

Boba: Authoring and Visualizing Multiverse Analyses

Yang Liu, Alex Kale, Tim Althoff, and Jeffrey Heer

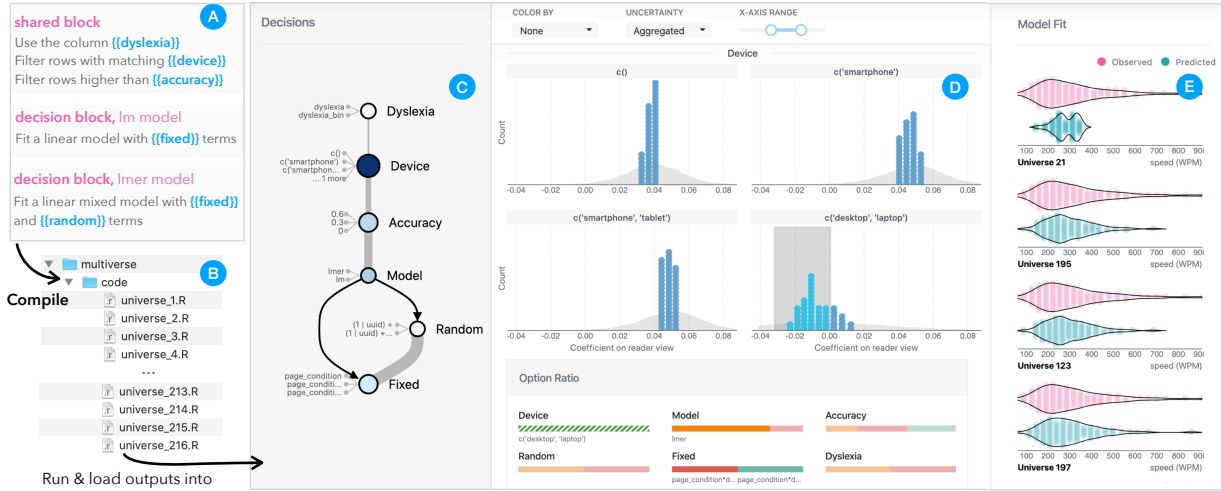


Fig. 1. Authoring and visualizing multiverse analyses with Boba. Users start by annotating a script with analytic decisions (a), from which Boba synthesizes a multiplex of possible analysis variants (b). To interpret the results from all analyses, users start with a graph of analytic decisions (c), where sensitive decisions are highlighted in darker blues. Clicking a decision node allows users to compare point estimates (d, blue dots) and uncertainty distributions (d, gray area) between different alternatives. Users may further drill down to assess the fit quality of individual models (e) by comparing observed data (pink) with model predictions (teal).

Abstract—Multiverse analysis is an approach to data analysis in which all “reasonable” analytic decisions are evaluated in parallel and interpreted collectively, in order to foster robustness and transparency. However, specifying a multiverse is demanding because analysts must manage myriad variants from a cross-product of analytic decisions, and the results require nuanced interpretation. We contribute Boba: an integrated domain-specific language (DSL) and visual analysis system for authoring and reviewing multiverse analyses. With the Boba DSL, analysts write the shared portion of analysis code only once, alongside local variations defining alternative decisions, from which the compiler generates a multiplex of scripts representing all possible analysis paths. The Boba Visualizer provides linked views of model results and the multiverse decision space to enable rapid, systematic assessment of consequential decisions and robustness, including sampling uncertainty and model fit. We demonstrate Boba’s utility through two data analysis case studies, and reflect on challenges and design opportunities for multiverse analysis software.

Index Terms—Multiverse Analysis, Statistical Analysis, Analytic Decisions, Reproducibility

1 INTRODUCTION

The last decade saw widespread failure to replicate findings in published literature across multiple scientific fields [2, 6, 35, 41]. As the replication crisis emerged [1], scholars began to re-examine how data analysis practices might lead to spurious findings. An important contributing factor is the flexibility in making analytic decisions [16, 17, 48]. Drawing inferences from data often involves many decisions: what are the cutoffs for outliers? What covariates should one include in statistical models? Different combinations of choices might lead to diverging results and conflicting conclusions. Flexibility in making decisions might inflate false-positive rates when researchers explore multiple alternatives and selectively report desired outcomes [48], a practice known as p-hacking [34]. Even without exploring multiple paths, fixating on a single analytic path might be less rigorous, as multiple justifiable alternatives might exist and picking one would be arbitrary. For example, a crowdsourced study [47] shows that well-

intentioned experts still produce large variations in analysis outcomes when analyzing the same dataset independently.

In response, prior work proposes *multiverse analysis*, an approach to outline all “reasonable” alternatives a-priori, exhaust all possible combinations between them, execute the end-to-end analysis per combination, and interpret the outcomes collectively [49, 50]. A multiverse analysis demonstrates the extent to which conclusions are robust to sometimes arbitrary analytic decisions. Furthermore, reporting the full range of possible results, not just those which fit a particular hypothesis or narrative, helps increase the transparency of a study [44].

However, researchers face a series of barriers when performing multiverse analyses. Authoring a multiverse is tedious, as researchers are no longer dealing with a single analysis, but hundreds of forking paths resulting from possible combinations of analytic decisions. Without proper scaffolding, researchers might resort to multiple, largely redundant analysis scripts [26], or rely on intricate control flow structure including nested for-loops and if-statements [54]. Interpreting the outcomes of a vast number of analyses is also challenging. Besides gauging the overall robustness of the findings, researchers often seek to understand what decisions are critical in obtaining particular outcomes (e.g., [49, 50, 59]). As multiple decisions might interact, understanding the nuances in how decisions affect robustness will require a comprehensive exploration, suggesting a need for an interactive interface.

To lower these barriers, we present Boba, an integrated domain-specific language (DSL) and visualization system for multiverse author-

• All authors are with the University of Washington. E-mails: yliu0, kalea, althoff, jheer@uw.edu.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

ing and interpretation. Rather than managing myriad analysis versions in parallel, the Boba DSL allows users to specify the shared portion of the analysis code only once, alongside local variations defining alternative analysis decisions. The compiler enumerates all compatible combinations of decisions and synthesizes individual analysis scripts for each path. As a meta-language, the Boba DSL is agnostic to the underlying programming language of the analysis script (e.g., Python or R), thereby supporting a wide range of data science use cases.

The Boba Visualizer facilitates assessment of the output of all analysis paths. We support a workflow where users view the results, refine the analysis based on model quality, and commit the final choices to making inference. The system first provides linked views of both analysis results and the multiverse decision space to enable a systematic exploration of how decisions do (or do not) impact outcomes. Besides decision sensitivity, we enable users to take into account sampling uncertainty and model fit by comparing observed data with model predictions [14]. After viewing the results, users can exclude models poorly suited for inference by adjusting a model fit threshold, or adopt a principled approach based on model averaging to incorporate model fit in inference. We discuss the implications of post-hoc refinement, along with other challenges in multiverse analysis, in our design reflections.

We evaluate Boba in a code comparison example and two data analysis case studies. We first demonstrate how the Boba DSL eliminates custom control-flows when implementing a real-world multiverse of considerable complexity. Then, in two multiverses replicated from prior work [49, 59], we show how the Boba Visualizer affords multiverse interpretation, enabling a richer understanding of robustness, decision patterns, and model fit quality via visual inspection. In both case studies, model fit visualizations surface previously overlooked issues and change what one can reasonably take away from these multiverses.

2 RELATED WORK

We draw on prior work on authoring and visualizing multiverse analyses, and approaches for authoring alternative programs and designs.

2.1 Multiverse Analysis

Analysts begin a multiverse analysis by identifying reasonable analytic decisions *a-priori* [37, 49, 50]. Prior work defines reasonable decisions as those with firm theoretical and statistical support [49], and decisions can span the entire analysis pipeline from data collection and wrangling to statistical modeling and inference [30, 56]. While general guidelines such as a decision checklist [56] exist, defining what decisions are reasonable still involves a high degree of researcher subjectivity.

The next step in multiverse analyses is to exhaust all compatible decision combinations and execute the analysis variants (we call a variant a *universe*). Despite the growing interest in performing multiverse analysis (e.g., [6, 9, 21, 36, 43]), few tools currently exist to aid authoring. Young and Holsteen [59] developed a STATA module that simplifies multimodel analysis into a single command, but it only works for simple variable substitution. *Rdfanalysis* [13], an R package, supports more complex alternative scenarios beyond simple value substitution, but the architecture assumes a linear sequential relationship between decisions. Our DSL similarly provides scaffolding for specifying a multiverse, but it has a simpler syntax, extends to other languages, and handles procedural dependencies between decisions.

After running all universes, the next task is to interpret results collectively. Some prior studies visualize results from individual universes by either juxtaposition [42, 49, 50] or animation [12]. Visualizations in other studies apply aggregation [11, 40], for example showing a histogram of effect sizes. The primary issue with juxtaposing or animating individual outcomes is scalability, though this might be circumvented by sampling [49]. Our visualizer shows individual outcomes, but overlays or aggregates outcomes in larger multiverses to provide scalability.

Besides the overall robustness, many studies also investigate which analytic decisions are most consequential. The simplest approach is a table [8, 10, 42, 50] where rows and columns map to decisions, and cells represents outcomes from individual universes. Simonsohn et al. [49] extend this idea, visualizing the decision space as a matrix beneath a plot of sorted effect sizes. These solutions might not scale as they

juxtapose individual outcomes, and the patterns of how outcomes vary might be difficult to identify depending on the spatial arrangements of rows and columns. Another approach [40] slices the aggregated distribution of outcomes along a decision dimension to create a trellis plot (*a.k.a.* small multiples [53]). The trellis plot shows how results vary given a decision, but does not convey what decisions are prominent given certain results. Our visualizer uses trellis plots and supplements it with brushing to show how decisions contribute to particular results.

Finally, prior work relies on various strategies to infer whether a hypothesized effect occurs given a multiverse. The simplest approach is counting the fraction of universes having a significant p-value [8, 50] and/or an effect with the same sign [11]. Young and Holsteen [59] calculate a robustness ratio analogous to the *t*-statistic. Simonsohn et al. [49] compare the actual multiverse results to a null distribution obtained from randomly shuffling the variable of interest. We build upon Simonsohn’s approach and use weighted model averaging based on model fit quality [58] to aggregate uncertainty across universes.

While multiverse analysis is a recent concept, prior work has developed visual analytics approaches for similar problems. For example, multiverse analysis fits into the broader definition of parameter space analysis [5, 46], a concept originally proposed for understanding inputs and outputs of simulation models. Visual analytics systems for pre-processing time-series data [3, 4] also propose ways to generate and visualize alternative results, for example via superposition.

2.2 Authoring Alternative Programs and Designs

Analysts often manage alternatives from exploratory work by duplicating code snippets and files, but these ad-hoc variants can be messy and difficult to keep track of [18, 26]. Provenance tracking tools, especially those with enhanced history interactions [26, 27], provide a mechanism to track and restore alternative versions. In Variolite [26], users select a chunk of code directly in an editor to create and version alternatives. We also allow users to insert local alternatives in code, but instead of assuming that users interact with one version at a time, we generate multiple variants mapping to possible combinations of alternatives.

A related line of work supports manipulating multiple alternatives simultaneously. Techniques like subjunctive interfaces [31, 32] and Parallel Pies [52] embed and visualize multiple design variants in the same space, and Parallel Pies allows users to edit multiple variants in parallel. Juxtapose [19] extends the mechanism to software development, enabling users to author program alternatives as separate files and edit code duplicates simultaneously with linked editing. A visualization authoring tool for responsive design [20] also enables simultaneous editing across variants. Our DSL uses a centralized template such that edits in the shared portion of code affect all variants simultaneously.

3 DESIGN REQUIREMENTS

Our overarching goal is to make it easier for researchers to conduct multiverse analyses. From prior literature and our past experiences, we identify barriers in authoring a multiverse and visualizing its results, and subsequently identify tasks that our tools should support.

3.1 Requirements for Authoring Tool

As noted in prior work [12, 30], specifying a multiverse is tedious. This is primarily because a multiverse is composed of many forking paths, yet non-linear program structures are not well supported in conventional tools [45]. One could use a separate script per analytic path, such that it is easy to reason with an individual variant, but these variants are redundant and difficult to maintain [26]. Alternatively, one could rely on control flows in a single script to simulate the nonlinear execution, but it is hard to selectively inspect and rerun a single path, and deeply nested control flows are thought to be a software development anti-pattern [33]. Instead, a tool should eliminate the need to write redundant code and custom control flows, while allowing analysts to simultaneously update variants and reason with a single variant. Compared to arbitrary non-linear paths from an iterative exploratory analysis, the forking paths in multiverses are usually highly systematic. We take advantage of this characteristic, and account for other scenarios common in existing multiverse analyses. We distill the following design requirements:

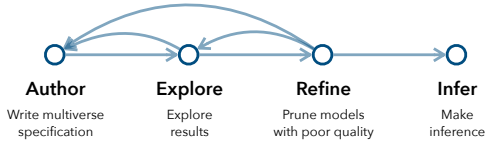


Fig. 2. The intended workflow for multiverse analysis in Boba.

R1: Multiplexing. Users should be able to specify a multiverse by writing the shared portion of the analysis source code along with analytic decisions, while the tool creates the forking paths for them. Users should also be able to reason about a single universe and update all universes simultaneously.

R2: Decision Complexity. Decisions come in varying degrees of complexity, from simple value replacements (e.g., cutoffs for excluding outliers) to elaborate logic requiring multiple lines of code to implement. The tool should allow succinct ways to express simple value replacements while at the same time support more complex decisions.

R3: Procedural Dependency. Existing multiverses [9, 50] contain *procedural dependencies* [30], in which a downstream decision only exists if a particular upstream choice is made. For example, researchers do not need to choose priors if using a Frequentist model instead of a Bayesian model. The tool should support procedural dependencies.

R4: Linked Decisions. Due to idiosyncrasies in implementation, the same conceptual decision can manifest in multiple forms. For example, the same set of parameters can appear in different formats to comply with different function APIs. Users should be able to specify different implementations of a high-level decision.

R5: Language Agnostic. Users should be able to author their analysis in any programming languages, as potential users are from various disciplines adopting different workflows and programming languages.

3.2 Requirements for Visual Analysis System

After executing all analytic paths in a multiverse to obtain corresponding results, researchers face challenges interpreting the results collectively. The primary task in prior work (Sect. 2) is understanding the robustness of results across all reasonable specifications. If robustness checks indicate conflicting conclusions, a natural follow-up task is to identify what decisions are critical to reaching a particular conclusion or what decisions produce large variations in results.

We also propose new tasks to cover potential blind spots in prior work. First, besides point estimates, a tool should convey appropriate uncertainty information to help users gauge the end-to-end uncertainty caused by both sampling and decision variations, and compare the variance between conditions. Second, it is important to assess the model fit quality to distinguish trustworthy models from the ones producing questionable estimates. Uncertainty information and fit issues become particularly important during statistical inference. Users should be able to propagate uncertainty in the multiverse to support judgments about the overall reliability of effects, and they should be able to refine the multiverse to exclude models with fit issues before making inferences.

We identify six tasks that our visual analysis system should support:

- T1: *Decision Overview* – gain an overview of the decision space to understand the multiverse and contextualize subsequent tasks.
- T2: *Robustness Overview* – gauge the overall robustness of findings obtained through all reasonable specifications.
- T3: *Decision Impacts* – identify what combinations of decisions lead to large variations in outcomes, and what combinations of decisions are critical in obtaining specific outcomes.
- T4: *Uncertainty* – assess the end-to-end uncertainty as well as uncertainty associated with individual universes.
- T5: *Model Fit* – assess the model fit quality of individual universes to distinguish trustworthy models from questionable ones.
- T6: *Inference* – perform statistical inference to judge how reliable the hypothesized effect is, while accounting for model quality.

Besides the tasks, our system should also support the following data characteristics (S1) and types of statistical analyses (S2). First, our visual encoding should be scalable to large multiverses and large input datasets. This is because the multiverse size increases exponentially

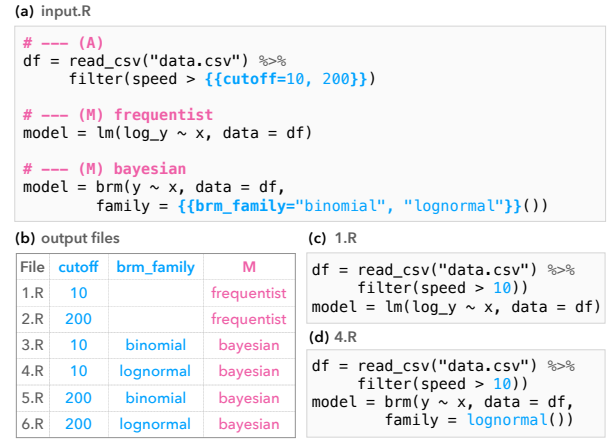


Fig. 3. An example Boba specification. The user annotates an R script (a) with two placeholder variables (blue) and three code blocks (pink). The compiler synthesizes six files (b). In the example output files (c) and (d), placeholder variables are replaced by their possible values, and only one version of the decision block M is present.

with the number of decisions, with the median size in practice being in the thousands [30]. The input datasets might also have arbitrarily many observations. Second, we should support common simple statistical tests in HCI research [39], including ANOVA and linear regressions.

3.3 Workflow

We propose a general workflow for multiverse analysis with four stages (Fig. 2). In this workflow, users *author* the multiverse specification, *explore* the results, *refine* the multiverse by pruning universes with poor model quality, and make *inference*. Users should be free to cycle between the first three stages, because upon exploring the results, users might discover previously overlooked alternatives, or notice that certain decisions are poorly suited for inference. In this case, they might iterate on their multiverse specification to include only decisions resulting in universes that seem “reasonable”. However, once users proceed to the *inference* stage, they should not return to any of the prior stages.

4 THE BOBA DSL

We design a domain-specific language (DSL) to aid the authoring of multiverse analyses. The DSL formally models an analysis decision space, providing critical structure that the visual analysis system later leverages. With the DSL, users annotate the source code of their analysis to indicate decision points and alternatives, and provide additional information for procedural dependencies between decisions. The specification is then compiled to a set of universe scripts, each containing the code to execute one analytic path in the multiverse. An example Boba specification for a small multiverse is shown in Fig. 3.

4.1 Language Constructs

The basic language primitives in the Boba DSL consist of source code, placeholder variables, blocks, constraints, and code graphs.

Source Code. The most basic ingredient of an annotated script is the source code (Fig. 3a, black text). The compiler treats the source code as a string of text, which according to further language rules will be synthesized into text in the output files. As the compiler is agnostic about the semantics of the source code, users are free to write the source code in any programming language (R5).

Placeholder Variables. Placeholder variables are useful to specify decisions points consisting of simple value substitution (R2). To define a placeholder variable, users provide an identifier and a set of possible alternative values that the variable can take up (Fig. 3a, blue text). To use the variable, users insert the identifier into any position in the source code. During synthesis, the compiler removes the identifier and replaces it with one of its alternative values. Variable definition may occur at the same place as its usage (Fig. 3a) or ahead of time inside the config block (supplemental Fig. 2).

Code Blocks. Code blocks (Fig. 3a, pink text) divide the source code into multiple snippets of one or more lines of code, akin to cells in a computational notebook. A block can be a *normal block* (Fig. 3a, block A), or a *decision block* (Fig. 3a, block M) with multiple versions. The content of a normal block will be shared by all universes, whereas only one version of the decision block will appear in a universe. Decision blocks allow users to specify alternatives that require more elaborate logic to define (R2). In the remainder of Sect. 4, *decision points* refer to placeholder variables and decision blocks.

With the constructs introduced so far, a natural way to express procedural dependency (R3) is to insert a placeholder variable in some, but not all versions of a decision block. For example, in Fig. 3, the variable `brm_family` only exists when bayesian of block M is chosen.

Constraints. By default, Boba assumes all combinations between decision points are valid. Constraints allow users to express dependencies between decision points, for example infeasible combinations, which will restrict the universes to a smaller set. Boba supports two types of constraints: procedural dependencies (R3) and linked decisions (R4).

A procedural dependency constraint is attached to a decision point or one of its alternatives, and has a conditional expression to determine when the decision/alternative should exist (Fig. 4b, orange text). Variables within the scope of the conditional expression are declared decision points, and the values are the alternatives that the decision points have taken up. For example, the first constraint in Fig. 4b indicates that `ECL computed` is not compatible with `NMO reported`.

The second type of constraint allows users to *link* multiple decision points, indicating that these decision points are different manifestations of a single conceptual decision (R4, see supplemental Fig. 2). Linked decisions have one-to-one mappings between their alternatives, such that the *i*-th alternatives are chosen together in the same universe. One-to-one mappings can also be expressed using multiple procedural dependencies, but linked decisions make them easier to specify.

Code Graph. Users may further specify the execution order between code blocks as a directed acyclic graph (DAG), where a parent block executes before its child. To create a universe, the compiler selects a linear path from the start to the end, and concatenates the source code of blocks along the path. Branches in the graph represent alternative paths that appear in different universes. Users can flexibly express complex dependencies between blocks with the graph, including procedural dependencies (R3). For example, to indicate that block `prior` should only appear after block `bayesian` but not block `frequentist`, the user simply makes `prior` a descendant of `bayesian` but not `frequentist`.

4.2 Compilation and Runtime

The compiler parses the input script, computes compatible combinations between decisions, and generates output scripts. More details about compilation are in the supplemental material. Besides executable universe scripts, the compiler also outputs a summary table that keeps track of all the decisions made in each universe, along with other intermediate data that can be ingested into the Boba Visualizer.

Boba infers the language of the input script based on its file extension and uses the same extension for output scripts. These output scripts might be run with the corresponding script language interpreter. Universe scripts log the results into separate files, which will be merged together after all scripts finish execution. Each universe must output a point estimate, along with other optional data such as a p-value, a model quality metric, or a set of sampled estimates to represent uncertainty. As the universe scripts are responsible for computations such as extracting point estimates and computing uncertainty, we provide language-specific utilities for a common set of model types to generate these visualizer-friendly outputs. We also provide a command-line tool for users to (1) invoke the compiler, (2) execute the generated universe scripts, (3) merge the universe outputs, and (4) invoke the visualizer as a local server reading the intermediate output files.

4.3 Example: Replicating a Real-World Multiverse

We use a real-world multiverse example [50] to illustrate how the Boba DSL eliminates the need for custom control flows otherwise required

Figure 4 consists of two panels, (a) and (b), showing code snippets. Panel (a) shows R code with various highlights: yellow for nested for-loops, blue for if-statements, and purple for nested if-statements. Panel (b) shows the Boba DSL specification, which is a more concise representation of the same logic, using decision points and procedural dependencies to replace the complex control flow of the R code.

Fig. 4. Specification of a real-world multiverse analysis [50] with five decisions and a procedural dependency. (a) Markup of the R code written by original authors, with custom control flow (nested for-loops and if-statements) highlighted. (b) Markup of the Boba DSL specification.

for authoring a multiverse in a single script. The multiverse, originally proposed by Steegen *et al.* [50], contains five decisions and a procedural dependency. Fig. 4a shows a markup of the R code implemented by the original authors (we modified the lines in purple to avoid computing infeasible paths). The script starts with five nested for-loops (yellow highlight) to repeat the analysis for every possible combination of the five decisions. Then, depending on the indices of the current decisions, the authors either index into an array, or use if-statements to define alternative program behaviors (blue highlight). Finally, to implement a procedural dependency, it is necessary to skip the current iteration when incompatible combinations occur (purple highlight).

The resulting script has multiple issues. First, the useful snippets defining multiverse behavior start at five levels of nesting at minimum. Such deeply nested code is often considered to be hard to read [33]. Second, nested control flows are not easily amenable to parallel execution. Third, without additional error-handling mechanisms, an error in the middle will terminate the program before any results are saved.

The corresponding specification in the Boba DSL is shown in Fig. 4b. The script starts directly with the preprocessing code shared by all universes. It then uses decision code blocks to define alternative snippets in decision `NMO` and `ECL`, and uses a placeholder variable to simulate the value array for a simpler decision `F`. It additionally specifies constraints (orange text) to signal incompatible paths. Compared to Fig. 4a, this script reduces the amount of boilerplate code needed for control-flows and does not require any level of nesting. The script compiles to 120 separate files. Errors in one universe no longer affect the completion of others due to distributed execution, it is trivial to execute universes in parallel, and users can selectively review and debug a single analysis.

5 THE BOBA VISUALIZER

Next, we introduce Boba Visualizer, a visual analysis system for interpreting the outputs from all analysis paths. We present the system features and design choices in a fictional usage scenario where Emma, an HCI researcher, uses the visualizer to explore a multiverse on data collected in her experiment. We construct the multiverse based on how the authors of a published research article [28] might analyze their data, but the name “Emma” and her workflow are fictional.

Emma runs an experiment to understand whether “Reader View” – a modified web page layout – improves reading speed for individuals with dyslexia. She assigns participants to use Reader View or standard websites, measures their reading speed, and collects other variables such as accuracy, device, and demographic information. She plans to build a model with reading speed as the dependent variable. To check whether her conclusion depends on idiosyncratic specifications, Emma identifies six analytic decisions, including the device type and accuracy cutoff used to filter participants, ways to operationalize dyslexia, the statistical model, and its random and fixed terms. She then writes a multiverse specification in the Boba DSL, compiles it to 216 analysis scripts, and runs all scripts to obtain a set of effect sizes. She loads these outputs into the Boba Visualizer.

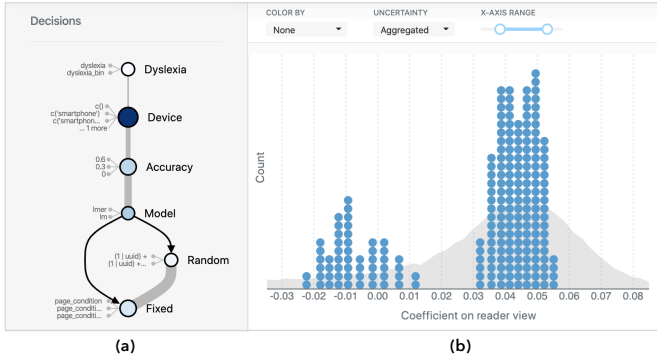


Fig. 5. **Decision view and outcome view.** (a) The decision view shows analytic decisions as a graph with order and dependencies between them, and highlights more sensitive decisions in darker colors. (b) The outcome view visualizes outputs from all analyses, including individual point estimates and aggregated uncertainty.

5.1 Outcome View

On system start-up, Emma sees an overview distribution of point estimates from all analyses (Fig. 5b). The majority of the coefficients are positive, but a smaller peak around zero suggests no effect.

The outcome view visualizes the final results of the multiverse, including point estimates (e.g., model coefficient of reader view, the independent variable encoding experimental conditions) and uncertainty information. By default, the chart contains outcomes from all universes in order to show the overall robustness of the conclusion (T2).

Boba visualizes one point estimate from each universe using a density dot plot [57] (Fig. 5b, blue dots). The x-axis encodes the magnitude of the estimate; dots in the same bin are stacked along the y-axis. To accommodate large multiverses (S1), we allow dots to overlap along the y-axis, which represents count. Density dot plots more accurately depict gaps and outliers in data than histograms [57]. One-to-one mapping between dots and universes affords direct manipulation interactions such as brushing and details-on-demand, as we introduce later.

Boba visualizes end-to-end uncertainty from both sampling and decision variations (T4) as a background area chart (Fig. 5b, gray area). When the uncertainty introduced by sampling variations is negligible, the area chart follows the dot plot distribution closely. In contrast, the mismatch of the two distributions in Fig. 5b indicates considerable sampling uncertainty. We compute the end-to-end uncertainty by aggregating over modeling uncertainty from all universes. Specifically, we first calculate $\hat{f}(x) = \sum_{i=1}^N f_i(x)$, where $f_i(x)$ is the sampling distribution of the i -th universe, and N is the overall multiverse size. Then, we scale the height of the area chart such that the total area under $\hat{f}(x)$ is approximately the same as the total area of dots in the dot plot.

Besides aggregated uncertainty, Boba allows users to examine uncertainty from individual universes (Fig. 7). In a dropdown menu, users can switch to view the probability density functions (PDFs) or cumulative distribution functions (CDFs) of all universes. A PDF is a function that maps the value of a random variable to its likelihood, whereas a CDF gives the area under the PDF. In both views, we draw a cubic basis spline for the PDF or CDF per universe, and reduce the opacity of the curves to visually “merge” the curves within the same space. There is again a one-to-one mapping between a visual element and a universe to afford interactions. To help connect point estimates and uncertainty, we draw a strip plot of point estimates beneath each PDFs/CDFs chart (Fig. 7, blue dashes), and show the corresponding sampling distribution PDF when users mouse over a universe in the dot plot.

5.2 Decision View

As the overall outcome distribution suggests conflicting conclusions, Emma wants to investigate what decisions lead to changes in results. She first familiarizes herself with the available decisions.

The decision view shows a graph of analytic decisions in the multiverse, along with their order and dependencies (Fig. 5a), helping users understand the decision space and inviting further exploration (T1).

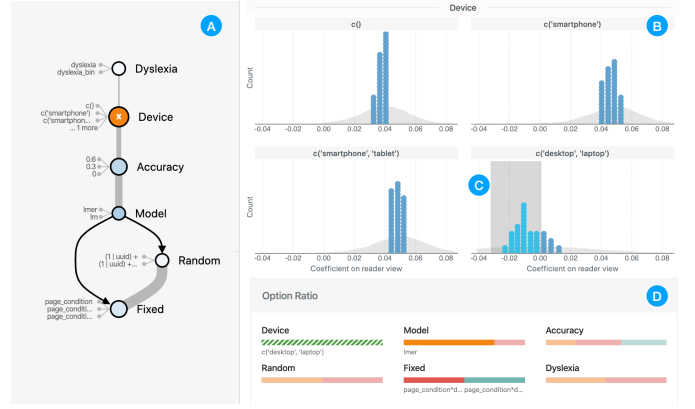


Fig. 6. **Facet and Brushing.** Clicking a node in the decision view (a) divides the outcome view into a trellis plot (b), answering questions like “does the decision lead to large variations in effect size?” Brushing a region in the outcome view (c) reveals dominant alternatives in the option ratio view (d), answering questions like “what causes negative results?”

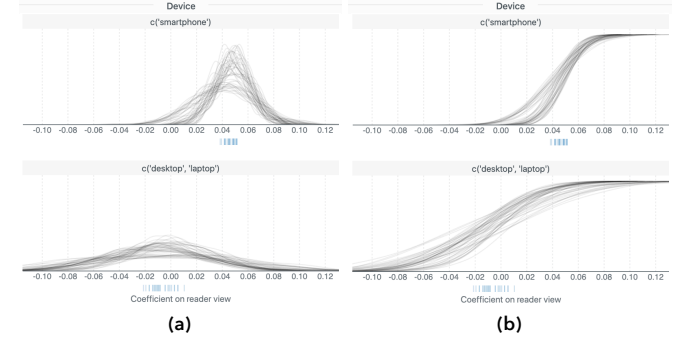


Fig. 7. **PDFs (a) and CDFs (b) views** visualize sampling distributions from individual universes. Toggling these views in a trellis plot allows users to compare the variance between conditions.

We adapt the design of Analytic Decision Graphs [30] to show decisions in the context of the analysis process. Nodes represent decisions and edges represent the relationships between decisions: light gray edges encode *temporal order* (the order that decisions appear in analysis scripts) and black edges encode *procedural dependencies*. To avoid visual clutter, we aggregate the information about alternatives, using the size of a node to represent the number of alternatives and listing a few example alternative values besides a node. Compared to viewing decisions in isolation, this design additionally conveys the analysis pipeline to help users better reason with the ramifications of a decision.

The underlying data structure for the graph is inferred from the Boba DSL specification. We infer decision names from variable identifiers. We extract temporal order as the order that decision points are first used in the specification, and detect procedural dependencies from user-specified constraints and code graph structure. After we extract the data structure, we apply a Sugiyama-style [51] flow layout algorithm, as implemented in Dagre [38], to compute the graph layout.

5.2.1 Sensitivity

When viewing the decision graph, Emma notes a sensitive decision “Device” which is highlighted in a darker color (Fig. 5a).

To highlight decisions that lead to large changes in analysis outcomes (T3), we compute the marginal sensitivity per decision and color the nodes using a sequential color scale. The color encoding helps draw the user’s attention to consequential decisions to aid initial exploration.

Boba implements two methods for estimating sensitivity. The first method is based on the F-Test in one-way ANOVA, which quantifies how much a decision shifts the means of results compared to variance (details in supplemental material). The second method uses the Kolmogorov–Smirnov (K–S) statistic, a non-parametric method to quantify the difference between two distributions. We first compute pairwise K–S statistics between all pairs of alternatives in decision D :

$$K = \left\{ \sup_x |f_i(x) - f_j(x)| : i, j \in \binom{S}{2} \right\}$$

where $f_i(x)$ is the empirical distribution function of results following the i -th alternative, and $S = \{1, 2, \dots, k\}$ where k is the number of alternatives in D . We then take the median of K as the sensitivity score s_D . In both methods, we map s_D to a single-hue color ramp of blue shades. As the F-Test relies on variance, which is not a reasonable measure for dispersion of some distributions, Boba uses the nonparametric K-S statistic by default. Users can override the default in the config file.

5.3 Facet and Brushing

Seeing that the decision “Device” has a large impact, Emma clicks on the node to further examine how results vary (Fig. 6a). She finds that one condition exclusively produces point estimates around zero (Fig. 6b) and it also has a much larger variance (Fig. 7).

Clicking a node in the decision graph facets the outcome distribution into a trellis plot, grouping subsets of universes by shared decision alternatives. This allows users to systematically examine the trends and patterns caused by a decision (T3). Akin to the overall outcome distribution, users can toggle between point estimates and uncertainty views to compare the variance between conditions. The trellis plot can be further divided on a second decision by shift-clicking a second node to show the interaction between two decisions. With faceting, users may comprehensively explore the data by viewing all univariate and bivariate plots. Sensitive decisions are automatically highlighted, so users might quickly find and examine consequential decisions as well.

What decisions lead to negative estimates? Emma brushes negative estimates in a subplot (Fig. 6c) and inspects option ratios (Fig. 6d).

Brushing a region in the outcome view updates the option ratio view. The option ratio view shows percentages of decision options to reveal dominating alternatives that produce specific results (T3).

The option ratio view visualizes each decision as a stacked bar chart, where bar segment length encodes the percentage of results coming from an alternative. When the user brushes a range of results, the bars are updated accordingly to reflect changes, and dominating alternatives (those having a higher percentage than default) are highlighted. For example, Emma notices that the `lmer` model (i.e., linear mixed-effect model in R) and two sets of fixed effects are particularly responsible for the negative outcomes in Fig. 6c. We color the bar segments using a categorical color scale to make bars visually distinguishable. Decisions used to divide a trellis plot are marked by a striped texture, as each trellis subplot only contains one alternative by definition.

5.4 Model Fit View

Now that Emma understands what decisions lead to null effects, she wonders if these results are from trustworthy models. She changes the color-by field to get an overview of model fit quality (Fig. 8a) and sees that the universes around zero have a poorer fit. She then uses a slider to remove universes that fail to meet a quality threshold (Fig. 8b).

Boba enables an overview of model fit quality across all universes (T5) by coloring the outcome view with a model quality metric (Fig. 8a). We use normalized root mean squared error (NRMSE) to measure model quality and map NRMSE to a single-hue colormap of blue shades where a darker blue indicates a better fit.

To obtain NRMSE, we first compute the overall mean squared prediction error (MSE) from a k -fold cross validation:

$$MSE = \frac{1}{k} \sum_{j=1}^k \frac{1}{n_j} \sum_{i=1}^{n_j} (y_i - \hat{y}_i)^2$$

where k is the number of folds (we set $k = 5$ in all examples), n_j is the size of the test set in the j -th fold, y_i is the observed value, and \hat{y}_i is the predicted value. We then normalize the MSE by the span of the maximum y_{max} and minimum y_{min} values of the observed variable:

$$NRMSE = \sqrt{MSE} / (y_{max} - y_{min})$$

We use k -fold cross validation [55] because metrics such as Akaike Information Criterion cannot be used to compare model fit across

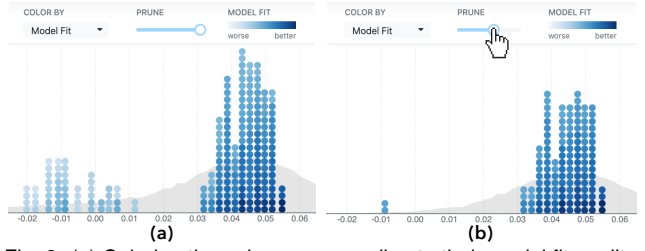


Fig. 8. (a) Coloring the universes according to their model fit quality. (b) Removing universes that fail to meet a model quality threshold.

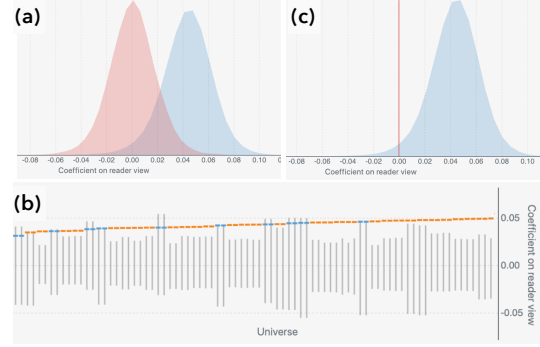


Fig. 9. **Inference views.** (a) Aggregate plot comparing the possible outcomes of the actual multiverse (blue) and the null distribution (red). (b) Detailed plot showing the individual point estimates and the range between the 2.5th and 97.5th percentile in the null distribution (gray line). Point estimates outside the range are colored in orange. (c) Alternative aggregate plot where a red line marks the expected null effect.

classes of models (e.g., hierarchical vs. linear) [15]. Prior work shows that cross validation performs better in estimating predictive density for a new dataset than information criteria [55], suggesting that it is a better approximation of out-of-sample predictive validity.

To further investigate model quality, Emma drills down to individual universes by clicking a dot in the outcome view. She sees in the model fit view (Fig. 1e) that a model gives largely mismatched predictions.

Clicking a result in the outcome view populates the model fit view with visual predictive checks, which show how well predictions from a given model replicate the empirical distribution of observed data [14], allowing users to further assess model quality (T5). The model fit visualization juxtaposes violin plots of the observed data and model predictions to facilitate comparison of the two distributions (see Fig. 1e). Within the violin plots, we overlay observed and predicted data points as centered density dot plots to help reveal discrepancies in approximation due to kernel density estimation. When the number of observations is large (S1), we plot a representative subset of data, sampled at evenly spaced percentiles, as centered quantile dotplots [25]. As clicking individual universes can be tedious, the model fit view suggests additional universes that have similar point estimates to the selected universe.

5.5 Inference

After an in-depth exploration, Emma proceeds to the final step, asking “given the multiverse, how reliable is the effect?” She confirms a warning dialog to arrive at the inference view (Fig. 9).

To support users in making inference and judging how reliable the hypothesized effect is (T6), Boba provides an inference view at the end of the analysis workflow, after users have engaged in exploration. Once in the inference view, all earlier views and interactions are inaccessible to avoid multiple comparison problems [60] arising from repeated inference. The inference view contains different plots depending on the outputs from the authoring step, so that users can choose between robust yet computationally-expensive methods and simpler ones.

A more robust inference utilizes the null distribution – the expected distribution of outcomes when the null hypothesis of no effect is true. In this case, the inference view shows an aggregate plot followed by a detailed plot (Fig. 9ab). The aggregate plot (Fig. 9a) compares the null

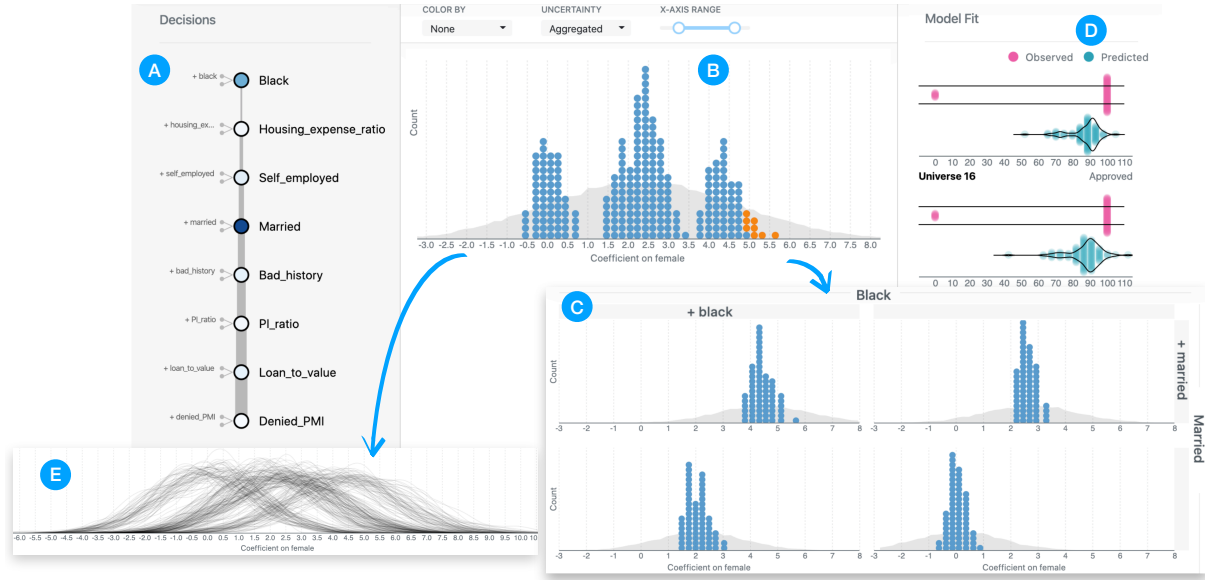


Fig. 10. A case study on how model estimates are robust to control variables in a mortgage lending dataset. (a) Decision view shows that *black* and *married* are two consequential decisions. (b) Overall outcome distribution follows a multimodal distribution with three peaks. (c) Trellis plot of *black* and *married* indicates the source of the peaks. (d) Model fit plots show that models produce numeric predictions while observed data is categorical. (e) PDFs of individual sampling distributions show significant overlap of the three peaks.

distribution (red) to possible outcomes of the actual multiverse (blue) across sampling and decision variations. The detailed plot (Fig. 9b) shows point estimates (colored dots) against 95% confidence intervals representing null distributions (gray lines) for each universe. Each point estimate is orange if it is outside the range, or blue otherwise. Underneath both plots, we provide descriptions (supplemental Fig. 1) to guide users in interpretation: For the aggregate plot, we prompt users to compare the distance between the averages of the two densities to the spread. For the detailed plot, we count the number of universes with the point estimate outside its corresponding range. If the null distribution is unavailable, Boba shows a simpler aggregate plot (Fig. 9c) where the expected effect size under the null hypothesis is marked with a red line.

To compute the null distribution, we permute the data with random assignment [49]. Specifically, we shuffle the column with the independent variable (reader view in this case) N times, run the multiverse of size M on each of the shuffled datasets, and obtain $N \times M$ point estimates. As reader view and speed are independent in the shuffled datasets, these $N \times M$ point estimates constitute the null distribution.

In addition, Boba enables users to propagate concerns in model fit quality to the inference view in two possible ways. The first way employs a model averaging technique called *stacking* [58] to take a weighted combination of the universes according to their model fit quality. The technique learns a simplex of weights, one for each universe model, via optimization that maximizes the log-posterior-density of the held-out data points in a k -fold cross validation. Boba then takes a weighted combination of the universe distributions to create the aggregate plot. While stacking provides a principled way to approach model quality, it can be computationally expensive. As an alternative, Boba excludes the universes below the model quality cutoff users provide in Sect. 5.4. The decisions of the cutoff and whether to omit the universes are made before a user enters the inference view.

6 CASE STUDIES

We evaluate Boba through a pair of analysis case studies, where we implement the multiverse using the Boba DSL and interpret the results using the Boba Visualizer. The supplemental material contains the Boba specifications of both examples, additional figures, and a video demonstrating all the interactions described below.

6.1 Case Study: Mortgage Analysis

The first case study demonstrates how analysts might quickly arrive at insights provided by summary statistics in prior work, while at the same time gaining a richer understanding of robustness patterns. We

also show that by incorporating uncertainty and model fit checks, Boba surfaces potential issues that prior work might have neglected.

Young *et al.* [59] propose a multimodel analysis approach to gauge whether model estimates are robust to alternative model specifications. Akin to the philosophy of multiverse analysis, the approach takes all combinations of possible control variables in a statistical model. The outputs are multiple summary statistics, including (1) an overall *robustness ratio*, (2) *uncertainty* measures for sampling and modeling variations, and (3) metrics reflecting the *sensitivity* of each variable.

As an example, the authors present a case study on mortgage lending, asking “are female applicants more likely to be approved for a mortgage?” They use a dataset of publicly disclosed loan-level information about mortgage, and fit a linear regression model with mortgage application acceptance rate as the dependent variable and female as one independent variable. In their multimodel analysis, they test different control variables capturing demographic and financial information. The resulting summary statistics indicate that the estimate is not robust to modeling decisions with large end-to-end uncertainty, and two control variables, *married* and *black*, are highly influential. These summary statistics provide a powerful synopsis, but may fail to convey more nuanced patterns in result distributions. The authors manually create additional visualizations to convey interesting trends in data, for instance the estimates follow a multimodal distribution. These visualizations, though necessary to provide a richer understanding of model robustness, are ad-hoc and not included in the software package.

We replicate the analysis in Boba. The Boba DSL specification simply consists of eight placeholder variables, each indicating whether to include a control variable in the model formula. Then, we compile the specification to 256 scripts, run them all, and start the Boba Visualizer.

We first demonstrate that the default views in the Boba Visualizer afford similar insights on uncertainty, robustness, and decision sensitivity. Upon launching the visualizer, we see a decision graph and an overall outcome distribution (Fig. 10). The decision view (Fig. 10a) highlights two sensitive decisions, *black* and *married*. The outcome view (Fig. 10b) shows that the point estimates are highly varied with conflicting implications. The aggregated uncertainty in the outcome view (Fig. 10b, background gray area) has a wide spread, suggesting that the possible outcomes are even more varied when taking both sampling and decision variability into account. These observations agree with the summary metrics in previous work, though Boba uses a different, non-parametric method to quantify decision sensitivity, as well as a different method to aggregate end-to-end uncertainty.

The patterns revealed by ad-hoc visualizations in previous work

We toggle the color-by drop-down menu so that each universe is colored by its model quality metric (Fig. 11b). The large estimates are almost exclusively coming from models with a poor fit. We further verify the model fit quality by picking example universes and examining the model fit view (Fig. 11c). The visual predictive checks confirm issues in model fit, for example the models fail to generate predictions smaller than 3 deaths, while the observed data contains plenty such cases.

Now that we have reasons to be skeptical of the large estimates, the remaining universes still support a small, positive effect. How reliable is the effect? We proceed to the inference view to compare the possible outcomes in the observed multiverse and the expected distribution under the null hypothesis (Fig. 11d). The two distributions are different in terms of mode and shape, yet they are highly overlapping, which suggests the effect is not reliable. The detail plot depicting individual universes (supplemental Fig. 1) further confirms this observation. Out of the entire multiverse, only 3 universes have point estimates outside the 2.5th and 97.5th percentile of the corresponding null distribution.

7 DISCUSSION

Through the process of designing, building, and using Boba, we gain insights into challenges that multiverse analysis poses for software designers and users. We now reflect on these challenges and additional design opportunities for supporting multiverse analysis.

While Boba is intended to reduce the gulf of execution for multiverse analysis, conducting a multiverse analysis still requires statistical expertise. The target users of our current work are experienced researchers and statisticians. Future work might attempt to represent expert statistical knowledge to lower the barriers for less experienced users. One strategy is to represent analysis goals in higher-level abstractions, from which appropriate analysis methods might be synthesized [22]. Another is to guide less experienced users through key decision points and possible alternatives [30], starting from an initial script.

Running all scripts produced by Boba can be computationally expensive due to their sheer number. Boba already leverages parallelism, executing universes across multiple processes. Still, scripts often perform redundant computation and the compiler may produce prohibitively many scripts. Future work should include optimizing multiverse execution, for example caching shared computation across universes, or efficiently exploring decision spaces via adaptive sampling.

As a new programming tool, Boba requires additional support to increase its usability, including code editor plugins, debugging tools, documentation, and community help. In this paper we assess the feasibility of Boba, with the understanding that its usability will need to be subsequently evaluated. Currently, Boba specifications are compiled into scripts in a specific programming language, so users can leverage existing debugging tools for the corresponding language.

However, debugging analysis scripts becomes difficult at the scale of a multiverse because a change that fixes a bug in one script might not fix bugs in others. When we attempt to run a multiverse of Bayesian regression models, for example, models in multiple universes do not converge for a variety of reasons including problems with identifiability and difficulties sampling parameter spaces with complex geometries. These issues are common in Bayesian modeling workflows and must be resolved by adjusting settings, changing priors, or reparameterizing models entirely. At the scale of multiverse analysis, debugging this kind of model fit issue is particularly difficult because existing tools for diagnostics and model checks (e.g., trace and pairs plots) are designed to assess one model at a time. While this points to a need for better debugging and model diagnostic tools in general, it also suggests that these tools must be built with a multiplexing workflow in mind if they are going to facilitate multiverse analysis.

Analysts must take care when reviewing and summarizing multiverse results, as a multiverse is not a set of randomly drawn, independent specifications. In general, the Boba Visualizer avoids techniques that assume universe results are independent and identically distributed. A possible venue for future work is to explicitly account for statistical dependence among universes to remove potential bias. Boba might also do more to aid the communication of results, for example helping to produce reports that communicate multiverse results [12].

Previous approaches to multiverse analysis have largely overlooked the quality of model fit, focusing instead on how to enumerate analysis decisions and display the results from the entire multiverse. We visualize model fit in two ways: we use color to encode the NRMSE from a k -fold cross validation in the outcome view, and use predictive checks to compare observed data with model predictions in the model fit view. Together these views show that a cross-product of analytic decisions can produce many universes with poor model fits, and many of the results that prior studies include in their overviews may not provide a sound base for subsequent inferences. The prevalence of fit issues, which are immediately apparent in the Boba Visualizer, calls into question the idea that a multiverse analysis should consist of a cross-product of all *a-priori* “reasonable” decisions.

We propose adding a step to the multiverse workflow where analysts must distinguish between what seems reasonable *a-priori* vs. *post-hoc*. Boba supports this step in two ways: in the inference view we can use model averaging to produce a weighted combination of universes based on model fit, or we can simply omit universes below a certain model fit threshold chosen by the users. The latter option relies on analysts making a post-hoc subjective decision and might be susceptible to p-hacking. However, one can pre-register a model quality threshold to eliminate this flexibility. Should we enable more elaborate and interactive ways to give users control over pruning? If so, how do we prevent analysts from unintentionally biasing the results? These questions remain future work.

Indeed, a core tension in multiverse analysis is balancing the imperative of transparency with the *need for principled reduction of uncertainty*. Prior work on researcher degrees of freedom in analysis workflows [24] identifies strategies that analysts use to make decisions (see also [7, 29]), including two which are relevant here: *reducing* uncertainty in the analysis process by following systematic procedures, and *suppressing* uncertainty by arbitrarily limiting the space of possible analysis paths. In the context of Boba, design choices which direct the user’s attention toward important information (e.g., highlighting models with good fit and decisions with a large influence on outcomes) and guide the user toward best practices (e.g., visual predictive checks) serve to push the user toward reducing rather than suppressing uncertainty. Allowing users to interact with results as individual dots in the outcome view while showing aggregated uncertainty in the background reduces the amount of information that the user needs to engage with in order to begin exploring universes, while also maintaining a sense of the range of possible outcomes. We believe that guiding users’ attention and workflow based on statistical principles is critical.

8 CONCLUSION

This paper presents Boba, an integrated DSL and visual analysis system for authoring and interpreting multiverse analyses. With the DSL, users annotate their analysis script to insert local variations, from which the compiler synthesizes executable script variants corresponding to all compatible analysis paths. We provide a command line tool for compiling the DSL specification, running the generated scripts, merging the outputs, and invoking the visual analysis system. We contribute a visual analysis system with linked views between analytic decisions and model estimates to facilitate systematic exploration of how decisions impact robustness, along with views for sampling uncertainty and model fit. We also provide facilities for principled pruning of “unreasonable” specifications, and support inference to assess effect reliability. Using Boba, we replicate two existing multiverse studies, gain a rich understanding of how decisions affect results, and find issues around uncertainty and model fit that change what we can reasonably take away from these multiverses. Boba is available as open source software at <https://github.com/uwdata/boba>.

ACKNOWLEDGMENTS

We thank the anonymous reviewers, UW IDL members, Uri Simonsohn, Mike Merrill, Ge Zhang, Pierre Dragicevic, Yvonne Jansen, Matthew Kay, Brian Hall, Abhraneel Sarma, Fanny Chevalier, and Michael Moon for their help. This work was supported by NSF Award 1901386.

REFERENCES

- [1] M. Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, 2016. doi: 10.1038/533452a
- [2] C. G. Begley and L. M. Ellis. Raise standards for preclinical cancer research. *Nature*, 483(7391):531–533, 2012. doi: 10.1038/483531a
- [3] J. Bernard, M. Hutter, H. Reinemuth, H. Pfeifer, C. Bors, and J. Kohlhammer. Visual-interactive preprocessing of multivariate time series data. In *Computer Graphics Forum*, vol. 38, pp. 401–412. Wiley Online Library, 2019. doi: doi.org/10.1111/cgf.13698
- [4] J. Bernard, T. Ruppert, O. Goroll, T. May, and J. Kohlhammer. Visual-interactive preprocessing of time series data. In *Proceedings of SIGRAD 2012*, number 81, pp. 39–48. Linköping University Electronic Press, 2012.
- [5] M. Booshehrian, T. Möller, R. M. Peterman, and T. Munzner. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. In *Computer Graphics Forum*, vol. 31, pp. 1235–1244. Wiley Online Library, 2012. doi: 10.1111/j.1467-8659.2012.03116.x
- [6] R. Border, E. C. Johnson, L. M. Evans, A. Smolen, N. Berley, P. F. Sullivan, and M. C. Keller. No support for historical candidate gene or candidate gene-by-interaction hypotheses for major depression across multiple large samples. *American Journal of Psychiatry*, 176(5):376–387, 2019. doi: 10.1176/appi.ajp.2018.18070881
- [7] N. Boukhelifa, M.-E. Perrin, S. Huron, and J. Eagan. How Data Workers Cope with Uncertainty : A Task Characterisation Study. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2017. doi: 10.1145/3025453.3025738
- [8] J. Cesario, D. J. Johnson, and W. Terrill. Is there evidence of racial disparity in police use of deadly force? analyses of officer-involved fatal shootings in 2015–2016. *Social psychological and personality science*, 10(5):586–595, 2019. doi: 10.1177/1948550618775108
- [9] M. Credé and L. A. Phillips. Revisiting the power pose effect: How robust are the results reported by carney, cuddy, and yap (2010) to data analytic decisions? *Social Psychological and Personality Science*, 8(5):493–499, 2017. doi: 10.1177/1948550617714584
- [10] E. Dejonckheere, E. K. Kalokerinos, B. Bastian, and P. Kuppens. Poor emotion regulation ability mediates the link between depressive symptoms and affective bipolarity. *Cognition and Emotion*, 33(5):1076–1083, 2019. doi: 10.1080/02699931.2018.1524747
- [11] E. Dejonckheere, M. Mestdagh, M. Houben, Y. Erbas, M. Pe, P. Koval, A. Brose, B. Bastian, and P. Kuppens. The bipolarity of affect and depressive symptoms. *Journal of personality and social psychology*, 114(2):323, 2018. doi: 10.1037/pspp0000186
- [12] P. Dragicevic, Y. Jansen, A. Sarma, M. Kay, and F. Chevalier. Increasing the transparency of research papers with explorable multiverse analyses. In *Proc. ACM Human Factors in Computing Systems*, pp. 65:1–65:15, 2019. doi: 10.1145/3290605.3300295
- [13] J. Gassen. A package to explore and document your degrees of freedom. <https://github.com/joachim-gassen/rdfanalysis>, 2019.
- [14] A. Gelman. A Bayesian Formulation of Exploratory Data Analysis and Goodness-of-Fit Testing. *International Statistical Review*, 2003.
- [15] A. Gelman, J. Hwang, and A. Vehtari. Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6):997–1016, 2014. doi: 10.1007/s11222-013-9416-2
- [16] A. Gelman and E. Loken. The garden of forking paths: Why multiple comparisons can be a problem, even when there is no “fishing expedition” or “p-hacking” and the research hypothesis was posited ahead of time. *Department of Statistics, Columbia University*, 2013.
- [17] A. Gelman and E. Loken. The statistical crisis in science. *American Scientist*, 102(6):460, 2014. doi: 10.1511/2014.111.460
- [18] P. J. Guo. *Software tools to facilitate research programming*. PhD thesis, Stanford University, 2012.
- [19] B. Hartmann, L. Yu, A. Allison, Y. Yang, and S. R. Klemmer. Design as exploration: Creating interface alternatives through parallel authoring and runtime tuning. In *Proc. ACM User Interface Software and Technology*, pp. 91–100, 2008. doi: 10.1145/1449715.1449732
- [20] J. Hoffswell, W. Li, and Z. Liu. Techniques for flexible responsive visualization design. In *Proc. ACM Human Factors in Computing Systems*, pp. 1–1, 2020. doi: 10.1145/3313831.3376777
- [21] Z. Jelveh, B. Kogut, and S. Naidu. Political language in economics. *Columbia Business School Research Paper*, (14-57), 2018. doi: 10.2139/ssrn.2535453
- [22] E. Jun, M. Daum, J. Roesch, S. E. Chasins, E. D. Berger, R. Just, and K. Reinecke. Tea: A high-level language and runtime system for automating statistical analysis. *CoRR*, abs/1904.05387, 2019.
- [23] K. Jung, S. Shavitt, M. Viswanathan, and J. M. Hilbe. Female hurricanes are deadlier than male hurricanes. *Proceedings of the National Academy of Sciences*, 111(24):8782–8787, 2014. doi: 10.1073/pnas.1402786111
- [24] A. Kale, M. Kay, and J. Hullman. Decision-making under uncertainty in research synthesis: Designing for the garden of forking paths. In *Proc. ACM Human Factors in Computing Systems*, pp. 202:1–202:14, 2019. doi: 10.1145/3290605.3300432
- [25] M. Kay, G. L. Nelson, and E. B. Hekler. Researcher-centered design of statistics: Why bayesian statistics better fit the culture and incentives of HCI. In *Proc. ACM Human Factors in Computing Systems*, pp. 4521–4532, 2016. doi: 10.1145/2858036.2858465
- [26] M. B. Kery, A. Horvath, and B. Myers. Variolite: Supporting exploratory programming by data scientists. In *Proc. ACM Human Factors in Computing Systems*, pp. 1265–1276, 2017. doi: 10.1145/3025453.3025626
- [27] M. B. Kery and B. A. Myers. Interactions for untangling messy history in a computational notebook. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 147–155, 2018. doi: 10.1109/VLHCC.2018.8506576
- [28] Q. Li, M. R. Morris, A. Fourny, K. Larson, and K. Reinecke. The impact of web browser reader views on reading speed and user experience. In *Proc. ACM Human Factors in Computing Systems*, pp. 524:1–524:12, 2019. doi: 10.1145/3290605.3300754
- [29] R. Lipshitz and O. Strauss. Coping with Uncertainty: A Naturalistic Decision-Making Analysis. *Organizational Behavior and Human Decision Processes*, 69(2):149–163, 1997. doi: 10.1006/obhd.1997.2679
- [30] Y. Liu, T. Althoff, and J. Heer. Paths explored, paths omitted, paths obscured: Decision points & selective reporting in end-to-end data analysis. In *Proc. ACM Human Factors in Computing Systems*, pp. 406:1–406:14, 2020. doi: 10.1145/3313831.3376533
- [31] A. Lunzer. Towards the subjunctive interface: General support for parameter exploration by overlaying alternative application states. In *Late Breaking Hot Topics, IEEE Visualization*, vol. 98, pp. 45–48, 1998.
- [32] A. Lunzer. Choice and comparison where the user wants them: Subjunctive interfaces for computer-supported exploration. In *Proceedings of INTERACT*, pp. 474–482, 1999.
- [33] S. McConnell. *Code complete*. Microsoft Press, 2 ed., 2004.
- [34] L. D. Nelson, J. Simmons, and U. Simonsohn. Psychology’s renaissance. *Annual Review of Psychology*, 69(1):511–534, 2018. doi: 10.1146/annurev-psych-122216-011836
- [35] Open Science Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251), 2015. doi: 10.1126/science.aac4716
- [36] A. Orben and A. K. Przybylski. The association between adolescent well-being and digital technology use. *Nature Human Behaviour*, 3(2):173, 2019.
- [37] C. J. Patel, B. Burford, and J. P. A. Ioannidis. Assessment of vibration of effects due to model specification can demonstrate the instability of observational associations. *Journal of Clinical Epidemiology*, 68(9):1046–1058, 2015. doi: 10.1016/j.jclinepi.2015.05.029
- [38] C. Pettitt. Dage. <https://github.com/dagreis/dage>, 2015.
- [39] C. Phelan, J. Hullman, M. Kay, and P. Resnick. Some prior(s) experience necessary: Templates for getting started with bayesian analysis. In *Proc. ACM Human Factors in Computing Systems*, pp. 479:1–479:12, 2019. doi: 10.1145/3290605.3300709
- [40] G. J. Poarch, J. Vanhove, and R. Berthele. The effect of bidialectalism on executive function. *International Journal of Bilingualism*, 23(2):612–628, 2019. doi: 10.1177/1367006918763132
- [41] F. Prinz, T. Schlange, and K. Asadullah. Believe it or not: How much can we rely on published data on potential drug targets? *Nature Reviews Drug Discovery*, 10(9):712, 2011. doi: 10.1038/nrd3439-c1
- [42] J. R. Rae, S. Gülgöz, L. Durwood, M. DeMeules, R. Lowe, G. Lindquist, and K. R. Olson. Predicting early-childhood gender transitions. *Psychological Science*, 2019. doi: 10.1177/0956797619830649
- [43] J. M. Rohrer, B. Egloff, and S. C. Schmukle. Probing birth-order effects on narrow traits using specification-curve analysis. *Psychological Science*, 28(12):1821–1832, 2017.
- [44] M. Rubin. Do p values lose their meaning in exploratory analyses? it depends how you define the familywise error rate. *Review of General Psychology*, 21(3):269–275, 2017. doi: 10.1037/gpr0000123
- [45] A. Rule, A. Tabard, and J. D. Hollan. Exploration and explanation in computational notebooks. In *Proc. ACM Human Factors in Computing Systems*, p. 32, 2018. doi: 10.1145/3173574.3173606

- [46] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014. doi: 10.1109/TVCG.2014.2346321
- [47] R. Silberzahn, E. L. Uhlmann, D. Martin, P. Anselmi, F. Aust, E. C. Awtry, Štěpán Bahník, F. Bai, C. Bannard, E. Bonnier, R. Carlsson, F. Cheung, G. Christensen, R. Clay, M. A. Craig, A. D. Rosa, L. Dam, M. H. Evans, I. F. Cervantes, N. Fong, M. Gamez-Djokic, A. Glenz, S. Gordon-McKeon, T. J. Heaton, K. Hederos, M. Heene, A. J. H. Mohr, F. Högden, K. Hui, M. Johannesson, J. Kalodimos, E. Kaszubowski, D. M. Kennedy, R. Lei, T. A. Lindsay, S. Liverani, C. R. Madan, D. Molden, E. Molleman, R. D. Morey, L. B. Mulder, B. R. Nijstad, N. G. Pope, B. Pope, J. M. Prenoveau, F. Rink, E. Robusto, H. Roderique, A. Sandberg, E. Schlüter, F. D. Schönbrodt, M. F. Sherman, S. A. Sommer, K. Sotak, S. Spain, C. Spörlein, T. Stafford, L. Stefanutti, S. Tauber, J. Ullrich, M. Vianello, E.-J. Wagenmakers, M. Witkowiak, S. Yoon, and B. A. Nosek. Many analysts, one data set: Making transparent how variations in analytic choices affect results. *Advances in Methods and Practices in Psychological Science*, 1(3):337–356, 2018. doi: 10.1177/2515245917747646
- [48] J. P. Simmons, L. D. Nelson, and U. Simonsohn. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11):1359–1366, 2011. doi: 10.1177/0956797611417632
- [49] U. Simonsohn, J. P. Simmons, and L. D. Nelson. Specification curve: Descriptive and inferential statistics on all reasonable specifications. *Available at SSRN 2694998*, 2015. doi: 10.2139/ssrn.2694998
- [50] S. Steegen, F. Tuerlinckx, A. Gelman, and W. Vanpaemel. Increasing transparency through a multiverse analysis. *Perspectives on Psychological Science*, 11(5):702–712, 2016. doi: 10.1177/1745691616658637
- [51] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981. doi: 10.1109/TSMC.1981.4308636
- [52] M. Terry, E. D. Mynatt, K. Nakakoji, and Y. Yamamoto. Variation in element and action: Supporting simultaneous development of alternative solutions. In *Proc. ACM Human Factors in Computing Systems*, pp. 711–718, 2004. doi: 10.1145/985692.985782
- [53] E. R. Tufte, N. H. Goeler, and R. Benson. *Envisioning information*. Graphics Press, 1990.
- [54] W. Vanpaemel, S. Steegen, F. Tuerlinckx, and A. Gelman. Multiverse analysis. <https://osf.io/zj68b/>, 2018.
- [55] A. Vehtari, A. Gelman, and J. Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and Estimating out-of-sample pointwise predictive accuracy using posterior simulations. *J Stat Comput*, 27(5):1413–1432, 2017. doi: 10.1007/s11222-016-9696-4
- [56] J. M. Wicherts, C. L. S. Veldkamp, H. E. M. Augusteijn, M. Bakker, R. C. M. van Aert, and M. A. L. M. van Assen. Degrees of freedom in planning, running, analyzing, and reporting psychological studies: A checklist to avoid p-hacking. *Frontiers in Psychology*, 7:1832, 2016. doi: 10.3389/fpsyg.2016.01832
- [57] L. Wilkinson. Dot plots. *The American Statistician*, 53(3):276–281, 1999.
- [58] Y. Yao, A. Vehtari, D. Simpson, and A. Gelman. Using stacking to average bayesian predictive distributions (with discussion). *Bayesian Analysis*, 13(3):917–1007, 2018. doi: 10.1214/17-BA1091
- [59] C. Young and K. Holsteen. Model uncertainty and robustness: A computational framework for multimodel analysis. *Sociological Methods & Research*, 46(1):3–40, 2017. doi: 10.1177/0049124115610347
- [60] E. Zraggen, Z. Zhao, R. Zeleznik, and T. Kraska. Investigating the effect of the multiple comparisons problem in visual analysis. In *Proc. ACM Human Factors in Computing Systems*, pp. 479:1–479:12, 2018. doi: 10.1145/3173574.3174053