Kaplan-Meier Simulation Study

Aim

In the following a simulation study is conducted, where we investigate the Kaplan-Meier estimator. The estimand is

$$S(\tau|A_0 = 1) = P(T \geq \tau|A_0 = 1), \quad S(\tau|A_0 = 0) = P(T \geq \tau|A_0 = 0)$$

For some τ. T is here the survival time *post* T2D diagnose. As mentioned the survival function will be estimated by the Kaplan-Meier estimator. Post T2D diagnose the setting is simpler, as there are no competing events, only death and censoring. The Kaplan-Meier estimator will then be an unbiased estimator of the survival probability.

The true value of $P(T \geq \tau|A_0 = a_0)$ can be estimated by a Monte Carlo approach. That is by simulating a large data set without censoring, and calculating the proportion of dead at time τ post T2D diagnose, in respectively treatment and placebo group.

After having estimated the true value, we generate B data sets with N individuals. For each data set we find the Kaplan-Meier estimate of $P(T \geq \tau|A_0 = a_0)$, the SE, and the upper and lower confidence bands. We plot a histogram of the estimates, calculate the bias of the estimator and the coverage of the associated CI.

We conduct the simulation study for the following 4 different parameter values

```
beta_L0_L <- c(1.5, 1.3, 2, 0.9)
beta_A0_L <- c(-1, -2, -3, -1.2)
beta_L_D <- c(0.9, 1, 0.5, 0.8)
beta_L0_D <- c(0.6, 0.7, 0, 0.6)
beta_A0_D <- c(0, 0, -0.1, -0.2)
```

And the following values of η and ν

```
eta <- c(0.1,0.3,0.1)
nu <- c(1.1,1.3,1.1)
```

True estimate

We simulate the large data

```
set.seed(836)
N <- 2*10^5
data_list <- list()

for(i in 0:3){
  data_list[[i*4 + 1]] <- simT2D(N = N, eta = eta, nu = nu, beta_L0_D = beta_L0_D[1],
                beta_L0_L = beta_L0_L[1], beta_A0_L = beta_A0_L[1],
                beta_L_D = beta_L_D[1], beta_A0_D = beta_A0_D[1], cens = 0)
  data_list[[i*4 + 2]] <- simT2D(N = N, eta = eta, nu = nu, beta_L0_D = beta_L0_D[2],
                beta_L0_L = beta_L0_L[2], beta_A0_L = beta_A0_L[2],
                beta_L_D = beta_L_D[2], beta_A0_D = beta_A0_D[2], cens = 0)
  data_list[[i*4 + 3]] <- simT2D(N = N, eta = eta, nu = nu, beta_L0_D = beta_L0_D[3],
                beta_L0_L = beta_L0_L[3], beta_A0_L = beta_A0_L[3],
```

```
                beta_L_D = beta_L_D[3], beta_A0_D = beta_A0_D[3], cens = 0)
  data_list[[i*4 + 4]] <- simT2D(N = N, eta = eta, nu = nu,, beta_L0_D = beta_L0_D[4],
                beta_L0_L = beta_L0_L[4], beta_A0_L = beta_A0_L[4],
                beta_L_D = beta_L_D[4], beta_A0_D = beta_A0_D[4], cens = 0)
}
```

We calculate the survival proportion $\tau = 1$ years post T2D diagnose for each data set.

```
tau <- 1
surv_prop1 <- numeric(16)
surv_prop0 <- numeric(16)

for(i in 1:16){
  # T2D events
  T2D_events <- data_list[[i]][Delta == 2]

  # T2D people
  T2D_peeps <- data_list[[i]][ID %in% T2D_events$ID]

  # Setting T_0 to debut time of diabetes
  T2D_peeps[, Time_T2D := Time - min(Time), by = ID]

  # Removing the new Time 0
  T2D_peeps <- T2D_peeps[Delta != 2]

  surv_prop1[i] <- nrow(T2D_peeps[Time_T2D > tau & A0 == 1]) / nrow(T2D_peeps[A0 == 1])
  surv_prop0[i] <- nrow(T2D_peeps[Time_T2D > tau & A0 == 0]) / nrow(T2D_peeps[A0 == 0])
}
```

We find the mean and variance

```
surv_prop_mean1 <- numeric(4)
surv_prop_mean0 <- numeric(4)
surv_prop_sd1 <- numeric(4)
surv_prop_sd0 <- numeric(4)

for(i in 1:4){
  surv_prop_mean1[i] <- mean(surv_prop1[0 : 3 * 4 + i])
  surv_prop_mean0[i] <- mean(surv_prop0[0 : 3 * 4 + i])
  surv_prop_sd1[i] <- sd(surv_prop1[0 : 3 * 4 + i])
  surv_prop_sd0[i] <- sd(surv_prop0[0 : 3 * 4 + i])
}
```

Kaplan-Meier simulation study

For each of set of parameter values, we simulate B data sets with N subjects, and calculate the KM estimate and CI's, we do this with use of the function `simStudyT2D`.

```
set.seed(836)
N <- 1000
B <- 500
res1 <- simStudyT2D(N = N, B = B, eta = eta, nu = nu, beta_L0_D = beta_L0_D[1],
                beta_L0_L = beta_L0_L[1], beta_A0_L = beta_A0_L[1],
                beta_L_D = beta_L_D[1], beta_A0_D = beta_A0_D[1], tau = tau)
res2 <- simStudyT2D(N = N, B = B, eta = eta, nu = nu, beta_L0_D = beta_L0_D[2],
```

```r
                 beta_L0_L = beta_L0_L[2], beta_A0_L = beta_A0_L[2],
                 beta_L_D = beta_L_D[2], beta_A0_D = beta_A0_D[2], tau = tau)
res3 <- simStudyT2D(N = N, B = B, eta = eta, nu = nu, beta_L0_D = beta_L0_D[3],
                 beta_L0_L = beta_L0_L[3], beta_A0_L = beta_A0_L[3],
                 beta_L_D = beta_L_D[3], beta_A0_D = beta_A0_D[3], tau = tau)
res4 <- simStudyT2D(N = N, B = B, eta = eta, nu = nu, beta_L0_D = beta_L0_D[4],
                 beta_L0_L = beta_L0_L[4], beta_A0_L = beta_A0_L[4],
                 beta_L_D = beta_L_D[4], beta_A0_D = beta_A0_D[4], tau = tau)
```

Histograms of estimate

```r
plot_data <- data.table(ests = c(res1[,'Est 0'], res1[,'Est 1'],
                                 res2[,'Est 0'], res2[,'Est 1'],
                                 res3[,'Est 0'], res3[,'Est 1'],
                                 res4[,'Est 0'], res4[,'Est 1']),
                        Lower = c(res1[,'Lower 0'], res1[,'Lower 1'],
                                  res2[,'Lower 0'], res2[,'Lower 1'],
                                  res3[,'Lower 0'], res3[,'Lower 1'],
                                  res4[,'Lower 0'], res4[,'Lower 1']),
                        Upper = c(res1[,'Upper 0'], res1[,'Upper 1'],
                                  res2[,'Upper 0'], res2[,'Upper 1'],
                                  res3[,'Upper 0'], res3[,'Upper 1'],
                                  res4[,'Upper 0'], res4[,'Upper 1']),
                        A0 = rep(c(rep(0, B), rep(1, B)),4),
                        Setting = c(rep("A", 2 * B), rep("B", 2 * B),
                                    rep("C", 2 * B),rep("D", 2 * B)),
                        true_val = c(rep(surv_prop0[1], B), rep(surv_prop1[1],B),
                                     rep(surv_prop0[2], B), rep(surv_prop1[2],B),
                                     rep(surv_prop0[3], B), rep(surv_prop1[3],B),
                                     rep(surv_prop0[4], B), rep(surv_prop1[4],B)))
pp <- ggplot(plot_data) +
  geom_histogram(aes(x = ests, y = ..density..), color = "white", fill = "steelblue")+
  geom_vline(aes(xintercept = true_val), color = "darkred")+
  facet_wrap( ~ A0 + Setting, ncol = 4)

ggsave("hist_sim_KM.jpeg", pp, width = 7, height = 5)
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Bias

```r
plot_data[,.(Bias=mean(ests - true_val)), by = .(Setting, A0)] |> knitr::kable()
```

Setting	A0	Bias
A	0	0.0025996
A	1	0.0051925
B	0	-0.0045115

Setting	A0	Bias
B	1	0.0076099
C	0	0.0075928
C	1	0.0169093
D	0	0.0066346
D	1	0.0065349

```
#plot_data[,.(Bias=mean(ests - true_val)), by = .(Setting, A0)] |> xtable::xtable()
```

Coverage

```
plot_data[,.(Coverage=mean(Lower <= true_val & true_val <= Upper)), by = .(Setting, A0)] |> knitr::kabl
```

Setting	A0	Coverage
A	0	0.928
A	1	0.950
B	0	0.932
B	1	0.938
C	0	0.936
C	1	0.926
D	0	0.958
D	1	0.934

```
plot_data[,.(Coverage=mean(Lower <= true_val & true_val <= Upper)), by = .(Setting, A0)] |> xtable::xtal
```

```
## % latex table generated in R 4.4.2 by xtable 1.8-4 package
## % Tue Apr 15 09:26:40 2025
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrr}
##   \hline
##  & Setting & A0 & Coverage \\
##   \hline
## 1 & A & 0.00 & 0.93 \\
##   2 & A & 1.00 & 0.95 \\
##   3 & B & 0.00 & 0.93 \\
##   4 & B & 1.00 & 0.94 \\
##   5 & C & 0.00 & 0.94 \\
##   6 & C & 1.00 & 0.93 \\
##   7 & D & 0.00 & 0.96 \\
##   8 & D & 1.00 & 0.93 \\
##    \hline
## \end{tabular}
## \end{table}
```