EVM: Incorporating Model Checking into Exploratory Visual Analysis

Alex Kale, Ziyang Guo, Xiao Li Qiao, Jeffrey Heer, Jessica Hullman

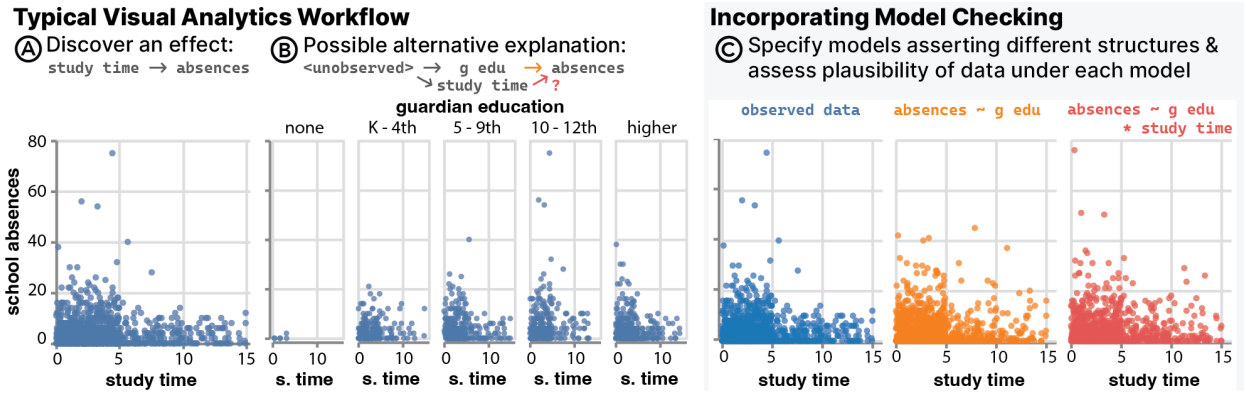

Fig. 1: Model checks modify the typical visual analytics workflow by enabling users to assess the plausibility of interpretations of discovered patterns. Ⓐ The analyst discovers that accounting for time spent studying appears to help explain student absences. Ⓑ The analyst facets this view by the highest level of education achieved by each student's guardian. He wonders if study time is predictive of absences after accounting for guardian education. Ⓒ The analyst specifies models asserting that absences are explained by either: guardian education alone (orange); or guardian education and study time (red). Seeing that predictions from the second model do a better job of capturing the largest numbers of absences, he concludes that both guardian education and study time are important explanatory variables.

Abstract— Visual analytics (VA) tools support data exploration by helping analysts quickly and iteratively generate views of data which reveal interesting patterns. However, these tools seldom enable explicit checks of the resulting interpretations of data—e.g., whether patterns can be accounted for by a model that implies a particular structure in the relationships between variables. We present EVM, a data exploration tool that enables users to express and check provisional interpretations of data in the form of statistical models. EVM integrates support for visualization-based model checks by rendering distributions of model predictions alongside user-generated views of data. In a user study with data scientists practicing in the private and public sector, we evaluate how model checks influence analysts' thinking during data exploration. Our analysis characterizes how participants use model checks to scrutinize expectations about data generating process and surfaces further opportunities to scaffold model exploration in VA tools.

Index Terms—Visualization, model checks, exploratory analysis

1 Introduction

Data analysts use exploratory visual analysis (EVA) tools such as Tableau to check their understanding of data, discover patterns, and seek potential explanations for those patterns. For example, imagine an analyst, Juan, contracted to investigate what factors contribute to school absences in a local school district. During exploration, Juan discovers an interesting looking pattern (Fig. 1 Ⓐ) where absences are associated with the number of hours students spend studying. Juan wonders what could explain this pattern. He begins faceting by other variables in the dataset provided by the school and notices that the highest level of education achieved by each student's guardian helps to explain large numbers of absences among students who study less (Fig. 1 Ⓑ). If Juan wants to scrutinize his interpretation of this pattern further, and he is comfortable with statistical programming, he might switch to a programming language like R or Python to specify and compare the predictions of models that do and do not assert this relationship. This kind of provisional "model checking" [24] could help Juan rule out some interpretations as less likely, or explore how variations on his interpretation might better capture what he sees.

Given their broad appeal to non-programmers, many users of visual analysis (VA) tools may not be comfortable shifting to statistical tools to do preliminary model development and checking. This points to a gap in current design approaches for EVA tools: they offer little support for helping analysts reason more explicitly about their provisional interpretations. Recent work [24] critiques the standard approach to designing visual analysis tools as implying a misconception that data exploration is "model free", in the sense that we rely primarily on visualizations for surfacing patterns and hypotheses early in analysis and rely primarily on modeling for confirmatory statistical inference late in analysis. A consequence of this "model free" account of data exploration is siloed tools for exploratory and confirmatory analysis, leaving EVA users under-equipped to resolve ambiguity about the underlying data generating process.

Imagine instead that an EVA interface enabled Juan to (1) fit a series of models asserting various plausible assumptions and structures about

- *Alex Kale is with the University of Chicago. E-mail: kalea@uchicago.edu.*
- *Ziyang Guo is with Northwestern University. E-mail: ziyangguo2027@u.northwestern.edu.*
- *Xiao Li Qiao is with Northwestern University. E-mail: emqiao@gmail.com.*
- *Jeffrey Heer is with the University of Washington. E-mail: jheer@uw.edu.*
- *Jessica Hullman is with the Northwestern University. E-mail: jhullman@northwestern.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.xxxx/TVCG.201x.xxxxxxx

the data and (2) visualize their predictions in comparison to observed data (Fig. 1 ©). Explicitly comparing models encoding different data interpretations can afford better-calibrated confidence in claims that Juan might make about the data generating process. Without such an explicit representation of expectations, analysts like Juan must *imagine what it could look like if*, e.g., only guardian education influences student absences and whether the observed data are plausible under that model, imagined counterfactuals that some recent work (e.g., [28]) suggests are difficult for many to accurately incorporate into visual inferences. At the same time, once an EVA tool makes it easy to generate model predictions for comparison with data, it is possible that the added formality provides a sense of false assurance to analysts, exacerbating risks of post-hoc inferences or overfit explanatory models, similar to p-hacking [46] or model selection via R-squared [67].

We explore the promise and pitfalls of explicit modeling support integrated in a visual data exploration tool via Exploratory Visual Modeling (`EVM`), an open-source prototype EVA application. `EVM` enables users to quickly specify statistical models representing plausible explanations of discovered patterns in data, and compare predictions from these models to the observed data. The key role of visualization in `EVM` is *showing discrepancies between patterns in data and expectations*, a design pattern dubbed **"model checks"** by Hullman and Gelman [24].

`EVM` contributes a drag-and-drop interface for chart construction (c.f., Polaris [47]) with model checking features including: an interactive "model bar" that enables analysts to flexibly specify regression models, a back-end layer that fits user-specified models and samples appropriate predictive distributions to show in the browser, and carefully designed defaults for model check visualizations that juxtapose user-generated views of data with model predictions. This interface enables users to quickly specify and check a wide variety of regression models against data on-the-fly, just as conventional EVA tools enable quick chart construction. Additionally, we present an evaluation of how 12 data scientists practicing in the private and public sectors use model checks when they are incorporated into EVA workflows. We characterize how model checks change participants' exploratory data analysis behavior compared to a baseline condition where they use a simple VA tool without model checking functionality, finding that model checks evoke different analysis behaviors depending on the user's previous experience with modeling. We discuss new design requirements for model checks in EVA, highlighting cognitive pitfalls of model-based data exploration, possible roles for model recommendation, and opportunities around new programming tools for visual modeling.

2 Related Work

We present relevant literature on graphical statistical inference and interactive model selection to contextualize our contributions.

2.1 Graphical statistical inference

Graphical statistical inference refers to visual methods for judging how well a statistical model describes observed patterns in data, which we refer to as *"model checks"* following Hullman and Gelman [24]. Tukey motivated visualizing model residuals to inspect where a provisional model might be wrong during exploratory data analysis [53]. Similarly, common model diagnostic tools such as QQ-plots [34] create visual tests that a model's assumptions are satisfied.

The best-known visualization formulation of model checks may be the visualization lineup protocol [6, 7, 61], in which many plots of simulated data are generated from a "null model" (i.e., representing a null hypothesis) and an impartial observer is asked to pick out a plot of the real data among the set of "null plots." The lineup procedure has been analogized to a formal statistical test, and shown to have equivalent power or even better power than a conventional statistical test in some scenarios [35]. However, the strict analogy between the lineup and a statistical test can be hard to ensure because creating null plots is a non-trivial challenge [54, 55], and the need for impartial observers is impractical. Others have proposed alternative analogies for graphical inference, such as Bayesian cognition, as a means of guiding visualization design and research [24, 30, 31, 65]. Inspired by Bayesian workflows, where visualizations are the primary means by which models are interrogated [15, 19], Hullman and Gelman [24] discuss how model check visualizations can have value for helping analysts understand *for which cases a model is wrong about the data*, without evoking an analogy to a statistical test or attempting to provide guarantees about error rates. In this approach, rejecting a model is not considered a win so much as realizing from a visualized model check what features of observed data remain yet to be explained.

While most interpretations of visualizations might be likened to checking an implicit model representing the viewer's expectations about the data [16, 17], attempts to design graphical user interfaces for visual analysis ('VA tools' hereafter) that make it easy to check provisional models have been much less prevalent than design approaches that emphasize interaction with observed data [24]. For example, while Tableau Software supports very simple regression modeling and construction of uncertainty intervals, at the time of this writing plotting residuals requires multiple data transformation and visualization steps. In part because of this lack of integration of model checking with VA tools, analysts may not always scrutinize patterns discovered using such tools to ask how exactly they might arise. Exceptions include early research systems developed by and for statisticians [4, 56], NorthStar created for predictive modeling [33], and recent research systems developed to study novel interfaces for eliciting analysts' expectations via natural language [9] and sketching [32]. In contrast to these efforts, we developed `EVM` to study how model check visualizations might benefit the broader populations of analysts that tools like Tableau target, assuming neither that our users would be statisticians nor that realizing model checking necessarily requires a new elicitation medium.

2.2 Visually-aided model selection

Descriptive accounts of exploratory visual analysis (EVA, e.g., [2]) acknowledge that it often alternates between open-ended tasks (e.g., flipping through filters looking for something interesting to explore a space of theories or models, a.k.a, abduction proper [13, 40]) and more focused exploration (e.g., trying to formulate and validate a hypothesis). Recent work in computer science [43, 68, 70] analogizing EVA to a multiple comparisons problem emphasizes what Tukey [50] referred to as "rough confirmatory analysis." In this stage, visual analysis plays a classification role in helping an analyst distinguish between signals that are so apparent that statistical modeling is not needed, versus where noise and confounding are so great that confirming perceived patterns is hopeless. `EVM` is designed specifically to support this rough confirmatory stage, in which analysts rely on their eyes to make often difficult judgments about signal versus noise ratios.

Tukey stressed multiplicity as a key issue in this intermediate stage of analysis, which proceeds a stage of initial exploration in which probability is not of interest, and precedes confirmatory analysis. For example, an analyst should be wary of "How many things might have been looked at? How many had a real chance to be looked at? How should the multiplicity decided upon, in answer to these questions, affect the resulting confidence sets and significance levels?" [51]. Whereas previous research [8, 9, 32] attempted to avoid risks of post-hoc inference by forcing analysts to specify models before seeing the results of queries, and suggested further mitigations through automated adjustment of test statistics, we opted to focus `EVM` on making support for model check visualizations as seamless as possible, without even providing numerical model summaries like p-values for users to exploit. We built `EVM` to investigate how VA users rely on unconstrained visual checking to search a space of plausible models, rather than presupposing a hypothesis testing framework where elicited models necessarily reflect a user's best-guess expectation. The multiple comparisons problem has led to valuable suggestions for EVA systems like holdout sets [68], which would be natural to support in future iterations of `EVM`.

Other research prototypes [21, 57, 58, 66] have been developed to support causal inference by representing user-defined queries in terms of directed acyclic graphs. These tools integrate data mining approaches into VA tools with the intention of helping users explore the plausibility that various causal structures explain their data. Although `EVM` has similar goals insofar as we aim to promote scrutiny about interpretations of data, we avoid automated modeling approaches in `EVM` based on the

design philosophy that visual model checking will be most meaningful when models originate from users' expectations.

3 Design Requirements

We designed and implemented EVM to support model checking during exploratory visual analysis. We describe the design rationale for EVM, highlighting where we envision model checks adding value to visual data exploration workflows.

Promote generative thinking. Analysts derive meaning from patterns they discover during visual data exploration based on how these patterns match or contradict their expectations. To evaluate possible patterns and expectations, analysts must answer the question, *"What would a new data sample look like given my provisional model?"* However, visual data exploration tools usually do not support generation of new hypothetical samples, leaving analysts to imagine how these might look. A primary design hypothesis behind EVM is that integrating model checking into visual data exploration will promote more explicit consideration of how data might have been generated.

Pattern-seeking, not data mining. Visual data exploration tools lend themselves to false discoveries [68] in part because they make it easy to view data but hard to check interpretations or connect visual patterns to expectations. Analysts may be distracted or mislead by small visual details on charts unless they seek these details deliberately based on questions about their data. For this reason, EVM avoids serving up comparisons or views of data that the user has not requested.

Eliciting regression models. In order to check users' provisional data interpretations, we require a computational representation of the assumptions and structure implied by an explanation. Part of the design philosophy behind model check visualizations is that regression models can provide a shared abstraction for humans and machines, and that models need not represent either a user's best-guess of the data generating process or a hypothesis test in order for visualized model predictions to provide a useful reference for making sense of data [24]. For this reason, EVM enables users to specify regression models using R syntax for model formulae [42, 44, 62] through a component called the model bar (see Section 4.2). Given that such regression formulae are shown to be a successful "interface" via their frequent use in the sciences, we assume they will be suitable for users with a wide range of backgrounds and experiences with statistics.

Model expansion workflow. One perspective in statistical theory suggests that tools should promote a modeling workflow where analysts consider and check *incremental* changes to models reflecting their provisional beliefs about data [15]. Although there is some contention that such forward stepwise selection of predictors could lead to biased parameter estimates arising from multiple comparisons and overfitting [22, 38], this is a risk primarily when the goal is hypothesis testing. In contrast, EVM is intended for visually scrutinizing data interpretations in order to explore a space of plausible models, not for hypothesis testing. While analysts' difficulty in clearly separating exploratory and confirmatory analysis can contribute to overfit explanations [18], and may also be a risk using EVM, a goal of EVM is to integrate better tools for preliminary winnowing of bad interpretations of visual findings that might otherwise go unchecked. To this end, a model expansion workflow helps analysts work up to complex models in terms of simpler models that assert a subset of the same structure. EVM facilitates this sort of cumulative assessment of what is and isn't helpful to model.

Decouple models from visualizations. It can be tempting to draw analogies between statistical models and visualizations (e.g., that Fig. 1 Ⓑ necessarily implies a model assuming an interaction between study time and guardian education), due to representational similarities at the software abstraction level between models [42, 44, 62] and visualizations [63]. However, in developing EVM we quickly realized that a more appropriate way to describe the relationship between models and visualizations is many-to-many: a model can imply multiple visual checks and a visualization can map to multiple model specifications [48]. Hence EVM decouples model specification from chart specification, allowing visualizations to show variables that are not predictors in a model, and models to learn relationships which are not directly visualized. This enables use cases like Figure 1 Ⓒ where model check visualizations can help users reason about the higher-dimensional patterns behind problematically low-dimensional [60] visualizations.

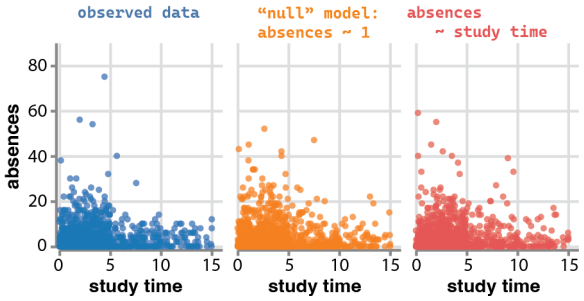

Fig. 2: Model check showing what the data would look like if absences was not associated with study_time (orange) vs if it was (red). Predictions from the two models look very similar.

Graft model outputs to user-specified visualizations. Model checking is inherently visual, but few principles exist for prescribing how to show model outputs alongside observed data [60], including how to sample from the model output and how to facilitate visual comparisons. We strove for smart defaults such that EVM juxtaposes model predictions with any user-generated visualization by simply adding an adjacent subplot for each fitted model. EVM always facets subplots of the model predictions in a layout that preserves the ability to *compare observed and predicted outcomes on a common scale*. We show model predictions as animated hypothetical outcome plots or HOPs [25] since this uncertainty visualization technique is helpful for showing reference distributions [27] and can be applied to any user-generated chart. When the analyst chooses to check a model that doesn't make sense (e.g., using a model that assumes discrete outcomes on continuous data), the resulting model check visualization sometimes becomes ill formed, signaling to the analyst that something has gone wrong. After discovering these failure modes during informal testing, we decided not to prevent them based on their value for detecting flawed models.

4 EVM: Exploratory Visual Modeling

We implemented EVM as a single-page web application, where users can generate views of data and models to check, connected to an R server that fits models and extracts predictions from them. The reader can interact with the prototype at https://mucollective.github.io/evm/ and find the project repository at https://github.com/MUCollective/evm/.

4.1 Usage scenario

Consider how Juan might interact with EVM to investigate factors that affect student absences (see Section 1). Initially, Juan might use EVM's drag-and-drop interface (Fig. 3 Ⓐ) to construct a series of bar charts, strip plots, and scatterplots examining each predictor variable in the dataset and its relationship with the outcome of interest, absences. This initial tour of the data would reveal interesting potential relationships such as associations between student absences and variables like weekly hours of study_time or level of guardian education (g_edu).

In order to scrutinize these relationships, Juan uses EVM's model bar (Fig. 3 Ⓓ) to express provisional interpretations to check against the data. First, Juan *evaluates the plausibility of the interpretation* that study_time helps explain absences. To do this, he specifies two models to check against the data: one "null" model asserts no relationship between study_time and absences; the other model asserts that study_time is predictive of absences (Fig. 2). In both models, Juan assumes that the empirical distribution of absences can be approximated by a negative binomial distribution because absences are an overdispersed count outcome. By comparing the data distribution to predictions from both of these models in a model check, Juan becomes less convinced that study_time is an important predictor.

Juan wonders if study_time might only seem predictive because of correlation with other explanatory variables. Returning to a visual exploration workflow, he starts faceting the relationship between absences and study_time by other factors. Juan discovers that the pattern looks stronger, with higher numbers of absences overall, when

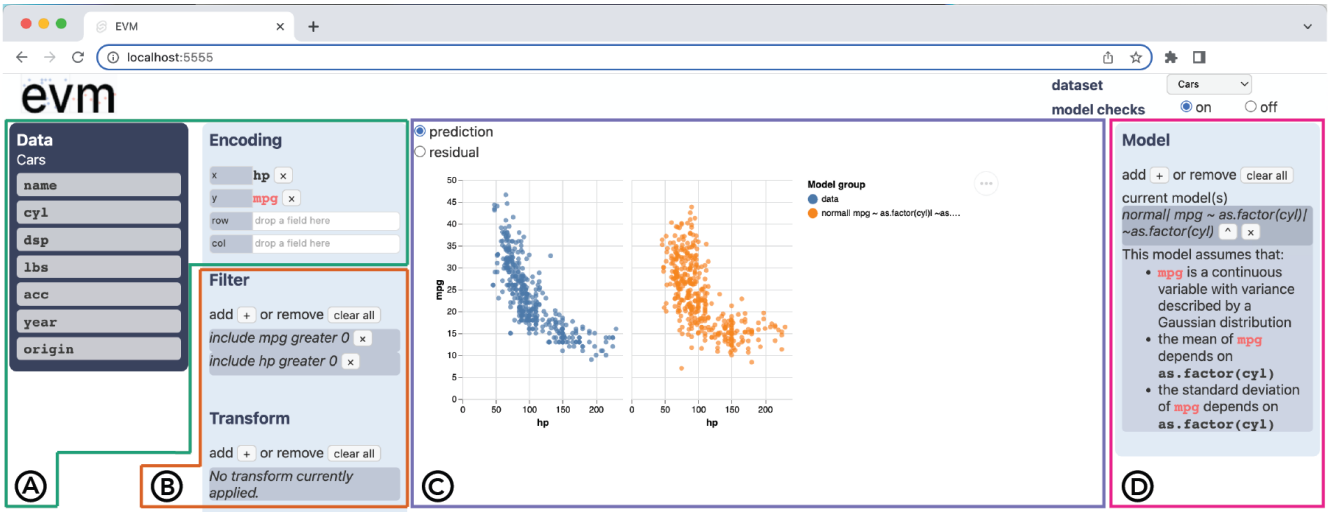

Fig. 3: A screenshot of EVM, annotated to show different components described in Section 4.2: Ⓐ Shelf construction; Ⓑ Filters & transformations; Ⓒ Chart panel; and Ⓓ Model bar. This model check assumes that the number of miles per gallon a car gets mpg is normally distributed with mean and variance both influenced by the number of cylinders in a car's engine cyl. Plotting horsepower hp against mpg results in a model check of how well the trade-off between hp and mpg is explained by the variable cyl, which is not shown directly in this view.

students have guardians with higher levels of education (Fig. 1 Ⓑ). Perhaps both study_time and g_edu reflect some unobserved factor such as a family's socioeconomic status. Juan wonders if it is still important to consider study_time after accounting for g_edu.

To better *understand correlated predictors*, Juan sets up a series of models to check against the data. He starts with a model asserting that only g_edu influences absences. EVM juxtaposes predictions from this model against Juan's scatterplot of the relationship between study_time and absences (Fig. 1 Ⓒ, left vs middle), revealing that the pattern in the data can be roughly accounted for by a model that asserts only an effect of g_edu. Juan tries adding another model asserting that both g_edu and study_time are predictive of absences to investigate whether using study_time as a predictor improves the model fit at all. The resulting model check shows that a model asserting influences of both g_edu and study_time does a better job of predicting the case with the largest number of absences (Fig. 1 Ⓒ, left vs middle vs right). Juan's interpretation of this model check depends on whether he thinks of this case as an outlier or the tail of the absences distribution. EVM's model checks help Juan arrive at the conclusion that, although g_edu and study_time are correlated, these predictors likely contain some non-overlapping information, something that he could not ascertain from exploratory visual analysis alone.

4.2 Overview of functionality

The basic visual analytics functionality of EVM resembles that of systems like Tableau Software. We based this aspect of EVM's design on PoleStar [64], a research prototype developed to mimic the interaction model of Tableau for the purpose of user testing. To generate visualizations in EVM, users rely on a **shelf construction** interface (Fig. 3 Ⓐ) where they drag "pills" representing variables in a dataset onto "shelves" representing x, y, row, and column encodings (in the sense of the grammar of graphics [63]). The resulting visualizations appear in the **chart panel** at the center the display (Fig. 3 Ⓒ). Smart defaults determine the chart types that render in the chart panel depending on the data types of the variables on the x and y encodings:

- *Bar charts* show univariate distributions for discrete variables.
- *Strip plots* show univariate distributions for continuous variables, and bivariate distributions for continuous vs. discrete variables.
- *Scatterplots* show bivariate distributions for continuous variables.
- *Heatmaps* show bivariate distributions for discrete variables.

EVM facets any of these chart types into a *trellis plot* [3, 52], consisting of multiple subplots arranged along the vertical and/or horizontal span of the chart panel, when the user defines a row and/or column encoding. In addition to chart specification, users of EVM can add **filters and transformations** by clicking the + icons (Fig. 3 Ⓑ). When adding filters, the user selects a variable based on which they will either include or exclude values less than (or equal to), greater than (or equal to), or (not) equal to a criterion. When adding transforms, the user selects a variable to which they can apply either a log odds or log transformation, two options provided in EVM because they are used to handle bounded distributions in logit normal and log normal models, respectively (see below). One can remove filters and transforms by clicking the X icons next to each filter or transform or by clicking remove all. For simplicity, EVM always applies filters before transforms and applies both in the order they are specified.

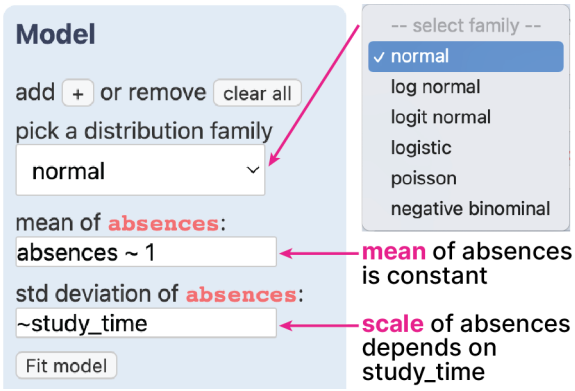

Fig. 4: The model bar specifying a model check where absences are assumed to be normally distributed with a constant mean and variance dependent on study_time.

The primary innovation of EVM is to add model-checking functionality to this style of visual analytics system. Users can specify models to check against visualizations using the **model bar**, an interface element not found in prior exploratory visualization tools (Fig. 3 Ⓓ). The model bar employs a similar design pattern to the filter and transform interfaces, in that users can add or remove models from the current set of candidate models. When the user chooses to add a model to the model bar, they select a *distribution family* from the following options: *normal (a.k.a. Gaussian)* for unbounded continuous outcomes; *log normal* for right-skewed continuous outcomes with a lower bound at zero; *logit normal* for continuous outcomes bounded at zero and one; *logistic* for binary outcomes; *Poisson* for count outcomes; and *negative binomial* for overdispersed count outcomes. We selected these families to cover common distributions of outcome variables without redundancy (see Appendix in Supplemental Materials). Based on the model family chosen by the user, EVM elicits a *model specification* in

Wilkinson-Pinheiro-Bates syntax [42,62], providing separate text boxes for location and scale sub-models (Fig. 4) when the user selects a distribution family with an explicit scale parameter. EVM parses these model specifications into assumptions and asserted structures and displays these as a list of natural language descriptions in the model bar.

4.3 Implementation

We implemented EVM as a Svelte application that runs in the browser and connects to an OpenCPU [41] deployment, which fits and processes models in R. We use Vega [45] to generate visualizations in EVM. We also wrote a custom R package named `modelcheck`, which contains functions to run a specified model in the selected family, add model predictions to the input data frame, calculate residuals, and merge together outputs from other operations without duplicating entries. We rely on a combination of base R and `tidyverse` [59] for data wrangling, as well as `gamlss` [44] models which fit quickly enough to keep latency suitably low for an interactive system.

In addition to fitting models, `modelcheck` handles two critical steps for uncertainty propagation that can pose challenges in modeling workflows: (1) sampling learned parameter values from the model to construct a predictive distribution and (2) back-transforming these predictions into the units of the data that users pass into the model bar. This guarantees that model check visualizations compare data and model outputs on the same scale. For example, predictions from log and logit normal model families are back-transformed using exponential and logistic inverse-link functions, respectively. This implementation facilitates quick model iteration in EVM's user interface; typically users would need to write analysis scripts to achieve similar model checks.

5 User Study

We designed EVM for visual analytics (VA) practitioners whose backgrounds in statistics vary, from having taken an introductory statistics course at one time to fluidity with statistical tools. In order to assess whether incorporating model checks into an interface like EVM would benefit at least some people currently using VA tools, we ran a user study targeting practicing data workers who were familiar with statistical models but who, unlike statisticians, typically might not incorporate modeling into exploratory workflows. Given the exploratory nature of our study, we conducted think-alouds followed by open-ended conversational interviews with users, rather than attempting a controlled experiment targeting anticipated benefits of model checking. To investigate how model checking changes VA workflows, we characterized users' analysis behaviors at baseline using a simple VA tool without model checking functionality as well as using model checks in EVM.

5.1 Participants

We recruited practicing data analysts ($n = 12$) to use EVM, drawing primarily from our professional network via Twitter and email. To be eligible, participants needed to (1) work with data regularly, (2) be familiar with visual analytics tools like Tableau, and (3) have some previous experience using regression models. We reasoned that recruiting real data analysts would provide greater ecological validity to any conclusions we draw. Of our 12 participants, 2 were academic researchers in computer science, 5 were data scientists working on business intelligence, 2 were data analysts working in healthcare, and 3 performed data-intensive work in government agencies or non-profits. This sample emphasized practices of non-statisticians using VA tools.

5.2 Datasets

We provided each participant with two cleaned datasets to explore using EVM, without and with model checks enabled. Our study design required that these datasets were realistic without requiring specialized domain knowledge to explore, contained non-trivial structure for participants to discover, and were roughly equal in size. We used two real datasets on forest fires [11,14] and student absences [12,14] in Portugal, which met the first two requirements. To limit potential confounding effects of dataset size on analysis behavior, we matched the number of variables available for exploration by dropping selected variables until there were ten variables per dataset, and we matched the number of records in the datasets by dropping rows at random from the larger dataset (student absences) until both datasets contained 517 observations. We also used aggregation and sampling procedures on a few variables to match the number of variables in each dataset that were discrete versus continuous (see Supplemental Materials).

5.3 Interview protocol

Our interview sessions were structured as a pre-post study design, where participants used a prototype visualization system with and without model check functionality, followed by a debriefing interview. Each session spanned 90 minutes total, split into three 30-minute sections: *think-aloud baseline*, *think-aloud with model checks*, and *debrief*. We focus here on the procedure and goals of our evaluation; see Supplemental Materials for full details of the user study protocol.

Think-aloud baseline. In the first 5 minutes, we introduced participants to a version of EVM where *model checks were not enabled*. This entailed a demonstration of chart specification, data filtering, and transforms. In the next 25 minutes, participants explored one dataset in a think-aloud protocol using EVM without model checks enabled. This step provided a baseline for characterizing data exploration behavior with a conventional VA tool at an individual level.

Think-aloud with model checks. The interviewer enabled model checking and spent 5 minutes using the cars dataset [14, 29] to demonstrate potential model checking use cases investigating the plausibility of assumed relationships and correlations between predictors (see Section 4.1). For example, to show how model checks can elucidate correlations between predictors, we showed the scenario in Figure 3 ©. This was followed by another 25 minutes of think-aloud data exploration on a second dataset, this time using EVM *with model checks enabled*. This second round of data exploration served as an intervention condition, assigned within-subjects to assess changes from baseline.

During think-aloud sessions, participants were instructed to spend 25 minutes exploring one of two datasets looking for potential influences on either area burned in forest fires or student absences. We asked participants to tell us about any observations or patterns they felt were worth having a colleague follow up on. The interviewer spoke only to prompt participants to say what they were thinking, to answer direct questions, and occasionally to help participants get unstuck if they encountered a bug or confusing edge case. Some participants, especially those who were less familiar with implementing regression models in R, needed clarification about model notation and underlying assumptions. The interviewer answered these questions. When participants hesitated or got confused about model specification, the interviewer made a note and asked about these instances later in the interview.

In the two think-alouds, the pairing of datasets with interface conditions (i.e., model checks disabled vs. enabled) was counterbalanced across subjects, but the order of interface conditions was not. Our rationale was to control for artifacts of exploring a particular dataset while also avoiding a complex experimental design. Counterbalancing the order of interface conditions would have told us whether users seemed to explore data differently in a typical VA tool after exploring data with model checks—a learning effect that is not the focus of our evaluation.

Debrief The last 30 minutes of each interview involved a semi-structured conversational interview with the participant about EVM. The semi-structured interview followed an interview guide, which consisted of the following lines of questioning:

- **Utility of model checks.** In what ways (if any) did you use model checks to help you think about the dataset? What specific visual cues on the resulting chart (if any) were interesting or helpful?
- **Generative thinking.** Did you find yourself thinking about the data generating process, or the underlying relationships that might explain the patterns you saw in the data? What kinds of assumptions (if any) did you make about the dataset? Did using model checks make these assumptions more salient or concrete?
- **Expressiveness and usability.** Did you have any difficulty using the model bar to express and check provisional interpretations of data? What if anything made it hard to use? What if anything do you think would make this kind of functionality easier to use?

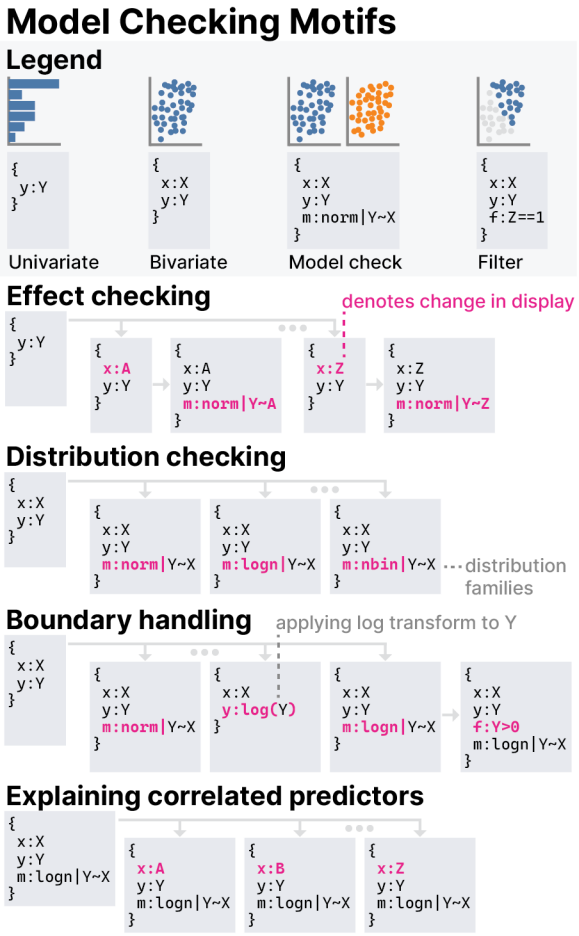

Fig. 5: Model checking motifs observed in our evaluation of EVM.

In each line of questioning, the interviewer referred back to specific examples of situations where the participant either created or attempted to specify a model check. The goal of this discussion was to elicit participants' reflections on how EVM influenced their analysis process and identify potential challenges to adoption of model checks in VA.

5.4 Analysis

We characterized participants' use of EVM with and without model check functionality through a qualitative analysis of interview recordings and transcripts. The interviewer reviewed each transcript and video, at first identifying episodes of interest that revealed patterns of visual data exploration behavior, uses of model checks, and potential improvements to EVM. The study team then analyzed these episodes of interest and summarized what participants said in terms of topical themes and design tensions, focusing especially on any difficulties that participants seemed to have expressing or interpreting model checks.

5.5 Results

We report on how patterns of data exploration behavior in a prototype VA tool differ with versus without model checking functionality. We also summarize what visual cues participants relied on to interpret model check visualizations and challenges that participants encountered using EVM's interface, with an eye toward the development of future VA tools incorporating model checks.

5.5.1 Data analysis motifs

Our analysis revealed patterns of visual data exploration behavior that we refer to as *"motifs"*, which were common sequences of operations in EVM that often seemed to serve the purpose of making data exploration systematic. For example, Figure 5 shows motifs specific to model checking (described below). Our notion of motifs is similar to Battle and Heer's "VA subtasks" [2]. However, unlike subtasks, motifs were not focused on answering questions specific to a given dataset, but rather these were procedures we observed participants apply repeatedly to surface and check relationships between different subsets of variables. These motifs help us describe users' analysis behavior at baseline—when using a simple drag-and-drop VA tool without model checking functionality—and then to describe the new behaviors that emerge when using model checks as implemented in EVM.

Behavior *without* model checks. Common visual exploration motifs we observed without model checking enabled facilitate use cases such as anomaly detection and pattern finding. Some of these motifs are similar to patterns of analysis behavior described in prior work. For example, these include the **univariate tour** viewing univariate summaries of each variable (7/12 participants) and the **bivariate tour** making pairwise comparisons of all predictors to look for correlations (6/12 participants). These are similar to automated summaries provided by previous systems like Voyager2 [64], which enable the user to get *"a cross tab of all the covariates" (P07)*. Some participants (4/12) worked even more systematically, applying these motifs to selected *"clusters of explanatory variables" (P10)* in turn.

Other motifs we observed without model checking functionality focused on explaining the distribution of a specific outcome variable of interest. For example, we observed **hunting for main effects** by cycling through each predictor variable comparing it to the outcome in a bivariate view (7/12 participants), which is sometimes interleaved with a univariate tour of predictors, and **hunting for interactions** by faceting bivariate plots of main effects by a sequence of third variables to look for interaction effects (9/12 participants). Another motif used by most participants (9/12) to account for conditional structure in the data was **filter toggling**, eyeballing a pattern before and after applying variations on a particular filter, which was also described in prior work on view sequencing in narrative visualizations [23]. All participants referred to expectations at some point when performing these distribution-explaining motifs, and some participants (8/12) would tell stories about the data in order to explain discovered patterns.

Behavior *with* model checks. When we introduced model checks to EVM in the second data exploration session, we noticed marked changes in the behavior of most participants. Without model checks enabled, all participants except *P12* briefly explored patterns across a broad set of available variables and then circled back to recheck relationships they had investigated previously. However, with model checks enabled, sequences of related operations became longer, and data exploration became less circuitous, consistent with findings of prior work that adding modeling functionality to VA tools leads to less breadth of analysis during insight-oriented data exploration [32].

Model checking tended to structure participants' thinking around one or two long chains of operations geared toward gradually improving models. Some of these model improvement motifs were concerned with finding an appropriate way to approximate the distribution of the outcome variable. For example (Fig. 5), we observed participants (7/12) **distribution checking** to hone in on a plausible distribution family and (5/12 participants) **boundary handling** by iteratively applying different distribution families, transforms, and filters in order to account for natural boundaries in the data (e.g., no counts below zero). Other model improvement motifs were more concerned with predictor selection. Similar to the demonstrated *understand-correlated-predictors* use case (see Section 4.1), we observed some participants (6/12) **explaining correlated predictors** by checking the patterns predicted by a provisional model against the domains of predictors *not* included in the model to see whether any structure in the data remains unaccounted for.

Although many of our participants exhibited these model improvement motifs, such motifs did not seem to benefit all participants equally. A subset of participants (5/12) used model improvement motifs to focus on developing a fine-grained understanding of the data generating process (DGP), demonstrating the kind of thinking we designed EVM's model checking functionality to elicit in users. For example, one participant said both that, *"When I fit a model, I was definitely thinking more about the second moment." (P09)* and that,

> *"With the initial [think-aloud session], I didn't think about bounds as much... I don't think it came up, but it was only*

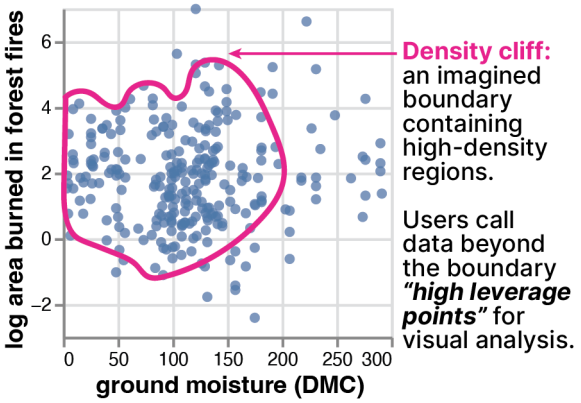

Fig. 6: Depiction of an imagined boundary we refer to as a "density cliff". Many heuristics for visual analysis with scatterplots and strip plots seem anchored to this kind of reference.

> *when trying to describe a model that I started putting these theoretical bounds on what values could take." (P09)*

In addition to paying increased attention to variance relationships and expected boundaries, these participants (5/12) mentioned that visual model checks calibrated their sense of uncertainty around outliers, undersampled regions of the data, and the tails of distributions. HOPs depicting uncertainty as sampled model predictions seem to clarify the possible structure of the data in regions where data is sparse.

However, for a smaller subset of participants, relying on the same model improvement motifs promoted *fixation* on trying to understand the underlying implementation of model checks (2/12 participants) or on superficial matching of the shape of the predictive distribution to the shape of the data without thought about what the relationships meant (2/12 participants). Two participants fixated on long sequences of model improvements and neglected to explore most of the dataset during the *think-aloud with model checks* portion of the interview.

Exceptions to this trend of longer operation sequences were two participants who always used model checks in a manner more akin to one-off statistical tests for specific relationships. They would specify models with versus without an effect of a particular predictor in order to see which model's predictive distribution seemed more in line with the data, similar to the demonstrated *evaluate-the-plausibility-of-an-interpretation* use case (see Section 4.1). We call this motif **effect checking** (Fig. 5). Most participants (8/12) used model check visualizations as provisional hypothesis tests at some point, interleaving visual analysis motifs for pattern discovery such as *hunting for main effects/interactions* with model checking motifs for vetting data interpretations such as *effect checking* or *explaining correlated predictors*.

5.5.2 Interpretation of model check visualizations

Most participants (8/12) interpreted model check visualizations primarily in terms of the match between the shape of the data and model predictions. The most salient visual cues for shape seemed to involve the concentration of data points in EVM's scatterplots and strip plots, with many participants (8/12) paying special attention to an imaginary boundary around high-concentration regions, which we refer to as a *"density cliff"* (Fig. 6). Some participants seemed self-aware about how points near a density cliff influenced their perception of a pattern, for example, *"I do see that again these kind of larger [fires] are occurring... but that's really cherry picking just a few high-leverage points, though, so I'm not sure that means a whole lot." (P01).* When data points were further away from a density cliff, some participants (3/12) referred to them as separate modes and others (8/12) referred to them as outliers depending on the number of points and how far they were from the density cliff. Although prior work finds that "regression by eye" down-weights outliers [10], our participants judged misfit using heuristics that were more sensitive to anomalies. When the shapes did not match between the data and model predictions, participants (4/12) interpreted this as a sign of misfit. A subset of participants frequently relied on residuals plots to assess model fit, with some (2/12 participants) describing asymmetry around zero as a helpful cue and others (4/12 participants) pointing out heteroskedasticity of prediction errors across the domain of a given predictor as a cue that the relationship with that predictor was not adequately captured by the model (Fig. 7).

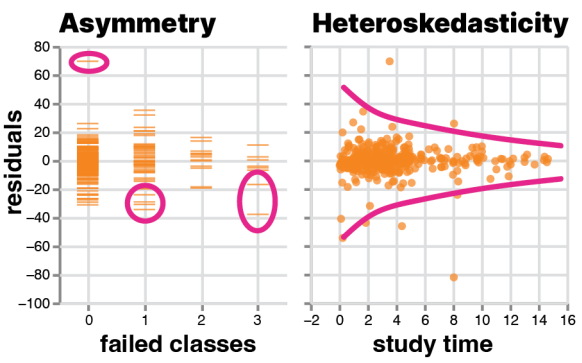

Fig. 7: Visual cues that participants used to interpret residuals.

5.5.3 Challenges using model checks in EVM

Many difficulties participants faced with EVM involved distrust of eyeballing. One participant bemoaned when struggling to visually differentiate the fit quality of two models, *"My lying eyes sometimes deceive me." (P01).* This issue is certainly not unique to model checks—indeed, most VA workflows tend to rely on such subtle visual inferences. However, all participants described similar dilemmas and wanted to supplement visual inferences with diagnostic metrics for models (e.g., information criteria, R-squared, regression coefficients). We avoided providing such diagnostics in EVM in part because we worried that participants would over-rely on them in ways that would interfere with our design goal of promoting generative thinking (see Section 3) and detract from our investigation of visual model checking.

Similarly, most participants (9/12) wanted the ability to derive summary statistics such as counts, means, and quartiles on the fly and to apply them to regions of the chart using brush interactions. For example, two participants wanted to brush out a region of a scatterplot and compute the number of data points in that region relative to the size of the dataset. Many (7/12 participants) wanted the ability to change the default visualizations of the tool by binning continuous variables or by recasting variables as different data types. These difficulties seemed to stem from the lack of granularity that EVM's scatterplots and strip plots provide for inspecting the relative density of regions on charts: *"It's either dense, medium dense, or not very dense." (P03).* Supporting histograms and density plots would have addressed some of these concerns, however, emphasizing visual aggregation may also lead to overconfidence in visual inferences [39], so defaults must navigate this trade-off. Seeking stronger visual signals about relative density led two participants to rely on highly inefficient exploration strategies such as scrolling through many faceted bar charts.

When using the model bar to express provisional models, most problems stem from the challenge of anticipating what accounts for misfit. Although some participants (5/12) said that model checks in EVM make it quick to try out models, and others (6/12 participants) said visual model checks made it easy to see misfit, they seemed to struggle to improve models using the interface. For example, all participants began modeling with a normal distribution, even though the outcome variable in both datasets had a lower bound at zero, making it likely that the modeling assumption of Gaussian distributed residuals would be violated. Upon seeing the resulting misfit, all participants except *P11* at first added more variables to their model specification rather than changing their choice of distribution family. Eventually, by using distribution checking or boundary handling motifs for iterative model checking, all participants except *P12* discovered that the choice of distribution family accounted for misfit. Participants may have added predictors before changing distributional assumptions because EVM's model bar makes it easy to dump additional predictors into subsequent model iterations, and they did not stop to rethink distributional assumptions.

Many participants (7/12) struggled when choosing among distribu-

tion families, which was often a contributor to misfit. This was especially common when trying to reason about the back-transformations (see Section 4.3) in the log normal and logit normal families. Multiple participants attributed some difficulties to lack of knowledge about the domain of the datasets (3/12) or to being rusty at specifying regression models in R (5/12). In contrast to prior work showing that VA users don't follow up on model misfit [8], we find that all participants attempted to reason about what might be wrong with misfit models.

Our results suggest the need for future work on guided model elicitation interfaces. Two participants suggested scaffolding a path of model exploration between the simplest possible intercept model and a "dredge model" including all available predictors. This is in line with practices in a Bayesian workflow [15, 19] as well as the desire of some participants (5/12) to avoid including multiple highly-correlated predictors in their model specifications, exemplified by this quote:

> *"How do we get the most parsimonious model? How do we remove things that are perhaps highly correlated, or even collinear, and have the best model with the fewest features? which is I think where I would go next, being able to prune the model a bit, not have things that I don't need." (P08).*

A few participants (3/12) noted situations where a single model could not account for distinct sub-populations within the data and requested support for partitioning the data into subsets to be modeled separately.

6 Discussion

Our work building and evaluating `EVM` points to new design requirements for model checks beyond those identified in Section 3, as well as articulatory gaps faced by users of visual analysis tools more broadly.

We find that **model checking can improve understanding of data generating process (DGP), but only when users avoid fixating on non-conceptual aspects of analysis**, such as the underlying implementation of statistical models or superficially matching model predictions to patterns in the data. Participants who used model checks to discover important attributes of the DGP such as boundedness or correlated predictors tended to be experienced at interpreting the relationship between modeling assumptions and data. In contrast, the smaller subset of participants who fixated on non-conceptual aspects of model checking were more familiar with use cases for statistical models that prioritize predictive accuracy over scrutinizing assumptions. This partially confirms our design hypothesis about the utility of model checks in visual analytics, but it also points to the need for model checking tools to accommodate users with fluency in different modeling approaches.

Although model checks help users identify misfit in models, **users require guidance about which incremental improvements to a model could plausibly improve fit**. For example, most participants added new predictors to a given model before considering whether they had chosen an appropriate distribution family for the data. Future work might generate modeling recommendations that nudge users to (re-)examine specific assumptions or asserted structures. More expressive tools that allow users to articulate specific aspects of misfit that concern them as input to a recommender are also well motivated—e.g., if a cluster of data points appear in the data distribution but not in model predictions.

Additionally, we present **analysis *"motifs"* that reflect procedural abstractions for visual analytics (VA) workflows**. Similar to patterns of visualization sequencing described in prior work [2, 23] and automated data summaries provided by previous systems such as Voyager2 [64], we think of these motifs as a workflow-level abstraction describing subroutines in visual analysis. In contrast, a chart-level abstraction such as the grammar of graphics [63] focuses on specifying analyses one visualization at a time. For our purposes, these motifs serve to characterize common sequences of operations within VA sessions. Looking to the future of VA tools, we envision interfaces that enable users to author and reuse motifs in data exploration workflows, extending the idea of "wildcard fields" in Voyager2 [64] such that users could specify a sequence of diagnostic visualizations and apply them across a series of provisional models reflecting competing data interpretations. Specifying model checking workflows at the level of motifs could further reduce the amount of effort required to rigorously check data interpretations without loss of expressiveness.

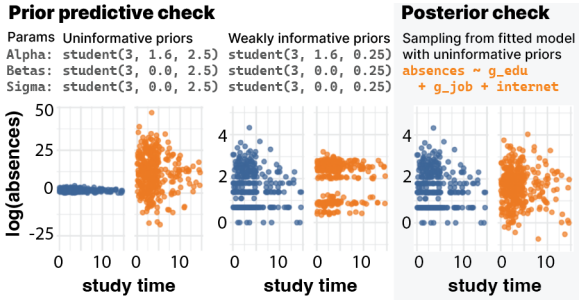

Fig. 8: Examples of prior vs posterior predictive checks for the same model. This model check examines whether study time is predictive of log absences after accounting for proxies of socioeconomic status. Uninformative priors (left) blow up the scale of the y-axis, making an unhelpful model check. Sampling from weakly informative priors (middle), rather than a fitted model (right), can lead to model checks that show a larger space of possible patterns that follow from a model's assumptions (e.g., homogeneity of variance), potentially increasing visual signal for judging whether a model is mis-specified.

Importantly, beyond need-finding for model checking interfaces, we also add to a line of work [8, 9, 32] demonstrating that **model checks can integrate cleanly into exploratory visual analysis workflows**, offering support to users transitioning between open-ended exploration tasks [2] and what Tukey [50] calls "rough confirmatory analysis." Participants who spend relatively little time working with programming interfaces appreciated that `EVM` offered access to modeling and diagnostics approaches which usually require scripting in statistical programming languages. For them, `EVM` offered a quick way to refine data interpretations and fight *"a false sense of security about the quality of the data" (P02)* that they say sets in when using graphical user interfaces for exploratory visual analysis. Similarly, participants who spend relatively more time using statistical tools viewed `EVM` as a way to test their preconceived notions by rapidly iterating on models. One of these participants described how, *"[Visual model checking] felt much less p-hacky than it might have if I'd been looking at the numbers. You know, I'm not just choosing the [model] with the best R-squared or whatever." (P08)*, echoing others for whom the ease of generating visual checks enabled a faster pace of analysis without loss of rigor.

6.1 Ongoing & future work

Investigating how to integrate **model recommendations** into a tool like `EVM` is a natural follow-up to our work. Recommendations should account for what is known about the user's understanding of the DGP at the time of recommendation, similar to the way that tools like Tisane [26] and Visual Causality Analyst [57, 58] anchor model suggestions on knowledge elicited from users. If the user's preferred model at a given moment during analysis represents their traversal of a *"model space"* they are searching, recommendations should be proximal to and informed by the user's current model. Rather than suggesting a single "best" proximal model for the user to consider next, recommender systems should highlight multiplicity, where multiple alternative models perform similarly and cannot be easily distinguished. Model recommendations should also be informed by the visual model checks an analyst creates and the specific patterns in data that they struggle to explain. We envision interfaces where analysts can directly select a pattern in a visualization that they wish to better explain—i.e., the kind of superficial pattern matching that some participants fixated on—and a recommendation engine would suggest additional models that could capture the pattern. This could help analysts remain focused on how visual patterns relate to their conceptual understanding of DGP, providing softer on-ramps to the kind of generative thinking that `EVM` facilitates.

In `EVM`, we implemented comparisons between observed data and predictions from a *fitted* model. However, there are many possible **ways to sample predictions** reflecting a given model, and the specific approach plays a large role in determining the visual appearance of a

model check. Specifying a model defines a *parameter space*, a latent multivariate distribution that is sampled from and further transformed in order to produce a prediction. Sampling fitted models as we did in EVM means sampling from this distribution in ways that minimize the discrepancy between the data and model predictions. However, Bayesian prior predictive checking offers an alternative approach, which helps to shed light on the importance of these low-level implementation choices (Fig. 8, see Supplemental Materials). In a prior predictive check, the choice of prior determines the scale of the predictive distribution, enabling this approach to show more possible structures in predictions that are consistent with a model specification but which are ruled out by the fitting process. Contrasting these approaches raises the question of whether enabling users to specify priors and sample from them would be a more direct way to assess users' expectations through visual checks, to the extent that elicited models represent users' beliefs as assumed by prior work [8, 9, 32]. However, designing for Bayesian predictive checks might increase the level of modeling complexity beyond what some VA users are familiar with, e.g., requiring procedures for fine tuning and justifying priors. Where concerns about model check implementation details threaten to pull analysts' attention away from DGP, we suggest providing more documentation of how a tool like EVM samples predictive distributions in order to provide transparency.

Critical to these endeavors is incorporation of **diagnostic metrics** for ensuring that a seemingly well-fitting model also has predictive power, such as cross validation on a holdout set. Using holdout sets rather than training data for model checks would increase the visual discrepancy between data and model predictions, providing stronger visual signals of potential misfit. Having access to diagnostics like cross validation metrics might help address the lack of confidence in eyeballing among some study participants, in addition to helping users of future tools like EVM avoid overfit explanations, overconfidence in mis-specified models, and misleading local extrema in model search space.

Related to issues of overfitting and iterative model development, it is crucial to develop **safeguards against post-hoc statistical inference** in tools like EVM. A potential failure mode with any VA tool is that users will make multiple comparisons before deciding what relationships are important [68], which can inflate false positive rates similar to p-hacking [46]. Anticipating the concern that incorporating modeling into VA tools could exacerbate these risks, we carefully design EVM to make it difficult to perform confirmatory hypothesis testing—e.g., by not providing p-values, learned coefficients, or other metrics conventional in statistical testing. Unlike previous work, which enforces a strict order of operations where VA tools only elicit models before showing the data [8, 9, 32], we opt for a more open-ended study of how people use modeling in VA when left to their own discretion. In the user study, we find that participants are relatively cautious about overtrusting models they fit, suggesting they do not view model check visualizations as a definitive inference method. However, we observe some participants using effect checking motifs (Fig. 5) in ways that resemble hypothesis testing, and this suggests opportunities for design patterns that distinguish between comparisons in service of model development versus proper confirmatory testing. As proposed above and in prior work [68], future tools like EVM should support holdout sets for model validation and confirmatory testing, which will require careful consideration of how to avoid leaking information when swapping training and holdout sets. In cases where holdout sets are unavailable, we envision that users could save models to test on future data when it becomes available. Future work should evaluate various regimes for controlling post-hoc inference, including algorithmic approaches (e.g., [5, 49]), to determine which are best suited to the VA setting.

Additionally, future research should develop tools for ensuring effective model check visualizations. Systems like EVM, along with prior work [24, 36], motivate a **model check grammar** to facilitate greater flexibility in traversing the design space of uncertainty visualizations required for model checking. This would be necessary in order to enable motif-based authoring systems where analysts create visual checks and apply them across a sequence of related views demarcated by, e.g., different variable selections, data transformations, or model iterations. How a visual check should change when model predictions are grafted onto it is determined by constraints that are difficult to express using Vega [45] due to the diffuseness and viscosity of its notation [20]. As a workaround, when implementing EVM we created a separate Vega template for each possible layout, similar to recent work on "parameterized declarative templates" [37]. However, extending this approach for model checks requires additional research and development on abstractions for model check visualizations in particular. To support engineering future tools, a model check grammar should include (1) layout constraints defining how data and model predictions are treated, (2) primitives for sampling from models in order to generate uncertainty visualizations and model diagnostics, and (3) interaction techniques that enable users to express which patterns reflect conceptually important misfit. More broadly, understanding what makes a model check visualization effective may not be equivalent to what makes a visualization of observed data effective, motivating empirical work.

6.2 Limitations

Large scale and high-dimensional data are open problems in VA tools that remain unaddressed by EVM. For the purpose of creating a proof-of-concept tool, we focused on model checks for relatively small datasets. However, some of our design choices need refinement to work at larger scales. For example, using HOPs [25] for higher-dimensional views requires careful interaction design (e.g., in node-link diagrams [69]). Relatedly, disaggregated views [39] showing one mark per data point become unwieldy at large sample sizes, due to limitations around visual crowding [1] and computer memory. Aggregation is a common approach to side-stepping these issues of scale, however, aggregated data and model outputs have fundamentally different meanings as summary statistics than disaggregated data and model predictions. Future work will need to resolve when such aggregation is and is not advisable.

Similarly, our choice to limit EVM to position encodings (i.e., x-axis, y-axis, row, column) rules out visualization techniques that might be more suitable for higher-dimensional data, where datasets have many variables that can each take on many values. EVM only enables viewing a subset of a high-dimensional dataset's features at one time, a limitation of many VA tools. Although model check visualizations carry information about variables that are not currently in view—i.e., a model can make predictions based on a variable that isn't visualized—analysts may still struggle to reason about the complex structure of correlated variables that underlies a particular view. Future work should more directly investigate whether using model checks to reason about variables that are not in view can solve the high-dimensionality problem, and the ways in which this approach might be error prone.

Some conditions of our user study design were difficult to control. When preparing datasets for participants to explore, we dropped certain variables and observations from the student absences dataset in order to make it match the size of the forest fires dataset. It's possible that these adjustments affected participants' ability to find patterns in the data, however, this did not come up explicitly during our interviews. Beyond the size of datasets, there were other factors which likely impact our results that were not possible to control for. These include each participant's level of interest in or familiarity with the provided datasets.

7 Conclusion

We present EVM, a proof-of-concept tool enabling analysts to express and check statistical models during visual data exploration. EVM is a design investigation into how visual analytics (VA) tools can incorporate statistical modeling to facilitate more rigorous thinking about a data generating process (DGP). This augments the typical process of visual pattern discovery with procedures for articulating and scrutinizing claims about a DGP, which we argue elevates VA tools from producing nebulous "insights" to vetting provisional data interpretations and providing a more concrete basis for further analysis. Our work demonstrates the potential of model check visualizations to better connect VA tools with the cognitive and statistical procedures by which analysts develop and evaluate their conceptual understanding of data.

Acknowledgments

We thank Justin Talbot and Andrew Gelman for providing early input about design requirements for incorporating regression models into graphical user interfaces for exploratory visual analysis. Jessica Hullman thanks NSF #2211939 and #1930642 for supporting this work.

References

[1] G. A. Alvarez. Representing multiple objects as an ensemble enhances visual cognition. *Trends in Cognitive Sciences*, 15(3):122–131, 2011. doi: 10.1016/j.tics.2011.01.003 9

[2] L. Battle and J. Heer. Characterizing exploratory visual analysis: A literature review and evaluation of analytic provenance in tableau. *Computer Graphics Forum*, 38(3):145–159, 2019. doi: 10.1111/cgf.13678 2, 6, 8

[3] R. A. Becker, W. S. Cleveland, and M.-J. Shyu. The visual design and control of trellis display. *Journal of Computational and Graphical Statistics*, 5(2):123–155, 1996. doi: 10.1080/10618600.1996.10474701 4

[4] R. A. Becker, W. S. Cleveland, and A. R. Wilks. Dynamic graphics for data analysis. *Statistical Science*, 2(4):355–383, 1987. doi: 10.1214/ss/1177013104 2

[5] R. Berk, L. Brown, A. Buja, K. Zhang, and L. Zhao. Valid post-selection inference. *The Annals of Statistics*, 41(2):802 – 837, 2013. doi: 10.1214/12-AOS1077 9

[6] A. Buja, D. Cook, H. Hofmann, M. Lawrence, E.-k. Lee, D. F. Swayne, and H. Wickham. Statistical inference for exploratory data analysis and model diagnostics. *Royal Society Philosophical Transactions*, Series A(367):4361–4383, 2009. doi: 10.1098/rsta.2009.0120 2

[7] A. Buja, C. Hurley, and J. McDonald. A data viewer for multivariate data. In *Proc. of the 18th Symposium on the Interface*, pp. 171–174, 1987. 2

[8] I. K. Choi, S. Mishra, T. Childers, K. Harris, N. K. Raveendranath, and K. Reda. Concept-driven visual analytics: An exploratory study of model- and hypothesis-based reasoning with visualizations. ACM Conference on Human Factors in Computing Systems - Proc., 5 2019. doi: 10.1145/3290605.3300298 2, 8, 9

[9] I. K. Choi, N. K. Raveendranath, J. Westerfield, and K. Reda. Visual (dis)confirmation: Validating models and hypotheses with visualizations. pp. 116–121. IEEE Proc. International Conference in Information Visualization, 7 2019. doi: 10.1109/IV-2.2019.00032 2, 8, 9

[10] M. Correll and J. Heer. Regression by eye: Estimating trends in bivariate visualizations. In *ACM Human Factors in Computing Systems (CHI)*, 2017. doi: 10.1145/3025453.3025922 7

[11] P. Cortez and A. Morais. A data mining approach to predict forest fires using meteorological data. In *J. Neves, M. F. Santos and J. Machado Eds., New Trends in Artificial Intelligence, Proc. of the 13th EPIA*, pp. 512–523. APPIA, 2007. 5

[12] P. Cortez and A. Silva. Using data mining to predict secondary school student performance. In *A. Brito and J. Teixeira Eds., Proc. of 5th FUture BUsiness TEChnology Conference (FUBUTEC)*, pp. 5–12. EUROSIS, 2008. 5

[13] B. Devezer, D. J. Navarro, J. Vandekerckhove, and E. O. Buzbas. The case for formal methodology in scientific reform. *Royal Soc. Open Science*, 8200805, 2020. doi: 10.1098/rsos.200805 2

[14] D. Dua and C. Graff. UCI machine learning repository, 2017. 5

[15] J. Gabry, D. Simpson, A. Vehtari, M. Betancourt, and A. Gelman. Visualization in bayesian workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182(2):389–402, 2019. doi: 10.1111/rssa.12378 2, 3, 8

[16] A. Gelman. A bayesian formulation of exploratory data analysis and goodness-of-fit testing. *International Statistical Review*, 71(2):369–382, 2003. doi: 10.1111/j.1751-5823.2003.tb00203.x 2

[17] A. Gelman. Exploratory data analysis for complex models. *Journal of Computational and Graphical Statistics*, 13(4):755–779, 2004. doi: 10.1198/106186004X11435 2

[18] 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*, 348:1–17, 2013. 3

[19] A. Gelman, A. Vehtari, D. Simpson, C. C. Margossian, B. Carpenter, Y. Yao, L. Kennedy, J. Gabry, P.-C. Bürkner, and M. Modrák. Bayesian workflow. *arXiv preprint*, 2020. doi: 10.48550/arXiv.2011.01808 2, 8

[20] T. R. G. Green. Cognitive dimensions of notations. In *Proc. of the 5th Conference of the British Computer Society, Human-Computer Interaction Specialist Group on People and Computers V*, p. 443–460. Cambridge University Press, USA, 1990. 9

[21] G. Guo, E. Karavani, A. Endert, and B. C. Kwon. Causalvis: Visualizations for causal inference. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3581236 2

[22] F. E. Harrell. *Regression modeling strategies: With applications to linear models, logistic regression, and survival analysis*. Springer-Verlag, 2001. doi: 10.1007/978-1-4757-3462-1 3

[23] J. Hullman, S. Drucker, N. Henry Riche, B. Lee, D. Fisher, and E. Adar. A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2406–2415, 2013. doi: 10.1109/TVCG.2013.119 6, 8

[24] J. Hullman and A. Gelman. Designing for interactive exploratory data analysis requires theories of graphical inference. *Harvard Data Science Review*, 2021. doi: 10.1162/99608f92.3ab8a587 1, 2, 3, 9

[25] J. Hullman, P. Resnick, and E. Adar. Hypothetical Outcome Plots Outperform Error Bars and Violin Plots for Inferences about Reliability of Variable Ordering. *PloS one*, 10(11):e0142444, 2015. doi: 10.1371/journal.pone.0142444 3, 9

[26] E. Jun, A. Seo, J. Heer, and R. Just. Tisane: Authoring statistical models via formal reasoning from conceptual and data relationships. In *Proc. of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22. ACM, New York, NY, USA, 2022. doi: 10.1145/3491102.3501888 8

[27] A. Kale, F. Nguyen, M. Kay, and J. Hullman. Hypothetical outcome plots help untrained observers judge trends in ambiguous data. *IEEE Trans. on Visualization and Computer Graphics*, 25(1):892–902, 2019. doi: 10.1109/TVCG.2018.2864909 3

[28] A. Kale, Y. Wu, and J. Hullman. Causal support: Modeling causal inferences with visualizations. *IEEE Trans. on Visualization and Computer Graphics*, 28, 2022. doi: 10.1109/TVCG.2021.3114824 2

[29] D. Kibler, D. W. Aha, and M. Albert. Instance-based prediction of real-valued attributes. *Computational Intelligence*, 5:51–57, 1989. 5

[30] Y.-S. Kim, P. Kayongo, M. Grunde-McLaughlin, and J. Hullman. Bayesian-assisted inference from visualized data. *IEEE Trans. on Visualization and Computer Graphics*, 27(2):989–999, 2021. doi: 10.1109/TVCG.2020.3028984 2

[31] Y.-S. Kim, L. A. Walls, P. Krafft, and J. Hullman. A bayesian cognition approach to improve data visualization. In *Proc. of the 2019 CHI Conference on Human Factors in Computing Systems*, p. 1–14. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300912 2

[32] R. Koonchanok, P. Baser, A. Sikharam, N. K. Raveendranath, and K. Reda. Data prophecy: Exploring the effects of belief elicitation in visual analytics. In *Proc. of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2021. doi: 10.1145/3411764.3445798 2, 6, 8, 9

[33] T. Kraska. Northstar: An interactive data science system. *Proceedings of the VLDB Endowment*, 11(12):2150–2164, 2018. doi: 10.14778/3229863.3240493 2

[34] A. Loy, L. Follett, and H. Hofmann. Variations of Q–Q Plots: The Power of Our Eyes! *American Statistician*, 70(2):202–214, 2016. doi: 10.1080/00031305.2015.1077728 2

[35] M. Majumder, H. Hofmann, and D. Cook. Validation of visual statistical inference, applied to linear models. *Journal of the American Statistical Association*, 108(503):942–956, 2013. doi: 10.1080/01621459.2013.808157 2

[36] A. M. McNutt. No grammar to rule them all: A survey of json-style dsls for visualization. *IEEE Trans. on Visualization and Computer Graphics*, 29(01):160–170, jan 2023. doi: 10.1109/TVCG.2022.3209460 9

[37] A. M. McNutt and R. Chugh. Integrated visualization editing via parameterized declarative templates. In *Proc. of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2021. doi: 10.1145/3411764.3445356 9

[38] A. J. Miller. Selection of subsets of regression variables. *Journal of the Royal Statistical Society. Series A (General).*, 147(3):389–425, 1984. doi: 10.2307/2981576 3

[39] F. Nguyen, X. Qiao, J. Heer, and J. Hullman. Exploring the Effects of Aggregation Choices on Untrained Visualization Users' Generalizations From Data. *Computer Graphics Forum*, 39(6):33–48, 2020. doi: 10.1111/cgf.13902 7, 9

[40] K. Oberauer and S. Lewandowsky. Addressing the theory crisis in psychology. *Psychonomic Bulletin & Review*, 26(5):1596–1618, 2019. doi:

10.3758/s13423-019-01645-2 2

[41] J. Ooms. The opencpu system: Towards a universal interface for scientific computing through separation of concerns. *arXiv:1406.4806 [stat.CO]*, 2014. doi: 10.48550/arXiv.1406.4806 5

[42] J. Pinheiro, D. Bates, S. DebRoy, D. Sarkar, E. authors, S. Heisterkamp, B. Van Willigen, and R-core. nlme: Linear and Nonlinear Mixed Effects Models, 2020. 3, 5

[43] X. Pu and M. Kay. The garden of forking paths in visualization: A design space for reliable exploratory visual analytics: Position paper. In *2018 IEEE Evaluation and Beyond-Methodological Approaches for Visualization (BELIV)*, pp. 37–45. IEEE, 2018. doi: 10.1109/BELIV.2018.8634103 2

[44] R. A. Rigby and D. M. Stasinopoulos. Generalized additive models for location, scale and shape. *Applied Statistics*, 54:507–554, 2005. doi: 10.1111/j.1467-9876.2005.00510.x 3, 5

[45] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer. Reactive vega: A streaming dataflow architecture for declarative interactive visualization. *IEEE Trans. on Visualization and Computer Graphics*, 2016. doi: 10.1109/TVCG.2015.2467091 5, 9

[46] 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 2, 9

[47] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Trans. on Visualization and Computer Graphics*, 8(1):52–65, 2002. doi: 10.1109/2945.981851 2

[48] J. Talbot. personal communication. 3

[49] R. J. Tibshirani, J. Taylor, R. Lockhart, and R. Tibshirani. Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association*, 111(514):600–620, April 2016. doi: 10.1080/01621459.2015.110 9

[50] J. W. Tukey. Data analysis, computation and mathematics. *Quarterly of Applied Mathematics*, 30(1):51–65, 1972. 2, 8

[51] J. W. Tukey. Exploratory data analysis as part of a larger whole. In *Proc. of the 18th conference on design of experiments in army research and development i. Washington, DC*, vol. 1010, 1972. 2

[52] J. W. Tukey. *Exploratory data analysis*. Addison-Wesley Pub, Reading, Mass, 1977. doi: 10.1002/bimj.4710230408 4

[53] J. W. Tukey and M. B. Wilk. Data analysis and statistics: An expository overview. In *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference*, pp. 695–709, 1966. doi: 10.1145/1464291.1464366 2

[54] S. Vanderplas. Designing graphics requires useful experimental testing frameworks and graphics derived from empirical results. *Harvard Data Science Review*, pp. 1–8, 2021. doi: 10.1162/99608f92.7d099fd0 2

[55] S. VanderPlas and H. Hofmann. Clusters beat trend!? Testing feature hierarchy in statistical graphics. *Journal of Computational and Graphical Statistics*, 26(2):231–242, 2017. doi: 10.1080/10618600.2016.1209116 2

[56] P. F. Velleman. Datadesk: an interactive package for data exploration, display, model building, and data analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(4):407–414, 2012. doi: 10.1002/wics.1208 2

[57] J. Wang and K. Mueller. The visual causality analyst: An interactive interface for causal reasoning. *IEEE Trans. on Visualization and Computer Graphics*, 22(1):230–239, 2016. doi: 10.1109/TVCG.2015.2467931 2, 8

[58] J. Wang and K. Mueller. Visual causality analysis made practical. *2017 IEEE Conference on Visual Analytics Science and Technology, VAST 2017 - Proc.*, (October):151–161, 2018. doi: 10.1109/VAST.2017.8585647 2, 8

[59] H. Wickham, M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi: 10.21105/joss.01686 5

[60] H. Wickham, D. Cook, and H. Hofmann. Visualizing statistical models: Removing the blindfold. *Statistical Analysis and Data Mining*, 8(4):203–225, Aug 2015. doi: 10.1002/sam.11271 3

[61] H. Wickham, D. Cook, H. Hofmann, and A. Buja. Graphical inference for infovis. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):973–979, 2010. doi: 10.1109/TVCG.2010.161 2

[62] G. N. Wilkinson and C. E. Rogers. Symbolic description of factorial models for analysis of variance. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 22(3):392–399, 1973. doi: 10.2307/2346786 3, 5

[63] L. Wilkinson. *The Grammar of Graphics*. 2005. doi: 10.1007/0-387-28695-0 3, 4, 8

[64] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proc. of the ACM Conference on Human Factors in Computing Systems (CHI)*, 2017. doi: 10.1145/3025453.3025768 4, 6, 8

[65] Y. Wu, L. Xu, R. Chang, and E. Wu. Towards a bayesian model of data visualization cognition. In *IEEE Visualization Workshop on Dealing with Cognitive Biases in Visualisations (DECISIVe)*, 2017. 2

[66] X. Xie, F. Du, and Y. Wu. A visual analytics approach for exploratory causal analysis: Exploration, validation, and applications. *IEEE Trans. on Visualization and Computer Graphics*, 27(2):1448–1458, 2021. doi: 10.1109/TVCG.2020.3028957 2

[67] T. Yarkoni and J. Westfall. Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, 12(6):1100–1122, 2017. PMID: 28841086. doi: 10.1177/1745691617693393 2

[68] E. Zgraggen, Z. Zhao, R. Zeleznik, and T. Kraska. Investigating the effect of the multiple comparisons problem in visual analysis. In *Proc. of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 1–12. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3174053 2, 3, 9

[69] D. Zhang, E. Adar, and J. Hullman. Visualizing uncertainty in probabilistic graphs with network hypothetical outcome plots (nethops). *IEEE Trans. on Visualization and Computer Graphics*, 28(1):443–453, 2022. doi: 10.1109/TVCG.2021.3114679 9

[70] Z. Zhao, L. De Stefani, E. Zgraggen, C. Binnig, E. Upfal, and T. Kraska. Controlling false discoveries during interactive data exploration. In *Proc. of the ACM International Conference on Management of Data*, pp. 527–540, 2017. doi: 10.1145/3035918.3064019 2