Empowering domain experts to author valid statistical analyses

Eunice Jun
Paul G. Allen School of Computer Science & Engineering, University of Washington
Seattle, WA, USA
emjun@cs.washington.edu

ABSTRACT

Reliable statistical analyses are critical for making scientific discoveries, guiding policy, and informing decisions. To author reliable statistical analyses, integrating knowledge about the domain, data, statistics, and programming is necessary. However, this is an unrealistic expectation for many analysts who may possess domain expertise but lack statistical or programming expertise, including many researchers, policy makers, and other data scientists. How can our statistical software help these analysts? To address this need, we first probed into the cognitive and operational processes involved in authoring statistical analyses and developed the theory of *hypothesis formalization*. Authoring statistical analyses is a dual-search process that requires grappling with assumptions about conceptual relationships and iterating on statistical model implementations. This led to our key insight: statistical software needs to help analysts translate what they know about their domain and data into statistical modeling programs. To do so, statistical software must provide programming interfaces and interaction models that allow statistical non-experts to express their analysis goals accurately and reflect on their domain knowledge and data. Thus far, we have developed two such systems that embody this insight: Tea and Tisane. Ongoing work on rTisane explores new semantics for more accurately eliciting analysis intent and conceptual knowledge. Additionally, we are planning a summative evaluation of rTisane to assess our hypothesis that this new way of authoring statistical analyses makes domain experts more aware of their implicit assumptions, able to author and understand nuanced statistical models that answer their research questions, and avoid previous analysis mistakes.

CCS CONCEPTS

• **Human-centered computing → User interface toolkits**.

KEYWORDS

statistical analysis; end-user programming; domain-specific languages; mixed-initiative systems; transparent statistics; validity

ACM Reference Format:
Eunice Jun. 2022. Empowering domain experts to author valid statistical analyses. In *The Adjunct Publication of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22 Adjunct), October 29-November 2, 2022, Bend, OR, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3526114.3558530

1 INTRODUCTION

Statistical analyses are critical to scientific research, policy making, and decision making. Scientists develop, compare, and assess theories using statistical models. Policy makers track disease, inform health recommendations, and allocate resources using statistical models. Individuals author analyses to inform their financial investments, health goals, and other daily decisions. Unreliable statistical models can lead to findings that do not generalize or reproduce, spurious estimations of disease spread, and a misinformed public.

There are numerous statistical tools that provide significant mathematical and computational control (e.g., R [15], Python [12], SPSS [14], and SAS [3]). The challenge of developing accurate statistical models lies not in access to such mathematical tools but rather in accurately applying them in conjunction with domain theory, data collection, and statistical knowledge [5, 10]. When translating a research question or hypothesis into a statistical model implementation, analysts engage in a dual-search process we call *hypothesis formalization* [5]. The first search process is focused on refining conceptual understanding and hypotheses. The second process iterates on statistical model formulations in mathematics and code. These two processes inform one another because, ultimately, statistical analyses must align conceptual knowledge, observations about data, and statistical methods. Yet, based on an assessment of 20 statistical tools, we found that statistical tools are focused on just the one search process for statistical model iteration.

Analysts' goals and current statistical software interfaces are misaligned. Current tools lack support for expressing, assessing, and iterating on conceptual and data assumptions. As a result, analysts may select sub-optimal statistical models that do not fully represent their conceptual knowledge or data collection procedures, change their original conceptual hypotheses in favor of ones they know how to assess with familiar statistical methods, or falsely assume their statistical models represent their discipline and data accurately, all strategies we observed in a lab study of 24 data scientists.

To enable domain experts to author valid statistical analyses and avoid these mistakes, this dissertation develops a new programming and interaction model for hypothesis formalization: Analysts express conceptual and data assumptions. A system leverages this knowledge to formulate applicable statistical models. Three systems embody this principle. Thus far, we have developed and conducted initial evaluations of two systems. Tea [6] allows analysts to express and assess their assumptions about data while authoring commonly used Null Hypothesis Significance Tests (e.g., Student's t-test, ANOVA). Tisane [7] goes one step higher and allows analysts to express their conceptual knowledge and data measurement

UIST '22 Adjunct, October 29-November 2, 2022, Bend, OR, USA

ACM ISBN 978-1-4503-9321-8/22/10.
https://doi.org/10.1145/3526114.3558530

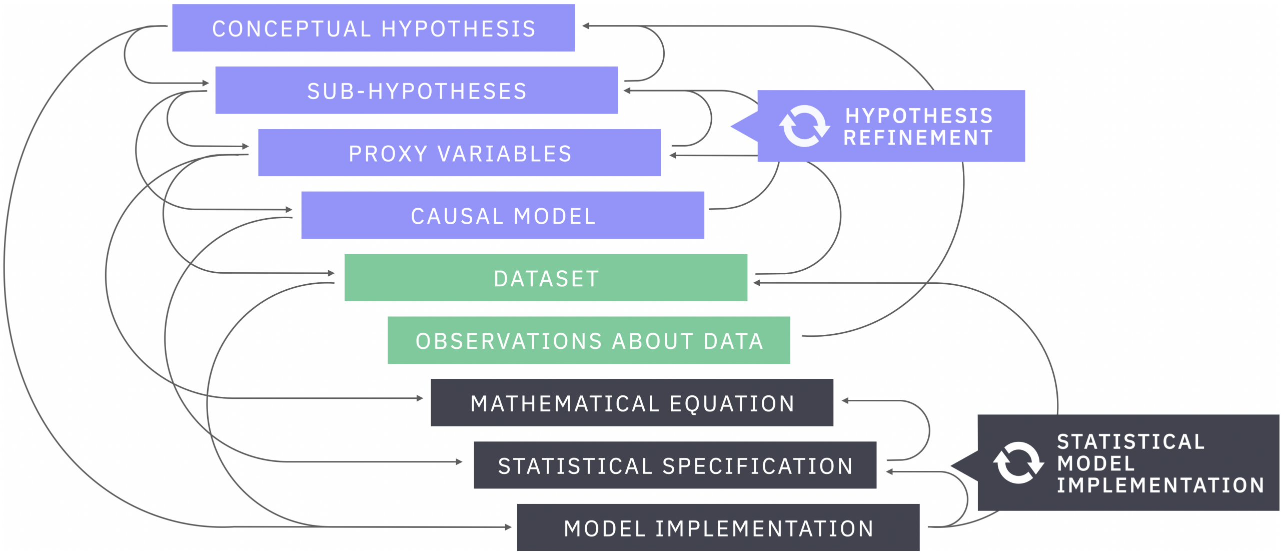


Figure 1: Hypothesis formalization. Hypothesis formalization is a dual-search process that involves hypothesis refinement and statistical model implementation in order to translate research questions and hypotheses into statistical model implementations. Currently, statistical software is only focused on statistical model implementation and overlooks the other half of hypothesis formalization. Tea [6] (see Section 2) pushes the level of abstraction up to assumptions of data (green). Tisane [7] (see Section 3 and rTisane (see Section 4 facilitate conceptual modeling (purple) to author statistical models, supporting hypothesis formalization end-to-end. Future work could further address iterations and updates to conceptual models from statistical models.

details to author generalized linear models with or without mixed effects, popular classes of statistical models that subsume the tests supported in Tea. We are currently building a third system rTisane to refine the semantics of Tisane so that analysts can more easily and accurately express their implicit conceptual models and become more aware of and reflect on their assumptions. We plan to finish developing and evaluate rTisane in a controlled lab study in the fall. As a whole, the dissertation contributes

- the theory of hypothesis formalization that describes how people translate their research questions and hypotheses into statistical models and suggests how statistical tools could better support this translation process,
- three systems for authoring statistical analyses that leverage analysts' implicit assumptions to guide the statistical authoring process, and
- (initial and ongoing) empirical results that illustrate the impact of this novel approach to authoring statistical analyses.

2 TEA: A HIGH-LEVEL LANGUAGE AND RUNTIME SYSTEM FOR NULL HYPOTHESIS SIGNIFICANCE TESTING

A wide variety of tools (such as SPSS [20], SAS [19], and JMP [17]), programming languages (e.g., R [18]), and libraries (including numpy [11], scipy [4], and statsmodels [13]), enable people to perform specific statistical tests. However, they do not address the fundamental problem that analysts may not know which statistical test to perform or how to verify that specific test assumptions about their data hold. To address this overlooked need, we designed Tea[1], a high-level declarative language for automating statistical test selection and execution. Tea is implemented as a Python package[2].

Tea provides a domain-specific language (DSL) for end-users to directly express their study designs, assumptions about the data, and hypotheses. Based on end-users' study designs and assumptions, Tea compiles these into logical constraints for selecting valid Null Hypothesis Significance Tests, executes these tests, and then outputs the results from these tests. properties. Figure 1 shows a sample Tea program.

```
import tea

tea.data("./UScrime.csv")

variables = [
    {
        'name': 'So',
        'data type': 'nominal',
        'categories': ['0', '1']
    },
    {
        'name': 'Prob',
        'data type': 'ratio'
    },
    {
        'name': 'Ineq',
        'data type': 'ratio'
    }
]

study_design = {
```

[1] named after Fisher's "Lady Tasting Tea" experiment [2]

[2] Tea is open-source: https://github.com/tea-lang-org/tea-lang

```
    'study type': 'observational study',
    'contributor variables': ['So', 'Prob'],
    'outcome variables': ['Prob', 'Ineq']
}

assumptions = {
    'Type I (False Positive) Error Rate': 0.05
}

tea.define_variables(variables)
tea.define_study_design(study_design)
tea.assume(assumptions)
tea.hypothesize(['Ineq', 'Prob'], ['Ineq ~ -Prob'])
```

Listing 1: Example Tea program. Analysts do not have to specify a statistical test. Instead, they express their assumptions about data and hypothesis directly. Tea will then infer and execute a set of valid statistical tests to assess the hypothesis.

Tea's key technical insight is that valid statistical test selection can be cast as a constraint satisfaction problem. This insight arises from two observations. First, common Null Hypothesis Significance Tests already make assumptions about when they are valid choices. For instance, the Student's t-test assumes that the data are normally distributed and the two groups being compared have equal variances. Tea uses these widely documented assumptions to select statistical tests. Second, these assumptions are operationally similar to logical constraints, suggesting that these assumptions should be encoded as constraints. Tea makes optimizations internally to address technical challenges of how exactly to represent statistical properties as constraints (e.g., Are booleans expressive enough? Do we need more complex theories? How do we avoid recursive constraints given that some statistical properties can only be evaluated using statistical tests?). For example, Tea avoids recursive constraints by introducing categories of statistical tests and directly evaluating those that check properties.

Tea is designed for statistical tests common to Null Hypothesis Significance Testing (NHST). While there are calls to incorporate other methods of statistical analysis [8, 9], Null Hypothesis Significance Testing (NHST) remains the norm in HCI and other disciplines. Therefore, Tea currently implements a module for NHST. In particular, Tea supports four classes of tests: correlation (parametric: Pearson's r, Pointbiserial; non-parametric: Kendall's τ, Spearman's ρ), bivariate mean comparison (parametric: Student's t-test, Paired t-test; non-parametric: Mann-Whitney U, Wilcoxon signed rank, Welch's t-test), multivariate mean comparison (parametric: F-test, Repeated measures one way ANOVA, Factorial ANOVA, Two-way ANOVA; non-parametric: Kruskal Wallis, Friedman), and comparison of proportions (Chi Square, Fisher's Exact). Tea also supports an implementation of bootstrapping [1].

We evaluated Tea against a corpus of 12 tutorials for statistical tests. We translated the tutorials into Tea programs, including any explicitly stated assumptions in the text as assumptions in the program. In nine cases, Tea replicated the tutorial authors' test choices and results. However, in the other three cases, Tea suggested more conservative alternatives and found that the authors made assumptions about the data not explicitly discussed in the tutorials or supported in the data. Thus, we conclude that Tea can replicate and even improve upon expert choices.

Furthermore, we simulated a comparison between Tea and novices. The simulation is based on two observations: (i) statistical novices often try to run and interpret results from any statistical test that executes and (ii) existing software packages will execute and provide results even if the test is not a valid choice for the hypothesis or data. We collected results from statistical tests that returned results with data even if the tests were not applicable and compared these to the tests that Tea recommended. We found two tutorials where following the results of any statistical test that executes, as novices commonly do, would lead to different conclusions based on statistical significance.

As our first attempt to support hypothesis formalization, Tea demonstrates that raising the level of abstraction to better match analysts' knowledge can lead to statistical analyses that assess analysts' hypotheses, apply to the data, and avoid mistakes novices and experts make. A limitation of Tea is that it provides limited support for conceptual modeling, or reasoning about how variables relate to one another conceptually beyond their statistical properties. We address this limitation and tackle a larger, more difficult class of analyses in Tisane.

3 TISANE: AUTHORING STATISTICAL MODELS VIA FORMAL REASONING FROM CONCEPTUAL AND DATA RELATIONSHIPS

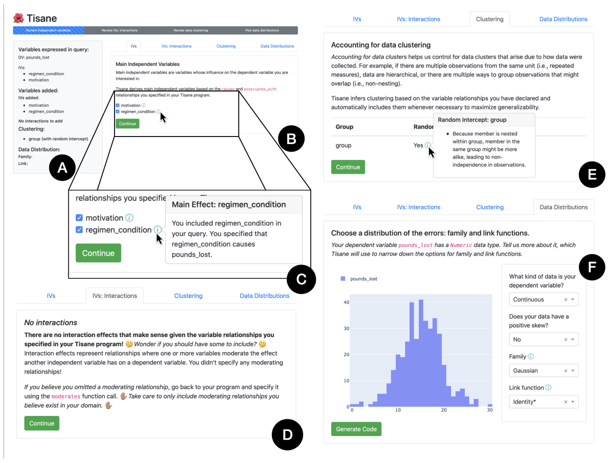
```
import tisane as ts

# Variable declarations
adult = ts.Unit("member")
motivation = adult.numeric("motivation")
pounds_lost = adult.numeric("pounds_lost")
group = ts.Unit("group")
regimen_condition = group.nominal("regimen_condition") #
    control vs. treatment
# Variable relationships
regimen_condition.causes(pounds_lost)
motivation.associates_with(pounds_lost)
adult.nests_within(group)
# 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)


# Specify data
data_path = "group_exercise.csv"
```

Listing 2: Example Tisane program. Tisane programs prioritize expressing conceptual and data measurement relationships.

As we found in our empirical research that led to our theory of hypothesis formalization [5], there is a mismatch between existing statistical interfaces and domain experts' analysis concerns. Domain experts are well-versed in how variables are related conceptually in their discipline and are aware of how data were collected, but incorporating this knowledge to formulate statistical models involves answering questions such as the following: Which variables do we include in our statistical model to account for potential confounding? Did the data collection procedure (e.g., due to repeated measures) introduce dependencies between observations that need to be explicitly modeled? How to account for this?



Variables expressed in query:
DV: pounds_lost
IVs:
- regimen_condition
- motivation

Variables added:
IVs added:
- motivation
- regimen_condition

No interactions to add

Clustering:
- group (with random intercept)

Data Distribution:
Family:
Link:

Main Independent Variables

Main independent variables are variables whose influence on the dependent variable you are interested in.

Tisane derives main independent variables based on the causes and associates_with relationships you specified in your Tisane program.

☑ motivation
☑ regimen_condition

Main Effect: regimen_condition

You included regimen_condition in your query. You specified that regimen_condition causes pounds_lost.

No interactions

There are no interaction effects that make sense given the variable relationships you specified in your Tisane program! 🤔 *Wonder if you should have some to include?* 🤔 Interaction effects represent relationships where one or more variables moderate the effect another independent variable has on a dependent variable. You didn't specify any moderating relationships!

If you believe you omitted a moderating relationship, go back to your program and specify it using the moderates function call. ✋ *Take care to only include moderating relationships you believe exist in your domain.* ✋

Accounting for data clustering

Accounting for data clusters helps us control for data clusters that arise due to how data were collected. For example, if there are multiple observations from the same unit (i.e., repeated measures), data are hierarchical, or there are multiple ways to group observations that might overlap (i.e., non-nesting).

Tisane infers clustering based on the variable relationships you have declared and automatically includes them whenever necessary to maximize generalizability.

Group	Random
group	Yes

Random Intercept: group

- Because member is nested within group, member in the same group might be more alike, leading to non-independence in observations.

Choose a distribution of the errors: family and link functions.

Your dependent variable pounds_lost has a Numeric data type. Tell us more about it, which Tisane will use to narrow down the options for family and link functions.

What kind of data is your dependent variable? Continuous

Does your data have a positive skew? No

Family: Gaussian

Link function: Identity*

Figure 2: Example Tisane GUI for disambiguation. Tisane asks analysts disambiguating questions about variables that are conceptually relevant and that analysts may have overlooked in their query.

Tisane [7] helps answer these and additional questions that are necessary for authoring generalized linear models with or without mixed effects. Tisane focuses on generalized linear models because they are prevalent in domains ranging from psychology to medicine to engineering and are difficult for even statistical experts to author. Tisane is implemented as a Python package[3].

Tisane provides a study design specification language so that analysts can express how variables conceptually relate to one another and how the data were collected (e.g., repeated measures, hierarchical nesting). Based on these expressions, Tisane constructs an internal graph representation. This graph is useful for deriving a space of candidate statistical models using the modified disjunctive criteria [16] from the causal modeling literature. Because there may be multiple potential statistical models, Tisane engages analysts in a disambiguation process where the system explains its rationale for specific model structures and asks targeted questions for formulate a final output model. The disambiguation process occurs in a GUI and involves explanations designed to be approachable for statistical non-experts. At the very end, Tisane generates a script for fitting and visualizing the statistical model.

To evaluate the feasibility of Tisane's programming and interaction model, we conducted six case studies with researchers. Researchers reported becoming more aware of their domain assumptions and able to focus on their research questions and analysis goals. The impact of Tisane on real-world analyses was most clear in two examples. First, an HCI researcher used Tisane and was able to catch a bug in their statistical models prior to submitting (and ultimately publishing) their paper at the ACM CHI conference. They had previously used linear mixed-effects models, but using Tisane, they learned that such a model would be inappropriate given that the dependent variable of interest was skewed and contained only positive values. Second, an external collaborator used Tisane to verify a statistical model they used for a health policy paper we co-authored together.

These initial results suggest that Tisane's novel programming and interaction model are viable. They have the potential to focus analysts on their domain and data knowledge while offloading the statistical model generation process to a system, and in doing so, can even help analysts avoid modeling mistakes. In ongoing work (see ??), we plan to more rigorously assess the impact of Tisane's programming and interaction model on analysis processes and authored statistical models. These evaluation plans motivate our work on rTisane.

4 RTISANE: REFINING SEMANTICS FOR CONCEPTUAL MODELING

Our ongoing research goals are to (i) identify what semantics more closely match how statistical non-experts reason about conceptual relationships between variables, (ii) implement these semantics in rTisane, and (iii) evaluate the impact of rTisane in a controlled lab study.

We conducted a qualitative lab study with five domain experts to understand what analysts want to express about their implicit conceptual assumptions. We found that despite being statistical non-experts, analysts are careful about expressing conceptual relationships. For example, they distinguish between assumed and hypothesized relationships. Some analysts also describe how a variable specifically responds to another ("...if A causes B, then by changing A, I can change B whereas `associates_ with` means that...if I can turn dial A, B might not change."). In contrary to a widely accepted belief that raising the level of abstraction in programming domains is desirable for non-experts, we found that more granular language constructs were easier understand and use precisely.

The primary result of the study has been that we have added new constructs that differentiate assumed relationships from hypothesized ones, provided more constructs for specifying conceptual relationships with increasingly levels of specificity (e.g., `causes(a, b)`, `whenThen(when=increases(a), then=increases(b))`) and replaced confusing constructs, such as `moderates`, with more specific ways to express how multiple variables relate to each another.

We have begun implementing rTisane[4], we have observed that a potential benefit of the updated semantics is not just that they are easier to understand and use but, more importantly, that they prompt reflection and increase analysts' awareness of conceptual assumptions. In this way, we hypothesize that the reflection rTisane promotes will lead to more awareness of assumptions throughout the analysis process and statistical models that better reflect analysts' knowledge and can answer their research questions.

Finally, I am planning to conduct a within-subjects user study comparing rTisane to a scaffolded workflow to test the following hypotheses:

- rTisane helps analysts become aware of their implicit assumptions.
- Analysts will approach statistical modeling differently with rTisane than without it.

[3] Tisane is open-source: https://github.com/emjun/tisane

[4] rTisane is implemented as an open-source R library: https://github.com/emjun/rTisane

- Analysts will make fewer analysis mistakes with rTisane than without it.

Throughout this dissertation, we have evaluated the feasibility of and assessed how Tea and Tisane can improve analyses. However, we have yet to probe into why and precisely how these systems impact data analysis practice. In ongoing work, I will evaluate the impact of rTisane in a controlled lab study. rTisane is an appropriate choice because it epitomizes this dissertation's testable insights: (i) analysts grapple with conceptual and data concerns while authoring statistical models, (ii) abstractions that allow analysts to express this knowledge can help them become more aware of implicit assumptions and (iii) statistical software can/should leverage this knowledge to generate statistical models that are better than those that analysts can come up with on their own.

5 CONCLUSION

This dissertation first presents *hypothesis formalization*, a new theory of how analysts must align their domain knowledge, data observations, and statistical methods to translate their research questions and hypotheses into statistical models. Then, based on this insight, we developed three new systems: Tea, Tisane, and rTisane. They provide novel programming and interaction models. Rather than specify the statistical models to fit—which analysts may not know—analysts express their implicit assumptions about their data and domain knowledge. Based on this higher-level knowledge, the systems suggest valid statistical models. Ongoing work evaluates if this new programming and interaction model increases awareness of implicit assumptions, changes the statistical analysis authoring process, and improves the statistical models authored. Future work can build on this work to integrate data analysis with experimental design, improve data science education, and support collaborative data analysis.

ACKNOWLEDGMENTS

Many thanks to my advisors Jeffrey Heer and René Just for their continual feedback and support. Thank you to all my collaborators and co-authors who made the projects in this dissertation possible. Thank you to all my external collaborators who have inspired and supported this work.

REFERENCES

[1] Bradley Efron. 1992. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*. Springer, 569–593.

[2] Ronald Aylmer Fisher. 1937. *The design of experiments*. Oliver And Boyd; Edinburgh; London.

[3] SAS Institute Inc. 2021. SAS. https://www.sas.com/

[4] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–2021. SciPy: Open source scientific tools for Python. http://www.scipy.org/

[5] 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"

[6] 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.

[7] 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*. 1–16.

[8] Maurits Kaptein and Judy Robertson. 2012. Rethinking statistical analysis methods for CHI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1105–1114.

[9] Matthew Kay, Gregory L Nelson, and Eric B Hekler. 2016. Researcher-centered design of statistics: Why Bayesian statistics better fit the culture and incentives of HCI. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 4521–4532.

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

[11] Travis E Oliphant. 2006. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA.

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

[13] 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.

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

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

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

[17] Wikipedia contributors. 2019. JMP (statistical software) — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=JMP_(statistical_software)&oldid=887217350. [Online; accessed 5-April-2019].

[18] Wikipedia contributors. 2019. R (programming language) — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=R_(programming_language)&oldid=890657071. [Online; accessed 5-April-2019].

[19] Wikipedia contributors. 2019. SAS (software) — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=SAS_(software)&oldid=890451452. [Online; accessed 5-April-2019].

[20] Wikipedia contributors. 2019. SPSS — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=SPSS&oldid=888470477. [Online; accessed 5-April-2019].