The Journal of Open Source Software

CausalTables.jl: Simulating and storing data for statistical causal inference in Julia

Salvador V. Balkus [1¶] and **Nima S. Hejazi** [1]

[1] Department of Biostatistics, Harvard T.H. Chan School of Public Health ¶ Corresponding author

DOI: 10.21105/joss.07580

Software

- Review
- Repository
- Archive

Editor: Oskar Laverny

Reviewers:

- @juliohm
- @olivierlabayle

Submitted: 26 November 2024
Published: 14 February 2025

Summary

Estimating the strength of causal relationships between treatment and response variables is an important problem across many scientific disciplines. `CausalTables.jl` is a package that supports causal inference in Julia by providing two important functionalities. First, it implements the `CausalTable`, bundling tabular data with a type of directed acyclic graph (DAG) encoding features' causes. Users can intervene on treatments and identify causal-relevant variables like confounders automatically. Second, the package's `StructuralCausalModel` interface simplifies simulating data from arbitrary causal structures – and unlike other packages, users can extract ground truth distributions conditional on the data generated in previous steps. In this way, `CausalTables.jl` makes it easier to develop and experimentally evaluate new statistical causal inference methods in Julia.

Statement of need

The quantitative science of causal inference has emerged over the past three decades as a set of formalisms for studying cause-and-effect relationships between variables from observed data (Hernán & Robins, 2020; Pearl, 2009). Causal inference techniques have helped scientists and decision-makers better understand important phenomena in fields ranging from medicine to economics. New software tools for causal inference are being developed at a rapid pace, but in the Julia language, there currently do not exist auxiliary tools designed to support their development. `CausalTables.jl` aims to provide such a tool.

Implementing and testing causal inference methods in Julia involves two main challenges. First, causal estimation requires identifying and modifying features based on their relationships with treatment and response variables, which might include confounders, mediators, or instruments. Their required format may differ depending on downstream packages; for instance, `MLJ.jl` (Blaom et al., 2020) requires Table input, while `GLM.jl` (Bates et al., 2023) needs a Matrix or `Formula`. Second, when evaluating a causal estimator on simulated data from a Structural Causal Model (SCM) (Pearl, 2009), one often desires access to the true ("oracle") conditional distributions of relevant variables in the SCM, as well as ground truth values of various causal estimands, in order to test whether the method works correctly.

`CausalTables.jl` addresses both challenges – the first via the `CausalTable` interface, which extends `Tables.jl` (Quinn et al., 2024) with causal identification routines, and the second via the `StructuralCausalModel`, which encodes a causal model as a sequence of conditional distributions from `Distributions.jl` (Besançon et al., 2021; Lin et al., 2019), providing random sampling and ground-truth computation. `CausalTables.jl` integrates seamlessly with established Julia packages, ensuring ease of use for statisticians and applied scientists alike.

Comparison to existing packages

While R and Python include many causal packages (Chen et al., 2020; van der Laan et al., 2024), Julia has relatively fewer. Recent Julia packages for causal inference include `TMLE.jl` (Labayle et al., 2024) and `CausalELM.jl` (Colby, 2024). These focus on specific estimators, rather than general data processing and simulation like `CausalTables.jl`. The package `CausalInference.jl` (Schauer et al., 2024) implements causal graphs and discovery algorithms, similar to CausalDAG (Squires, 2018) or DoWhy (Sharma & Kiciman, 2020) in Python and daggity (Textor et al., 2017) in R. That said, it is generally incompatible with the tabular data used in practice and does not support simulations. The simulation capabilities of `CausalTables.jl` are similar to those of probabilistic programming packages like `Turing.jl` (Ge et al., 2018) or `Gen.jl` (Cusumano-Towner et al., 2019). However, while other packages can *sample* data from SCMs, only `CausalTables.jl` allows extracting *closed-form distributions* conditional on data drawn in previous steps of the process.

Example 1: Data with causal structure

`CausalTables.jl` supports causal inference problems that involve estimating the effect of at least one treatment on at least one response. Using the `CausalTable` constructor, one can wrap an existing Table with causal structure:

```julia
using CausalTables

# Example data in Tables-compatible format
tbl = (W = [0.2, 0.4, 0.7],
       A = [false, true, true],
       Y = [0.8, 1.2, 2.3])

# Wrap data as CausalTable
ct_wrap = CausalTable(tbl; treatment = :A, response = :Y,
                           causes = (A = [:W], Y = [:W, :A]))
```

Convenience functions perform causal data processing. For example, the general `parents` function selects only features that cause a given variable; other functions, like `confounders`, select variables with more specific causal relationships.

```julia
parents(ct_wrap, :Y)
```

```
CausalTable
┌─────────┬───────┐
│       W │     A │
│ Float64 │  Bool │
├─────────┼───────┤
│     0.2 │ false │
│     0.4 │  true │
│     0.7 │  true │
└─────────┴───────┘
Summaries: NamedTuple()
Arrays: NamedTuple()
```

Example 2: Simulation with ground truth

An SCM defines causal structure by envisaging a data-generating process as random draws from a sequence of non-parametric structural equations, with each draw depending on the draws preceding it. For example:

$$\begin{aligned} W &\sim Beta(2,4) \\ A &\sim Bernoulli(0.5W + 0.2) \\ Y &\sim Normal(A + W, 1) \end{aligned}$$

This SCM can be implemented in `CausalTables.jl` and randomly sampled by enumerating the sequence of random variables along with labels of their causal roles:

```
using Distributions

# Define sequence of random variables
dgp = @dgp(
    W ~ Beta(2, 4),
    A ~ Bernoulli.(0.5 .* W .+ 0.2),
    Y ~ Normal.(W .+ A, 1)
)

# Define structural causal model
scm = StructuralCausalModel(dgp;
  treatment = :A, response = :Y
)

ct = rand(scm, 5) # randomly sample
```

Many causal estimands involve applying some intervention to a treatment. For instance, computing an ATE compares hypothetical responses had everyone been treated versus no one treated; one can apply these interventions on a `CausalTable` using the `intervene` function:

```
treated = intervene(ct, treat_all)
untreated = intervene(ct, treat_none)
```

After simulating data, the true ("oracle") distribution can be obtained using `condensity`. Other functions obtain specific features, such as `conmean` for the conditional mean. These help evaluate how well a causal estimator might perform if the true distribution were known; for example, the code below computes the "true" ATE plug-in estimate:

```
mean(conmean(scm, treated, :Y) .- conmean(scm, untreated, :Y))
```

```
1.0
```

`CausalTables.jl` also provides high-level functions to approximate the ground truth of common causal estimands, such as:

- Average treatment effects (`ate`) including among the treatment (`att`) and untreated (`atu`)
- Counterfactual means (`cfmean`) and differences (`cfdiff`)
- Average policy effects (`ape`)

Closing remarks

The goal of `CausalTables.jl` is to simplify causal inference in Julia. So far, it has been used to experimentally evaluate novel causal estimators for continuous treatments on network data (Balkus et al., 2024), and also been integrated into `TMLE.jl` (Labayle et al., 2024). As interest in causal inference grows, `CausalTables.jl` aims to provide a user-friendly foundation for practitioners to develop and test new causal methods in the Julia ecosystem.

Acknowledgements

Salvador Balkus acknowledges support from the National Institute of Environmental Health Sciences (award no. T32 ES007142) and the National Science Foundation (award no. DGE 2140743).

References

Balkus, S. V., Delaney, S. W., & Hejazi, N. S. (2024). The causal effects of modified treatment policies under network interference. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2412.02105

Bates, D., Noack, A., Kornblith, S., Bouchet-Valat, M., Borregaard, M. K., Arslan, A., White, J. M., Kleinschmidt, D., Alday, P., Lynch, G., Dunning, I., Mogensen, P. K., Lendle, S., Dilum Aluthge, Mousum Dutta, Pdeffebach, José Bayoán Santiago Calderón, P., Ayush Patnaik, Born, B., … König, B. (2023). *JuliaStats/GLM.jl: v1.9.0*. Zenodo. https://doi.org/10.5281/zenodo.3376013

Besançon, M., Papamarkou, T., Anthoff, D., Arslan, A., Byrne, S., Lin, D., & Pearson, J. (2021). Distributions.jl: Definition and modeling of probability distributions in the JuliaStats ecosystem. *Journal of Statistical Software*, *98*(16), 1–30. https://doi.org/10.18637/jss.v098.i16

Blaom, A. D., Kiraly, F., Lienart, T., Simillides, Y., Arenas, D., & Vollmer, S. J. (2020). MLJ: A Julia package for composable machine learning. *Journal of Open Source Software*, *5*(55), 2704. https://doi.org/10.21105/joss.02704

Chen, H., Harinen, T., Lee, J.-Y., Yung, M., & Zhao, Z. (2020). CausalML: Python package for causal machine learning. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2002.11631

Colby, D. (2024). *CausalELM.jl*. https://github.com/dscolby/CausalELM.jl

Cusumano-Towner, M. F., Saad, F. A., Lew, A. K., & Mansinghka, V. K. (2019). Gen: A general-purpose probabilistic programming system with programmable inference. *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 221–236. https://doi.org/10.1145/3314221.3314642

Ge, H., Xu, K., & Ghahramani, Z. (2018). Turing: A language for flexible probabilistic inference. *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, 1682–1690. https://doi.org/10.17863/CAM.42246

Hernán, M. A., & Robins, J. M. (2020). *Causal Inference: What If*. CRC Boca Raton, FL. ISBN: 9781420076165

Labayle, O., Beentjes, S., Khamseh, A., & Ponting, C. (2024). *TMLE.jl*. https://github.com/olivierlabayle/TMLE.jl

Lin, D., White, J. M., Byrne, S., Bates, D., Noack, A., Pearson, J., Arslan, A., Squire, K., Anthoff, D., Papamarkou, T., Besançon, M., Drugowitsch, J., Schauer, M., & contributors, other. (2019). *JuliaStats/Distributions.jl: a Julia package for probability distributions and associated functions*. Zenodo. https://doi.org/10.5281/zenodo.2647458

Pearl, J. (2009). *Causality*. Cambridge University Press. ISBN: 9780521895606

Quinn, J., Kamiński, B., Anthoff, D., Bouchet-Valat, M., Papp, T. K., Arakaki, T., Schouten, R., Robinson, N., mathieu17g, Samuel, O., Revels, J., ExpandingMan, Hanson, E., Blaom, A., Arslan, A., Ling, J., Chen, J., Day, J., Calderón, J. B. S., … Adenbaum, J. (2024). *JuliaData/Tables.jl: v1.12.0* (Version v1.12.0). Zenodo. https://doi.org/10.5281/zenodo.12753139

Schauer, M., Keller, M., & Wienöbst, M. (2024). *CausalInference.jl*. Zenodo. https://doi.org/10.5281/zenodo.13684767

Sharma, A., & Kiciman, E. (2020). DoWhy: An end-to-end library for causal inference. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2011.04216

Squires, C. (2018). *causaldag: creation, manipulation, and learning of causal models*. https://github.com/uhlerlab/causaldag

Textor, J., Zander, B. van der, Gilthorpe, M. S., Liśkiewicz, M., & Ellison, G. T. H. (2017). Robust causal inference using directed acyclic graphs: The r package "dagitty." *International Journal of Epidemiology*, dyw341. https://doi.org/10.1093/ije/dyw341

van der Laan, M., Coyle, J., Hejazi, N., Malenica, I., Phillips, R., & Hubbard, A. (2024). *Targeted Learning in R*. https://tlverse.org/tlverse-handbook/