Hypothesis Formalization: Empirical Findings, Software Limitations, and Design Implications

EUNICE JUN, University of Washington, USA
MELISSA BIRCHFIELD, University of Washington, USA
NICOLE DE MOURA, Eastlake High School, USA
JEFFREY HEER, University of Washington, USA
RENÉ JUST, University of Washington, USA

Data analysis requires translating higher level questions and hypotheses into computable statistical models. We present a mixed-methods study aimed at identifying the steps, considerations, and challenges involved in operationalizing hypotheses into statistical models, a process we refer to as *hypothesis formalization*. In a formative content analysis of 50 research papers, we find that researchers highlight decomposing a hypothesis into sub-hypotheses, selecting proxy variables, and formulating statistical models based on data collection design as key steps. In a lab study, we find that analysts fixated on implementation and shaped their analyses to fit familiar approaches, even if sub-optimal. In an analysis of software tools, we find that tools provide inconsistent, low-level abstractions that may limit the statistical models analysts use to formalize hypotheses. Based on these observations, we characterize hypothesis formalization as a dual-search process balancing conceptual and statistical considerations constrained by data and computation and discuss implications for future tools.

CCS Concepts: • **Human-centered computing** → **HCI theory, concepts and models**; **Empirical studies in HCI**; *Laboratory experiments*.

Additional Key Words and Phrases: statistical analysis; scientific discovery; theory of data analysis; mixed-methods

ACM Reference Format:
Eunice Jun, Melissa Birchfield, Nicole de Moura, Jeffrey Heer, and René Just. 2021. Hypothesis Formalization: Empirical Findings, Software Limitations, and Design Implications. *ACM Trans. Comput.-Hum. Interact.* 1, 1, Article 1 (January 2021), 27 pages. **https://doi.org/10.1145/3476980**

1 INTRODUCTION

Using statistics to answer real-world questions requires four steps: (i) translating high-level, domain-specific questions and hypotheses into specific statistical questions [16]; (ii) identifying statistical models to answer the statistical questions; (iii) implementing and executing these statistical models, typically with the help of software tools; and (iv) interpreting the results, considering the domain-specific questions and applying analytical reasoning.

For example, suppose a census researcher asked, "In the United States (U.S.), how does an individual's sex relate to their annual income?" Drawing upon their prior experiences and exploratory

Authors' addresses: Eunice Jun, **emjun@cs.washington.edu**, University of Washington, Seattle, Washington, USA; Melissa Birchfield, **mbirch2@cs.washington.edu**, University of Washington, Seattle, Washington, USA; Nicole de Moura, **nicoledemoura4@gmail.com**, Eastlake High School, Sammamish, Washington, USA; Jeffrey Heer, **jheer@cs.washington.edu**, University of Washington, Seattle, Washington, USA; René Just, **rjust@cs.washington.edu**, University of Washington, Seattle, Washington, USA.

1073-0516/2021/1-ART1 $15.00
https://doi.org/10.1145/3476980

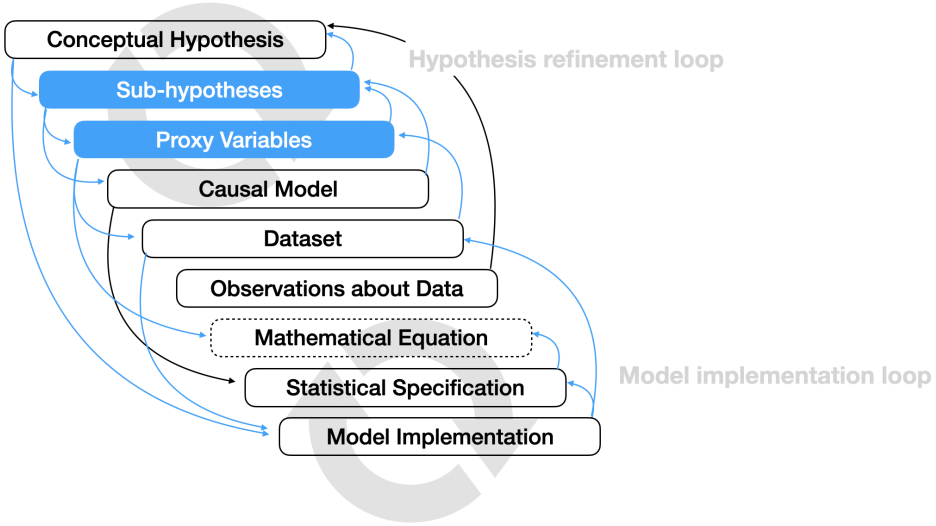

Fig. 1. **Definition and overview of the hypothesis formalization steps and process.** Hypothesis formalization is a dual-search process of translating a **conceptual hypothesis** into a statistical **model implementation**. Blue indicates steps and transitions that we identified. Black indicates steps and transitions discussed in prior work. "Mathematical Equation" (dashed box) was rarely an explicit step in our lab study but evident in our content analysis. Our findings (blue arrows) corroborate and subsume several of the transitions identified in prior work with greater granularity. When they do not, prior work's transitions are included in black. For example, analysts may operationalize a conceptual hypothesis as a causal model (found in prior work, see Figure 2) by first decomposing the conceptual hypothesis into sub-hypotheses and then identifying proxy variables to incorporate in a causal model (blue arrows above). Our definition of hypothesis formalization is a consequence of our synthesis of prior work, content analysis, lab study, and analysis of tools. Hypothesis formalization is a non-linear process. Analysts iterate over conceptual steps to refine their hypothesis in a *hypothesis refinement loop*. Analysts also iterate over computational and implementation steps in a *model implementation loop*. Data collection and data properties may also prompt conceptual revisions and influence statistical model implementation. As analysts move toward model implementation, they increasingly rely on software tools, gain specificity, and create intermediate artifacts along the way (e.g., causal models, observations about data, etc.).

data visualizations, the researcher knows that income in the U.S. is skewed, and they want to know how the distributions of income among males and females differ (step i). However, before implementing, they (implicitly) define their causal model: The researcher knows that other factors, such as education and race, may be associated with employment opportunities, which may then influence income. As such, they refine their conceptual hypothesis to consider the possible effects of race, education, sex, and their interactions on income. They plan to fit a generalized linear model with race, education, sex, and their two-way interactions as predictors of income (step ii). They start implementing a script to load and model data (step iii). The researcher receives a small table of results and is surprised to receive a convergence warning. After further investigation, they simplify their model and remove the interaction effects to see how that may affect convergence (revise step iii). This time, their model's inference algorithm converges, and they interpret the results (step iv), but they really want to study how sex and race interact, so they return to implementation (step iii) and proceed as before, iteratively removing and adding effects and changing computational parameters, and as a by-product shifting which high-level conceptual hypothesis is reflected in the model.

Performing statistical data analysis goes well beyond invoking the correct statistical functions in a software library. Analysts, such as the census researcher, must go back and forth between conceptual hypothesis and model implementation realities, grappling with domain knowledge, limitations of data, and statistical methods.

We refer to the process of translating a conceptual hypothesis into a computable statistical model as *hypothesis formalization*. This process is messy and under-scrutinized in prior work. Consequently, we investigate the steps, considerations, challenges, and tools involved. Based on our findings, we define hypothesis formalization as a dual-search process [43] that involves developing and integrating cognitive representations from two different perspectives—conceptual hypotheses and concrete model implementations. Analysts move back and forth between these two perspectives during formalization while balancing conceptual, data-driven, statistical, and implementation constraints. Figure 1 summarizes our definition and findings. Specifically, the paper addresses the following questions to develop our definition of hypothesis formalization:

- **RQ1 - Steps**: What is the range of steps an analyst might consider when formalizing a hypothesis? How do these steps compare to ones that we might expect based on prior work?
- **RQ2 - Process**: How do analysts think about and perform the steps to translate their hypotheses into model implementations? What challenges do they face during this process?
- **RQ3 - Tools**: How might current software tools influence hypothesis formalization?

To sensitive ourselves to the steps (**RQ1 - Steps**) and considerations (**RQ2 - Process**) involved in hypothesis formalization, we compared and contrasted existing models and descriptions of data analysis in prior work. We augmented our deep dive into prior work with a formative content analysis of 50 randomly sampled research papers from five different venues, including Psychological Science and Nature. We find that researchers decompose their research hypotheses into specific sub-hypotheses, derive proxy variables from theory and available data, and adapt statistical analyses to account for data collection procedures. A key takeaway from prior work and the formative content analysis was the "hypothesis refinement loop" in Figure 1.

To validate and deepen our understanding of hypothesis formalization (**RQ1 - Steps** and **RQ2 - Process**), we designed and conducted a lab study in which we observed 24 analysts develop and formalize hypotheses in-situ. We find that analysts foreground implementation concerns, even when brainstorming hypotheses, and try to fit their hypotheses and analyses to prior experiences and familiar tools, suggesting a strong influence of tools (**RQ3 - Tools**). Thus, the lab study reinforced the hypothesis refinement loop, surfaced the "model implementation loop," and raised questions about the role of tools.

To identify how tools may shape hypothesis formalization (**RQ3 - Tools**), we reviewed 20 statistical software tools. We find that although the tools support nuanced model implementations, their low-level abstractions can focus analysts on statistical and computational details at the expense of higher-level reasoning about initial hypotheses. Tools also do not aid analysts in identifying reasonable model implementations that would test their conceptual hypotheses, which may explain why analysts in our lab study relied on familiar approaches, even if sub-optimal. Furthermore, our tools review confirmed that the dual processes inform one another during hypothesis formalization.

Taken together, our findings help us define the hypothesis formalization framework, as summarized in Figure 1, and suggest **three design implications** for tools to more directly support hypothesis formalization: (i) show the relationships between related statistical models that seem syntactically different from each other, (ii) provide higher-level abstractions for expressing conceptual hypotheses and partial model specifications, and (iii) develop bidirectional computational assistance for authoring causal models and relating them to statistical models.

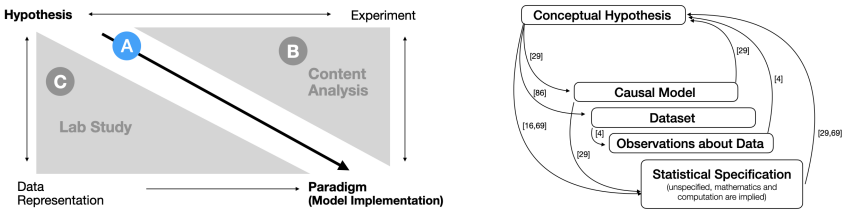

Fig. 2. **Relationship between hypothesis formalization and prior work.**
Left: Schunn and Klahr's four-space model of scientific discovery (stylized adaptation from Figure 1 in [72]), which includes unidirectional information flow from the hypothesis space to the paradigm space (which includes model implementation). Hypothesis formalization (A) is focused on a tighter integration and the information flow between hypothesis and paradigm spaces. Specifically, the information flow is bidirectional in hypothesis formalization. Our content analysis (B) and lab study (C) triangulate the four-space model to understand hypothesis formalization from complementary perspectives. *Right:* Hypothesis formalization steps also identified in prior work on theories of sensemaking, statistical thinking, and data analysis workflows (citations included to the right of the arrows). Hypothesis formalization is finer grained and involves more iterations. While prior work broadly refers to mathematical equations, partial model specifications, and computationally tuned model implementations as statistical specifications, hypothesis formalization differentiates them. As a whole, this paper provides empirical evidence for theorized loops between conceptual hypothesis and statistical specification (see Figure 1).

By defining and characterizing hypothesis formalization, we situate data analysis in a larger model of scientific discovery, identify specific problem solving strategies used in hypothesis formalization that demonstrate how data analysis (and science) is a practice, and identify opportunities for future software to improve the transparency and reproducibility of analyses by explicitly supporting pathways and loops through hypothesis formalization.

2 BACKGROUND AND RELATED WORK

Our work integrates and builds upon prior research on frameworks of scientific discovery, theories of sensemaking, statistical practices, and empirical studies of data analysts.

2.1 Dual-search Model of Scientific Discovery

Klahr and Simon characterized scientific discovery as a dual-search process involving the development and evaluation of hypotheses and experiments [43]. They posited that scientific discovery involved tasks specific to hypotheses (e.g., revising hypotheses) and to experiments (e.g., analyzing data collected from experiments), which they separated into two different "spaces," and tasks moving between them, which is where we place hypothesis formalization.

Extending Klahr and Simon's two-space model, Schunn and Klahr proposed a more granular four-space model involving data representation, hypothesis, paradigm, and experiment spaces [72, 73]. In the four-space model, conceptual hypothesizing still lies in the hypothesis space, and hypothesis testing and statistical modeling lies in the paradigm space. As such, hypothesis formalization is a process connecting the hypothesis and paradigm spaces. In Schunn and Klahr's four-space model, information flows unidirectionally from the hypothesis space to the paradigm space. Here we extend this prior research with evidence that hypothesis formalization involves both concept-to-implementation and implementation-to-concept processes. (see Figure 1). Figure 2 augments Schunn and Klahr's original diagram (Figure 1 in [72]) with annotations depicting how our content analysis of research papers and lab study triangulate a tighter dual-space search between hypothesis and paradigm spaces with a focus on hypothesis formalization. Our mixed-methods approach follows the precedent and recommendations of Klahr and Simon's [44] study of scientific discovery activities.

2.2 Theories of Sensemaking

Human beings engage in *sensemaking* to acquire new knowledge. Several theories of sensemaking [45, 66, 69] describe how and when human beings seek and integrate new data (e.g., observations, experiences, etc.) to develop their mental models about the world.

Russell et al. [69] emphasize the importance of building up and evaluating external representations of mental models, and define sensemaking as "the process of searching for a representation and encoding data in that representation to answer task-specific questions." External representations are critical because they influence the quality of conclusions reached at the end of the sensemaking process and affect how much time and effort is required in the process. Some representations may lead to insights more quickly. Russell et al. describe the iterative process of searching for and refining external representations in a "learning loop complex" that involves transitioning back and forth between (i) searching for and (ii) instantiating representations.

Grolemund and Wickham argued for statistical data analysis as a sensemaking activity [29]. They emphasize the (1) bidirectional nature of updating mental models of the world and hypotheses based on data and collecting data based on hypotheses and (2) the process of identifying and reconciling discrepancies between hypotheses and data. Their depiction of the analysis process parallels Klahr and Simon's framework of scientific discovery.

In this paper, we consider hypothesis formalization to be a learning loop [69] where the conceptual hypothesis is an external representation of a set of assumptions analysts may have about the world (e.g., an implicit causal model), that ultimately affects which models are specified and which results are obtained. We found that that there are smaller learning loops as analysts search for and revise intermediate representations, such as explicit causal models, mathematical equations, or partially specified models. The hypothesis and model refinement loops can themselves be smaller learning loops embedded in the larger loop of hypothesis formalization.

2.3 Statistical Thinking

Statistical thinking and practice require differentiating between *domain* and *statistical* questions. The American Statistical Association (ASA), a professional body representing statisticians, recommends that universities teach this fundamental principle in introductory courses (see Goal 2 in [16]).

Similarly, researchers Wild and Pfannkuch emphasize the importance of differentiating between and integrating statistical knowledge and context (or domain) knowledge when thinking statistically [63, 64, 86]. They propose a four step model for operationalizing ideas ("inklings") into plans for collecting data, which are eventually statistically analyzed. In their model, analysts must transform "inklings" into broad questions and then into precise questions that are then finally turned into a plan for data collection (see Figure 2 in [86]). Statistical and domain knowledge inform all four stages. However, it is unknown what kinds of statistical and domain knowledge are helpful, how they are used and weighed against each other, and when certain kinds of knowledge are helpful to operationalize inklings. Our work provides more granular insight into Wild and Pfannkuch's proposed model of operationalization and aims to answer when, how, and what kinds of statistical and domain knowledge are used during statistical data analysis.

More recently, in *Statistical Rethinking* [56], McElreath proposes that there are three key representational phases involved in data analysis: conceptual hypotheses, causal models underlying hypotheses (which McElreath calls "process models"), and statistical models. McElreath, like the ASA and Wild and Pfannkuch, separates domain and statistical ideas and discusses the use of causal models as an intermediate representation to connect the two. McElreath emphasizes that conceptual hypotheses may correspond to multiple causal and statistical models, and that the same statistical model may provide evidence for multiple, even contradictory, causal models and hypotheses.

McElreath's framework does not directly address how analysts navigate these relationships or how computation plays a role, both of which we take up in this paper.

Overall, our work provides empirical evidence for prior frameworks but also (i) provides more granular insight into *how* and *why* transitions between representations occur and (ii) scrutinizes the role of *software and computation* through close observation of analyst workflows in the lab as well as through a follow-up analysis of statistical software. Based on these observations, we also speculate on how tools might better support hypothesis formalization.

2.4 Empirical Studies of Data Analysts

Data analysis involves a number of tasks that involve data discovery, wrangling, profiling, modeling, and reporting [41]. Extending the findings of Kandel et al. [41], both Alspaugh et al. [1] and Wongsuphasawat et al. [87] propose exploration as a distinct task. Whereas Wongsuphasawat et al. argue that exploration should subsume discovery and profiling, Alspaugh et al. describe exploration as an alternative to modeling. The importance of exploration and its role in updating analysts' understanding of the data and their goals and hypotheses is of note, regardless of the precise order or set of tasks. Battle and Heer describe exploratory visual analysis (EVA), a subset of exploratory data analysis (EDA) where visualizations are the primary outputs and interfaces for exploring data, as encompassing both data-focused (bottom-up) and goal- or hypothesis-focused (top-down) investigations [4]. In our lab study, we found that (i) analysts explored their data before modeling and (ii) exploratory observations sometimes prompted conceptual shifts in hypotheses (bottom-up) but at other times were guided by hypotheses and only impacted statistical analyses (top-down). In this way, data exploration appears to be an important intermediate step in hypothesis formalization, blurring the lines between exploratory and confirmatory data analysis.

Decisions throughout analysis tasks can give rise to a "garden of forking paths" [25], which compounds for meta-analyses synthesizing previous findings [40]. Liu, Boukhelifa, and Eagan [50] proposed a broad framework that characterizes analysis alternatives using three different *levels of abstraction*: cognitive, artifact, and execution. *Cognitive* alternatives involve more conceptual shifts and changes (e.g., mental models, hypotheses). *Artifact* alternatives pertain to tooling (e.g., which software is used for analysis?), model (e.g., what is the general mathematical approach?), and data choices (e.g., which dataset is used?). *Execution* alternatives are closely related to artifact alternatives but are more fine-grained programmatic decisions (e.g., hyperparameter tuning). We find that hypothesis formalization involves all three levels of abstraction. We provide a more granular depiction of how these levels cooperate with one another.

Moreover, Liu, Althoff, and Heer [51] identified numerous decision points throughout the data lifecycle, which they call *end-to-end analysis*. They found that analysts often revisit key decisions during data collection, wrangling, modeling, and evaluation. Liu, Althoff, and Heer also found that researchers executed and selectively reported analyses that were already found in prior work and familiar to the research community. Hypothesis formalization is comprised of a subset of steps involved in end-to-end analysis. Thus, we expect hypothesis formalization will be an iterative process where domain norms will influence decision making. It is nonetheless valuable to provide insight into how a single iteration — from a domain-specific research question to a single instantiation of a statistical model (among many alternatives which may be subsequently explored) — occurs. Our depiction of hypothesis formalization aims to account for more domain-general steps and artifacts, but we recognize that domain expertise and norms may determine which paths and how quickly analysts move through hypothesis formalization.

In summary, our work differs in (i) scope and (ii) method from prior work in HCI on data analysis practices. Whereas hypothesis formalization has remained implicit in prior descriptions of data analysis, we explicate this specific process. While previous researchers have relied primarily on

post-analysis interviews with analysts, our lab study (Section 3) enables us to observe decision making during hypothesis formalization in-situ.

2.5 Formative Content Analysis

To complement our in-depth synthesis of prior work, we conducted a formative content analysis of 50 peer-reviewed publications from five different domains.

2.5.1 Methods. We randomly sampled ten papers published in 2019 from each of the following venues: (1) the Proceedings of the National Academy of Sciences (PNAS), (2) Nature, (3) Psychological Science (PS), (4) Journal of Financial Economics (JFE), and (5) the ACM Conference on Human Factors in Computing Systems (CHI). We sampled papers that used statistical analyses as either primary or secondary methodologies. Our sample represents a plurality of domains and recent practices. [1]

The first two authors iteratively developed a codebook to code papers at the paragraph-level. The codebook contained five broad categories: (i) research goals, (ii) data sample information, (iii) statistical analysis, (iv) results reporting, and (v) computation. Each category had more specific codes to capture more nuanced differences between papers. This tiered coding scheme enabled us to see general content patterns across papers and nuanced steps within papers. The first two authors reached substantial agreement (IRR = .69 - .72) even before resolving disagreements. The first three authors then (i) read and coded all sections of papers except the figures, tables, and auxiliary materials that did not pertain to methodology[2]; (ii) discussed and summarized the papers' goals and main findings to ensure comprehension and identify contribution types; and (iii) visualized each paper as a "reorderable matrix" [5].

We adapted Bertin's "reorderable matrix" [5], an interactive visualization technique for data exploration, in our analysis. We visualized each paper in our sample as a matrix where each row represented a code in our codebook and each column represented a coded paragraph. We fixed the order of paragraphs to match the paper's progression. We colored codes (rows) according to their categories in our codebook, repeatedly reordered the rows representing codes, and transposed the matrices to detect visual patterns in the papers. Figure 3 shows an example matrix.

The visual representation of papers' content and structure helped us notice common patterns across papers and guided our follow-up analyses and discussions about what steps (**RQ1 - Steps**) and considerations (**RQ2 - Process**) researchers reported having during hypothesis formalization. Across multiple papers, the matrices showed how researchers typically start with broader research goals that they decompose into specific hypotheses (i.e., hypothesis refinement) over the course of a paper section, for example. Within a single paper, the matrices visually showed patterns of how researchers motivated and pieced together multiple experiments and interpreted statistical results in order to make a primary scientific argument. Our supplementary materials include our codebook with definitions and examples as well as a summary, citation, and annotated matrix for each paper.

2.5.2 Findings. The content analysis confirmed prior findings on (i) the connection between hypotheses and causal models (e.g.,[56]), (ii) the importance of proxies to quantify concepts, and (iii) the constraints that data collection design and logistics place on modeling. Extending prior work,

[1] Google Scholar listed the venues among the top three in their respective areas in 2018. Venues were often clustered in the rankings without an obvious top-one, so we chose among the top three based on ease of access to publications (e.g., open access or access through our institution). Some papers were accepted and published before 2019, but the journals had included them in 2019 issues.

[2] PNAS and Nature papers included a materials and methods section after references that were distinct from extended tables, figures, and other auxiliary material. We coded the materials and methods sections in the appendices and included them in the content analysis. Our supplementary material describes our process in greater detail.

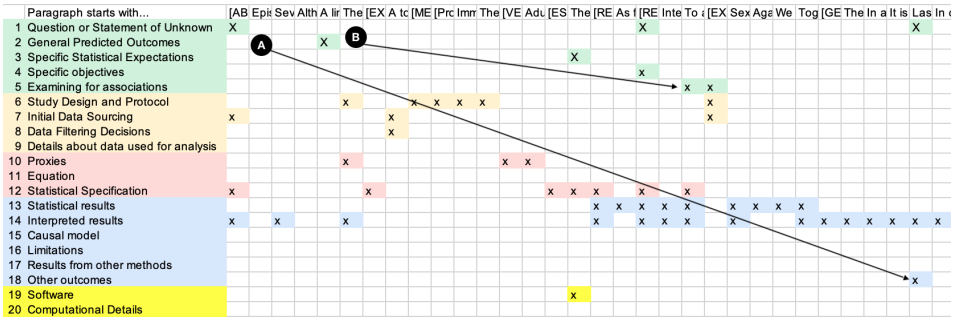

Fig. 3. **Formative content analysis: example reorderable matrix for [58].**
We visualized each paper in our sample as a "reorderable matrix" [5] to aid in detecting patterns in papers' structure and content that could indicate how researchers formalized their hypotheses. The rows represent the codes in our codebook, colored according to the five broad categories of codes: research goals (rows 1-5, green), sample information (rows 6-9, orange), statistical analysis details (rows 10-12, red), reporting of results (rows 13-18, blue), and computational details (rows 19-20, bright yellow). The columns are the paragraphs, which are indexed by their first sentences, ordered left to right. In a paragraph's column, there is an "X" for each code the paragraph received. Paragraphs have multiple codes if they contain multiple types of information. Among the ten visual patterns we noticed across our sample and subsequently looked for in each paper, two stand out in this paper. (A) As the paper progresses (visually moving left to right), the paper's focus shifts from research goals to sample information to statistical analysis to results, as indicated by the arrow labeled A. Largely expected, this pattern helps to validate our coding method. Also, there is only one paragraph that discusses statistical software. (B) Researchers discuss research goals and questions throughout the paper. Interestingly, in the middle of the paper, when the researchers discuss their goals in greater detail, the researchers discuss them in increasing specificity, as indicated by the arrow labeled B. We were able to detect this pattern across papers by iterating on how to order the research goal codes (rows 1-5, green). The final order lists codes in increasing specificity from top (row 1) to bottom (row 5). Pattern B suggests that researchers refine their hypotheses during hypothesis formalization, which may involve specifying proxies and statistical methods. Our supplementary materials discuss additional patterns in this paper and across our entire sample.

the content analysis also (i) suggested that decomposing hypotheses into specific objectives is a mechanism by which conceptual hypotheses relate to causal models; (ii) crystallized the hypothesis refinement loop involving conceptual hypotheses, causal models and proxies; and (iii) surfaced the dual-search nature of hypothesis formalization by suggesting that model implementation may shape data collection.

2.5.3 Limitations. The major limitation of analyzing published papers is the disconnect between actual and reported analytical practice. The pressures to write compelling scientific narratives [42] likely influence which aspects of hypothesis formalization are described or omitted. For instance, in practice, model implementations may constrain data collection more often than we found in our sample. Nevertheless, the lack of information in prior work and the content analysis suggests that hypothesis formalization remains an opaque process deserving of greater scrutiny. Hypothesis formalization may explain how analysts determine which tools to use and how domain expertise may influence the analytical conclusions reached.

2.6 Expected Steps in Hypothesis Formalization

Towards our first two research questions about what actions analysts take to formalize hypotheses (**RQ1 - Steps**) and why (**RQ2 - Process**), prior work and our formative content analysis suggest that hypothesis formalization involves steps in three categories: conceptual, data-based, and statistical.

Conceptually, analysts develop conceptual hypotheses and causal models about their domain that guide their data analysis. With respect to *data*, analysts explore data and incorporate insights from exploration, which can be top-down or bottom-up, into their process of formalizing hypotheses. The *statistical* concerns analysts must address involve mathematical and computational concerns, such as identifying a statistical approach (e.g., linear modeling), representing the problem mathematically (e.g., writing out a linear model equation), and then implementing those using software. In our work, we find evidence to support separating statistical considerations into concerns about mathematics, statistical specification in tools, and model implementation using tools.

A key observation about prior work is that there is a tension between iterative and linear workflows during hypothesis formalization. Although sensemaking processes involve iteration, concerns about methodological soundness, as evidenced in pre-registration efforts that require researchers to specify and follow their steps without deviation, advocate for, or even impose, more linear processes. More specifically, theories of sensemaking that draw on cognitive science, in particular [29, 69], propose larger iteration loops between conceptual and statistical considerations. Some textbooks and research concerning statistical thinking and practices [16, 86] appear less committed to iteration while other researchers and practitioners in applied statistics emphasize *workflows* for iterating on statistical models [24, 49, 88]. Workflows (e.g., model expansion) can help researchers start with simple models and build up to more complex ones by incrementally testing and refining their understanding of characterstics of the data, the model fitting algorithms, and computational settings [6, 23, 26]. Moreover, empirical work in HCI on data analysis embraces iteration during exploration and observes iteration during some phases of confirmatory data analysis, such as statistical model choice, but not in others, such as tool selection. In our work, we are sensitive to this tension and aim to provide more granular insight into iterations and linear processes involved in hypothesis formalization. We also anticipate that the steps identified in prior work will recur in our lab study, but we do not limit our investigation to these steps.

3 EXPLORATORY LAB STUDY

To address the limitation of the content analysis, understand analysts' considerations (**RQ2 - Process**) while formalizing their hypotheses (**RQ1 - Steps**), and examine the role of statistical software in this process (**RQ3 - Tools**), we designed and conducted a virtual lab study with freelance data workers who approach the hypothesis formalization and analysis process with expectations of rigor but without the pressure of publication.

3.1 Methods

Data workers: We recruited 24 data workers with experience in domains ranging from marketing to physics to education through Upwork (22) and by word of mouth (2).[3]

Twelve data workers held occupations as scientists, freelance data scientists, project managers, or software engineers. Six were currently enrolled in or had just finished graduate programs that involved data analysis. Five identified as current or recent undergraduates looking for jobs in data science. One was an educator. Data workers self-reported having significant experience on a 10-point scale adapted from a scale for programming experience [21] (min=2, max=10, mean=6.4, std=2.04) and would presumably have familiarity with hypothesis formalization.

The lab study enables us to contrast normative expert practices (found in prior work and our formative content analysis) to observed practices with data workers who are not statistical experts but

[3]We refer to our participants as data workers because they work with data but do not represent the entire population of data scientists, which may include statistical experts.

still work in real-world analysis settings (i.e., research, marketing, consulting). A benefit of studying these data workers is that they are likely to benefit most from new tools.

Protocol: We designed and conducted a lab study with three parts. Parts 1 and 3 were recorded and automatically transcribed using Zoom. We compensated data workers $45 for their time. The first author conducted the study and took notes throughout.

Part 1: Structured Tasks. Part 1 asked data workers to imagine they were leading a research team to answer the following research question: "What aspects of an individual's background and demographics are associated with income after they have graduated from high school?"[4] We asked data workers to complete the following tasks:

- *Task 1: Hypothesis generation.* Imagining they had access to any kind of data thinkable, data workers brainstormed at least three hypotheses related to the research question.
- *Task 2: Conceptual modeling.* Next, data workers saw a sample data schema and developed a conceptual model for one or more of their hypotheses. We used the term "conceptual model" instead of "causal model" to avoid (mis)leading data workers. We provided the following definition: "A conceptual model summarizes the process by which some outcome occurs. A conceptual model specifies the factors you think influence an outcome, what factors you think do not influence an outcome, and how those factors might interact to give rise to the outcome."
- *Task 3: Statistical model specification.* Finally, we presented data workers with a sample dataset and instructed them to specify but not implement a statistical model to test one or more of their hypotheses.

After the three tasks, we conducted a semi-structured interview with data workers about (i) their validity concerns[5] and (ii) experiences. To help us contextualize our observations and assess the generalizability of our findings, we asked data workers to compare the study's structure and tasks to their day-to-day data analysis practices.

Part 2: Take-home analysis. After the first Zoom session, data workers implemented their analyses using the previously shown dataset, shared any analysis artifacts (e.g., scripts, output, visualizations, etc.), and completed a survey about their implementation experience. Prior to Part 3, the first author reviewed all submitted materials and developed participant-specific questions for the final interview.

Part 3: Final Interview. The first author asked data workers to give an overview of their analysis process and describe the hypotheses they tested, how their analysis impacted their conceptual model and understanding, why they made certain implementation choices, what challenges they faced (if any), and any additional concerns about validity.

Materials: The data schema and dataset used in the study came from a publicly available dataset from the Pew Research Center [78]. Each task was presented in a separate document. All study materials are included as supplementary material.

Analysis: The first author reviewed the data workers' artifacts multiple times to analyze their content and structure; thematically analyzed notes and transcripts from data workers' Zoom sessions; and regularly discussed observations with the other authors throughout analysis.

3.2 Findings and Discussion

Eighteen of the 24 data workers we recruited completed all three parts of the study. The other six data workers completed only the first Zoom session. In our analysis, we incorporate data from all data workers for as far as they completed the study.

[4]We chose the open-ended research question about income after high school because we expected it to be widely approachable and require no domain expertise to understand.

[5]If data workers were unfamiliar with the term "validity," we rephrased the questions to be about "soundness" or "reliability."

We found that data workers had four major steps (**RQ1 - Steps**) and considerations (**RQ2 - Process**): (i) identifying or creating proxies, (ii) fitting their present analysis to familiar approaches, (iii) using their tools to specify models (**RQ3 - Tools**), and (iv) minimizing bias by relying on data. Data workers also faced challenges acquiring and incorporating domain and statistical knowledge (**RQ2 - Process**).

3.2.1 Data workers consider proxies and data collection while articulating hypotheses. We encouraged data workers to not consider the feasibility of collecting data while brainstorming hypotheses. Yet, while brainstorming hypotheses, data workers expressed concern with how to measure constructs [D2, D5, D8, D12, D18, D22, D24] and how to obtain data [D2, D6, D8, D9, D11, D21, D24].

For instance, D18, a computer science student who had worked on more than five data analysis projects, grappled with the idea of 'privilege' and how to best quantify it:

> "*I'm trying to highlight the fact that those who will be privileged before graduation...that experience will enable them to make again more money after graduation. I won't say 'privilege' because we need to quantify and qualify for that...it's just an abstract term.*"

Eventually, D18 wrote two separate hypotheses about 'privilege,' operationalizing it as parental income: (1) "People with higher incomes pre graduating, end up having higher differences between pre and post graduation incomes than those with lower incomes pre graduation." and (2) "People with parents with lower incomes tend to have lower incomes pre graduation than those with parents with higher incomes."

D18 continued to deliberate 'privilege' as measured by low and high income, saying, "*...again you need to be careful with low and high because these are just abstract terms. We need to quantify that. What does it mean to be 'low?' What does it mean to be 'high?'*". Finally, D18 decided to "*maybe use the American standards for low income and high income.*" Although an accepted "American standard" may not exist, D18 nevertheless believed that cultural context was necessary to specify because it could provide a normalizing scale to compare income during analysis, demonstrating how data workers plan ahead for statistical modeling while brainstorming and refining hypotheses.

Similarly, D2, a freelance data scientist, was very specific about how to measure personality: "More extraverted individuals (extraversion measured using the corresponding social network graph) are likely to achieve higher yearly income later in life."

In the presence of the data schema, more data workers were concerned with proxies [D2, D5, D6, D7, D8, D9, D16, D18, D21]. Some even adapted their working definitions to match the available data, similar to how researchers in the content analysis determined proxies based on data. For instance, D8, who hypothesized that "individuals interested in STEM fields tend to earn more post high school than individuals interested in other fields," operationalized "interest" as "Major" — a variable included in the data schema — even though they had previously brainstormed using other proxies such as club attendance in high school.

These data workers' closely related considerations of data and concept measurement demonstrate how conceptual hypotheses and data collection may inform each other, corroborating our findings from the content analysis.

3.2.2 Data workers consider implementation and tools when specifying statistical models. When we asked data workers to specify their models without considering implementation, we anticipated they would name specific statistical tests (e.g., "ANOVA"), approaches (e.g., "linear regression" or "decision trees"), or write mathematical models (e.g., $Y = B_0 + B_1X_{age} + B_2X_{gender}$) that they could then implement using their tools because (a) some researchers in the literature survey did so in their papers and (b) several data workers mentioned having years of analysis experience. However, despite the explicit instruction to disregard implementation, 16 data workers provided to-do lists or

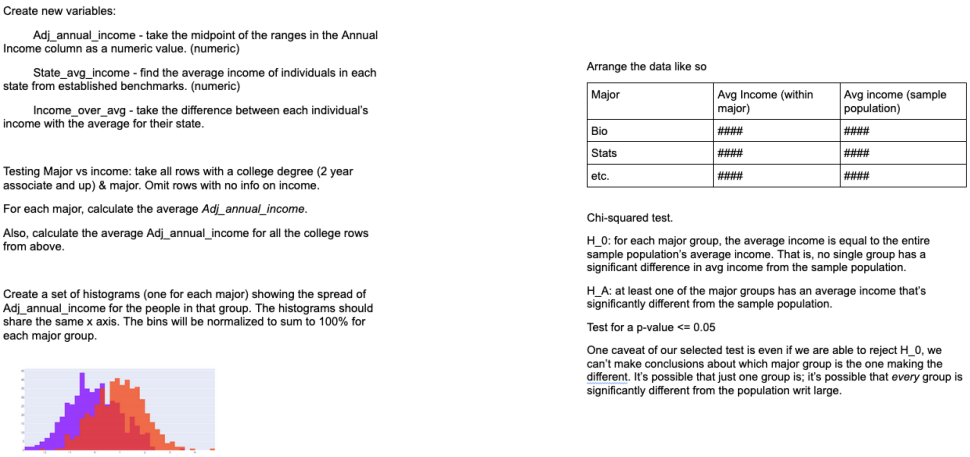

Fig. 4. **Sample statistical specification (D8).**

The lab study tasked analysts to specify their statistical models without considering implementation. We expected analysts would represent their statistical models using statistical test names or mathematical equations. Instead, most analysts specified statistical procedures for performing statistical models using todo lists and summaries of steps, which sometimes included mentions of software tools, showing that implementation was an important consideration and that tool familiarity may limit which statistical models analysts consider and implement. Data worker D8 specified their model through a combination of statistical test names (e.g., Chi-squared test) and a list (split across two pages) of detailed steps involved in creating new variables, cleaning and wrangling data, visualizing data, and testing their hypothesis.

summaries of steps to perform a statistical analysis as their model specifications [D1, D2, D3, D5, D7, D8, D9, D11, D12, D14, D16, D18, D20, D21, D22, D23, D24]. Of these 16 data workers, eight also named specific statistical tests in their descriptions [D3, D7, D8, D11, D12, D14, D18, D20].

For example, D8, a data science consultant with 7/10 analysis experience, specified a list of steps that included creating new variables that aggregated columns in the dataset, cleaning and wrangling the data, visualizing histograms, performing chi-squared test, and interpreting the statistical results. Notably, D8 also specified null and alternative hypotheses, which acted as an intermediate artifact during hypothesis formalization. Figure 4 shows D8's statistical specification.

Only four data workers named specific statistical methods without describing their steps [D4, D6, D15, D17]. Two data workers, D22, a neuroscientist by training with 8/10 analysis experience, and D19, an educator with 6/10 analysis experience, attempted to specify their models mathematically. D22 used the familiar R syntax: "Current Income ~ Educational attainment + Gender + Interactions of those two." On the other hand, D19 gave up because although they knew the general form of logistic regression, they did not know how to represent the specific variables in the model they wanted to perform.

The implementation and software details data workers discussed and included in their specifications suggest that data workers prefer to skip over mathematical equations and jump to specification and implementation in their tools. Although it is possible that study instructions primed data workers to respond about how they would perform, rather than represent, the task even after researcher clarifications, this would not explain the level of implementation detail data workers included. Nine data workers went so far as to mention specific libraries, even functions, that they would use to program their analyses [D3, D9, D12, D13, D14, D16, D19, D21, D23]. In their reflective interviews,

data workers also expressed that they often do not specify models outside of implementing them, which D19 succinctly described:

> "*I don't normally write this down because all of this is in a [software] library.*"

Data workers' statistical knowledge appears to be situated in the programs they write, and their knowledge of and familiarity with tools constrains the statistical methods they explore and consider. As such, tools may be a key point of intervention for guiding data workers toward statistical methods that may be unfamiliar but are best suited for their conceptual hypotheses.

3.2.3 Data workers try to fit analyses to previous projects and familiar approaches. Data workers spent significant thought and time categorizing their analyses as "prediction," "classification," or "correlation" problems [D2, D3, D7, D10, D11, D18, D19, D21, D22]. To categorize, data workers relied on their previous projects. While reflecting on their typical analysis process, D21, a software engineer working in healthcare, said (emphasis added),

> "*I usually tend to jump...to look at data and* ***match [the analysis problem] with similar patterns*** *I have seen in the past and start implementing that or do some rough diagrams [for thinking about parameters, data type, and implementation] on paper...and start implementing it.*"

Data workers also looked at variable data types (i.e., categorical or continuous) to categorize. For example, D3, a freelance analyst, pivoted from thinking about **predicting** income to **classifying** income groups (emphasis added) based on data type information:

> "*The income, the column, the target value here, is categorical. I think maybe it wouldn't be a bad idea to see what* ***classification*** *tasks, what we could do. So instead of trying to* ***predict*** *because we're not trying to* ***predict an exact number****, it seems...like more of a* ***classification*** *problem...*"

A provocative case of adhering to prior experiences was D6, a psychological research scientist. Although several data workers were surprised and frustrated that income was ordinal in the dataset with categories such as"Under $10K," "$10K to $20K," "$20K to $30K," up to "150K+", none went so far as D6 to synthetically generate normally distributed income data so that they could implement the linear regression models they had specified despite saying they knew that income was not normally distributed.

When asked further about the importance of normal data, D6 described how they plan analyses based on having normal data, strive to collect normally distributed, and rely on domain knowledge to transform the data to be normal when it may not be after collection:

> "*...I feel like having non normal data is something that's like hard for us to deal with. Like it just kind of messes everything up like. And I know, I know it's not always assumption of all the tasks, but just that we tend to try really hard to get our variables to be normally distributed. So, you know, we might like transform it or, you know, kind of clean it like clean outliers, maybe transform if needed...I mean, it makes sense because like a lot of measures we do use are like depressive symptoms or anxiety symptoms and kind of they're naturally normally distributed...I can probably count on my hand the number of non parametric tests I've like included in manuscripts.*"

D6's description of their day-to-day analyses exemplifies the dual-search nature of hypothesis formalization: Data workers (i) jump from hypothesis refinement to model specification or implementation with specific proxies in mind and then (ii) collect and manipulate their data to fit their model choices.

We recognize that data workers may have taken shortcuts for the study they would not typically make in real life. Nevertheless, the constraints we imposed by using a real-world dataset are to be expected in real-world analyses. Therefore, our observations still suggest that rather than consider the

nature and structure of their hypotheses and data to inform using new statistical approaches, which statistical pedagogy and theory may suggest, data workers may choose familiar statistical approaches and mold their new analyses after previous ones.

3.2.4 Data workers try to minimize their biases by focusing on data. Throughout the study, data workers expressed concern that they were biasing the analysis process. Data workers drew upon their personal experiences to develop hypotheses [D5, D10, D13, D15, D16, D20, D21, D24] and conceptual models [D8, D12, D20, D24]. D12, a data analysis project manager, described how their personal experiences may subconsciously bias their investigation by comparing a hypothetical physicist and social worker answering the same research question:

> "*Whereas a social worker by design...they're meant to look at the humanity behind the numbers [unlike a physicist]. So like, they may actually end up with different results...actually sitting in front of this data, trying to model it.*"

A few data workers even refused to specify conceptual models for fear of biasing the statistical analyses [D10, D11, D19]. On the surface, data workers resisted because they believed that some relationships, such as the effect of age on income, were too "obvious" and did not warrant documentation [D10, D11]. However, relationships between variables that were "obvious" to some data workers were not to others. For instance, D10, a business analyst, described how income would plateau with age, but other data workers, such as D18, assumed income would monotonically increase with age.

When we probed further into why D10, D11, and D19 rejected a priori conceptual models, they echoed D10's belief that conceptual models "put blinders on you." Even the data workers who created conceptual models echoed similar concerns of wanting to "[l]et the model do the talking" in their implementations [D3, D15, D18, D19]. Instead of conceptual modeling, D10 chose to look at all n-ary relationships in the dataset to determine which variables to keep in a final statistical model, saying,

> "*It's so easy to run individual tests...You can run hypothesis tests faster than you can actually think of what the hypothesis might be so there's no need to really presuppose what relationships might exist [in a conceptual model].*"

Of course, one could start from the same premise that statistical tests are so easy to execute and conclude that conceptual modeling is all the more important to prioritize analyses and prevent false discoveries.

Similarly, data workers were split on whether they focused their implementation exclusively on their hypotheses or examined other relationships in the dataset opportunistically. Nine data workers stuck strictly to testing their hypotheses [D1, D4, D5, D6, D7, D11, D13, D20, D24]. However, five data workers were more focused on exploring relationships in the dataset and pushed their hypotheses aside [D2, D3, D10, D16, D18], and an additional four data workers explored relationships among variables not previously specified in their hypotheses in addition to their hypotheses [D14, D15, D17, D21]. D18 justified their choice to ignore their hypotheses and focus on emergent relationships in the data by saying that they wanted to be "*open minded based on the data...open to possibilities.*"

Data workers' concerns about bias and choice of which relationships to analyze (hypothesis only vs. opportunistic) highlight the tension between the two searches involved in hypothesis formalization: concept-first model implementations and implementation-first conceptual understanding. Conceptual models are intermediate artifacts that could reconcile the two search processes and challenge data workers' ideas of what "data-driven" means. However, given some data workers' resistance to prior conceptual modeling, workflows that help data workers conceptually model as a way to reflect on their model implementations and personal biases may be more promising than ones that require them before implementation.

3.2.5 Data workers face challenges obtaining and integrating conceptual and statistical information. Based on data workers' information search behaviors and self-reports, we found that data workers faced challenges obtaining and integrating both domain and statistical knowledge.

Data workers consulted outside resources such as API documentation, Wikipedia, and the *Towards Data Science* blog throughout the study: one while brainstorming hypotheses [D13]; three while conceptual modeling [D12, D13, D22]; six while specifying statistical models [D3, D6, D12, D13]. Six data workers also mentioned consulting outside resources while implementing their analyses [D1, D3, D11, D14, D15, D21]. By far, statistical help was the most common.

Furthermore, when data workers reflected on their prior data analysis experiences, they detailed how collaborators provided domain and statistical expertise that are instrumental in formalizing hypotheses. Collaborators share data that help domain experts generate hypotheses [D9], critique and revise conceptual models and proxies [D4, D8], answer critical data quality questions [D10], and ensure statistical methods are appropriate [D5, D6, D22].

In the survey participants completed after implementing their analyses, the three most commonly reported challenges were (i) **formatting** the data [D1, D4, D5, D6, D13, D16, D18, D20, D21, D24], (ii) **identifying** which statistical analyses to perform with the data to test their hypotheses [D1, D11, D14, D18, D20, D21], and (iii) **implementing and executing** analyses using their tools [D1, D6, D7, D13, D20, D21]. Although we expected data workers would have difficulty wrangling their data based on prior work [41], we were surprised that identifying and executing statistical tests were also prevalent problems given that (a) data workers were relatively experienced and (b) could choose their tools. These results, together with our observations that data workers rely on their prior experiences and tools, suggest that data workers have difficulty adapting to new scenarios where new tools and statistical approaches may be necessary.

3.3 Takeaways from the Lab Study

After the first session, 13 out of the 24 data workers described all the tasks as familiar, and 10 described most of the tasks and process as familiar. Data workers commonly remarked that although the process was familiar, the order of the tasks was "opposite" of their usual workflows. In practice, data workers may start with model implementation before articulating conceptual hypotheses, which opposes the direction of data analysis that the ASA recommends [16]. Nevertheless, our observations reinforce the dual-search, non-linear nature of hypothesis formalization.

Moreover, one data worker, D24, a physics researcher who primarily conducted simulation-based studies expressed that the study and its structure felt foreign, especially because they had no control over data collection. Other data workers in the study also described the importance of designing and conducting data collection as part of their hypothesis formalization process [D4, D6, D9]. Designing data collection methods informs the statistical models data workers plan to use and helps to refine their conceptual hypotheses by requiring data workers to identify proxies and the feasibility of collecting the proxy measures, reinforcing what we saw in the content analysis. The remarks also suggest that disciplines practice variations of the hypothesis formalization process we identify based on discipline-specific data collection norms and constraints. For example, simulating data may sometimes take less time than collecting human subjects data, so data workers working with simulations may dive into modeling and data whereas others may need to plan experiments for a longer period of time.

Approximately half of the data workers had either just finished or were enrolled in undergraduate or graduate programs involving data analysis. As such, half of our sample likely has limited professional experience outside of their studies and/or freelance work on Upwork. Additionally, data work available on Upwork may be more narrowly focused and less representative of end-to-end data analysis or research projects expected of those with greater statistical expertise. Still, several data

workers in our study mentioned other employments where they gained professional experience working on larger analysis and research projects. Despite the limitations of recruiting participants from Upwork and word of mouth, our sample represents data workers who have training in a diversity of disciplines (e.g., medicine, psychology, business), are familiar with a range of statistical methods, and have experience using a broad range of statistical tools. As such, the data workers in our study may be representative of analysts who are likely to benefit most from new tools for supporting hypothesis formalization.

Finally, we found that data workers relied on prior experiences and tools to specify and formalize their hypotheses. Tools that scaffold the hypothesis formalization process by suggesting statistical models that operationalize the conceptual hypotheses, conceptual models, or partial specifications data workers create along the way may (i) nudge data workers towards more robust analyses that test their hypotheses, (ii) overcome limitations of data workers' prior experiences, and (iii) even expand data workers' statistical knowledge. Thus, we investigated how current tool designs serve (or under-serve) hypothesis formalization.

4 ANALYSIS OF SOFTWARE TOOLS

To understand how the design of statistical computing tools may support or hinder hypothesis formalization (**RQ3 - Tools**), we analyzed widely used software packages and suites. Throughout, we use the term "package" to refer to a set of programs that must be invoked through code, such as `lme4`, `scipy`, and `statsmodels`. We use the term "suite" to refer to a collection of packages that end-users can access either through code or graphical user interfaces (GUIs), such as SPSS, SAS, and JMP. We use the term "tool" to refer to both. Software packages were a unit of analysis because they are necessary for model implementation regardless of medium (e.g., computational notebook, CoLab, RStudio). As such, our findings apply to tools that provide wrappers around packages included in our sample.

4.1 Method

Sample: Our sampling procedure involved two phases: (i) identifying software packages and suites for model implementation (not visual analysis tools like Tableau) mentioned more than once across the content analysis and lab study and (ii) adding recommended packages and suites from online data science communities our lab participants mentioned or used (e.g., *Towards Data Science*). To identify these additional tools, we consulted online data analysis fora [8, 9, 28, 67]. The final sample included 20 statistical tools: 14 packages (R: 10, Python: 4); three suites that support in-tool programming; and three suites that do not support programming. Table 1 contains an overview of our sample and results.

Analysis: Four specific questions guided our analysis:

- **Specialization:** Data workers in the lab study eagerly named specific statistical tools they would use and looked up tool documentation during the tasks. This prompted us to ask, *How specialized are the tools, and how might specialization (or lack thereof) affect how end-users discover and use them to formalize hypotheses?*
- **Statistical Taxonomies:** Data workers in the lab study tried to mold their analyses to prior experiences and their taxonomies of statistical methods. We wondered what role tools play in this: *How do tools organize and group statistical models? How might tool organization and end-users' taxonomies interplay during hypothesis formalization?*
- **Model Expression:** Data workers in the lab study jumped to model implementation throughout the tasks. Only half provided names of statistical methods. We wondered if this was due to how tools enable end-users to express their models: *What notation must end-users use to express models in the tools?*

- **Computational Issues:** Data workers in the lab study described their statistical models using specific function calls. Similarly, although it was uncommon for researchers in the content analysis to specify the software tools they used, when they did, researchers specified the functions, parameters, and settings used. This prompted us to wonder about the importance of computational settings: *What specific kinds of computational control do tools provide end-users and how might that impact hypothesis formalization?*

To answer the four questions for each statistical tool, the first author read and took notes on published articles about tools' designs and implementations, API documentation and reference manuals, and available source code; followed online tutorials; consulted question-and-answer sites (e.g., StackExchange) when necessary; and analyzed sample data with the tools. The first author paid particular attention to tool organization, programming idioms, functions and their parameters, and tool failure cases. Table 1 contains citations for resources consulted in the analysis. The iterative analysis process involved discussions among the co-authors about how to evaluate the properties of tools from our perspectives as both tool designers/maintainers and end-users. Here, we focus on end-user (hereafter referred to as analyst) perspectives informed by our lab study and make callouts to details relevant for tool designers.

4.2 Findings and Discussion

We discuss our findings in light of our characterization of hypothesis formalization in Figure 1. We refer to specific steps and transitions in Figure 1 in **boldface**.

4.2.1 Specialization. Half the tools [T2, T3, T4, T5, T6, T7, T8, T9, T11, T12] in our sample are specialized in the scope of statistical analysis methods they support (e.g., `brms` supports Bayesian generalized linear multilevel modeling). `edgeR` [T3] provides multiple modeling methods but is specialized to the context of biological count data. Such specialized tools are vital to creating a widely adopted statistical computing ecosystem, such as R.

Despite its importance, tool specialization pushes computational concerns higher up the hypothesis formalization process. Specialized tools require analysts to consider computational settings while picking a statistical tool and, possibly, even while mathematically relating their variables. They fuse the last two steps of hypothesis formalization (**Statistical Specification** and **Model Implementation**). Ultimately, specialization requires analysts to have more (i) computational knowledge and (ii) foresight about their model implementations at the cost of focusing on conceptual or data-related concerns early in hypothesis formalization.

One way tool designers minimize the requisite computational knowledge and foresight while providing the benefits of specialized packages — which may be optimal for specific statistical models or data analysis tasks — is to provide micro-ecosystems of packages. For example, R's `tidymodels` [48] and `tidyverse` [85] create micro-ecosystems that use consistent API syntax and semantics across interoperable packages. They also push analysts towards what the tool designers believe to be best practices, such as the use of the tidy data format [84]. Tools that aim to support hypothesis formalization may consider fitting into or creating micro-ecosystems that provide tool support all along the process, focusing analysts on concepts, data, or model implementation at various points.

4.2.2 Statistical taxonomies. A consequence of tool specialization is the fragmented view of statistical approaches. For example, we observed analysts in the lab study who viewed the analysis as a classification task gravitate towards machine learning-focused libraries, such as `RandomForest` [T9], `Keras` [T11], and `scikit-learn` [T12]. Because classification can be implemented as logistic regression, any tool that supports logistic regression, such as the core `stats` library in R [T10], provides equally valid, alternative perspectives on the same analysis and hypothesis. However, tools

Table 1. **Overview of the software tools included in our analysis.**
Half of the tools are specialized for specific modeling use cases. Most tools use mathematical notation (T18–T20 (✓*) even use mathematical notation in their GUIs). Most tools also provide a wide range of computational control although sometimes they require additional packages [T5, T13]. Tool specialization, organization, notation, and computational control focus analysts on model implementation details, sometimes at the expense of focusing on their conceptual hypotheses.

ID	Tool name	Specialized Scope	Mathematical Notation	Computational Control	References
R Packages					
T1	MASS	—	✓	✓	[68]
T2	brms	✓	✓	✓	[13, 14]
T3	edgeR	✓	✓	✓	[17, 18]
T4	glmmTMB	✓	✓	✓	[11, 55]
T5	glmnet	✓	—	✓(additional)	[22, 32]
T6	lme4	✓	✓	✓	[2, 3]
T7	MCMCglmm	✓	✓	✓	[30, 31]
T8	nlme	✓	✓	✓	[65]
T9	RandomForest	✓	✓	✓(minimal)	[10]
T10	stats (core library)	—	✓	✓	[79]
Python Packages					
T11	Keras	✓	—	✓(minimal)	[19]
T12	Scikit-learn	✓	—	✓	[12, 61, 74]
T13	Scipy (scipy.stats)	—	—	✓(additional)	[36–38]
T14	Statsmodels	—	✓	—	[62, 75]
Suites, with DSLs for programming					
T15	Matlab (Statistics and ML Toolbox)	—	—	✓	[80, 81]
T16	SPSS	—	✓	✓	[76]
T17	Stata	—	✓	—	[52, 53, 77]
Suites, without programming					
T18	GraphPrism	—	✓*	✓	[27]
T19	JASP	—	✓*	—	[60]
T20	JMP	—	✓*	—	[35, 71]

obfuscate these connections and do not aid analysts in considering reasonable statistical models that may be unfamiliar or outside their personal taxonomy. This may explain why analysts adhered to their personal taxonomies during the lab study.

This problem carries over to tools that support numerous statistical methods. Ten tools in our sample intend to provide more comprehensive statistical support [T1, T10, T13, T14, T15, T16, T17, T18, T19, T20]. These tools group statistical approaches using brittle and inconsistent taxonomies based on data types [T17]; analysis classes that are both highly specific (e.g., "Item Response Theory") and vague (e.g., "Multivariate analyses") [T15, T16, T17, T18, T19, T20]; and disciplines or applications (e.g., "Epidemiology and related," "Direct Marketing") [T16, T17, T20]. Although well-intended to simplify statistical method selection, tools' taxonomies are at times misleading. For instance, JMP combines various linear models into a "Fit Model" option that is separate from "Predictive Modeling" and "Specialized Modeling," which are also distinct from the more general "Multivariate Methods." Once analysts select the "Fit Model" option, they can specify the "Personality" of their model as "Generalized Regression," "Generalized Linear Model," or "Partial Least Squares,"

among many others. This JMP menu structure implies that (i) a Partial Least Squares model is distinct from a regression model when it is in fact a type of regression model and (ii) regression is not useful for prediction, which is not the case.

In these ways, tools add a "Navigate taxonomies" step before the **Statistical Specification** step, requiring analysts to match their conceptual hypotheses with the tools' taxonomies, which may misalign with their personal taxonomies. One reason for this issue may be that tools do not leverage analysts' intermediate artifacts or understanding during hypothesis formalization. By the time analysts transition to **Statistical Specification**, they have refined their conceptual hypotheses, developed causal models, and made observations about data. However, tools' taxonomies require analysts to set these aside and consider another set of decisions imposed by tool-specific groupings of statistical methods. In this way, tool taxonomies may introduce challenges that detract from hypothesis formalization.

4.2.3 Model expression: Syntax and semantics. Fifteen tools in our sample provide analysts with interfaces that use mathematical notation to express statistical models [T1, T2, T3, T4, T6, T7, T8, T9, T10, T14, T16, T17, T18, T19, T20]. R and Python packages use symbolic mathematical syntax, and SPSS and Stata use natural language-like syntax. Expressing a linear model with Sex, Race, and their interaction as predictors of Annual Income involves the formula `AnnualIncome ~ Sex + Race + Sex*Race` in `lme4` and `AnnualIncome BY Sex Race Sex*Race` in SPSS. In a linear execution of steps involved in hypothesis formalization where analysts relate variables mathematically (**Mathematical Equation**) before specifying and implementing models using tools (**Statistical Specification**, **Model Implementation**), the mathematical interfaces match analysts' progression. However, in the lab study, analysts did not specify their models mathematically even when given the opportunity, suggesting that mathematical syntax may not adequately capture analysts' conceptual or statistical considerations.

Syntactic similarity between packages may lower the barrier to trying and adopting new statistical approaches that more directly test hypotheses and therefore benefit hypothesis formalization. At the same time, syntactic similarity may also introduce unmet expectations of semantic similarity. For example, `brms` [T2] uses the same formula syntax as `lme4` [T6], smoothing the transition between linear modeling and Bayesian linear modeling for analysts. However, based on syntactic similarity, analysts may incorrectly assume statistical equivalence in computed model values. For example, in `brms`, the model intercept is the mean of the posterior when all the independent variables are at *their means*, but in `lme4`, the intercept is the mean of the model when all the independent variables are at *zero*.

Conversely, tools introduce syntactic differences between statistical approaches that are for the most part semantically equivalent, which may lead to additional challenges in hypothesis formalization. For instance, an ANOVA with repeated measures and a linear mixed effects model are similar in intent but require two different function calls, one without a formula (e.g., `AnovaRM` in `statsmodels` [T14]) and another with (e.g., `mixedlm` in `statsmodels` [T14]). Even when considering only ANOVA, tools may provide similar syntax but implement different sums of squares procedures for partitioning variance (i.e., Type I, Type II, or Type III).[6] By default, R's `stats` core package [T10] uses Type I, `statsmodels` [T14] uses Type II, and `SPSS` [T16] uses Type III. The three different sum of squares procedures lead to different F-statistics and p-values, which may lead analysts to different conclusions. More importantly, the procedures encode different conceptual hypotheses. If analysts have theoretical knowledge or conceptual hypotheses about the order of independent variables,

[6]Type I is (a) sensitive to the order in which independent variables are specified because it assigns variance sequentially and (b) allows interaction terms. Type II (a) does not assign variance sequentially and (b) does not allow interaction terms. Type III (a) does not assign variance sequentially and (b) allows interaction terms. For an easy-to-understand blog post, see [46].

tools defaulting to Type I (e.g., R's `stats` core library) align the model implementation with the conceptual hypotheses. However, if analysts do not have such conceptual hypotheses, tools' default behavior would execute (without error) and silently respond to a conceptual hypothesis different from the one the analyst seeks to test. In this way, syntactic and semantic mismatches can create a rift between model implementations and conceptual hypotheses. Furthermore, the impact of tools' "invisible" model implementation choices reinforces the interplay between conceptual and model implementation concerns during hypothesis formalization.

4.2.4 Computational issues. Tools provide end-users with options for optimizers and solvers used to fit statistical models [T1, T2, T4, T6, T7, T8, T10, T11, T13, T16, T18], convergence criteria used for fitting models [T3, T6, T16, T18], and memory and CPU allocation [T2, T5, T12, T15], among more specific customizations. For instance, `lme4` [T6] allows analysts to specify the nonlinear optimizer and its settings (e.g., the number of iterations, convergence criteria, etc.) used to fit models. In `brms` [T2], analysts can also specify the number of CPUs to dedicate to fitting their models. Some computational settings are akin to performance optimizations, affecting computer utilization but not the results. However, not all computational changes are so well-isolated.

For example, the failure of a model's inference algorithm to converge (in **Model Implementation**) may prompt mathematical re-formulation (**Mathematical Equation**), which may cast **Observations about Data** in a new light, prompting **Causal Model** and **Conceptual Hypothesis** revision. In other words, computational failures and decisions may bubble up to conceptual hypothesis revision and refinement, which may then trickle back down to model implementation iteration, and so on. In this way, computational control can be another entry into the dual-search process of hypothesis formalization.

In theory this low-level control could help analysts formalize nuanced conceptual hypotheses in diverse computational environments. However, we found that tools do not currently provide feedback on the ramifications of these computational changes, introducing a gulf of evaluation [59]. Analysts can easily change parameters to fine-tune their computational settings, but how they should interpret their model implementations and revisions conceptually is unaddressed, suggesting opportunities for future tools to bridge the conceptual and model implementation gap.

4.3 Takeaways from the Analysis of Tools

Taken together, our analysis shows that tools can support a wide range of statistical models but expect analysts to have more statistical expertise than may be realistic. They provide limited guidance for analysts (i) to express and translate their conceptual and partially-formalized concerns and (ii) identify reasonable models. Tools also provide little-to-no feedback on the conceptual ramifications of model implementation iterations. These gaps reveal a misalignment between analysts' hypothesis formalization processes and tools' expectations and design. Possible reasons for this mismatch may be that tools do not scaffold or embody the dual-search nature of hypothesis formalization or leverage all the intermediate artifacts analysts may create (e.g., refined conceptual hypotheses, causal models, data observations, partial specifications, etc.) throughout the process.

5 IMPLICATIONS

Our findings suggest three opportunities for tools to facilitate the dual-search process and align conceptual hypotheses with statistical model implementations at various stages of hypothesis formalization.

5.1 Meta-libraries: Connecting Model Implementations with Mathematical Equations

Specialized tools, although necessary for sophisticated statistical computation, require a steep learning curve. *Meta-libraries* could allow analysts to specify their models in high-level code; execute the

models using the appropriate libraries in their knowledge bases; and then output library information, functions invoked, any computational settings used, the mathematical model that is approximated, and the model results. Libraries such as Parsnip [47] have begun to provide a unified higher-level interface that allows analysts to specify a statistical model using more "generically" named functions, parameter names, and symbolic formulae (when necessary). Parsnip then compiles and invokes various library-specific functions for the same statistical model.

Probabilistic programming languages (PPLs), such as Pyro [7], Stan [15], BUGS [54], PyMC [70], already enable the development of meta-libraries. PPLs support modular specification of data, probabilistic models, and probabilistic hypotheses. Existing libraries, including `brms`, provide higher-level APIs whose syntax uses symbolic formulae, for instance, and compile to programs in a PPL (i.e., Stan in the case of `brms`).

As already seen in Parsnip and tools using PPLs, meta-libraries could bring three benefits. First, they would provide simpler, less fragmented interfaces to analysts while continuing to take advantage of tool specialization. Second, meta-libraries that output complete mathematical representations would more tightly couple mathematical representations with implementations, providing an on-ramp for analysts to expand their statistical knowledge. Third, meta-libraries that show the mathematical representations alongside underlying libraries' function calls could show syntactical variation in underlying libraries, indirectly teaching analysts how they might express their statistical models in other tools, familiarizing analysts with new tools and models, and even mend fragmented views of identical models (e.g., ANOVA and regression).

Future meta-libraries could consider providing a higher-level, declarative interface that does not require analysts to write symbolic formulae. Designing such declarative meta-libraries would require formative elicitation studies (similar to natural programming studies such as [82]) on declarative primitives that are memorable, distinguishable, and reliably understood. An additional challenge would lie in maintaining support for various libraries executed under the hood, especially as libraries change their APIs, which would strengthen the case for meta-libraries. Although meta-libraries would not solve the problems involved in understanding how computational settings affect model execution or conceptual hypotheses, they could nevertheless provide scaffolding for analysts to more closely examine specific libraries, especially if multiple libraries execute the same model but do not all encounter the same computational bottlenecks.

5.2 High-level Libraries: Expressing Conceptual Hypotheses to Bootstrap Model Implementations

The absence of tools for directly expressing conceptual hypotheses may be an explanation for why data workers in the lab study dove into model implementation details. High-level libraries could allow analysts to specify data collection design (e.g., independent variables, dependent variables, controlled effects, possible random effects); variable data types; expected or known covariance relationships based on domain expertise; and hypothesized findings in a library-specific grammar. High-level libraries could compile these conceptual and data declarations into weighted constraints that represent the applicability of various statistical approaches, in a fashion similar to Tea [39], a domain-specific language for automatically selecting appropriate statistical analyses for common hypothesis tests. Libraries could then execute the appropriate statistical approaches, possibly by using a meta-library as described above.

In addition to questions of how to represent a robust taxonomy of statistical approaches computationally, another key challenge for developing high-level libraries is identifying a set of minimal yet complete primitives that are useful and usable for analysts to express information that is usually expressed at different levels of abstraction: conceptual hypotheses, study designs, and possibly even partial statistical model specifications. For instance, even if a conceptual hypothesis is expressible

in a library, it may be impossible to answer with a study design or partial statistical model that is expressed in the same program. An approach may be to draw upon and integrate aspects from existing high-level libraries and systems that aim to address separate steps of the hypothesis formalization process, such as Touchstone2 [20] for study design and Tea and Statsplorer [83] for statistical analysis.

5.3 Bidirectional Conceptual Modeling: Co-authoring Conceptual Models and Model Implementations

Conceptual, or causal, modeling was difficult for the analysts in the lab study. Some even resisted conceptual modeling for fear of biasing their analyses. Yet, implicit conceptual models were evident in the hypotheses analysts chose to implement and the sub-hypotheses researchers articulated in the content analysis.

Mixed-initiative systems that make explicit the connection between conceptual models and statistical model implementations could facilitate hypothesis formalization from either search process and allow analysts to reflect on their analyses without fear of bias. For example, a mixed-initiative programming environment could allow analysts to write an analysis script, detect data variables in the analysis scripts, identify how groups of variables co-occur in statistical models, and then visualize conceptual models as graphs where the nodes represent variables and the edges represent relationships. The automatically generated conceptual models would serve as templates that analysts could then manipulate and update to better reflect their internal conceptual models by specifying the kind of relationship between variables (e.g., correlation, linear model, etc.) and assigning any statistical model roles (e.g., independent variable, dependent variable). As analysts update the visual conceptual models, they could evaluate script changes the system proposes. In this way, analysts could externally represent their causal models while authoring analysis scripts and vice versa.

Although bidirectional programming environments already exist for vector graphics creation [33], they have yet to be realized in mainstream data analysis tools. To realize bidirectional, automatic conceptual modeling, researchers would need to address important questions about (i) the visual grammar, which would likely borrow heavily from the causal modeling literature; (ii) program analysis techniques for identifying variables and defining co-occurrences (e.g., line-based vs. function-based) in a way that generalizes to multiple statistical libraries; and (iii) adoption, as analysts who may benefit most from such tools (likely domain non-experts) may be the most resistant to tools that limit the number of "insights" they take away from an analysis.

6 DISCUSSION, LIMITATIONS, AND FUTURE WORK

Hypothesis formalization is a dual-search process of translating conceptual hypotheses into statistical model implementations. Due to constraints imposed by domain expertise, data, and tool familiarity, the same conceptual hypothesis may be formalized into different model implementations. A single model implementation may be useful for making multiple statistical inferences. The same model implementation may also formalize two possibly opposing hypotheses. To navigate these constraints, analysts use problem-solving strategies characteristic of the larger scientific discovery process [43, 72]. As such, hypothesis formalization exemplifies how data science is a design practice.

At a conceptual level, hypothesis formalization involves *hypothesis refinement*, which, to use Schunn and Klahr's language [72], is a *scoping* process. In the formative content analysis, we found that researchers *decomposed* their research goals and conceptual hypotheses into specific, testable sub-hypotheses and *concretized* constructs using proxies, born of theory or available data. Also, we found that analysts in the lab study also quickly converged on the need to specify established proxies or develop them based on the data schema presented. In hypothesis formalization, scoping incorporates domain- and data-specific observations to qualify the conceptual scope of researchers'

hypotheses. In other words, hypothesis refinement is an instance of *means-end analysis* [57], a problem-solving strategy that aims to recursively change the current state of a problem into sub-goals (i.e., increasingly specific objectives) in order to apply a technique (i.e., a particular statistical model) to solve the problem (i.e., test a hypothesis).

At the other computational endpoint of hypothesis formalization, *model implementation* also involves iteration. Through our analysis of software tools, we found that analysts must not only select tools among an array of specialized and general choices but also navigate tool-specific taxonomies of statistical approaches. These tool taxonomies may both differ from and inform analysts' personal categorizations, potentially explaining why analysts in our lab study relied on their personal taxonomies and tools. Based on their prior experience, analysts engage in *analogical reasoning* [34], finding parallels between the present analysis problem's structure and previously encountered ones or ones that fit a tool's design easily.

Upon selecting a statistical function, analysts may tune computational settings, choose different statistical functions or approaches, which they may tune, and so on. In this way, the model implementation loop in hypothesis formalization captures the "debugging cycles" analysts encounter, such as the census researcher in the introduction. The tool ecosystem as a whole supports diverse model implementations, even for the same mathematical equation. However, the tool interfaces provide low-level abstractions, such as interfaces using mathematical formulae that, based on our observations in the lab study, do not support the kind of higher-level conceptual reasoning required of hypothesis formalization.

The steps, considerations, and strategies we have identified are domain-general. Domain-specific expertise likely influences how quickly analysts switch between steps and strategies during the dual-search process. Domain experts, including researchers in our content analysis, may know which statistical model implementations and computational settings to use a priori and design their studies or specify their conceptual hypotheses in light of these expectations — incorporating means-end analysis and analogical reasoning strategies — more quickly. It may be these insights that analysts in our lab study sought when they looked online for conceptual and statistical help.

Future work could observe how domain experts perform hypothesis formalization and characterize when and how analysts draw upon their own or collaborators' expertise to circumvent iterations or justify early scoping decisions. These insights may also shed light on how pre-registration expectations and practices could be made more effective. Given the level of detail required of some pre-registration policies, researchers likely engage in a version of the hypothesis formalization process we have identified prior to registering their studies. Knowing how pre-registration fits into the hypothesis formalization process could improve the design and adoption of pre-registration practices.

Future work could also explore how hypothesis formalization may differ in machine learning settings. In this paper, our focus was on how analysts answer domain questions and test hypotheses using statistical methods and their domain knowledge. Our findings may not generalize to settings or methods where domain knowledge is less important, such as deep learning and other machine learning-based approaches.

Finally, our findings suggest opportunities for future tools to bridge steps involved in hypothesis formalization and guide analysts towards reasonable model implementations. Our analysis of tools suggest possibilities for tools to connect model implementations to their mathematical representations through meta-libraries, provide higher-level abstractions for more directly expressing conceptual hypotheses, and support automated conceptual modeling. Future system development and user testing are necessary to validate these implications and more readily support analysts translate their conceptual hypotheses into statistical model implementations.

7 ACKNOWLEDGEMENTS

We thank Philip Garrison and Manaswi Saha for their perspectives and feedback on our content analysis approaches. We are grateful to Yang Liu, Alex Kale, and the other UW Interactive Data Lab members for their feedback on early paper drafts and insightful conversation about hypothesis formalization.

REFERENCES

[1] Sara Alspaugh, Nava Zokaei, Andrea Liu, Cindy Jin, and Marti A Hearst. 2018. Futzing and moseying: Interviews with professional data analysts on exploration practices. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 22–31.

[2] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2014. Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823* (2014).

[3] Douglas Bates, Martin Mächler, Ben Bolker, Steve Walker, Rune H.B. Christensen, Henrik Singmann, Bin Dai, Fabian Scheipl, Gabor Grothendieck, Peter Green, and John Fox. 2019. Package 'lme4'. *CRAN* (2019). "https://cran.r-project.org/web/packages/lme4/lme4.pdf"

[4] Leilani Battle and Jeffrey Heer. 2019. Characterizing Exploratory Visual Analysis: A Literature Review and Evaluation of Analytic Provenance in Tableau. *Computer Graphics Forum (Proc. EuroVis)* (2019). "http://idl.cs.washington.edu/papers/exploratory-visual-analysis"

[5] Jacques Bertin. 2011. *Graphics and graphic information processing*. Walter de Gruyter.

[6] Michael Betancourt. 2020. Towards a Principled Bayesian Workflow. https://betanalpha.github.io/assets/case_studies/principled_bayesian_workflow.html

[7] Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. 2019. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research* 20, 1 (2019), 973–978.

[8] Igor Bobriakov. 2017. Top 15 Python Libraries for Data Science in 2017. *ActiveWizards in Medium* (2017). "https://medium.com/activewizards-machine-learning-company/top-15-python-libraries-for-data-science-in-in-2017-ab61b4f9b4a7"

[9] Igor Bobriakov. 2018. Top 20 Python libraries for data science in 2018. *ActiveWizards in Medium* (2018). "https://medium.com/activewizards-machine-learning-company/top-20-python-libraries-for-data-science-in-2018-2ae7d1db8049"

[10] Leo Breiman, Adele Cutler, Andy Liaw, and Matthew Wiener. 2018. Package 'randomForest'. (2018). "https://cran.r-project.org/web/packages/randomForest/randomForest.pdf"

[11] Mollie E Brooks, Kasper Kristensen, Koen J van Benthem, Arni Magnusson, Casper W Berg, Anders Nielsen, Hans J Skaug, Martin Machler, and Benjamin M Bolker. 2017. glmmTMB balances speed and flexibility among packages for zero-inflated generalized linear mixed modeling. *The R journal* 9, 2 (2017), 378–400.

[12] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. 2013. API design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238* (2013).

[13] Paul-Christian Bürkner et al. 2017. brms: An R package for Bayesian multilevel models using Stan. *Journal of statistical software* 80, 1 (2017), 1–28.

[14] Paul-Christian Bürkner and Maintainer Paul-Christian Buerkner. 2016. Package 'brms'. (2016).

[15] Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. Stan : A Probabilistic Programming Language. *Journal of Statistical Software* 76 (01 2017). https://doi.org/10.18637/jss.v076.i01

[16] Robert Carver, Michelle Everson, John Gabrosek, Nicholas Horton, Robin Lock, Megan Mocko, Allan Rossman, Ginger Holmes Roswell, Paul Velleman, Jeffrey Witmer, et al. 2016. Guidelines for assessment and instruction in statistics education (GAISE) college report 2016. (2016).

[17] Yunshun Chen, Aaron TL Lun, Davis J McCarthy, Matthew E Ritchie, Belinda Phipson, Yifang Hu, Xiaobei Zhou, Mark D Robinson, and Gordon K Smyth. 2020. Empirical Analysis of Digital Gene Expression Data in R (v3.30.3). (2020). "https://bioconductor.org/packages/release/bioc/html/edgeR.html"

[18] Yunshun Chen, David McCarthy, Matthew Ritchie, Mark Robinson, and Gordon Smyth. 2020. edgeR: differential analysisof sequence read count data. (2020). "https://bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf"

[19] François Chollet et al. 2015. Keras. https://keras.io.

[20] Alexander Eiselmayer, Chatchavan Wacharamanotham, Michel Beaudouin-Lafon, and Wendy Mackay. 2019. Touchstone2: An Interactive Environment for Exploring Trade-offs in HCI Experiment Design. (2019).

[21] Janet Feigenspan, Christian Kästner, Jörg Liebig, Sven Apel, and Stefan Hanenberg. 2012. Measuring programming experience. In *2012 20th IEEE International Conference on Program Comprehension (ICPC)*. IEEE, 73–82.

[22] Jerome Friedman, Trevor Hastie, Rob Tibshirani, Balasubramanian Narasimhan, Kenneth Tay, Noah Simon, and Junyang Qian. 2020. Package 'glmnet'. (2020). **"https://cran.r-project.org/web/packages/glmnet/index.html"**

[23] Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. 2019. Visualization in Bayesian workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 182, 2 (2019), 389–402.

[24] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. 2013. *Bayesian data analysis*. CRC press.

[25] Andrew Gelman and Eric Loken. 2013. 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).

[26] Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. 2020. Bayesian workflow. *arXiv preprint arXiv:2011.01808* (2020).

[27] LLC. GraphPad Software. 2020. GraphPad Prism 8 User Guide. (2020). **"https://www.graphpad.com/guides/prism/8/user-guide/index.htm"**

[28] Garrett Grolemund. 2019. Quick list of useful R packages. *R Studio Support* (2019). **"https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages"**

[29] Garrett Grolemund and Hadley Wickham. 2014. A cognitive interpretation of data analysis. *International Statistical Review* 82, 2 (2014), 184–204.

[30] Jarrod Hadfield. 2020. Package 'MCMCglmm'. (2020). **"https://cran.r-project.org/web/packages/MCMCglmm/MCMCglmm.pdf"**

[31] Jarrod D Hadfield et al. 2010. MCMC methods for multi-response generalized linear mixed models: the MCMCglmm R package. *Journal of Statistical Software* 33, 2 (2010), 1–22.

[32] Trevor Hastie and Junyang Qian. 2014. Glmnet vignette. (2014). **https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html**

[33] Brian Hempel, Justin Lubin, and Ravi Chugh. 2019. Sketch-n-Sketch: Output-Directed Programming for SVG. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 281–292.

[34] John H Holland, Keith J Holyoak, Richard E Nisbett, and Paul R Thagard. 1989. *Induction: Processes of inference, learning, and discovery*. MIT press.

[35] Bradley Jones and John Sall. 2011. JMP statistical discovery software. *Wiley Interdisciplinary Reviews: Computational Statistics* 3, 3 (2011), 188–194.

[36] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–2020. SciPy: Open source scientific tools for Python. **http://www.scipy.org/**

[37] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–2020. SciPy: Open source scientific tools for Python. **https://docs.scipy.org/doc/scipy/reference/stats.html**

[38] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–2020. SciPy: Open source scientific tools for Python. **https://docs.scipy.org/doc/scipy/reference/optimize.html**

[39] Eunice Jun, Maureen Daum, Jared Roesch, Sarah E Chasins, Emery D Berger, Rene Just, and Katharina Reinecke. 2019. Tea: A High-level Language and Runtime System for Automating Statistical Analysis. In *Proceedings of the 32nd Annual Symposium on User Interface Software and Technology*. ACM.

[40] Alex Kale, Matthew Kay, and Jessica Hullman. 2019. Decision-making under uncertainty in research synthesis: Designing for the garden of forking paths. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[41] Sean Kandel, Andreas Paepcke, Joseph M Hellerstein, and Jeffrey Heer. 2012. Enterprise data analysis and visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2917–2926.

[42] Norbert L Kerr. 1998. HARKing: Hypothesizing after the results are known. *Personality and social psychology review* 2, 3 (1998), 196–217.

[43] David Klahr and Kevin Dunbar. 1988. Dual space search during scientific reasoning. *Cognitive science* 12, 1 (1988), 1–48.

[44] David Klahr and Herbert A Simon. 1999. Studies of scientific discovery: Complementary approaches and convergent findings. *Psychological Bulletin* 125, 5 (1999), 524.

[45] Gary Klein, Jennifer K Phillips, Erica L Rall, and Deborah A Peluso. 2007. A data–frame theory of sensemaking. In *Expertise out of context*. Psychology Press, 118–160.

[46] Joos Korstanje. 2019. "ANOVA's three types of estimating Sums of Squares: don't make the wrong choice!". *Towards Data Science, Medium* (2019). "https://towardsdatascience.com/anovas-three-types-of-estimating-sums-of-squares-don-t-make-the-wrong-choice-91107c77a27a"
[47] Max Kuhn, Davis Vaughan, and RStudio. 2020. parsnip: A Common API to Modeling and Analysis Functions. "https://parsnip.tidymodels.org/"
[48] Max Kuhn and Hadley Wickham. 2020. *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles.* "https://www.tidymodels.org"
[49] Michael D Lee, Amy H Criss, Berna Devezer, Christopher Donkin, Alexander Etz, Fábio P Leite, Dora Matzke, Jeffrey N Rouder, Jennifer S Trueblood, Corey N White, et al. 2019. Robust modeling in cognitive science. *Computational Brain & Behavior* 2, 3 (2019), 141–153.
[50] Jiali Liu, Nadia Boukhelifa, and James R Eagan. 2019. Understanding the Role of Alternatives in Data Analysis Practices. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 66–76.
[51] Yang Liu, Tim Althoff, and Jeffrey Heer. 2019. Paths Explored, Paths Omitted, Paths Obscured: Decision Points & Selective Reporting in End-to-End Data Analysis. *arXiv preprint arXiv:1910.13602* (2019).
[52] StataCorp LLC. 2020. Language syntax. https://www.stata.com/manuals13/u11.pdf
[53] StataCorp LLC. 2020. Stata 16 Documentation. https://www.stata.com/features/documentation/
[54] David J. Lunn, Andrew Thomas, Nicky Best, and David Spiegelhalter. 2000. WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing* 10, 4 (01 Oct 2000), 325–337. https://doi.org/10.1023/A:1008929526011
[55] Arni Magnusson, Hans Skaug, Anders Nielsen, Casper Berg, Kasper Kristensen, Martin Maechler, Koen van Bentham, Ben Bolker, Nafis Sadat, Daniel Lüdecke, Russ Lenth, Joseph O'Brien, and Mollie Brooks. 2020. Package 'glmmTMB'. (2020). "https://cran.r-project.org/web/packages/glmmTMB/index.html"
[56] Richard McElreath. 2020. *Statistical rethinking: A Bayesian course with examples in R and Stan*. CRC press.
[57] Allen Newell, Herbert Alexander Simon, et al. 1972. *Human problem solving*. Vol. 104. Prentice-Hall Englewood Cliffs, NJ.
[58] Chi T. Ngo, Aidan J. Horner, Nora S. Newcombe, and Ingrid R. Olson. 2019. Development of Holistic Episodic Recollection. *Psychological Science* 30, 12 (2019), 1696–1706.
[59] Donald A Norman. 1986. Cognitive engineering. *User centered system design* 31 (1986), 61.
[60] University of Amsterdam. 2020. JASP: A Fresh Way to do Statistics. "https://jasp-stats.org/"
[61] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.
[62] Josef Perktold, Skipper Seabold, Jonathan Taylor, and statsmodels developers. 2020. Statsmodels v0.10.2 Reference Guide. (2020). "https://www.statsmodels.org/stable"
[63] M Pfannkuch. 1997. Statistical thinking: One statistician's perspective. *Research papers on stochastics education* (1997), 171–178.
[64] Maxine Pfannkuch, Chris J Wild, et al. 2000. Statistical Thinking an Statistical Practice: Themes Gleaned from Professional Statisticians. *Statistical science* 15, 2 (2000), 132–152.
[65] José Pinheiro, Douglas Bates, Saikat DebRoy, Deepayan Sarkar, EISPACK authors, Siem Heisterkamp, Bert Van Willigen, and R-core. 2020. Package 'nlme'. (2020). "https://cran.r-project.org/web/packages/nlme/nlme.pdf"
[66] Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, Vol. 5. McLean, VA, USA, 2–4.
[67] Tanu N Prabhu. 2019. Top Python Libraries Used In Data Science. *Towards Data Science, Medium* (2019). "https://towardsdatascience.com/top-python-libraries-used-in-data-science-a58e90f1b4ba"
[68] Brian Ripley, Bill Venables, Douglas M. Bates, Kurt Hornik, Albrecht Gebhardt, , and David Firth. 2020. Package 'MASS'. (2020). "https://cran.r-project.org/web/packages/MASS/MASS.pdf"
[69] Daniel M Russell, Mark J Stefik, Peter Pirolli, and Stuart K Card. 1993. The cost structure of sensemaking. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. ACM, 269–276.
[70] John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. 2016. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science* 2 (2016), e55.
[71] SAS. 2020. JMP. "https://www.jmp.com/en_us/home.html"
[72] Christian D Schunn and David Klahr. 1995. A 4-space model of scientific discovery. In *Proceedings of the 17th annual conference of the cognitive science society*. 106–111.
[73] Christian D Schunn and David Klahr. 1996. When and how to go beyond a 2-space model of scientific discovery. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society: July 12-15, 1996, University of*

California, San Diego, Vol. 18. Psychology Press, 25.
[74] scikit-learn developers. 2020. Scikit-Learn v0.23.2 Documentation. (2020). "https://scikit-learn.org/stable/"
[75] Skipper Seabold and Josef Perktold. 2010. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, Vol. 57. Scipy, 61.
[76] IBM SPSS. [n.d.]. SPSS Software. https://www.ibm.com/analytics/spss-statistics-software
[77] Stata. [n.d.]. Stata Software. https://www.stata.com/
[78] Michael Suh. 2014. Higher Education, Gender & Work Dataset. "https://www.pewsocialtrends.org/category/datasets/?download=20041"
[79] R Core Team and contributors worldwide. 2020. Package 'stats' v4.1.0. *CRAN* (2020). "https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html"
[80] Inc. The MathWorks. 2020. Matlab. "https://www.mathworks.com/"
[81] Inc. The MathWorks. 2020. Statistics and Machine Learning Toolbox. (2020). "https://www.mathworks.com/help/stats/index.html"
[82] Lea Verou, Tarfah Alrashed, and David Karger. 2018. Extending a Reactive Expression Language with Data Update Actions for End-User Application Authoring. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 379–387.
[83] Chat Wacharamanotham, Krishna Subramanian, Sarah Theres Volkel, and Jan Borchers. 2015. Statsplorer: Guiding novices in statistical analysis. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2693–2702.
[84] Hadley Wickham et al. 2014. Tidy data. *Journal of Statistical Software* 59, 10 (2014), 1–23.
[85] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, et al. 2019. Welcome to the Tidyverse. *Journal of Open Source Software* 4, 43 (2019), 1686.
[86] Chris J Wild and Maxine Pfannkuch. 1999. Statistical thinking in empirical enquiry. *International statistical review* 67, 3 (1999), 223–248.
[87] Kanit Wongsuphasawat, Yang Liu, and Jeffrey Heer. 2019. Goals, Process, and Challenges of Exploratory Data Analysis: An Interview Study. *arXiv preprint arXiv:1911.00568* (2019).
[88] Bin Yu and Karl Kumbier. 2020. Veridical data science. *Proceedings of the National Academy of Sciences* 117, 8 (2020), 3920–3929.