TP — Inférence causale en épidémiologie

David Hajage

Sommaire

Introduction

Bienvenue dans ce TP sur l'**inférence causale en épidémiologie** !

Scénario

Nous disposons d'une **étude de cohorte observationnelle** dont l'objectif est d'évaluer l'effet d'une exposition A (par exemple un traitement médical) sur la **survie à 3 ans**.

Les patients sont suivis pendant 3 ans avec des visites annuelles. À chaque visite, leur statut d'exposition peut changer : un patient initialement exposé peut arrêter, et inversement. Un facteur de confusion L, dépendant du temps, est mesuré à chaque visite. Une covariable initiale X est également disponible.

Deux questions causales

Ce TP vous guidera pour répondre à **deux questions causales distinctes** :

i Question 1 — Effet de l'exposition initiale (analogue-ITT)

Quelle serait la survie si tout le monde avait été exposé dès le début de l'étude ($A_0 = 1$), et quelle serait la survie si personne ne l'avait été ($A_0 = 0$) ?
Cette question porte sur l'*initiation* de l'exposition au temps 0, indépendamment de ce qui se passe ensuite (les patients peuvent arrêter ou débuter le traitement par la suite).
→ Abordée en **Parties 1 et 2** (G-computation puis IPTW)

i Question 2 — Effet de l'exposition maintenue (analogue per-protocol)

Quelle serait la survie si tout le monde avait été exposé et l'était resté tout au long du suivi ($\bar{a} = 1$), et si personne n'avait jamais été exposé ($\bar{a} = 0$) ?
Cette question porte sur le *maintien* de l'exposition tout au long du suivi. Elle nécessite de traiter les déviations de traitement comme une forme de censure.
→ Abordée en **Partie 3** (IPTW + IPCW)

Structure du TP

Partie	Méthode	Question	Durée estimée
Présentation	Analyse brute (Kaplan-Meier)	Description	~10 min
Partie 1	**G-computation**	Q1 — effet de l'exposition initiale	~20 min
Partie 2	**IPTW**	Q1 — même objectif, approche différente	~25 min
Partie 3	**IPTW + IPCW**	Q2 — effet de l'exposition maintenue	~30 min
Conclusion	Comparaison des résultats	Synthèse	~5 min

Les durées annoncées sont approximatives, vous pourrez terminer chez vous ce que vous n'auriez pas eu le temps de faire aujourd'hui. Vous pouvez télécharger le pdf de ce TP via le lien tout en bas de cette page.

Comment utiliser ce TP

Les exercices utilisent **WebR** : le code R s'exécute directement dans votre navigateur, sans installation.

- Chaque bloc de code peut être modifié et exécuté avec le bouton **Run Code**
- Les **solutions** sont là pour être consultées librement et sans hésitation. L'objectif de ce TP n'est pas de retrouver le code depuis une page blanche, mais de **suivre et comprendre chaque étape** : lisez-les comme un exemple commenté, exécutez-les, modifiez-les. Certaines analyses mobilisent des compétences R avancées — c'est normal de s'appuyer sur les solutions.
- Les variables créées sur une page ne sont **pas disponibles** sur les pages suivantes ; elles sont reconstruites automatiquement dans chaque section (voir *Note sur la session R* au début de chaque partie).

Présentation des données

```
#| context: setup
if (Sys.info()["nodename"] == "emscripten") {
  base_url <- sub("/[^/]*$", "/", quarto_page_url)
  download.file(
    paste0(base_url, "df.csv"),
    "df.csv", quiet = TRUE
  )
}
library(survival)
library(dplyr)
df <- read.csv("df.csv")
```

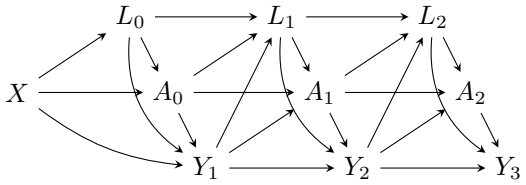
Présentation

Dans ce TP, nous utiliserons un jeu de données simulées (`df.csv`) contenant des informations sur **2000 individus** suivis dans le cadre d'une étude de cohorte observationnelle. L'objectif est d'évaluer l'effet d'un traitement A (exposé/non exposé, pouvant changer au cours du temps) sur la **mortalité à 3 ans**.

Les données sont au **format long** (*counting process*) : chaque individu contribue une ligne par période d'observation (ici, une ligne par visite annuelle).

Variable	Description
`id`	Identifiant patient
`T.start`	Début de la période d'observation (en années)
`T.stop`	Fin de la période d'observation (en années)
`A`	Traitement (0/1), **dépendant du temps**
`D`	Décès survenu avant `T.stop` (0/1)
`L`	Facteur de confusion **dépendant du temps** (0/1)
`X`	Covariable **initiale** (continue)

On suppose que les relations entre les variables suivent le DAG (*directed acyclic graph*) en temps discret ci-dessous, où Y_t désigne l'indicateur de décès à la période t ($t = 1, 2, 3$) :



Lecture du DAG :

- X (bleu foncé) — facteur de confusion **initial** : influence A_0, L_0 **et** Y_1.
- L_t (bleu clair) — facteur de confusion **dépendant du temps** : influencé par le traitement passé A_{t-1} et l'état de santé Y_t, il prédit à son tour A_t et Y_{t+1}.
- A_t (rouge) — exposition à chaque période, influencée par X, L_t et les valeurs passées.
- Y_t (gris) — indicateur de décès, influencé par A_{t-1} et L_{t-1}.

La structure $L_t \leftarrow A_{t-1}$ et $L_t \rightarrow A_t \rightarrow Y_{t+1}$ fait de L un **médiateur-confondant** dépendant du temps : les méthodes standards (régression ajustée sur L) ne suffisent pas — c'est l'objet des Parties 2 et 3.

Chargement et exploration

Chargez les données et affichez les premières lignes. Vérifiez les dimensions de la base et le nombre d'individus distincts.

```
#| context: interactive
library(survival)
library(dplyr)
df <- read.csv("df.csv")

## Affichez les premières lignes
```

```
library(survival)
library(dplyr)
df <- read.csv("df.csv")

head(df)
dim(df)                    # dimensions de la base
length(unique(df$id))      # nombre d'individus
sum(df$D)                  # nombre total de décès
df[df$id %in% 1:3, ]       # 3 premiers individus (plusieurs lignes chacun)
```

La base comporte **4376** lignes, **8** colonnes, **2000** individus distincts et **409** décès observés.

> **Note**
>
> **Format long :** chaque individu a autant de lignes qu'il a eu de périodes de suivi (en général 1 à 3, selon s'il est décédé ou perdu de vue). La colonne `T.start`/`T.stop` délimite chaque période. Le dernier enregistrement par individu indique son statut final (D) et son temps total de suivi.

Analyse brute (non ajustée)

Avant tout ajustement, comparons les courbes de survie selon le **traitement initial** A_0 (valeur de A à la première visite).

Créez la variable `A0` (valeur initiale de A pour chaque individu), puis tracez les courbes de Kaplan-Meier selon `A0`.

```
#| context: interactive
## Créez A0, puis tracez le Kaplan-Meier
df <- df |>
  group_by(id) |>
  mutate(A0 = first(A)) |>
  ungroup()
```

```
df <- df |>
  group_by(id) |>
  mutate(A0 = first(A)) |>
  ungroup()

km_brut <- survfit(Surv(T.start, T.stop, D) ~ A0, data = df)

plot(km_brut,
     col  = c("#1D2769", "#AC182E"), lwd = 2,
     xlab = "Temps (années)", ylab = "Probabilité de survie",
     main = "Kaplan-Meier brut selon A0 (non ajusté)")
legend("bottomleft",
       legend = c("A0 = 0 (non-exposés)", "A0 = 1 (exposés)"),
       col = c("#1D2769", "#AC182E"), lwd = 2)

## Survie à 3 ans
summary(km_brut, times = 3)
```

Que constatez-vous ? Cette différence brute a-t-elle une interprétation causale ?

> Réponse
>
> On observe une différence de survie entre les groupes $A_0 = 0$ et $A_0 = 1$. Cependant, cette différence **n'a pas d'interprétation causale** car les deux groupes ne sont pas comparables : les individus traités ($A_0 = 1$) et non traités ($A_0 = 0$) diffèrent sur X et L_0, qui sont à la fois des prédicteurs de l'exposition et du décès (facteurs de confusion).
> Les parties suivantes montrent comment corriger ce biais par G-computation et IPTW.

Ce que nous observons, ce que nous voulons estimer

L'analyse brute met en évidence une différence de survie entre les groupes $A_0 = 0$ et $A_0 = 1$. Mais cette différence **n'a pas d'interprétation causale directe** : les deux groupes ne sont pas

comparables à l'inclusion — les individus exposés et non exposés diffèrent sur X et L_0, qui prédisent à la fois l'exposition et le décès.

Les parties suivantes vont estimer deux effets causaux distincts à partir de ces données :

- **Parties 1 et 2** : effet de l'*initiation* de l'exposition au temps 0, ajusté sur les facteurs de confusion initiaux — d'abord par G-computation, puis par IPTW.
- **Partie 3** : effet du *maintien* de l'exposition tout au long du suivi, en traitant les déviations de stratégie comme une censure artificielle (IPTW + IPCW).

G-computation

```
#| context: setup
if (Sys.info()["nodename"] == "emscripten") {
  base_url <- sub("/[^/]*$", "/", quarto_page_url)
  download.file(
    paste0(base_url, "df.csv"),
    "df.csv", quiet = TRUE
  )
}
library(survival)
library(dplyr)
df <- read.csv("df.csv")
df <- df |>
  group_by(id) |>
  mutate(A0 = first(A), L0 = first(L)) |>
  ungroup()
```

Objectif — Question 1

Cette partie répond à la **Question 1** : quel serait l'effet d'une exposition initiale universelle ($a_0 = 1$ pour tous) par rapport à une absence totale d'exposition ($a_0 = 0$ pour tous) ?

On souhaite estimer les **courbes de survie contrefactuelles** :

$$S^{a_0=1}(t) = P(T^{a_0=1} > t) \quad \text{et} \quad S^{a_0=0}(t) = P(T^{a_0=0} > t)$$

La méthode utilisée ici est la **G-computation** (standardisation par régression) : on modélise le résultat Y en fonction de l'exposition et des covariables, puis on prédit les devenirs contrefactuels pour l'ensemble de la population.

> **Note**
>
> **Simplification pédagogique :** pour cette partie, le critère de jugement est traité comme binaire (décès en fin de suivi, oui/non). La G-computation avec un vrai modèle de survie est possible mais plus complexe. La Partie 2 traitera correctement le format de survie via l'IPTW.

Note sur la session R

Les variables suivantes sont disponibles dans vos chunks interactifs (construites dans le contexte de setup) :

— `df` : base de données au format long, avec `A0` (exposition initiale) et `L0` (facteur de confusion initial)

Étape 0 — Créer la base *baseline*

La G-computation s'applique sur une base **une ligne par individu**, avec les valeurs initiales des covariables et le statut final.

Créez `df_base` contenant : `id`, `A0`, `L0`, `X`, `time` (temps total de suivi = dernière valeur de `T.stop`), et `D` (statut final = dernière valeur de `D`).

```
#| context: interactive
## Créez df_base (1 ligne par individu)
```

```
df_base <- df |>
  group_by(id) |>
  summarise(
    A0   = first(A),
    L0   = first(L),
    X    = first(X),
    time = last(T.stop),
    D    = last(D)
  )
```

```
head(df_base)
nrow(df_base)
```

La base comporte **2000** lignes — une par individu. La variable `D` indique si l'individu est décédé avant la fin du suivi (3 ans).

Étape 1 — Estimer les modèles de résultat

La G-computation ajuste le modèle de résultat (D) séparément selon le groupe d'exposition A_0, en contrôlant pour les facteurs de confusion X et L_0.

Estimez deux modèles de régression logistique sur `D` : `mod1` sur les individus avec `A0 = 1`, `mod0` sur les individus avec `A0 = 0`.

```
#| context: interactive
## Modèle chez les exposés (A0 = 1)

## Modèle chez les non-exposés (A0 = 0)
```

```r
mod1 <- glm(D ~ X + L0,
            data   = df_base[df_base$A0 == 1, ],
            family = binomial)

mod0 <- glm(D ~ X + L0,
            data   = df_base[df_base$A0 == 0, ],
            family = binomial)

summary(mod1)
summary(mod0)
```

Étape 2 — Prédire les devenirs contrefactuels

Pour chaque individu (quelle que soit son exposition réelle), on prédit le risque de décès **comme s'il avait été exposé** (via `mod1`) et **comme s'il n'avait pas été exposé** (via `mod0`).

Calculez les vecteurs ***y1*** *et* ***y0*** *: probabilités de décès prédites sous* $a_0 = 1$ *et* $a_0 = 0$.

```r
#| context: interactive
## Prédictions contrefactuelles

## Probabilité de décès si tout le monde avait A0 = 1
y1 <- predict(mod1, newdata = df_base, type = "response")

## Probabilité de décès si tout le monde avait A0 = 0
y0 <- predict(mod0, newdata = df_base, type = "response")

head(data.frame(id = df_base$id, A0_obs = df_base$A0,
                pred_A0_1 = round(y1, 3),
                pred_A0_0 = round(y0, 3)))
```

Étape 3 — Calculer l'effet causal moyen (ATE)

Calculez $\widehat{E}(Y^1) = \overline{y_1}$, $\widehat{E}(Y^0) = \overline{y_0}$, *puis l'ATE et le RR.*

```r
#| context: interactive
## Moyennez les prédictions

E_Y1 <- mean(y1)
E_Y0 <- mean(y0)

cat("E(Y^1) =", round(E_Y1, 3), "\n")
cat("E(Y^0) =", round(E_Y0, 3), "\n")
cat("ATE    =", round(E_Y1 - E_Y0, 3), "\n")
cat("RR     =", round(E_Y1 / E_Y0, 3), "\n")
```

On obtient : $\widehat{E}(Y^1) \approx 0.125$, $\widehat{E}(Y^0) \approx 0.26$, $\widehat{ATE} \approx$ -0.135.

Interprétation

Que signifie cet ATE ? Comment le comparez-vous à l'analyse brute de la page précédente ?

Réponse

L'ATE estimé par G-computation est la différence de risque de décès entre les scénarios contrefactuels "tout le monde exposé" et "personne exposé", **après ajustement sur X et L_0**. Contrairement à l'analyse brute, cette estimation tient compte du fait que les individus exposés et non exposés ne sont pas comparables à l'inclusion.

Limites ici :

— On traite le décès comme binaire, en ignorant l'information temporelle.

— On n'ajuste que sur les confondeurs *initiaux* — les parties suivantes traitent les confondeurs dépendants du temps.

— Le résultat repose entièrement sur la bonne spécification du modèle de résultat D.

La Partie 2 répondra à la même **Question 1** en utilisant une approche différente (IPTW), ce qui permettra un contrôle croisé des deux estimations.

BONUS — Intervalle de confiance par bootstrap

Estimez un intervalle de confiance à 95% par bootstrap ($B = 200$ ré-échantillons).

```
#| context: interactive
set.seed(123)
B <- 200
ate_boot <- numeric(B)

for (b in 1:B) {
  ## 1. Ré-échantillonner df_base avec remise

  ## 2. Ajuster mod1 et mod0

  ## 3. Prédire y1, y0

  ## 4. Calculer l'ATE

}
quantile(ate_boot, c(0.025, 0.975))
```

```
set.seed(123)
B <- 200
ate_boot <- numeric(B)

for (b in 1:B) {
  idx  <- sample(nrow(df_base), replace = TRUE)
  df_b <- df_base[idx, ]
```

```r
  m1 <- glm(D ~ X + L0, data = df_b[df_b$A0 == 1, ], family = binomial)
  m0 <- glm(D ~ X + L0, data = df_b[df_b$A0 == 0, ], family = binomial)

  y1_b <- predict(m1, newdata = df_b, type = "response")
  y0_b <- predict(m0, newdata = df_b, type = "response")

  ate_boot[b] <- mean(y1_b) - mean(y0_b)
}

cat("ATE    :", round(mean(ate_boot), 3), "\n")
cat("IC 95% :", round(quantile(ate_boot, c(0.025, 0.975)), 3), "\n")
```

BONUS 2 — G-computation sur critère censuré

> **Note**
>
> Cette section est à faire **chez vous**, à votre rythme. Elle n'est pas traitée en TP. Elle montre comment étendre la G-computation vue précédemment pour traiter correctement le critère de jugement censuré.

Pourquoi changer d'approche ?

La G-computation de la partie principale modélise D comme un indicateur binaire (décédé avant $t = 3$ ans, oui/non). Deux limites importantes :

1. **Information temporelle perdue** : on sait *si* l'individu est décédé, mais on ignore *quand*.
2. **Censure traitée comme un non-événement** : un individu perdu de vue à $t = 1$ an est traité de la même façon qu'un individu suivi 3 ans sans événement.

La G-computation *complète* contourne ces deux problèmes en estimant directement des **courbes de survie contrefactuelles** :

$$\hat{S}^{a_0=1}(t) = P(T^{a_0=1} > t) \qquad \text{et} \qquad \hat{S}^{a_0=0}(t) = P(T^{a_0=0} > t)$$

Principe

L'idée est simple : on remplace les deux modèles logistiques par deux **modèles de Cox**, un chez les exposés et un chez les non-exposés.

Dans un modèle de Cox, la survie individuelle est :

$$\hat{S}_i^{a_0}(t) = \exp\Big(-\hat{H}_0^{a_0}(t) \times e^{\hat{\beta}^{a_0} \cdot (X_i,\, L_{0i})}\Big)$$

— $\hat{H}_0^{a_0}(t)$: hazard cumulatif de base (estimateur de Breslow), commun à tous les individus du groupe a_0

— $e^{\hat{\beta} \cdot (X_i, L_{0i})}$: **multiplicateur individuel** de risque (prédicteur linéaire exponentié), propre à chaque individu

La courbe de survie **marginale** (contrefactuelle) s'obtient en moyennant sur l'ensemble de la population :

$$\hat{S}^{a_0}(t) = \frac{1}{n} \sum_{i=1}^{n} \hat{S}_i^{a_0}(t)$$

Étape 1 — Deux modèles de Cox

> **Note**
>
> `df_base` doit avoir été créée lors de l'Étape 0 ci-dessus. Si nécessaire, relancez ce chunk d'abord.

Estimez deux modèles de Cox séparément chez les exposés ($A_0 = 1$) et chez les non-exposés ($A_0 = 0$), en contrôlant sur X et L_0. Utilisez `coxph(Surv(time, D) ~ X + L0, data = ...)`.

```
#| context: interactive
## Modèle de Cox chez les exposés (A0 = 1)

## Modèle de Cox chez les non-exposés (A0 = 0)

mod_cox1 <- coxph(Surv(time, D) ~ X + L0,
                  data = df_base[df_base$A0 == 1, ])
mod_cox0 <- coxph(Surv(time, D) ~ X + L0,
                  data = df_base[df_base$A0 == 0, ])

summary(mod_cox1)
summary(mod_cox0)
```

Étape 2 — Hazard cumulatif de base et prédicteurs linéaires individuels

L'estimation des courbes contrefactuelles nécessite deux ingrédients issus de chaque modèle de Cox :

- Le **hazard cumulatif de base** $\widehat{H}_0(t)$: c'est une fonction en escalier qui augmente à chaque temps d'événement. On l'obtient avec `basehaz(mod, centered = FALSE)`, qui renvoie un data frame avec les colonnes `time` et `hazard`.
- Le **prédicteur linéaire individuel** $\widehat{\text{lp}}_i = \widehat{\beta} \cdot (X_i, L_{0i})$: c'est un scalaire par individu, qui mesure à quel point cet individu a un risque plus ou moins élevé que la baseline. On l'obtient avec `predict(mod, newdata = df_base, type = "lp")`.

Extrayez le hazard cumulatif de base de chaque modèle, et calculez le prédicteur linéaire pour chaque individu de ***`df_base`*** *sous chaque scénario.*

```
#| context: interactive
## Hazard cumulatif de base de chaque modèle

## Prédicteur linéaire pour chaque individu (sous a0=1 et sous a0=0)
```

```
## Hazard cumulatif de base (estimateur de Breslow)
bh1 <- basehaz(mod_cox1, centered = FALSE)
bh0 <- basehaz(mod_cox0, centered = FALSE)

head(bh1)  # time = temps d'événement, hazard = H0(t) cumulé

## Prédicteur linéaire individuel lp_i = beta * (X_i, L0_i)
## Prédit pour chaque individu de df_base, quel que soit son A0 observé
lp1 <- predict(mod_cox1, newdata = df_base, type = "lp")
lp0 <- predict(mod_cox0, newdata = df_base, type = "lp")

head(data.frame(
  id         = df_base$id,
  A0_obs     = df_base$A0,
  lp_si_A1   = round(lp1, 3),
  lp_si_A0   = round(lp0, 3)
))
```

Étape 3 — Courbes de survie individuelles et marginalisation

On dispose maintenant de tout ce qu'il faut pour calculer $\widehat{S}_i^{a_0}(t)$ pour chaque individu i et chaque temps t, puis d'en faire la moyenne.

Astuce

`stepfun(knots, values)` crée une fonction en escalier à partir des nœuds et valeurs associées. Ici, `stepfun(bh1$time, c(0, bh1$hazard))` crée la fonction $\hat{H}_0^{(a_0=1)}(t)$: elle vaut 0 avant le premier événement, puis prend la valeur de Breslow correspondante à chaque nœud. On peut ensuite l'évaluer sur n'importe quelle grille de temps.
`outer(u, v)` calcule le produit extérieur de deux vecteurs : `outer(u, v)[i, j] = u[i] * v[j]`. Cela permet de calculer $-e^{\hat{\text{lp}}_i} \times \hat{H}_0(t_j)$ pour tous les couples (i, j) d'un coup, sans boucle, et d'obtenir directement la matrice des $n \times T$ valeurs $\log \hat{S}_i(t_j)$.

Calculez les courbes de survie marginales $\hat{S}^{a_0=1}(t)$ et $\hat{S}^{a_0=0}(t)$ sur une grille de temps.

```
#| context: interactive
## 1. Grille de temps

## 2. Hazard cumulatif de base évalué sur cette grille (stepfun)

## 3. Matrice des survies individuelles n × T (outer + exp)

## 4. Courbes marginales (colMeans)
```

```
## 1. Grille de temps : tous les temps d'événement de df_base
t_grid <- sort(unique(df_base$time))

## 2. H0(t) évalué sur t_grid via des fonctions en escalier
H1_fun  <- stepfun(bh1$time, c(0, bh1$hazard))
H0_fun  <- stepfun(bh0$time, c(0, bh0$hazard))
H1_grid <- H1_fun(t_grid)   # vecteur longueur T
H0_grid <- H0_fun(t_grid)

## 3. Survie individuelle : S_i(t_j) = exp(-H0(t_j) * exp(lp_i))
##    outer(-exp(lp), H_grid)[i, j] = -exp(lp[i]) * H_grid[j]
S1_mat <- exp(outer(-exp(lp1), H1_grid))  # matrice n × T
S0_mat <- exp(outer(-exp(lp0), H0_grid))

## 4. Courbes marginales : moyenne sur les individus (ligne = individu)
S1_marg <- colMeans(S1_mat)
S0_marg <- colMeans(S0_mat)

head(data.frame(t     = round(t_grid, 2),
                S_a1  = round(S1_marg, 3),
                S_a0  = round(S0_marg, 3)))
```

Étape 4 — Visualisation des courbes de survie contrefactuelles

Tracez les deux courbes de survie marginales contrefactuelles sur le même graphique.

```
#| context: interactive
## Courbes de survie contrefactuelles

plot(t_grid, S1_marg, type = "s", col = "blue", lwd = 2,
     ylim = c(0, 1),
     xlab = "Temps (années)",
     ylab = "Probabilité de survie",
     main = "G-computation (Cox) - Courbes contrefactuelles")
lines(t_grid, S0_marg, type = "s", col = "red", lwd = 2)
legend("bottomleft",
       c("Scénario a =1 (tous exposés)",
         "Scénario a =0 (aucun exposé)"),
       col = c("blue", "red"), lty = 1, lwd = 2, bty = "n")
```

Étape 5 — Différence de survie à $t = 3$ ans

Calculez $\hat{S}^{a_0=1}(3)$, $\hat{S}^{a_0=0}(3)$, et la différence de survie à 3 ans. Comparez avec l'ATE obtenu par G-computation logistique.

```
#| context: interactive
## Survie marginale à t = 3 sous chaque scénario, et différence

## Dernier temps <= 3 dans la grille
idx3 <- max(which(t_grid <= 3))

S1_3 <- S1_marg[idx3]
S0_3 <- S0_marg[idx3]

cat("S^(a0=1)(3) =", round(S1_3, 3), "\n")
cat("S^(a0=0)(3) =", round(S0_3, 3), "\n")
cat("Différence de survie à 3 ans :", round(S1_3 - S0_3, 3), "\n")
cat("Rapport de survie à 3 ans    :", round(S1_3 / S0_3, 3), "\n")
```

Discussion — comparaison avec l'approche binaire

Les deux G-computations (logistique et Cox) ciblent en principe le même estimand : $P(T^{a_0} \leq 3)$, ce qui correspond à $1 - S^{a_0}(3)$. En pratique, la différence de survie à $t = 3$ obtenue par les deux méthodes est souvent proche, mais peut légèrement différer car :

- la version Cox utilise **toute l'information temporelle** (pas seulement le statut binaire à 3 ans) ;
- la version Cox traite **correctement la censure** : un individu censuré à $t = 1$ an ne contribue qu'à l'estimation jusqu'à son temps de censure, sans être compté comme

"survivant" ou "décédé" à 3 ans ;
— les deux modèles imposent des hypothèses fonctionnelles différentes (effets proportionnels des hazards pour Cox, effets log-linéaires sur la cote pour la logistique).

La version Cox est donc plus rigoureuse et constitue l'approche standard pour la G-computation avec un critère de survie.

Intervalle de confiance par bootstrap (modèle de Cox)

Estimez un intervalle de confiance à 95% par bootstrap ($B = 200$ ré-échantillons) pour la différence de survie à $t = 3$ ans.

```
#| context: interactive
set.seed(42)
B <- 200
S1_boot <- numeric(B)
S0_boot <- numeric(B)

for (b in 1:B) {
  ## 1. Ré-échantillonner df_base avec remise

  ## 2. Ajuster mod_cox1 et mod_cox0

  ## 3. Extraire basehaz et prédicteurs linéaires

  ## 4. Calculer S^(a0=1)(3) et S^(a0=0)(3) et les stocker
  S1_boot[b] <- ...
  S0_boot[b] <- ...
}

diff_boot <- S1_boot - S0_boot
cat("IC 95% :", round(quantile(diff_boot, c(0.025, 0.975)), 3), "\n")
```

```
set.seed(42)
B <- 200
S1_boot <- numeric(B)
S0_boot <- numeric(B)

for (b in 1:B) {
  ## 1. Ré-échantillonnage avec remise
  idx <- sample(nrow(df_base), replace = TRUE)
  db  <- df_base[idx, ]

  ## 2. Modèles de Cox
  m1 <- coxph(Surv(time, D) ~ X + L0, data = db[db$A0 == 1, ])
```

```r
  m0 <- coxph(Surv(time, D) ~ X + L0, data = db[db$A0 == 0, ])

  ## 3. Basehaz et prédicteurs linéaires
  bh1b <- basehaz(m1, centered = FALSE)
  bh0b <- basehaz(m0, centered = FALSE)
  lp1b <- predict(m1, newdata = db, type = "lp")
  lp0b <- predict(m0, newdata = db, type = "lp")

  ## 4. H0 au temps 3 (dernière valeur <= 3 dans la baseline hazard)
  H1_3b <- if (any(bh1b$time <= 3)) tail(bh1b$hazard[bh1b$time <= 3], 1) else 0
  H0_3b <- if (any(bh0b$time <= 3)) tail(bh0b$hazard[bh0b$time <= 3], 1) else 0

  S1_boot[b] <- mean(exp(-H1_3b * exp(lp1b)))
  S0_boot[b] <- mean(exp(-H0_3b * exp(lp0b)))
}

diff_boot <- S1_boot - S0_boot
cat("Différence de survie à 3 ans (G-computation Cox, bootstrap) :\n")
cat("  Estimée :", round(mean(diff_boot), 3), "\n")
cat("  IC 95%  :", round(quantile(diff_boot, c(0.025, 0.975)), 3), "\n")
```

IPTW

```
#| context: setup
if (Sys.info()["nodename"] == "emscripten") {
  base_url <- sub("/[^/]*$", "/", quarto_page_url)
  download.file(
    paste0(base_url, "df.csv"),
    "df.csv", quiet = TRUE
  )
}
library(survival)
library(dplyr)
library(cobalt)
df <- read.csv("df.csv")
df <- df |>
  group_by(id) |>
  mutate(A0 = first(A), L0 = first(L)) |>
  ungroup()
```

Objectif — Question 1 (suite)

Cette partie répond à la **même Question 1** que la G-computation : estimer les courbes de survie contrefactuelles $S^{a_0=1}(t)$ et $S^{a_0=0}(t)$ ajustées sur les facteurs de confusion X et L_0.

L'**IPTW** adopte une stratégie opposée à la G-computation : au lieu de modéliser le résultat Y, il modélise l'**exposition** A_0 conditionnellement aux covariables, puis pondère les individus pour créer une pseudo-population équilibrée. Une fois les poids calculés, l'analyse de survie est directement réalisée par **Kaplan-Meier pondéré**, ce qui traite correctement la censure — aucune simplification du critère de jugement n'est nécessaire.

> **Note**
>
> **Lien avec la Partie 1 :** la G-computation avait traité D comme une variable binaire (décédé avant 3 ans, oui/non) pour simplifier. Avec l'IPTW, l'implémentation est tout aussi simple avec le critère censuré. Des **notes repliables** proposent le code et les résultats équivalents en version binaire, pour comparer directement les deux méthodes à la fin.

La variable `df` est disponible dans les chunks interactifs, avec `A0` et `L0` déjà propagées.

Étape 1 — Estimer le score de propension

Le score de propension est $e_i = P(A_0 = 1 \mid X = x_i,\ L_0 = l_i)$, estimé par régression logistique **sur les données de la première visite seulement**.

Estimez le modèle de propension ***mod.ps*** *et stockez les probabilités prédites dans* ***df$ps****.*

```
#| context: interactive
## Modèle logistique sur les données de la 1ère visite (T.start == 0)

## Prédire ps pour toutes les lignes de df
```

```
## Modèle de propension (données baseline uniquement)
mod.ps <- glm(A0 ~ X + L0,
              data   = df[df$T.start == 0, ],
              family = "binomial")

## Score de propension prédit pour tous les individus (toutes lignes)
df$ps <- predict(mod.ps, newdata = df, type = "response")

## Distribution du PS par groupe
summary(df$ps[df$A0 == 1])
summary(df$ps[df$A0 == 0])
```

Visualisez la distribution du score de propension selon A_0 pour vérifier le chevauchement (positivité) :

```
#| context: interactive
## Distribution du PS par groupe
```

```
## Histogrammes superposés
par(mfrow = c(1, 2))
hist(df$ps[df$A0 == 1], breaks = 20, col = "#AC182E80",
     main = "A0 = 1", xlab = "Score de propension", xlim = c(0, 1))
hist(df$ps[df$A0 == 0], breaks = 20, col = "#1D276980",
     main = "A0 = 0", xlab = "Score de propension", xlim = c(0, 1))
par(mfrow = c(1, 1))
```

Étape 2 — Calculer les poids IPTW

On calcule les poids **non stabilisés** $w_i = A_0/e_i + (1 - A_0)/(1 - e_i)$ et les poids **stabilisés** $w_i^s = P(A_0)/e_i$ si $A_0 = 1$, $P(A_0 = 0)/(1 - e_i)$ si $A_0 = 0$.

Calculez ***df$iptw*** *(non stabilisé) et* ***df$iptw.s*** *(stabilisé). Comparez leur distribution.*

```
#| context: interactive
## Poids non stabilisés

## Poids stabilisés (numérateur = P(A0) marginal)
```

```
## Distribution des poids

## Poids non stabilisés
df$iptw <- (df$A0 == 1) / df$ps + (df$A0 == 0) / (1 - df$ps)

## Probabilité marginale d'être exposé (numérateur des poids stabilisés)
p.A1 <- mean(df$A0[df$T.start == 0])
p.A0 <- 1 - p.A1

## Poids stabilisés
df$iptw.s <- ifelse(df$A0 == 1,
                    p.A1 / df$ps,
                    p.A0 / (1 - df$ps))

## Comparaison
cat("Poids non stabilisés - moyenne:", round(mean(df$iptw), 3),
    " / max:", round(max(df$iptw), 2), "\n")
cat("Poids stabilisés     - moyenne:", round(mean(df$iptw.s), 3),
    " / max:", round(max(df$iptw.s), 2), "\n")
```

> **Note**
>
> **Règle pratique :** les poids stabilisés ont une moyenne proche de 1 et une variance plus faible. Ils sont généralement préférés en pratique. Si certains poids sont très élevés (> 10–20), c'est un signe de violation de la positivité ou d'un modèle de propension mal spécifié.

Étape 3 — Vérifier l'équilibre

Avant toute analyse, il faut s'assurer que la pondération a **rétabli l'équilibre** entre les groupes $A_0 = 1$ et $A_0 = 0$ sur les covariables X et L_0.

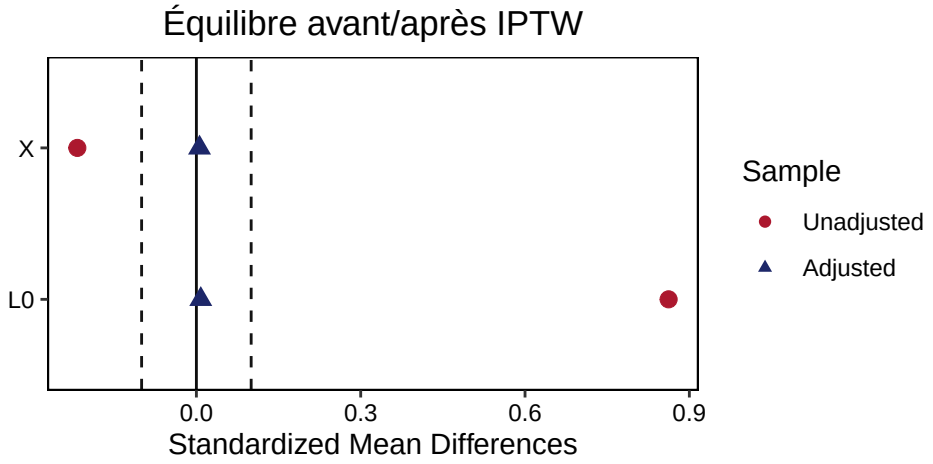
Utilisez `cobalt::bal.tab()` et `cobalt::love.plot()` pour comparer les différences standardisées avant et après pondération.

```
#| context: interactive
library(cobalt)

## Vérification de l'équilibre (données de la 1ère visite)

library(cobalt)

## Tableau des différences standardisées (avant et après pondération)
bal <- bal.tab(A0 ~ X + L0,
               data    = df[df$T.start == 0, ],
               weights = df$iptw.s[df$T.start == 0],
               method  = "weighting",
```

```
                binary  = "std",
                un      = TRUE)
bal

## Love plot avant/après
love.plot(bal,
          thresholds = c(m = 0.1),
          colors     = c("#AC182E", "#1D2769"),
          shapes     = c("circle", "triangle"),
          title      = "Équilibre avant/après IPTW")
```

Les différences standardisées après pondération sont-elles inférieures à 0,1 (seuil conventionnel) ?

Réponse



Si les SMD après pondération sont $< 0{,}1$ pour toutes les covariables, l'équilibre est satisfaisant. Un SMD résiduel $> 0{,}1$ suggère une mauvaise spécification du modèle de propension (variables manquantes, transformations nécessaires, interactions, etc.).
Rappel : n'utilisez **pas** de p-values pour évaluer l'équilibre — elles dépendent de la taille d'échantillon et ne mesurent pas la comparabilité des groupes.

Étape 4 — Analyse de survie pondérée (Kaplan-Meier)

Estimez les courbes de Kaplan-Meier pondérées par ***iptw.s****, tracez-les et obtenez la différence de survie à 3 ans.*

```
#| context: interactive
## Kaplan-Meier pondéré

km.iptw <- survfit(Surv(T.start, T.stop, D) ~ A0,
                   data    = df,
                   weights = iptw.s)

plot(km.iptw,
```

```
     col  = c("#1D2769", "#AC182E"), lwd = 2, conf.int = FALSE,
     xlab = "Temps (années)", ylab = "Probabilité de survie",
     main = "Kaplan-Meier pondéré (IPTW stabilisé)")
legend("bottomleft",
       legend = c("A0 = 0 (non-exposés)", "A0 = 1 (exposés)"),
       col = c("#1D2769", "#AC182E"), lwd = 2, bty = "n")

## Différence de survie à 3 ans
s3 <- summary(km.iptw, times = 3)$surv
cat("Survie à 3 ans - A0=0 :", round(s3[1], 3), "\n")
cat("Survie à 3 ans - A0=1 :", round(s3[2], 3), "\n")
cat("Différence     :", round(diff(s3), 3), "\n")
```

La différence de survie à 3 ans après IPTW est 0.167.

> **Note** — Version avec critère binaire (équivalent G-computation)
>
> Comme dans la Partie 1, on peut ignorer l'information temporelle et traiter D comme un indicateur binaire de décès à 3 ans. L'IPTW avec critère binaire revient à calculer la **moyenne pondérée de** D dans chaque groupe sur une base à une ligne par individu :
>
> $$\widehat{E}(D^{a_0}) = \frac{\sum\limits_{i\,:\,A_{0i}=a_0} w_i^s \cdot D_i}{\sum\limits_{i\,:\,A_{0i}=a_0} w_i^s}$$

```
## Une ligne par individu - D final et poids IPTW stabilisés
df_base_bin <- df |>
  group_by(id) |>
  summarise(A0 = first(A), D = last(D), iptw.s = first(iptw.s)) |>
  ungroup()

m1_bin <- weighted.mean(df_base_bin$D[df_base_bin$A0 == 1],
                        df_base_bin$iptw.s[df_base_bin$A0 == 1])
m0_bin <- weighted.mean(df_base_bin$D[df_base_bin$A0 == 0],
                        df_base_bin$iptw.s[df_base_bin$A0 == 0])

cat("E(D^1) =", round(m1_bin, 3), "\n")
cat("E(D^0) =", round(m0_bin, 3), "\n")
cat("ATE binaire (IPTW) =", round(m1_bin - m0_bin, 3))
```

> L'ATE binaire par IPTW est ≈ -0.15 — à comparer avec l'ATE G-computation de la Partie 1 dans la section suivante.

Comparaison G-computation vs IPTW

Les deux méthodes spécifient des modèles distincts pour répondre à la même question causale :

	G-computation (Partie 1)	IPTW (cette partie)
Modèle estimé	Résultat : $E(D \mid A_0, X, L_0)$	Exposition : $P(A_0 = 1 \mid X, L_0)$
Critère utilisé	Binaire *(simplification)*	Survie censurée *(complet)*
Mesure d'effet	Différence de risque de décès	Différence de survie à 3 ans
Estimée	-0.135	0.167

Lien entre les deux mesures : puisque $P(T \leq 3) = 1 - S(3)$, la différence de risque de décès vaut − la différence de survie à 3 ans. Les deux estimations doivent donc être de signe opposé et de valeur absolue proche.

Vérifiez cela : à partir de ***`df_base_bin`*** *(une ligne par individu, avec* `D` *final et* ***`iptw.s`****), calculez l'ATE par IPTW avec critère binaire, et vérifiez sa cohérence avec la différence de survie du KM pondéré.*

> **Avertissement**
>
> `km.iptw` doit avoir été calculé à l'Étape 4 pour que ce chunk fonctionne.

```
#| context: interactive
## Données : une ligne par individu (D final + poids IPTW)
df_base_bin <- df |>
  group_by(id) |>
  summarise(A0 = first(A), D = last(D), iptw.s = first(iptw.s)) |>
  ungroup()

## Calculez m1_bin, m0_bin, et l'ATE binaire

## Comparez avec la différence de survie du KM pondéré
s3 <- summary(km.iptw, times = 3)$surv
```

```
df_base_bin <- df |>
  group_by(id) |>
  summarise(A0 = first(A), D = last(D), iptw.s = first(iptw.s)) |>
  ungroup()

m1_bin <- weighted.mean(df_base_bin$D[df_base_bin$A0 == 1],
                        df_base_bin$iptw.s[df_base_bin$A0 == 1])
m0_bin <- weighted.mean(df_base_bin$D[df_base_bin$A0 == 0],
                        df_base_bin$iptw.s[df_base_bin$A0 == 0])
ate_iptw_bin <- m1_bin - m0_bin

cat("=== IPTW (critère binaire) ===\n")
cat("E(D^1)      =", round(m1_bin, 3), "\n")
cat("E(D^0)      =", round(m0_bin, 3), "\n")
cat("ATE binaire =", round(ate_iptw_bin, 3), "\n\n")

s3 <- summary(km.iptw, times = 3)$surv
```

```r
cat("=== IPTW (KM pondéré, critère censuré) ===\n")
cat("S^(a0=1)(3) =", round(s3[2], 3), "\n")
cat("S^(a0=0)(3) =", round(s3[1], 3), "\n")
cat("Diff survie =", round(diff(s3), 3), "\n\n")

cat("=== Vérification du lien ATE  -diff(survie) ===\n")
cat("ATE + diff(survie) =", round(ate_iptw_bin + diff(s3), 3),
    "  (doit être proche de 0)\n")
```

À retenir

Les estimations de référence calculées automatiquement :

- **G-computation** (binaire, Partie 1) : ATE ≈ -0.135
- **IPTW** (binaire) : ATE ≈ -0.15
- **IPTW** (KM pondéré, survie) : diff. de survie ≈ 0.167

Convergence : si G-computation et IPTW binaire donnent des ATE proches, les deux modèles (résultat et exposition) sont probablement bien spécifiés — c'est un argument de robustesse.
Signe opposé : la différence de survie IPTW doit être approximativement −ATE. Ici : -0.15 + 0.167 = 0.016 (proche de 0).
Critère censuré vs binaire : la différence de survie par KM pondéré est l'estimateur de référence de cette partie, car il exploite toute l'information temporelle et traite correctement la censure. La version binaire est proposée uniquement pour le parallèle avec la Partie 1.

La Partie 3 passe à la **Question 2** : l'effet de l'exposition *maintenue* tout au long du suivi.

IPCW

```
#| context: setup
if (Sys.info()["nodename"] == "emscripten") {
  base_url <- sub("/[^/]*$", "/", quarto_page_url)
  download.file(
    paste0(base_url, "df.csv"),
    "df.csv", quiet = TRUE
  )
}
library(survival)
library(dplyr)
df <- read.csv("df.csv")
df <- df |>
  group_by(id) |>
  mutate(A0 = first(A), L0 = first(L)) |>
  ungroup()

## Poids IPTW stabilisés (recalculés depuis partie 2)
mod.ps  <- glm(A0 ~ X + L0, data = df[df$T.start == 0, ], family = "binomial")
df$ps   <- predict(mod.ps, newdata = df, type = "response")
p.A1    <- mean(df$A0[df$T.start == 0])
df$iptw.s <- ifelse(df$A0 == 1, p.A1 / df$ps, (1 - p.A1) / (1 - df$ps))
```

Objectif — Question 2

Les Parties 1 et 2 ont estimé l'effet de l'*initiation* de l'exposition A_0, sans se préoccuper de ce qui se passe ensuite (les patients peuvent arrêter ou débuter le traitement par la suite). C'est l'**analogue-ITT**.

Cette partie répond à une **nouvelle question** : quel serait l'effet si les patients avaient *maintenu* leur stratégie d'exposition tout au long du suivi ?

$$S^{\bar{a}=1}(t) = P(T^{\bar{a}=1} > t) \quad \text{et} \quad S^{\bar{a}=0}(t) = P(T^{\bar{a}=0} > t)$$

Stratégie : grouper les individus selon A_0, puis **censurer artificiellement** tout individu qui dévie de sa stratégie initiale. Cette censure est très probablement **informative** (les patients qui changent de traitement diffèrent des autres) → on corrige par **IPCW**. Les poids IPTW de la Partie 2 sont conservés pour l'équilibre initial.

Note sur la session R

Les variables suivantes sont disponibles dans vos chunks interactifs (reconstruites dans le contexte de setup) :

- `df` : base avec `A0`, `L0` propagés
- `df$ps` : score de propension $P(A_0 = 1 \mid X, L_0)$
- `df$iptw.s` : poids IPTW stabilisés (de la Partie 2)

Étape 1 — Définir les deux groupes de stratégie

Créez `df.1` (individus avec `A0 == 1`) et `df.0` (individus avec `A0 == 0`).

```
#| context: interactive
## Groupe stratégie "toujours exposé"

## Groupe stratégie "jamais exposé"

df.1 <- df[df$A0 == 1, ]
df.0 <- df[df$A0 == 0, ]

cat("Individus avec A0=1 :", length(unique(df.1$id)), "\n")
cat("Individus avec A0=0 :", length(unique(df.0$id)), "\n")
```

Étape 2 — Censure artificielle dans le groupe A0 = 1

Pour la stratégie $\bar{a} = 1$, un individu **dévie** dès qu'il arrête le traitement ($A = 0$ à une visite ultérieure).

Dans `df.1`, créez `switchA` : 0 tant que l'individu suit la stratégie, 1 à la première déviation. Conservez toutes les lignes jusqu'à `switchA  1` (on garde la ligne de déviation pour estimer le modèle de poids).

```
#| context: interactive
## Somme cumulée de A=1 au fil du temps

## switchA = 0 si A=1 à toutes les visites jusqu'ici, 1 sinon (puis cumprod)

## Garder switchA <= 1

df.1 <- df.1 |>
  group_by(id) |>
  mutate(
    cumsumA = cumsum(A == 1),
    ## Si A=1 à toutes les visites, cumsumA == T.start + 1
    switchA = if_else(cumsumA == T.start + 1, 0L, 1L),
    switchA = cumsum(switchA)
```

```
  ) |>
  filter(switchA <= 1) |>
  ungroup()

table(df.1$switchA)
```

Étape 3 — Modèle de déviation et poids IPCW (groupe A0 = 1)

On estime la **probabilité de ne pas dévier** à chaque visite par régression logistique poolée (*pooled logistic regression*), puis on calcule les poids IPCW :

$$W_i(t) = \prod_{k=0}^{t} \frac{1}{P(\text{switch}_{ik} = 0 \mid X_i, L_{ik})} \qquad t = 0, 1, 2$$

Ajustez `wt.mod.1 <- glm(switchA ~ as.factor(T.start) + X + L, ...)` *sur* `df.1`*, stockez* `wt.denom` $= P(switch = 0)$ *prédit. Supprimez les lignes* `switchA == 1`*. Calculez* `wt` $=$ *produit cumulé de* `1/wt.denom`*.*

```
#| context: interactive
## Modèle de déviation (régression logistique poolée)

## Probabilité de ne PAS dévier

## Supprimer les lignes de déviation (switchA == 1)

## Poids IPCW = produit cumulé de 1/P(no switch)

## Modèle poolé de déviation dans df.1
wt.mod.1 <- glm(switchA ~ as.factor(T.start) + X + L,
                family = "binomial", data = df.1)

## Probabilité de ne pas dévier
df.1$wt.denom <- 1 - predict(wt.mod.1, type = "response", newdata = df.1)

## Supprimer la ligne de déviation
df.1 <- df.1[df.1$switchA == 0, ]

## Poids IPCW : produit cumulé de 1/P(no switch)
df.1 <- df.1 |>
  group_by(id) |>
  mutate(wt = cumprod(1 / wt.denom)) |>
  ungroup()

summary(df.1$wt)
```

Étape 4 — Répéter pour le groupe A0 = 0

Reproduire les étapes 2 et 3 pour ***df.0*** *(déviation = passer de $A = 0$ à $A = 1$). Empiler ensuite* ***df.1*** *et* ***df.0*** *dans* ***dfpp****.*

```
#| context: interactive
## Censure artificielle dans df.0 (déviation = A passe à 1)

## Modèle de déviation, wt.denom, suppression switchA==1, poids wt

## Empilement
## dfpp <- rbind(df.1, df.0)
```

```
## Censure artificielle dans df.0
df.0 <- df.0 |>
  group_by(id) |>
  mutate(
    cumsumA = cumsum(A == 0),
    switchA = if_else(cumsumA == T.start + 1, 0L, 1L),
    switchA = cumsum(switchA)
  ) |>
  filter(switchA <= 1) |>
  ungroup()

## Modèle poolé de déviation dans df.0
wt.mod.0      <- glm(switchA ~ as.factor(T.start) + X + L,
                     family = "binomial", data = df.0)
df.0$wt.denom <- 1 - predict(wt.mod.0, type = "response", newdata = df.0)
df.0          <- df.0[df.0$switchA == 0, ]
df.0          <- df.0 |>
  group_by(id) |>
  mutate(wt = cumprod(1 / wt.denom)) |>
  ungroup()

## Empilement
dfpp <- rbind(df.1, df.0)
cat("Lignes dans dfpp :", nrow(dfpp), "\n")
```

Étape 5 — Poids combinés IPTW × IPCW

Les poids IPCW (`wt`) corrigent la censure artificielle informative. Mais ils ne rééquilibrent pas encore les caractéristiques initiales entre $A_0 = 1$ et $A_0 = 0$. Il faut **combiner** avec les poids IPTW estimés en Partie 2.

Calculez ***dfpp$comb.wt = dfpp$iptw.s * dfpp$wt****, puis vérifiez la distribution.*

```
#| context: interactive
## Poids combinés

## Distribution

dfpp$comb.wt <- dfpp$iptw.s * dfpp$wt

cat("Poids IPCW seuls  - moy:", round(mean(dfpp$wt), 3),
    " / max:", round(max(dfpp$wt), 2), "\n")
cat("Poids combinés    - moy:", round(mean(dfpp$comb.wt), 3),
    " / max:", round(max(dfpp$comb.wt), 2), "\n")
```

> **Note**
>
> **Pourquoi multiplier ?** Les poids IPTW rééquilibrent les groupes sur les caractéristiques initiales (X, L_0). Les poids IPCW compensent la censure artificielle informative au cours du suivi (L_t, A_t). Les deux sources de biais sont indépendantes, donc leurs corrections se multiplient.

Étape 6 — Analyse de survie per-protocol

Tracez le Kaplan-Meier pondéré par `comb.wt` *dans* `dfpp`*. Comparez avec l'analogue-ITT de la Partie 2.*

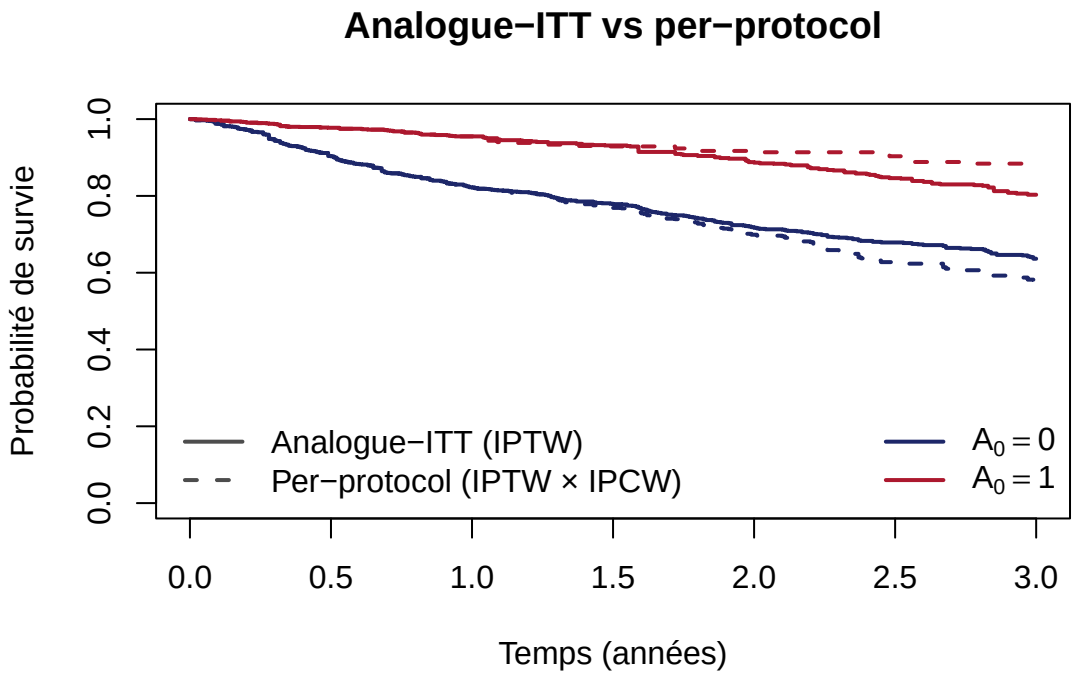
```
#| context: interactive
## Kaplan-Meier per-protocol (IPTW × IPCW)

km.pp <- survfit(Surv(T.start, T.stop, D) ~ A0,
                 data    = dfpp,
                 weights = comb.wt)

plot(km.pp,
     col  = c("#1D2769", "#AC182E"), lwd = 2, conf.int = FALSE,
     xlab = "Temps (années)", ylab = "Probabilité de survie",
     main = "Kaplan-Meier per-protocol (IPTW × IPCW)")
legend("bottomleft",
       legend = c("Stratégie ā=0 (jamais exposé)",
                  "Stratégie ā=1 (toujours exposé)"),
       col = c("#1D2769", "#AC182E"), lwd = 2, bty = "n")

## Différence de survie à 3 ans
s3.pp <- summary(km.pp, times = 3)$surv
cat("Survie à 3 ans - ā=0 :", round(s3.pp[1], 3), "\n")
cat("Survie à 3 ans - ā=1 :", round(s3.pp[2], 3), "\n")
cat("Différence     :", round(diff(s3.pp), 3), "\n")
```

La différence de survie à 3 ans (per-protocol) est 0.316.

Interprétation

Comparez l'effet per-protocol à l'effet analogue-ITT. Pourquoi peut-il différer ?

Réponse



— L'**analogue-ITT** estime l'effet d'**initier** l'exposition A_0, quelle que soit la compliance ultérieure. Il dilue l'effet si certains exposés arrêtent leur traitement.

— L'**analogue per-protocol** estime l'effet de **maintenir** l'exposition tout au long du suivi. Il peut être plus fort (si l'exposition continue est nécessaire) ou différent pour d'autres raisons.

Si les deux estimations convergent, l'arrêt du traitement n'a pas d'impact majeur sur la survie. Si elles divergent, la compliance joue un rôle.

Hypothèses supplémentaires nécessaires pour l'analogue-PP :

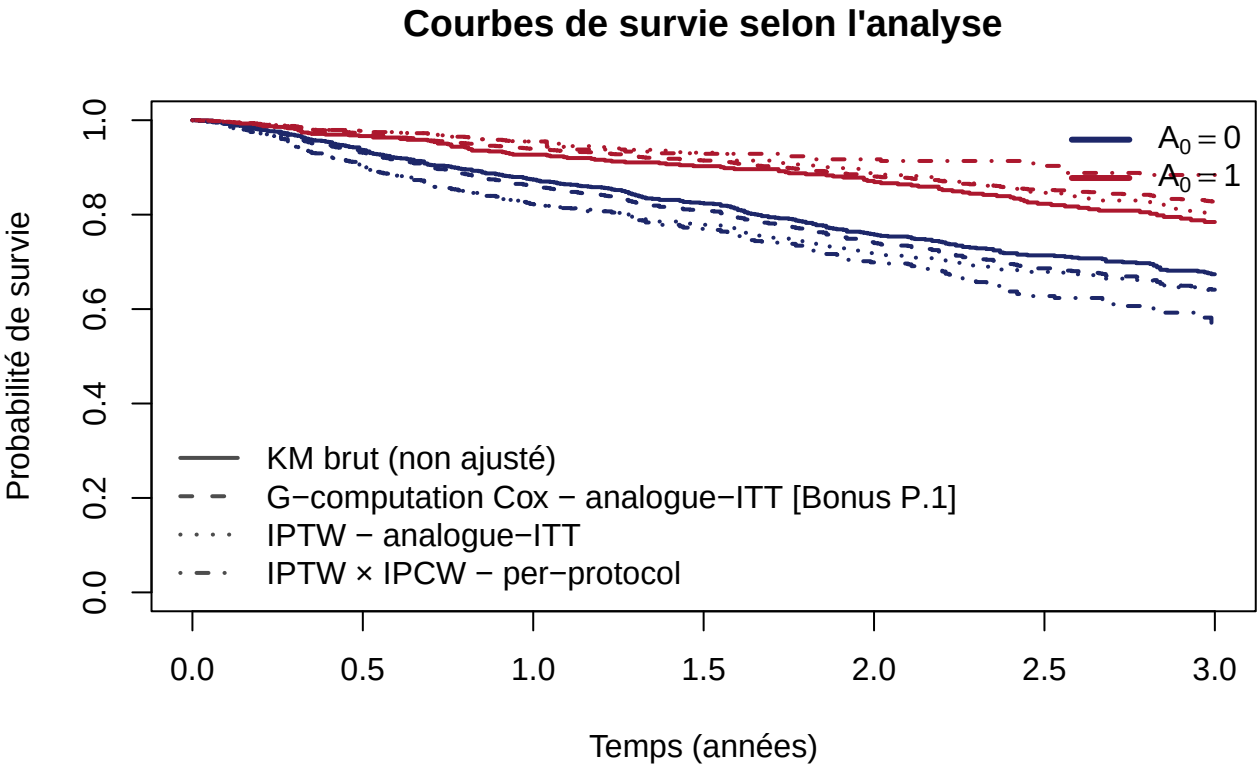
— Échangeabilité conditionnelle **au cours du temps** : L_t capture tous les facteurs qui prédisent à la fois la déviation et le décès.

— Positivité au cours du temps : à chaque visite, chaque individu doit avoir une probabilité positive de continuer ou d'arrêter.

Conclusion

```
Warning in survfit.coxph(fit, se.fit = FALSE): the model contains interactions;
the default curve based on columm means of the X matrix is almost certainly not
useful. Consider adding a newdata argument.
```

Comparaison graphique



Récapitulatif des estimations

Analyse	Estimand	Ajustement	S(3) — A =0	S(3) — A =1	Δ sur-vie
KM brut (non ajusté)	—	Aucun	0.674	0.784	0.111

Analyse	Estimand	Ajustement	S(3) — A =0	S(3) — A =1	Δ sur- vie
Q1 — G-computation Cox [Bonus P.1]	Analogue-ITT : effet d'initier l'exposition	Confusion initiale (X, L) — modèle du résultat	0.641	0.828	0.187
Q1 — IPTW	Analogue-ITT : effet d'initier l'exposition	Confusion initiale (X, L) — modèle de l'exposition	0.636	0.803	0.167
Q2 — IPTW × IPCW (per-protocol)	Analogue per-protocol : effet de maintenir l'exposition	Confusion initiale + déviation de stratégie	0.568	0.884	0.316

Discussion

Ce que nous apprenons de chaque analyse

— **Analyse brute :** la différence de survie observée entre $A_0 = 1$ et $A_0 = 0$ n'est pas interprétable causalement. Les groupes ne sont pas comparables à l'inclusion (X, L_0 diffèrent).

— **Analogue-ITT (G-computation Cox et IPTW — Q1) :** après ajustement sur les caractéristiques initiales, ces deux méthodes estiment le même estimand : l'effet d'*initier* l'exposition au temps 0, quelle que soit la compliance ultérieure. C'est l'estimand le plus proche d'un essai clinique en intention de traiter. La G-computation ajuste via un modèle du résultat ; l'IPTW via un modèle de l'exposition — si les deux convergent, la confiance dans le résultat est renforcée.

— **Analogue per-protocol (IPTW × IPCW — Q2) :** en censurant artificiellement les individus qui dévient de leur stratégie et en corrigeant ce biais par IPCW, on estime l'effet d'*initier et de maintenir* l'exposition. Cet estimand est plus proche d'un essai per-protocol.