Toward a Taxonomy of Trust for Probabilistic Machine Learning

Tamara Broderick[a*], Andrew Gelman[b,c], Rachael Meager[d], Anna L. Smith[e], and Tian Zheng[b]

[a]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology;
[b]Department of Statistics, Columbia University;
[c]Department of Political Science, Columbia University;
[d]Department of Economics, London School of Economics and Political Science;
[e]Department of Statistics, University of Kentucky.
[*]To whom correspondence should be addressed; E-mail: tamarab@mit.edu.

22 Oct 2022

To appear in *Science Advances*.

Abstract

Probabilistic machine learning increasingly informs critical decisions in medicine, economics, politics, and beyond. To aid the development of trust in these decisions, we develop a taxonomy delineating where trust in an analysis can break down: (1) in the translation of real-world goals to goals on a particular set of training data, (2) in the translation of abstract goals on the training data to a concrete mathematical problem, (3) in the use of an algorithm to solve the stated mathematical problem, and (4) in the use of a particular code implementation of the chosen algorithm. We detail how trust can fail at each step and illustrate our taxonomy with two case studies. Finally, we describe a wide variety of methods that can be used to increase trust at each step of our taxonomy. The use of our taxonomy highlights steps where existing research work on trust tends to concentrate and also steps where building trust is particularly challenging.

"Science, at its core, is a social phenomenon. It is a reflection of people, of our relationships, and of our institutions. When we provide inputs to the algorithm, when we program the device, when we design, test, and research, we are making human choices—choices that bring our social world to bear in a new and powerful way." — Alondra Nelson, Deputy Director for Science and Society, White House Office of Science and Technology Policy, 2021.

1. Introduction

Machine learning (ML) in general, and probabilistic methods in particular, are increasingly used to make major decisions in science, the social sciences, and engineering—with the potential to profoundly impact individuals' day-to-day lives. For instance, probabilistic methods have driven knowledge of the spread and effects of SARS-CoV-2 (*1–3*), underlie election predictions at the *Economist* (*4, 5*), and can guide our understanding of the efficacy of microloans in alleviating poverty (*6*). Given the large and growing impact of probabilistic ML, it behooves us to make sure that its outputs are useful for its users' stated purposes.

There are many potential points, though, where a data analysis pipeline may break down. This issue becomes especially pressing as statistics and ML workflows grow increasingly complex to face modern challenges. These challenges arise not just from the sheer size of the data but also from the inherent difficulty of the problems being studied. Big Data are messy data: confounded data rather than random samples, observational data rather than experiments, available data rather than direct measurements of underlying constructs of interest. To make relevant inferences from big data, we need to extrapolate from sample to population, from control to treatment group, and from measurements to latent variables. All of these steps require modeling. And Big Data needs Big Model: latent-variable models for psychological states or political ideologies, differential equation models in pharmacology, dynamic image analysis, and so forth. To meet modern challenges, we often fit models that are on the edge of our ability to compute and interact with in the real world. These models typically contain many assumptions and decision points. And we are

often pushed to adjust for more factors to capture the complexity of our world. But models that adjust for lots of factors are hard to estimate. Fitting big models to big data requires scalability, so we must often turn to approximations such as Markov chain Monte Carlo, variational inference, and expectation propagation.

As this pipeline becomes more elaborate, there are more potential points of failure. But other complex endeavors succeed thanks in large part to extensive infrastructure: e.g., software engineering has testing and construction has scaffolding. It is similarly necessary to build an infrastructure to support the trustworthy creation and deployment of methods in probabilistic ML. To this end, there exists a large literature addressing concerns of trust in data science—with key words including reproducibility, replicability, theoretical guarantees, stability, and interpretability.

To a new reader, it may not be obvious how these concerns relate to each other and what new work is needed in this area. In the present paper, we develop a *taxonomy of trust*, splitting the probabilistic ML workflow (or data analysis pipeline, or inferential chain) into distinct constituent parts where trust can fail. Just as testing in software engineering benefits from modularity, we hope that the modularity of our taxonomy can facilitate more targeted work on trust concerns. We will use our taxonomy to highlight concerns that are relatively well-studied and those that could benefit from more attention.

We start by delineating our focus as it relates to probabilistic machine learning and trust in Section 2. Then we lay out our taxonomy in Section 3 and illustrate its parts with two case studies. We finally use the taxonomy to organize and discuss different approaches to growing trust in a data analysis—including existing work on reproducibility and replicability (Section 4).

2. Our Focus

We here focus on probabilistic ML methods. Following the textbook *Machine Learning: a Probabilistic Perspective*, "we define machine learning as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty" ((*7*)). Further in accordance with (*7, 8*), we describe an ML method as probabilistic when it uses the tools of probability theory to handle uncertainty in the decision-making process, as in our examples.

In what follows, we concentrate on trust in the quality of decisions that are made using the results of a data analysis. We focus on the extent to which informed experts can trust these decisions; these experts could include the original data analysts—as well as peer data analysts in academia, industry, journalism, other forms of public service, or an independent capacity. Further stakeholders typically rely on experts to vouch for the products of a data analysis, so our trust focus remains relevant to these consumers as well.

Trust of lay consumers in a data analysis, though, may also rely on other factors. For instance, the broader sociopolitical climate can influence a broader sense (or lack) of trust in the scientific community and its outputs, but a discussion of these relationships is outside the scope of the present work. We point the interested reader to resources on human trust in AI (*9, 10*), trust in computing (*11*), and trust more generally (*12*). Likewise, concerns about privacy and security are important and related to trust in the use of data and its artifacts (*13, 14*) but are outside our purview.

3. Where trust can break down

We next turn to understanding where trust in the decisions made from a probabilistic ML analysis can fracture. Figure 1 gives a visualization of our *taxonomy of trust*; this graphic provides a map of how probabilistic ML analyses interact with the real world. We first give a high-level overview of its steps, then describe how it shows where trust can break down in a data analysis, and finally give two case studies as examples.
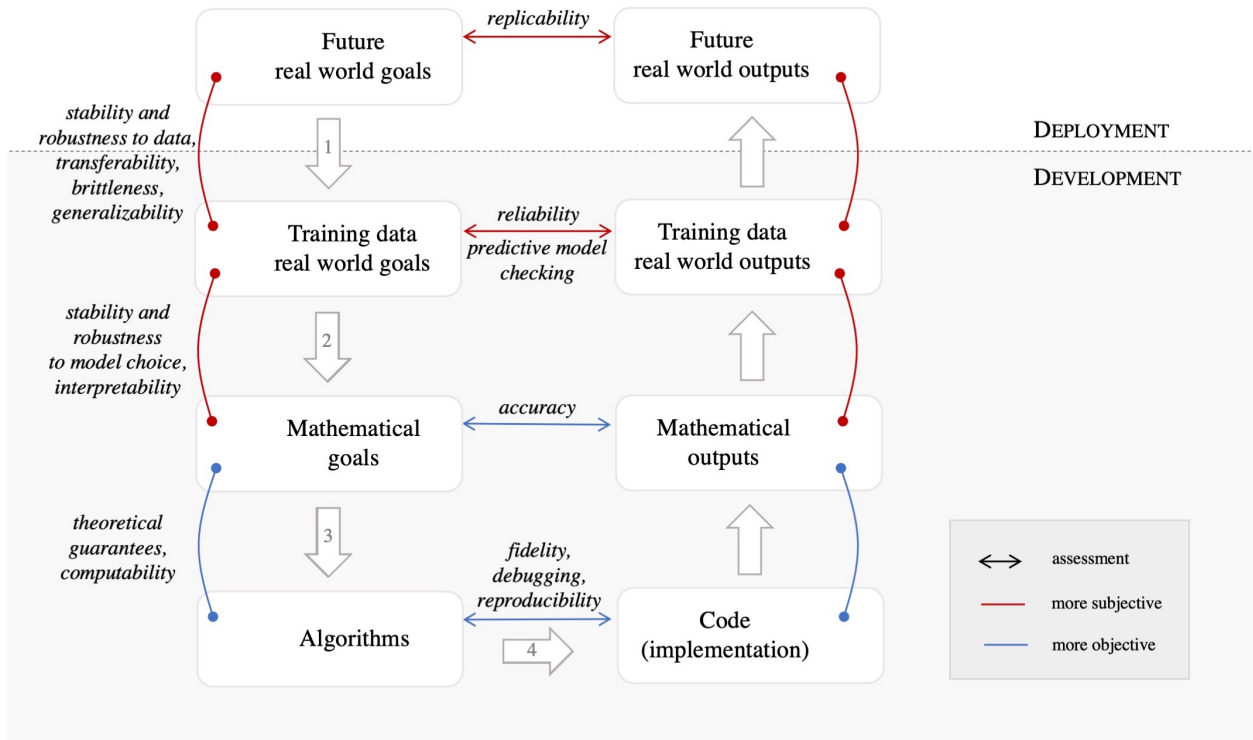
Figure 1: *Diagram illustrating steps where trust can break down in a data analysis workflow. See Section 3 for a detailed description of the (non-italicized) steps. See Section 4 for a discussion of the italicized connectors, which include many of the characteristics of Schwartz* et al. *(15) for cultivating trust in AI systems.*

We typically embark upon a data analysis because we seek to make an impact on our understanding of the world or generate future decision recommendations. From this perspective, an analysis is designed with "Future real-world goals" in mind (upper left of Figure 1). The data analyst then follows a rough workflow (on the left side of the figure) to turn this abstract goal into a concrete set of actions, which correspond to the numbered steps (arrows) in Figure 1.

1. To serve these real-world goals, data must be gathered and processed for analysis. At this point, the analyst has reduced the problem to "Training-data real-world goals."

2. The analyst chooses a model and expresses the real-world problem as a mathematical problem. The analyst has now reduced the problem to concrete "Mathematical goals."

3. The analyst chooses particular "Algorithms" to solve the mathematical problem.

4. The analyst runs the algorithm in practice using some particular "Code (implementation)."

We see the outputs of this workflow on the right side of Figure 1. The analyst runs their code and obtains a particular mathematical output. The output on the training data at hand is interpreted in the context of the real-world goal. Finally, the learned model is applied to future real-world outputs to make substantive conclusions and decisions.

Each numbered step in Figure 1 in turn represents a point where trust can break down in a data analysis.

1. Researchers typically face constraints in the cost of gathering, processing, or analyzing data, so the data set is necessarily a limited representation of the world. In choosing to approach our abstract question with a particular analysis of a particular set of data, we trust that the results from our data analysis are relevant to conclusions and decisions we will make at other places or times.

2. Essentially all models are misspecified. In choosing to express our problem with a model and mathematical formalism, we are trusting that the model and formalism adequately capture the substantive goals of the analysis.

3. Algorithms are often supported by theory, but based on assumptions that may not be perfectly achieved in practice. In choosing to solve our mathematical problem with a particular algorithm, we are trusting that the algorithm accomplishes the particular mathematical goal.

4. Code is a precise way to manifest an algorithm, but it is difficult to avoid bugs and conceptual errors—especially in large and complex code bases. In choosing to execute our algorithm via code, we are trusting that the code is faithful to the algorithm.

3.1. Case study: Microcredit efficacy

We use an example from applied economics that one of us has worked on (*6*) to demonstrate how the steps of Figure 1 relate to trust in a data analysis.

Future real-world goals. Economists and policymakers would like to know if programs promoting microcredit (small loans in developing countries) have an overall beneficial effect—and, if they do, to use that information to inform decisions on subsidizing or disbursing microcredit across many potential locations. A related question of interest to economists is whether, in a given context, expanding access to the services offered by local microlenders would be beneficial for a local population.

Step 1 ⇒ Training-data real-world goals

We analyzed data from seven different microfinance programs, each located in a different country. The countries range from Mexico to Mongolia, Bosnia to the Philippines, and the programs encompassed for-profit banks, government programs, and NGOs. Each individual dataset has been used to understand local efficacy of expanding access to microloans. In (*6*), we combined data across different settings and even somewhat different loan products to assess the geographically broader question of expanding microcredit in new locations.

In each of the individual studies, microcredit access was assigned in a randomized controlled trial (RCT) in order to prevent selection bias on the part of the microlender or borrowers from confounding the estimate of the effect of microcredit.

Moreover, researchers in each case decide what "beneficial" means, then what variables to measure to capture it, and further how to measure these variables using surveys or other tools. In the microcredit RCTs, the researchers all considered small-business profits as an outcome, but only five of the seven studies considered household consumption, and even fewer considered other components of well-being such as community health. All of these variables were carefully conceptualized and measured; e.g., based on contextual knowledge and previous research, someone decides whether, in rural Mexico, goats count as consumption or as investment. Some of the studies also recorded contextual covariates or characteristics of households to understand how they affect microcredit efficacy, but the relative importance of different factors in driving household outcomes (and thus likely moderating the effects of credit) such as consumption or business profits is itself an open research question.

Trust. Trust could break down in this step if the particular data analysis proved not to be useful for making future decisions. A researcher almost always makes choices that trade off the practicality and feasibility of collecting data and running the analysis against accurately capturing the state of the world. On the data collection side, there are an unending set of outcomes and characteristics of homes or small businesses that researchers could collect, but for time and funding reasons, the list must usually be limited.

The preference for randomized trials of microcredit reflects a desire to estimate unconfounded treatment effects even without being able to do any covariate adjustment, which partially addresses this concern, but might at times result in greater noise, or less representative sample sizes and thus greater extrapolation error.

Finally, the use of multiple studies provides hope that the results will generalize more than any single study, but full generalization relies on the assumption that the studies we have cover relevant aspects of a broader population of contexts in which we might make policy. And even if this assumption is satisfied, all studies occur in the past, but the policy decision gets made in the future; it is necessary to trust that the inference remains applicable under the new circumstances, or to make assumptions allowing appropriate adjustments to make this extrapolation.

Step 2 ⇒ Mathematical goals.

The original RCT papers and the later meta-analysis all required a statistical model and choice of inference procedure. Economists often choose linear models for their interpretability. We chose a hierarchical linear model to allow partial pooling across the different studies, reflecting both their shared information and idiosyncrasies. Moreover, we took a Bayesian approach and provided a prior as part of the model.

We reported Bayesian posterior summaries such as posterior means and variances. Given a hierarchical model and non-conjugate priors, these summaries must be approximated. So the mathematical goal is to report accurate approximations of these quantities for the particular chosen model.

Trust. Trust could break down in this step since we know that models are essentially always misspecified in practice. Linear models of conditional mean dependence are highly interpretable but cannot capture nonlinear trends that may exist (and can be captured by other models). Even if the policy intervention is represented by a binary variable, linearity in raw household outcomes is incompatible with linearity in log space, and the distinction matters for extrapolating to new settings. Focusing on the mean itself is an influential choice: the mean effect of microcredit may be positive even if just a small portion of people actually benefit and even if some are harmed; in contrast, what we might really wish to understand is whether many people benefit from microcredit, or the community as a whole experiences net benefit. Moreover, reporting a posterior mean and variance alone, without additional visualizations, can hide posterior multimodality, heavy tails, or other distributional features that might cause us to pause and dig deeper into an analysis.

Step 3 ⇒ Algorithms

We chose to approximate the Bayesian posterior mean and variance summaries using Hamiltonian Monte Carlo (HMC) (*16*). In related work (*17*), we have considered variational Bayesian (VB) approximations of these posterior quantities instead; in the latter case, we employed mean-field VB approximations (*18*) as well as linear response corrections (*19, 20*).

Trust. Trust could break down in this step if the algorithm did not in fact accomplish the mathematical goal. For instance, MCMC estimates of posterior expectations will converge to their exact values when an algorithm is run for enough time. But, despite a large literature on mixing rates, there are typically no concrete, computable bounds on performance after running a particular MCMC algorithm for a particular finite time. Analogously, VB will return a match to the exact posterior when used with a sufficiently large family

of distributions and enough computation time. But in practice, the family of approximating distributions is typically limited, and the Kullback-Leibler (KL) divergence between the approximating and true posterior will be strictly greater than zero. Even moderate KL divergence values can correspond to arbitrarily large discrepancies between exact and reported posterior means and variances (*21*). While there is work on using more expressive families of distributions in VB, these face practical challenges and lack computable guarantees on quality.

Step 4 ⇒ Code

As increasingly complex methods are used for data analysis, a practitioner will often turn to existing software packages. Here we used Stan for HMC (*22*). (*17*) developed new code for VB.

Trust. Trust could break down in this step if there are bugs in the code. There is robust software-engineering guidance on how to test code, including unit testing. Data analysts often rely on a software package for implementation and thus, to some extent, often outsource much (but certainly not all) of the establishment of trust in this step. Fitting the model in Stan, a well-tested probabilistic programming language, gave us confidence in the inferential part of our computation. But there is still the entire data and analysis pipeline to consider; recall the infamous Excel error of (*23*); see (*24*) and (*25*).

3.2. Case study: Election forecasting

We similarly walk through the parts of our taxonomy, and how they relate to trust, with an example from our development of a public poll-aggregation algorithm for election forecasting.

Future real-world goals. In spring 2020 one of us collaborated with a political science graduate student and a data journalist at the *Economist* magazine to produce an ongoing state-by-state and national-level forecast of the upcoming U.S. presidential election (*4, 5*). The forecast automatically updated as new polls arose during the campaign.

Step 1 ⇒ Training-data real-world goals.

A particular challenge of this problem is incorporating diverse sources of data: national polls, state polls, economic statistics, and past state and national election results—as well as knowledge reflecting political science understanding of day-to-day changes in voting behavior and survey responses.

Trust. By analogy to the microcredit example, trust could break down in this step if the data used in the model were deemed incomplete in the sense that there are other available data that could be used to noticeably improve the forecast. For example, a forecast based entirely on national polls would have problems staying up to date with the latest state poll results. Particularly relevant to the 2016 and 2020 elections were concerns about poll bias due to nonresponse; for instance, we might be concerned that fewer Republicans responded to polls, especially in key swing states.
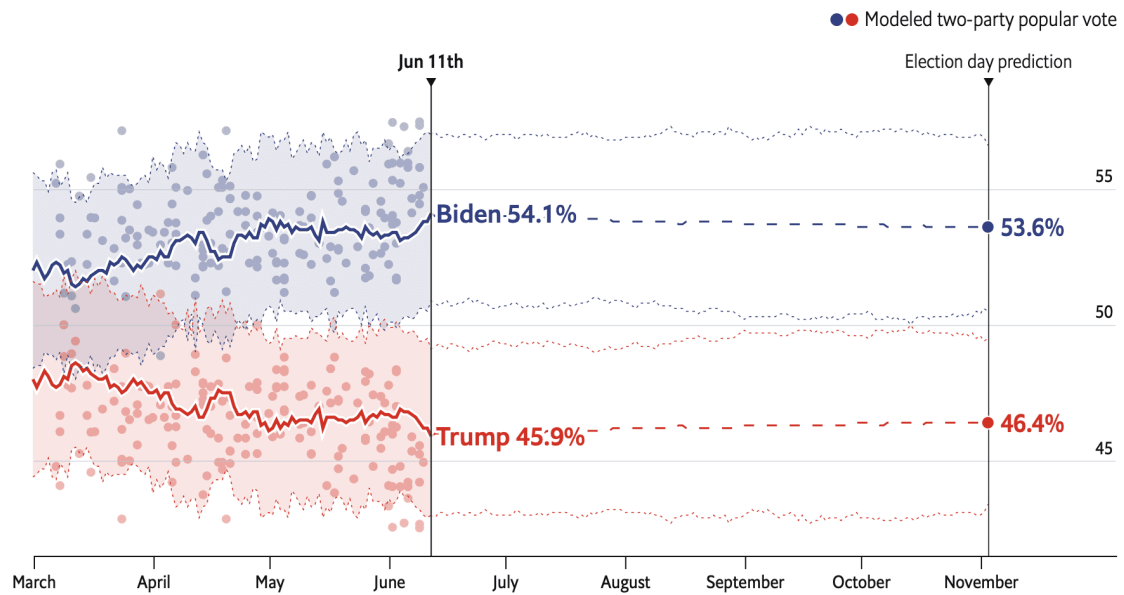
Step 2 ⇒ Mathematical goals

Constructing the model was more difficult than you might think. The probability distributions representing the underlying public opinion and survey data had to serve as a bridge connecting poll and economic data, political science understanding, and statistical structure (sampling error, nonsampling error, and variation of opinion over time in the 50 states).

Modeled popular vote on each day

The model first averages the polls, weighting them by their sample sizes and correcting their partisan biases. It then combines this average with our forecast based on non-polling data, pulling vote shares on each day slightly towards the final election-day projection.

●● Modeled two-party popular vote

Jun 11th · Election day prediction

Biden 54.1% — — — — — — — — — — — — — — — — — — — ● 53.6%

55

50

Trump 45.9% — — — — — — — — — — — — — — — ● 46.4%

45

March   April   May   June   July   August   September   October   November

Sources: US Census Bureau; MIT Election and Data Science Lab; 2016 Cooperative Congressional Election Study; US Bureau of Economic Analysis; American National Election Studies; 270towin.com; Gallup; FiveThirtyEight; YouGov

Forecast by *The Economist* with Andrew Gelman and Merlin Heidemanns, Columbia University

Figure 2: *Our election forecast for the* Economist *on the day of its release in June 2020. Before taking this forecast public, we went through a series of checks of the data, model, and fitting procedure. We made further changes to the model as the campaign went on.*

Concretely, we started with an existing forecasting model in political science (*26*), which in turn was based on earlier models for poll aggregation and election forecasting. The unique feature of our approach compared to other forecasts for the 2020 election was fitting a single model combining information from national polls, state polls, and economic and political fundamentals based on previous election outcomes.

Challenges in setting up this model included the following. We wanted to account for nonsampling error at both the state and national levels, including possible systematic biases favoring one party or the other; our error terms did not correct for biases, but they should allow the model to express forecasts with appropriate uncertainty. We aim to capture a time series of opinion changes during the campaign. We wanted to interpret the historical predictive power of economic fundamentals in the context of increasing political polarization. And we aimed to acknowledge unique features of the 2020 campaign. A particular statistical challenge arose in modeling the correlations of polling errors and opinion trends across states: there was nothing like the amount of data available to estimate a 50×50 covariance matrix from any purely empirical procedure, so we knew that strong assumptions would be necessary.

Trust. Trust could break down in this step if any of these strong assumptions or modeling choices could be replaced with other reasonable choices and lead us to substantively different conclusions. Indeed, some of our initial modeling choices regarding between-state correlations were flawed, as we noticed after seeing some predictive intervals that seemed implausible. Unfortunately, we only noticed some of these errors after the forecast went live, and we were forced to reboot our model and publicly explain our changes.

Step 3 ⇒ Algorithms

As with the microcredit case study, we used HMC to approximate and summarize the Bayesian posterior distribution on election outcomes and other posterior quantities of interest for political understanding. The dataset and model were small enough and our time scale was gentle enough (requiring updates every day, not every hour or minute) that there was no need to use a fast shortcut algorithm.

Trust. As discussed in the previous example, trust could break down if the approximation quality were poor.

Step 4 ⇒ Code

We used R and Stan for HMC; we also compiled and cleaned survey and election data.

Trust. Our general experience with Stan allowed us to trust its computation, so that the main concerns with the code were in the R scripts we wrote to prepare the data, set up covariance matrices, and postprocess the inferences.

4. Building trust

Above we have seen a number of ways that trust can break down across the taxonomy. But crucially there are also many ways for researchers to build trust. We next place existing work into our taxonomy. We start from the bottom of Figure 1 and work our way up. In particular, the italicized vertical connectors in Figure 1 give terms related to trust for the corresponding step. The horizontal connectors show terms that describe matching between goals and outputs at this level.

While we hope to highlight ways that analysts and researchers can grow trust, we do not intend to suggest the existence of a perfect analysis or a foolproof solution—just as we might improve software engineering through testing, but we do not expect even the most careful testing to entirely eliminate bugs. Likewise, we

try to distinguish between tools that would require an unusually large monetary cost, time budget, compute access, or other resources—and tools that are available to most analysts.

Algorithms and code

Fidelity. Trusting code to faithfully represent an algorithm (*fidelity*) is sometimes considered prosaic and perhaps even outside the purview of the data scientist, especially when using standard software packages. Yet this step forms the bedrock of the analysis; if trust breaks down here, trust in the entire analysis will fail. And indeed bugs in data analyses have led to unsupported justification for treatment of patients in clinical trials, as discussed by (*27*), and influential but unsupported economic policy advice, as discussed by (*24*) and (*25*).

Reproducibility. The name of the core issue here varies widely across the literature. In the present paper, we follow the usage of Conclusion 3-1 of the National Academies report on "Reproducibility and Replicability in Science" (*28*), labeling an analysis as *reproducible* if identical results are achieved when the same data are analyzed again with the same code. Despite this definition seeming somewhat limited, many data analyses (including some of our own in busy research projects!) do not meet this standard. For instance, some data analyses do not provide code. Some do not even describe their algorithm in enough detail for the reader to produce equivalent code (*29*). Moreover, unequal computing resources can form a barrier to actually checking reproducibility in practice; some modern analyses require resources enjoyed only by a handful of large companies (*30*). Even with full descriptions and code and adequate resources, checking reproducibility can be challenging and time-consuming. For instance, the rapid advances of the ML field give rise to continually changing software packages and dependencies; even when the precise set of dependencies used by a project is known, setting up the same ecosystem can be burdensome. And probabilistic methods often take advantage of randomness; without access to the random seed, two separate runs of the same code in the same environment may yield different results.

Supporting reproducibility. Nonetheless, these observations suggest clear ways to build trust at this step, via open community engagement, code sharing, and data sharing. Even when using standard software packages, analysts typically write their own code wrappers to call these packages for the specific data and model at hand, and this new code needs to be checked. (*31*) propose a gold standard where data analysts set up a single command that exactly reproduces the full analysis—and a silver standard that requires easy access to package dependencies and any random seeds. Modern computational tools that manage models (*32*) and aid reproducibility (*33*) can help researchers achieve the gold standard. It might be easy to overlook code that goes into data pre-processing, but this code is part of an analysis pipeline too; for instance, NASA failed to detect ozone reduction for much of the 1970s and 1980s (despite having appropriate sensors) due to an unexpected side effect of data pre-processing (*34*). (*35*) propose "datasheets for datasets" to fully detail data provenance, including any processing that may have gone into their creation. Permissive open licensing at every step of the pipeline can encourage reproducibility as well as proper attribution (*36, 37*).

Case study. During the election forecasting case study above, we opened up the process and results to a wide audience. Publication and daily updates in the *Economist* meant that thousands of readers would see each forecast. We made data and code publicly available and conveniently accessible (in this case, on GitHub). This ready access allowed outsiders to download the code, run our analysis themselves, and explore places where something could be going wrong. Outside readers did indeed discover problems, which we were able to track down to bugs and conceptual problems with our model. And the sense of public responsibility motivated us to check carefully when forecasts did not look right.

Mathematical goals

Theoretical guarantees.   At this level, we want to understand whether our algorithm achieves our stated mathematical goals. Often research will abstract the algorithm away from the code, for example by establishing *theoretical guarantees* on how well a particular algorithm can be expected to achieve a goal such as accuracy or compute time. These guarantees can help build trust. But they typically cannot be entirely relied upon to ensure accuracy: the exact set of assumptions needed for the guarantee may not hold in practice, or may not be possible to verify exactly in practice. A large amount of research concentrates at this level. Indeed, it is standard for any paper addressing Bayesian posterior approximation to come supported by theory. We likely see this abundance of research since the question of whether an algorithm achieves a precise mathematical goal has a relatively objective answer.

Checking the algorithm via the code.   Practical checks at this level can examine the actual instantiation of the algorithm in code—and either check the performance of the algorithm-code pair on representative problems, or provide post hoc checks on the results of the algorithm-code pair when run on the dataset at hand. For instance, as we have seen in the case studies, a sufficiently complex Bayesian analysis typically aims to approximate summaries of the posterior. (*38*), (*39*), and (*40*) present simulation-based methods for checking the *accuracy* of a Bayesian approximation algorithm (via its implemented code). We and others have also developed methods for checking approximate computation on a particular dataset of interest; for example, (*41*) and (*21*) have presented methods for evaluating variational inference. These new approaches can help to identify mismatch between the mathematical goals and the joint implementation of the algorithm and code.

Case study.   In the election forecasting case study, we simulated fake polling data and then saw how well the fitting procedure could recover the assumed parameters and underlying time series. The check proceeded as follows: (1) set hyperparameters of the chosen model to fixed, reasonable values; (2) simulate time series of national and state public opinion using the generative time series including the between-state correlation matrix assumed under the model; (3) simulate national and state polls at the same sample sizes and dates (in days before the election) as in the 2016 campaign, but with data set to the simulated underlying time series plus error, including both sampling and nonsampling error; and (4) fit the model to these simulated data. It is no surprise that we should be able to roughly recover the hyperparameters and time series, but given the complexity of the model it is useful to check and also to get a sense of the precision of the recovery.

Training data and real-world goals

To build trust at this level, we want to ensure that our mathematical goals are meeting our real-world data analysis goals, at least on our training data. Building trust here presents a new challenge relative to previous levels; in particular, we must now ask what it means to perform well on real-world goals. The answer to this question is necessarily more open to multiple perspectives and context-dependent than the mathematical problem an analyst reduces this question to in Step 2. Indeed, analysts often reduce the more abstract and ill-defined real-world problem to a standard mathematical problem both to apply familiar data-analysis tools but also to avoid claims that the analyst is lacking objectivity; see Table 1 of (*42*) and the surrounding text for a discussion of how multiple perspectives and context dependence relate to descriptions of subjectivity and objectivity in an analysis. We emphasize the distinction in character between this level and previous levels with the color of connectors in Figure 1.

The challenge of this level need not be due to communication difficulties between a data analyst and a domain expert. In both of our cases studies, a single person is both a data analyst and a domain expert. Yet there remain fundamental challenges.

Proxy measures to assess performance. We often choose proxy measures to assess performance of a method, so we might wonder how *stable* our results are to these choices. We may choose proxies when questions in the social sciences, sciences, and engineering are difficult to operationalize directly—or when these questions might require substantial new modeling and inference development. For example, we might use a mean effect to understand the efficacy of microcredit since a mean is straightforward and standard to capture via a linear model—even though it might be more appropriate to judge the effect of microcredit by the proportion of people to whom it is beneficial. As another example, many methods default to prediction with simple losses when the real-life goal or loss is difficult to quantify. For instance, 0-1 loss is a common choice of convenience in classification problems, but real-life classification loss is typically both unbalanced and difficult to quantify precisely. Prediction is generally an easy (statistical or mathematical) problem precisely because it is defined in terms of a clear ground truth that we can compare our predictions to. But even problems that are naturally framed as prediction, like medical diagnosis, may be mistranslated into a concrete optimization problem; e.g., consider a model for detecting cancerous skin legions that inadvertently trains on surgical ink markings (*43*).

These proxy measures of performance form part of a larger phenomenon in scientific research: hard questions often get translated into questions that are easier to answer but possibly substantively further from the question we want to ask. At its core, this critique is related to the question of how problem formulation can impact scientific outcomes (*44*). More broadly, it is important to consider whether we care about prediction or explanation (*45*): a particular analyst's real-world goals might be best addressed by optimizing a black box that makes highly accurate predictions, or by constructing a more *interpretable* model with strong explanatory power, or perhaps by doing something in between.

Proxy measures of performance can have *fairness* implications. For instance, an economic intervention can have positive average effects while only benefiting a small part of the population that is already well off. Or medical interventions tested only on men might expose women to unknown risks. See (*46–48*) for further discussion of fairness in machine learning, currently an active research area.

Stability to model choice. Just as we might choose proxy measures in part out of convenience, we might choose particular models to use in part out of convenience. In this case, we might ask about the *stability* of our decisions *under different reasonable modeling choices*. (*49, 50*) has advocated for the importance of stability of many forms, including stability under model choice, in data analysis.

Different representations or summaries of our data could also be seen as part of the choice of model, and we are interested in stability across reasonable choices of summary. As an example, consider choosing among competing network embeddings; (*51*) examine this problem within the framework of (*52*)'s principles for veridical data science: predicability, computability, and stability (PCS). For some problems, we may be interested in embeddings that preserve particular features of the observed network or in embeddings that translate to better performance for some downstream task (e.g., link prediction), rather than in estimating a true underlying embedding—or even mere features of that embedding, such as the manifold in which those embeddings lie, as in (*53, 54*).

Testing stability to model choice. To test stability to these various choices in practice, (*55*) documents variations among twenty-nine distinct research teams' analyses of a shared real-world goal, equipped with shared training data: whether soccer referees are more likely to give red cards to players with dark skin tones. Variations in adopted analytic strategies ranged in breadth from differences in which predictors are included, to how dependent observations are (or are not) adjusted for, to statistical modeling assumptions (e.g., linear or logistic regression).

Lower-cost tests of stability or robustness to model choice.   Convening multiple research teams for a problem is not always feasible, at least not at first, due to budget and time costs. So we might wish to have lower-cost checks usable by a single team. For instance, in (*56*), we construct prediction scores as a measure of agreement between data-generating mechanisms (for example, comparing preregistration or pilot data to realized experimental data); we show that modifications to the predictive model (used to construct the predictions) adjust the lens through which the prediction scores can pick up on varying types of differences between the underlying data generating mechanisms. This check goes beyond the mere reproduction of numerical outputs, but more broadly accounts for how the real world interacts with the inferential chain: how the real world goals are interpreted and formulated as mathematical goals.

Another body of literature looks at quantifying *robustness* of the conclusions of a data analysis *to likelihood and prior choice* in a Bayesian analysis (*57*), with tools to interface with MCMC (*58*) and variational methods (*20, 59*). A tricky aspect of operationalizing this work is deciding how to mathematically capture a range of reasonable models; the easiest option is to vary a hyperparameter of a model (that may control the model parametrically or nonparametrically) over some range that the analyst deems reasonable. Other options include visualizations to aid analysts and domain experts in making appropriate robustness checks (*60*).

Even the lower-cost tests, though, typically require additional thought and computation beyond the initial analysis. And even the most inexpensive and automated robustness checks are often not yet part of standard data-analysis software or generally an expected part of a data analysis. Perhaps due both to the challenge of subjectivity at this level as well as the additional effort required, we often do not see checks at this level in individual data analyses.

Future real-world goals

At this level, we want to build trust that the results from our particular data analysis are relevant to conclusions and decisions we will make at other places or times.

Replicability; robustness to data variation.   Again, building on Conclusion 3-1 in (*28*), we say that an analysis is *replicable* if similar results are achieved when the study is repeated with fresh data. A replicable study does not strictly require reproducibility (or matching of goals and outputs at other levels of Figure 1). But it is hard to imagine how a non-reproducible analysis could be successfully replicated.

We can still speak of *stability and robustness*, but now *to changing data* rather than changing model choices. For instance, ultimately we wish to apply the conclusions of our microcredit case study to decide whether to pursue microloans in new places in the future—or the same places but at necessarily different (future) times. If we concluded that microcredit alleviated poverty based on our original data, can we continue to trust these conclusions?

Testing replicability.   As in the previous case, the best option for testing replicability is to run the experiment many times in many different conditions. (*61*) brought replicability to the forefront of the scientific discussion via an attempt to replicate one hundred high-impact psychological studies. To the extent that teams are able to run full analyses that may shed light on future use cases, (*62*) propose "model cards" to document model details, training data, evaluation data, and evaluation metrics. But running a data analysis many times under different circumstances may be costly, and doing so does not eliminate the need to make future policy decisions using past analyses. To that end, there is a need for automated tools that can assess sensitivity and robustness to data changes that might reasonably reflect changes we expect to see before applying policy.

Data resampling schemes. Classical tools such as cross validation and the bootstrap can provide a notion of sensitivity by re-running a data analysis many times. Likewise, when many predictions and a ground truth are available, researchers can check if their uncertainty forecasts are *calibrated*; that is, one can check whether the proportion of times an event was predicted agrees with the forecasted uncertainty. But these checks rely on an implicit assumption that the data-generating distribution of future data is fundamentally the same as that of the data used for the checks. This assumption is often at least somewhat inappropriate. Recall that we hope to capture and apply universal truths from a data analysis. We therefore might not apply decisions from our data analysis to a wildly different context. But we still might expect, and wish to anticipate, small but substantive changes across data sets of interest. (*63*) have found that many conclusions of AI analyses do not hold up when used in real life despite being trained and tested in standard machine learning pipelines. In our own work, we reasonably expect regional differences in the effects of microcredit—and differences across time in elections. One option is to use cross-validation variants that respect temporal or spatial structure (*64*), but even these variants presuppose that future data fits a past trend or symmetry.

Stress tests. (*63*) suggests stress-testing data analyses to explore performance in practically relevant dimensions and to attempt to identify potential inductive biases. (*65*) also suggests running methods on a variety of meaningful data sets to diagnose different types of hidden issues. A challenge is finding appropriate data sets. One option that we have developed and applied to our microcredit case study is assessing sensitivity of an analysis to dropping a small amount of data (*66*). If an analysis is driven by a small proportion of its data, we might not expect it to generalize well to other scenarios. In theory, this sensitivity can be checked directly by re-running the data analysis with every small proportion of data removed; in practice, this naive approach is astronomically costly on even small data sets. So we develop an approximation that can be checked directly: when our approximation detects sensitivity, it returns the problematic small subset of data. So the analyst can drop this small data subset and re-run the analysis at the cost of just one additional data analysis.

Generalizability vs. security. The methods above address concerns about the *generalizability* of an analysis; populations may subtly differ across space or time, and these changes may in turn affect the efficacy of, e.g., an economic or medical intervention. We can be concerned about generalizability even when no security concerns are present—e.g., even if we do not suspect a nefarious actor may change our data, as in the study of adversarial examples and methods to counter them (*67*).

Indirect checks. The checks discussed so far at this level try to assess generalizability directly. Analysts may call on other forms of evidence, though, to try to understand whether decisions may generalize well from the particular training data at hand. One form of evidence is *explainability*. If we know that predictions of cancerous skin legions are based on surgical ink markings (*68*), we might expect that the method will generalize poorly to cases where experts have not already effectively made the predictions and marked them for us. By contrast, if the method's predictions have a convincing medical explanation, we may be more likely to trust the method to generalize well to new image sources. We direct the interested reader to a wide, and actively expanding, literature on explainability in AI methods, a full accounting of which is outside the scope of the present work; see e.g. (*69–73*).

Another indirect check is to assess whether the model is able to capture salient aspects of the data that are not directly part of reported summaries. Such Bayesian *model checking* originated with (*74*) and (*75*); this research arose in an era when computations were relatively simple so the focus was on fit of model to data. In modern work, Bayesian model checking often assesses whether the joint model-algorithm-code mechanism does a good job of describing the available training data, sometimes with additional granularity (*76*).

Case study: microcredit efficacy. We applied exactly the dropping-data check described above to our analysis of microcredit in (*77*). We hoped that various features of the analysis would decrease sensitivity: hierarchical Bayesian sharing of power across datasets, regularization from priors, and a tailored likelihood meant to more carefully reflect the data-generating process. But we still found that the average effects were sensitive to dropping less than 1% of the data—but the estimated variance in treatment effects across studies was more robust (*66*).

Case study: election forecasting. The election forecasting case study used sense checks related to the ideas above to assess whether its predictions could be trusted to generalize to the 2020 election. Some months elapsed from the start of our modeling process and the *Economist*'s launch of the forecast. This lag did not come from coding (as code is straightforward in Stan) or from running the model (the dataset including all polls was small enough that even without any real efforts at optimization the code ran in minutes) but rather from the steps of building and checking the model. Once we had code that compiled and ran, we fit it to data from 2008, 2012, and 2016; these were earlier elections where we had a large supply of state and national polls so we could mimic the 2020 forecasting we planned to do. Various early versions of our model produced results that did not make sense, for example with time trends seeming to vary too much from state to state, which implied that some variance parameter in our model was too large so that these time series were insufficiently constrained. This checking was valuable for its own sake and also because, to do so, we needed to design graphs whose forms we would use when plotting models fit to the 2020 campaign data.

Before releasing our predictions to the public in early June, 2020, we fit the model to the data available up to that point and checked that the inferences and our summary of them (see Figure 2) seemed reasonable, in the sense that they were consistent with our general understanding of the campaign and election. Any model is only as good as the data it includes, so even if the fit has no statistical errors we would be concerned if it produced forecasts that were far off from our general beliefs. We also examined the state-by-state forecasts from a competing model produced by the Fivethirtyeight.com team and found some implausible predictions, which we attributed to issues in how their model handled between-state correlations (*78*).

During the summer and fall of 2020, we continued to monitor our forecast. In particular we were concerned that, months before election day, the estimated probability of Biden winning the electoral college was over 90%, which did not seem to fully capture our uncertainties. We looked carefully at our code and performed more simulations, refitting our model under alternative scenarios such as removing polls from one state at a time, and eventually we discovered some bugs in our code as well as other places where we were unsatisfied with the model, most notably in our expression of the between-state correlation matrix. After a couple weeks of testing, we released an improved model along with a correction note on the *Economist* site.

Finally, we compared to the actual election outcomes. We found that our model performed well but not perfectly. The popular and electoral vote margins fell within our 90% forecast intervals, with 48 out of 50 states predicted correctly. But the election was closer than our point forecast, with Biden performing consistently worse than predicted in almost every state. Polls were off by about 2.5 percentage points in two-party vote share across the country; our forecast did not anticipate this error, but it was included as a possibility in the correlated polling error model. Including a correlated error term allowed the forecasts to acknowledge nonsampling error but is no substitute for predicting the direction of the bias.

5. Discussion

Perfect accuracy is not a requirement for trust. Indeed, we are able to trust many of the systems we interact with in our day-to-day lives despite their uncertain predictions: weather predictions are not always correct, and yet they are still useful; many people are injured or die in car crashes, and yet we rely on cars to travel

from place to place; we do not expect doctors to be able to comprehensively restore us to our previous health, and yet we trust our doctors. Even for probability models, our thresholds for acceptable accuracy may depend on the setting or type of model. For example, much of classical statistics developed out of agricultural settings. As a result, many statistical models are well-aligned with the physical sciences and essentially act as deterministic or mechanistic process models with noise. In these models, the error is often treated like a nuisance parameter and included because we wouldn't be able to estimate the model otherwise. Other statistical models are more descriptive and are fundamentally models of variability, such as in economics or the social sciences. A third class of models is even more disconnected from observable physical processes; these discovery or exploratory models are specifically designed to capture fuzziness. Examples include clustering models, topic models, and other unsupervised learning approaches. We naturally expect different levels of accuracy for models in each of these classes and, especially for discovery or exploratory models, defining accuracy can be more difficult.

In our work we have focused on fostering trust in a Bayesian context in a way different from the traditional notion of estimating the posterior probability that a model is true. Rather, we recognize that our models are assumptions, and trust relates in part to the robustness of our results and subsequent decisions to many types of explicit and implicit assumptions. There is also a large non-Bayesian literature on evaluating statistical and machine learning methods via predictive performance *(e.g., 79–81)*. These approaches have links to Bayesian predictive inference (*82*), just as there are links between deterministic and probabilistic models (*83*). However, we have found in applied work that real-world model improvements do not always show up as clear increases in prediction accuracy (*84*); this seeming discrepancy makes sense because a key goal in many applications is to make decisions that apply outside the scope of training data.

More generally, we highlight that trust itself is not binary. We have here laid out a variety of points in an analysis where trust can fail or be increased, but even at each point trust lies on a spectrum. (*31*) note that, even at the level of code reproducibility, there lies a spectrum of trust: from not providing code or details to providing them but without easy documentation or use all the way to providing well-documented code that duplicates an existing analysis with a single command. We analogously argue in (*56*) for treating trust as a spectrum, rather than a binary, at other levels of our present taxonomy. In almost all settings, there will be some variation (due to different operationalizations of the real world goals, or to stochastic elements in the algorithm, or even to random number generation in the code), even if only slight. Instead, we should focus on quantifying *how much* variation is present. In (*20*), we make the distinction between sensitivity and robustness as follows: sensitivity measures the (continuous or near-continuous) degree of change in some reported value due to, e.g., changing the model. That is, sensitivity is a well-defined continuous quantity. But robustness is a subjective judgment call based on sensitivity; we say an analysis is non-robust if a particular observed change is deemed important in that a change in reported value affects a particular actionable decision. We see a version of this distinction in our check for dropping data (*66*). We assess sensitivity by checking the following: for different sizes of dropped data subsets, we ask what is the biggest change we expect to see in our reported quantity of interest. Ultimately an analyst must decide if robustness is a concern by asking whether the change in data or model required to see a substantively different conclusion is too small.

References

1. S. Flaxman, *et al.*, *Nature* **584**, 257 (2020).

2. N. Van Doremalen, *et al.*, *New England Journal of Medicine* **382**, 1564 (2020).

3. R. J. Fischer, *et al.*, *Emerging Infectious Diseases* **26**, 2253 (2020).

4. M. Heidemanns, A. Gelman, G. E. Morris, *Harvard Data Science Review* **2** (2020).

5. A. Gelman, J. Hullman, C. Wlezien, G. E. Morris, *Judgment and Decision Making* **15**, 863 (2020).

6. R. Meager, *American Economic Journal: Applied Economics* **11**, 57 (2019).

7. K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (MIT Press, 2012).

8. K. P. Murphy, *Probabilistic Machine Learning: An Introduction* (MIT Press, 2022).

9. A. Jacovi, A. Marasović, T. Miller, Y. Goldberg, *FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (2021), pp. 624–635.

10. E. Toreini, *et al.*, *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (2020), pp. 272–283.

11. R. R. Hoffman, *Cognitive Systems Engineering* pp. 137–164 (2017).

12. G. Möllering, *Sociology* **35**, 403 (2001).

13. N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, G. Loukas, *Computer Science Review* **34**, 100199 (2019).

14. B. Liu, *et al.*, *ACM Computing Surveys (CSUR)* **54**, 1 (2021).

15. R. Schwartz, L. Down, A. Jonas, E. Tabassi, *NIST Special Publication* **1270** (2021).

16. R. M. Neal, *Handbook of Markov Chain Monte Carlo* pp. 113–162 (2011).

17. R. Giordano, T. Broderick, R. Meager, J. Huggins, M. Jordan, *ICML 2016 Workshop on #Data4Good: Machine Learning in Social Good Applications* (2016).

18. D. M. Blei, A. Kucukelbir, J. D. McAuliffe, *Journal of the American Statistical Association* **112**, 859 (2017).

19. R. J. Giordano, T. Broderick, M. I. Jordan, *Neural Information Processing Systems* (2015).

20. R. Giordano, T. Broderick, M. I. Jordan, *Journal of Machine Learning Research* **19** (2018).

21. J. Huggins, M. Kasprzak, T. Campbell, T. Broderick, *International Conference on Artificial Intelligence and Statistics* (2020), pp. 1792–1802.

22. Stan Development Team, Stan modeling language users guide and reference manual (2021). Version 2.27.

23. C. M. Reinhart, K. S. Rogoff, *American Economic Review* **100**, 573 (2010).

24. M. O'Brien, *Atlantic* **18** (2013).

25. T. Herndon, M. Ash, R. Pollin, *Cambridge Journal of Economics* **38**, 257 (2014).

26. D. A. Linzer, *Journal of the American Statistical Association* **108**, 124 (2013).

27. K. A. Baggerly, K. R. Coombes, *Annals of Applied Statistics* pp. 1309–1334 (2009).

28. National Academies of Sciences, Engineering, and Medicine, *Reproducibility and Replicability in Science* (National Academies Press, 2019).

29. B. Haibe-Kains, *et al.*, *Nature* **586**, E14 (2020).

30. W. D. Heaven, *MIT Technology Review* (2020).

31. B. J. Heil, *et al.*, *Nature Methods* **18**, 1132 (2021).

32. M. Vartak, *et al.*, *Proceedings of the Workshop on Human-In-the-Loop Data Analytics* (2016), pp. 1–3.

33. V. Stodden, F. Leisch, R. D. Peng, *Implementing Reproducible Research* (CRC Press, 2014).

34. NASA Earth Observatory, Research satellites for atmospheric sciences, 1978–present: Serendipity and stratospheric ozone, `https://earthobservatory.nasa.gov/features/RemoteSensingAtmosphere/remote_sensing5.php` (2001). Accessed November 2021.

35. T. Gebru, *et al.*, *Communications of the ACM* **64**, 86 (2021).

36. V. Stodden, *Computing in Science and Engineering* **11**, 35 (2008).

37. V. Stodden, *et al.*, *Science* **354**, 1240 (2016).

38. J. Geweke, *Journal of the American Statistical Association* **99**, 799 (2004).

39. S. R. Cook, A. Gelman, D. B. Rubin, *Journal of Computational and Graphical Statistics* **15**, 675 (2006).

40. S. Talts, M. Betancourt, D. Simpson, A. Vehtari, A. Gelman, *arXiv preprint arXiv:1804.06788* (2020).

41. Y. Yao, A. Vehtari, D. Simpson, A. Gelman, *Proceedings of Machine Learning Research* **80**, 5581 (2018).

42. C. Hennig, A. Gelman, *Journal of the Royal Statistical Society A* **180**, 967 (2017).

43. J. K. Winkler, *et al.*, *JAMA Dermatology* **155**, 1135 (2019).

44. S. Passi, S. Barocas, *Proceedings of the Conference on Fairness, Accountability, and Transparency* (2019), pp. 39–48.

45. G. Shmueli, *Statistical Science* **25**, 289 (2010).

46. S. Barocas, M. Hardt, A. Narayanan, *Fairness and Machine Learning* (2019). `http://www.fairmlbook.org`.

47. S. Mitchell, E. Potash, S. Barocas, A. D'Amour, K. Lum, *Annual Review of Statistics and Its Application* **8**, 141 (2021).

48. A. D. Selbst, D. Boyd, S. A. Friedler, S. Venkatasubramanian, J. Vertesi, *Proceedings of the Conference on Fairness, Accountability, and Transparency* (2019), pp. 59–68.

49. B. Yu, *Bernoulli* **19**, 1484 (2013).

50. B. Yu, *Harvard Data Science Review* **2** (2020).

51. O. G. Ward, Z. Huang, A. Davison, T. Zheng, *Statistical Analysis and Data Mining* **14**, 5 (2021).

52. B. Yu, K. Kumbier, *Proceedings of the National Academy of Sciences* **117**, 3920 (2020).

53. A. L. Smith, D. M. Asta, C. A. Calder, *Statistical Science* **34**, 428 (2019).

54. S. Lubold, A. G. Chandrasekhar, T. H. McCormick, Identifying the latent space geometry of network models through analysis of curvature, *Tech. rep.*, National Bureau of Economic Research (2020).

55. R. Silberzahn, *et al.*, *Advances in Methods and Practices in Psychological Science* **1**, 337 (2018).

56. A. L. Smith, T. Zheng, A. Gelman, *Columbia Academic Commons doi:10.7916/D82F95D0* (2020).

57. J. O. Berger, D. R. Insua, F. Ruggeri, *Robust Bayesian Analysis* (Springer, 2000), pp. 1–32.

58. P. Gustafson, *Robust Bayesian Analysis* (Springer, 2000), pp. 71–88.

59. R. Giordano, R. Liu, M. I. Jordan, T. Broderick, *Bayesian Analysis, to appear* (2022).

60. W. T. Stephenson, *et al.*, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, G. Camps-Valls, F. J. R. Ruiz, I. Valera, eds. (PMLR, 2022), vol. 151 of *Proceedings of Machine Learning Research*, pp. 3308–3331.

61. Open Science Collaboration, *Science* **349**, aac4716 (2015).

62. M. Mitchell, *et al.*, *Proceedings of the Conference on Fairness, Accountability, and Transparency* (2019), pp. 220–229.

63. A. D'Amour, *et al.*, *Journal of Machine Learning Research* **23**, 1 (2022).

64. S. Ghosh, W. Stephenson, T. D. Nguyen, S. Deshpande, T. Broderick, *Advances in Neural Information Processing Systems* **33**, 8741 (2020).

65. M. T. Ribeiro, T. Wu, C. Guestrin, S. Singh, *Association for Computational Linguistics (ACL)* (2020).

66. T. Broderick, R. Giordano, R. Meager, *arXiv:2011.14999* (2020).

67. A. Kurakin, I. J. Goodfellow, S. Bengio, *Artificial Intelligence Safety and Security* (CRC Press, 2018), pp. 99–112.

68. J. K. Winkler, *et al.*, *JAMA dermatology* **155**, 1135 (2019).

69. A. B. Arrieta, *et al.*, *Information Fusion* **58**, 82 (2020).

70. F. Doshi-Velez, *et al.*, *arXiv:1711.01134* (2017).

71. Y. Zhang, Q. V. Liao, R. K. Bellamy, *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (2020), pp. 295–305.

72. B. Mittelstadt, C. Russell, S. Wachter, *Proceedings of the Conference on Fairness, Accountability, and Transparency* (2019), pp. 279–288.

73. C. Rudin, *Nature Machine Intelligence* **1**, 206 (2019).

74. G. E. P. Box, *Journal of the Royal Statistical Society A* **143**, 383 (1980).

75. D. B. Rubin, *Annals of Statistics* **12**, 1151 (1984).

76. J. Gabry, D. Simpson, A. Vehtari, M. Betancourt, A. Gelman, *Journal of the Royal Statistical Society A* **182**, 389 (2019).

77. R. Meager, *American Economic Review* **112**, 1818 (2022).

78. A. Gelman, Reverse-engineering the problematic tail behavior of the fivethirtyeight presidential election forecast (2020). `https://statmodeling.stat.columbia.edu/2020/10/24/reverse/`.

79. H. Akaike, *Proceedings of the Second International Symposium on Information Theory* (1973), pp. 267–281.

80. S. Watanabe, *Algebraic Geometry and Statistical Learning Theory* (Cambridge University Press, 2009).

81. H. Jiang, B. Kim, M. Guan, M. Gupta, *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems* (2018), pp. 5546—-5557.

82. S. Watanabe, *Journal of Machine Learning Research* **11**, 3571 (2012).

83. A. Gelman, I. Leenen, I. V. Mechelen, P. D. Boeck, *Statistical Methodology* **7**, 187 (2018).

84. W. Wang, A. Gelman, *Statistics and Its Interface* **8**, 153 (2015).

Acknowledgments