Tisane: Authoring Statistical Models via Formal Reasoning from Conceptual and Data Relationships

Eunice Jun
emjun@cs.washington.edu
University of Washington
Seattle, Washington, USA

Audrey Seo
alseo@cs.washington.edu
University of Washington
Seattle, 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

ABSTRACT

Proper statistical modeling incorporates domain theory about how concepts relate and details of how data were measured. However, data analysts currently lack tool support for recording and reasoning about domain assumptions, data collection, and modeling choices in an integrated manner, leading to mistakes that can compromise scientific validity. For instance, generalized linear mixed-effects models (GLMMs) help answer complex research questions, but omitting random effects impairs the generalizability of results. To address this need, we present Tisane, a mixed-initiative system for authoring generalized linear models with and without mixed-effects. Tisane introduces a *study design specification language* for expressing and asking questions about relationships between variables. Tisane contributes an *interactive compilation* process that represents relationships in a graph, infers candidate statistical models, and asks follow-up questions to disambiguate user queries to construct a valid model. In case studies with three researchers, we find that Tisane helps them focus on their goals and assumptions while avoiding past mistakes.

KEYWORDS

statistical analysis; linear modeling; end-user programming; end-user elicitation; domain-specific language; transparent statistics; validity

ACM Reference Format:
Eunice Jun, Audrey Seo, Jeffrey Heer, and René Just. 2022. Tisane: Authoring Statistical Models via Formal Reasoning from Conceptual and Data Relationships. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3491102.3501888

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3501888

1 INTRODUCTION

Statistical models play a critical role in how people evaluate data and make decisions. Policy makers rely on models to track disease, inform health recommendations, and allocate resources. Scientists use models to develop, evaluate, and compare theories. Journalists report on new findings in science, which individuals use to make decisions that impact their nutrition, finances, and other aspects of their lives. Faulty statistical models can lead to spurious estimations of disease spread, findings that do not generalize or reproduce, and a misinformed public. The challenge in developing accurate statistical models lies not in a lack of access to mathematical tools, of which there are many (e.g., R [63], Python [52], SPSS [58], and SAS [24]), but in accurately applying them in conjunction with domain theory, data collection, and statistical knowledge [26, 38].

There is a mismatch between the interfaces existing statistical tools provide and the needs of analysts, especially those who have domain knowledge but lack deep statistical expertise (e.g., many researchers). Current tools separate reasoning about domain theory, study design, and statistical models, but analysts need to reason about all three *together* in order to author accurate models [26]. For example, consider a researcher developing statistical models of hospital expenditure to inform public policy. They collect data about individual hospitals within counties. Based on their domain knowledge, they know that counties have different demographics and that hospitals in these counties have different funding sources (private vs. public), all of which influence hospital spending. To model county-level and hospital-level attributes, the researcher may author a generalized linear mixed-effects model (GLMM) that accounts for clustering within counties. But which variables should they include? How do they account for this clustering? The three most common mistakes in modeling hierarchical data [9] lead to miscalibrated statistical power, "ecological fallacies" [49], and/or results that may not generalize, which impact not only the validity of research findings [3] but also enacted policies. How can the researcher avoid these issues?

To reduce threats to validity and improve analytical practices, **how might we derive (initial) statistical models from knowledge about concepts and data collection?** Inferring a statistical model raises two challenges: (1) How do we elicit the information necessary for inferring a statistical model? and (2) How do we infer a

statistical model, given this information? We present **Tisane, a system for integrating conceptual relationships, data collection details, and modeling choices when specifying generalized linear models (GLMs) and generalized linear mixed-effects models (GLMMs)**. GLMs and GLMMs are meaningful targets because they are commonly used (e.g., in psychology [9, 35], social science [29], and medicine [3, 5]) yet are easy to misspecify for statistical experts and non-experts alike [3, 9]. We designed Tisane to support researchers who are domain experts capable of supplying conceptual and data collection information but lack the statistical expertise or confidence to author GLM/GLMMs accurately.

Tisane provides a **study design specification language** for expressing relationships between variables. For example, the public health researcher can express that average county income is associated with hospital spending based on health economics theory or specify that hospitals exist within counties. Tisane compiles the explicitly stated relationships into an internal **graph representation** and then traverses the graph to infer candidate GLMs/GLMMs. In this process, Tisane engages analysts in **interactive compilation**. Analysts can query Tisane for a statistical model that explains a specific dependent variable from a set of independent variables. Based on the input query, Tisane asks analysts disambiguating questions to output a script for fitting a valid GLM/GLMM. Interactive compilation enables analysts to focus on their primary variables of interest as the system checks that analysts do not overlook relevant variables, such as potential confounders or data clustering that could compromise generalizability. Figure 1 provides an overview of this process.

To examine how Tisane affects real-world analyses, we conducted **case studies with three researchers**. The researchers described how Tisane focused them on their research goals, made them aware of domain assumptions, and helped them avoid past mistakes. Tisane even helped one researcher correct their model prior to submitting to the ACM Conference on Human Factors in Computing (CHI). These findings corroborate those from an earlier pilot study that informed our design process (see supplemental material).

We contribute (1) a study design specification language and graph representation for recording and reasoning about conceptual relationships between variables and data collection procedures (5), (2) an interactive compilation process that asks disambiguating questions and outputs code for fitting and visualizing a GLM/GLMM (6), and (3) three case studies with researchers that demonstrate the feasibility and benefit of prioritizing variable relationships to author linear models (7). We also provide an open-source Python implementation of Tisane.[1]

2 BACKGROUND AND RELATED WORK

We first provide brief background about data analysis practices, GLMs/GLMMs, and causal analysis. Then, we discuss how Tisane extends prior work on tools for conceptual reasoning, study design, and automated statistical analysis.

[1]Tisane is available for download on pip, a popular Python package manager. The source code is available at https://github.com/emjun/tisane.

2.1 Data Analysis Practices

Studies with analysts have found that data analysis is an iterative process that involves data collection; cleaning and wrangling; and statistical testing and modeling [21, 31, 32]. To formalize their hypotheses as statistical model programs, analysts engage in a dual-search process involving refinements to their conceptual understanding and iterations on model implementations, under constraints of data and statistical knowledge [26]. Analysts incorporate and refine their domain knowledge, study design, statistical models, and computational instantiations of statistical models while creating statistical model programs. Tisane facilitates one formalization cycle in this iterative process: deriving statistical models from conceptual knowledge and data measurement specifications.

2.2 Generalized Linear Models and Generalized Linear Mixed-effects Models

Tisane supports two classes of models that are widely applicable to diverse domains and data collection settings [3, 5, 35]: Generalized Linear Models (GLMs) and Generalized Linear Mixed-effects Models (GLMMs). Both GLMs and GLMMs consist of (i) a *model effects structure*, which can include main and interaction effects and (ii) *family* and *link* functions. The family function describes how the residuals of a model are distributed. The link function transforms the predicted values of the dependent variable. This allows modeling of linear and non-linear relationships between the dependent variable and the predictors. In contrast to transformations applied directly to the dependent variable, a link function does not affect the error distributions around the predicted values. The key difference between GLMs and GLMMs is that GLMMs contain random effects in their model effects structure. Random effects describe how individuals (e.g., a study participant) vary and are necessary in the presence of hierarchies, repeated measures, and non-nesting composition (5.2.2)[2].

Both GLMs and GLMMs assume that (i) the variables involved are linearly related, (ii) there are no extreme outliers, and (iii) the family and link functions are correctly specified. In addition, GLMs also assume that (iv) the observations are independent. Tisane's interactive compilation process guides users through specifying model effects structures, family and link functions to satisfy assumption (iii), and random effects only when necessary to pick between GLMs and GLMMs and satisfy assumption (iv).

2.3 Causal Analysis

There are multiple frameworks for reasoning about causality [44, 50]. One widespread approach is to use directed acyclic graphs (DAGs) to encode conditional dependencies between variables [20, 45, 56, 57]. If analysts can specify a formal causal graph, Pearl's "backdoor path criterion" [44, 46] explains the set of variables that control for confounding. However, in practice, specifying proper causal DAGs is challenging and error-prone for domain experts who are not also experts in causal analysis [60] due to uncertainty of empirical findings [61] and lack of guidance on which variables

[2]Traditionally, the term "mixed effects" refers to the simultaneous presence of "fixed" and "random" effects in a single model. We try to avoid these terms as there are many contradictory usages and definitions [18]. When we do use these terms, we use the definitions from Kreft and De Leeuw [29].

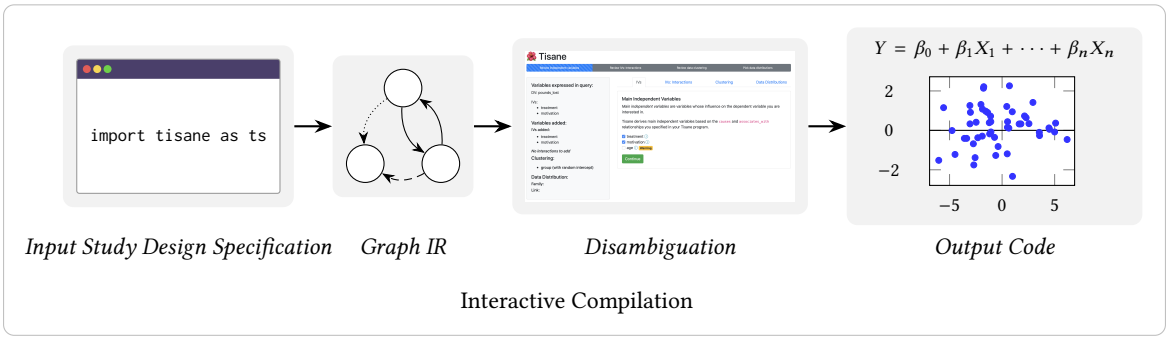


Figure 1: Overview of the Tisane system. Analysts specify a set of variable relationships (*Input Study Design Specification*). Tisane represents these in an internal graph (*Graph IR*). To infer a statistical model, Tisane engages analysts in an interactive compilation process that elicits additional input from analysts in a disambiguation process (*Disambiguation*) and outputs a script for fitting a valid GLM and visualizing its residuals (*Output Code*).

and relationships to include [67]. Accordingly, Tisane does not expect analysts to specify a formal causal graph. Instead, analysts can express causal relationships as well as "looser" association (not causal) relationships between variables in the study design specification language.

Prior work in the causal reasoning literature shows how linear models can be derived from causal graphs to make statistical inferences and test the motivating causal graph [56, 57]. Recently, VanderWeele proposed the "modified disjunctive cause criterion" [66] as a new heuristic for researchers without a clearly accepted formal causal model to identify confounders to include in a linear model, for example. The criterion identifies confounders in a graph based on expressed causal relationships. Tisane applies the modified disjunctive cause criterion when suggesting variables to include in a GLM or GLMM. Tisane does not automatically include variables to the statistical models because substantive domain knowledge is necessary to resolve issues of temporal dependence between variables, among other considerations [66]. To guide analysts through the suggestions, Tisane provides analysts with explanations to aid their decision making during disambiguation.

Finally, GLMs are not formal causal analyses. Tisane does not calculate average causal effect or other causal estimands. Rather, Tisane only utilizes insights about the connection between causal DAGs and linear models to guide analysts towards including potentially relevant confounders in their GLMs grounded in domain knowledge.

2.4 Tools for Conceptual Reasoning and Study Design

Tools such as Daggity [64] support authoring, editing, and formally analyzing causal graphs through code and a visual editor. Daggity requires users to specify a formal causal graph, which may not always be possible [60, 61, 67]. Although a knowledgeable analyst could use Daggity to identify a set of variables that control for confounding to include in a linear model, Daggity does not provide this support directly. In contrast, Tisane aims to (i) help analysts may not be able to formally specify causal graphs and (ii) scaffold the derivation of GLMs and GLMMs from causal graphs.

Several domain-specific languages [1, 55] and tools specialize in experiment design [4, 15, 62]. A primary focus is to provide researchers low-level control over trial-level and randomization details. For example, JsPsych [13] gives researchers fine-grained control over the design and presentation of stimuli for online experiments. At a mid-level of abstraction, Touchstone [36] is a tool for designing and launching online experiments. It also refers users to R and JMP for data analysis but does not help users author an appropriate statistical model. Touchstone2 [15] helps researchers design experiments based on statistical power. At a high-level of abstraction, edibble [62] helps researchers plan their data collection schema. Edibble aims to provide a "grammar of study design" that focuses users on their experimental manipulations in relation to specific units (e.g., participants, students, schools), the frequency and distribution of conditions (e.g., within-subjects vs. between-subjects), and measures to collect (e.g., age, grade, location) in order to output a table to fill in during data collection. While Tisane's study design specification language uses an abstraction level comparable to edibble, Tisane is focused on using the expressed data measurement relationships to infer a statistical model. Additionally, Tisane's SDSL provides conceptual relationships that are out of the scope of edibble but important for specifying conceptually valid statistical models.

2.5 Tools for Automated Statistical Analysis

Researchers have introduced tools that automate statistical analyses. Given a dataset, the Automatic Statistician [34] generates a report listing all "interesting" relationships (e.g., correlations, statistical models, etc.). Although apparently complete, the Automatic Statistician may overlook analyses that are conceptually interesting and difficult, if not impossible, to deduce from data alone. In contrast, Tisane prioritizes analyst-specified conceptual and data measurement relationships and uses them to bootstrap the modeling process. As a result, Tisane aims to ensure that statistical analyses are not only *technically* correct but also *conceptually* correct.

AutoML tools automate machine learning for non-experts. Tools such as Auto-WEKA [65], auto-sklearn [16], and H2O AutoML [30] aim to make statistical methods more widely usable. Tisane differs

from AutoML efforts in its focus on analysts who prioritize explanation, not just prediction, such as researchers developing scientific theories. As a result, Tisane provides support for specifying GLMMs, which some prominent AutoML tools, such as auto-sklearn [16], omit. Tisane ensures that inferred statistical models respect expressed conceptual relationships. Thus, Tisane programs can serve a secondary purpose of recording and communicating conceptual and data measurement assumptions. In addition, Tisane explains its suggestions to users and guides them in answering disambiguation questions whereas AutoML tools do not by default. Tisane's explanations are grounded in the variable relationships analysts specify. Although H2O AutoML offers a model explainability module [14], these "explanations" take the form of plots without conceptual exposition.

Tea [27] elicits end-user expertise through explicit hypotheses and study designs to automatically infer a set of valid Null Hypothesis Significance Tests. Tisane differs from Tea in three key ways. First, Tisane enables analysts to express more complex study designs, such as nested hierarchies that necessitate mixed effects modeling, which are not available in Tea. Second, because statistical modeling requires more conceptual expertise and oversight, Tisane is a mixed-initiative system while Tea is not. Third, Tisane outputs a single statistical model, whereas Tea outputs a set of statistical tests.

Recent work in the database community helps researchers answer causal questions about multilevel, or hierarchical, data [28, 51]. CaRL [51] provides a domain-specific language to express causal relationships between variables and a GUI to show researchers results. Like CaRL, Tisane leverages the insight that researchers have domain knowledge that a system can use to infer statistical methods. Whereas CaRL is focused on answering specific queries about average causal effect, Tisane supports authoring GLMs and GLMMs, which can address a range of non-causal questions.

In summary, whereas prior systems have supported reasoning about either concepts, study designs, or statistical models, Tisane integrates all three. Existing theories of data analysis [26, 38] illustrate how all three concerns are necessary and interconnected. Thus, a tool for integrating these concerns seems a promising way to reduce errors and the cognitive burden involved in model specification.

3 USAGE SCENARIO

To illustrate how a researcher might use Tisane, we compare two hypothetical researchers analyzing the same dataset. This scenario (simplified from [9]) illustrates workflow differences between Tisane and current tools (statsmodels in Python and lme4 in R). Michael and Bridget are health experts studying the effects of a new exercise regimen they have developed on weight loss. Their research question is "How much does the exercise regimen affect weight loss?" They recruited 386 adults to be part of 40 exercise groups focused on diet and weight management. The researchers randomly assigned 16 groups a control regimen and the other 24 groups an experimental regimen. The researchers measured the adults' motivation scores for weight loss at the beginning of the experiment and their total weight loss at the end of the experiment.

Michael uses `statsmodels` [47] to analyze the data[3]. Bridget uses Tisane. While both are experienced researchers, familiar with their shared field of study, Michael and Bridget are not statistics experts. They have both used GLMs in the past but neither has heard of GLMMs.

3.1 Workflow in Python using `statsmodels`

Michael takes a first attempt at creating a model. He loads the data and casts `regimen_condition` and `group` as categorical variables. The first model (Listing 1) that Michael tries has the dependent variable `pounds_lost` and the independent variables `regimen` (control vs. treatment) and `motivation`.

```
import statsmodels.api as sm
import statsmodels.formula.api as sm
import pandas as pd

data = pd.read_csv("data.csv")
data['regimen_condition'] = data['regimen_condition'].
    astype('str').astype("category")
data['group'] = data['group'].astype('str').astype("
    category")

m1 = sm.glm("pounds_lost ~ regimen_condition + motivation
    ", data=data)
```

Listing 1: Michael's first model attempt, from the usage scenario (Section 3). Michael specifies `pounds_lost` as his dependent variable, and `regimen` and `motivation` as his independent variables.

Although this model includes the primary variable of research interest (`regimen`) and a likely confounder (`motivation`), Michael realizes that this model overlooks the fact that the adults exercised in groups. The groups likely had group support and accountability, among other benefits of group cohesion that are difficult to measure and not included in the data.

In his second model (Listing 2), Michael adds `group` as an additional independent variable. Considering that this model accounts for variables pertaining to individual adults (i.e., `pounds_lost`, `motivation`) and to groups (i.e., `regimen_condition`), Michael supposes this model is good enough and accepts it as his final model.

```
m2 = sm.glm("pounds_lost ~ regimen_condition + motivation
     + group", data=data)
```

Listing 2: Michael's second model attempt. Building on his first model (Listing 1), Michael adds an additional independent variable, `group`.

3.2 Workflow with Tisane

Bridget starts by listing the variables of interest. Lines 3-8 in Listing 3 show how variables are declared with a data type, the name of the column in the data that corresponds to the variable. The observational units are `adult` and `group`. The measures `motivation` and `pounds_lost` pertain to individual adults while exercise regimen was administered to exercise groups. The Tisane program makes these details explicit. Because Bridget has data, she does not need to specify the cardinality of variables.

```
1 import tisane as ts
2
3 # Variable declarations
4 adult = ts.Unit("member")
5 motivation = adult.numeric("motivation")
6 pounds_lost = adult.numeric("pounds_lost")
7 group = ts.Unit("group")
8 regimen = group.nominal("regimen_condition") # control vs
    . treatment
```

Listing 3: The first snippet of the example Tisane program,

[3] The workflow in R is almost identical to the workflow in Python using `statsmodels`.

statistical model in line 15.

```
# Query Tisane for a statistical model
design = ts.Design(dv=pounds_lost, ivs=[regimen_condition
    , motivation]).assign_data("data.csv")
ts.infer_model(design=design)
```

Listing 5: The final snippet of the example Tisane program, continuing from Listing 4.

Bridget queries Tisane for a statistical model by specifying her study design's most important variables and assigning her study design's most important variables and assigning data. Based on the complete program, consisting of this listing as well as Listing 3 and Listing 4, Bridget engages in a disambiguation process (see Figure 2) to generate a final output statistical modeling script.

Tisane launches a GUI in the browser after executing the program.[4] In the GUI, Tisane asks Bridget to look over her choice of variables (Figure 2). As seen in panel B, there are no additional variables to consider. Bridget continues to the next tab, interaction effects (panel D). Tisane explains that interaction effects do not make sense for her model given that she did not express any moderating relationships. In this way, Tisane prevents her from adding arbitrary interaction effects without conceptual foundation. In the next tab (E), Bridget sees that Tisane has automatically included exercise group as a random intercept. (Tisane does not include a random slope for group because there is only one observation per adult in a group, see subsubsection 6.2.3.) Bridget has not heard of a random intercept before, but she reads the explanation, which explains that accounting for exercise groups is necessary since adults were in groups and groups received the regimens. In the last tab, Bridget answers questions about the dependent variable pounds_lost to identify family and link functions. She specifies that the dependent variable is continuous. She chooses the Gaussian family with the default link function. Finally, Bridget clicks on the button to generate code. Tisane's output script contains code to fit the statistical model and plot model residuals to inspect fit. Tisane helped Bridget author the following GLMM:

```
model = Lmer(
    formula="pounds_lost ~ regimen_condition +
    motivation + (1|group)",
    family="gaussian",
    data=df,
)
```

Listing 6: Bridget's output statistical model from using Tisane. Tisane suggests a GLMM with group as a random intercept. The output script contains code for fitting this model and inspecting it using a residuals plot.

3.3 Key differences in workflows and statistical results

Even with experience modeling in Python, Michael makes two common mistakes in authoring linear models [9, 29]: disaggregating

Figure 2: Example Tisane GUI for disambiguation from usage scenario. Tisane asks analysts disambiguating questions about variables that are conceptually relevant and that analysts may have overlooked in their query. (A) The left hand panel gives an overview of the model the analyst is constructing. (B) Based on the variable relationships analysts specify (Listing 4), Tisane infers candidate main effects that may be potential confounders. Tisane asks analysts if they would like to include these variables, explaining in a tooltip (C) why the variable may be important to include. (D) Tisane only suggests interaction effects if analysts specify moderating relationships in their specification. This way, Tisane ensures that model structures are conceptually justifiable. (E) From the data measurement relationships analysts provide (line 15 in Listing 4), Tisane automatically infers and includes random effects to increase generalizability and external validity of statistical findings. (F) Tisane assists analysts in choosing an initial family and link function by asking them a series of questions about their dependent (e.g., Is the variable continuous or about count data?). To help analysts answer these questions and verify their assumptions about the data, Tisane shows a histogram of the dependent variable.

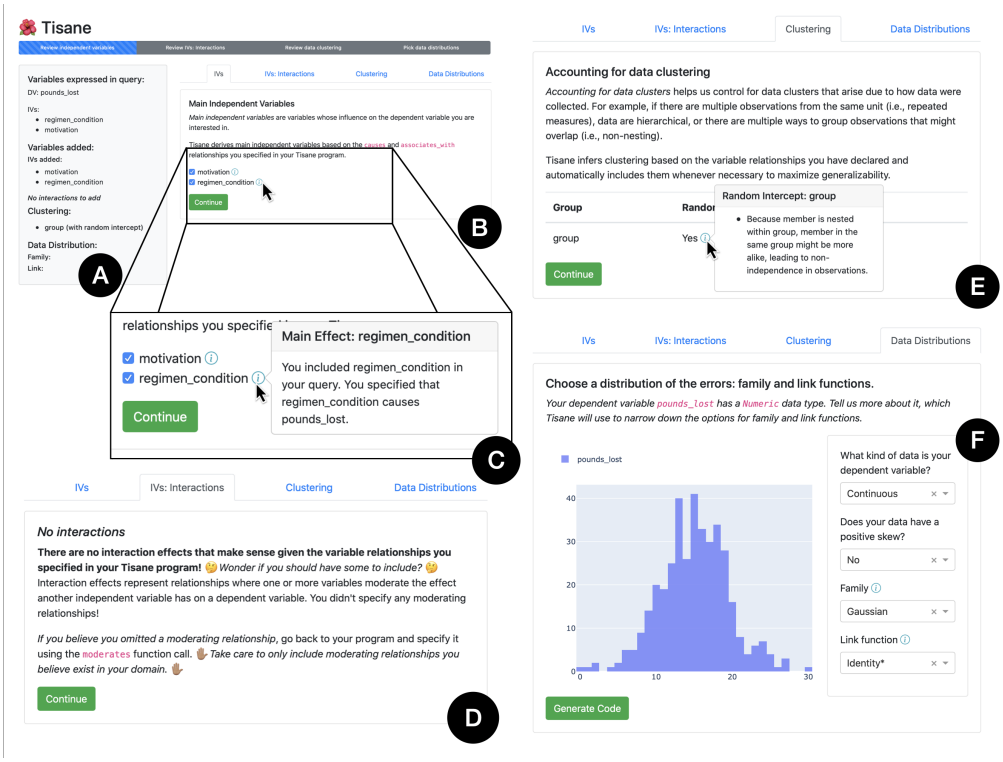


Figure 2: Example Tisane GUI for disambiguation from usage scenario. Tisane asks analysts disambiguating questions about variables that are conceptually relevant and that analysts may have overlooked in their query. (A) The left hand panel gives an overview of the model the analyst is constructing. (B) Based on the variable relationships analysts specify (Listing 4), Tisane infers candidate main effects that may be potential confounders. Tisane asks analysts if they would like to include these variables, explaining in a tooltip (C) why the variable may be important to include. (D) Tisane only suggests interaction effects if analysts specify moderating relationships in their specification. This way, Tisane ensures that model structures are conceptually justifiable. (E) From the data measurement relationships analysts provide (line 15 in Listing 4), Tisane automatically infers and includes random effects to increase generalizability and external validity of statistical findings. (F) Tisane assists analysts in choosing an initial family and link function by asking them a series of questions about their dependent (e.g., Is the variable continuous or about count data?). To help analysts answer these questions and verify their assumptions about the data, Tisane shows a histogram of the dependent variable.

the appropriate model, a GLMM, leads Michael to answer the research question differently than Bridget, artificially inflates statistical power [9], and compromises the generalizability of his findings [3].[5]

	$\beta_{\texttt{regimen\_condition}}$	p	$\beta_{\texttt{motivation}}$	p	$\beta_{\texttt{group}}$	p
Michael	1.628	.046	3.119	.000	Var.	Var.
Bridget	1.659	.005	3.193	.000	N/A	N/A

Table 1: The coefficients for each of the independent variables in Michael's and Bridget's models. "Var." stands for "Various," since there were multiple coefficients generated. The complete output tables for Michael's and Bridget's models are included in supplemental material.

4 DESIGN GOALS

We articulated four design goals based on prior research and our formative work. The supplemental material details our design process.

DG1 - Prioritize conceptual knowledge. Current tools require analysts to transition back and forth between their conceptual concerns and their statistical model specifications using math and/or code [26]. Analysts' conceptual knowledge remains implicit and hidden [38]. As a result, analysts may resort to familiar but suboptimal statistical methods [26] or accidentally overlook details that lead to conceptually inaccurate statistical models. One solution is to provide tools at a higher level of abstraction that allow analysts to express their conceptual knowledge directly. However, a higher level of abstraction alone is not enough. Tools must then leverage the expressed conceptual knowledge to guide analysis authoring.

[5]Another common mistake, not shown here, is to aggregate observations and use group means of the independent variables in the model, artificially deflating statistical power ("ecological fallacy" [49]). Kreft and De Leeuw [29] share an example where disaggregating vs. aggregating data lead to different signs for a fitted parameter. Unfortunately, we could not access the data to illustrate this here.

Tisane provides a high-level study design specification language that captures the motivation behind a study (Section 5). Tisane represents the specification in an internal graph representation to derive only conceptually accurate statistical model candidates. To arrive at an output statistical model, Tisane asks analysts disambiguating questions and provides them with suggestions and explanations based on their expressed variable relationships (Section 6). Importantly, Tisane does not fit or show modeling results during disambiguation to discourage statistical fishing. Although Tisane does not prevent researchers from re-starting and iterating on their Tisane program to attain specific statistical model findings, Tisane programs act as documentation for conceptual relationships that others could audit.

DG2 - Prioritize the validity of models. At present, the burden of valid statistics lies entirely on analysts. Tisane divides some of this burden by (i) ensuring correct application of methods (i.e., GLM vs. GLMM) and (ii) inferring models that increase the generalizability of results for GLMMs [2, 3]. Tisane helps analysts author GLMs and GLMMs that satisfy two assumptions: (i) observational dependencies and (ii) correct family and link functions. First, Tisane infers and constructs maximal random effects that account for dependencies due to repeated measures, hierarchical data, and non-nesting compositions. Maximal effects structures account for within-sample variability and thereby mitigate threats to external validity due to sampling biases from the choice of observational units and settings [54]. Second, Tisane narrows the set of viable family and link functions to match the dependent variable's data type (e.g., numeric). Tisane's GUI asks follow-up questions to determine the *semantic* type of variables (e.g., counts), further narrowing analysts' family and link function choices. The output script also plots model residuals against fitted values and provides tips (as comments) for interpreting the plot. The family and link functions Tisane suggests are intended to bootstrap an initial statistical model that analysts can examine and, if necessary, revise. This is how Tisane helps analysts avoid four common threats to statistical conclusion and external validity [11]: (i) violation of statistical method assumptions, (ii) fishing for statistical results, (iii) not accounting for the influence of specific units, and (iv) overlooking the influence of data collection procedures on outcomes.

DG3 - Give analysts guidance and control. Analysts may have insight into their research questions and domain that a system cannot capture. At the same time, analysts, especially those with less statistical experience, may lack the knowledge to select among many possible statistical models, which may inadvertently encourage cherry-picking based on observed results. Thus, Tisane adopts an interaction model that asks analysts specific questions to resolve modeling ambiguity rather than show multiple statistical models at the same time. Tisane also does not automatically select a a "best" model (e.g., highest R^2, easiest to interpret) but rather gives analysts suggestions and explanations to help them come to a statistical model that is valid and appropriate for their goals.

DG4 - Facilitate statistical planning without data. Experimental design best practices, such as pre-registration, encourage researchers to plan their statistical analyses prior to data collection. Tisane supports these best practices by not requiring that analysts provide data. If analysts do not have data, analysts must specify the cardinality at variable declaration. Without data, Tisane cannot validate variable declarations, but in this case, Tisane still guides analysts through the same interactive compilation process. The output Tisane script will include an empty file path and a comment directing analysts to specify the path to their data prior to execution. Analysts could attach this output Tisane script to their pre-registrations. After analysts collect data, they can re-run their previously specified Tisane program to validate and inspect their data. If Tisane does not issue any validation errors, analysts can proceed to execute their script.

5 STUDY DESIGN SPECIFICATION LANGUAGE AND GRAPH REPRESENTATION

Tisane provides a *study design specification language* (*SDSL*) for expressing relationships between variables. There are two key challenges in designing a specification from which to infer statistical models: (1) determining the set of relationships that are essential for statistical modeling and (2) determining the level of granularity to express relationships.

In Tisane's SDSL, analysts can express conceptual and data measurement relationships between variables. Both are necessary to specify the domain knowledge and study designs from which Tisane infers statistical models.

5.1 Variables

There are three types of data variables in Tisane's SDSL: (i) units, (ii) measures, and (iii) study environment settings. The **Unit** type represents entities that are observed and/or receive experimental treatments. In the experimental design literature, these entities are referred to as "observational units" and "experimental units," respectively. Entities can be both observational and experimental units simultaneously, so the SDSL does not provide more granular unit sub-types. The **Measure** type represents attributes of units and must be constructed through their units, e.g., `age = adult.numeric('age')`. Measures are proxies (e.g., minutes ran on a treadmill) of underlying constructs (e.g., endurance). Measures can have one of the following data types: numeric, nominal, or ordinal. Numeric measures have values that lie on an interval or ratio scale (e.g., age, minutes ran on a treadmill). Nominal measures are categorical variables without an ordering (e.g., race). Ordinal measures are ordered categorical variables (e.g., grade level in school). We included these data types because they are commonly taught and used in data analysis. The **SetUp** type represents study environment settings that are neither units nor measures. For example, time is often an environmental variable that differentiates repeated measures but is neither a unit nor a measure of a specific unit.

5.2 Relationships between Variables

In Tisane's SDSL, variables have relationships that fall into two broad categories: (1) *conceptual relationships* that describe how variables relate theoretically and (2) *data measurement relationships* that describe how the data was, or will be, collected. Below, we define each of the relationships in Tisane' SDSL and describe how Tisane internally represents these relationships as a graph (as illustrated in 3). 4 shows the graph representation constructed from the usage scenario.

Tisane's graph IR is a directed multigraph. Nodes represent variables, and directed edges represent relationships between variables. Tisane internally uses a graph intermediate representation (IR) because graphs are widely used for both conceptual modeling and statistical analysis, two sets of considerations that Tisane unifies.

Tisane's graph IR differs from two types of graphs used in data analysis: causal DAGs and path analysis diagrams. Unlike causal DAGs, Tisane's graph IR allows for non-causal relationships, moderating relationships (i.e., interaction effects), and data measurement relationships that are necessary for inferring random effects. Unlike path analysis diagrams that allow edges to point to other edges to represent interaction effects, Tisane represents interactions as separate nodes and only allows nodes as endpoints for edges. These design decisions simplify our statistical model inference algorithms and their implementation.

5.2.1 Conceptual relationships. Tisane's SDSL supports three conceptual relationships: causes, associates with, and moderates. Analysts can express that a variable **causes** or is **associated with** (but not directly causally related to) another variable. Variables associated with the dependent variable, for example, may help explain the dependent variable even if the causal mechanism is unknown. If analysts are aware of or suspect a causal relationship, they should use `causes`.

We chose to support both causal and associative relationships because formal causal DAGs are difficult for domain experts to specify [60, 61, 67], prior work has observed that researchers already use informal graphs that contain associative relationships when reasoning about their hypotheses and analyses [26], and GLMs/GLMMs can represent non-causal relationships. Finally, analysts can also express interactions where one (or more) variable (the *moderating variables*) **moderates** the effect of a *moderated variable* on another variable (the *target variable*).

Mediation relationships (where one variable influences another through a middle variable) are another common conceptual relationship. Tisane does not provide a separate language construct for mediation because mediations are expressible using two or more causal relationships. Furthermore, mediation analyses require specific analyses, such as structural equation modeling [23], that are out of Tisane's scope.

In the graph IR, a `causes` relationship introduces a causal edge from one node, the cause, to another node, the effect (3(a)). Because a variable cannot be both the cause and effect of the same variable, any pair of nodes can only have one causal edge between them. Furthermore, from a formal causal analysis perspective, associations may indicate the presence of a hidden, unobserved variable that mediates the causal effect of a variable on another or that influences two or more variables simultaneously. Thus, rather than inferring or requiring analysts to specify hidden variables, which may be unknown and/or unmeasurable, the `associates_with` relationship introduces two directed edges in opposing directions, representing the bidirectionality of association (3(b)). A `moderates` relationship creates a new node that is eventually transformed into an interaction term in the model, introduces associative edges between the new interaction node and the target (variable) node, creates associative edges between the moderated variable's node and the target node, and adds associative edges between the moderating variables' nodes and the target node if there is not a causal or associative edge already (3(c)). Furthermore, each interaction node inherits the attribution edges from the nodes of the moderating variables that comprise it. This means that every interaction node is also the attribute of at least one unit.[6]

5.2.2 Data measurement relationships. Study designs may have clusters of observations that need to be modeled explicitly for external validity. For example, in a within-subjects experiment, participants provide multiple observations for different conditions. An individual's observations may cluster together due to a hidden latent variable. Such clustering may be imperceptible during exploratory data visualization of a sample but can threaten external validity. GLMMs can mitigate three common sources of clustering that arise during data collection [8, 19, 29]:

- **Hierarchies** arise when one observational/experimental unit (e.g., adult) nests within another observational/experimental unit (e.g., group). This means that each instance of the nested unit belongs to one and only one nesting unit (many-to-one).
- **Repeated measures** introduce clustering of observations from the same unit instance (e.g., participant).
- **Non-nesting composition** arises when overlapping attributes (e.g., stimuli, condition) describe the same observational/experimental unit (e.g., participant) [19].

The above sources of clustering pose three problems for analysts. First, analysts must have significant statistical expertise to identify when data observations cluster. Second, they must know how to mitigate these clusters in their models. Third, with this knowledge, analysts must figure out how to express these types of clustering in their analytical tools. Even if analysts are not able to identify clustered observations, they are knowledgeable about how data were collected.

Thus, Tisane addresses the three problems by (i) eliciting data measurement relationships from analysts to infer clusters and (ii) formulating the maximal random effects structure, optimizing for external validity (6). Below, we describe language features for expressing data measurement relationships.

Nesting relationships: Hierarchies. **Hierarchies** arise when a unit (e.g., an `adult`) is nested within another unit (e.g., an exercise `group`). Researchers may collect data with hierarchies to study individual and group dynamics together or as a side effect of recruitment strategies. To express such designs, Tisane provides the `nests_within` construct. Conceptually, nesting is strictly between observational/experimental units, so Tisane type checks that the variables that nest are both Units. In the graph IR, a nesting relationship is encoded as an edge between two unit nodes (3(d)). There is one edge from the nested unit (e.g., `adult`) to the nesting unit (e.g., `group`) [7].

Frequency of measures: Repeated measures, Non-nesting composition. When a measure is declared through a unit, Tisane adds an

[6] In statistical terms, this means that within-level interactions have one unit while cross-level interactions may have two or more units.
[7] The GitHub repo contains a gallery of examples that include nesting relationships.

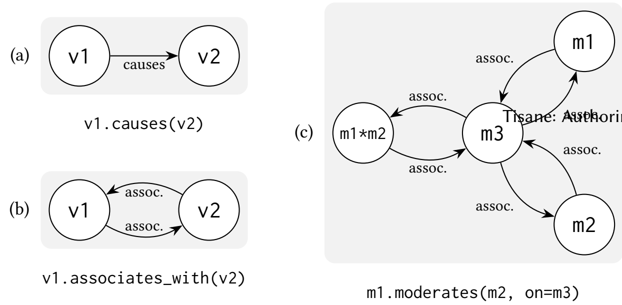


Figure 3: Code snippets of conceptual and data measurement relationships written in Tisane's specification language and their representation in Tisane's graph IR. Variables are named with u for units, m for measures, and v for variables that can be either units or measures. All edges depicted are those that are added due to the relationship. In the moderates example, we assume that m1 and m2 both belong to the same unit, and for simplicity, the attribution edge (labeled as "has") from m1 and m2's unit is not shown. For more complex examples, see the supplemental materials.

attribution edge ("has') from a unit node to a measure node (3(e)). A unit's measure can be taken one or more times in a study. The frequency of measurement is useful for detecting repeated measures and non-nesting composition. In **repeated measures** study designs, each unit provides multiple values of a measure, which are distinguished by another variable, usually time. **Non-nesting** [19] composition arises when measures describing the same unit overlap. For example, HCI researchers studying input devices might design them to utilize different senses (e.g., touch, sight, sound). Participants in the study may be exposed to multiple different devices, which act as experimental conditions or senses. The conditions are intrinsically tied to the devices, and participants can be described as having both conditions and devices, which overlap with one another. Such study designs introduce dependencies between observations [7] and hence violate the assumption of independence that GLMs make.

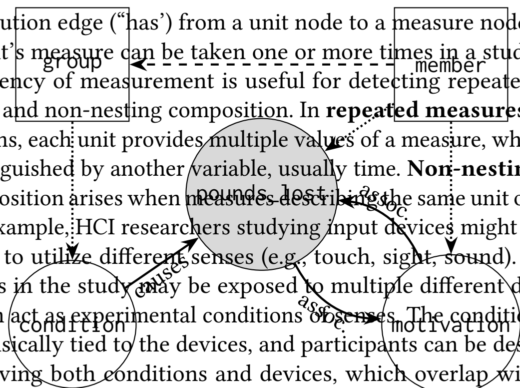


Figure 4: The graph representation of the variables and relationships from the usage scenario. causes edges are labeled with "causes". associates_with edges are labeled with "assoc." Dashed edges indicate nests_within relationships, and dotted edges indicate has relationships.

When analysts declare Measures, they specify the frequency of the observation through the number_of_instances parameter. This parameter accepts an integer, variable, a Tisane Exactly operator, or a Tisane AtMost operator. By default, the parameter is set to one. The Exactly operator represents the exact number of times a unit has a measure. The AtMost operator represents the maximum number of times a unit has a measure. Both operators are useful for specifying that a measure's frequency depends on another variable, which is expressible through the per function. For example, participants may use two devices *per* condition assigned: `device = subject.nominal('Input device', number_of_instances=ts.Exactly(2).per(condition))`. The per function uses the Tisane variable's cardinality by default but can instead use a data variable's number_of_instances by specifying use_cardinality=False as a parameter to per. Moreover, specifying a measure's number_of_instances to be an integer is syntactic sugar for using the Exactly operator. Specifying a variable is syntactic sugar for expressing ts.Exactly(1).per(variable).

To determine the presence of repeated measures or non-nesting composition, Tisane computes the number_of_instances of measures and their relationship to other measures. Measures that are declared with number_of_instances equal to one are considered to vary between-unit. Measures that are declared with number_of_instances greater than one, or a variable with cardinality greater than one are considered to vary within-unit as repeated measures. If there are instances of a measure per another "measure sharing the same unit, the measures are non-nesting.

6 STATISTICAL MODEL INFERENCE: INTERACTIVELY QUERYING THE GRAPH IR

After specifying variable relationships, analysts can query Tisane for a statistical model. Queries are constructed by specifying a study design with a dependent variable (the value to be predicted) and a set of independent variables (predictors). Tisane processes the query and generates a statistical model in four phases: (1) preliminary conceptual checks that validate the study design, (2) inference of possible effects structures and family and link functions, (3) input elicitation to disambiguate possible models, and (4) generation of a record of decisions during disambiguation and the final executable script. Given that the interactive process begins with an input program using Tisane and outputs a script for fitting a GLM or GLMM, we call this process *interactive compilation.*

6.1 Preliminary checks

At the beginning of processing a query, Tisane checks that every input study design is well-formed. This involves two conceptual correctness checks. First, every independent variable (IV) in the study design must either cause or be associated with the dependent variable (DV) directly or transitively. Second, the DV must not cause any of the IVs, since it would be conceptually invalid to explain a cause from any of its effects. If any of the above checks fail, Tisane issues a warning and halts execution. By using these two checks, the Tisane compiler avoids technically correct statistical models that have little to no conceptual basis (*G1 - Conceptual knowledge*). If the checks pass, Tisane moves on to the next phase.

6.2 Candidate statistical model generation

A GLM/GLMM is comprised of a model effects structure, family function, and link function. The model effects structure may consist of main, interaction, and random effects. Tisane utilizes variables' conceptual relationships to infer candidate main and interaction effects and data measurement relationships to infer random effects. Tisane infers family and link functions based on the data type of the DV in the query. The candidate statistical models that Tisane generates based on the graph and query seed an interactive disambiguation process.

The purpose of identifying candidate main effects beyond the ones analysts may have specified is to provoke consideration of erroneously omitted variables that are conceptually relevant and pre-empt potential confounding and multicollinearity issues that may arise.

6.2.1 Deriving Candidate Main Effects. To infer a statistical model, analysts specify a single dependent variable and a set of one or more IVs. After passing the checks described in 6.1, the query's independent variables are considered candidates. In addition, Tisane derives three additional sets of candidate main effects intended to control for confounding variables in the output statistical model.[8] The first two sets below are from the "modified disjunctive cause criterion" [66]:

[8]Tisane currently treats each input IV as a separate "exposure" variable for which to identify confounders. Tisane then combines all confounders into one statistical model.

a final executable script, and a record of decisions during disambiguation. Given that the interactive process begins with an input program using Tisane and outputs a script for fitting a GLM or GLMM, we call this process *interactive compilation*.

6.1 Preliminary checks

At the beginning of processing a query, Tisane checks that every input study design is well-formed. This involves two conceptual correctness checks. First, every independent variable (IV) in the study design must either cause or be associated with the dependent variable (DV) directly or transitively. Second, the DV must not cause any of the IVs, since it would be conceptually invalid to explain a cause from any of its effects. If any of the above checks fail, Tisane issues a warning and halts execution. By using these two checks, the Tisane compiler avoids technically correct statistical models that have little to no conceptual grounding (*DG1 - Conceptual knowledge*). If the checks pass, Tisane proceeds to the next phase.

6.2 Candidate statistical model generation

A GLM/GLMM is comprised of a model effects structure, family function, and link function. The model effects structure may consist of main, interaction, and random effects. Tisane utilizes variables' conceptual relationships to infer candidate main and interaction effects and data measurement relationships to infer random effects. Tisane infers family and link functions based on the data type of the DV in the query. The candidate statistical models that Tisane generates, based on the graph and query, seed an interactive disambiguation process.

The purpose of identifying candidate main effects beyond the ones analysts may have specified is to provoke consideration of erroneously omitted variables that are conceptually relevant and pre-empt potential confounding and multicollinearity issues that may arise.

6.2.1 Deriving Candidate Main Effects. In a query to infer a statistical model, analysts specify a single dependent variable and a set of one or more IVs. After passing the checks described in 6.1, the query's independent variables are considered candidates. In addition, Tisane derives three additional sets of candidate main effects intended to control for confounding variables in the output statistical model[8]. The first two sets below are from the "modified disjunctive cause criterion" [66]:

- **Causal parents.** For each IV in the query, Tisane finds its causal parents (see 5(a)).
- **Possible causal omissions.** Tisane looks to see if any other variables not included as IVs cause the DV (see in 5(b)). They are relevant to the DV but may have been erroneously omitted.
- **Possible confounding associations.** For each IV, Tisane looks for variables that are associated with both the IV and the DV (see in 5(c)). Because associations between variables can have multiple underlying causal structures, Tisane recommends variables with associative relationships with caution. Tisane issues a warning describing when not to include such a variable in the GUI (see Figure 3 in supplemental material).

Using the above rules, Tisane suggests a set of variables that are likely confounders of the variables of interest expressed in the query. There may be additional confounders due to unmeasured or unexpressed variables that are either not known or excluded from the graph. Tisane never automatically includes the candidate main effects in the output statistical model. Analysts must always specify a variable as an IV in the query or accept a suggestion (*DG3 - Guidance and control*).

If a graph only contains associates edges then the candidate main effects Tisane suggests are those that are directly associated with both the DV and an IV. If a graph has only causal edges, Tisane would suggest variables that directly cause the DV but were omitted from the query and the causal parents of IVs in case the parents exert causal influence on the DV through the IV or another variable that is not specified.

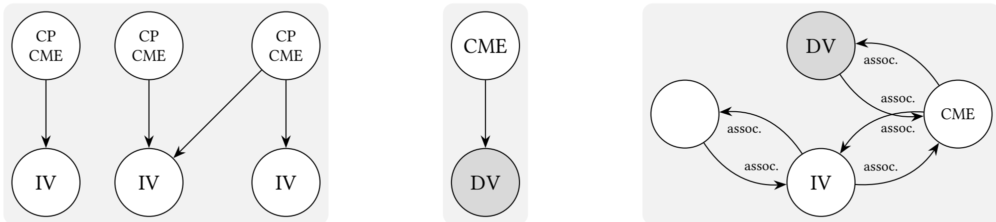
The total set of main effects, including variables the analyst has specified as IVs in their query and candidate main effects, are used to derive candidate interaction effects and random effects, which we discuss next.

6.2.2 Deriving Candidate Interaction Effects. An interaction between variables means that the effect of one variable (the *moderated* variable) on a *target* variable is moderated by another (non-empty) set of variables (the *moderating* variables). Tisane's SDSL already provides a primitive, `moderates`, to express interactions. As such, Tisane's goal in suggesting candidate interaction effects is to help analysts avoid omissions of conceptual relationships that are pertinent to an analyst's research questions or hypotheses (*DG1 - Conceptual knowledge*). Candidate interaction effects are the interaction nodes whose (i) moderated and moderating variables include two or more candidate main effects and (ii) target variable is the query's DV.

6.2.3 Deriving Candidate Random Effects. Random effects occur when there are clusters in the data, which occur when we have repeated measures, nested hierarchies, or non-nesting composition (as defined in Section 5.2.2). Tisane implements Barr et al.'s recommendations for specifying the maximal random effects structure of linear mixed effects models for increasing the generalizability of statistical results [2, 3].

To derive random effects, Tisane focuses on the data measurement edges in the graph IR. Using the graph IR, Tisane identifies unit nodes, looks for any nesting edges among them, and determines within- or between-subjects measures based on the frequency of observations for units. From these, Tisane generates random intercepts of units for the unit's measures that are between-subjects as well as the unit's measures that are within-subjects where each instance of the unit has only one observation per value of another variable. Tisane generates random slopes of a unit and its measure for all measures that are within-subjects where each instance of the unit has multiple observations per value of another variable. For interaction effects, random slopes are included for the largest subset of within-subjects variables (see [2]). Tisane handles correlation of random slopes and intercepts during disambiguation (section 6.3). Maximal random effects may lead to model convergence issues that

[8]Tisane currently treats each input IV as a separate "exposure" variable for which to identify confounders. Tisane then combines all confounders into one statistical model.



(a) *causal parents* (b) *possible causal omissions* (c) *possible confounding associations*

Figure 5: Graphs demonstrating causal parents, possible causal omissions, and possible confounding associations. In graphs (a) and (b) (left and middle), all edges are causal. Independent variables are marked "IV", discovered candidate main effects "CME", dependent variables "DV", and causal parents "CP".

analysts address by later removing or adding independent variables and random effects. Nevertheless, starting with a maximal valid model is important for ensuring that future revisions are also valid (*DG2 - Validity*).

6.2.4 Deriving Candidate Family and Link Functions. The DV's data type determines the set of candidate family and link functions. For example, numeric variables are not allowed to have binomial or multinomial distributions. Similarly, nominal variables are not allowed to have Gaussian distributions. Furthermore, each family has a set of possible link functions. For example, a Gaussian family distribution may have an Identity, Log, or Square Root link function. The statistics literature documents possible combinations of family and link functions for specific data types [41].

Tisane includes common family distributions as candidate families and their applicable link functions. In its current implementation, Tisane relies on `statsmodels` [53] for GLMs and `pymer4` [25] for GLMMs. As such, Tisane is limited to the family and link function pairings implemented in these libraries. As `statsmodels`' and `pymer4`'s support for GLMs grows in the future, Tisane can be extended.

6.3 Eliciting Analyst Input for Disambiguation

The disambiguation process provides an opportunity for analysts to explore the space of generated models based on their original query. Given our design considerations to prioritize conceptual knowledge (*DG1 - Conceptual knowledge*) and give analysts guidance (*DG3 - Guidance and control*), we designed a GUI to scaffold analysts' reasoning and elicit their input. For versatility, we implemented Tisane's GUI using Plotly Dash [10]. Analysts can either execute their Tisane programs and use the GUI inside a Jupyter notebook (no additional widgets needed) or run their Tisane programs in an IDE or terminal, in which case Tisane will open the GUI in a web browser. Figure 2 gives an overview of the GUI.

Candidate statistical models are organized according to (i) independent variables (main effects and interaction effects), (ii) data clustering (random effects), and (iii) data distribution (family and link functions). In the main effects tab, Tisane asks analysts if they would like to include additional or substitute main effects that Tisane infers to be conceptually relevant. In the interaction effects tab, Tisane suggests moderating relationships to include but does not automatically include them because analysts may not have specific hypotheses involving interactions (*DG3 - Guidance and control*). If analysts do not specify any moderating relationships, Tisane does not suggest any interaction effects, preventing analysts from including arbitrary interactions that may be conceptually unfounded (*DG1 - Conceptual knowledge, DG2 - Validity*).

In the data clustering tab, Tisane shows analysts which random effects it automatically includes based on the selected main and interaction effects. Unlike main and interaction effects, Tisane automatically includes random effects in order to maximize model generalizability (*DG2 - Validity*). If there is a random slope and random intercept pertaining to the same unit, Tisane asks analysts if they should be correlated or uncorrelated. We provide this option because analysts may have relevant domain expertise to make this decision (*DG3 - Guidance and control*). By default, Tisane correlates the random slope and random intercept.

The final tab, data distribution, helps analysts examine their data and select an initial family and link function to try. Appropriate selection of family and link functions depends on the data type of the dependent variable and the distribution of model residuals. Therefore, the selection can only be assessed after choosing a family and link function in the first place.

For an initial statistical model to consider, Tisane narrows the set of family functions considered based on the declared data type of variables (see 6.2.4) and lightweight viability checks, such as ensuring that a Poisson distribution is only applicable for variables that have nonnegative integer values. Tisane asks questions designed to uncover more semantically meaningful data types (e.g., counts) than are provided at variable declaration. Analysts without data can answer these questions as they are planning their studies (*DG4 - Statistical planning*). For the selected family candidate, Tisane automatically selects the default link function based on the defaults for `statsmodels` [47] and `pymer4` [25]. Analysts can then choose a different link function, as long as it is supported[9].

6.4 Output

There are two outputs of the interactive compilation: (i) an executable modeling script and (ii) a log of GUI choices. To increase transparency of the authoring process, Tisane provides a log of user selections in the GUI as documentation which the analyst can include in pre-registrations, for example (*DG4 - Statistical planning*).

[9] See supplemental material for a complete listing of Tisane's supported family and link function pairings.

6.4 Output

There are two outputs to interactive compilation: (i) a log of GUI choices and (ii) an executable modeling script. To increase transparency of the authoring process, Tisane provides a log of user selections in the GUI as documentation, which the analyst can include in pre-registrations, for example (*DG4 - Statistical planning*). In the output script, Tisane includes code to fit the model and plot residuals against fitted values in order to assess the appropriateness of family and link functions, as is typical when examining family and link functions. The output script also includes a comment explaining what to look for in the plots and an online resource for further reading. Should analysts revise their choice of family and link functions, they can re-generate a script through the Tisane GUI.

7 CASE STUDIES WITH RESEARCHERS

Given Tisane's novel focus on deriving and guiding analysts toward valid statistical models, we assessed how Tisane affects data analysis practices in three case studies with researchers. The following research questions guided the evaluation:

- **RQ1 - Workflow.** How does Tisane's programming and interaction model affect how analysts author models? Specifically, what does Tisane make noticeably easier or more difficult when conducting an analysis?
- **RQ2 - Cognitive fixation.** Where do researchers report spending more time or attention when using Tisane? How does this compare to their fixation during analyses typically?
- **RQ3 - Future possibilities.** When do researchers imagine using Tisane in future projects, if at all? What additional support do researchers want from Tisane?

We recruited researchers through internal message boards and individual contacts. We intentionally recruited researchers at different stages of the research process—study planning, data analysis for publication, and ongoing model building and maintenance. We believed this could help us more holistically evaluate Tisane's impact on data analysis. We met with researchers over Zoom (R1, R3) and in person (R2) to discuss their use cases, observe them use Tisane for the first time, and ask for open-ended feedback. We pointed researchers to the Tisane tutorial for installation instructions and examples but otherwise encouraged the researchers to work independently. We answered any questions researchers had while using Tisane. Each study session lasted approximately 2 hours. At the end, two of the three researchers (R1, R3) said they planned to use Tisane again over the next two months.

7.1 Case Study 1: Planning a new study

R1, a clinical psychology PhD student, had recently submitted a paper and was planning a follow-up. R1 reported that she had never taken a formal class on modeling techniques but taught herself for

In the output script, Tisane includes code to fit the model and plot residuals against fitted values in order to assess the appropriateness of family and link functions, as is typical when examining family and link functions. The output script also includes a comment explaining what to look for in the plots and an online resource for further reading. Should analysts revise their choice of family and link functions, they can re-generate a script through the Tisane GUI.

7 CASE STUDIES WITH RESEARCHERS

Given Tisane's novel focus on deriving and guiding analysts toward valid statistical models, we assessed how Tisane affects data analysis practices in three case studies with researchers. The following research questions guided the evaluation:

- **RQ1 - Workflow** How does Tisane's programming and interaction model affect how analysts author models? Specifically, what does Tisane make noticeably easier or more difficult when conducting an analysis?
- **RQ2 - Cognitive fixation** Where do researchers report spending more time or attention when using Tisane? How does this compare to their fixation during analyses typically?
- **RQ3 - Future possibilities** When do researchers imagine using Tisane in future projects, if at all? What additional support do researchers want from Tisane?

We recruited researchers through internal message boards and individual contacts. We intentionally recruited researchers at different stages of the research process—study planning, data analysis for publication, and ongoing model building and maintenance. We believed this could help us more holistically evaluate Tisane's impact on data analysis. We met with researchers over Zoom (R1, R3) and in person (R2) to discuss their use cases, observe them use Tisane for the first time, and ask for open-ended feedback. We pointed researchers to the Tisane tutorial for installation instructions and examples but otherwise encouraged the researchers to work independently. We answered any questions researchers had while using Tisane. Each study session lasted approximately 2 hours. At the end, two of the three researchers (R1, R3) said they planned to use Tisane again over the next two months.

7.1 Case Study 1: Planning a new study

R1, a clinical psychology PhD student, had recently submitted a paper and was planning a follow-up. R1 reported that she had never taken a formal class on modeling techniques but taught herself for her last paper. Her general workflow involved consulting with and mirroring what others in her research group did even if she did not completely understand why. R1 did not program often but said she had "enough coding experience to understand this kind of...[sample program]." Although familiar with Python, R1 preferred M+ [39] and SPSS [58]. She was interested in using Tisane to brainstorm new studies and research questions.

Using Tisane. After installation, R1 read through one of the computational notebook examples available in the Tisane GitHub repository. While reading, R1 asked clarifying questions about the variable types and syntax. R1 explained that the `Design` class felt novel because she had never seen the concept of a study design in data analysis code before. When the first two authors explained that it was supposed to be the equivalent of the statement of a study design in a paper, R1 remarked that usually, she "[kept] that in [her] head, which [she] probably shouldn't" (**RQ2 - Cognitive fixation**). Without a concrete data set, R1 preferred to walk through more examples rather than author a script of her own.

While reading an example, R1 drew a parallel between the tabs in SPSS dialogs for specifying models and the tabs in the Tisane GUI, noting that SPSS had a tab for control variables. R1 also wanted the ability to distinguish between "control variables" and other independent variables in the Tisane GUI. R1 explained that this would map more closely to how psychologists think about analyses. Future work could incorporate additional language constructs, such as a new data type for controls, for different groups of users (**RQ3 - Future possibilities**).

At the end of the study session, R1 remarked how Tisane "fills in a lot of the...gaps" in data analysis (**RQ1 - Workflow**, **RQ2 - Cognitive fixation**). The first gap R1 discussed was the *programming gap* between scientists and statistical tools. R1 believed that, for scientists who were not comfortable with programming, "they should probably be running less complex models, or first learn how to code" even if the complex models would be most appropriate. The second gap R1 discussed was the *statistical knowledge gap* in tools. R1 explained that in her experience, R provides support for more complex models but little guidance for what those models or statistical tests should be, requiring "top down assumption[s]." Thus, to R1, Tisane bridged the gap between tools like SPSS and R by requiring minimal programming and providing modeling support. Put another way, Tisane bridged the gulf of execution [43] for R1 that previous tools had not.

7.2 Case Study 2: Analyzing data for a paper submission

R2, a computer science PhD student, had conducted a within-subjects study where 47 participants used four versions of an app for one week each (four weeks total). The motivating research question was how the different app designs led to psychological dissociation. Although R2 had expected to collect multiple survey responses for each participant each day, they only had aggregate daily self-report measures due to an error in the database management system. In the past, R2 reported having extensively explored their data and consulting others, but for this paper, they had not explored their data prior to fitting models because they felt more confident in their modeling skills. For analyses, R2 preferred R but had general Python programming experience. Prior to using Tisane, R2 had authored linear mixed effects models in R for their study. They were interested in using Tisane to check their analyses prior to submitting their paper to CHI.

Using Tisane. R2 wrote their scripts by adapting an example from the Tisane GitHub repository. As R2 considered which conceptual relationships to add, they reasoned aloud about if they should state causal or associative relationships between various measures and dissociation (**RQ2 - Cognitive fixation**). After some deliberation, they said, "I don't feel comfortable [making a causal statement]," and instead specified `associates_with` relationships. R1's hesitation to assert causal relationships confirms prior findings that specifying formal causal graphs is difficult for domain

researchers [60, 61, 67] and our design choice to allow for association edges. In addition, R2 was initially unsure about how to specify the `number_of_instances` for their measures since their original study design was unbalanced. After asking for clarification about `number_of_instances`, R2 declared all the measures with the parameter `number_of_instances` set equal to `date`.

Next, R2 ran their script and used the Tisane GUI in a browser window. Based on Tisane's recommended family and link functions, R2 realized the models they had previously authored in R using a Gaussian family were inappropriate. Due to a bug that we have since fixed, Tisane suggested a Poisson family that R2 used to generate a script, but this was an invalid choice given that not all dependent variable values were nonnegative integers. R2 explored other family distributions and generated a new script using an Inverse Gaussian family. When executed, the second output script issued an error due to the model inference algorithm failing to converge. R2 made a note to look into this model further on their own.

Once finished using Tisane, R2 commented that their analysis with Tisane was more streamlined (**RQ1 - Workflow**) in contrast to their very first paper where they had tried "every single kind of model that [they] could" until finding "the one that fits best," even if it was "one that no one would have heard of." R2 also stated they would be interested in using Tisane earlier in their analysis process in the future (**RQ3 - Future possibilities**). Based on their experience with Tisane, R2 questioned their previously authored linear mixed effects model, and said it was "unnerving" to discover an issue so close to a deadline. At the same time, they expressed, "if it's incorrect, I should know before I submit." A day after the study, R2 contacted the authors to inform them that they had decided to update their analyses from linear mixed effects models to generalized linear mixed effects models. They reported using the Inverse Gaussian family after visualizing and checking the distribution of residuals with help from the output Tisane script. The Inverse Gaussian family was appropriate because their dependent variable's values were all nonnegative and displayed a slight positive skew. R2's experience with Tisane suggests that Tisane can help researchers catch errors and lead them to re-examine their data, assumptions, and conclusions.

7.3 Case Study 3: Developing models to inform future models

Employed on a research team, R3 analyzes health data at the county, state, and national levels to estimate health expenditure and inform public policy. R3 develops initial models that are used to validate and generate estimates for larger, more comprehensive models. Due to the scale of data and established collaborative workflows, R3 typically works in a terminal or RStudio through a computing cluster and had very little experience with Python. Despite working on statistical models every day, R3 described himself as "not...a great modeler." R3 was interested in using Tisane to determine what variables to include as random effects in a model.

Using Tisane. R3 used Tisane in a local Jupyter notebook as well as on his team's cluster. R3 used the Tisane API overview reference material on GitHub to start writing his program, which involved copying and pasting the functions with their type signatures and then modifying them to match his dataset and incrementally running the program. The most common mistake R3 made while authoring his Tisane program was to refer to variables using the string names in the dataset (e.g., `"year"`) instead of the variable's alias (e.g., `year_id`), an idiom common in R but not in Python.

While authoring his Tisane program, R3 found the `number_of_instances` parameter redundant, especially because his data is always "square." Every `state_name` in his data set had 30 rows of data, corresponding to the `year_id`s 1990-2019. This is in contrast to R2, whose study design was unbalanced and resulted in variable numbers of observations per participant that needed to be aggregated. Based on R3's feedback, we added functionality to infer `number_of_instances` for each unit, which analysts can inspect by printing the variable.

While giving open-ended feedback on Tisane, R3, similar to R1, liked how Tisane helped "fill [the] gap in...[his] knowledge" (**RQ2 - Cognitive fixation**). Given the diversity of models R3 works with, R3 found Tisane's focus on GLMs and GLMMs a "little limiting" and also wished to make Tisane "run without...the mouse" in a script, as is typical in his workflow (**RQ1 - Workflow**). Specifically, R3 described how he and his collaborators typically want to explore a space of models and run them in parallel. Nevertheless, R3 foresaw using Tisane in three types of modeling tasks common in his work: (i) exploratory modeling to determine if there are any interesting relationships between variables, (ii) authoring and comparing multiple models for prediction, and (iii) working out the precise model specification after identifying variables of interest (**RQ3 - Future possibilities**).

7.4 System changes and Takeaways

We fixed bugs and iterated on Tisane's GUI based on feedback from researchers. The largest change we made was to the data distributions tab. The data distributions tab we tested with researchers visualized the dependent variables against simulated distributions of family functions and included the results of the Shapiro-Wilk and D'Agostino and Pearson's normality tests. All three researchers reported becoming more aware of their data due to the visualizations. However, researchers' enthusiasm for the feature made us wary that visualizing the simulated data could mislead less careful analysts to believe that family and link functions pertain to variable distributions rather than the distributions of the model's residuals. To avoid such errors while still helping analysts become more aware of their data, we removed the simulated visualizations and normality tests and instead provide questions about the semantic nature of the dependent variable collected, as discussed in subsection 6.3.

Overall, Tisane streamlines the analysis process (**RQ1 - Workflow**) in part because researchers report formalizing their conceptual knowledge into statistical models more directly (R1, R2). Although Tisane does not eliminate the need for model revision, Tisane may scope the revisions analysts consider to significant issues instead of details that may detract from the analysis goals (R2). Additionally, researchers reported a perceived shift in their attention from keeping track of and analyzing all possible modeling paths to their research questions and data assumptions (**RQ2 - Cognitive fixation**) while planning a new study and analysis (R1) as well as while preparing a research manuscript (R2). Future adoption of Tisane may depend on the complexity of analyses (**RQ3**

- **Future possibilities**) (R3). For instance, Tisane may provide a streamlined alternative to false starts due to misspecifications for simpler analyses (R1, R2, R3). For more complex models and studies, Tisane may act more as a prototyping tool for statistical models, helping researchers start at a reasonable model that they can then revise (R2, R3).

8 DISCUSSION

In this work, our motivating question was "**How might we derive (initial) statistical models from knowledge about concepts and data collected?**" This question presented two challenges: (i) how to elicit the information necessary to author a GLM/GLMM and (ii) how to computationally infer a valid statistical model given this information. To address the first challenge, we designed and developed **Tisane's study design specification language**. To address the second challenge, we developed a **graph representation** that Tisane traverses to derive candidate statistical models. We also developed a novel interaction model that involves **interactive compilation** to address both challenges. Throughout the design process, we employed statistical methods and theory, theories of how people analyze data, and an iterative design process with researchers. When using Tisane, researchers in our case studies reported focusing more on their analysis goals and becoming more aware of their assumptions and even identified and avoided previous analysis mistakes. Below, we reflect on future opportunities for Tisane to further enhance statistical practice, interpretation of results, and the end-to-end data analysis pipeline.

Design for statistical validity. Campbell's theory of validity – encompassing statistical conclusion, internal, external, and construct validity [6, 11] – has influenced disciplines widely (e.g., [54]), including epidemiology (e.g., [37]), software engineering (e.g., [42]), and psychology (e.g., [6]). Viewed through the Campbellian framework, Tisane helps analysts avoid four common threats to statistical conclusion and external validity: (i) violation of statistical method assumptions, (ii) fishing for statistical results, (iii) not accounting for the influence of specific units, and (iv) overlooking the influence of data collection procedures on outcomes [11].

Tisane fills a need to align analysts' conceptual models with the statistical models they want to implement but find difficult to express with the current tools available. By integrating conceptual, data, and statistical concerns, Tisane facilitates the hypothesis formalization [26] process, which can be an error-prone and cognitively demanding process that existing tools do not yet support.

In the future, we plan to develop additional strategies for enhancing the validity of analyses authored with Tisane. As discussed in Section 6.3, our current approach to family and link functions is only an initial step. We look forward to developing and comparing multiple strategies for scaffolding the family and link function selection and revision process. For example, what if the Tisane GUI allowed analysts to fit multiple models that varied in their family and link functions, plotted each model's residuals against the predicted values, and gave analysts visual guides for comparing models? To avoid false discovery rate inflation, Tisane could partition analysts' data, fit models to only a subset, and output a script for fitting a selected model using another subset. Although possible for large datasets, this strategy would encounter limited statistical power for smaller datasets. Alternatively, what if Tisane calculated Bayes factors for variables in the models [12, 17, 48] after analysts tried multiple family and link combinations? Carefully balancing statistical rigor and usefulness to domain researchers who may be statistical non-experts deserves careful consideration.

Prevent p-hacking. Tisane generates a space of possible models from a set of conceptual and data measurement relationships. By querying Tisane for a model, analysts will only consider a set of models that are compatible with these relationships. As a result, Tisane helps analysts avoid unintentional p-hacking. Especially motivated p-hackers could specify questionable conceptual and data measurement relationships to manipulate the space of models Tisane generates. However, in this case, review or inspection of the Tisane program during pre-registration or peer-review, for example, could identify such malicious practices. In these ways, we believe that p-hacking is more difficult in Tisane than in existing analysis tools.

Future work to further discourage p-hacking could extend Tisane to conduct a sensitivity analysis on the space of possible models and only report models and results that are robust across the space. A challenge in this approach is that statistical non-experts may need more scaffolding to understand and interpret the results of sensitivity analyses.

Scaffold interpretation of statistical results. Tisane's focus is on authoring GLMs/GLMMs, but accurate interpretation is also necessary. For instance, analysts may need help interpreting what their statistical models and results mean in relation to their input conceptual models. Do the results suggest their conceptual model is correct? What kind of inferences should they make? Future work should address these interpretation challenges, which may require eliciting hypotheses and expected results from analysts.

Although researchers in our pilot or case studies did not presume Tisane helped with formal causal analysis, the ability to express causal relationships (`causes`) may lead some analysts to erroneously assume that their models assess causality. Changing the name of the language construct and/or building out support to interpret GLM/GLMM results may resolve this concern. One way to support accurate interpretation and reporting could be to output a figure representing the input conceptual model along with visual summaries of the data and/or statistical model for direct inclusion in publications. Tisane could also allow analysts to annotate their disambiguation decisions with their own rationale and provide a richer log of selections than currently supported. Tisane could even accept these augmented logs to save the state of the GUI in between analysis sessions.

Provide discipline-specific language support. When designing Tisane's study design specification language, we analyzed and developed language constructs common across existing libraries for study design (see supplemental material). In our case studies, we found that researchers had different conventions for describing their data ("unbalanced" (R2) vs. always "square" (R3)) and models (e.g., "controls" (R1) vs. "covariates" (R3)). This observation suggests opportunities to increase the usability of Tisane's SDSL by

providing syntactic sugar that may be more familiar to users. In the future, we plan to formally assess usability and identify "natural" programming [40] constructs that differ across disciplines.

An additional strategy for supporting more discipline-specific programming models and analysis needs is to integrate Tisane with existing study design libraries. For example, HCI researchers may find the lower-level randomization details that Touchstone2's interface [15] provides more natural. A system could summarize these details into the higher-level data measurement relationships in Tisane to bootstrap interactive compilation and output a possible statistical model. In this way, Tisane's graph IR can provide a "shared representation" [22] between study design tools and Tisane.

Integrate into end-to-end analysis workflows. Researchers in our case study were more comfortable with R. R1 and R3 expressed it could be helpful to have Tisane in R as an RStudio plug-in, for example, to fit into their workflows. As more users adopt Tisane, we will add an implementation in R.

Moreover, analysts may need to add or remove variables from Tisane's generated statistical models in order to accommodate model convergence failures, new data, or changing domain knowledge. However, adding or removing variables may subtly change the hypotheses analysts can test statistically. We look forward to extending Tisane to support model iteration, which presents two technical challenges: (i) recognizing when conceptual revisions are necessary and (ii) identifying and suggesting model changes that maintain conceptual validity or, at the very least, quantify conceptual shifts. Furthermore, in the model revision process, analysts may consider multiple alternatives. As R3 described, he preferred to run multiple variations of a model and compare them, a workflow akin to a multiverse analysis [59]. Given that Tisane already generates a combinatorial space of candidate statistical models, Tisane could generate a multiverse script for Boba [33] instead. A multiverse could help check the robustness of findings, and Boba's visual analyzer could help analysts further develop an understanding of their data and modeling choices. A multiverse may also help analysts explore and compare family and link combinations as well.

Tisane is one tool designed to enable analysts with limited statistical expertise to author valid statistical models. Tisane enables future possibilities and raises open research questions for creating an ecosystem of analysis tools that align tool interfaces with analysts' conceptual goals.

ACKNOWLEDGMENTS

We thank Yang Liu and Younghoon Kim for early feedback on Tisane's API; Leilani Battle, Matthew Conlen, Sherry Wu, and Rock Pang for feedback on early drafts of this paper; Tyler McCormick for feedback on the project and paper; and Maureen Daum for insightful conversations about how Tisane's graph IR relates to data models. We also thank the anonymous reviewers who provided valuable feedback.

REFERENCES

[1] Eytan Bakshy, Dean Eckles, and Michael S Bernstein. 2014. Designing and deploying online field experiments. In *Proceedings of the 23rd international conference on World wide web*. ACM, 283–292.
[2] Dale J Barr. 2013. Random effects structure for testing interactions in linear mixed-effects models. *Frontiers in psychology* 4 (2013), 328.
[3] Dale J Barr, Roger Levy, Christoph Scheepers, and Harry J Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language* 68, 3 (2013), 255–278.
[4] Graeme Blair, Jasper Cooper, Alexander Coppock, and Macartan Humphreys. 2019. Declaring and diagnosing research designs. *American Political Science Review* 113, 3 (2019), 838–859.
[5] Benjamin M Bolker, Mollie E Brooks, Connie J Clark, Shane W Geange, John R Poulsen, M Henry H Stevens, and Jada-Simone S White. 2009. Generalized linear mixed models: a practical guide for ecology and evolution. *Trends in ecology & evolution* 24, 3 (2009), 127–135.
[6] Donald T Campbell and Julian C Stanley. 2015. *Experimental and quasi-experimental designs for research*. Ravenio Books.
[7] Herbert H Clark. 1973. The language-as-fixed-effect fallacy: A critique of language statistics in psychological research. *Journal of verbal learning and verbal behavior* 12, 4 (1973), 335–359.
[8] Jacob Cohen. 1988. Statistical power analysis for the social sciences. (1988).
[9] Jacob Cohen, Patricia Cohen, Stephen G West, and Leona S Aiken. 2013. *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge.
[10] Plotly Dash Community. [n. d.]. Plotly Dash. https://plotly.com/dash/
[11] Thomas D Cook, Donald Thomas Campbell, and William Shadish. 2002. *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin Boston, MA.
[12] Claudia Czado and Adrian E Raftery. 2006. Choosing the link function and accounting for link uncertainty in generalized linear models using Bayes factors. *Statistical Papers* 47, 3 (2006), 419–442.
[13] Joshua R De Leeuw. 2015. jsPsych: A JavaScript library for creating behavioral experiments in a Web browser. *Behavior research methods* 47, 1 (2015), 1–12.
[14] H2OAutoML developers. 2021. H2OAutoML v3.32.1.6 Documentation: Model Explainability. (2021). https://docs.h2o.ai/h2o/latest-stable/h2o-docs/explain.html#model-explainability
[15] Alexander Eiselmayer, Chatchavan Wacharamanotham, Michel Beaudouin-Lafon, and Wendy Mackay. 2019. Touchstone2: An Interactive Environment for Exploring Trade-offs in HCI Experiment Design. (2019).
[16] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and Robust Automated Machine Learning. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf
[17] Alan E Gelfand and Dipak K Dey. 1994. Bayesian model choice: asymptotics and exact calculations. *Journal of the Royal Statistical Society: Series B (Methodological)* 56, 3 (1994), 501–514.
[18] Andrew Gelman. 2005. Why I don't use the term "fixed and random effects". https://statmodeling.stat.columbia.edu/2005/01/25/why_i_dont_use/
[19] Andrew Gelman and Jennifer Hill. 2006. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.
[20] Sander Greenland, Judea Pearl, and James M Robins. 1999. Causal diagrams for epidemiologic research. *Epidemiology* (1999), 37–48.
[21] Garrett Grolemund and Hadley Wickham. 2014. A cognitive interpretation of data analysis. *International Statistical Review* 82, 2 (2014), 184–204.
[22] Jeffrey Heer. 2019. Agency plus automation: Designing artificial intelligence into interactive systems. *Proceedings of the National Academy of Sciences* 116, 6 (2019), 1844–1850.
[23] Rick H Hoyle. 1995. *Structural equation modeling: Concepts, issues, and applications*. Sage.
[24] SAS Institute Inc. 2021. SAS. https://www.sas.com/
[25] Eshin Jolly. 2018. Pymer4: connecting R and Python for linear mixed modeling. *Journal of Open Source Software* 3, 31 (2018), 862.
[26] Eunice Jun, Melissa Birchfield, Nicole De Moura, Jeffrey Heer, and Rene Just. 2022. Hypothesis Formalization: Empirical Findings, Software Limitations, and Design Implications. In *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 29. Issue 1. "https://arxiv.org/abs/2104.02712"
[27] 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.
[28] Moe Kayali, Babak Salimi, and Dan Suciu. 2020. Demonstration of inferring causality from relational databases with CaRL. *Proceedings of the VLDB Endowment* 13, 12 (2020), 2985–2988.
[29] Ita GG Kreft, Ita Kreft, and Jan de Leeuw. 1998. *Introducing multilevel modeling*. Sage.
[30] Erin LeDell and Sebastien Poirier. 2020. H2O AutoML: Scalable Automatic Machine Learning. *7th ICML Workshop on Automated Machine Learning (AutoML)* (July 2020). https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf
[31] 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.

[32] 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).

[33] Yang Liu, Alex Kale, Tim Althoff, and Jeffrey Heer. 2020. Boba: Authoring and visualizing multiverse analyses. *IEEE Transactions on Visualization and Computer Graphics* (2020).

[34] James Lloyd, David Duvenaud, Roger Grosse, Joshua Tenenbaum, and Zoubin Ghahramani. 2014. Automatic construction and natural-language description of nonparametric regression models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[35] Steson Lo and Sally Andrews. 2015. To transform or not to transform: Using generalized linear mixed models to analyse reaction time data. *Frontiers in psychology* 6 (2015), 1171.

[36] Wendy E Mackay, Caroline Appert, Michel Beaudouin-Lafon, Olivier Chapuis, Yangzhou Du, Jean-Daniel Fekete, and Yves Guiard. 2007. Touchstone: exploratory design of experiments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1425–1434.

[37] Ellicott C Matthay and M Maria Glymour. 2020. A graphical catalog of threats to validity: Linking social science with epidemiology. *Epidemiology (Cambridge, Mass.)* 31, 3 (2020), 376.

[38] Richard McElreath. 2020. *Statistical rethinking: A Bayesian course with examples in R and Stan*. CRC press.

[39] Muthén & Muthén. [n. d.]. MPlus. https://www.statmodel.com/

[40] Brad A Myers, John F Pane, and Andy Ko. 2004. Natural programming languages and environments. *Commun. ACM* 47, 9 (2004), 47–52.

[41] John Ashworth Nelder and Robert WM Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)* 135, 3 (1972), 370–384.

[42] Amadeu Anderlin Neto and Tayana Conte. 2013. A conceptual model to address threats to validity in controlled experiments. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*. 82–85.

[43] Don Norman. 2013. *The design of everyday things: Revised and expanded edition*. Basic books.

[44] Judea Pearl. 1995. Causal diagrams for empirical research. *Biometrika* 82, 4 (1995), 669–688.

[45] Judea Pearl. 1995. Causal diagrams for empirical research. *Biometrika* 82, 4 (1995), 669–688.

[46] Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress* 19 (2000).

[47] Josef Perktold, Skipper Seabold, Jonathan Taylor, and statsmodels developers. 2020. Statsmodels v0.10.2 Reference Guide. (2020). "https://www.statsmodels.org/stable"

[48] Adrian E Raftery. 1996. Approximate Bayes factors and accounting for model uncertainty in generalised linear models. *Biometrika* 83, 2 (1996), 251–266.

[49] William S Robinson. 1950. Ecological correlations and the behavior of individuals. *Sociological Review* 15 (1950), 351–357.

[50] Donald B Rubin. 2004. Teaching statistical inference for causal effects in experiments and observational studies. *Journal of Educational and Behavioral Statistics* 29, 3 (2004), 343–367.

[51] Babak Salimi, Harsh Parikh, Moe Kayali, Lise Getoor, Sudeepa Roy, and Dan Suciu. 2020. Causal relational learning. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 241–256.

[52] Michel F Sanner et al. 1999. Python: a programming language for software integration and development. *J Mol Graph Model* 17, 1 (1999), 57–61.

[53] 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.

[54] William R Shadish. 2010. Campbell and Rubin: A primer and comparison of their approaches to causal inference in field settings. *Psychological methods* 15, 1 (2010), 3.

[55] N.J.A. Sloane and R.H. Hardin. 2017. Gosset: A General-purpose program for designing experiments. http://neilsloane.com/gosset/

[56] Peter Spirtes. 1994. Conditional independence in directed cyclic graphical models for feedback. (1994).

[57] Peter Spirtes, Thomas Richardson, Christopher Meek, Richard Scheines, and Clark Glymour. 1996. Using d-separation to calculate zero partial correlations in linear models with correlated errors. *Publisher: Carnegie Mellon University* (1996).

[58] IBM SPSS. 2021. SPSS Software. https://www.ibm.com/analytics/spss-statistics-software

[59] Sara Steegen, Francis Tuerlinckx, Andrew Gelman, and Wolf Vanpaemel. 2016. Increasing transparency through a multiverse analysis. *Perspectives on Psychological Science* 11, 5 (2016), 702–712.

[60] Etsuji Suzuki, Tomohiro Shinozaki, and Eiji Yamamoto. 2020. Causal diagrams: pitfalls and tips. *Journal of epidemiology* (2020), JE20190192.

[61] Etsuji Suzuki and Tyler J VanderWeele. 2018. Mechanisms and uncertainty in randomized controlled trials: A commentary on Deaton and Cartwright. *Social science & medicine (1982)* 210 (2018), 83–85.

[62] Emi Tanaka. 2021. Edibble: An R-package to construct designs using the grammar of experimental design. https://github.com/emitanaka/edibble

[63] R Core Team et al. 2013. R: A language and environment for statistical computing. (2013).

[64] Johannes Textor, Juliane Hardt, and Sven Knüppel. 2011. DAGitty: a graphical tool for analyzing causal diagrams. *Epidemiology* 22, 5 (2011), 745.

[65] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. 2013. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms. In *Proc. of KDD-2013*. 847–855.

[66] Tyler J VanderWeele. 2019. Principles of confounder selection. *European journal of epidemiology* 34, 3 (2019), 211–219.

[67] Priscilla Velentgas, Nancy A Dreyer, Parivash Nourjah, Scott R Smith, Marion M Torchia, et al. 2013. Developing a protocol for observational comparative effectiveness research: a user's guide. (2013).