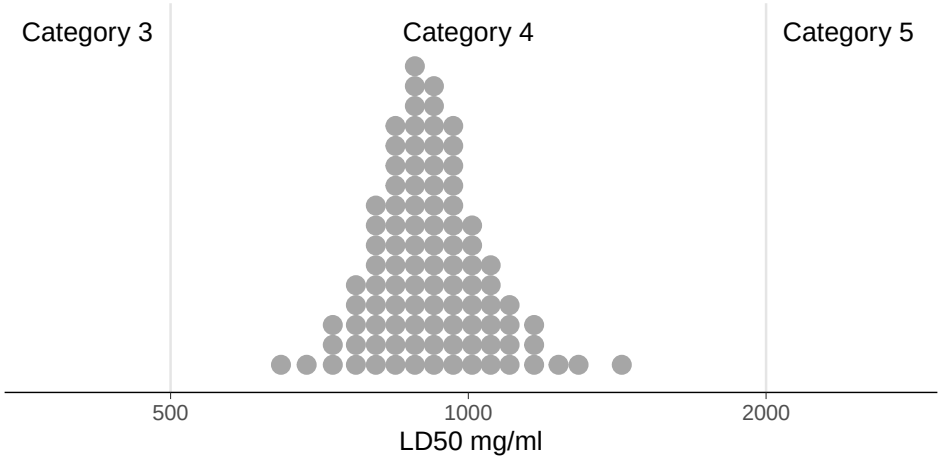

Figure 2: *Quantile dot plot of the posterior distribution of LD50 and the 1983 Swiss poison regulation category boundaries. We can be confident that the analyzed toxin belongs to Category 4 and no further experiments are needed.*

2.6. Exploring the model and the fitting process

If the goal were merely to fit this particular dataset, it could make sense to stop right here. But if we are intending to develop a method that would apply more generally, many further steps of statistical workflow could be taken to explore the model and data structure.

We could explore our understanding of the model by altering the conditions of the fake-data simulation, trying different parameter values and different values for the unmodeled data, which in this case are the number of groups, J, the vector of the number of rats in each group, n, and the vector of dose levels, x. Increasing the sample size would allow the parameters to be better identified from the data, and the fitted model should more closely estimate the assumed parameter values—this is a good way to check that the procedure is working well under ideal conditions. Decreasing the sample size would make the estimation problem harder, and it is a good way to get a sense of where the fitting procedure will break down. By changing the values of x in this logistic regression, you can change the difficulty of the problem: move the design points closer to each other and the slope becomes harder to estimate; shift them all far enough to the left or the right and the data points will all become 0's or 1's, which limits the amount of information that will be coming from the data. How to alter the simulation will depend on the particular example, but that is fine: the effort taken to work out these alternatives should be valuable for understanding the model-fitting process, that is, the map from data to inference.

Another way to understand the fit is to perturb the modeled data, y. This can be a powerful tool because then you are not restricted to what might come up in your model. For example, the data in Figure 1 are $y = (0, 1, 3, 5)$, for each dose representing the number of rats out of 5 in the experiment at that dose who died. You could change this to $y = (5, 1, 1, 5)$, which would say that all the rats died when they were given a very low dose or a very high dose, but they did well in between. This does not seem so consistent with the logistic regression model, but you could try fitting to see what happens. Or you could increase the sample sizes n from $(5, 5, 5, 5)$ to $(50, 50, 50, 50)$ and scale up y accordingly, in which case the resulting concentration of the likelihood, combined with the poor fit to the assumed family of curves, could make the model difficult to fit—an example of the "folk theorem of statistical computing," which states that computational problems often arise from modeling problems (Gelman 2008). For the data at hand there might not be a problem because the

observed proportions happen to be very close to a logistic curve (see Figure 1b), but sharpening the likelihood can effectively constrain the parameters enough that variational or simulation algorithms can have difficulty capturing or traversing the posterior distribution.

A different direction is to explicitly set up an alternative probabilistic model and simulate from it, to get a sense of the performance of the fitting procedure under model misspecification.

Finally, there are directions to expand the model, for example allowing the probability of death to vary by rat, which would imply overdispersion beyond the binomial distribution, and allowing the probabilities to depart from the logistic function. Given that the model fits the data well (see Figure 1b), there would seem to be no reason to add these complexities. A more fruitful direction could be to include additional data from other experiments and other toxins and then allow the logistic regression coefficients to vary, following the principle discussed in Section 1.3 of combining information from multiple sources. This would serve the larger goal of learning about more general conditions, not just the 20 rats and this one toxin in these data. Model and data expansion go together.

2.7. Bayesian and non-Bayesian workflows

The above Bayesian steps have clear non-Bayesian analogues. The fit can be performed with maximum likelihood estimation with uncertainties computed using the normal approximation to the likelihood, and if desired the small-sample bias and coverage problems can be corrected using a simulation study. The computation can still be checked using simulation from the fitted model, and problems of unstable estimation can be fixed using a regularization which might be mathematically equivalent to a weakly informative prior but would be evaluated based on its frequency properties conditional on some range of possible values of a and b, and also possible alternative models. Finally, the inputs to any statistical procedure can be perturbed as in Section 2.6. The point is that, even in this simple example where the model is specified ahead of time, experimentation can be used to understand and explore the many steps between data, model, and conclusions.

3. Discussion

There are many ways to organize statistical workflow, and our list of topics in Section 1 represents just one take on the problem. As we saw in Section 2, a model can fit to observed data but fail in similar examples, even for data that have been generated from the model being fit. It's good general practice to check a fitting procedure by applying it to simulated data and then repeating this process to catch possible problems.

No matter what statistical and machine learning methods you use, and whatever your philosophy of inference, our message is that serious workflow involves many steps that are often hidden from view. These steps include details of measurement, data collection, and data exclusion rules, along with initial values and tuning parameters for computational algorithms. Visualizations of data and fitted models can and should be integrated into the process of data analysis rather than relegated to an exploratory phase before the analysis begins and presentation graphics at the end. Explicit accounting for inferential goals can guide both modeling and model checking, and any statistical procedure can be understood by examining its properties when it is fit to some distribution of potential datasets.

We find it fruitful to consider the process of data analysis as a form of scientific or engineering exploration, not just of the data but of the models and procedures being used to analyze the data. The two key ways we do this exploration are by including additional information (whether labeled

as "new data" or "priors")—hence the importance of tools for combining information from multiple sources as discussed in Section 1.3—and by evaluating procedures using simulated data.

References

Berk, R., L. Brown, A. Buja, K. Zhang, and L. Zhao (2013). "Valid post-selection inference." *Annals of Statistics* 41, pp. 802–837.

Box, G. E. P. (1980). "Sampling and Bayes inference in scientific modelling and robustness (with discussion)." *Journal of the Royal Statistical Society A* 143, pp. 383–430.

Cai, T., P. Greengard, G. Goodrich, and A. Gelman (2024). "Approximate posterior recalibration." `https://sites.stat.columbia.edu/gelman/research/unpublished/Approximate_posterior_calibration.pdf`.

Chipman, H. A., E. D. Kolaczyk, and R. E. McCulloch (1997). "Adaptive Bayesian wavelet shrinkage." *Journal of the American Statistical Association* 92, pp. 1413–1421.

Cook, S., A. Gelman, and D. B. Rubin (2006). "Validation of software for Bayesian models using posterior quantiles." *Journal of Computational and Graphical Statistics* 15, pp. 675–692.

DiCiccio, T. J. and B. Efron (1996). "Bootstrap confidence intervals (with discussion)." *Statistical Science* 11, pp. 189–228.

Donoho, D. L. and I. M. Johnstone (1994). "Ideal spatial adaptation by wavelet shrinkage." *Biometrika* 81, pp. 425–455.

Firth, D. (1993). "Bias reduction of maximum likelihood estimates." *Biometrika* 80, pp. 27–38.

Gal, Y. and Z. Ghahramani (2016). "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning." *Proceedings of Machine Learning Research* 48, pp. 1050–1059.

Gelman, A. (2003). "A Bayesian formulation of exploratory data analysis and goodness-of-fit testing." *International Statistical Review* 71, pp. 369–382.

Gelman, A. (2004). "Exploratory data analysis for complex models (with discussion)." *Journal of Computational and Graphical Statistics* 13, pp. 755–779.

Gelman, A. (2008). "The folk theorem of statistical computing." *Statistical Modeling, Causal Inference, and Social Science*. 13 May. `https://statmodeling.stat.columbia.edu/2008/05/13/the_folk_theore/`.

Gelman, A. (2018). "Bayesians are frequentists." *Statistical Modeling, Causal Inference, and Social Science*. 17 Jun. `https://statmodeling.stat.columbia.edu/2018/06/17/bayesians-are-frequentists/`.

Gelman, A. (2024). "Before data analysis: Additional recommendations for designing experiments to learn about the world." *Journal of Consumer Psychology* 34, pp. 190–191.

Gelman, A. and J. Auerbach (2016). "Age-aggregation bias in mortality trends." *Proceedings of the National Academy of Sciences* 113, E816–E817.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2013). *Bayesian Data Analysis*. 3rd ed. London: CRC Press.

Gelman, A., J. L. Hill, and M. Yajima (2012). "Why we (usually) don't have to worry about multiple comparisons." *Journal of Research on Educational Effectiveness* 5, pp. 189–211.

Gelman, A., A. Jakulin, M. G. Pittau, and Y. S. Su (2008). "A weakly informative default prior distribution for logistic and other regression models." *Annals of Applied Statistics* 2, pp. 1360–1383.

Gelman, A., A. Vehtari, D. Simpson, C. C. Margossian, B. Carpenter, Y. Yao, P. C. Bürkner, L. Kennedy, J. Gabry, and M. Modrák (2020). "Bayesian workflow." `https://sites.stat.columbia.edu/gelman/research/unpublished/Bayesian_Workflow_article.pdf`.

Hullman, J. and A. Gelman (2021). "Designing for interactive exploratory data analysis requires theories of graphical inference (with discussion)." *Harvard Data Science Review* 3.3.

Krinsky, I. and A. L. Robb (1991). "Three methods for calculating the statistical properties of elasticities: A comparison." *Empirical Economics* 16, pp. 199–209.

Modrák, M., A. H. Moon, S. Kim, P. C. Bürkner, N. Huurre, K. Faltejsková, A. Gelman, and A. Vehtari (2025). "Simulation-based calibration checking for Bayesian computation: The choice of test quantities shapes sensitivity." *Bayesian Analysis* 20, pp. 461–488.

Pinheiro, J. C. and D. M. Bates (2000). *Mixed-Effects Models in S and S-Plus*. New York: Springer.

Racine-Poon, A., A. P. Grieve, H. Fluhler, and A. F. M. Smith (1986). "Bayesian methods in practice: Experiences in the pharmaceutical industry (with discussion)." *Applied Statistics* 35, pp. 93–150.

Rubin, D. B. (1984). "Bayesianly justifiable and relevant frequency calculations for the applied statistician." *Annals of Statistics* 12, pp. 1151–1172.

Rue, H., S. Martino, and N. Chopin (2009). "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations (with discussion)." *Journal of the Royal Statistical Society B* 71, pp. 319–382.

Savitsky, T. D. and J. Gershunskaya (2024). "Simulation-based calibration of uncertainty intervals under approximate Bayesian estimation." `https://arxiv.org/abs/2407.04659`.

Smullyan, R. (1979). *The Chess Mysteries of Sherlock Holmes*. New York: Knopf.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). "Dropout: A simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research* 15, pp. 1929–1958.

Taylor, J. and R. J. Tibshirani (2015). "Statistical learning and selective inference." *Proceedings of the National Academy of Sciences* 112, pp. 7629–7634.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Reading, Mass.: Addison-Wesley.

Weber, S., A. Gelman, D. Lee, M. Betancourt, A. Vehtari, and A. Racine-Poon (2018). "Bayesian aggregation of average data: An application in drug development." *Annals of Applied Statistics* 12, pp. 1583–1604.

Yao, Y., A. Vehtari, D. Simpson, and A. Gelman (2018). "Using stacking to average Bayesian predictive distributions (with discussion)." *Bayesian Analysis* 13, pp. 917–1003.

Yu, B. and R. L. Barter (2024). *Veridical Data Science: The Practice of Responsible Data Analysis and Decision Making*. Cambridge, Mass.: MIT Press.

Yu, B. and K. Kumbier (2020). "Veridical data science." *Proceedings of the National Academy of Sciences* 117, pp. 3920–3929.