BayCauRETM: R package for Bayesian Causal Inference for Recurrent Event Outcomes

11 September 2025

Summary

Observational studies often estimate the effect of medical treatments on the rate of a recurrent event within a specific follow-up window. Recurrent events (e.g., repeated hospitalizations, relapses) may occur multiple times during follow-up. Causal analysis is challenging because: 1) Recurrent events are jointly observed with a terminal event (e.g., death), which truncates future recurrences. 2) Both event counts and the terminal process are unobserved under dropout/censoring. 3) Patients initiate treatment at different times, yielding as many strategies as initiation times. Finally, 4) Treatments are not randomized, so formal causal methods are required to adjust for observed confounders.

This paper presents `BayCauRETM`, an `R` package for estimating causal effects of different treatment initiation strategies on a recurrent event outcome in the presence of death and censoring. Users specify an initiation time and supply a `data.frame` containing confounders and columns for censoring, death, and interval-specific recurrent counts, then define terminal and recurrent models via standard `R` formula syntax. Given these inputs, `BayCauRETM` performs causal adjustment and outputs adjusted expected event rates over follow-up under the specified initiation time. The package also provides diagnostic and visualization utilities.

Intended users include statisticians, epidemiologists, and health-services researchers analyzing observational data.

Statement of need

Standard software for time-to-event and recurrent-event data remains useful descriptively but generally does not target causal estimands while addressing complexities (1)-(4) above (Ghosh and Lin 2002; Schaubel and Zhang 2010; Janvin et al. 2024). Oganisian et al. (2024) developed Bayesian statistical methods that accommodate these complexities and conducted a thorough simulation-based validation of these methods. However, due to the focus on methodological development and validation, only proof-of-concept replication code was provided along with the paper. There is need for user-friendly, off-the-shelf software with readable help files that can implement the methods developed in Oganisian et al. (2024).`BayCauRETM` fills this methodological and practical gap by operationalizing the Bayesian approach in `R`. Its syntax is familiar to base-R users and mirrors standard regression functions such as `lm()` and `glm`, with extensive help pages accessible via `help()`. Thus, `BayCauRETM` provides the first user-friendly software for analyzing complex recurrent-event data while handling complexities (1)-(4) described in the Summary section above.

Data structure, model, and outputs

In this section, we provide an overview of the expected input data structure, models that are run under-the-hood, and expected outputs. We refer readers to Oganisian et al. (2024) for methodological details.

Data structure and preprocessing

The package expects longitudinal data in long, person-interval format. For follow-up time τ, the window $[0, \tau)$ is partitioned into K equal-length intervals $I_k = [\tau_{k-1}, \tau_k)$ for $k = 1, \ldots, K$ with $\tau_0 = 0$ and $\tau_K = \tau$. Each row represents a patient-interval; a subject has one row per interval at risk.

Required `data.frame` variables are: subject ID, interval index k, treatment indicator (0 until the initiation interval, then 1), interval-specific count of recurrent events, and a terminal-event indicator (0 up to death, 1 thereafter). Optional variables include baseline covariates and lagged history (e.g., one-interval lag of the event count).

Model specification

Each row contains a monotone death indicator at the start of interval k, T_k, a monotone treatment indicator by the end of interval k, A_k, the interval count Y_k and baseline covariates $L \in \mathcal{L}$.

Let $a(s) = (\underbrace{0, \ldots, 0}_{s-1}, 1, \ldots, 1)$ be the strategy that initiates treatment at interval $s \in \{1, 2, \ldots, K+1\}$. Define $T_k^{a(s)}$ as the potential death indicator for the subject following sequence $a(a)$ up to k and $Y_k^{a(s)}$ as potential number of events in interval k. The target is the difference in average potential incidence rates over follow-up under two initiation times:

$$\Delta(s, s') = \mathbb{E}\left[\frac{\sum_{k=1}^{K} Y_k^{a(s)}}{K - \sum_{k=1}^{K} T_k^{a(s)}}\right] - \mathbb{E}\left[\frac{\sum_{k=1}^{K} Y_k^{a(s')}}{K - \sum_{k=1}^{K} T_k^{a(s')}}\right]$$

The package runs a pair of discrete-time models conditional on shared treatment and covariate terms:

1. Discrete-time hazard model for the terminal event that models death at a given interval conditional on survival up to that interval:

$$\lambda_k(a_k, \bar{y}_{k-1}, l) = \Pr\big(T_k = 1 \mid T_{k-1} = 0,\, a_k, \bar{y}_{k-1}, l\big)$$

2. Distribution for the number of event occurrences in a given interval conditional on survival through that interval:

$$f(y_k \mid a_k, \bar{y}_{k-1}, l) = \Pr\big(Y_k = y_k \mid T_k = 0,\, a_k, \bar{y}_{k-1}, l\big)$$

Here, we use overbar notation to denote process history, e.g. $\bar{Y}_{k-1} = (Y_1, Y_2, \ldots, Y_{k-1})$. $f(y_k \mid a_k, \bar{y}_{k-1}, \ell)$ represents the Poisson probability mass function with conditional mean/intensity of the event-count $\mu_k(a_k, \bar{y}_{k-1}, l) = E[Y_k \mid A_k, \bar{Y}_{k-1}, L]$. Together, these two models multiply to form a joint model for the terminal and recurrent event occurrence at a given interval.

The functions in `BayCauRETM` implement the following models for the hazard and intensity, respectively:

$$\begin{aligned} \operatorname{logit} \lambda_k(a_k, \bar{y}_{k-1}, l) &= \beta_{0k} + l^\top \beta_L + y_{k-1}\beta_Y + \beta_A a_k, \\ \log \mu_k(a_k, \bar{y}_{k-1}, l) &= \theta_{0k} + l^\top \theta_L + y_{k-1}\theta_Y + \theta_A a_k, \end{aligned}$$

The time-varying intercepts $\{\beta_{0k}\}$ and $\{\theta_{0k}\}$ parameterize the baseline hazard and event intensity, respectively. They are assigned a first-order autoregressive (AR1) smoothing prior. See Oganisian et al. (2024) for more details.

Posterior inference and g-computation

`BayCauRETM` conducts full posterior inference for the models and back-ends to Stan (Carpenter et al. 2017) via the `rstan` package since the posterior is not available in closed form. Stan is a probabilistic programming language (PPL) that implements cutting edge Hamiltonian Monte Carlo methods to obtain these draws.

Using a given draw of the model parameters obtained via Stan. For each parameter draw, `BayCauRETM` simulate the joint death-recurrent process under $a(s)$ and $a(s')$ to obtain a posterior draw of $\Delta(s, s')$. Reporting over many draws yields posterior samples of $\Delta(s, s')$, as described by Oganisian et al. (2024). The posterior mean and the 95% credible interval (2.5th and 97.5th percentiles) are reported.

Detailed usage and example results are available on GitHub (see the demo PDF).

Acknowledgements

This work was partially funded by the Patient Centered Outcomes Research Institute (PCORI) Contract ME-2023C1-31348.

References

Carpenter, Bob, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2017. "Stan: A Probabilistic Programming Language." *Journal of Statistical Software* 76 (1): 1–32. https://doi.org/10.18637/jss.v076.i01.

Ghosh, Debashis, and D. Y. Lin. 2002. "Marginal Regression Models for Recurrent and Terminal Events." *Statistica Sinica*, 663–88.

Janvin, Matias, Jessica G Young, Pål C Ryalen, and Mats J Stensrud. 2024. "Causal Inference with Recurrent and Competing Events." *Lifetime Data Analysis* 30 (1): 59–118. https://doi.org/10.1007/s10985-023-09594-8.

Oganisian, Arman, Anthony Girard, Jon A Steingrimsson, and Patience Moyo. 2024. "A Bayesian Framework for Causal Analysis of Recurrent Events with Timing Misalignment." *Biometrics* 80 (4): ujae145. https://doi.org/10.1093/biomtc/ujae145.

Schaubel, Douglas E, and Min Zhang. 2010. "Estimating Treatment Effects on the Marginal Recurrent Event Mean in the Presence of a Terminating Event." *Lifetime Data Analysis* 16 (4): 451–77. https://doi.org/10.1007/s10985-009-9149-x.