Causal survival analysis

Estimation of the Average Treatment Effect (ATE) in Causal Survival Analysis : Practical Recommendations

Charlotte Voinot[1] CMEI, Sanofi R&D
Premedical, INRIA
INSERM
Université de Montpellier
Julie Josse[2] Premedical, INRIA
INSERM
Université de Montpellier
Bernard Sebastien[3] CMEI, Sanofi R&D

Last modified: 2024-09-10

Abstract

Keywords: Causal survival analysis, Weighting, Robustness, Observational study, Censoring mechanism

Contents

[1] Corresponding author: charlotte.voinot@sanofi.com

[2] Corresponding author: julie.josse@inria.fr

[3] Corresponding author: bernard.sebastien@sanofi.com

1 Introduction

Survival analysis is a highly active area of research with significant applications in biostatistics, engineering, social sciences, and more. This statistical field focuses on analyzing time-to-event data. One of the major challenges in survival analysis is dealing with censoring, which occurs when only partial information about an event is known. A tempting approach might be to remove this incomplete data, but this would lead to substantial bias, as partial information still holds value. Conversely, treating partial information as complete also introduce significant bias. Even if the event is not fully observed, it is crucial to consider the time up until censoring and not just the uncensored time for the analysis of these event times. This so-called censoring can be most effectively handled using survival analysis techniques, which depend on the specific characteristics of the censoring. One key characteristic of censoring is its type, which indicates where the lack of information lies. The most common type is *right-censoring*, where the event has not occurred by the time of the last observation, suggesting it will happen after this point. Left-censoring occurs when the event happened before the start of the study. Interval censoring occurs when the event happened between two observation times, but the exact time is unknown. Another crucial characteristic to consider is the censoring mechanism, as the properties for survival analysis methods often depend on it. The mechanism can be independent, also known as *non-informative censoring*, meaning that the probability of an individual being censored is independent of the covariates. Well-known estimators, such as the Kaplan-Meier (Kaplan and Meier 1958) or Cox estimators (Cox 1972), require an independent censoring mechanism to be consistent. In real-life conditions, conditionally independent censoring, or *informative censoring*, is more likely to occur. This means that the probability of being censored is dependent on the survival time and/or covariates, leading to selection bias if ignored. Techniques used in the context of independent censoring are not valid in this scenario.

On the other hand, *causal inference* (D. B. Rubin 1974; Hernán and Robins 2010; Imbens and Rubin 2015; Hernán MA 2020) aims to evaluate the effect of a treatment, policy or intervention on an outcome. It has acquired considerable importance like survival analysis in a variety of fields, including the social sciences, health and economics. In a dichotomous treatment assignment ($A = 1$: treated, $A = 0$: untreated), the outcome Y can be equal to $Y^{a=1}$ (the outcome that would have been observed under the treatment $a = 1$) if treated or $Y^{a=0}$ if not. These variables are referred to as *potential outcome*. The *individual treatment effect* (ITEs) is the main objective in causal inference. This measure (equals to $Y^{a=1} - Y^{a=0}$) allows to evaluate the impact of a treatment per individual. But, at most one of the potential outcomes can be observed per individual i.e. an observation cannot experiment both treatments. The *Average Treatment Effects* (ATE), one of fundamental estimand in causal analysis, get round this fundamental problem by average the ITEs. On possibility to evaluate causal effect is by running a randomized clinical trial (RCT). This experiment completely controls over treatment assignment which ensure balanced distribution of covariates between treated and control. RCTs are considered as gold standard to assess causal effect because one simple estimator (such as Difference in Means) can measure the causal treatment effect. However, due to restrictive inclusion/exclusion criteria, this controlled design can lead to a lack of generalization to the target population. Recently, the use of observational studies to evaluate real-world data (RWD) in clinical studies has been growing rapidly. But, contrary to RCTs, extracting causal evidence from observational data suppose to control for confounding variables because of confounding bias. In practice, under some strong causal assumptions such as unconfoundedness, estimation methods take this into account. ATE can be evaluated in using propensity-score based methods IPW, regression methods or augmented IPW (Imbens and Rubin 2015).

This article will focus on causal survival analysis which bridges the fields of causal inference and survival analysis: the aim is to assess the causal effect of a treatment or an intervention on a outcome which is a time until an event occurs in the presence of censoring. In this article, we will focus on estimating the Average Treatment Effect (ATE) on time to event data with static treatment assignment, baseline covariates and also right-censoring. To do so, in Section 2, we will define the causal treatment effect with survival outcome. In particular, we will focus on the Restricted Mean Survival Time (RMST) quantity that is easy to interpret. Then, in Section 3, we will define identifiability assumptions that are necessary to overcome the two main issues we presented above i.e. censoring mechanism and treatment assignment. Next, we will present and implement the corresponding estimators in the context of a randomized controlled trial. In the same way than RCT, Section 4 focuses on presenting and implementing estimators in the context of observational study.

Despite the abundant literature on causal survival estimators in our context, there are few available packages for these estimators. Then, for each of the estimator in Section 3 and Section 4, an implementation is proposed. Finally in Section 7, these estimators will be challenged by different simulation scenarios representing different context: RCT or observational study and independent or conditionally independent censoring and with good model specification or mis-specification of nuisance parameters or positivity violation of censoring. In this section, We will also provide practical recommendations on which estimators should be used in different context, based on convergence criteria, implementation complexity, computing time etc.

2 Context and Notations

2.1 Notations

Let's consider a sample of n i.i.d observations that are described by:

- X_i: the baseline covariates, $X \in \mathbb{R}^p$.
- A_i: the binary treatment, $A \in \{0, 1\}$.
- C_i: the time to censoring, $C \in \mathbb{R}^+$.
- $T_i(0)$: the survival time to the event of interest had the patient received control $A_i = 0$.
- $T_i(1)$: the survival time to the event of interest had the patient received treatment $A_i = 1$.
- $T_i = A_i T_i(1) + (1 - A_i) T_i(0)$, $T \in \mathbb{R}^+$: the observed outcome (see identifiability assumption Equation 1).
- $T_i \wedge \tau = min(T_i, \tau)$,: the truncated observed outcome at τ.
- $\Delta_i = I\{T_i \leq C_i\}$: the status of censoring, where $I\{\cdot\}$ is the indicator.
- $\Delta_i^\tau = I\{T_i \wedge \tau \leq C_i\}$: the status of censoring truncated at τ (introduced later)

- $\tilde{T}_i = T_i \wedge C_i = \min(T_i, C_i)$: the observed time. When an observation is censured, then its observed time is equal to the censoring time. The censoring time is type II censoring (right censoring).
- $S(t) = P(T \geq t)$: the survival curve, which represents a key function in survival analysis. It denotes the probability that an individual will survive beyond a given time t.

The observed data can be summarized as a quadruplet $(X_i, A_i, \Delta_i, \tilde{T}_i)$ represented in Table 1.

Table 1: Example of survival data with covariates, treatment, the censoring time, the status of censoring and the potential outcomes and observed outcomes.

ID	Covariates			Treatment	Censoring	Status	Outcomes			
ID	X_1	X_2	X_3	A	C	Δ	T(0)	T(1)	T	$\tilde{T}$
1	1	1.5	4	1	?	1	?	200	200	200
2	5	1	2	0	?	1	100	?	100	100
3	9	0.5	3	1	200	0	?	?	?	200

2.2 Definition of treatment effect

In causal inference, the primary goal is to estimate the individual causal effect of the treatment denoted as $\theta_i = T_i(1) - T_i(0)$ (D. B. Rubin 1974; Hernán and Robins 2010). However, this quantity cannot be observed because at most one outcome can be observed per sample (see Table 1). Furthermore, censoring may also mask outcomes (Turkson, ayiah-mensah, and Nimoh 2021).

Despite these challenges, certain identifiability assumptions, described in Section 2.3, enable us for estimating the average treatment effect (Díaz et al. 2019; Ozenne et al. 2020) (ATE) which is defined as follows:

Definition 2.1 (Causal effect: Average treatment effect in survival analysis (ATE))**.**

$$\theta = \mathbb{E}\left[y(T(1)) - y(T(0))\right]$$

- with $y(T) = T \wedge \tau = \min(T, \tau)$ with τ a fixed time horizon; then, $E(y(T))$ becomes the restricted mean survival time (RMST) at time τ (Chen and Tsiatis 2001).
- with $y(T) = I\{T > \tau\}$ the indicator of survival with τ a fixed time horizon.

In this article, we only focus on restricted mean survival time ($y(T) = T \wedge \tau = \min(T, \tau)$) as the estimand of interest.

There is a direct relationship between RMST and the survival curve. As a result, the expression $E(T \wedge \tau)$ can be also expressed as:

$$E(T \wedge \tau) = E\left(\int_0^{T \wedge \tau} 1dt\right) = E\left(\int_0^{\tau} I\{T > t\}dt\right) = \int_0^{\tau} E(I\{T > t\})dt = \int_0^{\tau} S(t)dt$$

With this expression, we can interpret θ_{RMST} as the mean difference between the survival function of treated and control until a fixed time horizon τ. RMST can be interpreted as the average survival time from baseline to a pre-specified time τ. Thus, a difference in RMST (θ_{RMST}) value of 10 days with $\tau = 200$ means that on average the treatment increases the survival time by 10 days at 200 days.

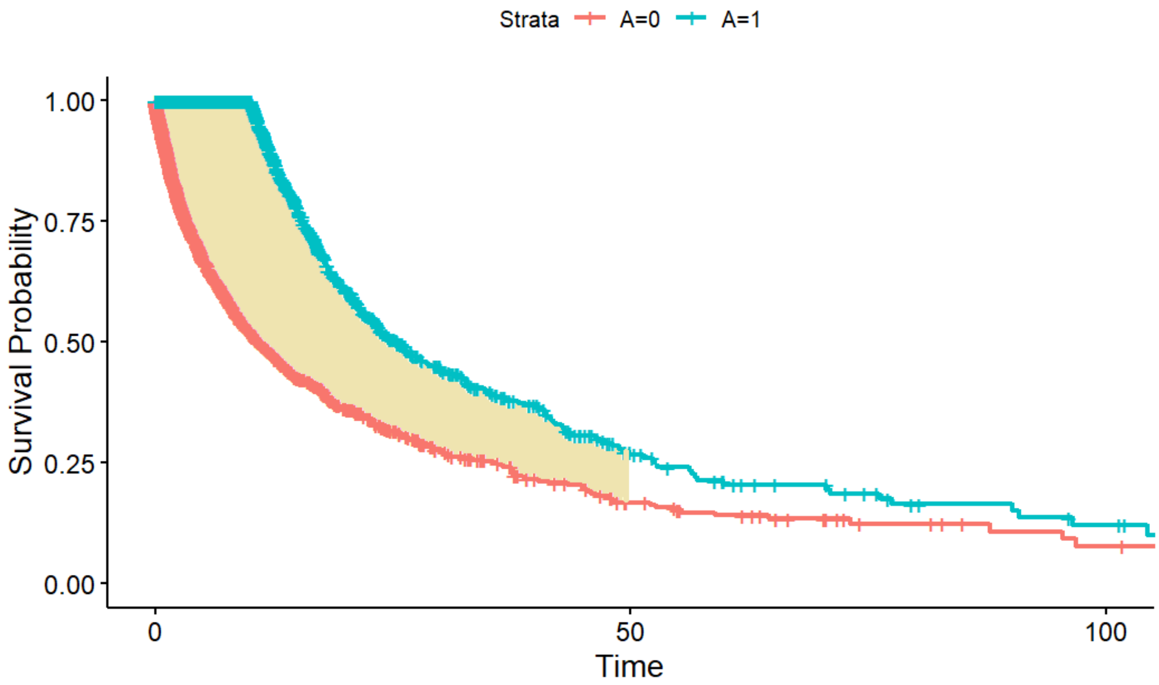


Figure 1: Plot of Kaplan-Meier survival curve for treated and control. The θ_{RMST} at $\tau = 50$ is represented in yellow

The difference in Restricted Mean Survival Time is a time-dependent measure that can be easily understood with the help of Figure 1. The average survival time will naturally vary based on the value of τ.

A very naive (and biased) estimator of the average treatment effect could be to compute the difference of the mean of the uncensored survival time between the treated and the untreated group:

```
# Naive estimator:
# Warning, this estimator does not take into account censoring
# This estimator is in all context biased
Naive <- function(data, tau) {
  # Remove censored observations
  data<- data[data$status == 1, ]
  # Compute the restricted survival time
  data$T_obs_tau <- ifelse(data$T_obs >= tau, tau, data$T_obs)

  # Compute the difference of the restricted mean survival time of treated
  # and control
  mean_naive <- mean(data$T_obs_tau[data$A == 1]) -
    mean(data$T_obs_tau[data$A == 0])

  return(mean_naive)
}
```

This estimator eliminates censored observations, which represents a considerable loss of information. A reliable estimator of θ_{RMST} must include censored observations but also be expressed with observed quantities. To do this, certain identifiability assumptions are necessary to allow the ATE to be identified in different contexts.

2.3 General identifiability assumptions

A general identifiability assumption, already well-known in the field of causal inference, needs to be introduced:

Assumption 1: Consistency and no-interference Assumption (Stable Unit Treatment Value Assumption: SUTVA)

$$T = AT(1) + (1 - A)T(0) \tag{1}$$

This Equation 1 means that unit i's potential outcomes do not depend on the treatments of other units. This is known as the no-interference assumption. Basically, this assumption can be violated in cases of infectious diseases. Additionally, Equation 1 means that there are no other versions of the treatment (Ding 2023).

As introduced in Section 1, it is crucial to consider the mechanism of censoring to identify appropriate estimators of θ_{RMST}.

2.3.1 Censoring mechanism

This article focuses only on right censoring with two mechanisms of censoring considered. The first one is independent censoring:

Assumption 2: Independent/ Non informative censoring

$$C \perp\!\!\!\perp T(0), T(1), X, A \tag{2}$$

Under Equation 2, subjects censored at time t are representative of all subjects who remain at risk at time t. It is as if the censored subjects were randomly selected from all subjects.

The case of dependent censoring considers that censoring is conditionally independent on covariates.

Assumption 3: Conditionally independent censoring

$$C \perp\!\!\!\perp T(0), T(1) | X_c, A \tag{3}$$

with X_c is the set of covariables which influence the censoring mechanism.

Under Equation 3, within subgroups represented by $X_c = x$, subjects censored at time t are representative of all subjects in their subgroup who remain at risk at time t. It is as if the censored subjects were randomly selected inside each subgroup. This assumption is very similar to the assumption of uncounfoundedness in causal inference.

Another assumption for identifiability of θ_{RMST} is required in case of conditionally independent censoring: we need to assume that all subjects have a positive probability to remain uncensored at their failure time.

Assumption 4: Positivity / Overlap for censoring

$$0 < P(C > t \mid X_c = x, A = a) < 1, \forall t \leq \tau. \tag{4}$$

If this assumption is violated, i.e. for a time t, $\mathbb{P}(C > t \mid X = x, A = a) = 0$ or $\mathbb{P}(C > t \mid X = x, A = a) = 1$, then this excludes that we have results observed after time t or respectively that we have only results observed after time t. This assumption is necessary to be able to balance the censoring mechanism within subgroup.

The value of τ can be adjusted to ensure that the hypothesis of positivity is met. For example, if we consider a clinical study where patients are followed for 5 years and some patients leave the study before the end, either because they have developed serious side effects (and have therefore decided to stop the treatment), or because their state of health has deteriorated so much that they have been transferred to a palliative care unit. In this case, censoring is dependent, as the probability of leaving the study is linked to the severity of the illness and the probability of remaining uncensored for severe patient at 5 years is zero. Consequently, the potential outcomes $T(0)$ and $T(1)$ are entirely unobserved beyond the one-year mark. To address this limitation, one approach is to adjust the threshold time τ such that each participant has a chance of remaining uncensored up to their revised threshold time.

Although this assumption is strong, it can be verified. One way to check it is by modeling the conditional probability of censoring using a Cox model or a survival forest and then assessing whether this probability is non-zero across all covariate subgroups.

The other assumptions of identifiability depend on the design of the study. We will start with the simplest but stronger design: the Randomized Clinical Trial.

3 Causal survival analysis with a Randomized Control Trial

Randomized clinical trials (RCTs) are the gold standard for establishing the effect of a treatment on an outcome, because treatment allocation is under control, which ensures (asymptotically) the balance of covariates between treated and controls, and thus avoids problems of confounding between covariables and treatment. The core assumption in a RCT is the random assignment of the treatment (D. B. Rubin 1974).

Assumption 5: Random treatment assignment

$$A \perp\!\!\!\perp (T(0), T(1), C, X) \tag{5}$$

Equation 5 implies that the treatment is given at random and is independent of both the potential outcomes and the covariates. It is like flipping a coin to decide the treatment assignment. In that case, as the covariates are balanced between treated and control (the treatment assignment does not depend on covariates), the causal effect is direct. The Figure 2 illustrates a simple causal graph when the study is randomized without censoring.

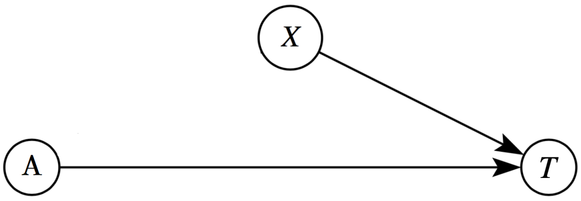


Figure 2: Illustration of a simple causal graph in RCT survival data without censoring (A is the treatment, X_t the confounding variables and T is the time to event outcome)

In the context of no censoring, causal effect is straightforward and can be measure with one simple estimator (such as OLS estimator). But in our case, censoring is present and the censoring mechanism has to be considered. In the next section, we will introduce identifiability of θ_{RMST} and the corresponding estimator in the simple case: independent censoring.

3.1 Independent censoring

3.1.1 Identifiability

Under Equation 5 (random treatment assignment) and Equation 2 (independent censoring), the difference in RMST, θ_{RMST}, can be identified as follows:

$$\begin{aligned}
\theta_{RMST} &= \mathbb{E}[T(1) \wedge \tau - T(0) \wedge \tau] \\
&= \int_0^\tau \mathbb{E}[I\{T(1) > t\} - I\{T(0) > t\}]dt && \text{(By definition)} \\
&= \int_0^\tau \mathbb{E}[I\{T(1) > t\}] - \mathbb{E}[I\{T(0) > t\}]dt && \text{(By linearity of expectation)} \\
&= \int_0^\tau \mathbb{E}[I\{T(1) > t|A = 1\}] - \mathbb{E}[I\{T(0) > t|A = 0\}]dt && \text{(Random treatment assignment As. 5)} \\
&= \int_0^\tau \mathbb{E}[I\{T > t|A = 1\}] - \mathbb{E}[I\{T > t|A = 0\}]dt && \text{(By consistency As.1)} \\
&= \int_0^\tau \mathbb{P}(T > t|A = 1) - \mathbb{P}(T > t|A = 0)dt
\end{aligned} \tag{6}$$

$\mathbb{P}(T > t|A = a)$ $(S(t, A = a))$ is the survival function and can be estimated using usual estimator of survival function such as Kaplan meier estimator, cox model, etc.

3.1.2 Estimation with Unadjusted Kaplan-meier

The straightforward estimator of the Equation 6 is the difference of Unadjusted Kaplan Meier estimator for the treated and control group:

Definition 3.1 (Unadjusted Kaplan meier estimator)**.**

$$\begin{aligned}
\hat{S}_{KM}(t \mid a) &= \prod_{j=1,t_j<=t} \left(1 - \frac{\sum_i I\{T_i = t_j, C_i \geq t_j, A_i = a\}}{\sum_k I\{T_k \geq t_j, C_k \geq t_j, A_k = a\}}\right) \\
&= \prod_{j=1,t_j<=t} \left(1 - \frac{\sum_i I\left\{\tilde{T}_i = t_j, \Delta_i = 1, A_i = a\right\}}{\sum_k I\left\{\tilde{T}_k \geq t_j, A_i = a\right\}}\right)
\end{aligned} \tag{7}$$

The corresponding θ_{RMST} is obtained in integrating from 0 to τ the difference between unadjusted Kaplan Meier estimator of the treated and controls (Equation 6):

$$\hat{\theta}_{RMST}(\tau) = \int_0^\tau \left(\hat{S}_1(t) - \hat{S}_0(t)\right) dt$$

3.1.2.1 Properties of Unadjusted Kaplan Meier estimator

Unadjusted Kaplan Meier for the survival function is a nonparametric MLE (maximum likelihood estimator). Its asymptotic properties has been derived firstly by Breslow and Crowley (1974). An other simple way of deriving the asymptotic properties is to consider Kaplan-Meier estimator as a Martingale process (*Survival and Event History Analysis: A Process Point of View* 2008). Kaplan-Meier is proved to be:

- Uniformly consistent: $\sup_{s\in[0,t]} |\hat{S}(s) - S(s)| \xrightarrow{\mathbb{P}} 0$.
- Asymptotically normal for a fixed t: $\sqrt{n}\left(\hat{S}\left(t\right) - S\left(t\right)\right) \xrightarrow[n\to\infty]{\mathcal{L}} \mathcal{N}\left(0, V^2\left(t\right)\right)$ with $V^2\left(t\right) = -S^2\left(t_0\right)\int_0^t \frac{S(du)}{S^2(u)G(u)}$.

To derive this asymptotic distribution, we need to assume that the proportion of the sample at risk at time t becomes stable as the sample size increases (*Survival and Event History Analysis: A Process Point of View* 2008) and that the censoring is not informative (independent).

The Restricted Mean Survival Time ($\int_0^\tau S(t)dt$) derived by an unadjusted Kaplan-Meier is proved to be:

- Almost surely consistent estimator of the true RMST

3.1.2.2 Implementation

For reasons of interpretability and transparency, we implement an Unadjusted Kaplan-Meier estimator and apply the rectangle integration method (detailed of this method in Section 9) between 0 and τ of the two survival curves (control and treated) to obtain θ_{RMST} (as presented in Figure 1). However, it is also possible to use existing packages to perform these calculations. For example, one can simulate a Kaplan-Meier curve using the survfit() function from the survival package (Therneau 2001) and then use the rmean() function to derive the restricted mean survival time for each survival curve by rectangle integration method also, and subsequently calculating the difference. These methods are equivalent and yield to the same results.

The following code includes several functions :

- Integral_rectangles computes the integral of a given function by the method of rectangle integration
- Kaplan_meier_handmade enables to compute the Kaplan-Meier estimator for both treated and control group.
- RMST_1 computes θ_{RMST} by using the two previous function.

```r
# Function to calculate the integral of a decreasing function using
# the rectangle method
# x corresponds to the x coordinate of the function to integrate
# y corresponds to the y coordinate
integral_rectangles <- function(x, y) {
  # Check if the lengths of x and y are the same
  if (length(x) != length(y)) {
    stop("Lengths of x and y must be the same")
  }

  # Calculate the width of each rectangle
  dx <- diff(x)

  # Initialize the sum
  integral_sum <- 0
  # Iterate through each rectangle and sum up the areas
  for (i in 1:(length(x) - 1)) {
    # Calculate the height of the current rectangle
    height <- min(y[i], y[i + 1])

    # Multiply the height by the width and add it to the sum
    integral_sum <- integral_sum + height * dx[i]
  }
  mean <- integral_sum + x[1]
  # Return the final integral sum
  return(mean)
}

# Kaplan-Meier estimator handmade implementation
# The database 'data' must be in the same form as that shown in
# notation (Table 1) and with the same variable name (status, T_obs)
Kaplan_meier_handmade <- function(data,
                                  status = data$status,
                                  T_obs = data$T_obs) {
  # Sort unique observed times
  Y.grid <- sort(unique(T_obs))

  # Initialize vectors for number of events, number at risk, and survival
  # probability
  d <- rep(NA, length(Y.grid))  # Number of events at time Y.grid[i]
  n <- rep(NA, length(Y.grid))  # Number at risk just before time Y.grid[i]
  S <- rep(NA, length(Y.grid))  # Survival probability at time Y.grid[i]

  # Loop over each unique observed time
  for (i in 1:length(Y.grid)) {
    d[i] <- sum(T_obs == Y.grid[i] & status == 1, na.rm = TRUE)  # Count events
    n[i] <- sum(T_obs >= Y.grid[i])  # Count at risk

    # Calculate survival probability
    S[i] <- cumprod(1 - d / n)[i]
  }

  # Create a data frame with the results
  df <- data.frame(d = d, n = n, S = S, T = Y.grid)

  return(df)
}

# Function to calculate RMST (Restricted Mean Survival Time):
# Method 1: Handmade KM with no truncation

# Two possibilities of computing RMST :
# - in using directly S_A1 and S_A0 (survival function of treated and control)
# - in using the dataframe and the function computes the survival functions
RMST_1 <- function(data = NULL, A1 = 1, A0 = 0, tau, S_A1 = NULL, S_A0 = NULL) {
  if (is.null(S_A1) & is.null(S_A0)) {
    # Subset data for treatment groups
    data1 <- data[data$A == A1,]
    data0 <- data[data$A == A0,]

    # Calculate Kaplan-Meier survival estimates
    S_A1 <- Kaplan_meier_handmade(data1, status = data1$status,
                                  T_obs = data1$T_obs)
    S_A0 <- Kaplan_meier_handmade(data0, status = data0$status,
                                  T_obs = data0$T_obs)

    # Restrict observations to those less than or equal to tau
    Y.grid1 <- data1$T_obs[data1$T_obs <= tau]
    Y.grid0 <- data0$T_obs[data0$T_obs <= tau]
  } else {
    # Restrict observations to those less than or equal to tau
    Y.grid1 <- S_A1$T[S_A1$T <= tau]
    Y.grid0 <- S_A0$T[S_A0$T <= tau]
  }

  # Filter survival estimates to restricted observations
  S_A1 <- S_A1 %>%
    dplyr::filter(T %in% Y.grid1)
  S_A0 <- S_A0 %>%
    dplyr::filter(T %in% Y.grid0)

  # Check if there is any event at tau for S_A1
  if (!any(S_A1$T == tau)) {
    new_row <- tibble(T = tau, S = S_A1$S[nrow(S_A1)])
    S_A1 <- dplyr::bind_rows(S_A1, new_row)
  }

  # Check if there is any event at tau for S_A0
  if (!any(S_A0$T == tau)) {
    new_row <- tibble(T = tau, S = S_A0$S[nrow(S_A0)])
    S_A0 <- dplyr::bind_rows(S_A0, new_row)
  }

  # Calculate integrals from 0 to tau of survival probabilities
  intA1 <- integral_rectangles(S_A1$T, S_A1$S)
  intA0 <- integral_rectangles(S_A0$T, S_A0$S)
  RMST1 <- intA1 - intA0

  return(list(RMST=RMST1, intA1=intA1,intA0=intA0))
}
```

It is important to adapt the integration method to the type of function being integrated. In our case, since the Kaplan-Meier estimator is a step function, the most appropriate integration method is the rectangle method. However, if the survival function is derived from a parametric model (such as the Weibull or Exponential model), it is preferable to use a more precise method, such as the trapezoidal method (details of this method can be found in Section 9) to derive θ_{RMST}.

When the censoring becomes conditionally independent on covariates, the previous estimators are not sufficient as it considers only independent censoring. The next section will introduce other strategies to overcome this issue.

3.2 Conditional independent censoring

Under Equation 5 (random treatment assignment) and Equation 3 (conditional independent censoring), a strategy to make the identification of the θ_{RMST} possible can be to use a censoring unbiased transformation. A notable advantage of this transformation is that it enables the use of causal inference estimators in the same way as with fully observed data.

3.2.1 Notion of censoring unbiased transformation

This notion has been introduced by Fan and Gijbels (1994a) for local polynomial modelling adapted to regression problems. The objective of this transformation is to adjust for the censoring effect by transforming the data in an unbiased way. In other word, it creates an estimated unbiased fully observed population.

The method consists in transforming the observed data mentioned in Section 2 : $O_i = (X_i, A_i, \Delta_i, \tilde{T}_i)$ into (X_i, A_i, T_i^*) according to:

$$T^* = \Delta\phi_1(\mathbf{X}, A, \tilde{T}) + (1 - \Delta)\phi_2(\mathbf{X}, A, \tilde{T}) \tag{8}$$

with $\phi_1(.,.) =$ and $\phi_2(.,.)$ the transformation functions on respectively uncensored and censored observations.
The basic requirement is that $E(T^*|X, A) = E(T \wedge \tau|X, A)$. It allows to create a population that would have been observed if there was no censoring. Several transformations has been proposed in the literature such as the Buckley-James transformation (Buckley and James 1979) (one of the earliest transformation detailed in Section 9) or the inverse probability of censoring transformation (IPC) firstly introduced by Koul, Susarla, and Ryzin (1981) for right-censored regression.

In the following section, we begin by considering the IPC (inverse probability of censoring) transformation, then the Buckley-James transformation. After defining these transformations, we present the identifiability formula using IPC transformation for θ_{RMST} and finally an estimation strategy. Then, identifiability formula and estimator in using Buckley-James transformation.

3.2.1.1 The inverse probability of censoring transformation (IPC transformation)

The inverse probability censoring weighting approach has been introduced in survival analysis by Koul, Susarla, and Ryzin (1981) to overcome bias due to conditionally independent censoring.

The IPC transformation is given by :

$$T^*(\tau) = \frac{\widetilde{T} \wedge \tau * \Delta^\tau}{S_c(\widetilde{T} \wedge \tau|X, A)}$$

where $S_c(T \wedge \tau|X, A = a)$ is the survival function of remain uncensored truncated at τ given the covariate X in the treatment arm $A = a$ and $\Delta^\tau = I\{T \wedge \tau < C\}$ is the status of the individual truncated at τ.

Considering Equation 8, this transformation corresponds to the case where $\phi_1 = \frac{\widetilde{T}\wedge\tau}{S_c(\widetilde{T}\wedge\tau|X,A)}$ and $\phi_2 = 0$. It does not consider censored observations.

This transformation can be also identified by the following formula:

$$
\begin{aligned}
E\left[T\wedge\tau \mid A,X\right] &= E\left[E\left[1\{T\wedge\tau < C\} \mid A,X,T\right]\cdot\frac{T\wedge\tau}{S_C(\widetilde{T}\wedge\tau \mid A,X)}\middle| A,X\right] \quad (1)\\
&\text{(In color, the terms are equal)}\\
&= E\left[E\left[\Delta^\tau \mid A,X,T\right]\cdot\frac{T\wedge\tau}{S_C(\widetilde{T}\wedge\tau \mid A,X)}\middle| A,X\right] \quad (2)\\
&= E\left[E\left[\Delta^\tau \mid A,X,T\right]\cdot E\left[\frac{T\wedge\tau}{S_C(\widetilde{T}\wedge\tau \mid A,X)}\middle| A,X,T\right]\middle| A,X\right] \quad (3)\\
&= E\left[E\left[\Delta^\tau\cdot\frac{T\wedge\tau}{S_C(\widetilde{T}\wedge\tau \mid A,X)}\middle| A,X,T\right]\middle| A,X\right] \text{ (By conditional censoring As.3)} \quad (4)\\
&= E\left[\frac{\Delta^\tau\cdot T\wedge\tau}{S_C(\widetilde{T}\wedge\tau \mid A,X)}\middle| A,X\right] \text{ (Law of total probability)} \quad (5)\\
&= E\left[\frac{\Delta^\tau\cdot \widetilde{T}\wedge\tau}{S_C(\widetilde{T}\wedge\tau \mid A,X)}\middle| A,X\right] \; (\Delta^\tau\cdot T\wedge\tau = \Delta^\tau\cdot\widetilde{T}\wedge\tau) \quad (6)\\
&= \mathbb{E}[T^*|A,X] \quad (7)
\end{aligned}
\tag{9}
$$

The term in color are equal because $E\left[1\{T\wedge\tau < C\} \mid A,X,T\right] = E\left[1\{\widetilde{T}\wedge\tau < C\} \mid A,X,T\right] = S_C(\widetilde{T}\wedge\tau \mid A,X)$. Also, the equality in the line 6 of Equation 9 is easily proven by the fact that Δ^τ, the indicator of censoring, selects observations which declare the event:

$$
\begin{aligned}
(\widetilde{T}\wedge\tau) * \Delta^\tau &= (\widetilde{T}\wedge\tau) * I\{T\wedge\tau < C|A=0\}\\
&= min(T\wedge\tau, C) * I\{T\wedge\tau < C|A=0\}\\
&= (T\wedge\tau) * I\{T\wedge\tau < C|A=0\}
\end{aligned}
\tag{10}
$$

The particularity of this transformation is that it doesn't take into account the censored observation by weighting only the uncensored observations. It exits other censoring unbiased transformation such as Buckley James that consider also the censored observations.

3.2.1.2 Buckley James unbiased censoring transformation

This transformation has been introduced by Buckley and James (1979) (BJ). This is the earliest unbiased censoring transformation:

$$
\begin{aligned}
T^*(O,\tau) &= \Delta^\tau * (\widetilde{T}\wedge\tau) + (1-\Delta^\tau) * \mathbb{E}[T\wedge\tau|X,A,T\wedge\tau > \widetilde{T}\wedge\tau]\\
&= \Delta^\tau * (\widetilde{T}\wedge\tau) + (1-\Delta^\tau) * Q_S(C|X,A)
\end{aligned}
$$

with $Q_S(t|x,a) = E[T\wedge\tau|X=x, A=a, T\wedge\tau > t]$

This transformation uses the uncensored observed values of the time-to-event $\widetilde{T} \wedge \tau$ directly in the formula (as uncensored observations are fully complete) while the censored values of the time-to-event outcome $\widetilde{T} \wedge \tau$ are extrapolated from an estimator $Q_S(t|x,a)$ which corresponds to the expected remaining survival time given the covariates and treatment for censored observation.

The function Q_S can be expressed in an other way:

$$\begin{aligned}
Q_S(t|x,a) &= E[T \wedge \tau \mid X = x, A = a, T \wedge \tau > t] \\
&= \frac{E[T \wedge \tau . I\{T \wedge \tau > t\} \mid X = x, A = a]}{P(T \wedge \tau > t|X, A)} \qquad \text{(by the law of conditional expectation)} \\
&= \int_{-\infty}^{+\infty} \frac{t.I\{T \wedge \tau > t\}.dF(T \wedge \tau|X, A)}{P(T \wedge \tau > t|X, A)} \qquad \text{(by def of Riemann–Stieltjes integral)} \\
&= \frac{1}{P(T \wedge \tau > t|X, A)} \int_{t}^{+\infty} t.dF(T \wedge \tau|X, A) \\
&= \frac{1}{S(T \wedge \tau|X, A)} \int_{t}^{+\infty} t.dF(T \wedge \tau|X, A) \\
Q_S(t|x,a) &= \frac{1}{S(T \wedge \tau|X = x, A = a)} \int_{t}^{+\infty} t.d(1 - S(T \wedge \tau|X = x, A = a))
\end{aligned}$$

with F the function of the cumulative probability ($F(t) = \mathbb{P}(t \leq T)$) and $\bar{F}$ the complementary cumulative probability ($S(t) = 1 - F(t)$).

Considering Equation 8, this transformation can be expressed as:

$$T^* = \begin{cases} \phi_1(X, A, \widetilde{T} \wedge \tau) = \tilde{T} \wedge \tau \quad \text{if} \quad \Delta^\tau = 1 \\ \phi_2(X, A, \widetilde{T} \wedge \tau) = \mathbb{E}[T \wedge \tau|X, A, T \wedge \tau \geq \tilde{T} \wedge \tau] \quad \text{else } \Delta^\tau = 0 \end{cases}$$

Exactly than the inverse probability of censoring transformation, the Buckley James transformation verify $E(T^*|X, A) = E(T \wedge \tau|X, A)$ (proof in Section 9).

Unfortunately, the two previous transformation depends on nuisance parameters:

- the censoring distribution $S_c(t|x, a)$ for the IPC weighting transformation.
- the conditional survival distribution $Q_S(t|x, a)$ for the Buckley-James transformation.

If this nuisance parameters are not well estimated, the estimator using the transformation will be biased (Fan and Gijbels 1994b).

3.2.2 Identifiability using IPC transformation

Thus, under Equation 5 (random treatment assignment) and Equation 3 (conditionally independent censoring), the θ_{RMST} can be identified as follows:

$$
\begin{aligned}
\theta_{\text{RMST}} &= \mathbb{E}\left[T(1)\wedge\tau - T(0)\wedge\tau\right] \quad (1)\\
&= \mathbb{E}\left[T(1)\wedge\tau\right] - \mathbb{E}\left[T(0)\wedge\tau\right]\\
&= \mathbb{E}\left[\mathbb{E}\left[T(1)\wedge\tau \mid A=1, X\right]\right] - \mathbb{E}\left[\mathbb{E}\left[T(0)\wedge\tau \mid A=0, X\right]\right] \quad (2)\\
&\text{(Law of total probability and Ignorability)}\\
&= \mathbb{E}\left[\mathbb{E}\left[T^*(1) \mid A=1, X\right]\right] - \mathbb{E}\left[\mathbb{E}\left[T^*(0) \mid A=0, X\right]\right] \quad (3)\\
&\text{(IPC transformation)}\\
&= \mathbb{E}\left[T^*(1) \mid A=1\right] - \mathbb{E}\left[T^*(0) \mid A=0\right] \quad (4)\\
&\text{(Law of total probability)}\\
&= \mathbb{E}\left[\frac{\widetilde{T}(1)\wedge\tau\cdot\Delta^\tau}{S_c(T\wedge\tau \mid A=1, X)} \mid A=1\right] - \mathbb{E}\left[\frac{\widetilde{T}\wedge\tau\cdot\Delta^\tau}{S_c(T\wedge\tau \mid A=0, X)} \mid A=0\right] \quad (5)\\
&= \int_0^\tau \mathbb{E}\left[\frac{1\{T(1)\geq t\}\cdot\Delta^\tau}{S_c(T\wedge\tau \mid A=1, X)} \mid A=1\right] - \mathbb{E}\left[\frac{1\{T(0)\geq t\}\cdot\Delta^\tau}{S_c(T\wedge\tau \mid A=0, X)} \mid A=0\right] dt \quad (6)
\end{aligned}
\tag{11}
$$

The IPC transformation used in this identifiability equation has been detailed in the Section 3.2.1.1.

3.2.2.1 Estimation with IPCW Kaplan Meier

Under Assumptions Equation 5 (random treatment assignment), Equation 3 (conditional censoring) and Equation 4 (Positivity for censoring), the Unadjusted KM estimator in Definition 3.1 misjudges the real survival probabilities (Willems et al. 2018). Correction for the presence of conditionally independent censoring is a necessity. The adjusted IPCW (inverse probability of censoring weighting) Kaplan Meier estimator (J. M. Robins and Rotnitzky 1992; J. M. Robins and Finkelstein 2000) can be used to estimate the causal treatment effect represented in the identifiability Equation 11 by applying the weight $\frac{\Delta^\tau}{S_c(T\wedge\tau|A=1,X)}$ on all observations into Kaplan-Meier estimator:

Definition 3.2 (IPCW adjusted Kaplan Meier estimator)**.**

$$
\hat{S}_{IPCW-KM}(t \mid A=a) = \prod_{j=1, t_j <= t}\left(1 - \frac{\sum_i \hat{w}_i(t_j, X_i) * I\left\{T_i = t_j, C_i \geq t_j, A_i = a\right\}}{\sum_k \hat{w}_k(t_j, X_k) * I\left\{T_k \geq t_j, C_k \geq t_j, A_k = a\right\}}\right)
$$

- $\hat{w}_i(t, X_i) = \frac{\Delta_i^\tau}{\hat{S}_c(t|X_i, A_i)}$ is the inverse of the probability of remain uncensored given the covariates X_i.
- $\Delta^\tau = I\{T\wedge\tau < C\} = I\{\widetilde{T}\geq\tau\} + I\{\widetilde{T}\leq\tau\}.\Delta$ (proof in Section 9).
- $\hat{S}_c(t|X_i, A_i)$ is based on the fit of parametric, semi-parametric (for example a Cox model) or even non-parametric model (such as survival forest) for censoring with X_i and A_i the covariates.

The probability of remaining uncensored depends on the covariates and the treatment, so subjects with the same covariates have the same probability of remaining uncensored. Thus, this estimator give extra weight to subjects who are not censored in the same group of subject. It compensates the conditionally independent censoring and offsets the selection bias induced by this dependency (Howe et al. 2016).

At every time point t, each subject i is given a weight which is inversely proportional to the estimated probability of having remained uncensored until time t.

In the exact same way than before in Section 3.1.2, the corresponding θ_{RMST} is obtained in integrating from 0 to τ the difference between adjusted Kaplan Meier estimator of the treated and controls (Equation 12) :

$$\hat{\theta}_{RMST} = \int_0^\tau \hat{S}_{IPCW-KM}(t, A = 1) - \hat{S}_{IPCW-KM}(t, A = 0)dt \tag{12}$$

3.2.2.1.1 Properties of IPCW Kaplan Meier

Robins (1993) (fameuse publi introuvable) shows that our IPCW estimate $\hat{S}_{IPCW-KM}(t, a)$ is guaranteed to be consistent and asymptotically normal under Equation 3 and the model of conditional censoring is correct (J. Robins, Rotnitzky, and Bonetti 2004). Also, J. M. Robins and Finkelstein (2000) shows that IPCW estimate is asymptotically more efficient than the standard Kaplan-Meier estimator for failure in treatment arm a whenever the latter estimator is consistent (i.e., whenever censoring is independent).

3.2.2.1.2 Implementation

The following code includes several functions :

- Adjusted.KM allows to compute the adjusted IPCW survival curve for treated and control (stratification on treatment) from given weights, times and events.
- Estimate_survival_function enables to compute the probability of remain uncensored over the time, S_c in using cox model or survival forest (survival_forest function from grf (Tibshirani et al. 2017)). It allows cross-fitting for survival forest (n.folds>1).
- IPCW_Kaplan_meier computes IPCW KM estimator by using the previous functions.

```r
# Kaplan-Meier adjusted
# Times of event
# Failures:  1 if event, 0 if censored
# Variable:  1 if treated, 0 if control
# Weights:  weight of the individual
adjusted.KM <- function(times, failures, variable, weights = NULL) {
  # Sanity checks
  if (sum(times < 0) > 0) {
    stop("Error: times must be positive")
  }
  if (!is.null(weights) && sum(weights < 0) > 0) {
    stop("Error: weights must be superior to 0")
  }
  if (sum(failures != 0 & failures != 1) > 0) {
    stop("Error: failures must be a vector of 0 or 1")
  }
  # If 'weights' is NULL, initialize 'w' with ones of the same length as 'times',
  # otherwise use 'weights'
  w <- if (is.null(weights)) rep(1, length(times)) else weights

  # Create a DataFrame 'data' with columns t (times), f (failures),
  # v (stratification variable: often treatment variable), and w (weights)
  data <- data.frame(t = times, f = failures, v = variable, w = w)

  # Remove rows from the DataFrame where the stratification variable is NA
  data <- data[!is.na(data$v),]
  # Initialize an empty DataFrame to store the Kaplan-Meier results
  table_KM <- data.frame(times = NULL, n.risk = NULL, n.event = NULL,
                         survival = NULL, variable = NULL)

  # Loop over each unique value of the stratification variable
  for (i in unique(variable)) {
    # Subset the data for the current stratification variable value
    d <- data[data$v == i,]

    # Create a sorted vector of unique event times, including time 0 and the
    # maximum time
    tj <- c(0, sort(unique(d$t[d$f == 1])), max(d$t))

    # Calculate the number of events at each time point
    dj <- sapply(tj, function(x) {
      sum(d$w[d$t == x & d$f == 1])
    })

    # Calculate the number of individuals at risk at each time point
    nj <- sapply(tj, function(x) {
      sum(d$w[d$t >= x])
    })

    # Compute the cumulative product for the survival probabilities
    st <- cumprod((nj - dj) / nj)

    # Append the results to the Kaplan-Meier table
    table_KM <- rbind(table_KM, data.frame(T = tj, n = nj, d = dj,
                                           S = st, variable = i))
  }
  return(table_KM)
}


# Estimate survival function with covariates for each individual at each time Y.grid
# Type of model can be cox or survival forest (n.fold must be completed in this case)

# This function is used also to compute S_c with status = censor.status
estimate_survival_function <- function(data, X.names,
                                       Y.grid,
                                       type_of_model = "cox",
                                       T_obs = "T_obs",
                                       status = "status",
                                       n.folds = NULL) {
  if (type_of_model == "cox") {
    # Formula for cox model (single learner: A as a covariate,
    # T-learner: Stratified fit on A)
    # Here only T-learner
    outcome <- paste0('Surv(', T_obs, ',', status, ')')
    formula <- as.formula(paste(outcome, paste(c(X.names),
                                               collapse = " + "), sep = " ~ "))
    data.1 <- data%>%
      filter(A==1)

    data.0 <- data%>%
      filter(A==0)

    # Cox model fitting stratified on A=1
    fitS1 <- suppressWarnings(coxph(formula, data = data.1, x = TRUE))
    # Suppress NA in coefficient
    fitS1$coefficients[is.na(fitS1$coefficients)] <- 0

    # Cox model fitting on A=0
    fitS0 <- suppressWarnings(coxph(formula, data = data.0, x = TRUE))
    # Suppress NA in coefficient
    fitS0$coefficients[is.na(fitS0$coefficients)] <- 0

    fit.pred1 <- predictCox(fitS1, newdata = data, times = Y.grid,
                            type = "survival")
    fit.pred0 <- predictCox(fitS0, newdata = data, times = Y.grid,
                            type = "survival")
    S_hat1 <- fit.pred1$survival
    S_hat0 <- fit.pred0$survival

  } else { # Survival forest
    # Initialization
    n <- nrow(data)
    fit.pred1 <- matrix(NA, nrow = n, ncol = length(Y.grid))
    fit.pred0 <- matrix(NA, nrow = n, ncol = length(Y.grid))

    if (n.folds > 1) {# Cross-fitting
      # Split the dataset into n-folds
      indices <- split(seq(n), sort(seq(n) %% n.folds))

      # For each index in each split
      for (idx in indices) {
        # Fit survival forest on observations removed from idx (training set) and A=1
        # A is not included in covariates (T-learner)
        forest.grf1 <- survival_forest(X = as.matrix(data[-idx & data[,"A"] == 1,
                                                          X.names]),
                                       Y = data[-idx & data[,"A"] == 1, T_obs],
                                       D = data[-idx & data[,"A"] == 1, status],
                                       failure.times = Y.grid)
        # Fit survival forest on observations removed from idx (training set) and A=0
        # A is not included in covariates (T-learner)
        forest.grf0 <- survival_forest(X =  as.matrix(data[-idx & data[,"A"] == 0,
                                                           X.names]),
                                       Y = data[-idx & data[,"A"] == 0, T_obs],
                                       D = data[-idx & data[,"A"] == 0, status],
                                       failure.times = Y.grid)
        # Prediction on idx to avoid overfitting
        fit.pred1[idx,] <- predict(forest.grf1,  as.matrix(data[idx, X.names]),
                                   failure.times = Y.grid)$predictions

        fit.pred0[idx,] <- predict(forest.grf0,  as.matrix(data[idx, X.names]),
                                   failure.times = Y.grid)$predictions
      }
    } else {# No cross-fitting
      # Fit survival forest on all observations with A=1
      # A is not included in covariates (T-learner)
      forest.grf1 <- survival_forest(X =  as.matrix(data[data[,"A"] == 1,
                                                         X.names]),
                                     Y = data[data[,"A"] == 1, T_obs],
                                     D = data[data[,"A"] == 1, status],
                                     failure.times = Y.grid)

      # Fit survival forest on all observations with A=0
      # A is not included in covariates (T-learner)
      forest.grf0 <- survival_forest(X =  as.matrix(data[data[,"A"] == 0,
                                                         X.names]),
                                     Y = data[data[,"A"] == 0, T_obs],
                                     D = data[data[,"A"] == 0, status],
                                     failure.times = Y.grid)

      # Prediction on all observations
      fit.pred1 <- predict(forest.grf1,  as.matrix(data[, X.names]),
                           failure.times = Y.grid)$predictions
      fit.pred0 <- predict(forest.grf0,  as.matrix(data[, X.names]),
                           failure.times = Y.grid)$predictions
    }

    S_hat1 <- fit.pred1
    S_hat0 <- fit.pred0
  }

  # Associate the corresponding Survival curve to the observation
  S_hat <- S_hat1 * data$A + (1 - data$A) * S_hat0

  return(list('S_hat' = S_hat, "S_hat1" = S_hat1, "S_hat0" = S_hat0, "T" = Y.grid))
}

# IPCW Kaplan-Meier estimator with restricted tau
IPCW_Kaplan_meier <- function(data, tau,
                              X.names.censoring,
                              nuisance_censoring = "cox",
                              n.folds = NULL) {
```

```
  # Compute of truncated T_obs, status and censored status
  data$T_obs_tau <- ifelse(data$T_obs >= tau, tau, data$T_obs)
  data$censor.status_tau <- 1 - as.numeric((data$T_obs >= tau) |
                                             (data$T_obs < tau & data$status == 1))
  data$status_tau <- as.numeric((data$T_obs >= tau) |
                                  (data$T_obs < tau & data$status == 1))
  Y.grid <- sort(unique(data$T_obs_tau))

  # Estimate probability of remaining uncensored based on nuisance model
  S_C_hat <- estimate_survival_function(data = data, X.names = X.names.censoring,
                                        Y.grid = Y.grid, T_obs = "T_obs_tau",
                                        status = "censor.status_tau",
                                        type_of_model = nuisance_censoring,
                                        n.folds = n.folds)

  # Select the probability of censoring for each observe T_obs_tau from the all
  # curve
  data$S_C <- S_C_hat$S_hat[cbind(1:nrow(data), match(data$T_obs_tau, Y.grid))]

  # Compute IPC weights
  data$weights <- data$status_tau / data$S_C

  # Compute the adjusted IPCW Kaplan Meier
  S <- adjusted.KM(times = data$T_obs, failures = data$status,
                   variable = data$A, weights = data$weights)

  # Compute difference in RMST
  RMST <- RMST_1(S_A1 = S[S$variable == 1,], S_A0 = S[S$variable == 0,], tau = tau)

  return(list(RMST = RMST$RMST,
              intA1 = RMST$intA1,
              intA0 = RMST$intA0,
              weights = data$weights))
}
```

It exists other solutions to identify θ_{RMST}. One of them use directly the results using pseudo-observations from the unbiased censoring transformation in Section 3.2.1.1 or Section 3.2.1.2 without using survival function. In the following section, we will introduce the identifiability formulae using Buckley-James transformation known as the best restoration transformation. It will give us another estimator described in Section 3.2.3.1 to estimate θ_{RMST}.

3.2.3 Identifiability using BJ transformation

Under Equation 5 (random treatment assignment) and Equation 3 (conditionally independent censoring), another identifiability formula can be derived from line 4 in Equation 11 in using Buckley-James transformation (Section 3.2.1.2):

$$
\begin{aligned}
\theta_{\text{RMST}} &= \mathbb{E}\left[T(1) \wedge \tau - T(0) \wedge \tau\right] \\
&= \mathbb{E}\left[T^{*}(1) \mid A = 1\right] - \mathbb{E}\left[T^{*}(0) \mid A = 0\right] \\
&= \mathbb{E}\left[\Delta^{\tau} * (\widetilde{T}(1) \wedge \tau) + (1 - \Delta^{\tau}) * Q_S(C|X, A) \mid A = 1\right] - \\
&\qquad \mathbb{E}\left[\Delta^{\tau} * (\widetilde{T}(0) \wedge \tau) + (1 - \Delta^{\tau}) * Q_S(C|X, A) \mid A = 0\right] \\
&\text{(BJ transformation)} \\
&= \mathbb{E}\left[\Delta^{\tau} * (\widetilde{T} \wedge \tau) + (1 - \Delta^{\tau}) * Q_S(C|X, A) \mid A = 1\right] - \\
&\qquad \mathbb{E}\left[\Delta^{\tau} * (\widetilde{T} \wedge \tau) + (1 - \Delta^{\tau}) * Q_S(C|X, A) \mid A = 0\right] \\
&\text{(By consistency)}
\end{aligned}
\tag{13}
$$

The straightforward corresponding estimator from Equation 13 is described below.

3.2.3.1 Estimation of BJ estimator

Based on the identifiability formula Equation 13, it is possible to implement BJ estimator directly without using survival function:

Definition 3.3 (BJ estimator)**.**

$$
\begin{aligned}
\theta_{RMST} = \frac{1}{n_1} * \sum_{i=1}^{n_1} &\left[\Delta_i^{\tau} * (\widetilde{T}_i \wedge \tau) + (1 - \Delta_i^{\tau}) * \hat{Q}_S(C_i|X, A) \mid A = 1\right] - \\
&\frac{1}{n_0} * \sum_{j=1}^{n_0} \left[\Delta_j^{\tau} * (\widetilde{T}_j \wedge \tau) + (1 - \Delta_j^{\tau}) * \hat{Q}_S(C_j|X, A) \mid A = 0\right]
\end{aligned}
$$

with n_1 corresponds to the number of observations in the treated group, n_0 corresponds to the number of observations in the control group and $\hat{Q}_S(\widetilde{T}\wedge\tau \mid X, A) = \frac{1}{\hat{S}(\widetilde{T}\wedge\tau|X,A)} \int_{\widetilde{T}\wedge\tau}^{+\infty} \widetilde{T} \wedge \tau . d\hat{F}(\widetilde{T} \wedge \tau \mid X, A)$ the estimation function of the remaining survival function ($E[\widetilde{T} \wedge \tau \mid X, A, \widetilde{T} \wedge \tau > t]$)

This estimator is the illustration that θ_{RMST} can be computed directly without integrate the restricted survival function. It does the average of the RMST using fully complete pseudo-observation from Buckley-James transformation of each group. This estimator behaves as if there is no censoring.

3.2.3.1.1 Properties of Buckley-James estimator

BJ transformation is considered as the best predictor of the original response in the sens that $\mathbb{E}(T - T^{*}(O))^2 \leq \mathbb{E}(T - T^{*})^2$ among all censoring unbiased transformation. It can be regarded as the best restoration (Fan and Gijbels 1994b). But, as presented in Section 3.2.1.2, the properties of BJ estimator is dependent on the model specification of Q_s. If the model specification is correct, then BJ transformation is considered as the best predictor of the original response among all censoring unbiased transformation.

3.2.3.1.2 Implementation

The following code includes several functions :

- Q_t_hat computes $Q_s(t|x, a)$ for each timepoint and individuals which uses the previously implemented function estimate_survival_function.
- Q_Y compute specifically $Q_s(T_i \wedge \tau|X_i, A_i)$.

- BJ computes the θ_{RMST} by implementing the Buckley-James estimator and in using the previous functions.

The previous estimators are suitable for RCT settings but not for more complex contexts such as observational studies. The next section will introduce more sophisticated estimators to measure the Average Treatment Effect (ATE) in the context of observational studies.

4 Causal survival analysis with an observational study

In contrast to RCT, observational data (for example data from registries, electronic health reports, national health data system…) are collected without interventions on the treatment allocation. The Figure 3 presents an illustration of a simple causal graph in observational survival data without censoring to show that causal effect is no longer easily identifiable.

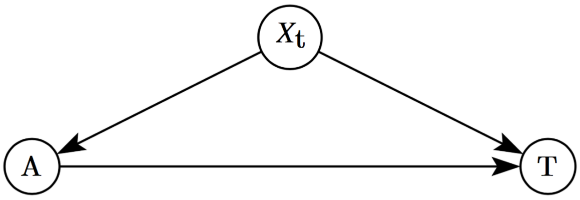


Figure 3: Illustration of a simple causal graph in observational survival data without censoring (A is the treatment, X_t the confounding variable and T is the time to event outcome)

In this context, treated and control group can be unbalanced due to the lack of controlled design. As a consequence, causal effect of treatment is obscured by the effect of confounding variables X_t that have both impact on time to event outcome T and treatment allocation A.

The assumption of Equation 5 (randomized treatment assignment) presented in Section 3 is no longer verified. To enable the identifiability of causal estimand, some additional assumptions on the treatment allocation are required. These assumptions are classical for causal inference with observational data and can be extended to identify θ_{RMST}:

Assumption 5: Conditional exchangeability / Uncounfoundedness

$$A \perp\!\!\!\perp (T(0), T(1)) | X_t \tag{14}$$

with X_t the set of covariates that are related both to treatment's assignment and outcomes.

Under Assumption Equation 14, the treatment assignment is randomly assigned conditionally on the covariates X. It is as if the treatment for all subjects were randomly selected inside each subgroup. Also, this assumption assumes that there are no unmeasured confounders as unobserved confounders make it impossible to separate correlation and causality.

Assumption 6: Positivity / Overlap for treatment

$$1 > P(A = a \mid X_t = x) > 0 \tag{15}$$

The Equation 15 requires adequate overlap of the covariates distribution between group of treatment. This means that all observations in the study have non-zero probability of being treated.

In addition to the confounding bias, the censoring bias has to be considered as seen in Section 3. Therefore, the assumptions regarding the censoring mechanism discussed in Section 2.3.1 remain applicable. In the following section, we will present the identifiability formula, the corresponding estimator and implementation when censoring is independent in an observational study. Following this, we will present several identifiability formulae when censoring is conditionally dependent on X_c, along with their estimators and implementations.

In the following section, we will consider that the variables $X_t = X_c = X$.

4.1 Independent censoring

4.1.1 Identifiability

Under Equation 14 (Uncounfoundedness) and Equation 2 (independent censoring), the RMST can be identified as follows:

$$
\begin{aligned}
\theta &= \mathbb{E}[T(1) \wedge \tau - T(0) \wedge \tau] \\
&= \int_0^\tau \mathbb{E}[I\{T(1) > t\}] - \mathbb{E}[I\{T(0) > t\}]dt \\
&\text{(By linearity)} \\
&= \int_0^\tau \mathbb{E}\left[\mathbb{E}[I\{T(1) > t\}|X]\right] - \mathbb{E}\left[\mathbb{E}[I\{T(0) > t\}|X]\right] \\
&\text{(Law of total probability and Ignorability)} \\
&= \int_0^\tau \mathbb{E}\left[\frac{\mathbb{E}[I\{T(1) > t|X\}] * {\color{blue}\mathbb{E}[A|X\}]}}{{\color{blue}e(X)}}\right] - \mathbb{E}\left[\frac{\mathbb{E}[I\{T(0) > t|X\}] * {\color{red}\mathbb{E}[1 - A|X\}]}}{{\color{red}1 - e(X)}}\right] dt \\
&\text{(In color, the terms are equal)} \\
&= \int_0^\tau \mathbb{E}\left[\frac{\mathbb{E}[I\{T(1) > t\} * A|X\}}{e(X)}\right] - \mathbb{E}\left[\frac{\mathbb{E}[I\{T(0) > t\} * (1 - A)|X]]}{1 - e(X)}\right] dt \qquad (16) \\
&\text{(By unconfoundedness)} \\
&= \int_0^\tau \mathbb{E}\left[\frac{I\{T(1) > t\} * A}{e(X)}\right] - \mathbb{E}\left[\frac{\mathbb{E}[I\{T(0) > t\} * (1 - A)}{1 - e(X)}\right] dt \\
&\text{(Law of total probability)} \\
&= \int_0^\tau \mathbb{E}\left[\frac{I\{T(1) > t|A = 1\} * A}{e(X)}\right] - \mathbb{E}\left[\frac{I\{T(0) > t|A = 0\} * (1 - A)}{1 - e(X)}\right] dt \\
&= \int_0^\tau \mathbb{E}\left[\frac{I\{T > t|A = 1\} * A}{e(X)}\right] - \mathbb{E}\left[\frac{I\{T > t|A = 0\} * (1 - A)}{1 - e(X)}\right] dt \\
&\text{(By consistency)}
\end{aligned}
$$

This inverse probability treatment weighting approach is similar to the method for IPW estimator used in causal inference to correct the bias due to some confounding variables. The use of propensity score in causal inference has been introduced by Rosenbaum and Rubin (1983) and extended to survival analysis by Xie and Liu (2005).

4.1.1.1 Estimation with IPTW Kaplan Meier

Under Equation 14 (Uncounfoundedness) and Equation 2 (independent censoring), an adjusted Kaplan Meier estimator is easily derived from the identifiability Equation 16. This estimator includes a weighting term to take into account that the treated and control groups are unbalanced. This weighted estimator is called the inverse probability of treatment weighted Kaplan Meier estimator (IPTW-KM) (Xie and Liu 2005):

Definition 4.1 (IPTW Kaplan Meier estimator).

$$\hat{S}_{IPTW-KM}(t \mid A=a) = \prod_{j=1, t_j<=t} \left(1 - \frac{\sum_i \hat{w}_i * I\left\{T_i = t_j, C_i \geq t_j, A_i = a\right\}}{\sum_k \hat{w}_k * I\left\{T_k \geq t_j, C_k \geq t_j, A_k = a\right\}}\right) \tag{17}$$

with:

- $\hat{w}_i = \frac{A_i}{\hat{e}(X_i)} + \frac{1-A_i}{1-\hat{e}(X_i)}$ the inverse of the propensity score.
- $e(X_i) = P(A_i = 1|X_i)$ is the propensity score.

In the exact same way than before, the corresponding θ_{RMST} is obtained in integrating from 0 to τ the difference between IPTW adjusted Kaplan Meier estimator of the treated and controls.

4.1.1.1.1 Properties of IPTW Kaplan Meier estimator

When the propensity score is known, the estimator IPTW KM is a consistent estimate and is the maximum pseudo-likelihood estimate. Its variance when $A = 1$ is estimated by:

$$\operatorname{Var}\left[\hat{S}^1(t)\right] = \left(S^1(t)\right)^2 \sum_{j:t_j \leq t} \frac{1-s_j^1}{M_j s_j^1}$$

where $M_j = \left(\sum_{i:T_i \geq t_j} \frac{1}{e_i}\right)^2 / \sum_{i:T_i \geq t_j} \left(\frac{1}{e_i}\right)^2$

When the propensity score is unknown but estimated given X, A in using parametric or non-parametric model, then the corresponding variance is estimated by:

$$\operatorname{Var}\left[\hat{S}^1(t)\right] = \left(S^1(t)\right)^2 \sum_{j:t_j \leq t} \frac{1-\hat{s_j}^1}{\hat{M_j} \hat{s_j}^1}$$

Thus, the estimator is consistent if the propensity estimators uniformly converges to the true probability ($\max_{i:T_i \geqslant t_j} (1/\hat{e}_i) / \sum_{i:T_i \geqslant t_j} 1/\hat{e}_i \underset{n\to\infty}{\to} 0$) (Xie and Liu 2005).

4.1.1.1.2 Implementation

The following code includes several functions :

- Estimate_propensity_score enables to compute the probability of being treated over the time $e(X)$ in using logistic regression or probability forest (probability_forest function from grf (Tibshirani et al. 2017)). It allows cross-fitting for probability forest (n.folds>1).
- IPTW_Kaplan_meier computes the presented estimator and regroups the previous function and adjusted.KM presented in Section 3.2.2.1.2.

```r
# Function to estimate propensity score
estimate_propensity_score <- function(data, treatment_covariates,
                                      type_of_model = "glm", n.folds = NULL) {
  # Generalized Linear Model (GLM)
  if (type_of_model == "glm") {
    outcome <- 'A'
    f <- as.formula(paste(outcome, paste(c(treatment_covariates),
                                         collapse = " + "), sep = " ~ "))
    fitA <- glm(f, data = data, family = binomial(link = "logit"))
    e_hat <- predict(fitA, newdata = data, type = "response")
  }

  # Probability Forest (only for continuous variables,
  # categorical variables need one-hot encoding)
  if (type_of_model == "probability forest" && !is.null(n.folds)) {
    # Initialization
    n <- nrow(data)
    e_hat <- rep(NA, n)
    A <- data$A

    # Cross-fitting to avoid overfitting
    if (n.folds > 1) {
      # Split the dataset into n folds
      indices <- split(seq(n), sort(seq(n) %% n.folds))

      # Learn and predict for each fold
      for (idx in indices) {

        # Learn on all data except idx
        propensity_model <- probability_forest(
          as.matrix(data[-idx, treatment_covariates]),
          as.factor(A[-idx]))

        # Predict on idx
        e_hat[idx] <- predict(
          propensity_model,
          newdata = as.matrix(data[idx, treatment_covariates]))$predictions[, 2]
      }
    }
    # No cross-fitting
    else if (n.folds == 0 | n.folds == 1) {

      propensity_model <- probability_forest(
        as.matrix(data[, treatment_covariates]),
        as.factor(A))

      e_hat <- predict(
        propensity_model,
        newdata = as.matrix(data[, treatment_covariates]))$predictions[, 2]
    }
  }
  return(e_hat)
}

# Function to calculate IPTW Kaplan-Meier
IPTW_Kaplan_meier <- function(data, tau, X.names.propensity,
               nuisance_propensity = "glm", n.folds = NULL) {
```

```
    # Estimate propensity scores
    data$e_hat <- estimate_propensity_score(
      data,
      treatment_covariates = X.names.propensity,
      type_of_model = nuisance_propensity,
      n.folds = n.folds)

    # Truncate observed times at tau
    data$T_obs_tau <- pmin(data$T_obs, tau)

    # Define censoring status at tau
    data$status_tau <- as.numeric((data$T_obs >= tau) |
                                  (data$T_obs < tau & data$status == 1))

    # Calculate weights
    data$weights <- (data$A) * (1 / data$e_hat) + (1 - data$A) / (1 - data$e_hat)

    # Adjusted Kaplan-Meier estimator
    S <- adjusted.KM(
      times = data$T_obs,
      failures = data$status,
      variable = data$A,
      weights = data$weights)

    # Calculate RMST from the adjusted survival curves
    RMST <- RMST_1(S_A1 = S[S$variable == 1,],
                   S_A0 = S[S$variable == 0,],
                   tau = tau)

    return(list("intA0" = RMST$intA0, "intA1" = RMST$intA1, "RMST" = RMST$RMST))
  }
```

The next section will present several identifiability formulae when conditional independent censoring is met in an observational study.

4.2 Conditional independent censoring

Under Equation 14 (uncounfoundedness) and Equation 3 (conditional independent censoring), the causal effect is affected both by confounding variables (confounding bias) and by conditional censoring. In this context, θ_{RMST} can be identified using the conditional survival function without the need for weighting corrections. This approach results in the following identifiability formula and plug-in estimator.

4.2.1 Identifiability of θ_{RMST} by using conditional survival function

Under Equation 14 (uncounfoundedness), the θ_{RMST} can be identified very easily as follow:

$$\begin{aligned}\theta &= \mathbb{E}\left[T(1)\wedge\tau - T(0)\wedge\tau\right]\\ &= \mathbb{E}\left[\mathbb{E}\left[T(1)\wedge\tau - T(0)\wedge\tau \mid X\right]\right]\\ &\text{(Law of total probability)}\\ &= \mathbb{E}\left[\mathbb{E}\left[T(1)\wedge\tau \mid X, A=1\right] - \mathbb{E}\left[T(0)\wedge\tau \mid X = X, A=0\right]\right]\\ &\text{(Uncounfoundedness and Positivity of treatment)}\\ &= \mathbb{E}\left[\mathbb{E}\left[T\wedge\tau \mid X, A=1\right] - \mathbb{E}\left[T\wedge\tau \mid X, A=0\right]\right]\\ &\text{(Consistency)}\end{aligned} \tag{18}$$

This identifiability formula stands out from the others in this rather complex context because of its simplicity. The plug-in estimator derived from this formula is the g-formula. In the identification formula, it seems that this corresponding estimator does not need the Equation 3 and Equation 4 to be verified (conditional independent censoring and positivity for censoring). But these assumptions are implicit and necessary to allow an estimation of the conditional restricted mean survival.

4.2.1.1 Estimation with G-formula estimator

Introduced by J. Robins (1986), it is well-known in causal inference for overcoming confounding bias. It can be extended to survival data.

It offers an alternative to IPCW in leveraging the regression formulation. In addition, instead of fitting distinct models for the censoring mechanism and treatment probability, these estimators focus on modeling the conditional mean of the outcome. By applying these models to each treatment arm and subsequently marginalizing over the empirical covariate distributions of the target population, the corresponding expected outcome is derived (J. Robins 1986):

Definition 4.2 (G-formula plug-in estimator)**.**

$$\widehat{\theta}_{\text{g-formula}}(\tau) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{F}\left(X_i, 1\right) - \hat{F}\left(X_i, 0\right)\right)$$

with $\hat{F}(x,a) \triangleq \mathbb{E}[T\wedge\tau \mid X = x, A = a]$ the estimation of the conditional survival function.

Generally, $F(x,a)$ estimator can be obtained in fitting parametric (i.e. Weibull distribution), semi-parametric (i.e. Cox model) or event non-parametric model (i.e. survival random forest). $F(x,a)$ can be implemented in two ways:

- Fitting only one model adjusted on both covariates and treatment $F(a,x)$, and predicting on the all data if everyone had the treatment and on the all data if everyone had the control. This method is known as S-learner (single learner).
- Fitting two different estimators in adjusting on covariates. One estimator is fitted on treated observation $F(1,x)$ and one estimator on control $F(0,x)$, then the two fitted models predict on the all data set (as if all observations had received the treatment and respectively as if they had all received the control). This method is known as T-learner (two-learner).

The choice between the T-learner and the S-learner can be made according to the type of model used to estimate $F(x,a)$. For example, in a model where the treatment effect is expected to be additive and we wish to use a Cox model, a S-learner will probably not be suitable because of the violation of the proportional hazard hypothesis for variable A. In this case, the T-learner will therefore be preferred.

4.2.1.1.1 Properties of G-formula plug-in estimator

In Chen and Tsiatis (2001), they show the asymptotic normality of the G-estimator when the conditional survival probability $S(t|a,x) = P(T \wedge \tau > t \mid X = x, A = a)$ is estimated with a Cox model and the cumulative hazard function with a Breslow estimator for S-learner or T-learner. The asymptotic properties of G-formula can be obtained through the asymptotic properties of the influence functions φ (as the variance will be equal to $\mathbb{E}(\varphi^2)$). An expression of the asymptotic variance of G-formula estimator has been detailed in Chen and Tsiatis (2001):

$$\begin{aligned}
&(1-\pi) g_1^T \Sigma_1 g_1 + \int_0^L \frac{h_1^2(t)\lambda_1(t)}{s_1^{(0)}(t,\beta_1)} dt \\
&+ \pi g_0^T \Sigma_0 g_0 + \int_0^L \frac{h_0^2(t)\lambda_0(t)}{s_0^{(0)}(t,\beta_0)} dt \\
&+ \operatorname{var}\left[\int_0^L \left\{S_1(u \mid Z_i) - S_0(u \mid Z_i)\right\} du\right]
\end{aligned}$$

where

- $n_0 = \sum_{i=1}^n I(A_i = 0)$
- $\hat{\Sigma}_0 = n_0^{-1} \sum_{i=1}^n \int_0^L (1 - A_i) \times \left[\frac{S_0^{(2)}(t,\hat{\beta}_0)}{S_0^{(0)}(t,\hat{\beta}_0)} - \left\{\frac{S_0^{(1)}(t,\hat{\beta}_0)}{S_0^{(0)}(t,\hat{\beta}_0)}\right\}^{\otimes 2}\right] \times dN_i(t)$
- $\hat{g}_0 = {n_0}^{-1} \hat{\Sigma}_0^{-1} \sum_{i=1}^n \int_0^L \int_0^u \hat{S}_0(u \mid Z_i) e^{\hat{\beta}_0^T Z_i} \times \left\{\bar{Z}_0\left(t, \hat{\beta}_0\right) - Z_i\right\} d\hat{\Lambda}_0(t) du$
- $\hat{h}_0(t) = n^{-1} \sum_{i=1}^n \int_t^L \hat{S}_0(u \mid Z_i) e^{\hat{\beta}_0^T Z_i} du$
- $\widehat{\operatorname{var}}\left\{\int_0^L S_0(u \mid Z_i) du\right\}$ is obtained using the moment estimator:

$$n^{-1} \sum_{i=1}^n \left\{\int_0^L \hat{S}_0(u \mid Z_i) du - n^{-1} \sum_{j=1}^n \int_0^L \hat{S}_0(u \mid Z_j) du\right\}^2$$

Moreover, G-formula plug-in estimator is simple to implement and also very stable.

4.2.1.1.2 Implementation

The following code implements several function :

- Expected_survival computes the integral by the trapezoidal rule (described in Section 9) of a given survival function.
- g_formula_T_learner computes the presented G-formula plug in estimator in using two-learners survival forest or cox regression. It allows cross-fitting for survival forest (n.folds>1).
- g_formula_S_learner computes the presented G-formula plug in estimator in using a single learner survival forest or cox regression. It allows also cross-fitting for survival forest (n.folds>1).

```
# Compute the area under the survival curve for each individual using the
# Trapezoidal rule.
# S.hat: predicted survival function for each individual.
expected_survival <- function(S.hat, Y.grid) {
  # Y.grid: vector of time at which to evaluate the survival estimates
  # (same as S.hat).
  # Calculate the distance between each time point.
  grid.diff <- diff(c(0, Y.grid, max(Y.grid)))

  # Compute the area under each survival curve.
  area <- c(base::cbind(1, S.hat) %*% grid.diff)

  return(area)
}

# Function to estimate the g-formula Two-learner.
g_formula_T_learner <- function(data,
                                X.names.outcome,
                                tau,
                                nuisance_survival = "cox",
                                n.folds = NULL) {
  # Compute min(T_obs,tau)
  data$T_obs_tau <- ifelse(data$T_obs >= tau, tau, data$T_obs)

  # Y.grid is the grid of time points where we want to estimate the
  # survival function.
  Y.grid <- sort(unique(data$T_obs_tau))

  # Subset data for A == 0.
  data0 <- data %>% filter(A == 0)

  # Subset data for A == 1.
  data1 <- data %>% filter(A == 1)

  # Cox
  if (nuisance_survival == "cox") {
    outcome <- 'Surv(T_obs, status)'

    # Learn Cox regression on two datasets: A|X.
    f <- as.formula(paste(outcome, paste(c(X.names.outcome), collapse = " + "),
                          sep = " ~ "))

    # Fit the two models on the covariates of time Y.grid.
    fitS0 <- cph(f, data = data0, y = TRUE, x = TRUE, times = Y.grid)
    fitS1 <- cph(f, data = data1, y = TRUE, x = TRUE, times = Y.grid)

    # Predict survival probabilities for each individual at each Y.grid.
    fit.pred1 <- predictCox(fitS1, newdata = data, times = Y.grid,
                            type = "survival")
    fit.pred0 <- predictCox(fitS0, newdata = data, times = Y.grid,
                            type = "survival")

    # Survival probabilities for each individual at each Y.grid.
    S_hat1 <- fit.pred1$survival
    S_hat0 <- fit.pred0$survival
  } else {
    # Survival forest.
    # Initialize objects
    n <- nrow(data)
    fit.pred1 <- matrix(NA, nrow = n, ncol = length(Y.grid))
    fit.pred0 <- matrix(NA, nrow = n, ncol = length(Y.grid))

    if (n.folds > 1) {
      # Split the dataset into n-folds.
      indices <- split(seq(n), sort(seq(n) %% n.folds))

      # For all index in each split.
      for (idx in indices) {
        # Fit survival forest on all observations with A=1 except idx
        forest.grf1 <- survival_forest(
          X = as.matrix(data[-idx & data[, "A"] == 1, X.names.outcome]),
          Y = data[-idx & data[, "A"] == 1, "T_obs"],
          D = data[-idx & data[, "A"] == 1, "status"],
          failure.times = Y.grid
        )
        # Fit survival forest on all observations with A=0 except idx
        forest.grf0 <- survival_forest(
          X = as.matrix(data[-idx & data[, "A"] == 0, X.names.outcome]),
          Y = data[-idx & data[, "A"] == 0, "T_obs"],
          D = data[-idx & data[, "A"] == 0, "status"],
          failure.times = Y.grid
        )
        # Predict on idx
        fit.pred1[idx, ] <- predict(
          forest.grf1, as.matrix(data[idx, X.names.outcome]),
          failure.times = Y.grid)$predictions

        fit.pred0[idx, ] <- predict(
          forest.grf0, as.matrix(data[idx, X.names.outcome]),
          failure.times = Y.grid)$predictions
      }
    } else if (n.folds == 0 | n.folds == 1) {
      # If no cross-fitting
      # Fit survival forest on all observation with A=1
      forest.grf1 <- survival_forest(
        X = as.matrix(data[data[, "A"] == 1, X.names.outcome]),
        Y = data[data[, "A"] == 1, "T_obs"],
        D = data[data[, "A"] == 1, "status"],
        failure.times = Y.grid
      )
      # Fit survival forest on all observation with A=0
      forest.grf0 <- survival_forest(
        X = as.matrix(data[data[, "A"] == 0, X.names.outcome]),
        Y = data[data[, "A"] == 0, "T_obs"],
        D = data[data[, "A"] == 0, "status"],
        failure.times = Y.grid
      )
      # Predict on all observations
      fit.pred1 <- predict(forest.grf1, as.matrix(data[, X.names.outcome]),
                           failure.times = Y.grid)$predictions
      fit.pred0 <- predict(forest.grf0, as.matrix(data[, X.names.outcome]),
                           failure.times = Y.grid)$predictions
    }

    S_hat1 <- fit.pred1
    S_hat0 <- fit.pred0
  }

  # Compute the area under each survival curve until max(Y.grid) = tau.
  E_hat1 <- expected_survival(S_hat1, Y.grid)
  E_hat0 <- expected_survival(S_hat0, Y.grid)

  # Calculate the mean difference.
  theta_g_formula <- mean(E_hat1 - E_hat0)

  return(theta_g_formula)
}

# Function to estimate the g-formula Single-learner.
g_formula_S_learner <- function(data,
                                X.names.outcome,
                                tau,
                                nuisance_survival = "cox",
                                n.folds = NULL) {
  # Compute min(T_obs,tau)
  data$T_obs_tau <- ifelse(data$T_obs >= tau, tau, data$T_obs)

  # Y.grid is the grid of time points where we want to estimate the
  # survival function.
  Y.grid <- sort(unique(data$T_obs_tau))

  # Set A=0 for all data
  data0 <- data
  data0$A <- 0

  # Set A=1 for all data
  data1 <- data
  data1$A <- 1

  # Cox
  if (nuisance_survival == "cox") {
    outcome <- 'Surv(T_obs, status)'
    # Learn Cox regression on one datasets and add A as covariate
    f <- as.formula(paste(outcome, paste(c(X.names.outcome,"A"),
                                         collapse = " + "),
                          sep = " ~ "))

    # Fit the two models on the covariates of time Y.grid.
    fitS <- cph(f, data = data, y = TRUE, x = TRUE, times = Y.grid)

    # Predict survival probabilities for each individual at each Y.grid.
    fit.pred1 <- predictCox(fitS, newdata = data1, times = Y.grid,
                            type = "survival")
    fit.pred0 <- predictCox(fitS, newdata = data0, times = Y.grid,
                            type = "survival")

    # Survival probabilities for each individual at each Y.grid.
    S_hat1 <- fit.pred1$survival
    S_hat0 <- fit.pred0$survival
  } else {
    # Survival forest.
    # Initialize objects
    n <- nrow(data)
    fit.pred1 <- matrix(NA, nrow = n, ncol = length(Y.grid))
    fit.pred0 <- matrix(NA, nrow = n, ncol = length(Y.grid))

    if (n.folds > 1) {
      # Split the dataset into n-folds.
      indices <- split(seq(n), sort(seq(n) %% n.folds))

      # For all index in each split.
      for (idx in indices) {
        # Fit survival forest on all observations except idx (add A as covariate)
        forest.grf <- survival_forest(
          X = as.matrix(data[-idx, c(X.names.outcome,"A")]),
          Y = data[-idx, "T_obs"],
          D = data[-idx, "status"],
          failure.times = Y.grid
        )
        # Predict on idx
        fit.pred1[idx, ] <- predict(
          forest.grf, as.matrix(data1[idx, c(X.names.outcome,"A")]),
          failure.times = Y.grid)$predictions

        fit.pred0[idx, ] <- predict(
          forest.grf, as.matrix(data0[idx, c(X.names.outcome,"A")]),
          failure.times = Y.grid)$predictions
      }
    } else if (n.folds == 0 | n.folds == 1) {
      # If no cross-fitting
      # Fit survival forest on all observation (add A as covariate)
      forest.grf <- survival_forest(
        X = as.matrix(data[, c(X.names.outcome,"A")]),
        Y = data[, "T_obs"],
        D = data[, "status"],
        failure.times = Y.grid
      )

      # Predict on all observations
      fit.pred1 <- predict(
        forest.grf, as.matrix(data1[, c(X.names.outcome,"A")]),
        failure.times = Y.grid)$predictions
      fit.pred0 <- predict(
        forest.grf, as.matrix(data0[, c(X.names.outcome,"A")]),
        failure.times = Y.grid)$predictions
    }

    S_hat1 <- fit.pred1
    S_hat0 <- fit.pred0
  }

  # Compute the area under each survival curve until max(Y.grid) = tau.
  E_hat1 <- expected_survival(S_hat1, Y.grid)
  E_hat0 <- expected_survival(S_hat0, Y.grid)

  # Calculate the mean difference.
  theta_g_formula <- mean(E_hat1 - E_hat0)

  return(theta_g_formula)
}
```

It exists other type of estimators valid in this context. The following section will present the identifiability formulae, corresponding estimators and implementations of another estimator with double correction weighting.

4.2.2 Identifiability of θ_{RMST} by double weighting with IPC transformation

Under Equation 14 (uncounfoundedness) and Equation 3 (conditional independent censoring), θ_{RMST} can be also identified as follow:

$$
\begin{aligned}
\theta_{RMST} &= \mathbb{E}[T(1) \wedge \tau - T(0) \wedge \tau] \\
&= \int_0^\tau \mathbb{E}[I\{T(1) > t\}] - \mathbb{E}[I\{T(0) > t\}]dt \\
&\text{(By linearity)} \\
&= \int_0^\tau \mathbb{E}\left[\mathbb{E}[I\{T(1) > t\}|X, A]\right] - \mathbb{E}\left[\mathbb{E}[I\{T(0) > t\}|X, A]\right] \\
&\text{(Law of total probability)} \\
&= \int_0^\tau \mathbb{E}\left[1(\{T(1) \geq t\}) * \frac{\Delta^\tau}{S_C(T \wedge \tau \mid A, X)} * \frac{A}{e(X)}|A = 1\right] - \\
&\mathbb{E}\left[1(\{T(0) \geq t\}) * \frac{\Delta^\tau}{S_C(T \wedge \tau \mid A, X)} * \frac{1 - A}{1 - e(X)}|A = 0\right] dt \\
&\text{(By def of IPCW and IPTW)} \\
&= \int_0^\tau \mathbb{E}\left[1(\{T \geq t\}) * \frac{\Delta^\tau}{S_C(T \wedge \tau \mid A, X)} * \frac{A}{e(X)}|A = 1\right] - \\
&\mathbb{E}\left[1(\{T \geq t\}) * \frac{\Delta^\tau}{S_C(T \wedge \tau \mid A, X)} * \frac{1 - A}{1 - e(X)}|A = 0\right] dt \\
&\text{(By consistency)}
\end{aligned}
\tag{19}
$$

The identifiability formula Equation 19 uses the two previous weighting correction: IPCW presented in Section 3.2 and IPTW in Section 4.1.

4.2.2.1 Estimation with IPCW-IPTW Kaplan Meier

The IPTW-IPCW Kaplan Meier estimator can be used to estimate the causal treatment effect according to the identifiability formula Equation 19:

Definition 4.3 (IPTW-IPCW Kaplan Meier estimator)**.**

$$
\hat{S}_{IPTW-IPCW}(t \mid A = a) = \prod_{j=1, t_j <= t} \left(1 - \frac{\sum_i \hat{w}_i(t, X_i) * I\{T_i = t_j, C_i \geq t_j, A_i = a\}}{\sum_i \hat{w}_i(t, X_i) * I\{T_i \geq t_j, C_i \geq t_j, A_i = a\}}\right) \tag{20}
$$

with $\hat{w}_i(t, X_i) = \frac{\Delta_i^\tau}{\hat{S}_C(\widetilde{T} \wedge \tau | A_i, X_i)} * (\frac{A_i}{\hat{e}(X_i)} + \frac{1 - A_i}{1 - \hat{e}(X_i)})$ the corresponding weight including the inverse of the estimated propensity score ($\hat{e}(X)$) and the inverse estimated probability of remain uncensored given the covariates ($\hat{S}_c(t|X)$) and the censoring status at τ (Δ^τ) already presented in Section 3.2 and Section 4.1.

This IPTW-IPCW Kaplan Meier estimator enables a balance between treatment and control groups and between censored and uncensored individuals. The corresponding θ_{RMST} is the integral of the difference between the survival curve with $A = 1$ and the $A = 0$.

4.2.2.1.1 Properties of IPTW-IPCW Kaplan Meier estimator

This estimator converges almost surely and uniformly to the true θ_{RMST} if the two nuisance parameters $\hat{e}(X_i)$ and $\hat{S}_C\left(\widetilde{T} \wedge \tau \mid A_i, X_i\right)$ are well estimated (Schaubel and Wei 2011). Like the IPTW Kaplan-Meier (in Section 4.1.1.1) or the IPCW Kaplan-Meier (in Section 3.2.2.1), this estimator is even more subject to extreme weights since it is weighted by two inverse probabilities. This makes it very sensitive to positivity for censoring and processing.

4.2.2.1.2 Implementation

The following code implements the IPTW-IPCW Kaplan Meier function, which uses the previously implemented functions:

- estimate_survival_function (detailed and implemented in Section 3.2.2.1.2) which computes the probability of remaining uncensored over time S_c using either a Cox model or a survival forest.
- estimate_propensity_score (detailed and implemented in Section 4.1.1.1.2) which computes the propensity score of each observation using a logistic model or a probability forest.
- adjusted.KM (detailed and implemented in Section 3.2.2.1.2) which computes an adjusted Kaplan Meier estimator.
- RMST_1 (detailed and implemented in Section 3.1.2.2) which computes RMST.

```r
IPTW_IPCW_Kaplan_meier <- function(data,
                                   X.names.propensity,
                                   X.names.censoring,
                                   tau,
                                   nuisance_propensity = "glm",
                                   nuisance_censoring = "cox",
                                   n.folds = NULL) {
  # Censoring time to tau if observed time exceeds tau
  data$T_obs_tau <- ifelse(data$T_obs >= tau, tau, data$T_obs)

  # Create censoring status for tau
  data$censor.status_tau <- 1 - as.numeric((data$T_obs >= tau) |
                                           (data$T_obs < tau & data$status == 1))

  # Create status at tau
  data$status_tau <- as.numeric((data$T_obs >= tau) |
                                (data$T_obs < tau & data$status == 1))

  # Grid of unique observed times truncated at tau
  Y.grid <- sort(unique(data$T_obs_tau))

  # Estimate propensity scores
  data$e_hat <- estimate_propensity_score(data,
                                          treatment_covariates = X.names.propensity,
                                          type_of_model = nuisance_propensity,
                                          n.folds = n.folds)

  # Estimate survival function for censoring
  S_C_hat <- estimate_survival_function(data, X.names = X.names.censoring,
                                        Y.grid = Y.grid, T_obs = "T_obs_tau",
                                        status = "censor.status_tau",
                                        type_of_model = nuisance_censoring,
                                        n.folds = n.folds)

  # Get estimated survival probabilities for censoring
  data$S_C <- S_C_hat$S_hat[cbind(1:nrow(data), match(data$T_obs_tau, Y.grid))]

  # Calculate weights
  data$weights <- data$status_tau / data$S_C *
                  (data$A * (1 / data$e_hat) +
                     (1 - data$A) * (1 / (1 - data$e_hat)))

  # Compute adjusted Kaplan-Meier estimator
  S <- adjusted.KM(times = data$T_obs_tau,
                   failures = data$status_tau,
                   variable = data$A,
                   weights = data$weights)

  # Compute Restricted Mean Survival Time (RMST)
  RMST <- RMST_1(S_A1 = S[S$variable == 1, ],
                 S_A0 = S[S$variable == 0, ],
                 tau = tau)

  # Return RMST and ATE for treated and not treated groups
  return(list(RMST = RMST$RMST, ATE_treated = RMST$intA1,
              ATE_not_treated = RMST$intA0))
}
```

As in Section 3.2.3, it exists other solutions to identify θ_{RMST}. One of them apply directly the inverse propensity weighting on the pseudo-observation without using survival function. It leads to the following identifiability formula.

4.2.3 Identifiability of θ_{RMST} by double weighting with BJ transformation

Under Equation 14 (uncounfoundedness), the θ_{RMST} can be identified as follow:

$$
\begin{aligned}
\theta &= \mathbb{E}[T(1)\wedge\tau - T(0)\wedge\tau] \quad (1)\\
&= \mathbb{E}[T(1)\wedge\tau] - \mathbb{E}[T(0)\wedge\tau] \quad (2)\\
&\text{(By linearity)}\\
&= \mathbb{E}\left[\mathbb{E}[T(1)\wedge\tau \mid X]\right] - \mathbb{E}[T(0)\wedge\tau \mid X] \quad (3)\\
&\text{(By the total probability law)}\\
&= \mathbb{E}\left[\mathbb{E}[(T\wedge\tau) \mid A, X]\left(\frac{A}{e(X)} - \frac{1-A}{1-e(X)}\right)\right] \quad (4)\\
&\text{(def of IPTW)}\\
&= \mathbb{E}\left[\mathbb{E}[T^* \mid A, X]\left(\frac{A}{e(X)} - \frac{1-A}{1-e(X)}\right)\right] \quad (5)\\
&= \mathbb{E}\left[(\Delta^\tau * (\widetilde{T}\wedge\tau) + (1-\Delta^\tau) * Q_S(C|X,A)) * \left(\frac{A}{e(X)} - \frac{1-A}{1-e(X)}\right)\right] \quad (6)\\
&\text{(by Buckley James transformation)}
\end{aligned}
\tag{21}
$$

The transition from line 5 to line 6 has been explained by the Buckley-James transformation explained in Section 3.2.1.2. As in the identifiability formula Equation 18 of G-formula plug-in estimator, this identifiability formula do need implicitly for Equation 3 (conditional independent censoring) or Equation 4 (positivity for censoring) to be verified.

The next section will start with the estimation based on the identifiability formula in Equation 21.

4.2.3.1 Estimation with IPTW-BJ estimator

Based on the identifiability of Equation 21, the IPTW-BJ estimator is defined as follow:

Definition 4.4 (IPTW-BJ estimator).

$$\hat{\theta}_{\mathrm{IPTW-BJ}}(\tau) = \frac{1}{n}\sum_{i=1}^{n}\left(\Delta^{\tau}\widetilde{T}\wedge\tau + (1-\Delta^{\tau})\hat{Q}_S(\widetilde{T}\wedge\tau|X,A)\right)\left(\frac{A_i}{\hat{e}\left(X_i\right)} - \frac{1-A_i}{1-\hat{e}\left(X_i\right)}\right).$$

Exactly than simple BJ estimator (introduced in Section 3.2.3.1), this estimator is easier to implement than the weighting survival function.

4.2.3.1.1 Properties of IPTW-BJ estimator

Exactly than IPTW-IPCW (described in Section 4.2.2.1.1), IPTW-Bj estimator converges almost surely and uniformly to the true θ_{RMST} if the two nuisance parameters $\hat{e}(X_i)$ and $\hat{Q}_S\left(\widetilde{T}\wedge\tau \mid A_i, X_i\right)$ are well estimated (Andersen, Syriopoulou, and Parner 2017).

4.2.3.1.2 Implementation of IPTW-BJ estimator

The following code implements the IPTW-BJ function, which uses the previously implemented functions:

- estimate_survival_function (detailed and implemented in Section 3.2.2.1.2) which computes the probability of remaining uncensored over time S_c using either a Cox model or a survival forest.
- estimate_propensity_score (detailed and implemented in Section 4.1.1.1.2) which computes the propensity score of each observation using a logistic model or a probability forest.

```r
# In using the min
IPTW_BJ <- function(data,
                    X.names.propensity,
                    X.names.outcome,
                    tau,
                    nuisance_propensity = "glm",
                    nuisance = "cox",
                    n.folds = NULL) {
  # Minimum of T_obs and tau
  data$T_obs_tau <- ifelse(data$T_obs >= tau, tau, data$T_obs)

  # Grid of unique observed times truncated at tau
  Y.grid <- sort(unique(data$T_obs_tau))

  # Indicator for min(T, tau) < C
  data$status_tau <- as.numeric((data$T_obs >= tau) |
                                (data$T_obs < tau & data$status == 1))


  # Estimate propensity scores
  data$e_hat <- estimate_propensity_score(data,
                                          treatment_covariates = X.names.propensity,
                                          type_of_model = nuisance_propensity,
                                          n.folds = n.folds)


  # Estimation of Q_s
  Q_t <- Q_t_hat(data, tau, X.names.outcome, nuisance, n.folds)
  data$Q_y <-  Q_Y(data,tau,Q_t)

  # BJ transformation
  data$Y <-  data$status_tau * data$T_obs_tau +
                             (1 - data$status_tau) * data$Q_y

  # IPTW on BJ transformation
  data$RST <- data$Y * (data$A/data$e_hat-(1-data$A)/(1-data$e_hat))

  RMST <- mean(data$RST)

  # Return RMST and other relevant metrics
  return(RMST)
}
```

As seen in the previous properties, the estimators with inverse weighting such as IPTW-IPCW or BJ-IPTW are subject to instability and sensible to mis-specification of nuisance models. Regression based estimators such as G-formula are also subject to mis-specification of the outcome model. A possibility to overcome these issues is to use doubly robust corrections that is by property more robust to mis-specification. The next section will discuss the identifiability of θ_{RMST} using doubly robust methods. First, we will introduce the concept of the influence function, which enables the construction of the augmented version of IPTW. The properties of this AIPTW estimator will be disclosed. Next, we will present the augmented version of IPCW, another unbiased censoring transformation. Finally, we will describe the corresponding estimator with double augmented weighting corrections for treatment and censoring and explain its theoretical properties.

4.2.4 Identifiability of θ_{RMST} by augmented corrections

The following sub-sections explain how to derive the augmented weighting based on a semi-parametric approach, firstly to correct for the confounding effect, and secondly to correct for the conditionally independent censoring. In the next part, we consider a fully observed population (as there is no censoring).

4.2.4.1 Augmented weighting for treatment from semi-parametric approach

The semi-parametric approach allows to create estimators $\sqrt{n}$ consistent with valid confidence interval. It is a model free functional of the observed data distribution based on the derivation of the efficient influence function (Hines et al. 2022).

Influence function is used to measure how much an estimator changes in response to a slight perturbation in the sample distribution. To introduce how to identify the augmented version of IPTW in using influence function, let's say we have a distribution P, a perturbed distribution $\bar{P}$ and that we have access to a "mixture" distribution $P_\epsilon = P + \epsilon(\bar{P} - P)$ where $\epsilon \in [0; 1]$ is a slight perturbation (Fisher and Kennedy 2021). P_0 corresponds to the true distribution (unknown) and the objective is to estimate $\psi(P_0)$.

In following the method to derive influence function (detailed in Section 9), the influence function for the plug-in estimator $\psi(P) = \mathbb{E}_P[\mathbb{E}_P(T \wedge \tau | X, A = 1)]$ which satisfies Equation 22 (in Section 9) is defined as (detail in Kennedy (2023)):

$$\varphi(\psi) = \frac{A}{e(X)}(T \wedge \tau - F(X, A = 1)) + F(X, A = 1) - \psi$$

Thus the corresponding bias-corrected estimator is:

$$\psi(P) = \mathbb{P}_n \left[\hat{F}(X, A = 1) + \frac{A.(T \wedge \tau - \hat{F}(X, A = 1))}{\hat{e}(X)} \right]$$

where $\mathbb{P}_n$ means the corresponding sample average, $\hat{F}(X)$ is the estimated conditional survival and $\hat{e}(X)$ is the estimated propensity score.

When we come back to our estimand of interest θ_{RMST}, $\psi(P) = \mathbb{E}_P[\mathbb{E}_P(T \wedge \tau | X, A = 1) - \mathbb{E}_P(T \wedge \tau | X, A = 0)]$, the efficient influence function is then:

$$\begin{aligned} \varphi(\psi) = {} & F(X, A = 1) - F(X, A = 0) + \\ & \frac{A}{e(X)}(T \wedge \tau - F(X, A = 1)) - \frac{1 - A}{1 - e(X)}(T \wedge \tau - F(X, A = 0)) \end{aligned}$$

Finally, the bias-corrected estimator can be defined as:

$$\begin{aligned} \hat{\theta}_{AIPTW} &= \mathbb{E}[\mathbb{E}(T \wedge \tau | X, A = 1)] - \mathbb{E}[\mathbb{E}(T \wedge \tau | X, A = 0)] \\ &= \mathbb{E}\left[\underbrace{\hat{F}(X, A = 1)}_{1} + \underbrace{\frac{A.(T \wedge \tau - \hat{F}(X, A = 1))}{\hat{e}(X)}}_{2} \right] - \\ &\quad \mathbb{E}\left[\underbrace{\hat{F}(X, A = 0)}_{1} + \underbrace{\frac{(1 - A).(T \wedge \tau - \hat{F}(X, A = 0))}{1 - \hat{e}(X)}}_{2} \right] \end{aligned}$$

This augmented inverse probability of treatment weighting (AIPTW) estimator does direct conditional survival (1) and IPW to conditional survival residuals (2).

In the exact same way, AIPTW can also be defined as:

$$\begin{aligned}\hat{\theta}_{AIPTW} &= \mathbb{E}[\mathbb{E}(T \wedge \tau|X, A = 1)] - \mathbb{E}[\mathbb{E}(T \wedge \tau|X, A = 0)] \\ &= \mathbb{E}\left[\underbrace{\frac{A.T \wedge \tau}{\hat{e}(X)}}_{1} + \underbrace{\hat{F}(X, A = 1) * \frac{\hat{e}(X) - A}{\hat{e}(X)}}_{2}\right] - \\ &\quad \mathbb{E}\left[\underbrace{\frac{(1 - A) * T \wedge \tau}{1 - \hat{e}(X)}}_{1} + \underbrace{\hat{F}(X, A = 0) * \frac{(1 - \hat{e}(X)) - A}{1 - \hat{e}(X)}}_{2}\right]\end{aligned}$$

In that case, AIPTW does direct IPTW adjustment (1) and conditional survival adjustment on IPTW residuals (2).

This estimator is an adaptation of the AIPW estimator in causal inference (James M. Robins and Zhao 1994, 1995; Chernozhukov et al. 2016).

4.2.4.1.1 Properties of AIPTW estimator

The resulting estimators from semi-parametric approach have been described to be doubly robust. In our case, double robustness means that the estimator is consistent if either the treatment model $\hat{F}(X, A)$ or the outcome model $\hat{e}(X)$ is correctly specified. Based on what we saw in the previous section, this double robustness properties can be explained by the decomposition of $\hat{\psi} - \psi$:

$$\begin{aligned}\widehat{\psi} - \psi &= \psi(\widehat{\mathbb{P}}) + \mathbb{P}_n\{\varphi(Z; \widehat{\mathbb{P}})\} - \psi(\mathbb{P}) \\ &= (\mathbb{P}_n - \mathbb{P})\{\varphi(Z; \widehat{\mathbb{P}})\} + R_2(\widehat{\mathbb{P}}, \mathbb{P}) \\ &= \underbrace{(\mathbb{P}_n - \mathbb{P})\{\varphi(Z; \mathbb{P})\}}_{\text{S*}} + \underbrace{(\mathbb{P}_n - \mathbb{P})\{\varphi(Z; \widehat{\mathbb{P}}) - \varphi(Z; \mathbb{P})\}}_{\text{T}_1} + \underbrace{R_2(\widehat{\mathbb{P}}, \mathbb{P})}_{\text{T}_2}\end{aligned}$$

$S^* = o_{\mathbb{P}}(1/\sqrt{n})$ by the central limit theorem. T_1, known as the empirical process, is of smallest order (sample average of a short term as $\varphi(Z; \widehat{\mathbb{P}})$ converge to $\varphi(Z; \mathbb{P})$) when working with not too complex estimators (Donsker estimator) and in using sample splitting. T_2, the crucial one, can often converge to zero under some conditions such as sparsity or smoothness (negligible under nonparametric conditions). The double robustness properties can be explained by the remainder term T_2.

By Cauchy-Schwarz theorem, this remainder term T_2 can be lower and gives:

$$\left|R_2(\widehat{\mathbb{P}}, \mathbb{P})\right| \leq \left(\frac{1}{\epsilon}\right) \int \left|e(x) - \widehat{e}(x)\right| \left|F(x) - \widehat{F}(x)\right| d\mathbb{P}(x) \leq \left(\frac{1}{\epsilon}\right) \left\| \widehat{e} - e\| \|\widehat{F} - F\|\right.$$

where $e(x) = P(A = 1 \mid X = x)$ with known and $\hat{e}(x) = \hat{P}(A = 1 \mid X = x)$, and similarly for $F(x) = \mathbb{E}_P(Y \mid X = x, A = 1)$. If $\widehat{e}(x) \geq \epsilon$ with $\epsilon > 0$ and that the combination product of $\|\widehat{e} - e\|$ and $\|\widehat{F} - F\|$ give $o_{\mathbb{P}}(\frac{1}{\sqrt{n}})$, thus, $\left|R_2(\widehat{\mathbb{P}}, \mathbb{P})\right| = o_{\mathbb{P}}(\frac{1}{\sqrt{n}})$. In other words, the residual term tends towards zero even if one of the estimators $widehatF(x)$ or $widehate(x)$ converges slowly while the other compensates by converging more quickly and vice-versa.

To conclude, AIPTW is root-n consistent, asymptotically normal (even in non-parametric approach) and attains the nonparametric efficiency bound (no estimator can have smaller mean squared error than this estimator) (Kennedy 2023).

This estimator only considers correction for treatment in case of confounding effect. In our context of Equation 14 (uncounfoundedness) and Equation 3 (conditional independent censoring), this estimator is not sufficient as it does not consider censoring at all. This augmented treatment weighting could be associated with transformations already mentioned above but an augmented transformation based on the semi-parametric approach also exists for censoring to overcome this problem and form an estimator that is also doubly robust for censoring.

4.2.4.2 Augmented weighting for conditionally independent censoring from semi-parametric approach

The augmented weighting to correct conditionally independent censoring is considered as an augmented censoring unbiased transformation (simple transformations were introduced in Section 3.2.1).

This augmented transformation from Fan and Gijbels (1994a) is inspired by the Buckley James transformation (Buckley and James 1979) and Inverse probability censoring transformation (Koul, Susarla, and Ryzin 1981) (detailed in Section 3.2.1.2 and Section 3.2.1.1) with an augmentation terms (Tsiatis 2006; Laan and Robins 2003).

This transformation can be expressed as (D. Rubin and Laan 2007):

$$\begin{aligned}
T^{\star}(O) &= T_{S,S_c^{\star}}(O) \\
&= \frac{\widetilde{T}\wedge\tau\Delta^{\tau}}{S_c(\widetilde{T}\wedge\tau \mid X,A)} + \frac{Q_S(\widetilde{T}\wedge\tau \mid X,A)(1-\Delta^{\tau})}{S_c(\widetilde{T}\wedge\tau \mid X,A)} \\
&\quad - \int_{-\infty}^{\widetilde{T}\wedge\tau} \frac{Q_S(\widetilde{T}\wedge\tau \mid X,A)}{S_c^2(\widetilde{T}\wedge\tau \mid X,A)}\, d(1 - S_c(\widetilde{T}\wedge\tau \mid X,A))
\end{aligned}$$

With $Q_S(\widetilde{T}\wedge\tau \mid X,A) = \frac{1}{S(\widetilde{T}\wedge\tau|X,A)}\int_{\widetilde{T}\wedge\tau}^{+\infty}\widetilde{T}\wedge\tau.dF(\widetilde{T}\wedge\tau \mid X,A)$ the estimation function of the remaining survival function ($E[\widetilde{T}\wedge\tau \mid X,A,\widetilde{T}\wedge\tau > t]$)

4.2.4.2.1 Properties of AIPCW estimator

As derived from semiparametric approach, AIPCW estimator has double robustness properties. The augmented inverse probability of censoring transformation is unbiased if either $S_c(t|x,a)$ or $Q_S(t|x,a)$ is correctly estimated. It gives a clear advantage to the previous transformation in Section 3.2.1.1. Also, this estimator is root-n consistent, asymptotically normal (even in non-parametric approach) and attains the nonparametric efficiency bound (details in Laan and Robins (2003)).

In order to obtain a complete correction for confounding effect and conditionally independent censoring, the both estimator can be combined.

4.2.4.3 Double augmented weighting correction

The augmented inverse probability of treatment weighting (AIPTW) and the augmented inverse probability of censoring weighting (AIPCW) can be combined to form an augmented estimator of RMST with conditional censored data .

Definition 4.5 (AIPTW- AIPCW estimator (IPTW on pseudo-observation and correction on IPTW residuals)).

$$\hat{\theta}_{AIPTW-AIPCW} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{A_i}{\hat{e}\left(X_i\right)} - \frac{1-A_i}{1-\hat{e}\left(X_i\right)}\right)\hat{T}^*_{DR}$$
$$+ \hat{F}\left(X_i, A=1\right)\left(1-\frac{A_i}{\hat{e}\left(X_i\right)}\right) - \hat{F}\left(X_i, A=0\right)\left(1-\frac{1-A_i}{1-\hat{e}\left(X_i\right)}\right)$$

with

$\hat{T}^*_{DR} = \frac{\widetilde{T}_i \wedge \tau \cdot \Delta^\tau}{\hat{S_C}\left(\widetilde{T}_i \wedge \tau | X_i\right)} + \frac{Q_{\hat{S}}(\tilde{T}_i \wedge \tau | X, A) \cdot (1-\Delta^\tau)}{\hat{S_C}(\tilde{T}_i \wedge \tau | X_i)} - \int_0^{\widetilde{T}_i \wedge \tau} \frac{Q_{\hat{S}}(c | X_i, A_i)}{\hat{S_C}^2(c | X_i, A_i)} d(1-\hat{S_C}(c \mid X_i, A_i))$ which corresponds to the previous AIPCW estimator in Section 4.2.4.2.

Basically, the above AIPTW-AIPCW applies direct IPTW on the augmented unbiased pseudo-population (that would have been observed if there was no censoring) and regression correction on IPTW residuals.

This estimator can be also written as:

Definition 4.6 (AIPTW- AIPCW estimator (Conditional survival and correction on survival residuals with pseudo-observation)).

$$\hat{\theta}_{AIPTW-AIPCW} = \frac{1}{n}\sum_{i=1}^{n}\hat{F}\left(X_i, A=1\right) - \hat{F}\left(X_i, A=0\right)$$

$$+ \left(\frac{A_i * (\hat{T}^*_{DR} - \hat{F}\left(X_i, A=1\right))}{\hat{e}\left(X_i\right)} - \frac{(1-A_i) * (\hat{T}^*_{DR} - \hat{F}\left(X_i, A=0\right))}{1-\hat{e}\left(X_i\right)}\right)$$

In this case, the estimator applies direct conditional survival and the residuals of the conditional survival (based on the augmented unbiased pseudo-population) is weighted by IPTW.

4.2.4.3.1 Properties of AIPTW-AIPCW estimator

The estimator AIPTW-AIPCW involves nuisance models for the outcome $\hat{S}$ to estimate both Q_S and F, for the treatment to estimate the propensity score e and for the censoring to estimate S_c. As introduced in Section 4.2.4.1.1 and Section 4.2.4.2.1, this estimator achieves double robustness, root-n consistency and is asymptotically normal and attains the nonparametric efficiency bound (Kennedy 2023). Concerning the double robustness, the estimator is expected to be consistent only if:

- $\hat{S}_c(t|a,x)$ and $\hat{e}(x)$ are consistent.
- $\hat{S}(t|a,x)$ is consistent.

In addition, this method has to incorporate cross-fitting (Zheng and Laan 2012; Chernozhukov et al. 2016) to provide an efficient (reduce possible overfitting) and unbiased method. Also, this estimator is also refereed as the locally efficient estimator written as the solution of the efficient influence curve equation.

In the exact same way than AIPTW and AIPCW, the AIPTW-AIPCW estimator enables the use of machine-learning estimation of nuisance functions while preserving the root-n consistency of the AIPTW-AIPCW estimator (Chernozhukov et al. 2016).

4.2.4.3.2 Implementation

The following code implements the AIPTW-AIPCW function, which uses other tool functions:

- Estimate_hazard_function which computes the instantaneous hazard function of the corresponding survival function necessary to compute the third term of AIPCW. This function compute the instantaneous hazard function as a forward difference of $-\log\left(\hat{S}(t \mid x)\right)$.
- Integrate which integrate from 0 to $T_i \wedge \tau$ for each individuals an integrand also to compute the third term of AIPCW.
- AIPCW which finally compute pseudo observation based on all previous functions, estimate_survival_function (detailed and implemented in Section 3.2.2.1.2), Q_t_hat function and Q_Y function (implemented in Section 3.2.3.1.2). The third term of the transformation is simplified by: $\int_0^{T_i \wedge \tau} \frac{\lambda_c(s|A_i,X_i)}{\hat{S}_C(s|A_i,X_i)} * Q_S(s \mid X_i, A_i)ds$ (detail in Section 9).

```r
# Tool functions
# Compute hazard function from survival function
estimate_hazard_function <- function(S_hat, Y.grid) {

  Y.grid[Y.grid==0]<-0.001

  # Calculate differences between successive elements in Y.grid
  Y.diff <- diff(c(0, Y.grid))

  # Get the number of columns in S_hat
  grid.length <- ncol(S_hat)

  # Compute -log of survival probabilities (cumulative hazard function),
  # Add 1 as the first value of survival function to ensure that lambda(0)=0
  log.surv.C <- -log(base::cbind(1, S_hat))

  # Calculate differences of -log survival probabilities to have
  # the instantaneous hazard function
  h_hat <- log.surv.C[, 2:(grid.length + 1)] - log.surv.C[, 1:grid.length]

  # Divide each column of h_hat by the corresponding element in Y.diff
  h_hat <- sweep(h_hat, 2, Y.diff, "/")

  # Return the estimated hazard function
  return(h_hat)
}

integrate <- function(integrand, Y.grid, times) {
  # Create a filter matrix to indicate which elements are within the time
  # interval
  filter <- sapply(1:length(Y.grid), function(i) {
    return(as.numeric(i <= findInterval(times, Y.grid)))
  })
  # Apply the filter to the integrand
  integrand_filtered <- filter * integrand

  # Sum the rows of the filtered integrand to get the integrated values
  integrated_value <- rowSums(integrand_filtered)

  # Return the integrated values
  return(integrated_value)
}


# DR censoring transformation
AIPCW <-function(data,
                 tau,
                 X.names.censoring = c("X1","X2","X3","X4"),
                 X.names.outcome = c("X1","X2","X3","X4"),
                 nuisance_Qt = "cox",
                 nuisance_censoring = "cox",
                 n.folds = NULL,
                 h_C_hat = NULL,
                 method_aipw = 1) {

  # Truncate observed times at tau
  data$T_obs_tau <- pmin(data$T_obs, tau)

  # Define status at tau
  data$status_tau <-  as.numeric((data$T_obs > tau) |
                                  (data$T_obs <= tau &  data$status == 1 ))

  data$censor.status_tau <- 1- as.numeric(
    (data$T_obs > tau) | (data$T_obs <= tau &  data$status == 1 ))

  Y.grid <- sort(unique(data$T_obs_tau))

  # Estimate survival function for censoring
  S_C_hat <- estimate_survival_function(data = data,X.names.censoring,
                                        type_of_model = nuisance_censoring,
                                        n.folds = n.folds,
                                        Y.grid = Y.grid,
                                        T_obs = "T_obs_tau",
                                        status = "censor.status_tau")

  Y.index <- findInterval(data$T_obs_tau, Y.grid)

  data$S_C_hat_T_obs_tau <- S_C_hat$S_hat[cbind(seq_along(Y.index), Y.index)]


  if (is.null(h_C_hat)) {
      h_C_hat <- estimate_hazard_function(S_C_hat$S_hat,Y.grid)
  }

  # Compute Q.t.hat
  Q.t.hat <- Q_t_hat(data = data,
                     X.names = X.names.outcome,
                     tau = tau,
                     nuisance = nuisance_Qt,
                     n.folds = n.folds)

  # Compute Q.Y.hat
  data$Q.Y.hat <- Q_Y(data = data, tau, Q.t.hat)

  # Compute first term
  data$first_term <- (data$T_obs_tau * data$status_tau) /
    data$S_C_hat_T_obs_tau

  # Compute second term
  data$second_term <- (data$Q.Y.hat * (1 - data$status_tau)) /
    data$S_C_hat_T_obs_tau

  Y.diff <- diff(c(0, Y.grid))

  # Compute integrand for the third term
  integrand <- sweep( ( (h_C_hat) / S_C_hat$S_hat )* (Q.t.hat), 2, Y.diff, "*")

  # Compute third term
  data$third_term <- integrate(integrand, Y.grid, data$T_obs_tau)

  # Compute pseudo outcome
  pseudo_outcome <- data$first_term + data$second_term - data$third_term

  return(pseudo_outcome)
}
```

Then, we can compute the AIPTW-AIPCW estimator which uses all the previous functions and estimate_propensity_score (detailed and implemented in Section 4.1.1.1.2) which computes the propensity score of each observation using a logistic model or a probability forest:

```
AIPTW_AIPCW <- function(data,
                        tau,
                        X.names.propensity = c("X1", "X2", "X3", "X4"),
                        X.names.censoring = c("X1", "X2", "X3", "X4"),
                        X.names.outcome = c("X1", "X2", "X3", "X4"),
                        nuisance_propensity = "glm",
                        nuisance_regression = "cox",
                        nuisance_censoring = "cox",
                        nuisance_Qt = "cox",
                        n.folds = NULL) {
```

```r
# Estimate propensity scores
data$e_hat <- estimate_propensity_score(
  data = data,
  treatment_covariates = X.names.propensity,
  type_of_model = nuisance_propensity,
  n.folds = n.folds
)

# Prepare data for censoring model
data$T_obs_tau <- ifelse(data$T_obs >= tau, tau, data$T_obs)

data$censor.status_tau <- 1 - as.numeric((data$T_obs >= tau) |
                                         (data$T_obs < tau & data$status == 1))

data$status_tau <- as.numeric((data$T_obs >= tau) |
                              (data$T_obs < tau & data$status == 1))

# Create unique time grid
Y.grid <- sort(unique(data$T_obs_tau))

if (nuisance_regression == "cox") {
  # Survival formula for Cox model
  outcome <- 'Surv(T_obs, status)'

  # Split data by treatment group
  data.0 <- data %>% filter(A == 0)
  data.1 <- data %>% filter(A == 1)

  # Construct formula
  f <- as.formula(paste(outcome, paste(X.names.outcome, collapse = " + "),
                        sep = " ~ "))

  # Fit Cox model to each subset
  fitS0 <- cph(f, data = data.0, y = TRUE, x = TRUE, times = Y.grid)
  fitS1 <- cph(f, data = data.1, y = TRUE, x = TRUE, times = Y.grid)

  # Predict survival on the time grid
  fit.pred1 <- predictCox(fitS1, newdata = data, times = Y.grid,
                          type = "survival")
  fit.pred0 <- predictCox(fitS0, newdata = data, times = Y.grid,
                          type = "survival")

  # Survival probabilities for each individual
  S_hat1 <- fit.pred1$survival
  S_hat0 <- fit.pred0$survival

} else {
  # Initialize prediction matrices for survival forest
  n <- nrow(data)
  fit.pred1 <- matrix(NA, nrow = n, ncol = length(Y.grid))
  fit.pred0 <- matrix(NA, nrow = n, ncol = length(Y.grid))

  if (n.folds > 1) {
    # Split indices into n subsets
    indices <- split(seq(n), sort(seq(n) %% n.folds))

    for (idx in indices) {
      # Fit survival forest model to each subset
      forest.grf1 <- survival_forest(
        X = as.matrix(data[-idx & data[, "A"] == 1, X.names.outcome]),
        Y = data[-idx & data[, "A"] == 1, "T_obs_tau"],
        D = data[-idx & data[, "A"] == 1, "status_tau"],
        failure.times = Y.grid
      )

      forest.grf0 <- survival_forest(
        X = as.matrix(data[-idx & data[, "A"] == 0, X.names.outcome]),
        Y = data[-idx & data[, "A"] == 0, "T_obs_tau"],
        D = data[-idx & data[, "A"] == 0, "status_tau"],
        failure.times = Y.grid
      )

      # Predict survival probabilities
      fit.pred1[idx, ] <- predict(
        forest.grf1, as.matrix(data[idx, X.names.outcome]),
        failure.times = Y.grid)$predictions

      fit.pred0[idx, ] <- predict(
        forest.grf0, as.matrix(data[idx, X.names.outcome]),
        failure.times = Y.grid)$predictions
    }
  } else {
    # Fit survival forest model without subset splitting
    forest.grf1 <- survival_forest(
      X = as.matrix(data[data[, "A"] == 1, X.names.outcome]),
      Y = data[data[, "A"] == 1, "T_obs_tau"],
      D = data[data[, "A"] == 1, "status_tau"],
      failure.times = Y.grid
    )

    forest.grf0 <- survival_forest(
      X = as.matrix(data[data[, "A"] == 0, X.names.outcome]),
      Y = data[data[, "A"] == 0, "T_obs_tau"],
      D = data[data[, "A"] == 0, "status_tau"],
      failure.times = Y.grid
    )

    # Predict survival probabilities
    fit.pred1 <- predict(
      forest.grf1, as.matrix(data[, X.names.outcome]),
      failure.times = Y.grid)$predictions

    fit.pred0 <- predict(
      forest.grf0, as.matrix(data[, X.names.outcome]),
      failure.times = Y.grid)$predictions
  }

  S_hat1 <- fit.pred1
  S_hat0 <- fit.pred0
}

# Compute area under the survival curve up to tau
data$E_hat1 <- expected_survival(S_hat1, Y.grid)
data$E_hat0 <- expected_survival(S_hat0, Y.grid)

# Compute IPW-weighted residuals
data$IPW_res <- data$E_hat1 * (1 - data$A / data$e_hat) -
  data$E_hat0 * (1 - (1 - data$A) / (1 - data$e_hat))

# Compute AIPCW weights
TDR <- AIPCW(
  data = data,
  tau = tau,
  X.names.censoring = X.names.censoring,
  X.names.outcome = X.names.outcome,
  nuisance_Qt = nuisance_Qt,
  nuisance_censoring = nuisance_censoring,
  n.folds = n.folds
)

data$TDR <- TDR

# Compute AIPCW-weighted residuals
data$AIPCW_w <- data$TDR * (data$A / data$e_hat -
                             (1 - data$A) / (1 - data$e_hat))

# Compute regression residuals
data$reg <- data$E_hat1 - data$E_hat0
data$reg_res <- data$A / data$e_hat * (data$TDR - data$E_hat1) -
  (1 - data$A) / (1 - data$e_hat) * (data$TDR - data$E_hat0)

# Compute estimators
# na.rm = TRUE to remove NA for the mean calculation
AIPTW_AIPCW_IPW_res <- mean(data$AIPCW_w + data$IPW_res, na.rm = TRUE)
AIPTW_AIPCW_reg_res <- mean(data$reg + data$reg_res, na.rm = TRUE)

return(list(AIPTW_AIPCW_reg_res = AIPTW_AIPCW_reg_res,
                       AIPTW_AIPCW_IPW_res = AIPTW_AIPCW_IPW_res))
}
```

This function returns the two estimators introduced previously in Section 4.2.4.3.

In the previous section, we discussed methods for implementing various estimators. However, there are existing packages that can directly compute θ_{RMST} in certain contexts. The next section will introduce these packages and explain the specific conditions under which they are applicable.

5 Available packages to compute θ_{RMST}

To date, there are very few packages that offer functions with a direct application of the methods presented above. The package selection is based on criteria:

- Calculation of ATE or CATE in a survival analysis framework similar to our case study (static treatment assignment and binary treatment, baseline covariates, right-censoring type)
- The package is not archived

The function that requires minimal user intervention is grf's (Tibshirani et al. 2017) causal_survival_forest or survRM2 ("survRM2: Comparing Restricted Mean Survival Time" 2015)'s rmst2. The other packages provide functions that demand significant user effort, such as calculating nuisance models, which account for the largest proportion of error.

5.1 SurvRM2 Packages

The following function is based on the package survRM2 ("survRM2: Comparing Restricted Mean Survival Time" 2015) which allows to compute the RMST under Equation 5 (random treatment assignment) and Equation 2 (independent censoring). It performs two-sample comparisons using the restricted mean survival time (RMST) as a summary measure of the unadjusted survival time distribution presented in Section 3.1.2.

```
library(survRM2)
RMST_survRM2 <- function(data, tau) {

  ATE_pack <- rmst2(data$T_obs, data$status, arm = data$A, tau = tau)

  RMST <- ATE_pack[[5]][1]

  return(RMST)
}
```

5.2 RISCA

RISCA package (Foucher, Le Borgne, and Chatton 2019) allows to compute θ_{RMST} with unadjusted survival function (as in Section 3.1.2), with S_{IPTW} adjusted survival function (as in Section 4.1.1.1) and g-computation, a maximum-likelihood substitution estimator of the g-formula (as in Section 4.2.1.1) (Chatton et al. 2022). The weights for the adjusted IPTW survival function have to be computed before using the function.

- The function rmst() allows to compute the RMST for a given time horizon τ and a survival function. The survival function has to be computed before using it. To be equivalent to

Equation 7, we stratify the survival function by treatment and compute the RMST for treated and control, then we compute θ_{RMST} by doing the difference.

- The function ipw.survival() allows to estimate confounder-adjusted survival curves by weighting the individual contributions by the inverse of the probability to be in the group of treatment (as Equation 17). This function could be also used with other weights such as IPTW-IPCW (as Equation 20) as the weights (propensity score) have to be computed before using it.
- The function gc.survival() allows to estimate the marginal effect of an exposure or a treatment by G-computation for a censored times-to-event, the Q-model being specified by a Cox model. This function is computed like a single learner.

```
# Function to estimate RMST using unadjusted method
RISCA_unadj <- function(data,
                        tau) {
  # Fit survival curves stratified by treatment group
  fit <- survfit(Surv(T_obs, status) ~ A, data = data)
  res <- summary(fit)

  # Calculate RMST for treatment group A=1
  RMST_A1 <- rmst(
    times = res$time[as.character(res$strata) == "A=1"],
    surv.rates = res$surv[as.character(res$strata) == "A=1"],
    max.time = tau,
    type = "s"
  )

  # Calculate RMST for treatment group A=0
  RMST_A0 <- rmst(
    times = res$time[as.character(res$strata) == "A=0"],
    surv.rates = res$surv[as.character(res$strata) == "A=0"],
    max.time = tau,
    type = "s"
  )

  # Compute ATE as the difference in RMST between groups
  ATE_RISCA_unadj <- RMST_A1 - RMST_A0
  return(ATE_RISCA_unadj)
}

# Function to estimate RMST using IPTW method
RISCA_iptw <- function(data,
                       tau,
                       X.names.propensity,
                       nuisance_propensity = "glm",
                       n.folds = NULL) {

  # Estimate propensity scores
  e_hat <- estimate_propensity_score(
    data,
    treatment_covariates = X.names.propensity,
    type_of_model = nuisance_propensity,
    n.folds = n.folds
  )

  # Compute inverse probability weights
  weighted <- (data$A / e_hat) + ((1 - data$A) / (1 - e_hat))

  # Fit weighted survival curves
  IPW_pack <- ipw.survival(
    times = data$T_obs,
    failures = data$status,
    variable = data$A,
    weights = weighted
  )

  # Calculate RMST for treatment group A=1 using weighted survival curve
  RMST_RISCA_A1 <- rmst(
    times = IPW_pack$table.surv$times[IPW_pack$table.surv$variable == 1],
    surv.rates = IPW_pack$table.surv$survival[IPW_pack$table.surv$variable == 1],
    max.time = tau,
    type = "s"
  )

  # Calculate RMST for treatment group A=0 using weighted survival curve
  RMST_RISCA_A0 <- rmst(
    times = IPW_pack$table.surv$times[IPW_pack$table.surv$variable == 0],
    surv.rates = IPW_pack$table.surv$survival[IPW_pack$table.surv$variable == 0],
    max.time = tau,
    type = "s"
  )

  # Compute ATE as the difference in RMST between groups
  ATE_RISCA_IPW <- RMST_RISCA_A1 - RMST_RISCA_A0
  return(ATE_RISCA_IPW)
}

# Function to estimate RMST using G-formula method
RISCA_gf <- function(data,
                     tau,
                     X.names.outcome) {

  # Define the outcome formula for the Cox model
  outcome <- paste(c('Surv(', "T_obs", ',', "status", ')'), collapse = "")
  formula <- as.formula(paste(outcome, paste(c(X.names.outcome, 'A'),
                                             collapse = " + "), sep = " ~ "))

  # Fit the Cox proportional hazards model
  cox.cdt <- coxph(formula, data = data, x = TRUE)
  summary(cox.cdt)

  # Compute the marginal effect of the treatment (ATE) using the G-formula
  gc.ate <- gc.survival(
    object = cox.cdt,
    data = data,
    group = "A",
    times = "T_obs",
    failures = "status",
    max.time = tau,
    iterations = 100,
    effect = "ATE",
    n.cluster = 1
  )

  # Extract the ATE
  ATE_RISCA_gf <- gc.ate$delta[[1]]
  return(ATE_RISCA_gf)
}
```

5.3 Causal survival forest

In this section, the algorithm of causal survival forest will be introduced. It exists other way of using forest algorithm for observational study with conditionally independent censoring such as weighted IPCW causal forest but the causal survival estimator showed better performance (Cui et al. 2023).

5.3.1 Theory of causal survival forest

The causal survival forest (Cui et al. 2023) is an adaptation of the causal forest algorithm of (Athey, Tibshirani, and Wager 2018) in a context of time to event output and censoring.

This estimator is mainly used in the case of heterogeneous treatment effect estimation in observational setting by estimating the conditional average treatment effect (CATE):

$$\theta(x) = E[y(T_i(1)) - y(T_i(0))|X_i = x]$$

The assumption of consistency (Equation 1), conditional independent censoring (Equation 3), positivity of censoring (Equation 4), unconfoundedness of treatment assignment (Equation 14) and positivity of propensity score (Equation 15) play a fundamental role in the identifiability and estimation of the CATE using causal survival forest.

To introduce it, let's begin in a context without censoring. The corresponding estimator adjusts for treatment effect by solving the localized equation:

$$\sum_{i=1}^{n} \alpha_i(x)\psi_\theta(X_i, T_i \wedge \tau, A_i, \hat{e}, \hat{F}) = 0$$

with $\psi_\theta(X_i, T_i \wedge \tau, A_i, \hat{e}, \hat{F}) = [A_i - \hat{e}(X_i)].[T_i \wedge \tau - \hat{F}(X_i) - \theta(A_i - \hat{e}(X_i)]$, θ is the conditional average treatment effect, $\alpha_{bi}(x) = \frac{\mathbf{1}(\{X_i \in L_b(x)\})}{|L_b(x)|}$, $\alpha_i(x) = \frac{1}{B}\sum_{b=1}^{B} \alpha_{bi}(x)$ with $L_b(x)$ the set of training observations falling in the same leaf as x in the tree b. In other words, the weighting α is used to express heterogeneity in θ_{RMST} by measuring the relevance of the i-th sample to fitting θ at x. The weights sum is equal to 1.

In the presence of censoring, another adjustment has to correct the censoring bias. The use of AIPCW transformation introduced in Section 4.2.4.2, enable to complete the expression of $\psi_\theta(X_i, T_i \wedge \tau, A_i, \hat{e}, \hat{F})$:

$$\psi_\theta\left(X_i, T_i \wedge \tau, A_i, \Delta_i^\tau; \hat{e}, \hat{F}, \hat{\lambda}_a^C, \hat{S}_a^C, \hat{Q}_a\right) = \\ \left(\frac{\hat{Q}_{A_i}\left(T_i \wedge \tau \mid X_i\right) + \Delta_i^\tau\left[T_i \wedge \tau - \hat{Q}_{A_i}\left(T_i \wedge \tau \mid X_i\right)\right] - \hat{F}\left(X_i\right) - \theta\left(A_i - \hat{e}\left(X_i\right)\right)}{\hat{S}_{A_i}^C\left(T_i \wedge \tau \mid X_i\right)}\right. \\ \left. - \int_0^{T_i \wedge \tau} \frac{\hat{\lambda}_{A_i}^C\left(s \mid X_i\right)}{\hat{S}_{A_i}^C\left(s \mid X_i\right)}\left[\hat{Q}_{A_i}\left(s \mid X_i\right) - \hat{F}\left(X_i\right) - \theta\left(A_i - \hat{e}\left(X_i\right)\right)\right] ds\right)\left(A_i - \hat{e}\left(X_i\right)\right)$$

where $Q_a(t \mid x) = \mathbb{E}\left[T_i \wedge \tau \mid X_i = x, A_i = a, T_i \wedge \tau > t\right]$ is the conditional expectation of the survival time, $\hat{\lambda}_a^C(t \mid x)$

Exactly than illustrated before, the causal survival forest solves this localized equation:

$$\sum_{i=1}^n \alpha_i(x) \psi_\theta\left(X_i, T_i \wedge \tau, A_i, \Delta_i^\tau; \hat{e}, \hat{F}, \hat{\lambda}_a^C, \hat{S}_a^C, \hat{Q}_a\right) = 0$$

In the method from Athey and Wager (2019), the splitting method relies on favoring split that increase the heterogeneity of the estimates as fast as possible. The following Δ-criterion has to be large:

$$\Delta\left(C_1, C_2\right) := \frac{n_{C_1} n_{C_2}}{n_P^2}\left(\hat{\theta}_{C_1}(\mathcal{J}) - \hat{\theta}_{C_2}(\mathcal{J})\right)^2$$

where $\hat{\theta}_{C_1}$ and $\hat{\theta}_{C_2}$ are solutions to the estimating equation computed in the children. and $n_P = |\{i \in \mathcal{J} : X_i \in P\}|$, the number of observations in the parent and n_{C_j} for the number of observations in each child node.

In reality, an approximate criterion $\widetilde{\Delta}\left(C_1, C_2\right)$ based on gradient approximation is used for computational reason.

5.3.2 Properties of Causal survival forest

This estimator is Neyman-orthogonal in the sense discussed in Chernozhukov et al. (2016), and attains a $1/\sqrt{n}$ rate of convergence for τ under 4-th root rates for the nuisance components, provided we use cross-fitting and that assumptions detailed above hold (Cui et al. 2023; Kennedy 2023).

5.3.2.1 Implementation

This causal survival forest is implemented in using grf package (Tibshirani et al. 2017):

```
# Function to estimate RMST using Causal Survival Random Forest (CSRF)
CSRF <- function(data, X.names, tau) {
  # Select and convert covariates to matrix
  X <- data %>%
    dplyr::select(all_of(X.names)) %>%
    as.matrix()
```

```r
  # Select and convert observed times to matrix
  Y <- data %>%
    dplyr::select(T_obs) %>%
    as.matrix()

  # Select and convert treatment assignment to matrix
  W <- data %>%
    dplyr::select(A) %>%
    as.matrix()

  # Select and convert event status to matrix
  D <- data %>%
    dplyr::select(status) %>%
    as.matrix()

  # Set target and horizon for the causal forest
  target <- "RMST"
  horizon <- tau

  # Fit a causal survival forest
  cf <- causal_survival_forest(X = X, Y = Y, W = W, D = D, horizon = horizon)

  # Predict using the fitted forest
  cf.predict <- predict(cf)

  # Estimate the average treatment effect (ATE)
  ATE_csf <- average_treatment_effect(cf)

  # Return the estimated ATE
  return(ATE_csf[[1]])
}
```

6 Summary of the estimators

In this section, we use tables to summarise the notations and estimators presented and their consistency under mis-specification.

Here's a summary of the notation used in the previous sections:

Table 2: Notations reminder for all estimators

Symbol	Description
X	Covariates
A	Treatment indicator ($A = 1$ for treatment, $A = 0$ for control)
T	Time to event
$T(1), T(0)$	Potential time to event respectively with and without treatment
S_1, S_0	Potential survival curve $(S_a(t) = p(T(a) > t))_{a \in 0,1}$ of the potential time to event
C	Censoring time
$\widetilde{T}$	Observed time ($T \wedge C$)

Symbol	Description
Δ	Censoring indicator (or status) $\mathbb{I}(\{T \leq C\})$
Δ^τ	Censoring indicator of the restricted time (or restricted status) $\mathbb{I}(\{\widetilde{T} > \tau\}) + \mathbb{I}(\{\widetilde{T} \leq \tau\}) \cdot \Delta$
$(t_1, t_2, \ldots, t_D)$	D ordered distinct times to event in the sample
$e(x)$	Propensity score $\mathbb{E}[A \vert X = x]$
$F(a, x)$	$\mathbb{E}[T \wedge \tau \mid X = x, A = a]$
$S(t \vert a, x)$	Conditional survival function, $p[T \wedge \tau > t \vert X = x, A = a]$ for $t \leq \tau$
$S_C(t \vert a, x)$	conditional survival function of the censoring $p(C > t \vert X, A)$ for $t \leq \tau$
$Q_S(t \vert x, a)$	$\mathbb{E}[T \wedge \tau \mid X = x, A = a, T \wedge \tau > t]$

As seen in the previous sections, the estimators do not all have the same nuisance parameters to estimate. The following table summarizes the estimators used in the simulation and the nuisance parameters used to estimate them:

Table 3: Nuisance parameter to compute for all estimators

Estimator	Context of application	Outcome model	Censoring model	Treatment model
Unadjusted KM	RCT & Independent censoring			
————————	———————	———————-	————————	———————
IPCW-KM	RCT & Dependent censoring		✓ (S_c)	
BJ		✓ (Q_S)		
————————	———————	———————-	————————	———————
IPTW-KM	Obs & Independent censoring			✓ (e)
————————	———————	———————-	————————	———————
IPCW-IPTW-KM	Obs & Dependent censoring		✓ (S_c)	✓ (e)
G-formula		✓ (F)		
IPTW-BJ		✓ (Q_S)		✓ (e)
AIPTW-AIPCW		✓ (Q_S, F)	✓ (S_c)	✓ (e)

Also, the estimators do not all have the same sensitivity to the mis-specification of these nuisance models. The table below shows the consistency of the estimators under different mis-specification scenarios

Table 4: Consistency of estimator under model mis-specification. When all the nuisances models are mis-specified none of the estimators is consistent. ✓ indicates consistency of the estimator, ☒ mean that the nuisance model is necessary for the estimator and that it is not consistent when the model is mis-specified and empty boxes indicate that the nuisance model not needed in the estimator thus mis-specfication has no impact.

Estimator	Context of application	mis. outcome model	mis. censoring model	mis. treatment model	mis. outcome and censoring	mis. outcome and treatment	mis. censoring and treatment
Unadjusted KM	RCT & independent censoring	~	~	~	~	~	~
———————	————————-	——————-	——————-	—————-	——————-	——————-	——————-
IPCW-KM	RCT & dependent censoring	~	☒	~	~	~	~
BJ	—	☒	~	~	~	~	~
———————	————————-	——————-	——————-	—————-	——————-	——————-	——————-
IPTW-KM	Obs & independent censoring	~	~	☒	~	~	~
———————	————————-	——————-	——————-	—————-	——————-	——————-	——————-
IPTW-IPCW	Obs & dependent censoring	~	☒	☒	~	~	~
G-formula	—	☒	~	~	~	~	~
IPTW-BJ	—	☒	~	☒	~	~	~
AIPTW-AIPCW	—	✓	✓	✓	☒	☒	✓

As discussed in Section 4.2.4.3.1, the only estimator that remains robust to mis-specification is the AIPTW-AIPCW.

After having presented the estimators and their theoretical properties, we will now apply them to simulation sets.

7 Simulations

7.1 Data generating process

The proposed data generation processes simulate RCT and observational data. Among the RCT and observational data, we propose two versions, one with an independent censoring mechanism and another with conditionally independent censoring. Furthermore, among the observational data, we will consider a simple version (where nuisance parameters will be well estimated by parametric and semi-parametric models such as the Cox model, logistic regression …), a more complex version that will implement non linear relationships between variables and a version with covariate interactions.

7.1.1 RCT

We conduct two simple simulations to simulate RCTs studies, baseline covariates with no time dependency. The first simulation represents a scenario with independent censoring and the second one with conditionally independent censoring.

The time of event and the censoring time (when there is dependency between the censoring time and the covariates) is simulated using the cumulative hazard inversion method for exponential models (details in Annex Section 9).

For the simulation, n samples $(X_i, A_i, C, T_i(0), T_i(1))$ are generated in the following way:

- $X \sim \mathcal{N}\left(\mu = [1, 1, -1, 1]^\top, \Sigma = I_4\right)$.
- $e(X) = 0.5$ (constant) for the propensity score ($P(A = 1|X) = 0.5$).
- $\lambda(0)(X) = 0.01 \cdot \exp\left\{0.5X_1 + 0.5X_2 - 0.5X_3 + 0.5X_4\right\}$ hazard for the event time $T(0)$.
- The hazard for the censoring time C:
  - For scenario 1: $\lambda_c = 0.03$ does not depend on covariates.
  - For scenario 2: $\lambda_c(X) = 0.03 \cdot \exp\left\{0.7X_1 + 0.3X_2 - 0.25X_3 - 0.1X_4 - 0.2A\right\}$.
- $T(1) = T(0) + 10$.
- the event time is $T = AT(1) + (1 - A)T(0)$.
- The observed time is $\widetilde{T} = \min(T, C)$.
- The status is $\Delta = 1(T \leq C)$.
- The threshold time τ is set to 25.

The observed samples are $(X_i, A_i, \Delta_i, \widetilde{T}_i)$ represented previously in Table 1.

```
# scenario:
############ RCT
# RCT1:  Random treatment assignment + independent censoring
# RCT2:  Random treatment assignment + dependent censoring (conditional on X)

simulate_data_RCT <- function(n, mu = c(1, 1, -1, 1),
                              sigma = diag(4),
                              colnames_cov = c("X1", "X2", "X3", "X4"),
                              tau,
                              coefT0 = 0.01,
                              parsS = c(0.5, 0.5, -0.5, 0.5),
                              coefC = 0.03,
                              parsC = c(0.7, 0.3, -0.25, -0.1),
                              parsC_A = c(-0.2),
                              scenario = "RCT2",
                              mis_specification="none") {

  if (scenario == "RCT1") {
    # Generate X from a multivariate normal distribution
    X <- MASS::mvrnorm(n, mu, sigma)
    X <- as.data.frame(X)
    colnames(X) <- colnames_cov

    # Treatment variable selection: all X
    X_treatment <- as.matrix(X)

    # Propensity score: constant for random assignment
    e <- rep(0.5, n)

    # Random treatment assignment
    A <- sapply(e, FUN = function(p) rbinom(1, 1, p))

    # Outcome variable selection: all X
    X_outcome <- as.matrix(X)

    # Simulate the outcome using the cumulative hazard inversion method
    epsilon <- runif(n, min = 1e-8, max = 1)
    T0 <- -log(epsilon) / (coefT0 * exp(X_outcome %*% parsS))

    # Simulate independent censoring time
    epsilon <- runif(n, min = 1e-8, max = 1)
    C <- -log(epsilon) / coefC

    # T(1) = T(0) + 10
    T1 <- T0 + 10

    # True survival time
    T_true <- A * T1 + (1 - A) * T0

    # Observed time
    T_obs <- pmin(T_true, C)

    # Status indicator
    status <- as.numeric(T_true <= C)
    censor.status <- as.numeric(T_true > C)

    # Restricted survival time
    T_obs_tau <- pmin(T_obs, tau)
    status_tau <- as.numeric((T_obs > tau) | (T_obs <= tau & status == 1))

  } else if (scenario == "RCT2") {
    # Generate X from a multivariate normal distribution
    X <- MASS::mvrnorm(n, mu, sigma)
    X <- as.data.frame(X)
    colnames(X) <- c("X1", "X2", "X3", "X4")

    # Treatment variable selection: all X
    X_treatment <- as.matrix(X)

    # Propensity score: constant for random assignment
    e <- rep(0.5, n)

    # Random treatment assignment
    A <- sapply(e, FUN = function(p) rbinom(1, 1, p))

    # Outcome variable selection: all X
    X_outcome <- as.matrix(X)

    # Simulate the outcome using the cumulative hazard inversion method
    epsilon <- runif(n, min = 1e-8, max = 1)
    T0 <- -log(epsilon) / (coefT0 * exp(X_outcome %*% parsS))

    # Simulate dependent censoring time
    X_censoring <- as.matrix(cbind(X,A))
    parsC <- c(parsC,parsC_A)

    epsilon <- runif(n, min = 1e-8, max = 1)
    C <- -log(epsilon) / (coefC * exp(rowSums(X_censoring %*% diag(parsC))))

    # T(1) = T(0) + 10
    T1 <- T0 + 10

    # True survival time
    T_true <- A * T1 + (1 - A) * T0

    # Observed time
    T_obs <- pmin(T_true, C)

    # Status indicator
    status <- as.numeric(T_true <= C)
    censor.status <- as.numeric(T_true > C)

    # Restricted survival time
    T_obs_tau <- pmin(T_obs, tau)
    status_tau <- as.numeric((T_obs > tau) | (T_obs <= tau & status == 1))
  }


  # Combine all data into a single data frame
  data_target_population <- data.frame(X, tau, A, T0, T1, C, T_obs, T_obs_tau,
                                       status, censor.status, status_tau, e)

  return(data_target_population)
}
```

```
# data_rct1 simulate the data from RCT with independent censoring
data_rct1 <- simulate_data_RCT(n=2000,
                               tau=25,
                               scenario="RCT1",
                               coefC = 0.03)

# data_rct2 simulate the data from RCT with dependent censoring
data_rct2 <- simulate_data_RCT(n=2000,
                               tau=25,
                               scenario="RCT2",
                               coefC = 0.03,
                               parsC = c(0.7,0.3,-0.25,-0.1),
                               parsC_A = c(-0.2))


# data_rct2 simulate the data from RCT with dependent censoring
data_rct2 <- simulate_data_RCT(n=2000,
                               tau=25,
                               scenario="RCT2",
                               coefC = 0.002,
                                parsC =  c(2,-4,-5,0.2),
                               parsC_A = c(-0.2))
```

7.1.2 Observational study

In the same way as above, we carried out two simulations of an observational study. The only difference lies in the simulation of the propensity score, which is no longer constant.

For the simulation, n samples $(X_i, A_i, C, T_i(0), T_i(1))$ in the same way than Section 7.1.1, except :

- $\text{logit}\{e(X)\} = -1X_1 - 1X_2 - 2.5X_3 - 1X_4$ for the propensity score (A).
- The hazard for the censoring time C:
  - For scenario 2: $\lambda_c(X) = 0.03 \cdot \exp\{0.7X_1 + 0.3X_2 - 0.25X_3 - 0.1X_4\}$.

The observed samples are $(X_i, A_i, \Delta_i, \widetilde{T}_i)$ represented in Table Table 1.

```
############ Observational
# Obs1:  Treatment assignment dependent on X + independent censoring
# Obs2:  Treatment assignment dependent on X + dependent censoring (conditional on X)

# Function to simulate observational data for two scenarios: Obs1 and Obs2
simulate_data_obs <- function(n,
                              mu = c(1, 1, -1, 1),
                              sigma = diag(4),
                              colnames_cov = c("X1", "X2", "X3", "X4"),
                              tau,
                              coefT0 = 0.01,
                              parsS = c(0.5, 0.5, -0.5, 0.5),
                              parsA = c(-1, -1, -2.5, -1),
                              parsC_A = c(0),
                              coefC = 0.03,
                              parsC = c(0.7, 0.3, -0.25, -0.1),
                              scenario = "Obs2") {

  # Generate covariates X from a multivariate normal distribution
  X <- mvrnorm(n, mu, sigma)
  X <- as.data.frame(X)
  colnames(X) <- colnames_cov

  # Propensity score model based on X
  e <- rowSums(as.matrix(X) %*% diag(parsA))
  e <- plogis(e)  # Transform to probability scale

  # Treatment assignment based on the propensity score
  A <- sapply(e, FUN = function(p) rbinom(n = 1, size = 1, prob = p))

  # Outcome model based on X
  X_outcome <- as.matrix(X)
  epsilon <- runif(n, min = 0.00000001, max = 1)
  T0 <- -log(epsilon) / (coefT0 * exp(X_outcome %*% parsS))

  # Define treatment effect (shift in survival time due to treatment)
  T1 <- T0 + 10

  if (scenario == "Obs1") {
    # Scenario 1: Independent censoring
    C <- -log(runif(n, min = 0.00000001, max = 1)) / coefC

  } else if (scenario == "Obs2") {
    # Scenario 2: Dependent censoring based on X
    X_censoring <- as.matrix(cbind(X,A))
    parsC <- c(parsC,parsC_A)

    C <- -log(runif(n, min = 0.00000001, max = 1)) /
      (coefC * exp(rowSums(X_censoring %*% diag(parsC))))

  } else {
    stop("Invalid scenario. Choose 'Obs1' or 'Obs2'.")
  }

  # Determine the true survival time based on treatment
  T_true <- A * T1 + (1 - A) * T0

  # Observed time is the minimum of the true survival time and censoring time
  T_obs <- pmin(T_true, C)

  # Status indicator: 1 if the event (death) occurred, 0 if censored
  status <- as.numeric(T_true <= C)
```

```
  # Restricted survival time (censored at tau)
  T_obs_tau <- pmin(T_obs, tau)
  status_tau <- as.numeric((T_obs > tau) | (T_obs <= tau & status == 1))

  # Compile the simulated data into a data frame
  DATA_target_population <- data.frame(X, tau, A, T0, T1, C, T_obs, T_obs_tau,
                                       status, status_tau, e)

  return(DATA_target_population)
}
```

```
# Observational data with no informative censoring
data_obs1 <- simulate_data_obs(n=2000,tau=25,scenario="Obs1")

# Observational data simulation with dependent censoring
data_obs2 <- simulate_data_obs(n=2000,tau=25,scenario="Obs2",
                               coefC = 0.03, parsC = c(0.7,0.3,-0.25,-0.1))
```

7.1.3 Observational study with nonlinear relationships

We refer to a simulation where the effect of treatment and censoring cannot be captured by a simple parametric (or semi-parametric) model but well estimated by probability forest for propensity model or survival forest for conditional survival or censoring model.

For the simulation, n samples $(X_i, A_i, C, T_i(0), T_i(1))$ are generated in the following way (similar to the scenario 4 in Cui et al. (2023)):

- $X \sim \mathcal{N}\left(\mu = [1,1,1]^\top, \Sigma = I_3\right)$.
- T is generated from a Poisson distribution with mean $X_2 + X_3 + max(0; X_1 - 0,3)A$.
- C from a Poisson distribution with mean $1 + log(1 + exp(X_3))$.
- The propensity score is $e(x) = [(1 + exp(-X_1))(1 + exp(-X_2))]^{-1}$

The maximum follow-up time is $h = 3$. Note that for subjects with $X_1 < 0,3$, treatment does not affect survival time. The horizon time τ is fixed at 2.

```
simulate_data_complex <- function(n = 2000, tau, parsC = c(0,0,1)) {
  # Load necessary library
  library(MASS)

  # Generate covariates
  X <- mvrnorm(n, mu = c(1, 1, 1), Sigma = diag(3))
  X <- as.data.frame(X)
  colnames(X) <- c("X1", "X2", "X3")

  # Convert data frame to matrix for matrix operations
  X_treatment <- as.matrix(X)

  # Generate treatment
```

```
e <- 1 / ((1 + exp(-X_treatment[, "X1"])) * (1 + exp(-X_treatment[, "X2"])))
A <- sapply(e, function(p) rbinom(1, size = 1, prob = p))

# Generate potential outcomes
lambda_1 <- X_treatment[, "X2"] + X_treatment[, "X3"] +
            pmax(0, X_treatment[, "X1"] - 0.3) * 1
lambda_0 <- X_treatment[, "X2"] + X_treatment[, "X3"]

T1 <- rpois(n, lambda_1)
T0 <- rpois(n, lambda_0)

T1[is.na(T1)] <- 0
T0[is.na(T0)] <- 0

T_true <- T1 * A + T0 * (1 - A)

# Generate censoring time
lambda_C <- 1 + log(1 + exp(parsC[1]*X_treatment[, "X1"] + parsC[2]*X_treatment[, "
C <- rpois(n, lambda_C)

# Observed time and status
T_obs <- pmin(T_true, C)
status <- as.numeric(T_true <= C)
censor_status <- as.numeric(T_true > C)

# Restricted survival time
T_obs_tau <- pmin(T_obs, tau)
status_tau <- as.numeric((T_obs > tau) | (T_obs <= tau & status == 1))

# Create the final data frame
DATA_target_population <- data.frame(
  X1 = X$X1,
  X2 = X$X2,
  X3 = X$X3,
  tau = tau,
  A = A,
  T1 = T1,
  T0 = T0,
  T_true = T_true,
  C = C,
  T_obs = T_obs,
  T_obs_tau = T_obs_tau,
  status = status,
  censor_status = censor_status,
  status_tau = status_tau,
  e = e
)

return(DATA_target_population)
}
```

7.1.4 Observational study with covariates interaction

For this simulation with covariates interaction, n samples $(X_i, A_i, C, T_i(0), T_i(1))$ are generated in the following way:

- $X \sim \mathcal{N}\left(\mu = [1, 1, -1, 1, -2, -5]^\top, \Sigma = I_6\right)$.
- T is generated from a Poisson distribution with mean $X_2 + X_3 + max(0; X_1 - 0, 3)A$.
- C from a Poisson distribution with mean $1 + log(1 + exp(X_3))$.
- The propensity score is $e(x) = plogis$

```
# DGP for mis-specification

simulate_data_mis <- function(n,
                              mu = c(1, 1, -1, 1, -2, -5),
                              sigma =  matrix(c(1, 0, 0, 0, 0, 0,
                                               0, 1, 0, 0, 0, 0,
                                               0, 0, 1, 0, 0, 0,
                                               0, 0, 0, 1, 0, 0,
                                               0, 0, 0, 0, 1, 0,
                                               0, 0, 0, 0, 0, 1),
                                              nrow = 6, byrow = TRUE),
                              colnames_cov = c("X1", "X2", "X3", "X4", "X5", "X6"),
                              parsA = matrix(c(0.1, 0.05, -0.05, 0, 0, 0,
                                               0.05, -0.2, 0, 0.05, 0, 0,
                                               -0.05, 0, 0.3, -0.05, 0, 0,
                                               0, 0.05, -0.05, -0.1, 0, 0,
                                               0, 0, 0, 0, 0, 0,
                                               0, 0, 0, 0, 0, 0),
                                             nrow = 6, byrow = TRUE),
                              tau) {

  # Generate X from a multivariate normal distribution
  X <- MASS::mvrnorm(n, mu, sigma)
  X <- as.data.frame(X)
  colnames(X) <- colnames_cov

  # Treatment variable selection: all X
  X_treatment <- as.matrix(X)

  # Propensity score model based on X
     # Coefficients linéaires pour X1, X2, X3, X4
   # Coefficients pour interactions X1:X2, X1:X3, X2:X4, X3:X4

# Calcul du score de propension avec interactions
  e <- rowSums(as.matrix(X) %*% parsA)
# Transformer en échelle de probabilité
  e <- plogis(e)

  # Treatment assignment based on the propensity score
  A <- sapply(e, FUN = function(p) rbinom(n = 1, size = 1, prob = p))

  # Outcome variable selection: all X
  X_outcome <- as.matrix(X)

  lambda <- X[,1]2 + X[,2]2 + X[,1] * X[,2] + X[,1] * X[,3] + X[,2] * X[,4]

  # Simulate the outcome using the cumulative hazard inversion method
  epsilon <- runif(n, min = 1e-8, max = 1)
  T0 <- -log(epsilon) / lambda
  T0 <- pmax(T0, 1e-8)

  # Simulate independent censoring time
  censoring_lambda <- abs(2*X[,1]2 - X[,2]2 + 1.5*X[,1] * X[,3] - X[,2] * X[,4])
  epsilon <- runif(n, min = 1e-8, max = 1)
  C <- -log(epsilon) / censoring_lambda
  C <- pmax(C, 1e-8)

  # T(1) = T(0) + 10
  T1 <- T0 + 0.1

  # True survival time
  T_true <- A * T1 + (1 - A) * T0

  # Observed time
  T_obs <- pmin(T_true, C)

  # Status indicator
  status <- as.numeric(T_true <= C)
  censor.status <- as.numeric(T_true > C)

  # Restricted survival time
  T_obs_tau <- pmin(T_obs, tau)
  status_tau <- as.numeric((T_obs > tau) | (T_obs <= tau & status == 1))
    # Compile the simulated data into a data frame
  DATA_target_population <- data.frame(X, tau, A, T0, T1, C, T_obs, T_obs_tau,
                                       status, status_tau, e)

  return(DATA_target_population)
}
```

```
mis <- simulate_data_mis(n=1000,tau=0.45)
summary(mis)
```

```
     X1                 X2                 X3                 X4
```

```
 Min.   :-2.1507   Min.   :-1.8085   Min.   :-3.9403   Min.   :-2.3465  
 1st Qu.: 0.3736   1st Qu.: 0.3457   1st Qu.:-1.7170   1st Qu.: 0.3091  
 Median : 1.0151   Median : 0.9768   Median :-1.0223   Median : 0.9587  
 Mean   : 1.0104   Mean   : 1.0154   Mean   :-1.0253   Mean   : 0.9899  
 3rd Qu.: 1.6638   3rd Qu.: 1.6833   3rd Qu.:-0.3412   3rd Qu.: 1.7100  
 Max.   : 3.8680   Max.   : 4.5686   Max.   : 2.3546   Max.   : 3.9184  
       X5               X6               tau               A        
 Min.   :-5.207   Min.   :-8.667   Min.   :0.45   Min.   :0.000  
 1st Qu.:-2.743   1st Qu.:-5.673   1st Qu.:0.45   1st Qu.:0.000  
 Median :-1.980   Median :-5.029   Median :0.45   Median :0.000  
 Mean   :-2.006   Mean   :-5.024   Mean   :0.45   Mean   :0.432  
 3rd Qu.:-1.250   3rd Qu.:-4.357   3rd Qu.:0.45   3rd Qu.:1.000  
 Max.   : 0.801   Max.   :-2.095   Max.   :0.45   Max.   :1.000  
       T0                  T1                  C                  T_obs       
 Min.   :  0.00000   Min.   :  0.1000   Min.   :  0.00004   Min.   : 0.00000  
 1st Qu.:  0.04017   1st Qu.:  0.1402   1st Qu.:  0.08103   1st Qu.: 0.03777  
 Median :  0.14317   Median :  0.2432   Median :  0.23684   Median : 0.10612  
 Mean   :  1.09025   Mean   :  1.1903   Mean   :  1.33284   Mean   : 0.26790  
 3rd Qu.:  0.41258   3rd Qu.:  0.5126   3rd Qu.:  0.81286   3rd Qu.: 0.23747  
 Max.   :150.05873   Max.   :150.1587   Max.   :116.25138   Max.   :12.54542  
   T_obs_tau            status          status_tau           e         
 Min.   :0.00000   Min.   :0.000   Min.   :0.000   Min.   :0.2421  
 1st Qu.:0.03777   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.3827  
 Median :0.10612   Median :1.000   Median :1.000   Median :0.4257  
 Mean   :0.15848   Mean   :0.509   Mean   :0.572   Mean   :0.4257  
 3rd Qu.:0.23747   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:0.4685  
 Max.   :0.45000   Max.   :1.000   Max.   :1.000   Max.   :0.6334  
```

```
group_0 <- mis %>%
  dplyr:: filter(A == 0)%>%
  dplyr:: select(X1,X2,X3,X4,C,T1,T0,status_tau,T_tild=T_obs)

group_1 <- mis %>%
  dplyr:: filter(A == 1)%>%
  dplyr:: select(X1,X2,X3,X4,C,T1,T0,status_tau,T_tild=T_obs)

# Summary statistics
summary_group_0 <- summary(group_0)
summary_group_1 <- summary(group_1)

print(paste("Descriptive statistics for group A=0:  ",nrow(group_0)))
```

```
[1] "Descriptive statistics for group A=0:   568"
```

```
print(summary_group_0)
```

```
       X1                X2                X3                X4         
 Min.   :-2.1507   Min.   :-1.7281   Min.   :-3.9403   Min.   :-1.9070  
 1st Qu.: 0.2929   1st Qu.: 0.3801   1st Qu.:-1.8305   1st Qu.: 0.3102  
 Median : 0.9713   Median : 0.9598   Median :-1.1043   Median : 1.0268  
 Mean   : 0.9180   Mean   : 1.0333   Mean   :-1.1072   Mean   : 1.0279
 3rd Qu.: 1.5860   3rd Qu.: 1.6694   3rd Qu.:-0.4812   3rd Qu.: 1.7990
 Max.   : 3.1871   Max.   : 4.5686   Max.   : 2.3546   Max.   : 3.5219
       C                   T1                 T0              status_tau
 Min.   :  0.00124   Min.   :  0.1000   Min.   :  0.00000   Min.   :0.0000
 1st Qu.:  0.08563   1st Qu.:  0.1411   1st Qu.:  0.04115   1st Qu.:0.0000
 Median :  0.24782   Median :  0.2493   Median :  0.14932   Median :1.0000
 Mean   :  1.49889   Mean   :  1.2040   Mean   :  1.10400   Mean   :0.6496
 3rd Qu.:  0.86343   3rd Qu.:  0.4991   3rd Qu.:  0.39913   3rd Qu.:1.0000
 Max.   :116.25138   Max.   :150.1587   Max.   :150.05873   Max.   :1.0000
     T_tild
 Min.   : 0.00000
 1st Qu.: 0.02229
 Median : 0.08125
 Mean   : 0.26627
 3rd Qu.: 0.21756
 Max.   :12.54542
```

```
print(paste("Descriptive statistics for group A=1:  ",nrow(group_1)))
```

```
[1] "Descriptive statistics for group A=1:   432"
```

```
print(summary_group_1)
```

```
       X1                X2                X3                X4
 Min.   :-1.5951   Min.   :-1.8085   Min.   :-3.9182   Min.   :-2.3465
 1st Qu.: 0.4464   1st Qu.: 0.2633   1st Qu.:-1.6539   1st Qu.: 0.2769
 Median : 1.1222   Median : 1.0258   Median :-0.8833   Median : 0.8754
 Mean   : 1.1318   Mean   : 0.9919   Mean   :-0.9176   Mean   : 0.9399
 3rd Qu.: 1.8580   3rd Qu.: 1.7031   3rd Qu.:-0.1766   3rd Qu.: 1.6389
 Max.   : 3.8680   Max.   : 4.2860   Max.   : 1.7908   Max.   : 3.9184
       C                   T1                 T0              status_tau
 Min.   : 0.000045   Min.   :  0.1000   Min.   :  0.00000   Min.   :0.0000
 1st Qu.: 0.073958   1st Qu.:  0.1374   1st Qu.:  0.03741   1st Qu.:0.0000
 Median : 0.225836   Median :  0.2352   Median :  0.13522   Median :0.0000
 Mean   : 1.114500   Mean   :  1.1722   Mean   :  1.07217   Mean   :0.4699
 3rd Qu.: 0.748010   3rd Qu.:  0.5431   3rd Qu.:  0.44313   3rd Qu.:1.0000
 Max.   :30.169166   Max.   :124.3334   Max.   :124.23342   Max.   :1.0000
     T_tild
 Min.   :0.000045
 1st Qu.:0.073958
 Median :0.129099
 Mean   :0.270038
 3rd Qu.:0.252867
 Max.   :5.587860
```

7.2 Data description

This section will present the data description of the presented simulation in the Section 7.1.1, Section 7.1.2 and Section 7.1.3 to enhance their characteristics.

7.2.1 RCT with independent censoring (RCT scenario 1)

The summary by group of treatment of the generated (observed and unobserved) data set RCT with independent censoring (in Section 7.1.1) is displayed below:

```
[1] "Descriptive statistics for group A=0:   1004"
```

```
       X1                X2                X3                X4         
 Min.   :-2.5819   Min.   :-2.5121   Min.   :-4.0221   Min.   :-2.0244  
 1st Qu.: 0.2992   1st Qu.: 0.2789   1st Qu.:-1.6441   1st Qu.: 0.3567  
 Median : 0.9989   Median : 0.9967   Median :-1.0485   Median : 0.9759  
 Mean   : 0.9954   Mean   : 0.9611   Mean   :-0.9925   Mean   : 0.9865  
 3rd Qu.: 1.6996   3rd Qu.: 1.6877   3rd Qu.:-0.3009   3rd Qu.: 1.6703  
 Max.   : 3.6611   Max.   : 4.3149   Max.   : 1.7407   Max.   : 4.2426  
       C                  T1               T0                status      
 Min.   :  0.00015   Min.   : 10.01   Min.   :  0.0082   Min.   :0.0000  
 1st Qu.: 10.53987   1st Qu.: 13.20   1st Qu.:  3.2027   1st Qu.:0.0000  
 Median : 25.68229   Median : 19.06   Median :  9.0635   Median :1.0000  
 Mean   : 35.89328   Mean   : 32.60   Mean   : 22.6048   Mean   :0.6843  
 3rd Qu.: 50.61685   3rd Qu.: 34.26   3rd Qu.: 24.2627   3rd Qu.:1.0000  
 Max.   :255.22184   Max.   :432.75   Max.   :422.7543   Max.   :1.0000  
     T_tild         
 Min.   :  0.00015  
 1st Qu.:  2.39790  
 Median :  6.31256  
 Mean   : 11.26565  
 3rd Qu.: 14.72062  
 Max.   :154.38944  
```

```
[1] "Descriptive statistics for group A=1:   996"
```

```
       X1                X2                X3                X4         
 Min.   :-2.2623   Min.   :-2.6415   Min.   :-4.0367   Min.   :-1.9487  
 1st Qu.: 0.3024   1st Qu.: 0.2927   1st Qu.:-1.6225   1st Qu.: 0.2906  
 Median : 1.0006   Median : 0.9633   Median :-1.0385   Median : 0.9556  
 Mean   : 0.9404   Mean   : 0.9759   Mean   :-0.9967   Mean   : 0.9779  
 3rd Qu.: 1.6155   3rd Qu.: 1.6894   3rd Qu.:-0.3372   3rd Qu.: 1.6063  
 Max.   : 3.8497   Max.   : 3.9792   Max.   : 2.0577   Max.   : 3.8953  
       C                  T1               T0                status      
 Min.   :  0.09123   Min.   : 10.01   Min.   :  0.0051   Min.   :0.0000  
 1st Qu.:  9.39562   1st Qu.: 12.96   1st Qu.:  2.9601   1st Qu.:0.0000  
 Median : 22.77789   Median : 18.85   Median :  8.8478   Median :1.0000  
 Mean   : 33.70311   Mean   : 35.23   Mean   : 25.2284   Mean   :0.5161  
 3rd Qu.: 48.74494   3rd Qu.: 32.80   3rd Qu.: 22.7990   3rd Qu.:1.0000  
 Max.   :255.94921   Max.   :934.11   Max.   :924.1055   Max.   :1.0000  
     T_tild         
 Min.   :  0.09123  
 1st Qu.:  9.39562  
 Median : 13.26062  
 Mean   : 16.44041  
 3rd Qu.: 19.87937  
 Max.   :107.78376  
```

The tables summarize the covariates $X1$, $X2$, $X3$, and $X4$, the censoring time C, the true time T1 when all observations receive treatment $A = 1$ (T(1)), the true time T0 when all observations receive treatment $A = 0$ (T(0)), the event status (1 if the event occurs, 0 if censored), and $\tilde{T}$, the observed time $min(C, T)$.

Covariates are balanced between groups, and censoring times are the same (independent censoring). However, there are more censored observations in the treated group ($A = 1$) than in the control group ($A = 0$). This is due to the higher instantaneous hazard of the event in the treated group (with $T_1 = T_0 + 10$) compared to the constant hazard of censoring.

7.2.2 RCT with conditionally independent censoring (RCT scenario 2)

The summary of the generated (observed and unobserved) data set RCT with conditionally independent censoring (in Section 7.1.1) stratified by treatment is displayed below. As a reminder, the difference between the RCT scenario 1 and 2 is that the censoring time is dependent of the covariates.

```
[1] "Descriptive statistics for group A=0:  1017"
```

```
       X1                X2                X3                X4         
 Min.   :-2.4268   Min.   :-2.3300   Min.   :-4.6121   Min.   :-1.6562  
 1st Qu.: 0.3347   1st Qu.: 0.3007   1st Qu.:-1.7572   1st Qu.: 0.3066  
 Median : 1.0485   Median : 0.9851   Median :-1.0348   Median : 0.9918  
 Mean   : 1.0152   Mean   : 0.9809   Mean   :-1.0287   Mean   : 0.9991  
 3rd Qu.: 1.6885   3rd Qu.: 1.6233   3rd Qu.:-0.2572   3rd Qu.: 1.6750  
 Max.   : 3.7386   Max.   : 4.3895   Max.   : 2.5533   Max.   : 4.0856  
       C                   T1               T0               status      
 Min.   :0.000e+00   Min.   : 10.00   Min.   :  0.0019   Min.   :0.0000  
 1st Qu.:0.000e+00   1st Qu.: 13.17   1st Qu.:  3.1718   1st Qu.:0.0000  
 Median :7.000e+00   Median : 18.53   Median :  8.5322   Median :0.0000  
 Mean   :5.546e+10   Mean   : 31.55   Mean   : 21.5450   Mean   :0.4926  
 3rd Qu.:7.260e+02   3rd Qu.: 32.55   3rd Qu.: 22.5534   3rd Qu.:1.0000  
 Max.   :5.628e+13   Max.   :476.72   Max.   :466.7175   Max.   :1.0000  
   status_tau         T_tild        
 Min.   :0.0000   Min.   :  0.0000  
 1st Qu.:0.0000   1st Qu.:  0.0641  
 Median :1.0000   Median :  1.7182  
 Mean   :0.5025   Mean   : 10.0275  
 3rd Qu.:1.0000   3rd Qu.:  8.8023  
 Max.   :1.0000   Max.   :321.2499  
```

```
[1] "Descriptive statistics for group A=1:  983"
```

```
       X1                X2                X3                X4         
 Min.   :-2.2557   Min.   :-2.0405   Min.   :-4.1515   Min.   :-1.7082  
 1st Qu.: 0.2823   1st Qu.: 0.3112   1st Qu.:-1.6233   1st Qu.: 0.3196  
 Median : 0.9674   Median : 1.0016   Median :-0.9310   Median : 1.0202  
 Mean   : 0.9721   Mean   : 1.0152   Mean   :-0.9558   Mean   : 1.0150  
 3rd Qu.: 1.6457   3rd Qu.: 1.7202   3rd Qu.:-0.3168   3rd Qu.: 1.6728  
 Max.   : 4.5763   Max.   : 3.9533   Max.   : 2.3873   Max.   : 4.1300  
       C                   T1               T0               status      
 Min.   :0.000e+00   Min.   : 10.00   Min.   :  0.0018   Min.   :0.0000  
 1st Qu.:0.000e+00   1st Qu.: 12.75   1st Qu.:  2.7546   1st Qu.:0.0000  
 Median :1.700e+01   Median : 19.49   Median :  9.4890   Median :0.0000  
 Mean   :1.469e+08   Mean   : 33.16   Mean   : 23.1644   Mean   :0.4771
 3rd Qu.:2.122e+03   3rd Qu.: 35.01   3rd Qu.: 25.0111   3rd Qu.:1.0000
 Max.   :1.416e+11   Max.   :735.07   Max.   :725.0715   Max.   :1.0000
   status_tau          T_tild
 Min.   :0.0000   Min.   :  0.0000
 1st Qu.:0.0000   1st Qu.:  0.2272
 Median :0.0000   Median : 10.4855
 Mean   :0.4944   Mean   : 17.9511
 3rd Qu.:1.0000   3rd Qu.: 20.7128
 Max.   :1.0000   Max.   :735.0715
```

Covariates are balanced between the two groups. However, censoring times differ between groups due to conditionally independent censoring based on covariates. There are more censored observations in the treated group ($A = 1$) compared to the control group ($A = 0$). The summary statistics do not reveal the difference between independent and dependent censoring. Dependent censoring affects the rate of censoring among sub-groups without necessarily changing the overall level of censoring.

7.2.3 Observational study with independent censoring (Obs scenario 1)

The summary of the generated (observed and unobserved) data set observational study with independent censoring (in Section 7.1.2) stratified by treatment is displayed below to enhance the difference with the other scenario.

```
[1] "Descriptive statistics for group A=0:   1157"
```

```
       X1                 X2                 X3                  X4
 Min.   :-1.8009   Min.   :-1.9985   Min.   :-3.32798   Min.   :-1.7649
 1st Qu.: 0.5641   1st Qu.: 0.5386   1st Qu.:-1.05456   1st Qu.: 0.5412
 Median : 1.2603   Median : 1.2071   Median :-0.54328   Median : 1.2490
 Mean   : 1.2425   Mean   : 1.1976   Mean   :-0.50370   Mean   : 1.2324
 3rd Qu.: 1.9081   3rd Qu.: 1.8527   3rd Qu.: 0.02922   3rd Qu.: 1.9138
 Max.   : 4.8590   Max.   : 4.3389   Max.   : 2.25228   Max.   : 4.1068
       C                  T1                T0                 status
 Min.   :  0.073   Min.   : 10.00   Min.   :  0.0006   Min.   :0.0000
 1st Qu.:  9.172   1st Qu.: 12.55   1st Qu.:  2.5546   1st Qu.:0.0000
 Median : 23.352   Median : 16.98   Median :  6.9789   Median :1.0000
 Mean   : 32.044   Mean   : 30.67   Mean   : 20.6651   Mean   :0.6975
 3rd Qu.: 43.789   3rd Qu.: 29.86   3rd Qu.: 19.8565   3rd Qu.:1.0000
 Max.   :195.332   Max.   :540.94   Max.   :530.9352   Max.   :1.0000
     T_tild
 Min.   :  0.00058
 1st Qu.:  1.96094
 Median :  5.14433
 Mean   : 10.07490
 3rd Qu.: 11.78254
 Max.   :114.47224
```

```
[1] "Descriptive statistics for group A=1:   843"
```

```
       X1                 X2                  X3                X4
 Min.   :-2.3010   Min.   :-2.83740   Min.   :-4.151   Min.   :-1.9836
 1st Qu.: 0.0546   1st Qu.: 0.08575   1st Qu.:-2.199   1st Qu.: 0.1672
 Median : 0.6766   Median : 0.75497   Median :-1.647   Median : 0.7905
 Mean   : 0.7254   Mean   : 0.77081   Mean   :-1.652   Mean   : 0.7934  
 3rd Qu.: 1.3397   3rd Qu.: 1.49550   3rd Qu.:-1.093   3rd Qu.: 1.4306  
 Max.   : 3.6487   Max.   : 3.53143   Max.   : 0.621   Max.   : 4.2590  
       C                   T1               T0                status      
 Min.   :  0.00592   Min.   : 10.01   Min.   :  0.0136   Min.   :0.0000  
 1st Qu.: 10.10487   1st Qu.: 12.93   1st Qu.:  2.9292   1st Qu.:0.0000  
 Median : 23.94352   Median : 18.42   Median :  8.4170   Median :0.0000  
 Mean   : 34.22613   Mean   : 32.46   Mean   : 22.4646   Mean   :0.4994  
 3rd Qu.: 46.48944   3rd Qu.: 33.63   3rd Qu.: 23.6304   3rd Qu.:1.0000  
 Max.   :279.86477   Max.   :832.34   Max.   :822.3368   Max.   :1.0000  
     T_tild         
 Min.   :  0.00592  
 1st Qu.: 10.02526  
 Median : 13.30012  
 Mean   : 16.63317  
 3rd Qu.: 20.16314  
 Max.   :122.32726  
```

The covariates between the two groups of treatment are unbalanced because of dependent treatment assignation. The mean of $X1$, $X2$, $X3$ and $X4$ is bigger in the control group than in the treated group. The censoring times have the same distribution (independent censoring). There are more censored observation in the treated group (A=1) than in the control group (A=0) for the same reason than in the RCT scenario.

7.2.4 Observational study with conditionally independent censoring (Obs scenario 2)

The summary of the generated (observed and unobserved) data set Observational study with conditionally independent censoring (in Section 7.1.2) stratified by treatment is displayed below. As a reminder, the difference between the observational scenario 1 and 2 is that the censoring time is dependent of the covariates.

```
[1] "Descriptive statistics for group A=0:   1148"
```

```
       X1                 X2                 X3                  X4         
 Min.   :-2.9608   Min.   :-2.0788   Min.   :-3.26244   Min.   :-2.3498  
 1st Qu.: 0.5271   1st Qu.: 0.5365   1st Qu.:-1.04880   1st Qu.: 0.5755  
 Median : 1.1936   Median : 1.1840   Median :-0.50796   Median : 1.2202  
 Mean   : 1.1708   Mean   : 1.1844   Mean   :-0.48580   Mean   : 1.2262  
 3rd Qu.: 1.8007   3rd Qu.: 1.8476   3rd Qu.: 0.05169   3rd Qu.: 1.8539  
 Max.   : 5.0260   Max.   : 4.0387   Max.   : 2.28333   Max.   : 4.8956  
       C                   T1                 T0                 status      
 Min.   :  0.0023   Min.   :  10.00   Min.   :   0.0035   Min.   :0.0000  
 1st Qu.:  2.2240   1st Qu.:  12.72   1st Qu.:   2.7198   1st Qu.:0.0000  
 Median :  6.2227   Median :  18.68   Median :   8.6756   Median :0.0000  
 Mean   : 14.5069   Mean   :  35.10   Mean   :  25.0975   Mean   :0.4451  
 3rd Qu.: 16.8693   3rd Qu.:  33.71   3rd Qu.:  23.7135   3rd Qu.:1.0000  
 Max.   :474.0298   Max.   :1086.23   Max.   :1076.2283   Max.   :1.0000  
   status_tau          T_obs                 e           
 Min.   :0.0000   Min.   :  0.00233   Min.   :0.000047  
 1st Qu.:0.0000   1st Qu.:  1.25975   1st Qu.:0.020126  
 Median :0.0000   Median :  3.45205   Median :0.097693  
 Mean   :0.4869   Mean   :  8.03617   Mean   :0.202671  
 3rd Qu.:1.0000   3rd Qu.:  9.18200   3rd Qu.:0.312103  
 Max.   :1.0000   Max.   :157.30490   Max.   :0.994291  
```

```
[1] "Descriptive statistics for group A=1:   852"
```

```
       X1                X2                X3                X4         
 Min.   :-2.4447   Min.   :-2.5303   Min.   :-4.3781   Min.   :-1.9567  
 1st Qu.: 0.1274   1st Qu.: 0.1341   1st Qu.:-2.2291   1st Qu.: 0.1086  
 Median : 0.7615   Median : 0.8209   Median :-1.6710   Median : 0.7839  
 Mean   : 0.7528   Mean   : 0.7966   Mean   :-1.6691   Mean   : 0.7531  
 3rd Qu.: 1.3815   3rd Qu.: 1.4049   3rd Qu.:-1.1335   3rd Qu.: 1.4255  
 Max.   : 3.9001   Max.   : 3.8710   Max.   : 0.8336   Max.   : 4.0204  
       C                  T1               T0                status      
 Min.   :  0.0037   Min.   : 10.01   Min.   :  0.0063   Min.   :0.0000  
 1st Qu.:  2.8219   1st Qu.: 13.20   1st Qu.:  3.2026   1st Qu.:0.0000  
 Median :  8.6565   Median : 19.15   Median :  9.1477   Median :0.0000  
 Mean   : 17.3327   Mean   : 32.48   Mean   : 22.4786   Mean   :0.2077  
 3rd Qu.: 18.5587   3rd Qu.: 34.43   3rd Qu.: 24.4296   3rd Qu.:0.0000  
 Max.   :490.5082   Max.   :437.40   Max.   :427.4008   Max.   :1.0000  
   status_tau       T_obs                e          
 Min.   :0.00   Min.   :  0.0037   Min.   :0.00491  
 1st Qu.:0.00   1st Qu.:  2.8219   1st Qu.:0.58938  
 Median :0.00   Median :  8.6565   Median :0.85791  
 Mean   :0.25   Mean   : 11.5889   Mean   :0.74947  
 3rd Qu.:0.25   3rd Qu.: 14.8659   3rd Qu.:0.96011  
 Max.   :1.00   Max.   :129.7375   Max.   :0.99976  
```

The covariates between the two groups are unbalanced. The censoring time is dependent on the covariates also, as the covariates are unbalanced between the two groups, the censoring time is also unbalanced. In particular, the mean of $X1$, $X2$, $X3$ and $X4$ is bigger in the control group than in the treated group. Also, the number of events is bigger in the control than treated group.

7.2.5 Observational study with nonlinear relationships and conditionally independent censoring (Non parametric scenario)

The summary of the generated (observed and unobserved) data set complex observational study (conditionally independent censoring) stratified by treatment is displayed below.

```
[1] "Descriptive statistics for group A=0:   1010"
```

```
       X1                 X2                 X3                C        
 Min.   :-2.16240   Min.   :-2.17115   Min.   :-2.0001   Min.   : 0.00  
 1st Qu.: 0.05797   1st Qu.: 0.08377   1st Qu.: 0.3329   1st Qu.: 1.00  
 Median : 0.79415   Median : 0.73163   Median : 0.8981   Median : 2.00  
 Mean   : 0.77888   Mean   : 0.76304   Mean   : 0.9490   Mean   : 2.32  
 3rd Qu.: 1.47214   3rd Qu.: 1.44729   3rd Qu.: 1.5400   3rd Qu.: 3.00  
 Max.   : 4.21249   Max.   : 4.16871   Max.   : 4.2701   Max.   :10.00  
       T1               T0              status           T_obs      
 Min.   : 0.000   Min.   : 0.000   Min.   :0.0000   Min.   :0.000  
 1st Qu.: 1.000   1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.:0.000  
 Median : 2.000   Median : 1.000   Median :1.0000   Median :1.000  
 Mean   : 2.441   Mean   : 1.798   Mean   :0.7149   Mean   :1.152  
 3rd Qu.: 4.000   3rd Qu.: 3.000   3rd Qu.:1.0000   3rd Qu.:2.000  
 Max.   :12.000   Max.   :11.000   Max.   :1.0000   Max.   :8.000
   status_tau          e
 Min.   :0.0000   Min.   :0.04591
 1st Qu.:1.0000   1st Qu.:0.28346
 Median :1.0000   Median :0.41892
 Mean   :0.7752   Mean   :0.42402
 3rd Qu.:1.0000   3rd Qu.:0.54921
 Max.   :1.0000   Max.   :0.89462
```

```
[1] "Descriptive statistics for group A=1:   990"
```

```
       X1                X2                X3                C
 Min.   :-1.4970   Min.   :-1.670   Min.   :-2.5028   Min.   : 0.000
 1st Qu.: 0.6144   1st Qu.: 0.570   1st Qu.: 0.3067   1st Qu.: 1.000
 Median : 1.2975   Median : 1.204   Median : 0.9788   Median : 2.000
 Mean   : 1.3142   Mean   : 1.198   Mean   : 1.0021   Mean   : 2.434
 3rd Qu.: 1.9752   3rd Qu.: 1.819   3rd Qu.: 1.7061   3rd Qu.: 3.000
 Max.   : 4.1971   Max.   : 4.101   Max.   : 5.1139   Max.   :10.000
       T1               T0              status           T_obs
 Min.   : 0.000   Min.   : 0.000   Min.   :0.0000   Min.   : 0.000
 1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:0.0000   1st Qu.: 1.000
 Median : 3.000   Median : 2.000   Median :0.0000   Median : 2.000
 Mean   : 3.321   Mean   : 2.232   Mean   :0.4758   Mean   : 1.815
 3rd Qu.: 5.000   3rd Qu.: 3.000   3rd Qu.:1.0000   3rd Qu.: 3.000
 Max.   :13.000   Max.   :11.000   Max.   :1.0000   Max.   :10.000
   status_tau          e
 Min.   :0.0000   Min.   :0.08642
 1st Qu.:0.0000   1st Qu.:0.42719
 Median :1.0000   Median :0.56179
 Mean   :0.6343   Mean   :0.55223
 3rd Qu.:1.0000   3rd Qu.:0.68076
 Max.   :1.0000   Max.   :0.92101
```

The observations are the same than the previous scenario: The covariates and the censoring time between the two groups are unbalanced.

To be able to evaluate the estimators, we need to know the true θ_{RMST} at time τ.

7.2.6 True value of RMST

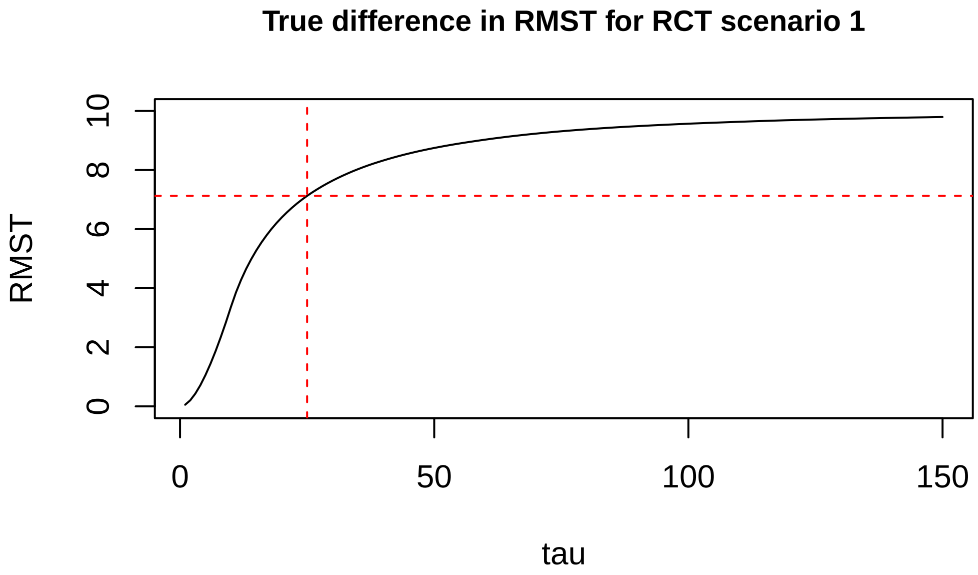
θ_{RMST} is a time-dependent value. Therefore, the ground truth for θ_{RMST} must be calculated at the required restricted time.

The following implementation computes the true θ_{RMST} for each of the previous simulations. This is feasible because the simulations produce data that include hypothetical scenarios not observable in real life (within the data frame, we have access to T1, the outcome if the patient had been treated, and T0, the outcome if the patient had not been treated). Thus, calculating of the true θ_{RMST} becomes straightforward:

```r
# Function to calculate ground truth for RCT and Observational datasets
ground_truth <- function(tau,
                         data) {
```

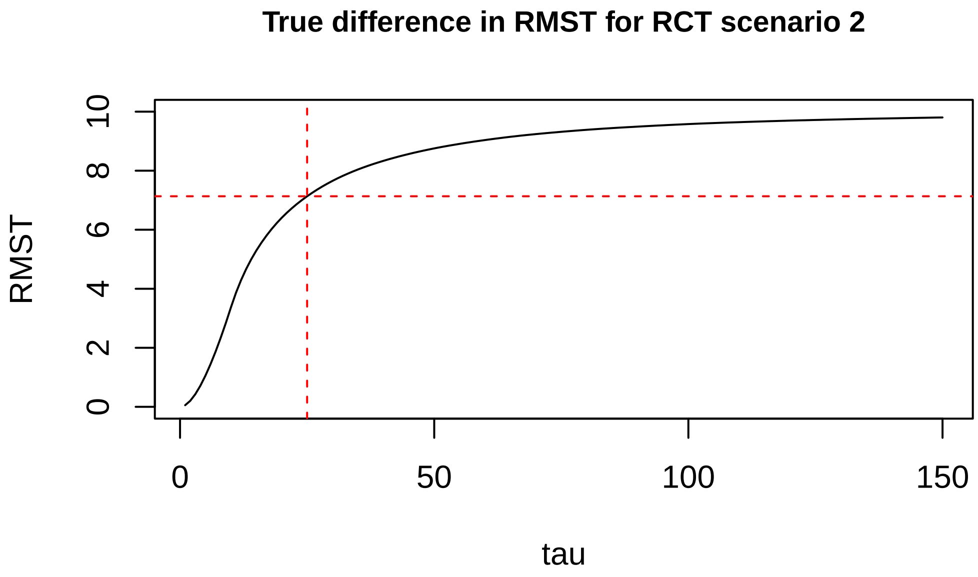
```
  # Compute RMST with the true T1
  data$T1_tau <- ifelse(data$T1 >= tau, tau, data$T1)

  # Compute RMST with the true T0
  data$T0_tau <- ifelse(data$T0 >= tau, tau, data$T0)

  # Compute the difference in RMST if everyone had the treatment
  # and if everyone had the control
  truth <- mean(data$T1_tau) - mean(data$T0_tau)

  return(truth)
}
```

The time-dependent ground truth for all the setting are displayed bellow:

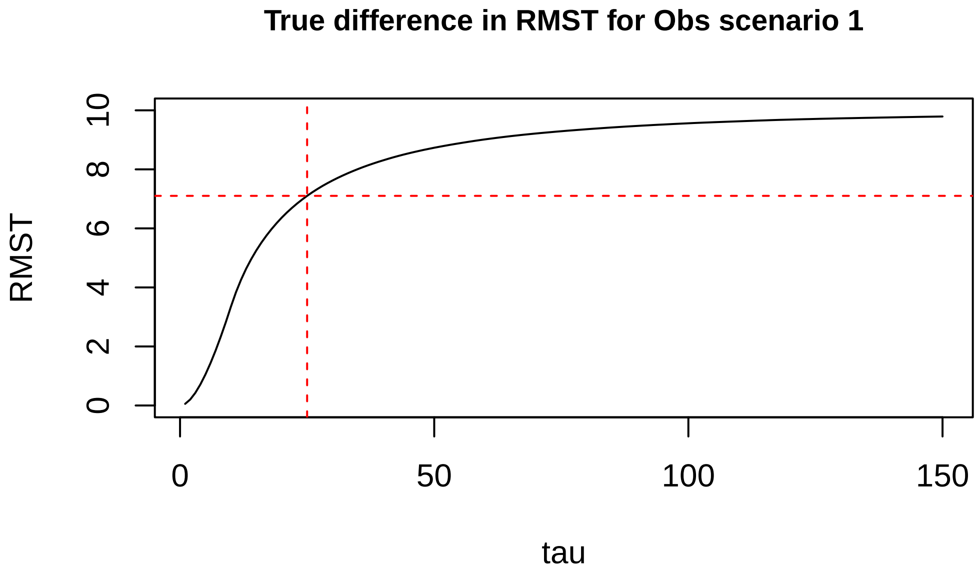


```
[1] 7.126142
```

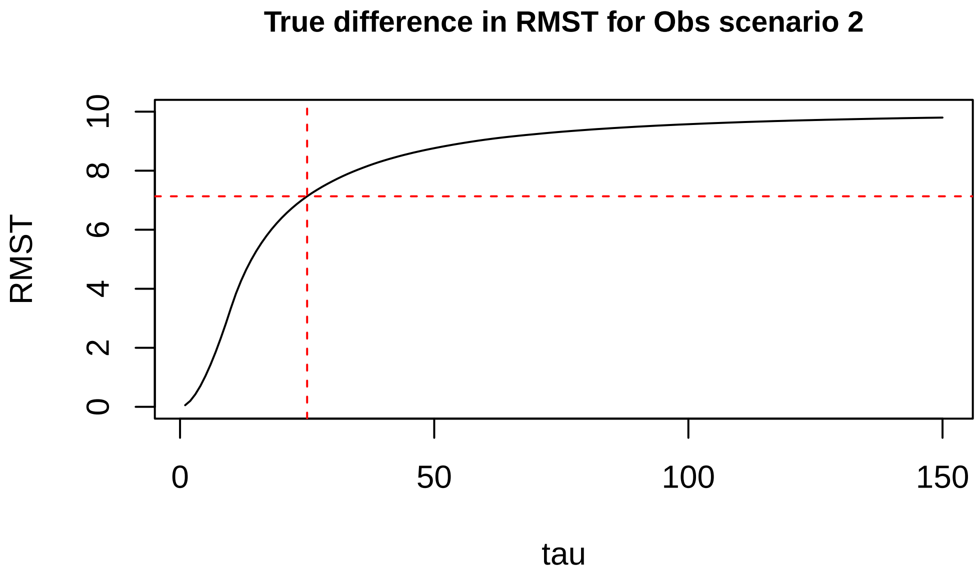
```
[1] "The ground truth for RCT scenario 1 at time 25 is 7.1"
```



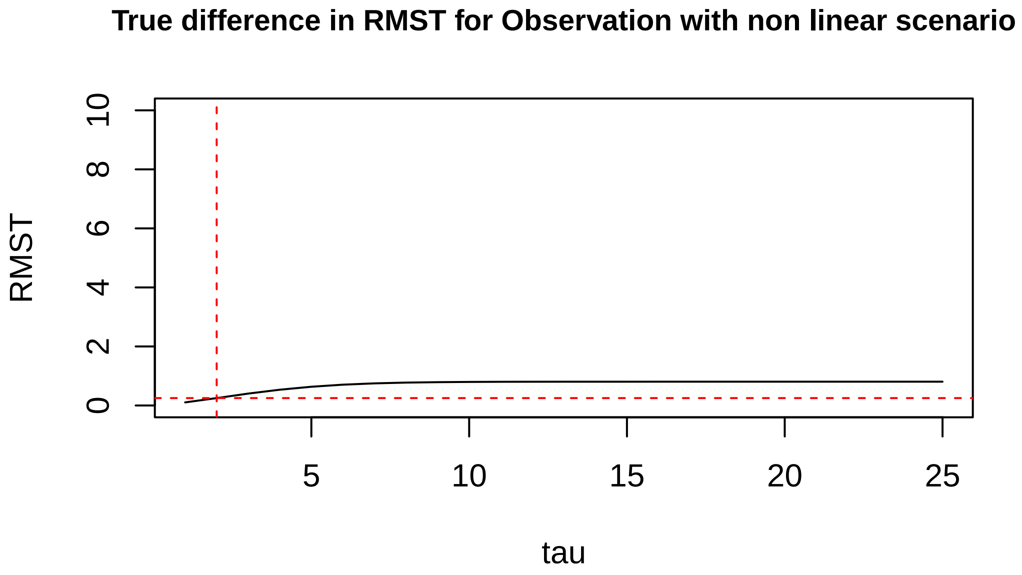
```
[1] 7.133358
[1] "The ground truth for RCT scenario 2 at time 25 is 7.1"
```



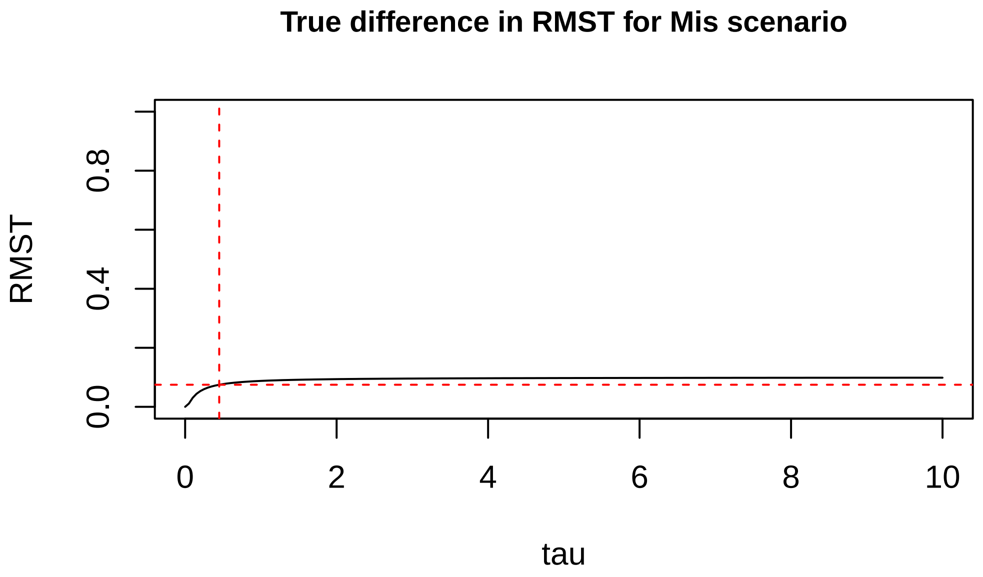
```
[1] 7.10183
[1] "The ground truth for Obs scenario 1 at time 25 is 7.1"
```



```
[1] 7.132828
[1] "The ground truth for Obs scenario 2 at time 25 is 7.1"
```



```
[1] 0.2506333
[1] "The ground truth for Observation with non linear scenario at time 2 is 0.3"
```



```
[1] 0.07484053
[1] "The ground truth for mis scenario at time 0.45 is 0"
```

The following section will apply the previous estimator on the presented simulation to evaluate their performances.

7.3 Estimation of the RMST

This section provides an evaluation of the previous estimators when the nuisance parameter specification is correct, as well as when there are mis-specifications of nuisance models or violations of the censoring positivity assumption.

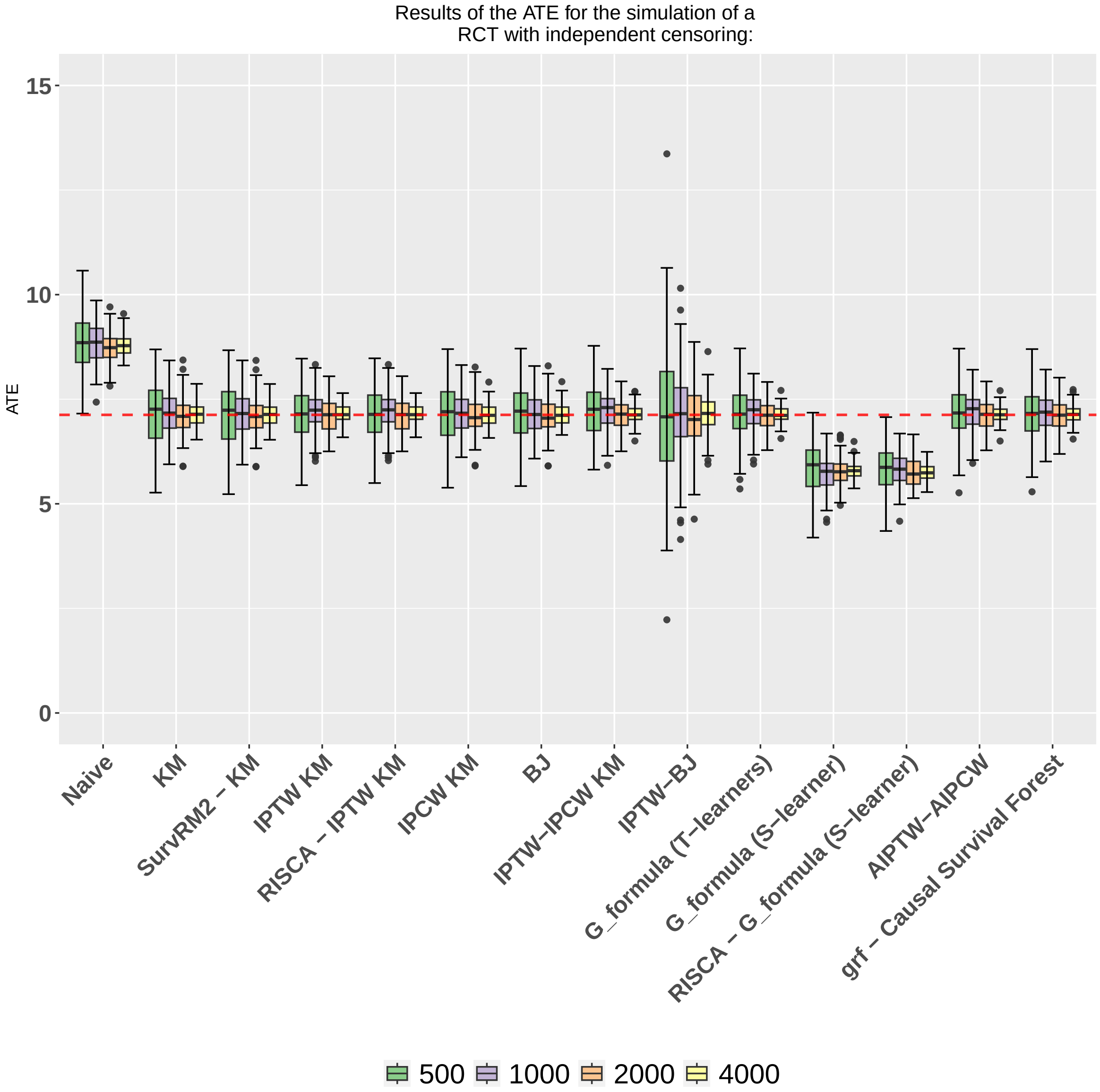
7.3.1 Correct specification of the nuisance parameters

In the case of RCT and observational study presented in Section 7.1.1 and Section 7.1.2, the nuisance models are estimated by Cox model for the conditional censoring model and for the conditional survival and by logistic regression for the propensity model. For the non parametric simulation presented in Section 7.1.3, the nuisance models are estimated by survival forest for conditional survival and conditional censoring, by a probability forest for the propensity model. Default tuning parameters were used for different forest-based methods. Additionally, cross-fitting with five folds is applied to these flexible models.

All the estimators detailed in Section 3 and Section 4 are computed 100 times at each sample size: 500, 1000, 2000, 4000 observations.

The estimations θ_{RMST} of all the presenting DGP are computed below:

The results are presented in boxplot for each sample size. The true value of θ_{RMST} is presented as red dotted line for $\tau = 25$. RCT scenario 1 results are displayed bellow:



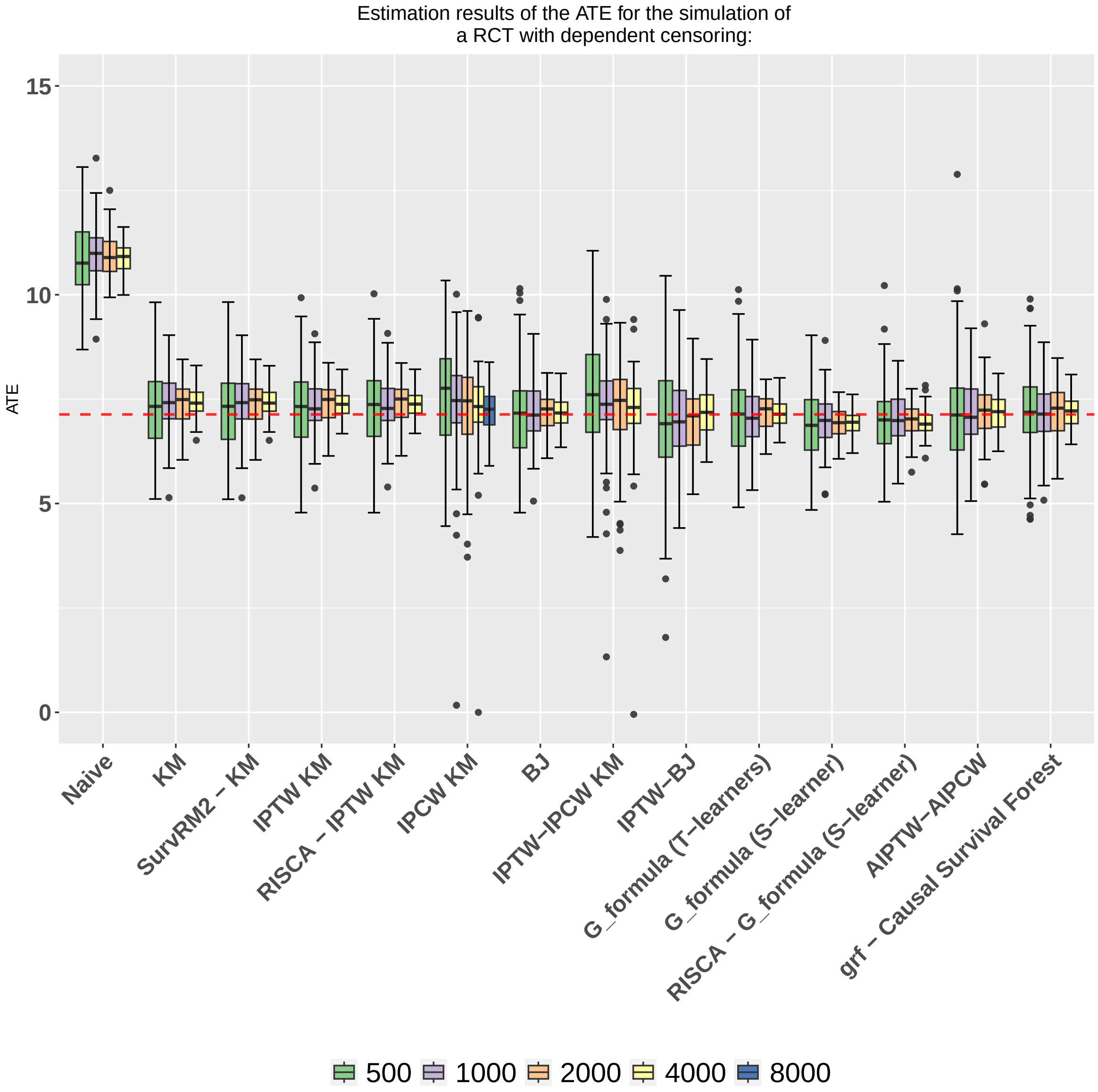
The boxplot above shows the distribution of the θ_{RMST} estimates for the RCT scenario 1 (with independent censoring).

In the context of correct specification of the nuisance parameter in using parametric and semi-parametric models, for the simulation of RCT and independent censoring , all the estimators converge except the Naive estimator and the G-formula estimator from RISCA package. Naive estimator is always biased because it does not take into account the censored observation at all in removing them. It applies then directly the θ_{RMST} formula on the uncensored observation. The bias of G-formula from RISCA package and our G-formula S-learner is due to the violation of proportional hazard assumption as the treatment effect is additive ($T(1) = T(0) + 10$).

When examining the variance of the estimators, the G-formula estimator consistently shows the lowest variance. The other estimators also maintain similarly low variances, even with small sample sizes, except for the IPTW-BJ estimator, which exhibits a significantly higher variance in small sample sizes. This increased variance can be attributed to the sensitivity of inverse probability weighting when applied directly to small sample size data without augmented correction. Among the convergent estimators AIPCW-AIPTW estimator, causal_survival_forest() from grf and G-formula (in using Two Learners) are the most efficient with a small sample size. They are very close to the true θ_{RMST} value even with 500 observations.

Surprisingly, Causal Survival Forest in this scenario converge as fast as AIPTW-AIPCW estimator. Generally, estimating linear models in using flexible regressions take more time to converge than a simple parametric estimation but this convergence can be explained due to the simple dataset design thanks to RCTs. Even if all these estimators are convergent, in the context of RCT and independent censoring, estimators with weights (such as IPTW KM, IPCW KM, IPTW-IPCW KM, IPTW-BJ or AIPTW-AIPCW) and also Causal Survival Forest don't really make sense to use. Indeed, they don't improve convergence, implement unnecessary weights and increase the computational time. Thus, in considering the speed of convergence as well as the computational time and difficulty of implementation in this simple RCT scenario, G-formula (T-learners) seems to be the most suitable with the smaller variance even at small sample size.

In the exact same way, the estimations in the context of RCT with conditionally independent censoring is displayed below. The red dashed line represents the true θ_{RMST} for $\tau = 25$.



In considering the simulation of RCT and conditionally independent censoring, exactly as before, the naive estimator is always biased. As expected, the unadjusted Kaplan Meier (KM) and its equivalent from SurvRM2 package, the adjusted estimator for treatment Kaplan Meier (IPTW KM) and its equivalent from RISCA package are always biased also. These estimators do not correct for the dependent censoring.

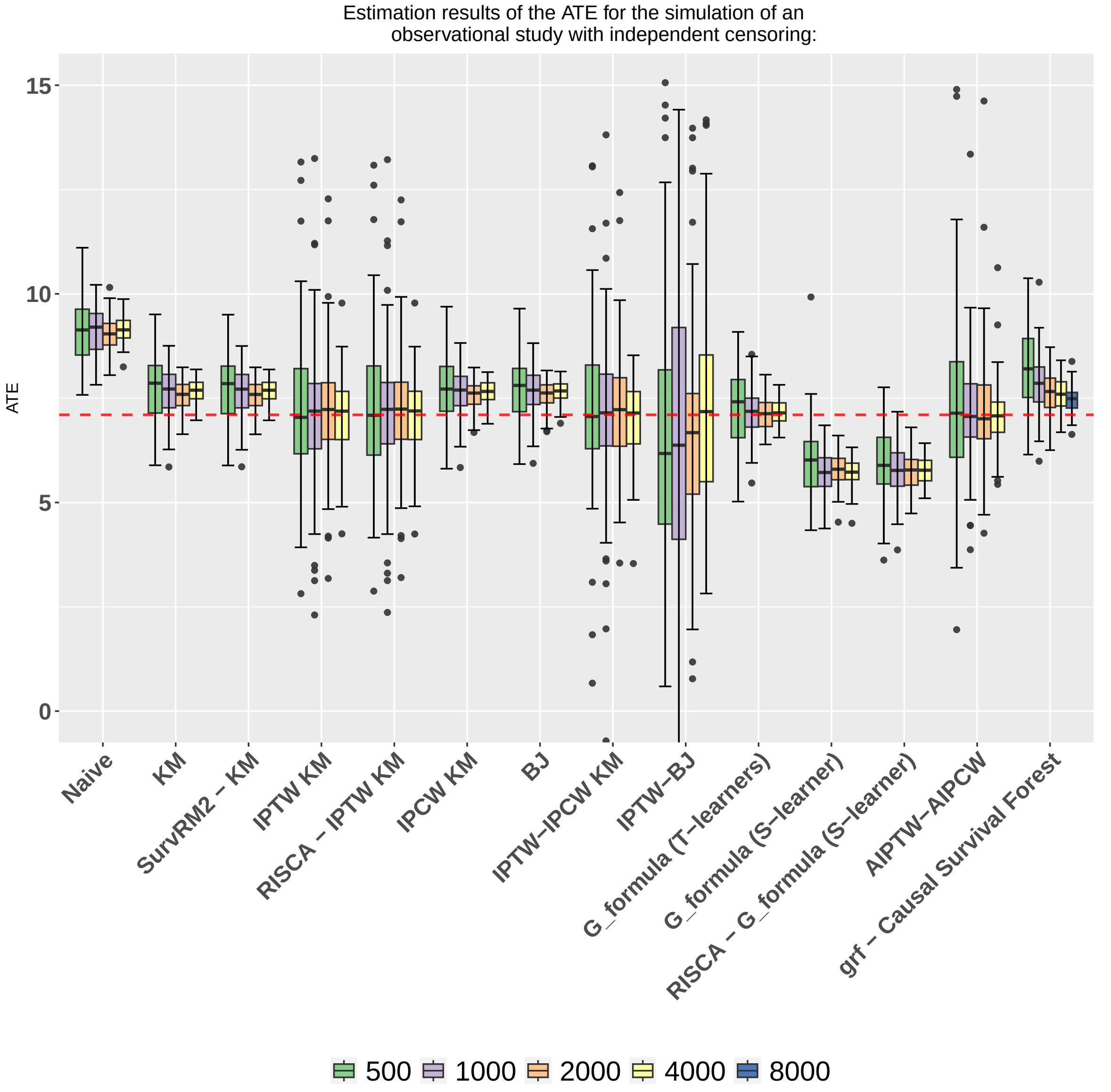
Surprisingly, the IPCW Kaplan-Meier and IPTW-IPCW Kaplan-Meier estimators appear slightly biased until 4,000 observations. This bias can be attributed to the simulation settings involving conditionally independent censoring, where the censoring rate was exaggerated to nearly 70% in the treatment group and 50% in the control group. Although the positivity of censoring is maintained, the probability of censoring in some sub-groups can be very close to 0 or 1. Additionally, the IPCW correction uniquely weights uncensored observations, which, in our case, accounts for only about 30% of the data. This, combined with the instability of weighting (or double-weighting for IPTW-IPCW), likely explains the difficulty of the IPTW-IPCW Kaplan-Meier and IPCW Kaplan-Meier estimators in converging with finite sample size. Additionally, it can explained also the high variability of this estimator. The estimators using Buckley-James transformation for censoring only (BJ estimator) and using Buckley-James with treatment correction (IPTW-BJ) are converging faster than those using IPC transformation with a better variance for BJ estimator. Both of them are unbiased even at 500 observations.

In the exact same way than RCT with independent censoring, G-formula (S-learner) and its equivalent from RISCA are biased. G-formula (T-learners) plug-in estimator, Causal Survival Forest and AIPCW-AIPTW demonstrate high efficiency even with small sample sizes. The estimator with the lowest variability is the G-formula (T-learners) and the Causal Survival Forest. Overall, the variability of all the estimators is greater than in the context of independent censoring.

However, exactly than RCT with independent censoring it seems excessive to use AIPCW-AIPTW which includes 3 nuisances parameters or Causal Survival Forest which uses flexible regressions to compute θ_{RMST} of a RCT with conditionally independent censoring. Then, in this context of good specification of nuisance model in RCT, G-formula (T-learners) is the estimator to use.

The boxplot below shows the distribution of the θ_{RMST} estimates for the Observational study with independent censoring. The red dashed line represents the true θ_{RMST} for $\tau = 25$.

```
Warning: Removed 51 rows containing non-finite values (`stat_boxplot()`).
Removed 51 rows containing non-finite values (`stat_boxplot()`).
```

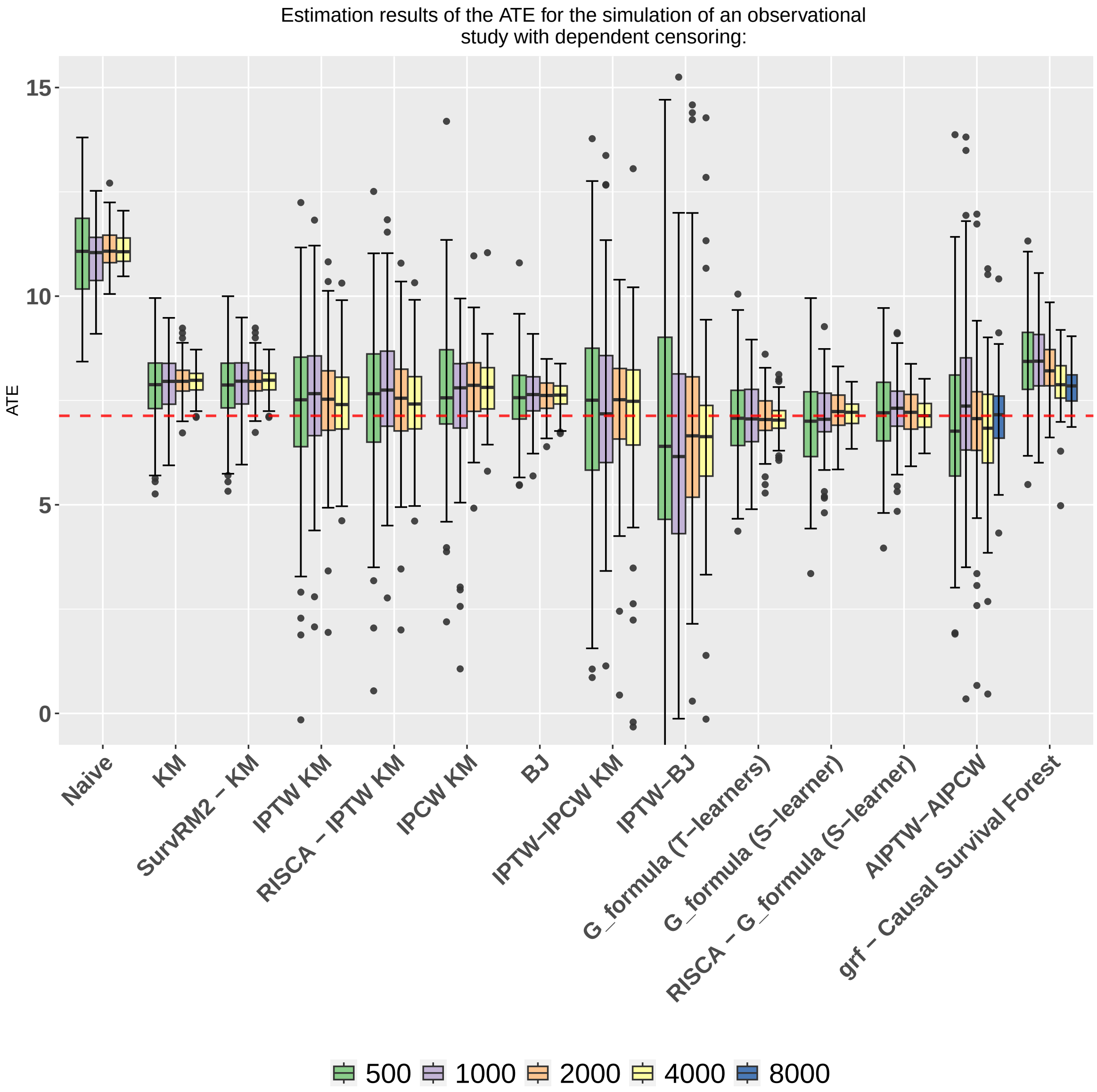


In the simulation of an observational study with independent censoring, the key difference compared to RCT simulation is the introduction of confounding bias. As a result, estimators that do not account for this bias, such as unadjusted Kaplan-Meier and IPCW Kaplan-Meier, along with their equivalents, exhibit the expected bias. In contrast, estimators like IPTW Kaplan-Meier, IPTW-IPCW Kaplan-Meier, IPTW-BJ, G-formula (T-learners), and AIPCW-AIPTW successfully converge. However, the IPTW-BJ estimator displays very high variability, likely due to the direct inverse probability weighting of the propensity score as we focus on observational study. The nuisance model of propensity score can struggle to converge at small sample size and predict probability near 0 or 1. The IPTW-IPCW Kaplan-Meier exibits also a lot of outliers. As before, the double-weighting is known to cause great instability due to the potential product of extrem weighting. The global variability is still less than IPTW-BJ wich has only one weighting correction.

The Causal Survival Forest from the grf package appears biased, likely due to the introduction of confounding bias, which adds complexity in the data, as well as the linear nature of the simulation. Thanks to the results with 8,000 observations of the Causal Survival Forest, we can see that it tends to converge asymptotically. In this context, the top-performing estimators are AIPCW-AIPTW, which converges the fastest (from 500 observations), and the G-formula, which consistently maintains the lowest variance over time and converges from 1,000 observations. The choice between these two estimators can be decided by the confidence in convergence of the nuisance models. While there is great confidence in the specification of outcome model, G-formula can be applied. Otherwise, AIPTW-AIPCW can be used.

The boxplot below shows the distribution of the θ_{RMST} estimates for the Observational study with conditionally independent censoring. The red dashed line represents the true θ_{RMST} for $\tau = 25$.

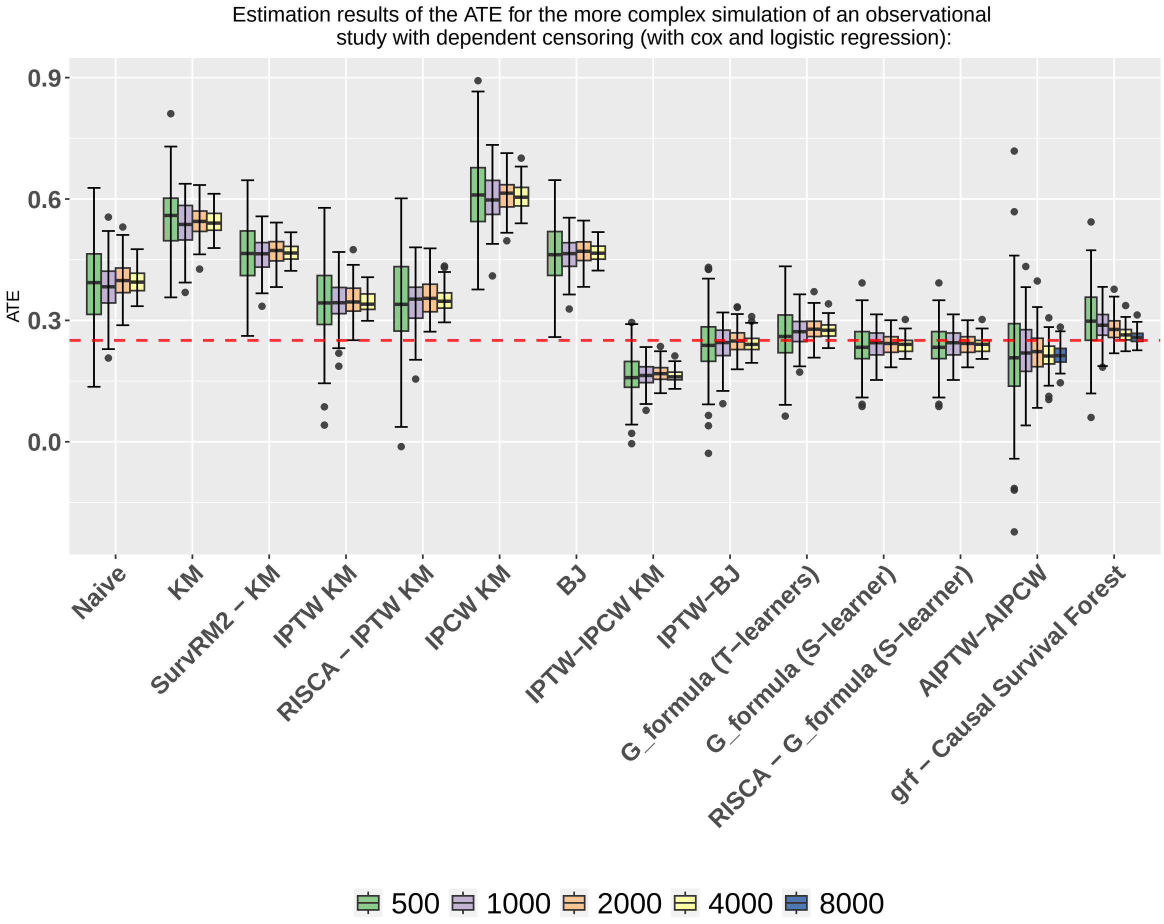
```
Warning: Removed 65 rows containing non-finite values (`stat_boxplot()`).
Removed 65 rows containing non-finite values (`stat_boxplot()`).
```



In the simulation of observational study with conditionally independent censoring, all estimators which do not consider correction for conditionally independent censoring and for confounding bias such as KM, IPCW KM, IPTW KM and their equivalent are biased. Contrary to the exposed properties, IPTW-IPCW Kaplan-Meier and IPTW-BJ estimators are biased even for 4,000 observations. As exposed previously, in the conditionally independent censoring simulation, in addition to conditionally independent censoring, the censoring rate was exaggerated to nearly 80% in the treatment group and 50% in the control group and IPCW considers only uncensored observation. Added to this, the complexity of the data set due to confounding bias and conditionally independent censoring, the nuisance parameters for IPTW-IPCW Kaplan-Meier struggle to converge even if nuisance models are well specified. The bias of IPTW-BJ estimator can be explained, in the same way than before, by the instability of the direct inverse probability weighting on the estimated complete data. This estimator is known to be unstable like IPTW-IPCW Kaplan-Meier.

AIPTW-AIPCW estimator is fluctuating around the true value with slight bias but appears to converge at 8,000 observations. This can be explained by the non negligeable number of nuisance models to estimate in a complex setting. Causal Survival Forest seems to slowly converge to the true θ_{RMST} but is still clearly biased even at 8,000 observations. This estimator exhibit its non-parameteric convergence rate in this complex context.
G-formula single learner and its equivalent from RISCA seems to have a very slight bias compared to the previous simulation. The proportional hazard is still violated but the bias is less important. It would seem that conditionally independent censoring makes it easier for the Cox model, which adjusts for the covariates and the treatment, to converge even if the treatment does not satisfy the proportional hazard hypothesis. This also clearly illustrates that the bias induced by the violation on the nuisance outcome parameter can vary from one to two.
Once again, the top-performing estimator in this context is G-formula (T-learners), which converge with lower variances compared to others.

The boxplot below shows the distribution of the θ_{RMST} estimates for the observational study with conditionally independent censoring in the context of non-parametric simulation (Section 7.1.3) when the nuisance models are estimated by Cox model and logistic regression. The red dashed line represents the true θ_{RMST} for $\tau = 2$.

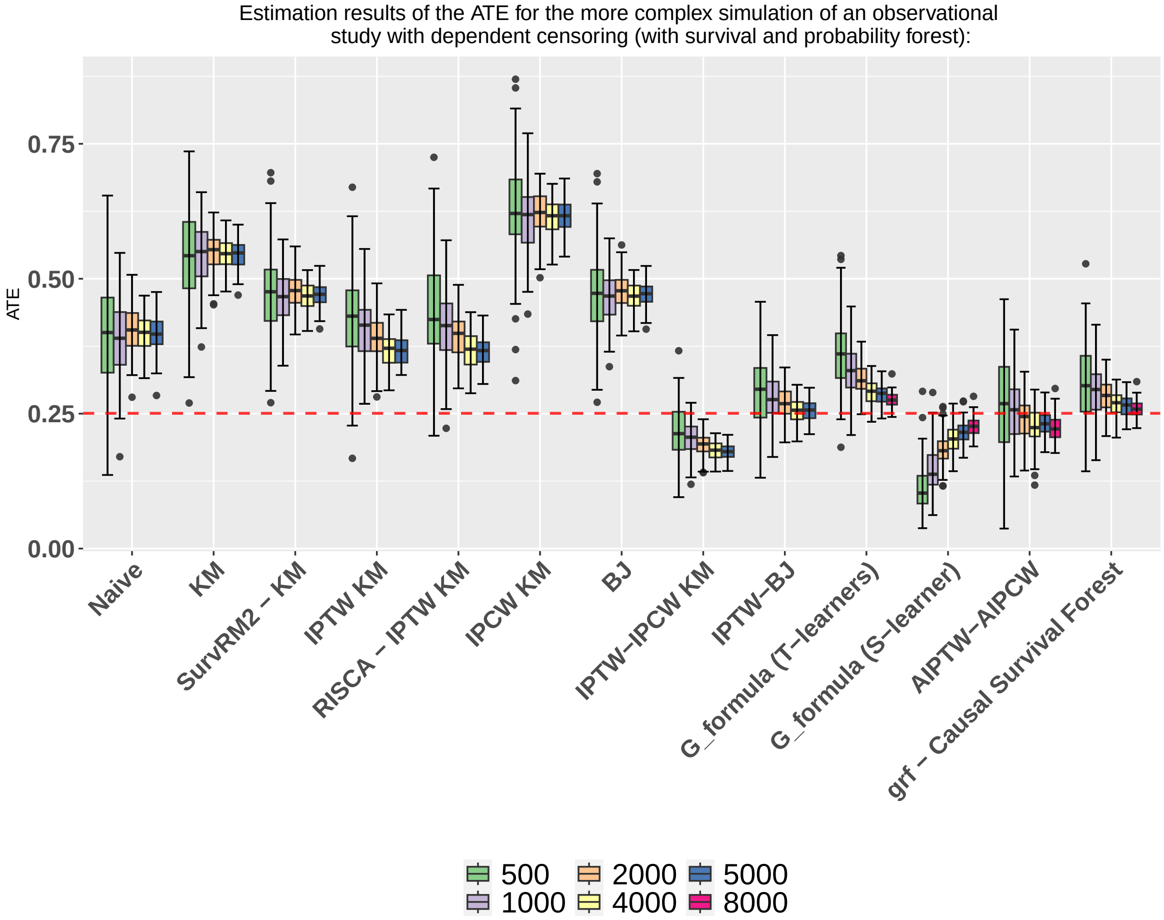


In this non-parametric simulation, nuisance parameters are estimated using the Cox model and logistic regression, except for the grf function, which uses non-parametric causal survival forests. While the propensity score is well-captured by logistic regression, the conditional probability of survival is not adequately modeled by the Cox model due to the violation of the proportional hazards assumption. Similarly, the violation of proportional hazards affects the modeling of the conditional probability of being uncensored, though less severely.

The estimators expected to converge under observational and dependent censoring include IPTW-IPCW Kaplan-Meier, IPTW-BJ, G-formula, AIPTW-AIPCW, and Causal Survival Forest. IPTW-IPCW shows the largest bias among these, likely due to the Cox model's poor fit for conditional censoring, its instability, and sensitivity to misspecification. Surprisingly, both G-formula S-learner estimates θ_{RMST} well despite the violation of proportional hazards. As seen in prior simulations, the bias from this violation is unpredictable, but by chance, the S-learner aligns with the true value, unlike the G-formula T-learner, which has a larger bias.

IPTW-BJ converges quickly to the true θ_{RMST}. As mentioned, IPTW is well-estimated by logistic regression, and the small bias from the Cox model's conditional survival estimate is mitigated by the Buckley-James transformation. This transformation introduces conditional survival for censored observations while retaining uncensored data for complete observations. The small proportion of censored data (35%) may explain this estimator's convergence.

Given these results, it is unsurprising that AIPTW-AIPCW fails to converge, as only the propensity score is well-estimated. This is insufficient for the AIPTW-AIPCW estimator to converge. In this simulation, the only converging estimators are IPTW-BJ, due to specific conditions, and the Causal Survival Forest, which uses flexible regression for nuisance model estimation.

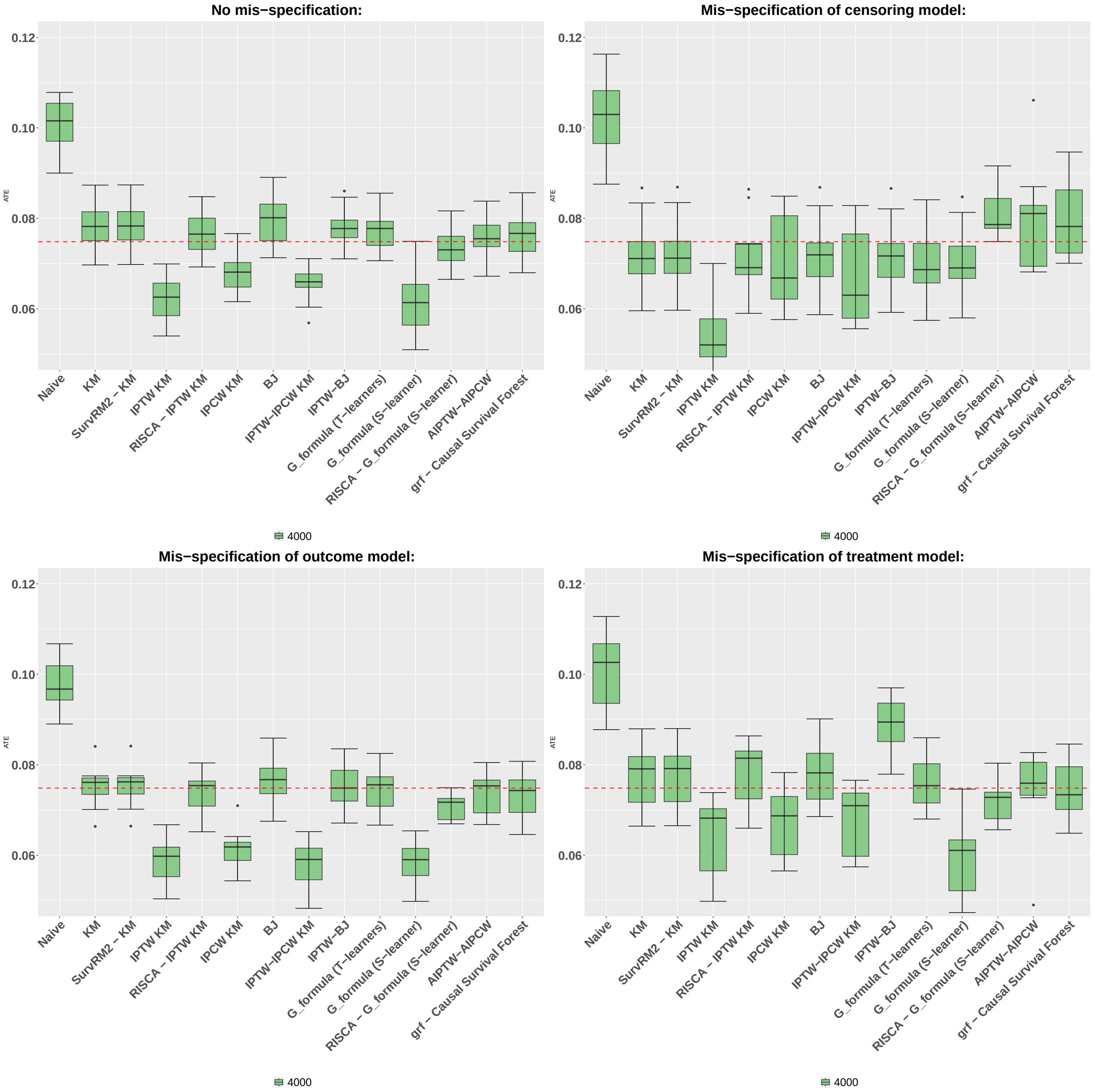


In the complex setting, all nuisance models are computed using flexible models. This simulation refers to an observational study with conditionally independent censoring also. As expected, all the estimators without correction for conditionally independent censoring and confounding bias are biased. It concerns Naive, KM, IPTW KM, IPCW KM and BJ estimator and their equivalents. The estimator IPTW-BJ has a small bias but tends to converge after 5,000 observations. G-formula (T-learners), G-formula (S-learner) and Causal Survival Forest tend to converge after 5,000 observations. Causal Survival Forest has less bias than the others. In Foster, Taylor, and Ruberg (2011), they introduce the estimator Virtual Twins (VT) which corresponds to our G-formula with survival random forest to estimate conditional survival probability. It has proved that this estimator lacks orthogonality properties (Chernozhukov et al. 2016). Additionally, if ML algorithms are not from Donsker class or ML algorithms whose entropy increases with the size of the sample used, this estimator becomes predisposed to bias (Vaart 1998). As a result, the asymptotic normality of VT is compromised. Also, the article Cui et al. (2023) shows that VT has lower efficiency than causal survival forest which can be seen in our simulation. Then, G-formula seems not to be the most efficient estimator to use in non-parametric setting. Also, S-learner has better behavior than T-learners due to the fact that the learner is not stratified by treatment then it has more observations to learn the outcome model. In general, the finite sample bias of all the estimators can be explained by the convergence rate of non parametric model. AIPTW-AIPCW has the same behavior than in observational and conditionally independent censoring with parametric estimation of nuisance models, its finite sample properties are very good but it still has sort of fluctuation around the true value of ATE. Its variability is low and has few outliers. Even if it's fluctuating, the bias is low compared to the other estimators. The top-performing estimators in this context are IPTW-BJ asymptotically but AIPTW-AIPCW estimator in finite sample size, which converge sooner but tends to fluctuate.

7.3.2 Mis-specification of the nuisance parameters

In this section, we will challenge the estimators in introducing mis-specification for the nuisance parameters. To do so, the simulation with interaction will be used and mis-specification is introduced by selecting only one covariate with the less impact on the model. All functions that cannot introduce mis-specification specifically on treatment, censoring or outcome are removed of the following graph. It concerns G-formula from RISCA package and the causal survival forest from grf. First, we will introduce mis-specification for the treatment model only, for the censoring model only, for the outcome model only, then for treatment and censoring, for treatment and outcome , for outcome and censoring and finally for all the nuisance models.

The nuisance models are estimated with flexible model such as probability forest and survival forest.



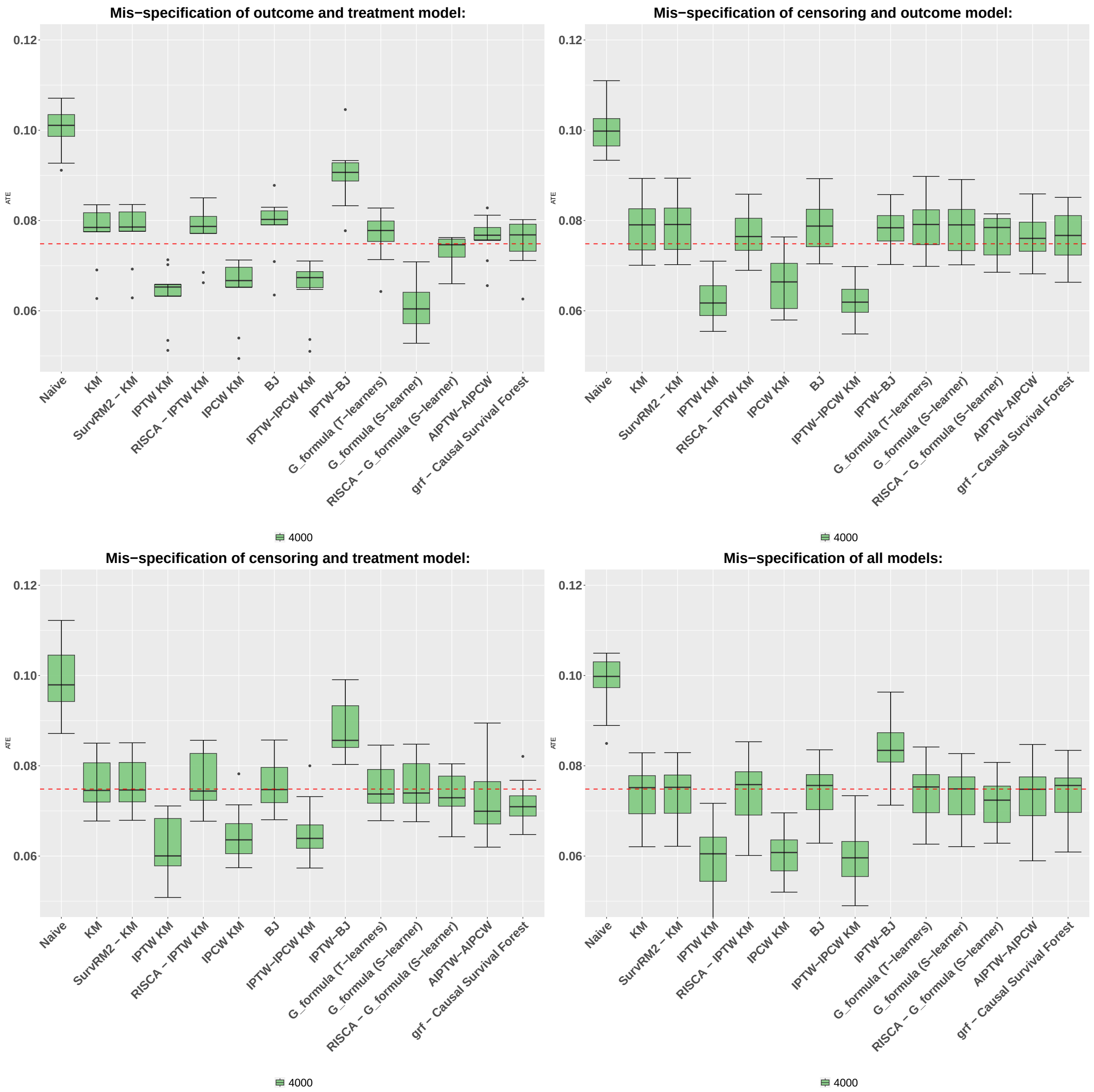
REVOIR : We refer to an observational and dependent censoring study, thus the expected consistent estimators are IPTW-IPCW Kaplan-Meier, IPTW-BJ estimator, G-formula, AIPTW-AIPCW and Causal survival forest.

In globality, when one nuisance model is mis-specified, the consistent estimators are AIPTW-AIPCW and causal survival forest. In all these setup, the two estimators are convergent at 4,000 observations.

When the censoring model is mis-specified, the estimators with censoring correction are biased. **The variation of G-formula estimator and other compared to the simulation without mis-specification can be due to the low repetitions**.

G-formula and IPTW-BJ estimator seems to converge when the mis-specification of the outcome model is present but in comparing it to the simulation without mis-specification, the results is different. It shows that it goes near the true value this time but only by chance. As expected, when the treatment model is mis-specified, IPTW-BJ estimator is biased.

The combined graph regroup the θ_{RMST} value of 4 different setting when there is mis-specification on all nuisance parameters (bottom right) or when two nuisance models are mis-specified either the censoring and the treatment models (top left), the outcome and treatment models (top right) or the outcome and censoring models (bottom left). The dot line represents the true value of the ATE.



On hold :

When the censoring and treatment models are mis-specified, G-formula (T-learners) estimator remains unbiased. The estimator IPTW-IPCW which is based on the two nuisance models is completely biased. Same result for IPTW-BJ estimator. and surprisingly AIPTW-AIPCW estimator become slightly biased.

When the outcome and treatment models are mis-specified, all the estimators are biased. **AIPTW-AIPCW is considered as non convergent because of its huge variability.** It is the same results when the mis-specification concerns outcome and censoring models. *

When all the nuisance models are mis-specified, all the converging estimators in the context of no mis-specification are biased.

Résultats pas trop normaux pour les biais de AIPTW-AIPCW surtout dans le contexte “censoring and treatment models mis-specified”, il devrait converger. Je refais avec 100 simulations –> Peut-être dû au fait que l’outcome model converge lentement (à étudier

après avoir fait tourner avec plus de simulations)

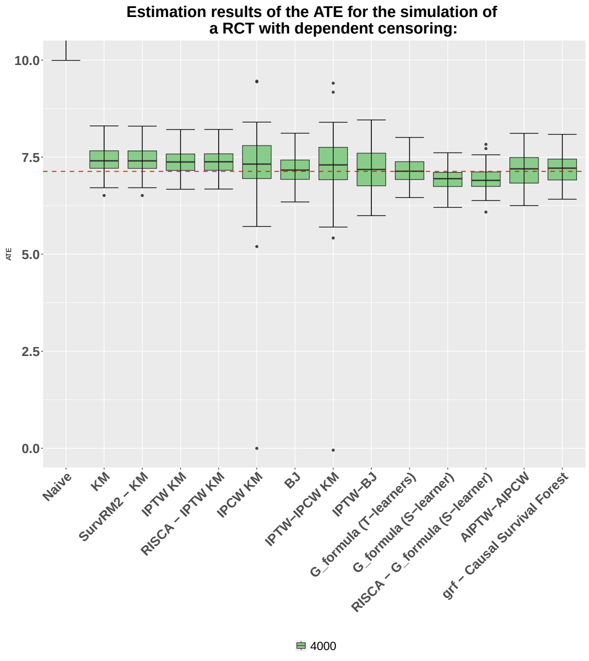
Résultats pas normaux pour la convergence de AIPTW-AIPCW “outcome and censoring models mis-specified”, il ne devrait pas converger (mais il a une forte variabilité, je refais avec 100 simulations –> Va prendre du temps)

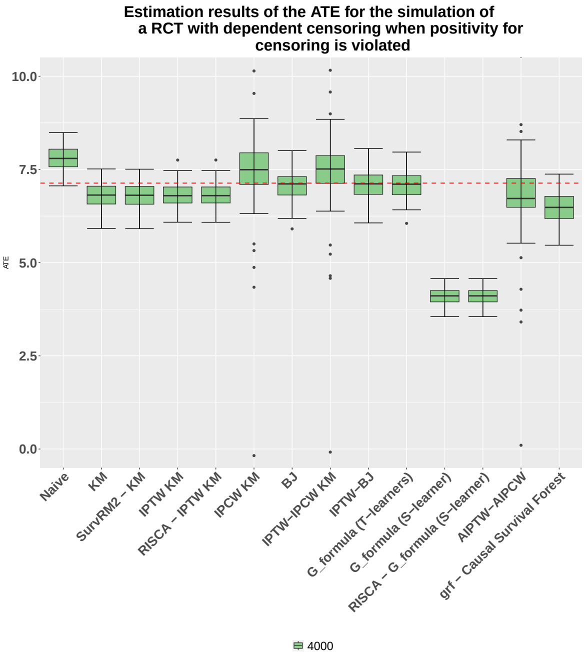
7.3.3 Violation of positivity assumption for censoring

In this section, the objectives is to show the impact of the violation of positivity assumption for censoring on the convergence of estimators. To enable this violation, we use the exact same simulation for RCT and conditionally independent censoring and Observational and conditionally independent censoring but with stronger relationship with covariates:

- For our RCT: $\lambda_c(X) = 0.002 \cdot \exp\{2X_1 - 4X_2 - 5X_3 + 0.2X_4 - 0.2A\}$.
- For our observational study: $\lambda_c(X) = 0.002 \cdot \exp\{2X_1 - 4X_2 - 5X_3 + 2X_4\}$.

The violation of positivity censoring can be verified in verifying the probability of remain uncensored for all combination of covariates (with cox model if proportional hazard is verified or with survival forest if not).

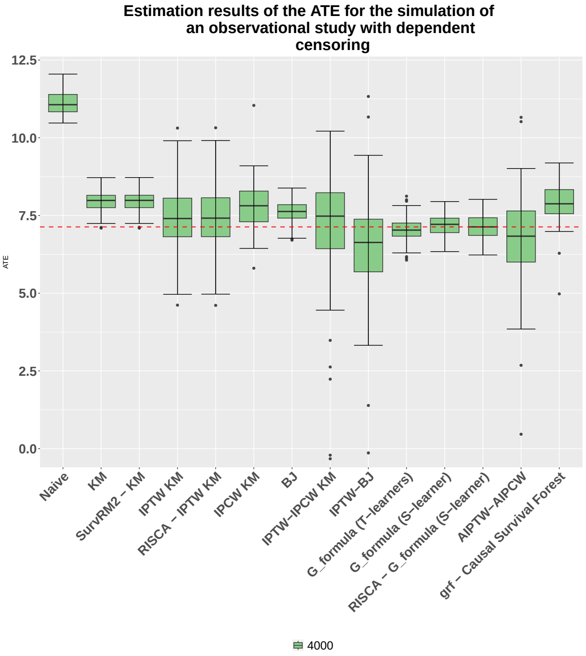


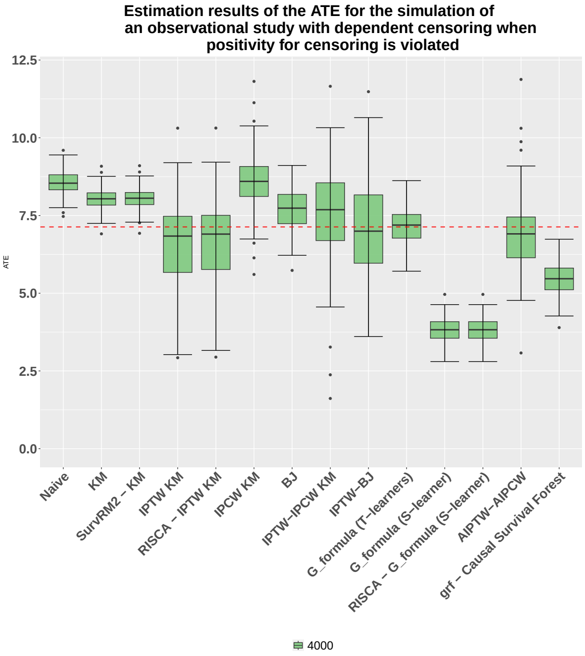


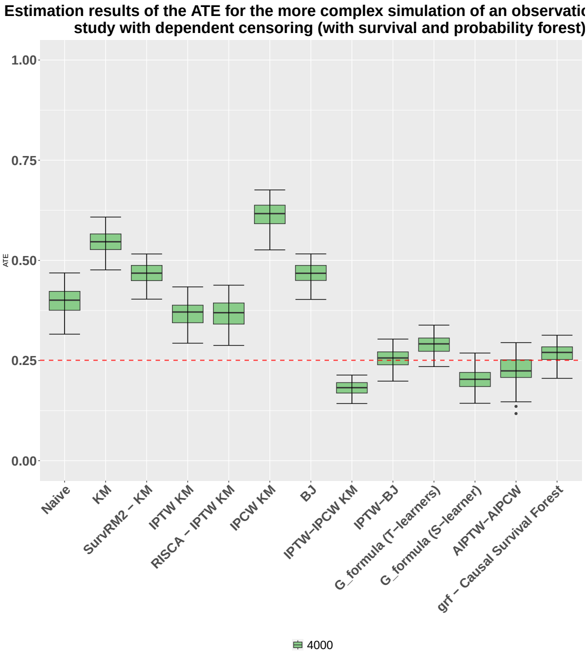
When the positivity of censoring is violated in our RCT context with conditionally independent censoring, all estimators that adjust for censoring using inverse probability weighting (IPW) methods, such as IPCW Kaplan-Meier, IPTW-IPCW Kaplan-Meier, AIPTW-AIPCW, or Causal Survival Forest, become biased under this violation. The G-formula (S-learner) and its equivalent exhibit even stronger bias. Conversely, the G-formula (T-learners), the IPTW-BJ estimator, and the BJ estimator continue to converge accurately.

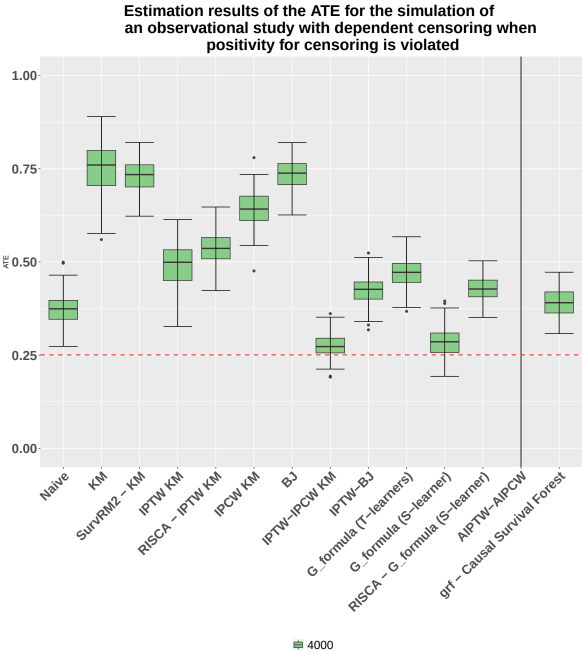
```
Warning: Removed 11 rows containing non-finite values (`stat_boxplot()`).
Removed 11 rows containing non-finite values (`stat_boxplot()`).
```

```
Warning: Removed 14 rows containing non-finite values (`stat_boxplot()`).
Removed 14 rows containing non-finite values (`stat_boxplot()`).
```









Peut être refaire tourner avec plus de dependence car pas évident ici When the positivity of censoring is violated in our observational context with conditionally independent censoring, all estimators that adjust for censoring using inverse probability weighting (IPW) methods, such as IPTW-IPCW Kaplan-Meier, AIPTW-AIPCW, or Causal Survival Forest, become a bit more biased under this violation. The G-formula (S-learner) and its equivalent exhibit even stronger bias. Conversely, the G-formula (T-learners) and the IPTW-BJ estimator have the same behavior.

8 Conclusion and perspective

In this study on causal survival analysis, we examined a broad range of estimators across multiple simulation settings, particularly focusing on treatment and censoring mechanisms. While the majority of the empirical results align well with the theoretical properties, notable exceptions include estimators based on IPCW transformations, such as the IPTW-IPCW Kaplan-Meier, which failed to converge in non-parametric settings, and the IPCW Kaplan-Meier, which showed some bias under high censoring conditions.

In this study, we made available our code for the estimators tested, addressing a significant gap in current resources. Despite the growing need for causal survival analysis tools, few packages exist that offer robust implementations of various estimators. By providing these codes, we aim to support further research and practical applications of causal inference in survival contexts, where available software is currently limited.

For users, we recommend different estimators based on the context of their data and modeling assumptions. In a parametric setting with correctly specified nuisance parameters, the G-formula is highly efficient due to its low variance and quick convergence, especially in simpler scenarios like randomized controlled trials. In non-parametric settings with no misspecification, estimators like the IPTW-BJ estimator, Causal Survival Forest or AIPTW-AIPCW perform well, though the latter two may require more computing resources. When uncertainty exists around nuisance parameters, robust estimators like AIPTW-AIPCW or Causal Survival Forest are preferable due to their resilience to mis-specification.

Notably, the G-formula and IPTW-BJ estimators implicitly rely on the positivity assumption, although it is not explicitly included in their identifiability conditions. In parametric settings, violations of positivity tend to have minimal impact, as the extrapolation needed to handle these violations is feasible but risky. However, in non-parametric contexts, violating the positivity assumption can lead to erroneous extrapolations, making these estimators unsuitable for use.

A key limitation of our simulations is the use of larger datasets with relatively simple relationships, which may not reflect the complexity of real-world scenarios. Most survival analysis datasets are smaller and more intricate, so the stability of certain estimators observed here may not fully generalize. It would be valuable to test these methods on real-world datasets to better assess their performance in practical applications.

Looking ahead, one promising avenue for improving these estimators is through optimizing variable selection for the conditional censoring, conditional survival, and treatment models. It has been shown that adding precision variables in causal inference enhances the variance of G-formula-like estimators. A potential area of exploration would be to investigate whether similar improvements can be made in causal survival analysis by refining the selection of covariates that influence censoring and survival outcomes.

References

Andersen, Per K., Elisavet Syriopoulou, and Erik T. Parner. 2017. "Causal Inference in Survival Analysis Using Pseudo-Observations." *Statistics in Medicine* 36 (17): 2669–81. https://doi.org/https://doi.org/10.1002/sim.7297.

Athey, Susan, Julie Tibshirani, and Stefan Wager. 2018. "Generalized Random Forests." https://arxiv.org/abs/1610.01271.

Athey, Susan, and Stefan Wager. 2019. "Estimating Treatment Effects with Causal Forests: An Application." https://arxiv.org/abs/1902.07409.

Breslow, N., and J. Crowley. 1974. "A Large Sample Study of the Life Table and Product Limit Estimates Under Random Censorship." *The Annals of Statistics* 2 (3): 437–53. https://doi.org/10.1214/aos/1176342705.

Buckley, and James. 1979. "Linear regression with censored data." *Biometrika* 66 (3): 429–36. https://doi.org/10.1093/biomet/66.3.429.

Chatton, Arthur, Florent Le Borgne, Clémence Leyrat, and Yohann Foucher. 2022. "G-Computation and Doubly Robust Standardisation for Continuous-Time Data: A Comparison with Inverse Probability Weighting." *Statistical Methods in Medical Research* 31 (4): 706–18. https://doi.org/10.1177/09622802211047345.

Chen, Pei-Yun, and Anastasios A. Tsiatis. 2001. "Causal Inference on the Difference of the Restricted Mean Lifetime Between Two Groups." *Biometrics* 57 (4): 1030–38. https://doi.org/https://doi.org/10.1111/j.0006-341X.2001.01030.x.

Chernozhukov, Victor, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. 2016. "Double/Debiased Machine Learning for Treatment and Causal Parameters."

Cox, D. R. 1972. "Regression Models and Life-Tables." *Journal of the Royal Statistical Society: Series B (Methodological)* 34 (2): 187–202. https://doi.org/https://doi.org/10.1111/j.2517-6161.1972.tb00899.x.

Cui, Yifan, Michael R Kosorok, Erik Sverdrup, Stefan Wager, and Ruoqing Zhu. 2023. "Estimating heterogeneous treatment effects with right-censored data via causal survival forests." *Journal of the Royal Statistical Society Series B: Statistical Methodology* 85 (2): 179–211. https://doi.org/10.1093/jrsssb/qkac001.

Díaz, Iván, Elizabeth Colantuoni, Daniel F Hanley, and Michael Rosenblum. 2019. "Improved Precision in the Analysis of Randomized Trials with Survival Outcomes, Without Assuming Proportional Hazards." *Lifetime Data Analysis* 25 (3): 439—468. https://doi.org/10.1007/s10985-018-9428-5.

Ding, Peng. 2023. *A First Course in Causal Inference*. https://arxiv.org/abs/2305.18793.

Fan, Jianqing, and Irene Gijbels. 1994a. *Local Polynomial Modelling and Its Applications*. https://api.semanticscholar.org/CorpusID:118744555.

Fan, Jianqing, and Irène Gijbels. 1994b. "Censored Regression: Local Linear Approximations and Their Applications." *Journal of the American Statistical Association* 89 (426): 560–70. https://doi.org/10.1080/01621459.1994.10476781.

Fisher, Aaron, and Edward H. Kennedy. 2021. "Visually Communicating and Teaching Intuition for Influence Functions." *The American Statistician* 75 (2): 162–72. https://doi.org/10.1080/00031305.2020.1717620.

Foster, Jared C, Jeremy M G Taylor, and Stephen J Ruberg. 2011. "Subgroup Identification from Randomized Clinical Trial Data." *Stat. Med.* 30 (24): 2867–80.

Foucher, Yohann, Florent Le Borgne, and Arthur Chatton. 2019. "RISCA: Causal Inference and Prediction in Cohort-Based Analyses." *CRAN: Contributed Packages*. The R Foundation. https://doi.org/10.32614/cran.package.risca.

Hernán MA, Robins JM. 2020. *Causal Inference: What If*. Boca Raton: Chapman & Hall/CRC.

Hernán, Miguel A. 2010. "The Hazards of Hazard Ratios." *Epidemiology (Cambridge, Mass.)* 21 (1): 13—15. https://doi.org/10.1097/ede.0b013e3181c1ea43.

Hernán, Miguel A, and James M Robins. 2010. "Causal Inference." CRC Boca Raton, FL.

Hines, Oliver, Oliver Dukes, Karla Diaz-Ordaz, and Stijn Vansteelandt. 2022. "Demystifying Statistical Learning Based on Efficient Influence Functions." *The American Statistician* 76 (3): 292–304. https://doi.org/10.1080/00031305.2021.2021984.

Howe, Chanelle J, Stephen R. Cole, Bryan Lau, Sonia Napravnik, and Joseph J. Eron. 2016. "Selection Bias Due to Loss to Follow up in Cohort Studies." *Epidemiology* 27 1: 91–97. https://api.semanticscholar.org/CorpusID:21993915.

Imbens, Guido W., and Donald B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press.

James M. Robins, Andrea Rotnitzky, and Lue Ping Zhao. 1994. "Estimation of Regression Coefficients When Some Regressors Are Not Always Observed." *Journal of the American Statistical Association* 89 (427): 846–66. https://doi.org/10.1080/01621459.1994.10476818.

———. 1995. "Analysis of Semiparametric Regression Models for Repeated Outcomes in the Presence of Missing Data." *Journal of the American Statistical Association* 90 (429): 106–21. https://doi.org/10.1080/01621459.1995.10476493.

Kaplan, E. L., and Paul Meier. 1958. "Nonparametric Estimation from Incomplete Observations." *Journal of the American Statistical Association* 53 (282): 457–81. https://doi.org/10.1080/01621459.1958.10501452.

Kennedy, Edward H. 2023. "Semiparametric Doubly Robust Targeted Double Machine Learning: A Review." https://arxiv.org/abs/2203.06469.

Koul, H., V. Susarla, and J. Van Ryzin. 1981. "Regression Analysis with Randomly Right-Censored Data." *The Annals of Statistics* 9 (6): 1276–88. https://doi.org/10.1214/aos/1176345644.

Laan, Mark J. van der, and James M. Robins. 2003. *Unified Methods for Censored Longitudinal Data and Causality*. https://api.semanticscholar.org/CorpusID:62474250.

Martinussen, Torben, Stijn Vansteelandt, and Per Andersen. 2020. "Subtleties in the Interpretation of Hazard Contrasts." *Lifetime Data Analysis* 26 (October). https://doi.org/10.1007/s10985-020-09501-5.

Ozenne, Brice Maxime Hugues, Thomas Harder Scheike, Laila Stærk, and Thomas Alexander Gerds. 2020. "On the Estimation of Average Treatment Effects with Right-Censored Time to Event Outcome and Competing Risks." *Biometrical Journal* 62 (3): 751–63. https://doi.org/https://doi.org/10.1002/bimj.201800298.

Robins, James. 1986. "A New Approach to Causal Inference in Mortality Studies with a Sustained Exposure Period—Application to Control of the Healthy Worker Survivor Effect." *Mathematical Modelling* 7 (9): 1393–1512. https://doi.org/https://doi.org/10.1016/0270-0255(86)90088-6.

Robins, James M., and Dianne M. Finkelstein. 2000. "Correcting for Noncompliance and Dependent Censoring in an AIDS Clinical Trial with Inverse Probability of Censoring Weighted (IPCW) Log-Rank Tests." *Biometrics* 56 (3): 779–88. https://doi.org/https://doi.org/10.1111/j.0006-341X.2000.00779.x.

Robins, James M., and Andrea Rotnitzky. 1992. "Recovery of Information and Adjustment for Dependent Censoring Using Surrogate Markers." In *AIDS Epidemiology: Methodological Issues*, edited by Nicholas P. Jewell, Klaus Dietz, and Vernon T. Farewell, 297–331. Boston, MA: Birkhäuser Boston. https://doi.org/10.1007/978-1-4757-1229-2_14.

Robins, James, Andrea Rotnitzky, and Marco Bonetti. 2004. "Discussion of the Frangakis and Rubin Article." *Biometrics* 57 (2): 343–47. https://doi.org/10.1111/j.0006-341X.2001.00343.x.

Rosenbaum, Paul R., and Donald B. Rubin. 1983. "The central role of the propensity score in observational studies for causal effects." *Biometrika* 70 (1): 41–55. https://doi.org/10.1093/biomet/70.1.41.

Rubin, Daniel, and Mark J. van der Laan. 2007. *The International Journal of Biostatistics* 3 (1). https://doi.org/doi:10.2202/1557-4679.1052.

Rubin, Donald B. 1974. "Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies." *Journal of Educational Psychology* 66 (5): 688–701. https://doi.org/10.1037/h0037350.

Schaubel, Douglas E., and Guanghui Wei. 2011. "Double Inverse-Weighted Estimation of Cumulative Treatment Effects under Nonproportional Hazards and Dependent Censoring." *Biometrics* 67 (1): 29–38. https://doi.org/10.1111/j.1541-0420.2010.01449.x.

Survival and Event History Analysis: A Process Point of View. 2008. New York, NY: Springer New York. https://doi.org/10.1007/978-0-387-68560-1_1.

"survRM2: Comparing Restricted Mean Survival Time." 2015. *CRAN: Contributed Packages*. The R Foundation. https://doi.org/10.32614/cran.package.survrm2.

Therneau, Terry M. 2001. "Survival: Survival Analysis." *CRAN: Contributed Packages*. The R Foundation. https://doi.org/10.32614/cran.package.survival.

Tibshirani, Julie, Susan Athey, Erik Sverdrup, and Stefan Wager. 2017. "Grf: Generalized Random Forests." *CRAN: Contributed Packages*. The R Foundation. https://doi.org/10.32614/cran.package.grf.

Tsiatis, Anastasios A. 2006. *Semiparametric Theory and Missing Data*. https://api.semanticscholar.org/CorpusID:118005650.

Turkson, Anthony, Francis ayiah-mensah, and Vivian Nimoh. 2021. "Handling Censoring and Censored Data in Survival Analysis: A Standalone Systematic Literature Review." *International Journal of Mathematics and Mathematical Sciences* 2021 (September): 1–16. https://doi.org/10.1155/2021/9307475.

Vaart, A. W. van der. 1998. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Willems, SJW, A Schat, MS van Noorden, and M Fiocco. 2018. "Correcting for Dependent Censoring in Routine Outcome Monitoring Data by Applying the Inverse Probability Censoring Weighted Estimator." *Statistical Methods in Medical Research* 27 (2): 323–35. https://doi.org/10.1177/0962280216628900.

Xie, Jun, and Chaofeng Liu. 2005. "Adjusted Kaplan–Meier Estimator and Log-Rank Test with Inverse Probability of Treatment Weighting for Survival Data." *Statistics in Medicine* 24 (20): 3089–3110. https://doi.org/https://doi.org/10.1002/sim.2174.

Zheng, Wenjing, and Mark J van der Laan. 2012. "Targeted Maximum Likelihood Estimation of Natural Direct Effects." *The International Journal of Biostatistics* 8 (1).

9 Annex

9.1 Link between RMST and survival probabilities

Survival probabilities and RMST are linked as follows:

$$\begin{aligned}\theta_{RMST}(\tau) &= = E\left[\int_0^\tau I\{T(1) > t\}dt - \int_0^\tau I\{T(0) > t\}dt\right] \\ &= \int_0^\tau \mathbb{E}[I\{T(1) > t\}]dt - \int_0^\tau \mathbb{E}[I\{T(0) > t\}]dt \\ &= \int_0^\tau S_1(t)dt - \int_0^\tau S_0(t)dt \\ &= \int_0^\tau [S_1(t) - S_0(t)]dt\end{aligned}$$

with $S_a(t) = P(T(a) > t)$, the probability of surviving at time t when treatment $A = a$.

9.2 Cumulative hazard inversion method

When X, a random continuous variable, follow an exponential law (X ~ $\varepsilon(\lambda)$): the corresponding repartition function is: $F_\lambda(x) = P(X \geq x) = 1 - \exp(-\lambda x)$ and the density function is $f_\lambda(x) = \lambda \exp(-\lambda x)$.

F_λ is bijective from $\mathcal{R}^+$ to $]0;1[$ thus, F_λ^{-1} exists and is also bijective from $]0;1[$ to $\mathcal{R}^+$. The inverse of the repartition function is: $F^{-1}(u) = \frac{-log(1-u)}{\lambda}$ where U ~ $\mathcal{U}(0,1)$ and $\frac{-log(1-u)}{\lambda}$ ~ $\varepsilon(\lambda)$.

In knowing that 1-U ~$\mathcal{U}(0,1)$, we can also simulate X as: $\frac{-log(u)}{\lambda}$ ~ $\varepsilon(\lambda)$.

Following this results, in the case where T, the survival time, follow an exponential distribution (T ~ $\varepsilon(\lambda)$). The variable T can be simulated as: $F_\lambda^{-1}(U) = \frac{-log(U)}{\lambda}$ where U ~ $\mathcal{U}(0,1)$.

9.3 Trapezoidal method for integration (TO CHANGE)

The trapezoidal integration method is a numerical technique used to estimate the integral of a function over a given interval by approximating the area under the curve with trapezoids. This method is often employed when the function lacks a simple analytical form or when the integral cannot be computed exactly.

Suppose we want to estimate the integral of a function $f(x) over the interval $[a, b]$. The trapezoidal method divides this interval into n sub intervals of width h, where $h = \frac{b-a}{n}$.

Each sub interval is approximated by a trapezoid whose bases are the values of the function $f(x)$ at the endpoints of the sub interval.

The general formula for the area of a trapezoid is: $A = \frac{(b_1+b_2)\times h}{2}$ where b_1 and b_2 are the lengths of the parallel bases of the trapezoid, and h is its height.

To estimate the integral of $f(x)$ over each sub interval, we calculate the area of each trapezoid and sum them up.

The formula for the trapezoidal integration method for a single pair of trapezoids is:

$$\text{Area of trapezoid} = \frac{(f(x_i) + f(x_{i+1})) \times h}{2}$$

where $ x_i $ and $ x_{i+1} $ are the lower and upper limits of sub interval $ i $ respectively.

To estimate the integral over the entire interval, we sum the areas of all the trapezoids:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} \frac{(f(x_i) + f(x_{i+1})) \times h}{2}$$

Below is an example of the trapezoidal integration method used to estimate the integral of $ f(x) = xˆ2 $ over the interval $[0, 1]$ with $ n = 4 $ sub intervals:

$$\int_0^1 x^2\,dx \approx \frac{(f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + f(x_4)) \times h}{2}$$

Where

$$h = \frac{1-0}{4} = \frac{1}{4}$$

Thus,

$$\int_0^1 x^2 dx \approx \frac{(0 + 2(1/16) + 2(1/4) + 2(9/16) + 1) \times 1/4}{2}$$

9.4 Other expression of Hazard ratio

To prove that the hazard ratio is the ratio of the log of survival probabilities:

$$HR = \frac{log(P(T > t|A = 1))}{log(P(T > t|A = 0))}$$

Let's consider t fixed, the survival probabilities can be expressed for each group up to t as follows (based on the fact that $\Lambda(t) = \int_0^t \lambda(u)du = -ln(S(t))$):

$$
\begin{aligned}
log(P(T > t|A = 1)) &= -\int_0^t \lambda_0(s) exp(\beta_1 X_1 + ... + \beta_p X_p + \beta A) ds \\
&= -\int_0^t \lambda_0(s) exp(\beta_1 X_1 + ... + \beta_p X_p + \beta) ds
\end{aligned}
$$

$$
log(P(T > t|A = 0)) = -\int_0^t \lambda_0(s) exp(\beta_1 X_1 + ... + \beta_p X_p) ds
$$

Dividing these two probabilities, we get:

$$
\begin{aligned}
\frac{log(P(T > t|A = 1))}{log(P(T > t|A = 0))} &= \frac{-\int_0^t \lambda_0(s) exp(\beta_1 X_1 + ... + \beta_p X_p + \beta A) ds}{-\int_0^t \lambda_0(s) exp(\beta_1 X_1 + ... + \beta_p X_p) ds} \\
&= \exp(\beta)
\end{aligned}
$$

9.5 Identifiability of Δ^τ

$\Delta^\tau = I\{T \wedge \tau < C|A = 1\}$ cannot be computed directly as we don't have access to $T \wedge \tau$ but to $\widetilde{T} \wedge \tau$. Here's an expression of Δ^τ in integrate $\widetilde{T} \wedge \tau$:

$$
\begin{aligned}
\Delta^\tau &= I\{T \wedge \tau < C\} \\
&= I\{min(T, \tau) < C\} \\
&= \underbrace{I\{C > \tau\}.I\{T \geq \tau\}}_{1} + \underbrace{I\{C > T\}.I\{T \leq \tau\}}_{2} \\
&= I\{\widetilde{T} \geq \tau\} + I\{\widetilde{T} \leq \tau\}.I\{C \geq \widetilde{T}\} \\
&= I\{\widetilde{T} \geq \tau\} + I\{\widetilde{T} \leq \tau\}.\Delta
\end{aligned}
$$

As $\widetilde{T} = min(C, T)$ and that C and T is superior to τ, then the first term of the sum is equal to $I\{\widetilde{T} \geq \tau\}$.

The second term $I\{C > T\}.I\{T \leq \tau\}$ is equal to $I\{\widetilde{T} \leq \tau\}.I\{C \geq \widetilde{T}\}$ because when C is superior to T, then $\widetilde{T} = T$.

9.6 Buckley-James trasnformation

$$\begin{aligned}
\mathbb{E}[T^*|X,A] &= \mathbb{E}[T^*\Delta^\tau|X,A] + \mathbb{E}[T^*(1-\Delta^\tau)|X,A] \\
&= \mathbb{E}[\Delta^\tau(\tilde{T}\wedge\tau)|X,A] + \\
&\quad \mathbb{E}[(1-\Delta^\tau)\mathbb{E}[T\wedge\tau|X,A,T\wedge\tau \geq \tilde{T}\wedge\tau]|X,A] \\
&= \mathbb{E}[\Delta^\tau(\tilde{T}\wedge\tau)|X,A] + \\
&\quad \mathbb{E}[(1-\Delta^\tau)\mathbb{E}[T\wedge\tau|X,A,T\wedge\tau \geq C]|X,A] \\
&= \mathbb{E}[\mathbf{1}\{T\wedge\tau < C\}(\tilde{T}\wedge\tau)|X,A] + \\
&\quad \mathbb{E}[\mathbf{1}\{T\wedge\tau \geq C\}\mathbb{E}[T\wedge\tau|X,A,T\wedge\tau \geq C]|X,A] \\
&= \mathbb{E}[\tilde{T}\wedge\tau|X,A,T\wedge\tau < C]\mathbb{P}[T\wedge\tau < C|X,A] + \\
&\quad \mathbb{E}[\mathbb{E}[T\wedge\tau|X,A,T\wedge\tau \geq C]|X,A,T\wedge\tau \geq C]\mathbb{P}[T\wedge\tau \geq C|X,A] \\
&= \mathbb{E}[T\wedge\tau|X,A,T\wedge\tau < C]\mathbb{P}[T\wedge\tau < C|X,A] + \\
&\quad \mathbb{E}[T\wedge\tau|X,A,T\wedge\tau \geq C]\mathbb{P}[T\wedge\tau \geq C|X,A] \\
&= \mathbb{E}[\mathbf{1}\{T\wedge\tau < C\}(T\wedge\tau)|X,A] + \\
&\quad \mathbb{E}[\mathbf{1}\{T\wedge\tau \geq C\}(T\wedge\tau)|X,A] \\
&= \mathbb{E}[T\wedge\tau|X,A]
\end{aligned}$$

9.7 Influence function

9.7.1 Taylor expansion of a function

The Taylor expansion of a function v around a point 1 is given by:

$$\begin{aligned}
T(P_0) = v(0) &= v(1) + v'(1)(0-1) - R_2 \\
&= T(P_1) + \frac{\partial}{\partial\epsilon}T(P_\epsilon)\bigg|_{\epsilon=1}(0-1) - R_2,
\end{aligned}$$

where R_2 is the remainder term of the Taylor expansion.

The von Mises expansion is a generalization of the Taylor expansion for functions in an infinite dimension.

9.7.2 Derive a corrected estimator with influence function

Based on the notation in Section 4.2.4.1, the direction of the fluctuation is derived by the score function:

$$s_{\epsilon=0}(O) = \frac{d}{d\epsilon}\log\left(\frac{dP_\epsilon}{dP_0}(O)\right)\bigg|_{\epsilon=0} = \frac{\frac{d}{d\epsilon}dP_\epsilon(O)}{dP_\epsilon(O)}\bigg|_{\epsilon=0}$$

The goal is to find a quantity that can correct the bias of the estimator $\psi(\bar{P})$ due to the perturbation. For the moment, the estimand of interest is $\psi(P) = \mathbb{E}_P[\mathbb{E}_P(T\wedge\tau|X,A=1)]$.

Recall the von Mises expansion (can be seen as distributional analog of a Taylor expansion see Annex Section 9):

$$\begin{aligned}\psi(\bar{P})-\psi(P)&=\int\varphi(z;\bar{P})d(\bar{P}-P)(z)+R_2(\bar{P},P)\\ \implies \psi(P)&=\psi(\bar{P})-\int\varphi(z;\bar{P})d(\bar{P}-P)(z)-R_2(\bar{P},P)\end{aligned}\tag{22}$$

where z is the observed data, $\varphi(z;P)$ is the influence function defined as a mean-zero (thus $\int\varphi(z;\bar{P})d(\bar{P})(z)=0$), finite variance function and $R_2(\bar{P},P)$ is a second-order remainder term.

This expansion shows that plug-in estimator such as G-formula (presented in Section 4.2.1.1) has first-order bias and suggests how to correct it by estimating the bias term $-\int\varphi(z;\bar{P})d(\bar{P}-P)(z)$ (Kennedy 2023) (a straightforward estimator could be the sample average of the influence function on the distribution P). Thus, the corresponding bias-corrected estimator is $\hat{\psi}=\psi(\hat{\mathbb{P}})+\mathbb{P}\{\varphi(Z,\hat{\mathbb{P}})\}$.

It exists several methods to compute influence function, the most general is to compute the following pathwise derivative $\psi^{'}(P_\epsilon)$ derived from Equation 22:

$$\begin{aligned}\left.\frac{\psi\left(P_\epsilon\right)-\psi(P)}{\epsilon}\right|_{\epsilon=0}&=\left.\frac{d}{d\epsilon}\psi\left(P_\epsilon\right)\right|_{\epsilon=0}\\ &=\left.\int\varphi(O;P)\underbrace{\frac{d}{d\epsilon}dP_\epsilon(O)}_{\equiv s_\epsilon(O)dP_\epsilon(O)}\right|_{\epsilon=0}\\ &=\left.\int\varphi(O;P)s_\epsilon(O)dP_\epsilon(O)\right|_{\epsilon=0}\\ &=\mathbb{E}_{P_\epsilon}\left[\varphi(O;P)s_\epsilon(O)\right]\big|_{\epsilon=0}\end{aligned}\tag{23}$$

One simple method from Kennedy (2023) to compute this quantity Equation 23 is to:

1- Consider that data are discrete.

2- Treat the influence function as derivatives to allow the differentiation rules.

Then derive the correction term.

9.8 The augmented inverse probability of censoring transformation

In the context of non parametric regression (from D. Rubin and Laan (2007)):

$$\begin{aligned}Y^\star(O)&=Y^\star_{\bar{F},\bar{G}}(O)\\ &=\frac{Y\Delta}{\bar{G}(Y\mid W)}+\frac{Q_{\bar{F}}(W,C)(1-\Delta)}{\bar{G}(C\mid W)}-\int_{-\infty}^{\tilde{Y}}\frac{Q_{\bar{F}}(W,c)}{\bar{G}^2(c\mid W)}dG(c\mid W)\\ &=T_1+T_2-T_3.\end{aligned}$$

First observe that,

$$
\begin{aligned}
E\left[T_1 \mid W\right] &= E\left[\left.\frac{Y\Delta}{\bar{G}_1(Y \mid W)}\right| W\right] \\
&= E\left[E\left[\left.\frac{Y\Delta}{\bar{G}_1(Y \mid W)}\right| W, Y\right]\middle| W\right] \\
&= E\left[\left.\frac{Y}{\bar{G}_1(Y \mid W)} P(\Delta = 1 \mid W, Y)\right| W\right] \\
&= E\left[\left.\frac{Y}{\bar{G}_1(Y \mid W)} \bar{G}(Y \mid W)\right| W\right] \\
&= \int_{-\infty}^{\tau} y \frac{\bar{G}(y \mid W)}{\bar{G}_1(y \mid W)} dF(y \mid W).
\end{aligned}
$$

Next note that,

$$
\begin{aligned}
E\left[T_2 \mid W\right] &= E\left[\left.\frac{Q_1(W, C)(1-\Delta)}{\bar{G}_1(C \mid W)}\right| W\right] \\
&= E\left[E\left[\left.\frac{Q_1(W, C)(1-\Delta)}{\bar{G}_1(C \mid W)}\right| W, C\right]\middle| W\right] \\
&= E\left[\left.\frac{Q_1(W, C)}{\bar{G}_1(C \mid W)} P(\Delta = 0 \mid W, C)\right| W\right] \\
&= E\left[\left.\frac{Q_1(W, C)}{\bar{G}_1(C \mid W)} \bar{F}(C \mid W)\right| W\right] \\
&= E\left[\left.\frac{\bar{F}(C \mid W)}{\bar{F}_1(C \mid W)} \int_C^{\tau} y dF_1(y \mid W) \bar{G}_1^{-1}(C \mid W)\right| W\right] \\
&= \int_{-\infty}^{\infty} \frac{\bar{F}(c \mid W)}{\bar{F}_1(c \mid W)} \bar{G}_1^{-1}(c \mid W) \left\{\int_c^{\tau} y dF_1(y \mid W)\right\} dG(c \mid W) \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left\{\frac{\bar{F}(c \mid W)}{\bar{F}_1(c \mid W)} \bar{G}_1^{-1}(c \mid W) 1(c < y < \tau) y\right\} dF_1(y \mid W) dG(c \mid W) \\
&= \int_{-\infty}^{\tau} y \left\{\int_{-\infty}^{y} \frac{\bar{F}(c \mid W)}{\bar{F}_1(c \mid W)} \bar{G}_1^{-1}(c \mid W) dG(c \mid W)\right\} dF_1(y \mid W).
\end{aligned}
$$

Finally, observe that,

$$
\begin{aligned}
E\left[T_3 \mid W\right] &= E\left[\left.\int_{-\infty}^{\min(Y,C)} \frac{Q_1(W,c)}{\bar{G}_1^2(c \mid W)} dG_1(c \mid W)\right| W\right] \\
&= E\left[\left.\int_{-\infty}^{\infty} 1(Y>c)1(C>c)\frac{Q_1(W,c)}{\bar{G}_1^2(c \mid W)} dG_1(c \mid W)\right| W\right] \\
&= \int_{-\infty}^{\infty} P(Y>c, C>c \mid W)\frac{Q_1(W,c)}{\bar{G}_1^2(c \mid W)} dG_1(c \mid W) \\
&= \int_{-\infty}^{\infty} P(Y>c \mid W)P(C>c \mid W)\frac{Q_1(W,c)}{\bar{G}_1^2(c \mid W)} dG_1(c \mid W) \\
&= \int_{-\infty}^{\infty} \bar{F}(c \mid W)\bar{G}(c \mid W)\frac{Q_1(W,c)}{\bar{G}_1^2(c \mid W)} dG_1(c \mid W) \\
&= \int_{-\infty}^{\infty} \frac{\bar{G}(c \mid W)}{\bar{G}_1^2(c \mid W)}\frac{\bar{F}(c \mid W)}{\bar{F}_1(c \mid W)}\left.\int_c^{\tau} y dF_1(y \mid W)\right\} dG_1(c \mid W) \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left\{\frac{\bar{G}(c \mid W)}{\bar{G}_1^2(c \mid W)}\left\{\frac{\bar{F}(c \mid W)}{\bar{F}_1(c \mid W)} 1(c<y<\tau)y\right\} dF_1(y \mid W) dG_1(c \mid W)\right. \\
&= \int_{-\infty}^{\tau} y\left\{\int_{-\infty}^{y} \frac{\bar{F}(c \mid W)}{\bar{F}_1(c \mid W)}\frac{\bar{G}(c \mid W)}{\bar{G}_1^2(c \mid W)} dG_1(c \mid W)\right\} dF_1(y \mid W) \\
&= \int_{-\infty}^{\tau} y\int_{-\infty}^{y} \frac{\bar{F}(c \mid W)}{\bar{F}_1(c \mid W)}\frac{\bar{G}(c \mid W)}{\bar{G}_1^2(c \mid W)}\frac{dG_1}{dG}(c \mid W) dG(c \mid W) dF_1(y \mid W)
\end{aligned}
$$

Thus, combining the previous expression, we see that,

$$
\begin{aligned}
E\left[Y^{\star}(O) \mid W\right] = E\left[T_1 + T_2 - T_3 \mid W\right] &= E\left[T_1 \mid W\right] + E\left[T_2 \mid W\right] - E\left[T_3 \mid W\right] \\
&= \int_{-\infty}^{\tau} y\frac{\bar{G}(y \mid W)}{\bar{G}_1(y \mid W)} dF(y \mid W) \\
&+ \int_{-\infty}^{\tau} y\left\{\int_{-\infty}^{y} \frac{\bar{F}(c \mid W)}{\bar{F}_1(c \mid W)}\bar{G}_1^{-1}(c \mid W) dG(c \mid W)\right\} dF_1(y \mid W) \\
&- \int_{-\infty}^{\tau} y\int_{-\infty}^{y} \frac{\bar{F}(c \mid W)}{\bar{F}_1(c \mid W)}\frac{\bar{G}(c \mid W)}{\bar{G}_1^2(c \mid W)}\frac{dG_1 dG}{dG}(c \mid W) dF_1(y \mid W) \\
&= \int_{-\infty}^{\tau} y\frac{\bar{G}(y \mid W)}{\bar{G}_1(y \mid W)} dF(y \mid W) \\
&+ \int_{-\infty}^{\tau} y\left\{\int_{-\infty}^{y} \frac{\overline{F}(c|W)}{\overline{F}_1(c|W)}\left[\frac{1}{\bar{G}_1(c \mid W)}\right.\right. \\
&\left.\left. - \frac{\bar{G}^2(c \mid W)}{\bar{G}_1^2(c \mid W)}\frac{dG_1}{dG}(c \mid W)\right] dG(c \mid W)\right\} dF_1(y \mid W) \\
= &\int_{-\infty}^{\tau} y\frac{\bar{G}(y \mid W)}{\bar{G}_1(y \mid W)} dF(y \mid W) \\
&- \int_{-\infty}^{\tau} y\left\{\int_{-\infty}^{y} \frac{\bar{F}(c \mid W)}{\bar{F}_1(c \mid W)}\left[\frac{d}{dc}\frac{\bar{G}(c \mid W)}{\bar{G}_1(c \mid W)}\right] dc\right\} dF_1(y \mid W).
\end{aligned}
$$

If $G = G_1$, then $\frac{d}{dc}\frac{\bar{G}(c|W)}{\bar{G}_1(c|W)} = 0$, so:

$$
E\left[Y^{\star}(O) \mid W\right] = \int_{-\infty}^{\tau} y\frac{\bar{G}(y \mid W)}{\bar{G}(y \mid W)} dF(y \mid W) = \int_{-\infty}^{\tau} y dF(y \mid W) = m(W).
$$

If $F = F_1$, then it becomes,

$$\begin{aligned}
E\left[Y^{\star}(O) \mid W\right] &= \int_{-\infty}^{\tau} y \frac{\bar{G}(y \mid W)}{\bar{G}_1(y \mid W)} dF(y \mid W) \\
&- \int_{-\infty}^{\infty} y \left\{ \int_{-\infty}^{y} \frac{d}{dc} \frac{\bar{G}(c \mid W)}{\bar{G}_1(c \mid W)} dc \right\} dF(y \mid W) \\
&= \int_{-\infty}^{\tau} y \left\{ \frac{\bar{G}(y \mid W)}{\bar{G}_1(y \mid W)} - \int_{-\infty}^{y} \frac{d}{dc} \bar{G}(c \mid W) \right. \\
&= \int_{-\infty}^{\tau} y \left\{ \frac{\bar{G}(y \mid W)}{\bar{G}_1(y \mid W)} - \left[\frac{\bar{G}(y \mid W)}{\bar{G}_1(y \mid W)} - \frac{\bar{G}(-\infty \mid W)}{\bar{G}_1(-\infty \mid W)} \right] \right\} dF(y \mid W) \\
&= \int_{-\infty}^{\tau} y \left\{ \frac{\bar{G}(y \mid W)}{\bar{G}_1(y \mid W)} - \frac{\bar{G}(y \mid W)}{\bar{G}_1(y \mid W)} + \frac{1}{1} \right\} dF(y \mid W) \\
&= \int_{-\infty}^{\tau} y dF(y \mid W) \\
&= m(W).
\end{aligned}$$

It proves that $E\left[Y^{\star}(O) \mid W\right] = m(W) = E[Y|W]$, if $G = G_1$ or $F = F_1$.

9.8.0.1 Simplification for implementation of AIPCW

$$\begin{aligned}
\int_0^{\bar{T} \wedge T} \frac{Q_s(c|X_i, A_i)}{S_c^2(c|X_i, A_i)} d(1 - S_c(c|X_i, A_i)) &= \int_0^{\bar{T} \wedge T} \frac{Q_s(c|X_i, A_i)}{S_c^2(c|X_i, A_i)} \times -dS_c(c|X_i, A_i) \\
&= -\int_0^{\bar{T} \wedge T} \frac{dS_c(c|X_i, A_i)}{S_c(c|X_i, A_i)} \times \frac{Q_s(c|X_i, A_i)}{S_c(c|X_i, A_i)} \\
&= -\int_0^{\bar{T} \wedge T} d \log(S_c(c|X_i, A_i)) \times \frac{Q_s(c|X_i, A_i)}{S_c(c|X_i, A_i)} ds \\
&= \int_0^{\bar{T} \wedge T} \frac{\lambda_c(s|X_i, A_i) Q_s(c|X_i, A_i)}{S_c(c|X_i, A_i)} ds
\end{aligned}$$

9.9 Classical methods to evaluate survival data

9.9.1 Cox model

The Cox proportional hazards estimator is mainly used to evaluate treatment effect in clinical studies (add ref). It is the most used model in survival analysis.

This estimator of the instantaneous hazard function is often referred to as a ***semi-parametric estimator*** due to its mixed nature, combining parametric and nonparametric components.

The Cox survival model, introduced by David R. Cox, is considered semi-parametric because it has two distinct components:

- ***Parametric Component***: The model includes a parametric term that specifies the functional form of the effect of covariates on the relative hazard. However, it does not impose particular restrictions on the form of the baseline survival function.

- ***Nonparametric Component***: The baseline instantaneous hazard function (the hazard function for an individual with all covariates equal to zero) is not specified. Instead, it is estimated in a nonparametric manner, allowing the model to adapt to different shapes of the survival function.

In the case of two states survival models, it can be defined as follows:

$$\lambda_i(t) = \lambda_0(t)exp(\beta_1 X_{1i} + ... + \beta_p X_{pi}) \tag{24}$$

This model is called ***proportional hazards*** because the ratio of instantaneous hazards is constant over time:

$$\frac{\lambda_i(t)}{\lambda_j(t)} = exp(\beta_1 X_{1i} + ... + \beta_p X_{pi} - \beta_1 X_{1j} - ... - \beta_p X_{pj})$$

is independent of time.

The model is also ***log-linear***:

$$\ln(\lambda_i(t)) = \ln(\lambda_0(t)) + \beta_1 X_{1i} + ... + \beta_p X_{pi}$$

.

Basically, cox model provides variables coefficients in maximizing the partial likelihood of Cox model. The underlying principle is that the parameter values that maximize the likelihood of the observed data are the most plausible values, given the statistical model chosen. The maximization of the partial likelihood is calculated instead of the total likelihood to significantly reduce computation time. However, censoring and lifetime must be independent, and censoring must be non-informative.

The exponential of the corresponding coefficients is the hazard ratio:

- For a binary variable coded 0/1: $HR = \exp(\text{coef})$
- For a binary variable coded a/b: $HR = \exp(\text{coef} \cdot (b - a))$
- For a continuous variable, exp(coef) corresponds to the hazard ratio for a one-unit increase in the variable. For example, for “age” variable, each additional year of life multiply the instantaneous risk of death by a factor of 1,01, an increase of 1%.

But even if the variable corresponds to treatment assignment and that the data fit all the model requirement (log linearity, hazard proportional), this quantity is not a causal quantity and can lead to major confounding bias when the data is observational.

Implementation

```
# Cox model
library(survival)
Cox_model <- function(data,tau,X.names=c("X1","X2","X3","X4")){
  data$T_obs_tau <- ifelse(data$T_obs>=tau,tau,data$T_obs)
  data$status_tau <- as.numeric((data$T_obs>=tau) | (data$T_obs<tau &  data$status ==
  outcome <- paste(c('Surv(',"T_obs_tau",',',"status_tau",')'),collapse="")
  f <- as.formula(paste(outcome, paste(c(X.names,'A'), collapse = " + "), sep = " ~ "
  fitS <- suppressWarnings(coxph(f, data=data, x=TRUE))
  fitS$coefficients[is.na(fitS$coefficients)] <- 0
  return(fitS)}
```

9.9.2 Hazard ratio from cox model are not a causal measure

Analyses of time to event endpoint in RCTs in epidemiological studies focus mainly on HR by the use of Cox model. However, the HR is not known to be a causal measure for different reasons (Hernán 2010; Martinussen, Vansteelandt, and Andersen 2020).

Explanation

If we focus on the expression on HR based on Cox model:

In a Cox proportional hazards model, we model the hazard function as a function of time and covariates. Mathematically, this can be expressed as in Equation 24:

$$exp(\beta) = \frac{lim_{h\to 0}P(t \leq T < t+h|T \geq t, A=1)}{lim_{h\to 0}P(t \leq T < t+h|T \geq t, A=0)}$$

If we have two treatment groups, say $T = 1$ for the treated group and $T = 0$ for the control group, then the hazard ratio between the two groups at a certain time t is $\exp(\beta)$, where β is the coefficient associated with the treatment variable T.

HR can be expressed in terms of survival probabilities as follows (explained in annex Section 9.4):

$$exp(\beta) = \frac{log(P(T > t|A=1))}{log(P(T > t|A=0)}$$

Under Equation 5, the hazard ratio could be seen as:

$$exp(\beta) = \frac{log(P(T(1) > t))}{log(P(T(0) > t)}$$

This shows that under the proportional hazards assumption, the HR is a function of the survival function of the same population if everyone were treated and if everyone were not treated. But the log function make the interpretation of HR as a causal measure difficult.

Also, under Equation 5, the HR could be expressed as:

$$exp(\beta) = \frac{lim_{h\to 0}P(t \leq T(1) < t+h|T(1) \geq t)}{lim_{h\to 0}P(t \leq T(0) < t+h|T(0) \geq t)}$$

The group with and without treatment will fail to be comparable over the time if the treatment affects the outcome (Hernán 2010). Also, the interpretation of HR is complicated because of its non collapsibility. The HR is not a causal measure because it is not a weighted average of the stratum-specific HRs.

This paradox is due to what is called non-collapsibility of the Hazard Ratio. The average effect on a population could not be written as a weighted sum of effects on sub-populations.

9.10 Schoenfeld residuals test

The proportional hazards (PH) assumption can be checked using statistical tests and graphical diagnostics based on the ***scaled Schoenfeld residuals*** to test the proportional hazards assumption for each covariate included in a Cox refression model fit.

In knowing that β is the solution of the maximum likelihood estimate (explained in the next part):

$$\sum_{i=1}^{n}(x_i - E(x_i|R(t_i))) = 0$$

Firstly, let's define the Schoenfeld residuals at time T_i :

$$r_i^{\text{schoenfeld}} = x_i - E(x_i|R(t_i))$$

with m covariates in cox model : $r_i = (r_{i,1}, ..., r_{i,m})$

where x_i is the covariate vector of the individual experiencing the i-th event and $R(t_i)$ is the risk set at time t_i.

Basically, the Schoenfeld residuals measure the difference between the value of the covariate and the

The scaled Schoenfeld residuals are defined as follows :

$$\hat{r}_{i,m}^{\text{scaled}} = \frac{r_{i,m}}{\hat{V}_m}$$

with $\hat{V}_m$ the variance of $r_{i,m}$

To test the time dependent coefficient, for a single covariate X_m, the proportional hazards is expanded as :

$$\beta_m(t) = \beta_m + \beta_m * g_m(t)$$

where $g_m(t)$ is a predictable process.

Thus, the scaled Schoenfeld residuals are defined as follows :

$$\hat{r}_{i,m}^{\text{scaled}} = \frac{r_{i,m}(\beta_m)}{\hat{V}(\beta_m, t_i)}$$

with $\hat{V}_m$ the variance of $r_{i,m}$

The esperance of the scaled Schoenfeld residuals is equal to :

$$E(\hat{r}_{i,m}^{\text{scaled}}) = \beta_m(t_i) - \beta_m$$

Grambsch and Therneau have shown that the variance is stable over time, thus in plotting the values of $r_m^{\text{scaled}} + \beta_m$ against time, the plot should be flat if the proportional hazards assumption holds. Then a score test can be used to test the null hypothesis.

Basically, this Schoenfeld residuals measure the scaled difference between ***observed and expected values of covariates under the hypothesis of constant risk*** to assess whether the effect of covariates on risk is constant over time.

All these assumptions can be checked in using the function cox.zph() from survival package.

The statistic of the test is defined as follows :

- Null hypothesis : H_0 : ***The scaled schoenfeld residuals are not time dependents*** $(\beta_m(t_i) = \beta_m)$
- Alternative hypothesis : H_1 :The scaled schoenfeld residuals are time dependents

The proportional risk hypothesis is confirmed by a non-significant relationship between residuals and time, and refuted by a significant relationship.

Thus a ***low p-value indicates***:

- the Schoenfeld residuals are ***not constant over time***
- there is evidence that the ***variable/predictor may be time-dependent***

Thus proportional-hazards assumption (made when generating the coxph model) may be violated by this variable.

In our example, variable meal.cal and ph.karno have p-value<5% thus the proportional hazards hypothesis is violated. To resolve this problem, we can consider a time dependent covariate :

A violations of proportional hazards assumption can be resolved by:

- Adding covariate*time interaction : The interaction have to be known in analysing the type of interaction
- Stratification : The stratification is a way of splitting the population into subgroups according to the covariate.
- Partition the time axis : The time axis is divided into several intervals and the Cox model is applied to each interval.
- Use a different model such as accelerated failure time model or additive hazard model.

Session information

```
R version 4.2.3 (2023-03-15 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22631)

Matrix products: default

locale:
[1] LC_COLLATE=French_France.utf8  LC_CTYPE=French_France.utf8
[3] LC_MONETARY=French_France.utf8 LC_NUMERIC=C
[5] LC_TIME=French_France.utf8

attached base packages:
[1] splines   stats     graphics  grDevices datasets  utils     methods
[8] base

other attached packages:
 [1] gridExtra_2.3             forecast_8.21.1
 [3] survRM2_1.0-4             RISCA_1.0.5
 [5] mosaicData_0.20.4         ggformula_0.12.0
 [7] Matrix_1.6-5              lattice_0.20-45
 [9] tune_1.2.0                reticulate_1.35.0
[11] relsurv_2.2-9             date_1.2-42
[13] riskRegression_2023.12.21 grf_2.3.1
[15] rms_6.7-1                 Hmisc_5.1-1
[17] dplyr_1.1.4               survminer_0.4.9
[19] ggpubr_0.6.0              ggplot2_3.4.4
[21] MASS_7.3-58.2             survival_3.5-3

loaded via a namespace (and not attached):
  [1] backports_1.4.1      workflows_1.1.4      plyr_1.8.9
  [4] listenv_0.9.1        TH.data_1.1-2        digest_0.6.34
  [7] foreach_1.5.2        htmltools_0.5.7      yardstick_1.3.1
 [10] parsnip_1.2.1        fansi_1.0.6         magrittr_2.0.3
 [13] checkmate_2.3.1      cluster_2.1.4       doParallel_1.0.17
 [16] mosaicCore_0.9.4.0   recipes_1.0.10      globals_0.16.2
 [19] mets_1.3.4           gower_1.0.1         xts_0.13.2
 [22] sandwich_3.1-0       hardhat_1.3.1       timechange_0.3.0
 [25] tseries_0.10-55      rsample_1.2.1       dials_1.2.1
 [28] colorspace_2.1-0     haven_2.5.4         xfun_0.42
 [31] jsonlite_1.8.8       zoo_1.8-12          iterators_1.0.14
 [34] glue_1.7.0           gtable_0.3.4        nnls_1.5
 [37] ipred_0.9-14         MatrixModels_0.5-3  car_3.1-2
 [40] kernlab_0.9-32       future.apply_1.11.1 shape_1.4.6
 [43] quantmod_0.4.26      abind_1.4-5         SparseM_1.81
 [46] scales_1.3.0         mvtnorm_1.2-4       mosaic_1.9.1
 [49] rstatix_0.7.2        Rcpp_1.0.12         xtable_1.8-4
 [52] cmprsk_2.2-11        htmlTable_2.4.2     GPfit_1.0-8
 [55] foreign_0.8-84       km.ci_0.5-6         Formula_1.2-5
 [58] stats4_4.2.3         lava_1.7.3          prodlim_2023.08.28
 [61] SuperLearner_2.0-29  glmnet_4.1-8        htmlwidgets_1.6.4
 [64] RColorBrewer_1.1-3   farver_2.1.1        pkgconfig_2.0.3
 [67] nnet_7.3-18          utf8_1.2.4          caret_6.0-94
 [70] labeling_0.4.3       tidyselect_1.2.0    rlang_1.1.3
 [73] DiceDesign_1.10      reshape2_1.4.4      munsell_0.5.0
 [76] tools_4.2.3          cli_3.6.2           generics_0.1.3
 [79] ggridges_0.5.6       broom_1.0.5         evaluate_0.23
 [82] stringr_1.5.1        fastmap_1.1.1       yaml_2.3.8
 [85] ModelMetrics_1.2.2.2 knitr_1.45          timereg_2.0.5
 [88] survMisc_0.5.6       purrr_1.0.2         future_1.33.1
 [91] nlme_3.1-162         quantreg_5.97       gam_1.22-3
 [94] compiler_4.2.3       rstudioapi_0.15.0   curl_5.2.0
 [97] png_0.1-8            ggsignif_0.6.4      tibble_3.2.1
[100] lhs_1.1.6            statmod_1.5.0       stringi_1.8.3
[103] cubature_2.1.0       forcats_1.0.0       urca_1.3-3
[106] KMsurv_0.1-5         vctrs_0.6.5         pillar_1.9.0
[109] lifecycle_1.0.4      furrr_0.3.1         lmtest_0.9-40
[112] data.table_1.15.0    R6_2.5.1            renv_1.0.4
[115] parallelly_1.37.0    codetools_0.2-19    polspline_1.1.24
[118] withr_3.0.0          fracdiff_1.5-3      multcomp_1.4-25
[121] hms_1.1.3            parallel_4.2.3      quadprog_1.5-8
[124] labelled_2.12.0      grid_4.2.3          rpart_4.1.19
[127] timeDate_4032.109    tidyr_1.3.1         class_7.3-21
[130] rmarkdown_2.25       carData_3.0-5       TTR_0.24.4
[133] pROC_1.18.5          numDeriv_2016.8-1.1 lubridate_1.9.3
[136] base64enc_0.1-3
```

10 TO DO LIST

- Problème sur mis-spec simulation
- Correct mis-specification, ajouter des détails:
  - RCT 2: Ajouter 8,000 obs pour IPCW KM + IPTW-IPCW KM qui ne convergent pas franchement

  - OBS 1:
    - Ajouter 8,000 obs pour Causal_survival_forest qui ne converge pas franchement
    - Ajouter un autre plot avec g-formula forest et aiptw-aipcw forest
  - OBS 2: AJouter 8,000 observations pour causal_survival_forest
  - Complex:
    - Ajouter 8,000 obs pour AIPTW-AIPCW et Causal_survival forest
    - Ajouter g-formula parametric
    - Investiguer pourquoi IPTW-IPCW est biaisé
- Prendre en compte les commentaires (jsq page 57)
- Refaire toutes les propriétés après avoir lu cours
- Refaire mis-specification sur le DGP avec interaction
- Faire une violation de positivité sur le DGP non linéaire pour voir si l'extrapolation est ok

ON HOLD: - Refaire tourner simulation avec mis-specification avec 100 simulations chacune (pour le moment que 10) - Refaire tourner la positivité RCT 2 avec 100 simulations pour que ce soit comparable - Intensifié la violation à la positivité de la censure pour Obs 2 et faire tourner avec 100 simulations

11 Question

- Faut-il appeler les packages à chaque fois qu'ils sont utilisés ?