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 sampling.

```
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.

```
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.

```
# 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_model
```

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

4. MCMC Diagnostics

Evaluate convergence and identify any problematic chains.

```
# MCMC Diagnosis
message("Checking convergence...")
```

```
## Checking convergence...
```

```
rstan::check_hmc_diagnostics(fit$stan_fit)
```

```
##
## Divergences:
```

```
## 64 of 4000 iterations ended with a divergence (1.6%).
## Try increasing 'adapt_delta' to remove the divergences.
```

```
##
## Tree depth:
```

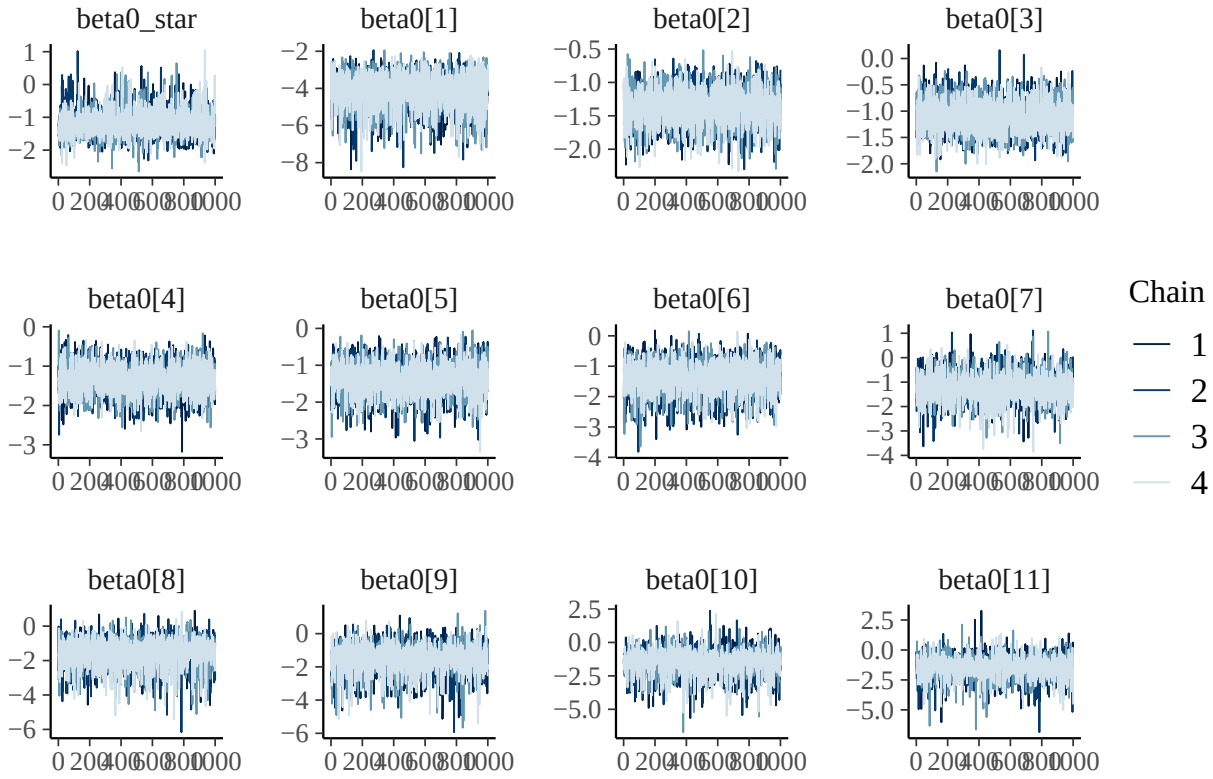
```
## 0 of 4000 iterations saturated the maximum tree depth of 15.

##
## Energy:

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

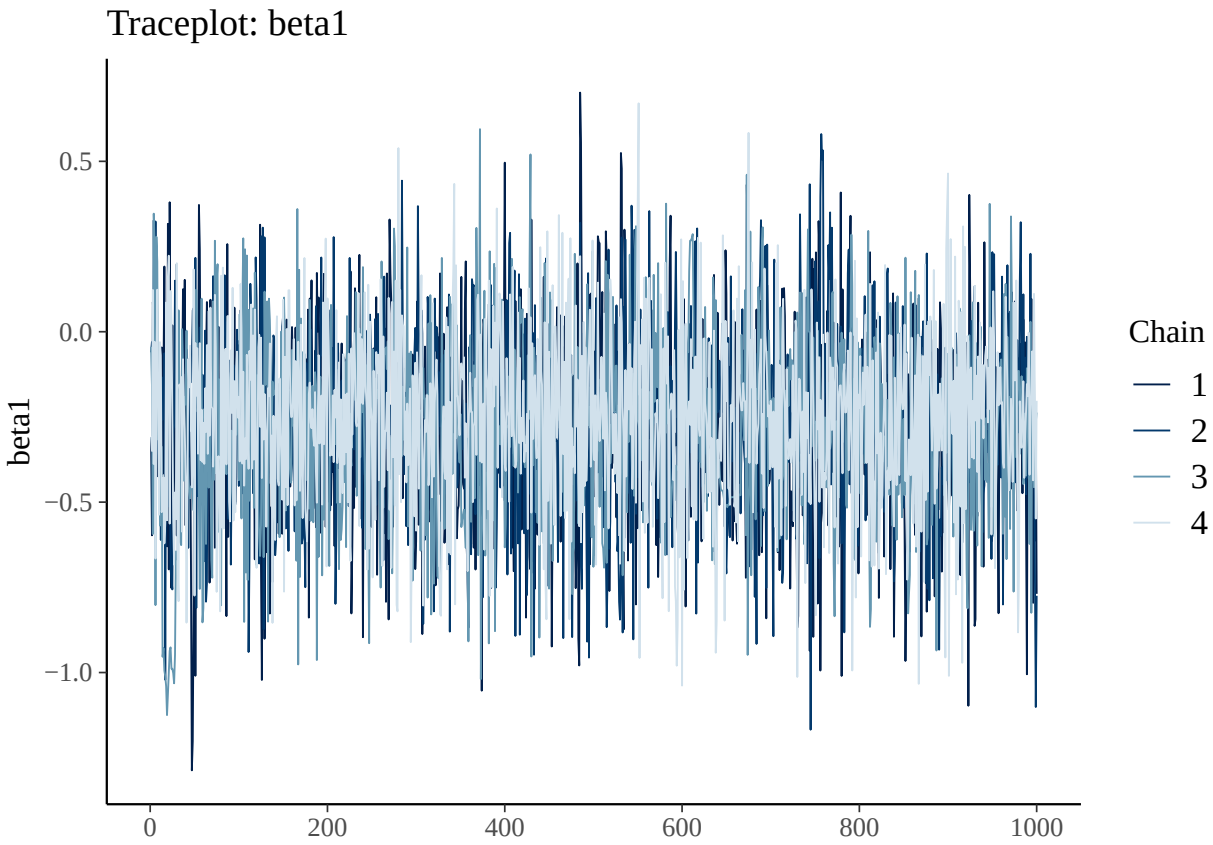
```
diag <- mcmc_diagnosis(fit, pars_to_check = c("beta0","beta1","theta0","theta1","theta_lag"))
```

```
## ----- MCMC Rhat & Effective Sample Size -----
##     Parameter    n_eff      Rhat
## 1    beta0[1] 3643.418 0.9998607
## 2    beta0[2] 4171.389 0.9996020
## 3    beta0[3] 3020.079 1.0012541
## 4    beta0[4] 4689.237 0.9994648
## 5    beta0[5] 4480.888 1.0004749
## 6    beta0[6] 3288.313 0.9998238
## 7    beta0[7] 4414.037 1.0009878
## 8    beta0[8] 3431.080 1.0013344
## 9    beta0[9] 3096.053 0.9996591
## 10  beta0[10] 2855.590 1.0011239
## 11  beta0[11] 3152.262 0.9997751
## 12      beta1 2655.593 1.0019275
## 13  theta0[1] 2212.840 1.0057596
## 14  theta0[2] 3920.799 1.0006062
## 15  theta0[3] 3140.715 1.0049544
## 16  theta0[4] 4333.342 0.9996669
## 17  theta0[5] 2903.138 1.0056761
## 18  theta0[6] 1723.776 1.0026391
## 19  theta0[7] 2310.635 1.0027139
## 20  theta0[8] 2264.325 1.0015809
## 21  theta0[9] 1480.434 1.0057404
## 22 theta0[10] 4139.792 1.0001993
## 23 theta0[11] 3100.707 1.0001999
## 24     theta1 4520.103 1.0010792
## (Values close to Rhat = 1 and large n_eff indicate good convergence.)
```

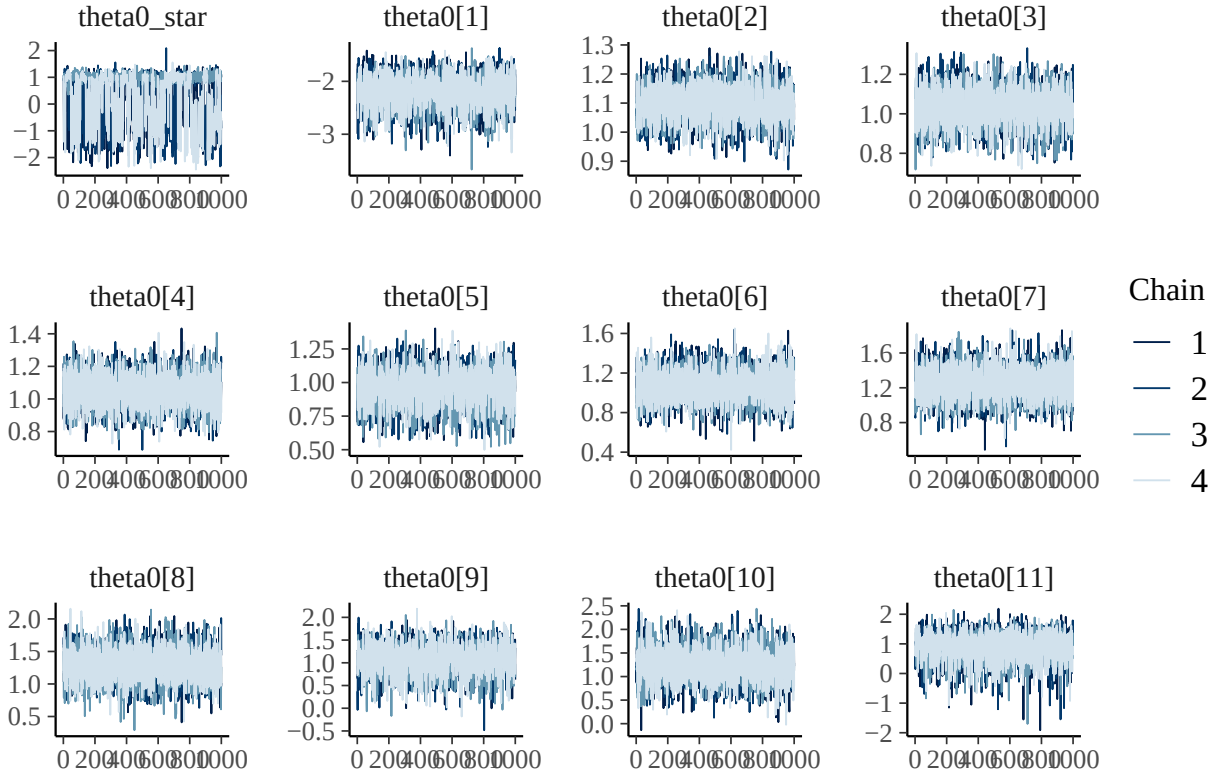
```
plot(diag)
```

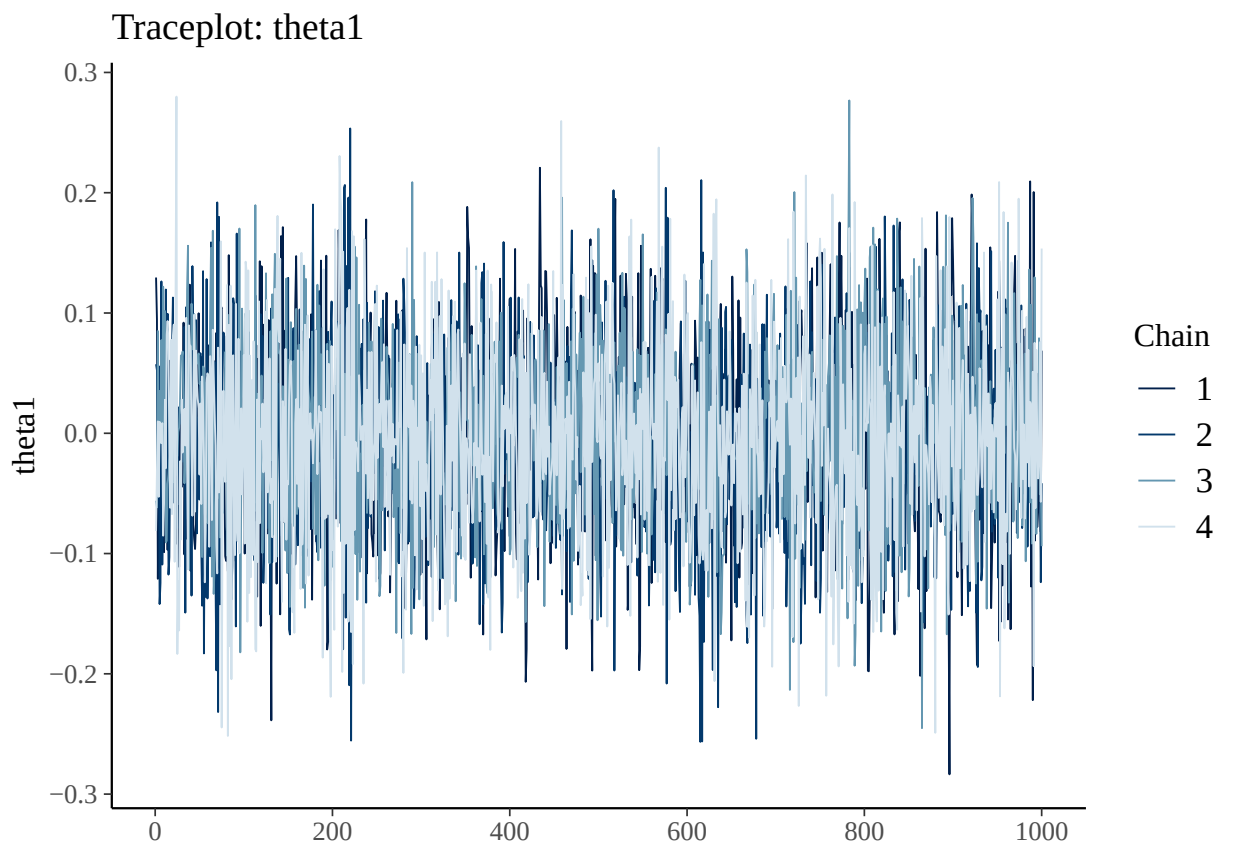
Traceplot: beta0




Traceplot: beta1
beta1
0.5
0.0
−0.5
−1.0
0
200
400
600
800
1000
Chain
1
2
3
4

Traceplot: theta0





5. G-Computation

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

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

message("Running g-computation...")
```

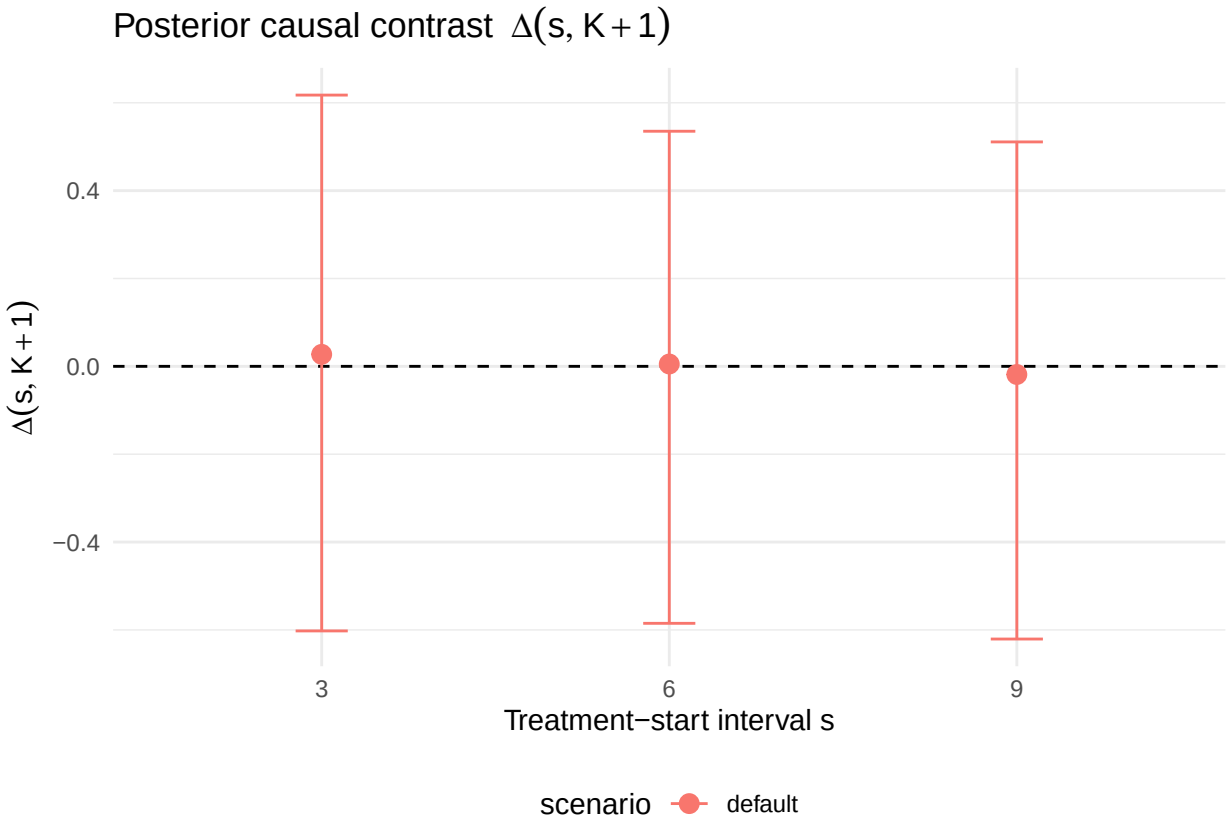
```
## Running g-computation...
```

```
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  0.027297245 -0.6023125 0.6175602
##  6  0.005338704 -0.5850453 0.5351935
##  9 -0.018524600 -0.6207968 0.5109547
```

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

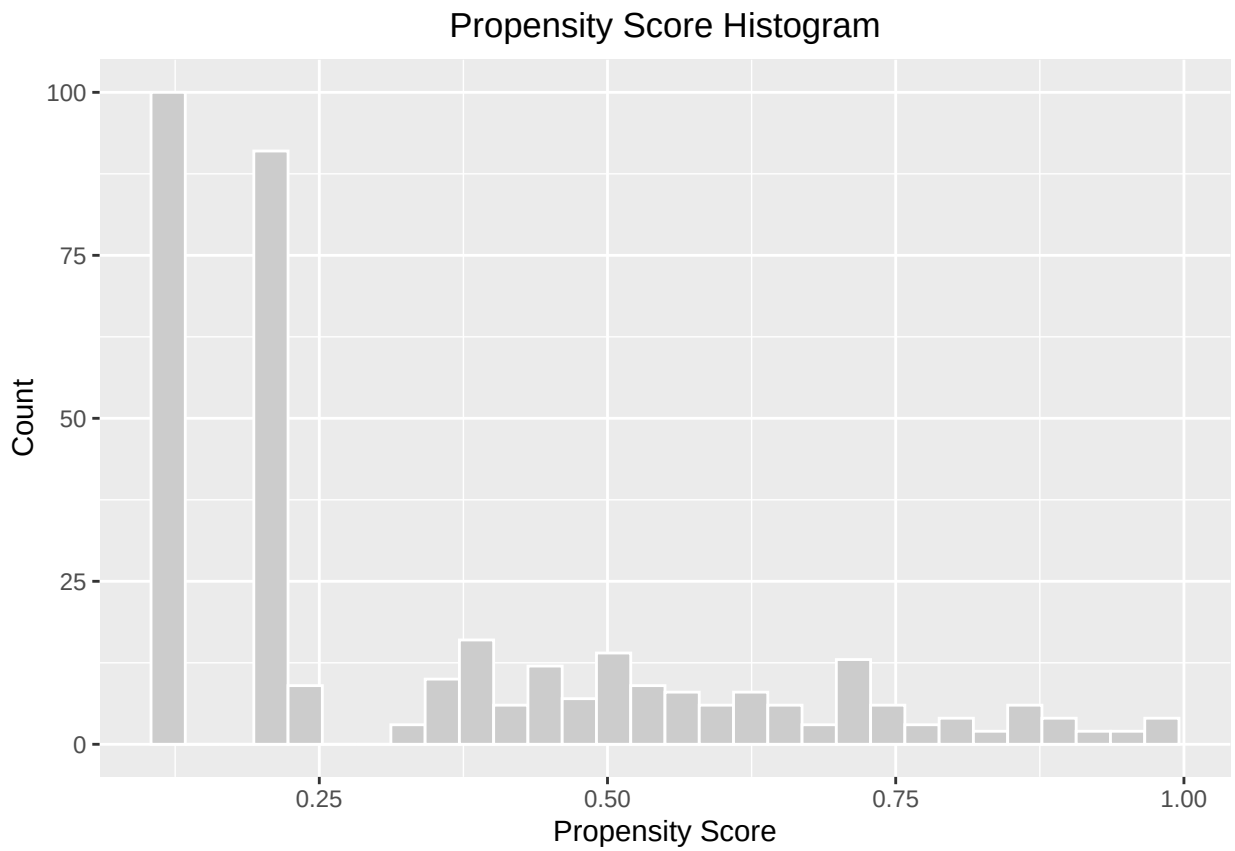



```r
plot(gcomp, interactive = TRUE, ref_line = 0)
```

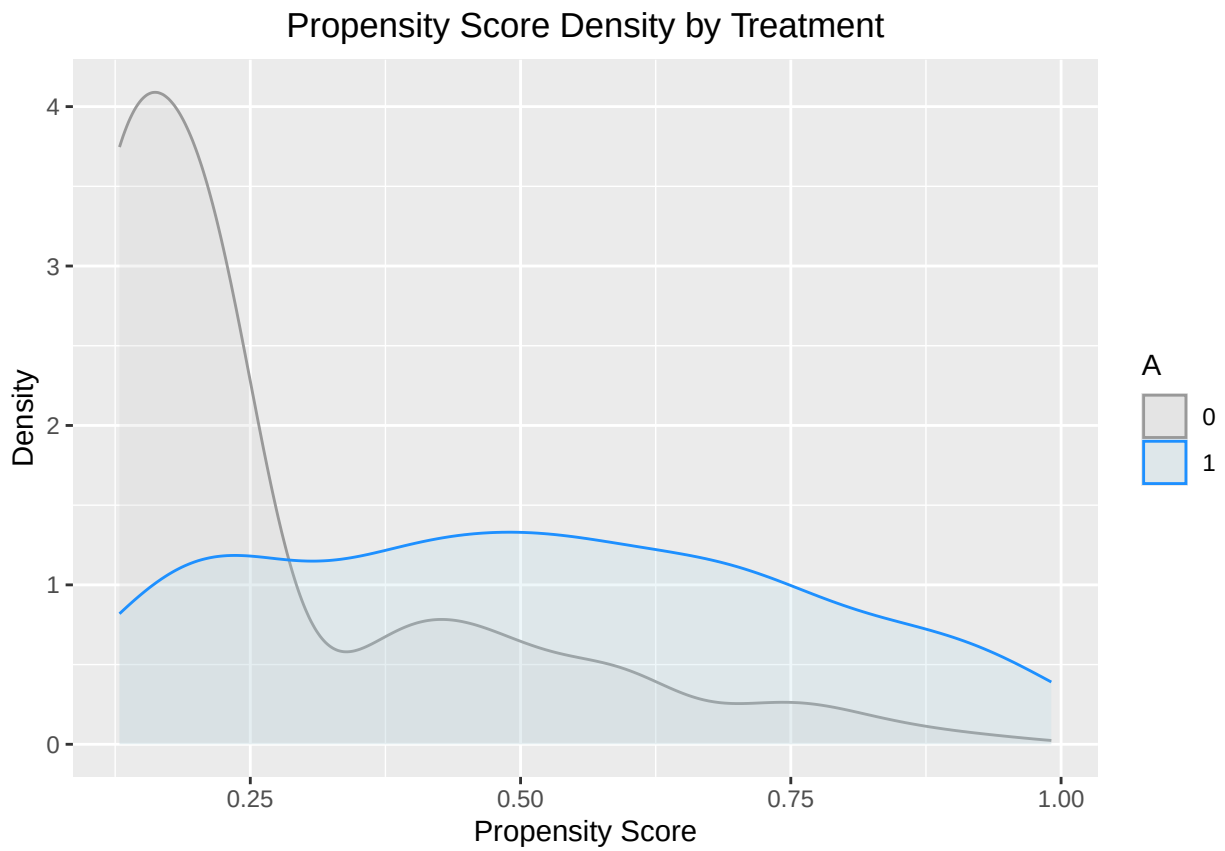
6. Propensity Score Diagnostics

Assess overlap and positivity for model validity.

```r
ps_diag <- propensity_score_diagnostics(
  fit$data_preprocessed,
  treat_col = "A",
  covariates = c("lagYk","k_idx")
)
plot(ps_diag, type = "histogram")
```



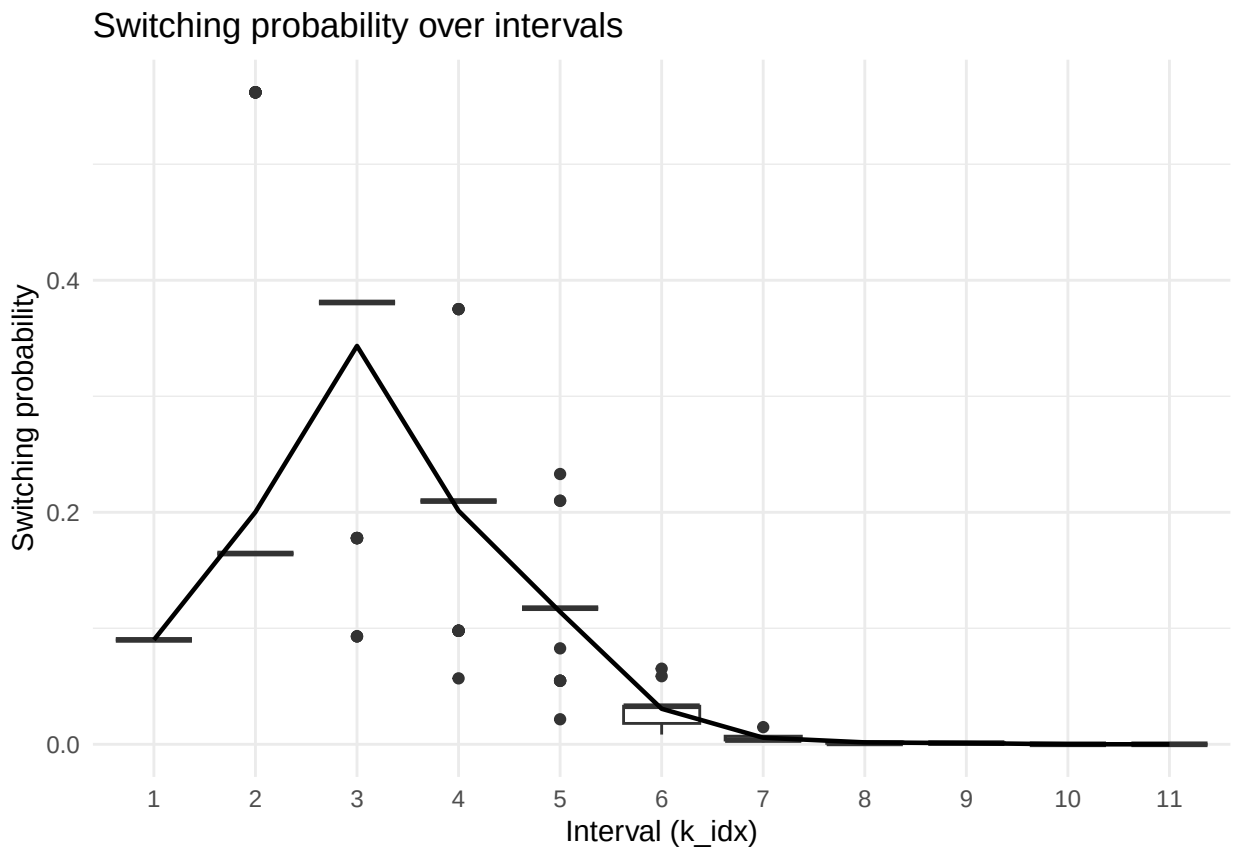

```
plot(ps_diag, type = "density")
```



7. Switching Probability Summary

Visualize the probability of treatment switching across time intervals.

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



8. Summary of Causal Estimates

This table summarizes posterior distributions of model parameters and g-computation estimates.

```
sum_tbl <- result_summary_table(
  fit_out   = fit,
  gcomp_out = gcomp,
  s_vec     = s_vec,
  format    = "kable",
  pars_to_report = c("beta0","beta1","theta0","theta1","theta_lag")
)
print(sum_tbl)
```

```
##
##
## Table: Posterior Parameters
##
## |Parameter  |       Mean|      X2.5.|     X97.5.|      Rhat|    n_eff|      MCSE|  CI_width|
## |:----------|----------:|----------:|----------:|---------:|--------:|---------:|---------:|
## |theta0[8]  |  1.2778096|  0.8170051|  1.7859695| 1.0015809| 2264.325| 0.0052266| 0.9689644|
## |theta0[7]  |  1.2422718|  0.8949164|  1.5951514| 1.0027139| 2310.635| 0.0037906| 0.7002350|
## |theta0[10] |  1.2326209|  0.5583714|  1.9492660| 1.0001993| 4139.792| 0.0053465| 1.3908946|
## |theta0[2]  |  1.0969824|  0.9780719|  1.2114804| 1.0006062| 3920.799| 0.0009469| 0.2334086|
## |theta0[9]  |  1.0814387|  0.4039912|  1.6528517| 1.0057404| 1480.434| 0.0080535| 1.2488605|
```

```
## |theta0[6]  |  1.0713936|  0.7704645|  1.3744647| 1.0026391| 1723.776| 0.0037316| 0.6040002|
## |theta0[4]  |  1.0478263|  0.8464265|  1.2390124| 0.9996669| 4333.342| 0.0015203| 0.3925859|
## |theta0[3]  |  1.0336144|  0.8470569|  1.2028357| 1.0049544| 3140.715| 0.0016008| 0.3557787|
## |theta0[5]  |  0.9582156|  0.6867505|  1.2025883| 1.0056761| 2903.138| 0.0024122| 0.5158378|
## |theta0[11] |  0.8980481| -0.1739855|  1.6704040| 1.0001999| 3100.707| 0.0081792| 1.8443895|
## |theta1     | -0.0045741| -0.1565815|  0.1529654| 1.0010792| 4520.103| 0.0011812| 0.3095469|
## |beta1      | -0.2738372| -0.8349362|  0.2539825| 1.0019275| 2655.593| 0.0052570| 1.0889187|
## |beta0[3]   | -1.0746173| -1.6278035| -0.5065139| 1.0012541| 3020.079| 0.0052647| 1.1212896|
## |beta0[7]   | -1.2202484| -2.2784033| -0.1526570| 1.0009878| 4414.037| 0.0079061| 2.1257463|
## |beta0[4]   | -1.3227703| -2.0599208| -0.6460827| 0.9994648| 4689.237| 0.0052212| 1.4138381|
## |beta0[5]   | -1.3640904| -2.2492749| -0.6078952| 1.0004749| 4480.888| 0.0061163| 1.6413797|
## |beta0[6]   | -1.3774038| -2.4994385| -0.5047122| 0.9998238| 3288.313| 0.0084864| 1.9947263|
## |beta0[2]   | -1.3808169| -1.9009925| -0.8952062| 0.9996020| 4171.389| 0.0039650| 1.0057864|
## |beta0[11]  | -1.3960660| -3.3607202|  0.1180596| 0.9997751| 3152.262| 0.0146346| 3.4787798|
## |beta0[10]  | -1.4313400| -3.3580137| -0.0561994| 1.0011239| 2855.590| 0.0148535| 3.3018144|
## |beta0[9]   | -1.4793041| -3.3338234| -0.1845534| 0.9996591| 3096.053| 0.0137655| 3.1492699|
## |beta0[8]   | -1.5055170| -3.3313261| -0.3513822| 1.0013344| 3431.080| 0.0126382| 2.9799440|
## |theta0[1]  | -2.2065630| -2.8607747| -1.6549254| 1.0057596| 2212.840| 0.0064082| 1.2058493|
## |beta0[1]   | -4.1251431| -6.1958689| -2.6873879| 0.9998607| 3643.418| 0.0147233| 3.5084810|
##
##
##
## Table: delta(s, K+1)
##
## |  s|       Mean|      X2.5.|    X97.5.| CI_width|
## |--:|----------:|----------:|---------:|--------:|
## |  3|  0.0272972| -0.6023125| 0.6175602| 1.219873|
## |  6|  0.0053387| -0.5850453| 0.5351935| 1.120239|
## |  9| -0.0185246| -0.6207968| 0.5109547| 1.131752|
```