Bayesian Causal Inference for Recurrent Events with Timing Misalignment

Yuqin

2025-08-15

This Markdown of the Bayesian framework was proposed by Oganisian et al. (2024) for causal inference with recurrent events subject to timing misalignment. We use a semiparametric Bayesian model to estimate the average causal effect of a time-varying treatment on the recurrent event rate, accounting for terminal events and censoring.

1. Setup

We configure Stan and set parallelization options for efficient MCMC workflow. We fix the random seed, enable parallel sampling, and set Hamiltonian Monte Carlo controls. The defaults (warmup = 500, M (posterior draws) = 500, 4 chains) are modest for illustration.

```r
use_multicore <- TRUE   # FALSE for single core
if (use_multicore) {
  cores <- max(1, detectCores() - 2)
} else {
  cores <- 1
}
options(mc.cores = cores)
rstan_options(auto_write = TRUE)

warmup <- 500
M      <- 500
iter   <- warmup + M
B      <- 50
s_vec  <- c(3, 6, 9)
```

2. Data Preprocessing

This step prepares the longitudinal dataset for analysis, including normalization and handling of missing values.

The package expects a long panel with: unique subject id (id), discrete time index (k), treatment (Ak), recurrent count (Yk), terminal event indicator (Tk), and optional covariates (here L.1, L.2).

History lagYk (default 0 at each subject's first interval) captures short-term dependence. In our design, lagged history terms are fully flexible: users may supply their own lag variables (e.g., lagYk, higher-order lags, or custom functions of past outcomes) directly in the dataset and reference them in the formulas, or omit them entirely. If no lag variable is provided but lag terms are specified in the formulas, the function automatically constructs the appropriate lag internally. This guarantees valid defaults while allowing investigators to encode any history structure they need.

Continuous covariates are standardized to improve sampler geometry and prior interpretability.

Remove observations with missing values for variables used in the model formulas; if imputation is required, perform it upstream and re-use the same transform for prediction.

K equals the number of unique time intervals and controls the length of each model's piecewise baseline.

```r
load("data.Rdata")
df_fit <- df %>%
  filter(id %in% 1:100) %>%
  arrange(id, k) %>%
  mutate(k_fac = as.integer(factor(k, levels = sort(unique(k))))) %>%
  group_by(id) %>%
  mutate(
    lagYk = if ("lagYk" %in% names(.)) replace_na(lagYk, 0) else lag(Yk, default = 0)
  ) %>%
  ungroup() %>%
  drop_na(Tk, Yk, Ak, L.1, L.2) %>%
  mutate(
    L.1 = as.numeric(scale(L.1)),
    L.2 = as.numeric(scale(L.2))
  )
K <- length(unique(df_fit$k_fac))
```

3. Bayesian Model Fitting

We now fit the joint model for the recurrent events and terminal process using Stan.

We fit a joint Bayesian model for (i) the terminal process Tk and (ii) the recurrent evernts Yk, sharing the discrete time structure and history recoords. Each sub-model includes: the current treatment, lagged outcomes (e.g. I(lagYk^2)), standardized covariates (L.1, L.2), and a piecewise-constant baseline (time_baseline_T[j] / time_baseline_Y[j]).

We expose variable names in diagnostics and summaries as follows:

- beta_T:L.1, beta_T:L.2, beta_Y:L.1, beta_Y:L.2: covariate (L.1, L.2 here) effects in T/Y models.
- theta_T_lag:I(lagYk^2), theta_Y_lag:I(lagYk^2): coefficients for lagged outcome features.
- time_baseline_T[j], time_baseline_Y[j]: j-th time-interval baselines capturing secular time trends.
- treatment_effect_T, treatment_effect_Y: treatment effects for terminal and recurrent processes.

```r
# Stan fit
fit <- fit_causal_recur(
  data      = df_fit,
  K         = K,
  id_col    = "id",
  time_col  = "k_fac",
  treat_col = "Ak",
  lag_col   = "lagYk",
  formula_T = Tk ~ Ak + I(lagYk^2) + L.1 + L.2,
  formula_Y = Yk ~ Ak + I(lagYk^2) + L.1 + L.2,
  cores      = cores,
  verbose    = TRUE
)
```

```
## Loading pre-compiled Stan model from: D:/Program/R/R-4.4.3/library/BayCauRETM/stan/causal_recur_mode

## Sampling (4 chains * 2000 iter, cores=30)...
```

4. MCMC Diagnostics

Evaluate convergence and identify any problematic chains. We report R-hat (target 1.00), effective sample size (should be large), and show traceplots grouped by model block.

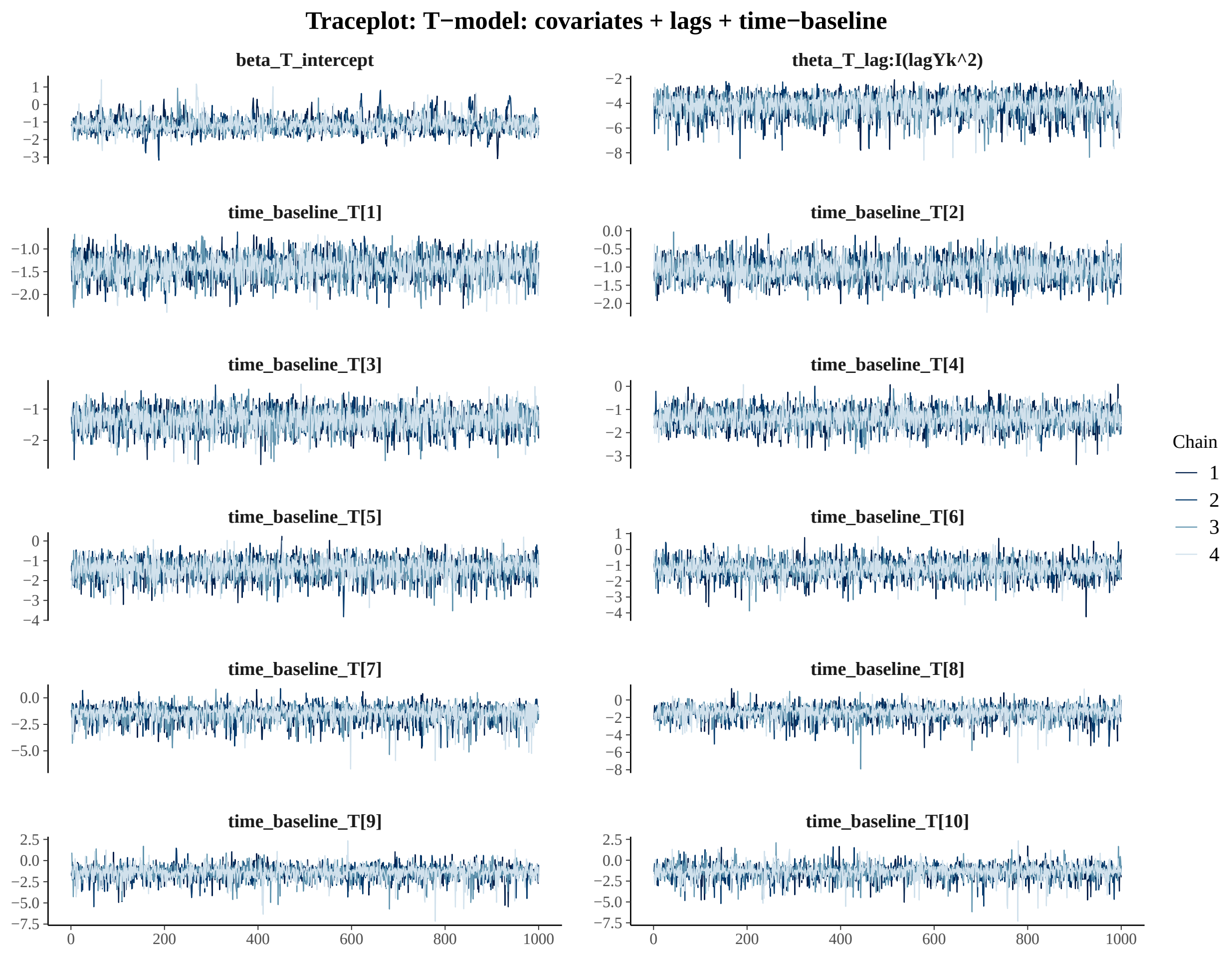
```
# MCMC Diagnosis
rstan::check_hmc_diagnostics(fit$stan_fit)
```

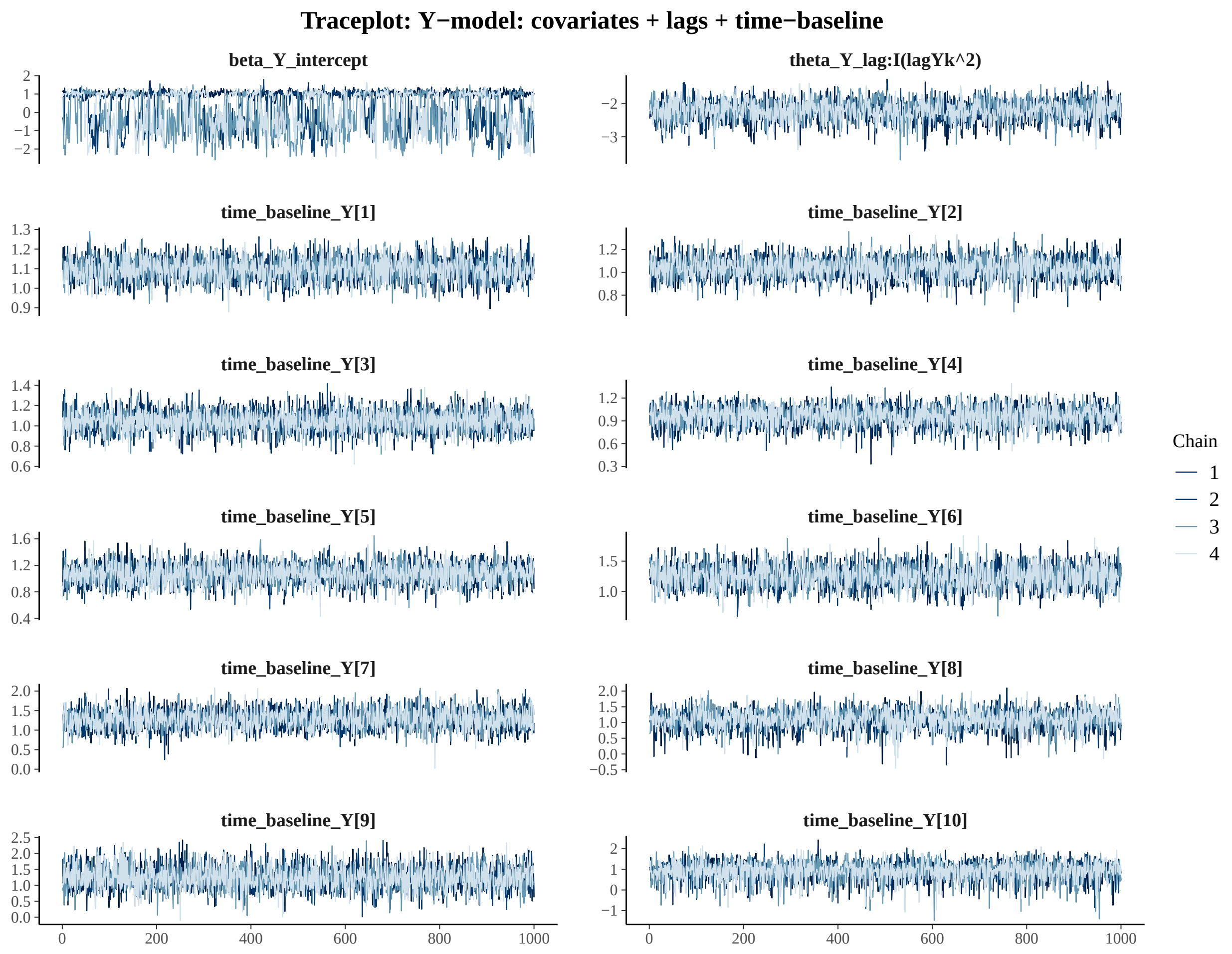
```
##
## Divergences:

## 20 of 4000 iterations ended with a divergence (0.5%).
## Try increasing 'adapt_delta' to remove the divergences.

##
## Tree depth:

## 0 of 4000 iterations saturated the maximum tree depth of 15.

##
## Energy:

## E-BFMI indicated no pathological behavior.
```

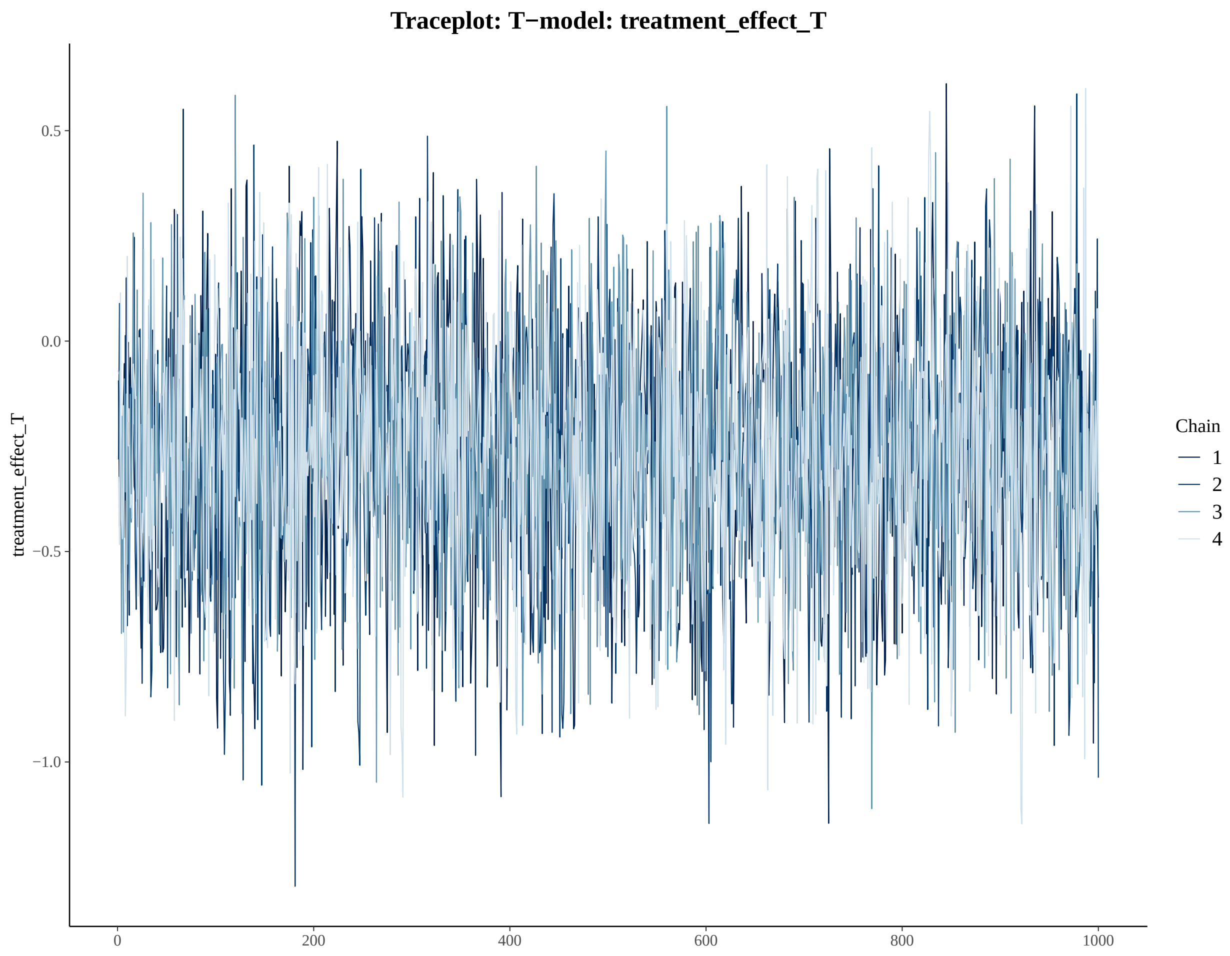
```
diag <- mcmc_diagnosis(fit)
```

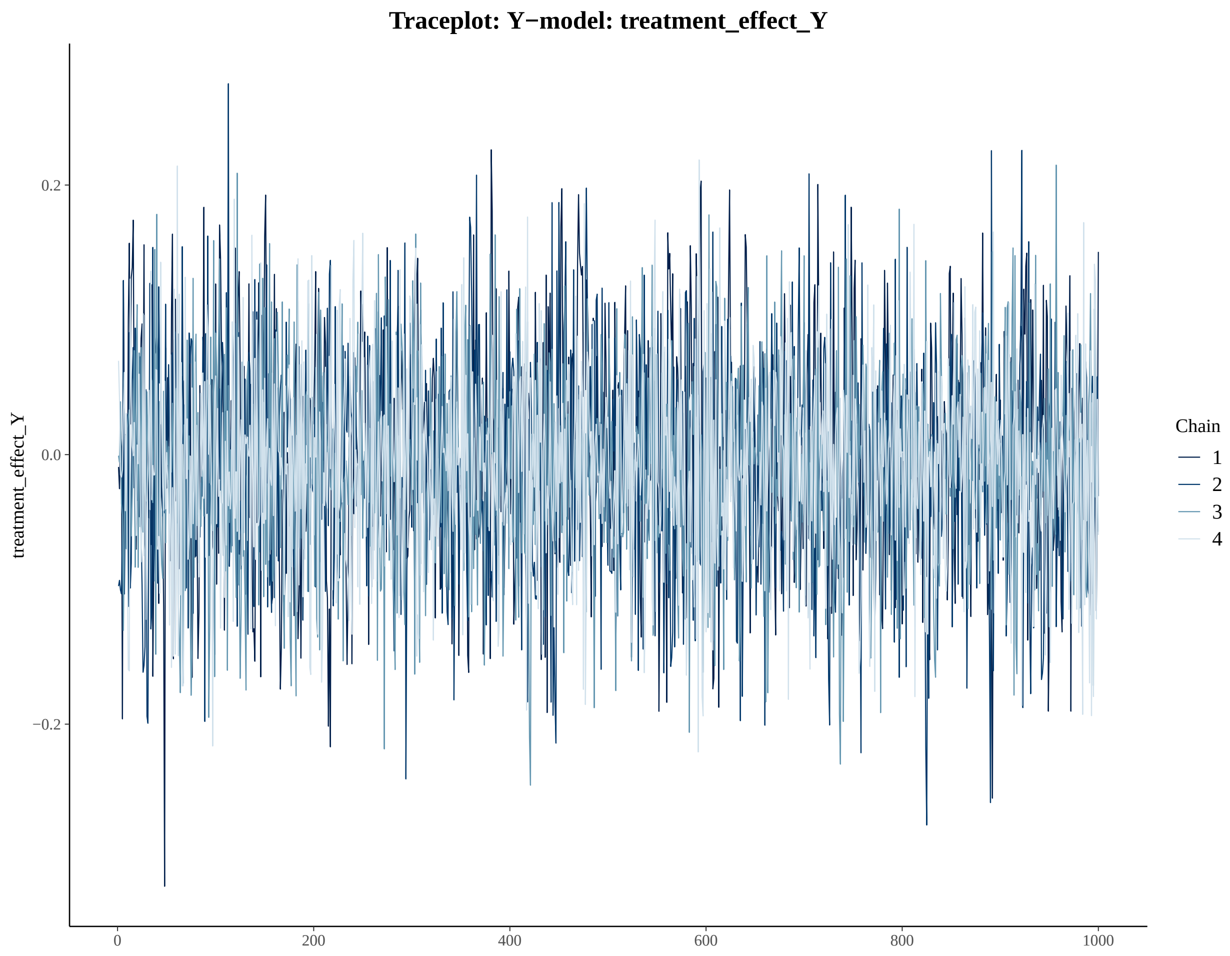
```
## ----- MCMC Rhat & Effective Sample Size -----
##                  Parameter     n_eff      Rhat
## 1  theta_T_lag:I(lagYk^2) 4043.2213 0.9996728
## 2      time_baseline_T[1] 3748.7310 0.9991667
## 3      time_baseline_T[2] 4128.5189 0.9997099
## 4      time_baseline_T[3] 4875.4662 0.9997434
## 5      time_baseline_T[4] 4218.5829 0.9993660
## 6      time_baseline_T[5] 4303.6104 0.9998946
## 7      time_baseline_T[6] 4068.7230 0.9999841
## 8      time_baseline_T[7] 4138.3207 0.9997244
## 9      time_baseline_T[8] 3583.9016 1.0000311
## 10     time_baseline_T[9] 3756.8256 0.9996320
## 11    time_baseline_T[10] 3803.8491 0.9994364
## 12 theta_Y_lag:I(lagYk^2) 1471.7730 1.0063451
## 13     time_baseline_Y[1] 3990.3140 0.9997153
## 14     time_baseline_Y[2] 3601.4790 1.0038586
## 15     time_baseline_Y[3] 4888.8775 0.9993624
## 16     time_baseline_Y[4] 2953.4266 1.0037303
## 17     time_baseline_Y[5] 3802.9640 1.0012266
## 18     time_baseline_Y[6] 3178.7094 1.0019557
## 19     time_baseline_Y[7] 3655.0890 1.0011517
## 20     time_baseline_Y[8]  832.0401 1.0122300
## 21     time_baseline_Y[9] 4496.1384 1.0008715
## 22    time_baseline_Y[10] 4670.2911 1.0011264
## 23     treatment_effect_T 4099.7557 0.9995201
## 24     treatment_effect_Y 3750.7110 1.0015077
## 25     theta_lag_extra[1] 5507.9220 1.0012521
## (Values close to Rhat = 1 and large n_eff indicate good convergence.)
```

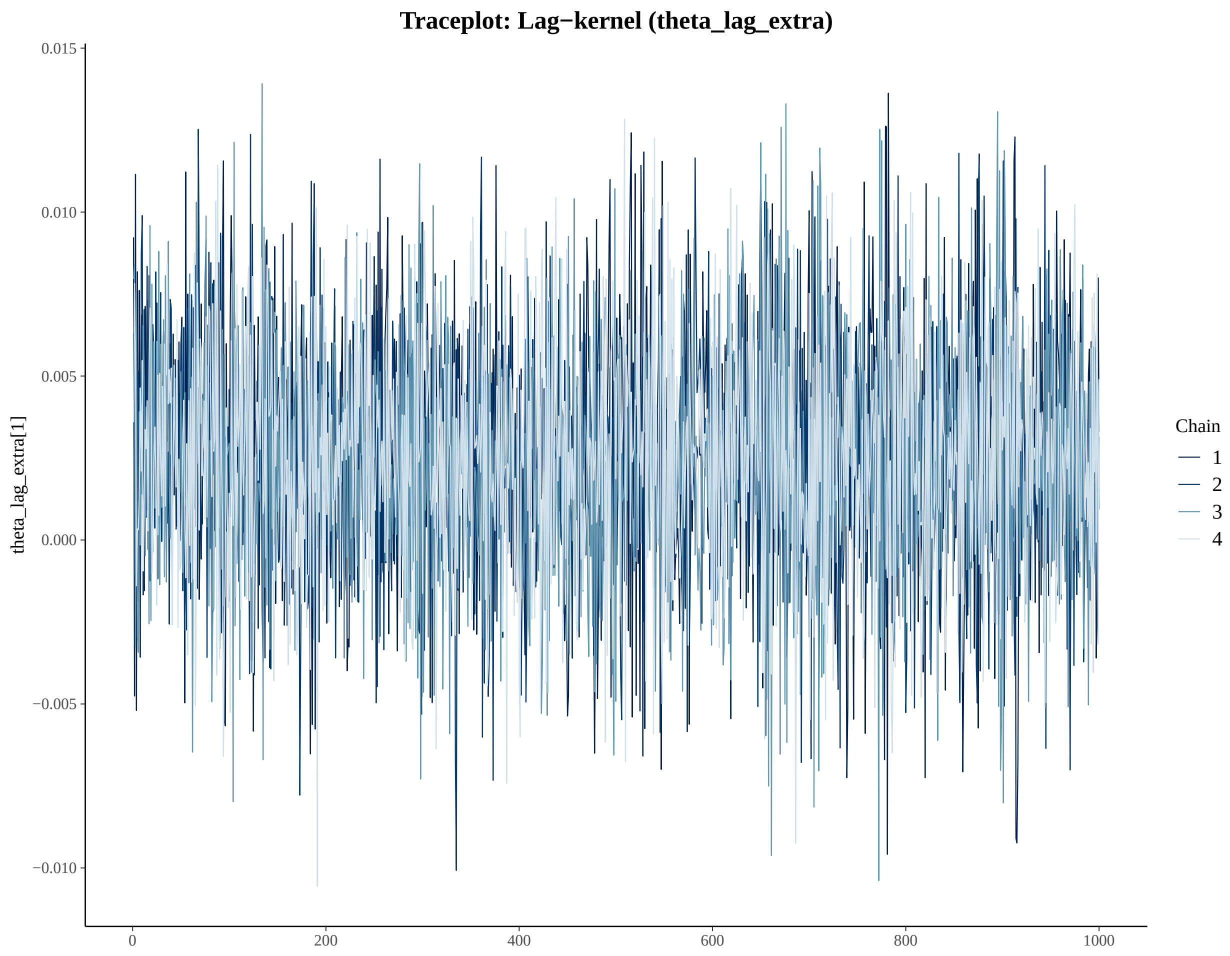
```
plot(diag)
```




Traceplot: Y−model: covariates + lags + time−baseline
beta_Y_intercept
theta_Y_lag:I(lagYk^2)
time_baseline_Y[1]
time_baseline_Y[2]
time_baseline_Y[3]
time_baseline_Y[4]
time_baseline_Y[5]
time_baseline_Y[6]
time_baseline_Y[7]
time_baseline_Y[8]
time_baseline_Y[9]
time_baseline_Y[10]
Chain
1
2
3
4
0
200
400
600
800
1000


Traceplot: T−model: treatment_effect_T
treatment_effect_T
0.5
0.0
−0.5
−1.0
0
200
400
600
800
1000
Chain
1
2
3
4


Traceplot: Y−model: treatment_effect_Y
treatment_effect_Y
0.2
0.0
−0.2
0
200
400
600
800
1000
Chain
1
2
3
4



5. G-Computation

We simulate potential outcomes under hypothetical treatment strategies to estimate causal contrasts.

We target a marginal causal estimand that contrasts the expected cumulative outcome trajectory under different treatment initiation strategies. Formally, let

$$\Delta(s, K+1) \;=\; \mathbb{E}\{Y_{1:K+1}(s)\} \;-\; \mathbb{E}\{Y_{1:K+1}(\text{ref})\}\,,$$

where $s \in s_{\text{vec}}$ denotes a treatment-start interval strategy and "ref" denotes the reference strategy (e.g., no intervention). Here $Y_{1:K+1}(s)$ is the cumulative number of events through interval $K+1$ under strategy s.

Estimation proceeds via Bayesian g-computation. For each posterior draw of model parameters, we simulate forward the joint process of survival and recurrent events under the intervention sequence $A_{1:K}(s)$, average across subjects, and repeat for the reference strategy. The causal contrast $\Delta(s, K+1)$ is then obtained as the posterior distribution of their difference:

$$\Delta(s, K+1)^{(m)} \;=\; \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{\sum_{k=1}^{K} Y_{ik}^{(m)}(s)}{K-\sum_{k=1}^{K} T_{ik}^{(m)}(s)} - \frac{\sum_{k=1}^{K} Y_{ik}^{(m)}(\text{ref})}{K-\sum_{k=1}^{K} T_{ik}^{(m)}(\text{ref})}\right\},$$

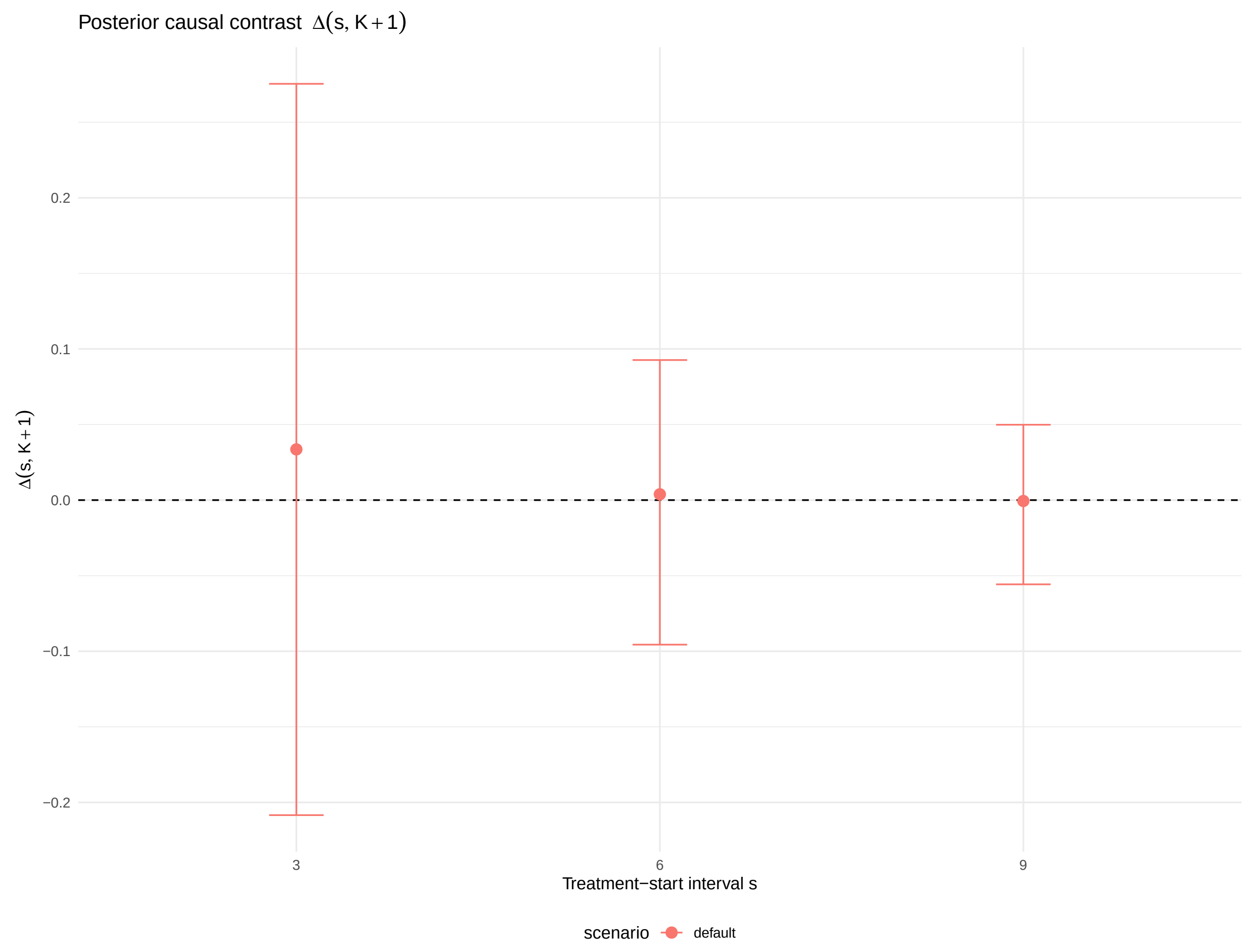
where $Y_{ik}^{(m)}(s)$ and $T_{ik}^{(m)}(s)$ denote simulated event and death indicators for subject i at interval k under posterior draw m. The full posterior of $\Delta(s, K+1)$ is summarized by its mean and credible interval. Wide intervals centered near zero indicate weak evidence for a treatment effect, in which case one may consider larger samples, stronger priors, or alternative summaries such as median effects or exceedance probabilities.

This simulation-based g-computation ensures that both survival and recurrent event processes are attributed consistently to each strategy, thereby yielding a valid causal interpretation of treatment-timing contrasts.

```r
baseline_df <- fit$data_preprocessed %>%
  group_by(pat_id) %>%
  slice_min(order_by = k_idx, n = 1) %>%
  arrange(pat_id) %>%
  ungroup()

gcomp <- g_computation(
  fit_out = fit,
  s_vec   = s_vec,
  B       = B,
  cores   = cores
)

print(gcomp)
```

```
## Causal contrast delta(s, K+1) summary:
##   s         Mean       X2.5.     X97.5.
##   3  0.033559993 -0.20843574 0.27544139
##   6  0.003859415 -0.09563426 0.09267411
##   9 -0.000597300 -0.05573113 0.04986188
```

```r
plot(gcomp, ref_line = 0)
```



```r
plot(gcomp, interactive = TRUE, ref_line = 0)
```

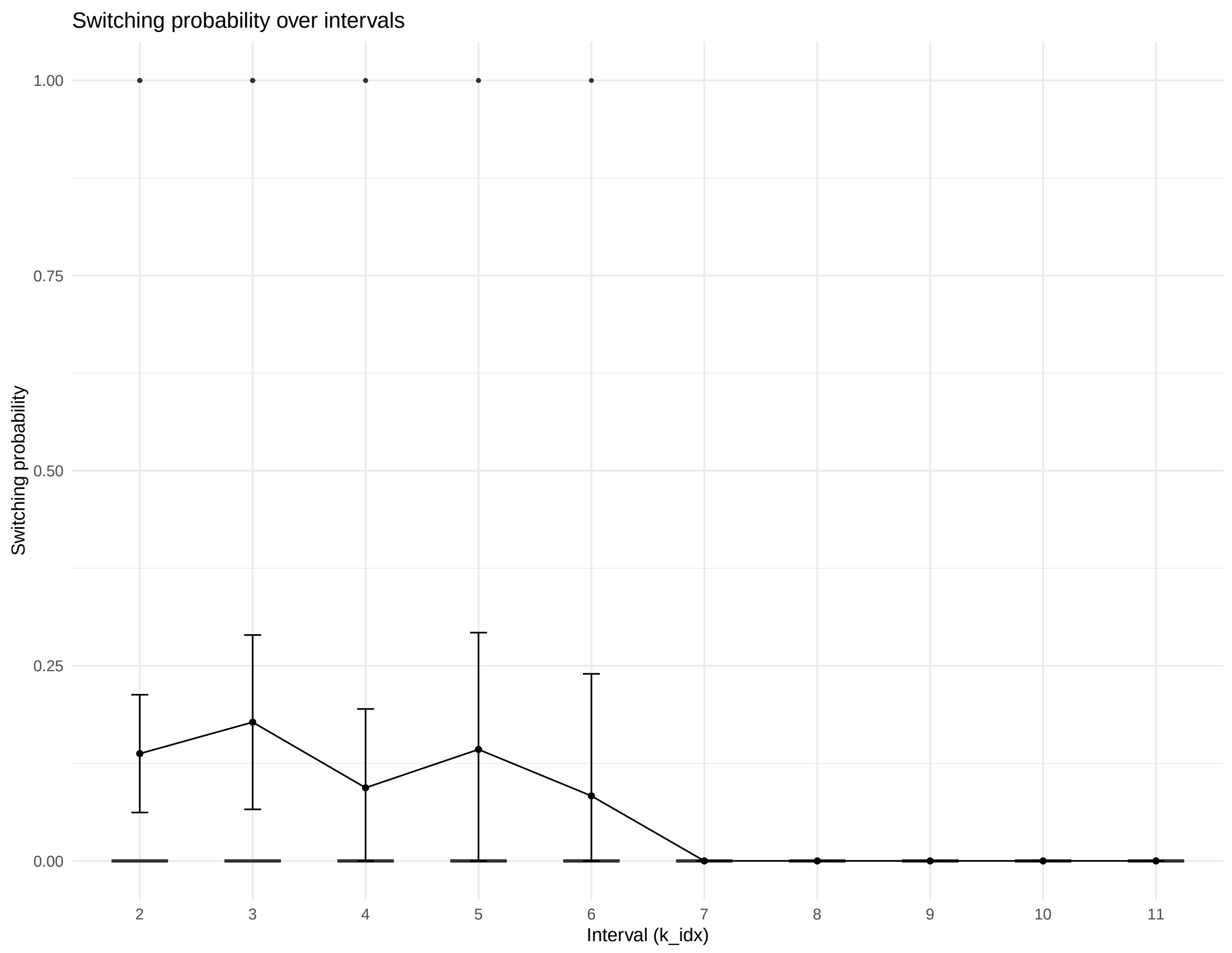
6. Switching Probability Summary

The switching probability summary quantifies how often subjects switch treatment across follow-up intervals. For each interval k, we estimate the within-subject probability of switching.

Boxplots show the empirical distribution of switching probabilities across subjects at each interval. Wide boxes indicate heterogeneity in switching behavior. Dots represent individual subjects whose switching probabilities are extreme relative to the median. Solid line connects the interval-specific means, summarizing the average probability of switching across the cohort. Horizontal bars at each point show approximate credible intervals (uncertainty bands) around the mean estimate.

Large dispersion or systematic drifts in switching probabilities across intervals kkk reveal time-structured treatment policies (e.g., delayed initiation or systematic stopping). This contextualizes causal estimates and aids interpretation of the contrast $\Delta(s, K{+}1)$.

```r
sw_diag <- switching_probability_summary(fit$data_preprocessed)
plot(sw_diag, type = "boxplot")
```



7. Summary of Causal Estimates

This table summarizes posterior distributions of model parameters and g-computation estimates. Parameter tables report Mean, 2.5%, 97.5%, R-hat, n_eff, MCSE, and CI width. Effects close to 0 with tight intervals imply little evidence of impact. Baseline blocks (time_baseline_*) quantify secular time risk; do not interpret them as treatment effects.

```
sum_tbl <- result_summary_table(
  fit_out   = fit,
  gcomp_out = gcomp,
  s_vec     = s_vec,
  format    = "kable"
)
print(sum_tbl)
```

```
## ----- Posterior Parameters -----
##
##
## |Parameter                |       Mean|      X2.5.|     X97.5.|      Rhat|      n_eff|      MCSE| CI_width|
## |:------------------------|----------:|----------:|----------:|---------:|----------:|--------
:|--------:|
## |time_baseline_Y[7]     |  1.272860|  0.811835|  1.760846| 1.001152| 3655.0890| 0.004064| 0.949011|
## |time_baseline_Y[6]     |  1.243500|  0.901395|  1.612853| 1.001956| 3178.7094| 0.003291| 0.711458|
## |time_baseline_Y[9]     |  1.238499|  0.575628|  1.983811| 1.000872| 4496.1384| 0.005271| 1.408183|
## |time_baseline_Y[1]     |  1.097261|  0.981489|  1.209320| 0.999715| 3990.3140| 0.000919| 0.227831|
## |time_baseline_Y[8]     |  1.086027|  0.428340|  1.649305| 1.012230|  832.0401| 0.010501| 1.220965|
## |time_baseline_Y[5]     |  1.071126|  0.769642|  1.362145| 1.001227| 3802.9640| 0.002475| 0.592503|
## |time_baseline_Y[3]     |  1.047825|  0.844407|  1.242295| 0.999362| 4888.8775| 0.001468| 0.397888|
## |time_baseline_Y[2]     |  1.038567|  0.852173|  1.211018| 1.003859| 3601.4790| 0.001519| 0.358845|
## |time_baseline_Y[4]     |  0.958851|  0.693154|  1.202924| 1.003730| 2953.4266| 0.002398| 0.509769|
## |time_baseline_Y[10]    |  0.904039| -0.147061|  1.666172| 1.001126| 4670.2911| 0.006516| 1.813233|
## |treatment_effect_Y     | -0.007025| -0.159476|  0.142381| 1.001508| 3750.7110| 0.001251| 0.301857|
## |treatment_effect_T     | -0.268343| -0.832832|  0.283079| 0.999520| 4099.7557| 0.004439| 1.115911|
## |time_baseline_T[2]     | -1.085123| -1.650292| -0.492085| 0.999710| 4128.5189| 0.004615| 1.158207|
## |time_baseline_T[6]     | -1.228294| -2.389829| -0.182637| 0.999984| 4068.7230| 0.008422| 2.207192|
## |time_baseline_T[3]     | -1.340576| -2.059847| -0.696947| 0.999743| 4875.4662| 0.005001| 1.362900|
## |time_baseline_T[4]     | -1.371349| -2.224765| -0.576268| 0.999366| 4218.5829| 0.006356| 1.648497|
## |time_baseline_T[10]    | -1.379758| -3.266730|  0.114876| 0.999436| 3803.8491| 0.013388| 3.381606|
## |time_baseline_T[5]     | -1.393497| -2.469617| -0.515554| 0.999895| 4303.6104| 0.007363| 1.954062|
## |time_baseline_T[1]     | -1.403884| -1.945598| -0.912718| 0.999167| 3748.7310| 0.004248| 1.032880|
## |time_baseline_T[9]     | -1.414690| -3.302739|  0.010557| 0.999632| 3756.8256| 0.012975| 3.313296|
## |time_baseline_T[8]     | -1.468155| -3.288208| -0.159727| 1.000031| 3583.9016| 0.013100| 3.128481|
## |time_baseline_T[7]     | -1.514010| -3.389412| -0.336074| 0.999724| 4138.3207| 0.011709| 3.053339|
## |theta_Y_lag:I(lagYk^2) | -2.212505| -2.846122| -1.642556| 1.006345| 1471.7730| 0.008031| 1.203566|
## |theta_T_lag:I(lagYk^2) | -4.163495| -6.156514| -2.730886| 0.999673| 4043.2213| 0.013650| 3.425628|
##
## ----- g-computation delta(s, K+1) -----
##
##
## |  s|      Mean|     X2.5.|   X97.5.| CI_width|
## |--:|---------:|---------:|--------:|--------:|
## |  3|  0.033560| -0.208436| 0.275441| 0.483877|
## |  6|  0.003859| -0.095634| 0.092674| 0.188308|
## |  9| -0.000597| -0.055731| 0.049862| 0.105593|
```