Mediation Workshop

Benoît Lepage

2024-05-03

Contents

1 Introduction **5**

2 Software **7**

3 Data sets **9**

3.1 General presentation of the data used in our examples 9

3.2 Data generating mechanisms 10

4 Baron and Kenny, structural equation models **13**

4.1 Baron and Kenny approach 13

5 Traditional regression models **19**

5.1 Estimation of the Average Total Effect (ATE) 19

5.2 Two-way decomposition 22

5.3 Three-way decomposition 29

5.4 Four-way decomposition 33

6 G-computation **41**

6.1 Estimation of the Average Total Effect (ATE) 41

6.2 Estimation of Controlled Direct Effects (CDE) 43

6.3 Estimation of Natural Direct (NDE) and Indirect Effects (NIE) . 50

6.4 Estimation of "Marginal" Randomized/Interventional Natural Direct (NDE) and Indirect Effects (NIE) 50

6.5 Using the `CMAverse` package for 2-way, 3-way and 4-way decomposition 58

7 Inverse Probability of Treatment Weighting (IPTW) 63

7.1 Estimation of the Average total effect 63

7.2 Estimation of the Controlled direct effect (CDE) 65

8 Marginal structural models 69

8.1 MSM for the Average Total Effect (ATE) 69

8.2 MSM for Controlled Direct Effects 74

8.3 MSM for Natural Direct and Indirect Effects 82

9 Targeted Maximum Likelihood Estimation (TMLE) 87

9.1 TMLE for the ATE 88

9.2 TMLE of the Controlled direct effect (CDE) 96

10 Appendix A: Data generating mechanisms 101

10.1 First causal model: Data generating mechanism without mediator-outcome confounder affected by the exposure 101

10.2 Second causal model: Data generating mechanism with mediator-outcome confounder affected by the exposure 103

10.3 Simulation of the four data sets used in examples 105

11 Appendix B: Calculation of the true causal quantities 111

11.1 True causal quantities without mediator-ouctome confounder affected by the exposure 111

11.2 True causal quantities with mediator-ouctome confounder affected by the exposure 151

Chapter 1

Introduction

The objective of this document is to provide practical examples for the Expanse report *"Mediation Analysis: a Starting Guide for Epidemiologists"* with R scripts corresponding to the different estimation methods presented in the report.

Chapter 2

Software

The examples given in this workshop have been elaborated for R (version 4.2.2).

Depending on the estimator, some R packages might be necessary:

- COMPLETE ONCE ALL THE EXAMPLES ARE DONE

Chapter 3

Data sets

3.1 General presentation of the data used in our examples

Four data sets have been simulated, each containing 7 variables:

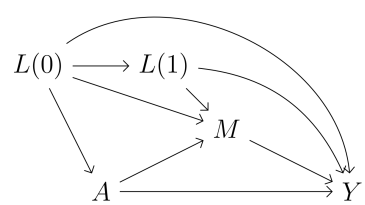
- 2 baseline confounders (denoted $L(0)$ in the DAGs):
  - `L0_male`, a binary variable indicating the sex of the participant (1 for men, 0 for women);
  - `L0_parent_low_educ_lv`, a binary variable indicated if the parents of the participants had a low level of education (1 for a low educational level, 0 for a high educational level);
- 1 exposure of interest (denoted A in the DAGs):
  - `A0_ace`, a binary variable indicating if the participants had been exposed to a high level of "Adverse childhood experience";
- 1 confounder of the mediator-outcome relationship (denoted $L(1)$ in the DAGs):
  - `L1`, a binary variable indicating if the participant has a low educational level (1 for a low educational level, 0 for a high educational level);
- 1 mediator of interest (denoted M in the DAGs):
  - `M_smoking`, a binary variable indicating if the participant is a smoker (1 for smokers, 0 for non-smokers);
- 2 outcomes (denoted Y in the DAGs):

  - `Y2_death`, a binary variable indicating the occurrence of death before 60 years of age (1 if dead, 0 if alive);
  - `Y2_qol`, a quantitative variable corresponding to a quality of life measurement.

3.2 Data generating mechanisms

The 4 data generating mechanisms used to simulate the data sets are described in chapter 4 of the *Expanse "Mediation analysis" report*:

- The first two data sets are simulated from a causal model where confounders of the mediator-outcome relationship ($L(1)$) are not affected by the exposure A (Figure 3.1),
  - The data set `df1.csv` is simulated from the statistical model $\mathcal{M}_1$, which does not contain any $A * M$ interaction effect on the outcome Y.
  - The data set `df1_int.csv` is simulated from the statistical model $\mathcal{M}_{1*}$, which contains an $A * M$ interaction effect on the outcome Y.

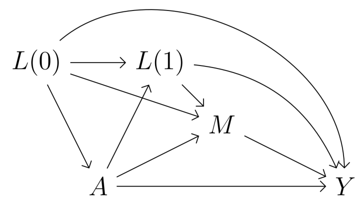


DAG of models $\mathcal{M}_1$ and $\mathcal{M}_{1*}$

Figure 3.1: Causal model 1

- The next two data sets are simulated from a causal model where confounders of the mediator-outcome relationship ($L(1)$) are affected by the exposure A (Figure 3.2),
  - The data set `df2.csv` is simulated from the statistical model $\mathcal{M}_2$, which does not contain any $A * M$ interaction effect on the outcome Y.
  - The data set `df2_int.csv` is simulated from the statistical model $\mathcal{M}_{2*}$, which contains an $A * M$ interaction effect on the outcome Y.

The R functions used to simulate these 4 data sets are given in the Appendix A @ref(appendix_a).



DAG of models $\mathcal{M}_2$ and $\mathcal{M}_{2*}$

Figure 3.2: Causal model 2

The Appendix B @ref(appendix_b) describes how the true values for the estimands of the causal quantities of interest given in Table 2 of the *Expanse "Mediation analysis" report* were calculated. Those true values are the theoretical values expected under the causal and statistical models $\mathcal{M}_1$, $\mathcal{M}_{1*}$, $\mathcal{M}_2$ and $\mathcal{M}_{2*}$. Estimations that will be obtained from the data sets `df1.csv`, `df1_int.csv`, `df2.csv`, and `df2_int.csv` will be slightly different from the true values because of sample variability.

Chapter 4

Baron and Kenny, structural equation models

The Baron and Kenny approach can be applied if we make the assumption that no confounder of the $M \rightarrow Y$ relationship is affected by the exposure A. As a consequence we will use the `df1.csv` data set simulated from the Causal model 1 (Figure 3.1). We will also assume that there is no $A * M$ interaction effect on the outcome Y in the following examples. Such interaction effects can be dealt with using traditional regression models in very similar approaches described in chapter 5.

4.1 Baron and Kenny approach

The Baron & Kenny approach relies on sequential and step-wise estimation of linear regression models:

- A model for the total effect of the exposure A on the outcome Y (conditional on baseline confounders $L(0)$)

$$\mathbb{E}(Y \mid A, L(0)) = \theta_0 + \theta_A A + \theta_{L(0)} L(0)$$

- A model to test if the exposure A has an effect on the mediator M (conditional on baseline confounders $L(0)$ of the $A - M$ relationship)

$$\mathbb{E}(M \mid A, L(0)) = \beta_0 + \beta_A A + \beta_{L(0)} L(0)$$

- A model to estimate the direct effect of the exposure A on the outcome Y as well as the effect of the mediator M on the outcome, adjusted for baseline confounders $L(0)$ and confounders of the $M-Y$ relationship $L(1)$

$$\mathbb{E}(Y \mid A, M, L(1), L(0)) = \gamma_0 + \gamma_A A + \gamma_M M + \gamma_{L(0)} L(0) + \gamma_{L(1)} L(1)$$

The total effect is given by the θ_A coefficient from the 1st model.

The direct effect is given by the γ_A coefficient from the 3rd model.

The indirect effect can be calculated using:

- the "difference in coefficient" method based on the 1st and 3rd models: $\theta_A - \gamma_A$,
- or the "product of coefficient" method based on the 2nd and 3rd models: $\beta_A \times \gamma_M$.

```
## Model 1 to estimate the total effect:
model.tot.A.QoL <- lm(Y_qol ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                      data = df1)
summary(model.tot.A.QoL)
# Coefficients:
#                          Estimate Std. Error t value Pr(>|t|)
#   (Intercept)             71.8820     0.2155 333.565  < 2e-16 ***
#   A0_ace                  -5.0961     0.3486 -14.617  < 2e-16 *** <- Total effect
#   L0_male                 -1.1486     0.2194  -5.235 1.68e-07 ***
#   L0_parent_low_educ_lv  -3.4441     0.2295 -15.005  < 2e-16 ***

# The total effect of ACE on Quality of life is approximately equal to -5.1,
# given by the A0_ace coefficient:
model.tot.A.QoL$coefficients["A0_ace"]
# -5.096057

## Model 2 to estimate the effect of the exposure on the mediator
## because the mediator is binary, we might want to use a logistic or probit regressio
## for example
logit.model.A.M <- glm(M_smoking ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                       data = df1, family = "binomial")
summary(logit.model.A.M)
# effects estimated on the logit scale:
# Coefficients:
#                          Estimate Std. Error z value Pr(>|z|)
#   (Intercept)            -1.27152    0.04584 -27.736  < 2e-16 ***
#   A0_ace                  0.56168    0.06537   8.592  < 2e-16 *** <- effect of A on M
#   L0_male                 0.25455    0.04425   5.753 8.77e-09 ***
```

```
#   L0_parent_low_educ_lv  0.32683    0.04731   6.908 4.91e-12 ***
exp(coefficients(logit.model.A.M)["A0_ace"])
# Odds ratio = 1.753609  for the effect of ACE on the mediator (probability of smoking)

## Model 3 to estimate the direct effect of the exposure (conditional on the outcome) and
## the effect of M on Y, adjusted for confounders of the A-Y and M-Y relationships
model.A.M.QoL <- lm(Y_qol ~ A0_ace + M_smoking + L1 + L0_male + L0_parent_low_educ_lv,
                    data = df1)
summary(model.A.M.QoL)
# Coefficients:
#                          Estimate Std. Error t value Pr(>|t|)
#   (Intercept)            74.7858     0.2130 351.178  < 2e-16 ***
#   A0_ace                 -3.9650     0.3212 -12.345  < 2e-16 *** <- Direct effect
#   M_smoking              -8.7138     0.2221 -39.237  < 2e-16 *** <- effect of M on Y
#   L1                     -3.4252     0.2212 -15.483  < 2e-16 ***
#   L0_male                -0.7193     0.2017  -3.566 0.000364 ***
#   L0_parent_low_educ_lv  -2.8876     0.2112 -13.674  < 2e-16 ***

# The direct effect is approximately -4.0 given by the A0_ace coefficient
model.A.M.QoL$coefficients["A0_ace"]
# -3.965038

## Following the Baron & Kenny Steps, we would conclude that :
# - There is a significant total effect of ACEs on Quality of Life (Model 1)
# - There is a significant effect of ACEs on the mediator (smoking) (Model 2)
# - There is a significant effect of the mediator (smoking) on Qol (model 3)
# - The direct effect is significantly non-null
# => Conclusion: Smoking partially mediates the relationship between ACEs and QoL

### Estimation of the indirect effect:
### We can apply the difference in coefficient method to estimate the indirect effect:
### substract the direct effect from the Total effect:
ind.effect.dif.meth <- (model.tot.A.QoL$coefficients["A0_ace"] -
                          model.A.M.QoL$coefficients["A0_ace"])

# -1.131019
# the indirect effect is approximately -1.1
# The confidence interval of the indirect effect can be computed by bootstrap.

# Because the mediator is binary and we applied a logistic regression for Model 2,
# we cannot apply the product of coefficients combining a coefficient from
# Model 2 (logit scale) and from Model 3 (difference scale)

### Model 2bis
# Surprisingly, another possibility is to run a linear model of the binary mediator
```

```
# instead of the logistic regression to apply the "product of coefficient method"
# in order to estimate the indirect effect:
linear.model.A.M <- lm(M_smoking ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                       data = df1)
summary(linear.model.A.M)
# Coefficients:
#                          Estimate Std. Error t value Pr(>|t|)
#   (Intercept)           0.215398   0.008929  24.123  < 2e-16 ***
#   A0_ace                0.127180   0.014446   8.804  < 2e-16 ***
#   L0_male               0.052363   0.009091   5.760 8.67e-09 ***
#   L0_parent_low_educ_lv 0.065806   0.009511   6.919 4.83e-12 ***

## product of coefficient method:
ind.effect.prod.meth <- (linear.model.A.M$coefficients["A0_ace"] *
                           model.A.M.QoL$coefficients["M_smoking"])
# -1.108213
# which also gives an indirect effect of approximately -1.1
```

The Baron & Kenny approach is usually applied for continuous outcomes, using linear regressions. It is less adapted for binary outcomes.

However, as for the binary mediator, some authors suggested that using linear regressions of the mediator and the outcome could still give some results.

```
### Baron & Kenny approach for binary outcomes:
## Model 1: linear model of the probability of death to estimate the total effect:
model.tot.A.death <- lm(Y_death ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                        data = df1)
summary(model.tot.A.death)
# Coefficients:
#                          Estimate Std. Error t value Pr(>|t|)
#   (Intercept)           0.135282   0.007909  17.104  < 2e-16 ***
#   A0_ace                0.060247   0.012796   4.708 2.53e-06 *** <- Total effect
#   L0_male               0.050285   0.008053   6.244 4.43e-10 ***
#   L0_parent_low_educ_lv 0.059565   0.008425   7.070 1.65e-12 ***
# On a risk difference scale the total effect of ACE on the probability of death is
# approximately +6.0%

## Model 3: linear model to estimate the direct effect of the exposure (conditional on
## the outcome) and the effect of M on Y, adjusted for confounders of the A-Y
## and M-Y relationships
model.A.M.death <- lm(Y_death ~ A0_ace + M_smoking + L1 + L0_male + L0_parent_low_educ_
                      data = df1)
summary(model.A.M.death)
#   Coefficients:
```

```
#                         Estimate Std. Error t value Pr(>|t|)
#   (Intercept)           0.098691   0.008460  11.666  < 2e-16 ***
#   A0_ace                0.051509   0.012759   4.037 5.45e-05 *** <- Direct effect
#   M_smoking             0.064751   0.008822   7.340 2.31e-13 *** <- effect of M on Y
#   L1                    0.075533   0.008788   8.595  < 2e-16 ***
#   L0_male               0.047490   0.008013   5.927 3.19e-09 ***
#   L0_parent_low_educ_lv 0.055676   0.008389   6.637 3.36e-11 ***


# The direct effect is approximately +5.2% given by the A0_ace coefficient
model.A.M.death$coefficients["A0_ace"]
# 0.05150901

# The indirect effect can be calculated by the "difference in coefficient" method
# using coefficients from models 1 and 3
model.tot.A.death$coefficients["A0_ace"] - model.A.M.death$coefficients["A0_ace"]
# 0.008737889, i.e. approximately 0.9%

# or the product of coefficients using the previous model 2bis and model 3:
linear.model.A.M$coefficients["A0_ace"] * model.A.M.death$coefficients["M_smoking"]
# 0.008234978, i.e. approximately 0.8%
```

Chapter 5

Traditional regression models

Traditional regression models can be applied in the absence of an intermediate confounder $L(1)$ of the $M - Y$ relationship affected by the exposure A (Causal model 1). They can be used for two-way, three-way and four-way decomposition of the average total effect.

In the following examples, we use the `df1_int.csv` data set with a $A \star M$ interaction effect on the outcome.

```
df1_int <- read.csv(file = "df1_int.csv")
```

If we assumed that there was no $A \star M$ interaction, then the `A0_ace:M_smoking` interaction terms should be removed from the models below (applicable if we use the `df1.csv` data set).

5.1 Estimation of the Average Total Effect (ATE)

The average total effect is the difference between the mean outcome had the whole population been exposed to adverse childhood experience (ACE), compared to the mean outcome had the whole population been unexposed to ACE: $\text{ATE} = \mathbb{E}(Y_{A=1}) - \mathbb{E}(Y_{A=0})$.

For the quantitative outcome, the ATE of the adverse childhood experience A on the quality of life score Y can be estimated using a traditional linear regression of `Y_qol` on `A0_ace`, adjusted for the baseline confounders `L0_male` and `L0_parent_low_educ_lv`.

For the binary outcome (death), we can estimate a risk difference applying a Generalized Linear Model with a Gaussian distribution and identity link, as suggested by Naimi *et al* (Naimi and Whitcomb 2020).

The regression coefficient of the exposure variable A is used to estimate the risk difference or the average difference.

$$\mathbb{E}(Y \mid A, L(0)) = \alpha_0 + \alpha_A A + \alpha_{L(0)} L(0) \tag{5.1}$$

$$\hat{\Psi}^{\text{ATE}}_{\text{trad}} = \hat{\alpha}_A$$

```
# For quantitative outcomes, apply a linear regression of Y on A (A0_ace),
# adjusted for the baseline confounders L(0):
trad_ATE_qol <- lm(Y_qol ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                   data = df1_int)

# For binary outcomes, apply a GLM of Y on A with a Gaussian distribution and
# identity link, adjusted for the baseline confounders:
trad_ATE_death <- glm(Y_death ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                      family = gaussian("identity"),
                      data = df1_int)

# Use the regression coefficient of the exposure (A0_ace) to estimate the ATE
ATE_trad_qol <- coefficients(trad_ATE_qol)["A0_ace"]
# -7.210089

ATE_trad_death <- coefficients(trad_ATE_death)["A0_ace"]
# 0.07720726
```

The estimation of 95% confidence intervals could be obtained directly from the linear regression with quantitative outcomes (equation (5.1)). However, using a robust (sandwich) variance estimator or applying a bootstrap procedure is recommended (Naimi and Whitcomb 2020).

```
library(sandwich)

# for the Quality of Life outcome
ATE_trad_qol <- list(ATE = coef(trad_ATE_qol)["A0_ace"],
                     lo = coef(trad_ATE_qol)["A0_ace"] - qnorm(0.975) *
                       sqrt(sandwich(trad_ATE_qol)["A0_ace","A0_ace"]),
                     hi = coef(trad_ATE_qol)["A0_ace"] + qnorm(0.975) *
                       sqrt(sandwich(trad_ATE_qol)["A0_ace","A0_ace"]))

ATE_trad_qol
# ATE = -7.210089 , IC95% = [-7.978234 ; -6.441944]
```

```r
# for death outcome
ATE_trad_death <- list(ATE = coef(trad_ATE_death)["A0_ace"],
                       lo = coef(trad_ATE_death)["A0_ace"] - qnorm(0.975) *
                         sqrt(sandwich(trad_ATE_death)["A0_ace","A0_ace"]),
                       hi = coef(trad_ATE_death)["A0_ace"] + qnorm(0.975) *
                         sqrt(sandwich(trad_ATE_death)["A0_ace","A0_ace"]))
ATE_trad_death
# ATE = 0.07720726 , IC95% = [0.04945859 ; 0.1049559]

# 95% CI calculation applying a bootstrap procedure
library(boot)
bootfunc <- function(data,index){
  boot_dat <- data[index,]
  mod.qol <- lm(Y_qol ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                data = boot_dat)
  mod.death <- glm(Y_death ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                   family = gaussian("identity"),
                   data = boot_dat)
  est <- c(coef(mod.qol)["A0_ace"],
           coef(mod.death)["A0_ace"])
  return(est)
}

set.seed(1234)
boot_est <- boot(df1_int,bootfunc,R=2000)

# the 95% CI for the estimation of the ATE of ACE on QoL is:
boot.ci(boot_est, index = 1, type = "norm")
# (-7.978, -6.444 )

# the 95% CI for the estimation of the ATE of ACE on death is:
boot.ci(boot_est, index = 2, type = "norm")
# ( 0.0502,  0.1040 )
```

Alternatively for binary outcomes, the total effect conditional on baseline confounders can be expressed on an Odds Ratio scale OR^{TE}, using the logistic regression (5.2).

$$\text{logit}P(Y = 1 \mid A, L(0)) = \alpha_0 + \alpha_A A + \alpha'_{L(0)} L(0) \tag{5.2}$$

$$\text{OR}^{\text{TE}} \mid L(0) = \exp \hat{\alpha}_A$$

```
TE_death_model <- glm(Y_death ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                      family = "binomial",
                      data = df1_int)
res_TE_death <- summary(TE_death_model)
tot.effect.death.OR <- list(OR = exp(coef(res_TE_death)["A0_ace","Estimate"]),
                            lo = exp(coef(res_TE_death)["A0_ace","Estimate"] -
                                       qnorm(0.975) *
                                       coef(res_TE_death)["A0_ace","Std. Error"]),
                            hi = exp(coef(res_TE_death)["A0_ace","Estimate"] +
                                       qnorm(0.975) *
                                       coef(res_TE_death)["A0_ace","Std. Error"]))
tot.effect.death.OR
# OR = 1.523254 , 95% CI = [ 1.323317 ; 1.753398]
```

5.2 Two-way decomposition

In order to carry-out two-way decomposition mediation analyses, with a binary mediator and a continuous outcome, Valeri and VanderWeele suggest using the following linear regression of the outcome and logistic regression of the mediator:(Valeri and VanderWeele 2013)

$$\mathbb{E}(Y \mid A, M, L(0), L(1)) = \gamma_0 + \gamma_A A + \gamma_M M + \gamma_{A*M}(A*M) + \gamma'_{L(0)}L(0) + \gamma'_{L(1)}L(1) \tag{5.3}$$

$$\text{logit} P(M = 1 \mid A, L(0), L(1)) = \beta_0 + \beta_A A + \beta'_{L(0)}L(0) + \beta'_{L(1)}L(1) \tag{5.4}$$

If the outcome is binary, they suggest using the following logistic regression of the outcome instead of the previous linear regression:

$$\text{logit} P(Y \mid A, M, L(0), L(1)) = \gamma_0 + \gamma_A A + \gamma_M M + \gamma_{A*M}(A*M) + \gamma'_{L(0)}L(0) + \gamma'_{L(1)}L(1) \tag{5.5}$$

```
trad_qol_am <- lm(Y_qol ~ A0_ace + M_smoking + A0_ace:M_smoking +
                    L0_male + L0_parent_low_educ_lv + L1,
                  data = df1_int)
gamma.A.q <- coef(trad_qol_am)["A0_ace"]
gamma.M.q <- coef(trad_qol_am)["M_smoking"]
gamma.AM.q <- coef(trad_qol_am)["A0_ace:M_smoking"]

trad_m <- glm(M_smoking ~ A0_ace + L0_male + L0_parent_low_educ_lv + L1,
              family = "binomial",
              data = df1_int)
```

```
beta.0 <- coef(trad_m)["(Intercept)"]
beta.A <- coef(trad_m)["A0_ace"]

trad_death_am <- glm(Y_death ~ A0_ace + M_smoking + A0_ace:M_smoking +
                       L0_male + L0_parent_low_educ_lv + L1,
                     family = "binomial",
                     data = df1_int)
gamma.A.d <- coef(trad_death_am)["A0_ace"]
gamma.M.d <- coef(trad_death_am)["M_smoking"]
gamma.AM.d <- coef(trad_death_am)["A0_ace:M_smoking"]
```

5.2.1 Controlled Direct Effect

The Controlled Direct Effect is defined as $\text{CDE}_m = \mathbb{E}(Y_{A=1,M=m}) - \mathbb{E}(Y_{A=0,M=m})$:

For continuous outcome, using parameters from equation (5.3), it can be estimated by:

$$\text{CDE}_m = \hat{\gamma}_A + \hat{\gamma}_{A*M} \times m$$

```
### For a continuous outcome
# setting the mediator to M=0
trad_CDE_qol_m0 <- gamma.A.q + gamma.AM.q * 0
trad_CDE_qol_m0
# -3.715265
# setting the mediator to M=1
trad_CDE_qol_m1 <- gamma.A.q + gamma.AM.q * 1
trad_CDE_qol_m1
# -9.330657
```

For binary outcomes, using parameters from equation (5.5), it can be estimated on the OR scale by:

$$OR^{\text{CDE}_m} = \exp\left(\hat{\gamma}_A + \hat{\gamma}_{A*M} \times m\right)$$

```
### For a binary outcome
## setting the mediator to M=0
trad_OD_CDE_death_m0 <- exp(gamma.A.d + gamma.AM.d * 0)
trad_OD_CDE_death_m0
# OR_CDE_{M=0} = 1.442942

## setting the mediator to M=1
trad_OD_CDE_death_m1 <- exp(gamma.A.d + gamma.AM.d * 1)
trad_OD_CDE_death_m1
# OR_CDE_{M=1} = 1.461464
```

5.2.2 Natural Direct and Indirect effects

The Pure Natural Direct Effect (PNDE) and the Total Natural Indirect Effect (TNIE) are defined as:

- $\text{PNDE} = \mathbb{E}\left(Y_{A=1,M_{A=0}}) - \mathbb{E}(Y_{A=0,M_{A=0}}\right)$,
- $\text{TNIE} = \mathbb{E}\left(Y_{A=1,M_{A=1}}) - \mathbb{E}(Y_{A=1,M_{A=0}}\right)$.

Alternatively, one can use the Total Natural Direct Effect (TNDE) and the Pure Natural Indirect Effect (PNIE):

- $\text{TNDE} = \mathbb{E}\left(Y_{A=1,M_{A=1}}) - \mathbb{E}(Y_{A=0,M_{A=1}}\right)$,
- $\text{PNIE} = \mathbb{E}\left(Y_{A=0,M_{A=1}}) - \mathbb{E}(Y_{A=0,M_{A=0}}\right)$.

With a continuous outcome and a binary mediator, the PNDE and TNDE can be estimated using the linear regression of the outcome (equation (5.3)) and the logistic regression of the mediator (equation (5.4)):

$$\text{PNDE} \mid L(0), L(1) = \hat{\gamma}_A + \hat{\gamma}_{A*M} \frac{exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))}{1 + exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))}$$

$$\text{TNIE} \mid L(0), L(1) = (\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 1) \left[\frac{exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))}{1 + exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))} - \frac{exp(\hat{\beta}_0 + \hat{\beta}_A \times}{1 + exp(\hat{\beta}_0 + \hat{\beta}_A} \right.$$

The alternative TNDE ad PNIE can be estimated by:

$$\text{TNDE} \mid L(0), L(1) = \hat{\gamma}_A + \hat{\gamma}_{A*M} \frac{exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))}{1 + exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))}$$

$$\text{PNIE} \mid L(0), L(1) = (\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 0) \left[\frac{exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))}{1 + exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))} - \frac{exp(\hat{\beta}_0 + \hat{\beta}_A \times}{1 + exp(\hat{\beta}_0 + \hat{\beta}_A} \right.$$

Conditional on participants for which $L(0) = 0$ and $L(1) = 0$, these expressions are simplified:

$$\text{PNDE}\Big|(L(0) = 0, L(1) = 0) = \hat{\gamma}_A + \hat{\gamma}_{A*M} \frac{exp(\hat{\beta}_0)}{1 + exp(\hat{\beta}_0)}$$

$$\text{TNIE}\Big|(L(0) = 0, L(1) = 0) = (\hat{\gamma}_M + \hat{\gamma}_{A*M}) \left[\frac{exp(\hat{\beta}_0 + \hat{\beta}_A)}{1 + exp(\hat{\beta}_0 + \hat{\beta}_A)} - \frac{exp(\hat{\beta}_0)}{1 + exp(\hat{\beta}_0)} \right]$$

and

$$\text{TNDE}\Big|(L(0)=0,L(1)=0) = \hat{\gamma}_A + \hat{\gamma}_{A*M}\frac{exp(\hat{\beta}_0+\hat{\beta}_A)}{1+exp(\hat{\beta}_0+\hat{\beta}_A)}$$

$$\text{PNIE}\Big|(L(0)=0,L(1)=0) = \hat{\gamma}_M\left[\frac{exp(\hat{\beta}_0+\hat{\beta}_A)}{1+exp(\hat{\beta}_0+\hat{\beta}_A)} - \frac{exp(\hat{\beta}_0)}{1+exp(\hat{\beta}_0)}\right]$$

```
### For a continuous outcome, in the subgroup with L(0)=0 and L(1)=0
## The PNDE and TNIE are:
trad_PNDE_qol <- gamma.A.q + gamma.AM.q * (exp(beta.0)) / (1 + exp(beta.0))
trad_PNDE_qol
# -4.845089

trad_TNIE_qol <- (gamma.M.q + gamma.AM.q) *
  (exp(beta.0 + beta.A) / (1 + exp(beta.0 + beta.A)) -
     exp(beta.0) / (1 + exp(beta.0)))
trad_TNIE_qol
# -1.50119

## The TNDE and PNIE are:
trad_TNDE_qol <- gamma.A.q +
  gamma.AM.q * exp(beta.0 + beta.A) / (1 + exp(beta.0 + beta.A))
trad_TNDE_qol
# -5.436773

trad_PNIE_qol <- gamma.M.q *
  (exp(beta.0 + beta.A) /
     (1 + exp(beta.0 + beta.A)) - exp(beta.0) / (1 + exp(beta.0)))
trad_PNIE_qol
# -0.9095061
```

For binary outcomes, total, direct and indirect effects can be expressed on relative risk or odds ratio scales:

The total effect risk ratio is equal to :

$$\text{RR}^{\text{TE}} = \frac{\mathbb{E}(Y_1)}{\mathbb{E}(Y_0)} = \frac{\mathbb{E}(Y_{1,M_1})}{\mathbb{E}(Y_{0,M_0})}$$

The total effect risk ratio can be decomposed as the product of the PNDE risk ratio and the TNIE risk ratio:

$$\text{RR}^{\text{TE}} = \frac{\mathbb{E}(Y_{1,M_1})}{\mathbb{E}(Y_{0,M_0})} = \frac{\mathbb{E}(Y_{1,M_0})}{\mathbb{E}(Y_{0,M_0})} \times \frac{\mathbb{E}(Y_{1,M_1})}{\mathbb{E}(Y_{1,M_0})} = \text{RR}^{\text{PNDE}} \times \text{RR}^{\text{TNIE}}$$

Similarly, the total effect risk ratio can be decomposed as the product of the TNDE risk ratio and the PNIE risk ratio:

$$\mathrm{RR}^{\mathrm{TE}} = \frac{\mathbb{E}(Y_{1,M_1})}{\mathbb{E}(Y_{0,M_0})} = \frac{\mathbb{E}(Y_{1,M_1})}{\mathbb{E}(Y_{0,M_1})} \times \frac{\mathbb{E}(Y_{0,M_1})}{\mathbb{E}(Y_{0,M_0})} = \mathrm{RR}^{\mathrm{TNDE}} \times \mathrm{RR}^{\mathrm{PNIE}}$$

PNDE, TNIE, TNDE and PNIE can also be given on the OR scale,

$$\mathrm{OR}^{\mathrm{PNDE}} = \frac{\frac{P\left(Y_{A=1,M_{A=0}}=1\right)}{1-P\left(Y_{A=1,M_{A=0}}=1\right)}}{\frac{P\left(Y_{A=0,M_{A=0}}=1\right)}{1-P\left(Y_{A=0,M_{A=0}}=1\right)}} \quad , \quad \mathrm{OR}^{\mathrm{TNIE}} = \frac{\frac{P\left(Y_{A=1,M_{A=1}}=1\right)}{1-P\left(Y_{A=1,M_{A=1}}=1\right)}}{\frac{P\left(Y_{A=1,M_{A=0}}=1\right)}{1-P\left(Y_{A=1,M_{A=0}}=1\right)}}$$

and

$$\mathrm{OR}^{\mathrm{TNDE}} = \frac{\frac{P\left(Y_{A=1,M_{A=0}}=1\right)}{1-P\left(Y_{A=1,M_{A=0}}=1\right)}}{\frac{P\left(Y_{A=0,M_{A=0}}=1\right)}{1-P\left(Y_{A=0,M_{A=0}}=1\right)}} \quad \text{and} \quad \mathrm{OR}^{\mathrm{PNIE}} = \frac{\frac{P\left(Y_{A=1,M_{A=1}}=1\right)}{1-P\left(Y_{A=1,M_{A=1}}=1\right)}}{\frac{P\left(Y_{A=1,M_{A=0}}=1\right)}{1-P\left(Y_{A=1,M_{A=0}}=1\right)}}.$$

If the outcome is rare, we have $P(Y=1) \approx \frac{P(Y=1)}{1-P(Y=1)}$ so that, $\mathrm{OR}^{\mathrm{PNDE}}$ AND $\mathrm{OR}^{\mathrm{TNIE}}$ can be estimated using the logistic model of the outcome (equation (5.5)) and the logistic model of the mediator (equation (5.4)):

$$\mathrm{OR}^{\mathrm{PNDE}} \mid L(0), L(1) \approx \frac{\exp(\hat{\gamma}_A \times 1)\left[1 + \exp(\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 1 + \hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))\right]}{\exp(\hat{\gamma}_A \times 0)\left[1 + \exp(\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 0 + \hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))\right]}$$

and

$$\mathrm{OR}^{\mathrm{TNIE}} \mid L(0), L(1) \approx \frac{\left[1 + \exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))\right]\left[1 + \exp(\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 1 + \hat{\beta}_0 + \hat{\beta}_A\right.}{\left[1 + \exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))\right]\left[1 + \exp(\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 1 + \hat{\beta}_0 + \hat{\beta}_A\right.}$$

Similarly, if the outcome is rare, $\mathrm{OR}^{\mathrm{TNDE}}$ AND $\mathrm{OR}^{\mathrm{PNIE}}$ can be estimated using the logistic regression models for the outcome and the mediator (equations (5.5) and (5.4)):

$$\mathrm{OR}^{\mathrm{TNDE}} \mid L(0), L(1) \approx \frac{\exp\left(\hat{\gamma}_A \times 1\right)\left[1 + \exp\left(\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 1 + \hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1)\right)\right]}{\exp\left(\hat{\gamma}_A \times 0\right)\left[1 + \exp\left(\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 0 + \hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1)\right)\right]}$$

and

$$\mathrm{OR}^{\mathrm{PNIE}} \mid L(0), L(1) \approx \frac{\left[1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1)\right)\right]\left[1 + \exp\left(\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 0 + \hat{\beta}_0 + \hat{\beta}\right.\right.}{\left[1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1)\right)\right]\left[1 + \exp\left(\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 0 + \hat{\beta}_0 + \hat{\beta}\right.\right.}$$

Conditional on participants for which $L(0) = 0$ and $L(1) = 0$, these expressions are simplified:

$$\mathrm{OR}^{\mathrm{PNDE}}\Big|(L(0)=0, L(1)=0) \approx \frac{\exp\left(\hat{\gamma}_A\right)\left[1+\exp\left(\hat{\gamma}_M+\hat{\gamma}_{A*M}+\hat{\beta}_0\right)\right]}{\left[1+\exp\left(\hat{\gamma}_M+\hat{\beta}_0\right)\right]}$$

and

$$\mathrm{OR}^{\mathrm{TNIE}}\Big|(L(0)=0, L(1)=0) \approx \frac{\left[1+\exp\left(\hat{\beta}_0\right)\right]\times\left[1+\exp\left(\hat{\gamma}_M+\hat{\gamma}_{A*M}+\hat{\beta}_0+\hat{\beta}_A\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\right)\right]\times\left[1+\exp\left(\hat{\gamma}_M+\hat{\gamma}_{A*M}+\hat{\beta}_0\right)\right]}$$

Similarly, conditional on $L(0) = 0$ and $L(1) = 0$,

$$\mathrm{OR}^{\mathrm{TNDE}}\Big|(L(0)=0, L(1)=0) \approx \frac{\exp\left(\hat{\gamma}_A\right)\left[1+\exp\left(\hat{\gamma}_M+\hat{\gamma}_{A*M}+\hat{\beta}_0+\hat{\beta}_A\right)\right]}{\left[1+\exp\left(\hat{\gamma}_M+\hat{\beta}_0+\hat{\beta}_A\right)\right]}$$

$$\mathrm{OR}^{\mathrm{PNIE}}\Big|(L(0)=0, L(1)=0) \approx \frac{\left[1+\exp\left(\hat{\beta}_0\right)\right]\times\left[1+\exp\left(\hat{\gamma}_M+\hat{\beta}_0+\hat{\beta}_A\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\right)\right]\times\left[1+\exp\left(\hat{\gamma}_M+\hat{\beta}_0\right)\right]}$$

```r
### For a binary outcome, in the subgroup with L(0)=0 and L(1)=0
## The PNDE and TNIE are (on the OR scale):
trad_OR_PNDE_death <- exp(gamma.A.d) *
  (1 + exp(gamma.M.d + gamma.AM.d + beta.0 )) /
  (1 + exp(gamma.M.d + beta.0))
trad_OR_PNDE_death
# 1.448035

trad_OR_TNIE_death <- (1 + exp(beta.0)) *
  (1 + exp(gamma.M.d + gamma.AM.d + beta.0 + beta.A)) /
  ((1 + exp(beta.0 + beta.A)) * (1 + exp(gamma.M.d + gamma.AM.d + beta.0)))
trad_OR_TNIE_death
# 1.050029

## The TNDE and PNIE are (on the OR scale):
trad_OR_TNDE_death <- exp(gamma.A.d) *
  (1 + exp(gamma.M.d + gamma.AM.d + beta.0 + beta.A)) /
  (1 + exp(gamma.M.d + beta.0 + beta.A))
trad_OR_TNDE_death
# 1.450344

trad_OR_PNIE_death <- (1 + exp(beta.0)) *
  (1 + exp(gamma.M.d + beta.0 + beta.A)) /
  ((1 + exp(beta.0 + beta.A)) * (1 + exp(gamma.M.d + beta.0)))
trad_OR_PNIE_death
# 1.048358
```

The `regmedint` package (Regression-Based Causal Mediation Analysis with Interaction and Effect Modification Terms) can be used for two-way decomposition. Estimations of the CDE, PNDE, TNIE, TNDE and PNIE presented above can be obtained as we show in the following example.

For continuous outcomes:

```
library(regmedint)
regmedint_cont <- regmedint(data = df1_int,
                           ## Variables
                           yvar = "Y_qol",                    # outcome variable
                           avar = "A0_ace",                   # exposure
                           mvar = "M_smoking",                # mediator
                           cvar = c("L0_male",                # confounders
                                    "L0_parent_low_educ_lv",
                                    "L1"),
                           #eventvar = "event",     # only for survival outcome
                           ## Values at which effects are evaluated
                           a0 = 0,
                           a1 = 1,
                           m_cde = 0,              # mediator level for the CDE
                           c_cond = c(0,0,0),                 # covariate level
                           ## Model types
                           mreg = "logistic",
                           yreg = "linear",
                           ## Additional specification
                           interaction = TRUE,
                           casecontrol = FALSE)
summary(regmedint_cont)
#### Mediation analysis
#             est         se          Z            p      lower      upper
# cde  -3.7152652 0.41600219  -8.930879 0.000000e+00 -4.5306145 -2.8999159
# pnde -4.8450888 0.35052810 -13.822255 0.000000e+00 -5.5321113 -4.1580663
# tnie -1.5011902 0.20821830  -7.209694 5.608847e-13 -1.9092905 -1.0930898
# tnde -5.4367728 0.34049175 -15.967414 0.000000e+00 -6.1041244 -4.7694213
# pnie -0.9095061 0.12266064  -7.414817 1.219025e-13 -1.1499166 -0.6690957
# te   -6.3462790 0.38788368 -16.361294 0.000000e+00 -7.1065170 -5.5860409
# pm    0.2365465 0.02947624   8.024991 1.110223e-15  0.1787742  0.2943189

# note: te = total effect = (pnde + tnie) = (tnde + pnie)
#       pm = proportion mediated = tnie / te
```

For binary outcomes:

```
regmedint_bin <- regmedint(data = df1_int,
                           ## Variables
                           yvar = "Y_death",              # outcome variable
                           avar = "A0_ace",               # exposure
                           mvar = "M_smoking",            # mediator
                           cvar = c("L0_male",            # confounders
                                    "L0_parent_low_educ_lv",
                                    "L1"),
                           #eventvar = "event",     # only for survival outcome
                           ## Values at which effects are evaluated
                           a0 = 0,
                           a1 = 1,
                           m_cde = 0,               # mediator level for the CDE
                           c_cond = c(0,0,0),                # covariate level
                           ## Model types
                           mreg = "logistic",
                           yreg = "logistic",
                           ## Additional specification
                           interaction = TRUE,
                           casecontrol = FALSE)
results.binary <- summary(regmedint_bin)

# taking the exponential of the estimations
exp(results.binary$summary_myreg[c("cde","pnde","tnie","tnde","pnie","te"),
                                 c("est","lower","upper")])
#### Mediation analysis
#           est    lower    upper
# cde  1.442942 1.191195 1.747893
# pnde 1.448035 1.245470 1.683545
# tnie 1.050029 1.013842 1.087509
# tnde 1.450344 1.257042 1.673371
# pnie 1.048358 1.029285 1.067783
# te   1.520479 1.316954 1.755457
```

5.3 Three-way decomposition

In order to carry-out a three-way decomposition with standard regressions, we will use the same models as for the two-way decomposition (equations (5.3), (5.5) and (5.4)).

(VanderWeele 2013) defines:

- the PNDE $= \mathbb{E}\left(Y_{A=1,M_{A=0}}\right) - \mathbb{E}(Y_{A=0,M_{A=0}}\Big)$,

- the PNIE $= \mathbb{E}\left(Y_{A=0,M_{A=1}}) - \mathbb{E}(Y_{A=0,M_{A=0}}\right)$,
- and the mediated interactive effect MIE $= \mathbb{E}\left(\left[Y_{1,1} - Y_{1,0} - Y_{0,1} - Y_{0,0}\right] \times [M_1 - M_0]\right)$.

The sum of these 3 components is equal to the Average total effect (ATE).

With a continuous outcome and a binary mediator, the PNDE and PNIE can be estimated as for the two-way decomposition (section 5.2.2) using the linear regression of the outcome (equation (5.3)) and the logistic regression of the mediator (equation (5.4)).

The mediated interactive effect can be estimated using the same equations (5.3) and (5.4), by:

$$\textsc{mie} \mid (L(0), L(1)) = \hat{\gamma}_{A*M} \left[\frac{\exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))} - \frac{\exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)} L}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)}} \right.$$

Conditional on participants for which $L(0) = 0$ and $L(1) = 0$, the expression is simplified:

$$\text{MIE}\Big|(L(0) = 0, L(1) = 0) = \hat{\gamma}_{A*M} \left[\frac{\exp(\hat{\beta}_0 + \hat{\beta}_A)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_A)} - \frac{\exp(\hat{\beta}_0)}{1 + \exp(\hat{\beta}_0)} \right]$$

```
### For a continuous outcome, in the subgroup with L(0)=0 and L(1)=0
## The PNDE is:
trad_PNDE_qol <- gamma.A.q + gamma.AM.q * (exp(beta.0)) / (1 + exp(beta.0))
trad_PNDE_qol
# -4.845089

## The PNIE is:
trad_PNIE_qol <- gamma.M.q *
  (exp(beta.0 + beta.A) /
     (1 + exp(beta.0 + beta.A)) - exp(beta.0) / (1 + exp(beta.0)))
trad_PNIE_qol
# -0.9095061

## The MIE is:
trad_MIE_qol <- gamma.AM.q *
  (exp(beta.0 + beta.A) / (1 + exp(beta.0 + beta.A)) -
     exp(beta.0) / (1 + exp(beta.0)))
trad_MIE_qol
# -0.591684
```

With a binary outcome, the total effect, the direct and indirect effects can be expressed using risk ratios or odds ratios. In order to express the mediated interactive effect, (VanderWeele 2013) suggested decomposing the *excess relative* *risk of the total effect* ($\text{RR}^{\text{TE}} - 1$), which enables the expression of the mediated interactive effect on an additive scale.

On the difference scale, the total effect can be decomposed as the sum of the PNDE, the PNIE and the MIE:

$$\begin{aligned}
\mathbb{E}(Y_1) - \mathbb{E}(Y_0) = \; & \left[\mathbb{E}\left(Y_{1M_0}\right) - \mathbb{E}\left(Y_{0M_0}\right)\right] + \left[\mathbb{E}\left(Y_{0M_1}\right) - \mathbb{E}\left(Y_{0M_0}\right)\right] \\
& + \left[\left[\mathbb{E}\left(Y_{1M_1}\right) - \mathbb{E}\left(Y_{1M_0}\right)\right] - \left[\mathbb{E}\left(Y_{0M_1}\right) - \mathbb{E}\left(Y_{0M_0}\right)\right]\right] \\
= \; & \text{PNDE} + \text{PNIE} \\
& + \text{MIE}
\end{aligned}$$

Dividing by $\mathbb{E}(Y_0) = \mathbb{E}\left(Y_{0M_0}\right)$, we obtain the excess relative risk of the total effect decomposition:

$$\begin{aligned}
\frac{\mathbb{E}(Y_1)}{\mathbb{E}(Y_0)} - 1 = \; & \left[\frac{\mathbb{E}\left(Y_{1M_0}\right)}{\mathbb{E}\left(Y_{0M_0}\right)} - 1\right] + \left[\frac{\mathbb{E}\left(Y_{0M_1}\right)}{\mathbb{E}\left(Y_{0M_0}\right)} - 1\right] \\
& + \left[\frac{\mathbb{E}\left(Y_{1M_1}\right)}{\mathbb{E}\left(Y_{0M_0}\right)} - \frac{\mathbb{E}\left(Y_{1M_0}\right)}{\mathbb{E}\left(Y_{0M_0}\right)} - \frac{\mathbb{E}\left(Y_{0M_1}\right)}{\mathbb{E}\left(Y_{0M_0}\right)} + 1\right]
\end{aligned}$$

where the first component is the *excess relative risk due to the PNDE*, the second component is the *excess relative risk due to the PNIE* and the third component is the *mediated excess relative risk due to interaction.*

If the outcome is rare, relative risks are approximately equal to odds ratios, and the 3 components of the excess relative risk can be estimated using the logistic regression of the outcome (equation (5.5)) and the logistic regression of the mediator (equation (5.4)).

The component of the excess relative risk due to the PNDE is approximately equal to:

$$\text{RR}^{\text{PNDE}} - 1 \approx \frac{\exp\left[\hat{\gamma}_A(1-0)\right]\left[1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)}L(0) + \hat{\beta}'_{L(1)}L(1) + \hat{\gamma}_M + \hat{\gamma}_{A*M} \times 1\right)\right]}{\left[1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)}L(0) + \hat{\beta}'_{L(1)}L(1) + \hat{\gamma}_M + \hat{\gamma}_{A*M} \times 0\right)\right]} - 1$$

The component of the excess relative risk due to the PNIE is approximately equal to:

$$\text{RR}^{\text{PNIE}} - 1 \approx \frac{\left[1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)}L(0) + \hat{\beta}'_{L(1)}L(1)\right)\right]\left[1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)}L(0) + \hat{\beta}'_{L(1)}L(1) + \hat{\gamma}_M +\right.\right.}{\left[1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)}L(0) + \hat{\beta}'_{L(1)}L(1)\right)\right]\left[1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)}L(0) + \hat{\beta}'_{L(1)}L(1) + \hat{\gamma}_M +\right.\right.}$$

The component of the excess relative risk due to the mediated interactive effect is approximately equal to:

$$\begin{aligned}
\text{RERI}_{mediated} \approx \; & \frac{\exp[\hat{\gamma}_A(1-0)]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times1+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times1\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times0\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times1+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]} \\
& - \frac{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times1+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times0\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times0\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times1+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]} \\
& - \frac{\exp[\hat{\gamma}_A(1-0)]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times1\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times0\right)\right]} + 1
\end{aligned}$$

Conditional on participants for which $L(0) = 0$ and $L(1) = 0$, these expressions are simplified:

$$\text{RR}^{\text{PNDE}} - 1 \approx \frac{\exp\left(\hat{\gamma}_A\right)\left[1 + \exp\left(\hat{\beta}_0 + \hat{\gamma}_M + \hat{\gamma}_{A*M}\right)\right]}{\left[1 + \exp\left(\hat{\beta}_0 + \hat{\gamma}_M\right)\right]} - 1$$

$$\text{RR}^{\text{PNIE}} - 1 \approx \frac{\left[1 + \exp\left(\hat{\beta}_0\right)\right]\left[1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_A + \hat{\gamma}_M\right)\right]}{\left[1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_A\right)\right]\left[1 + \exp\left(\hat{\beta}_0 + \hat{\gamma}_M\right)\right]} - 1$$

and

$$\text{RERI}_{\text{mediated}} \approx \frac{\exp(\hat{\gamma}_A)\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A+\hat{\gamma}_M+\hat{\gamma}_{A*M}\right)\right]\left[1+\exp\left(\hat{\beta}_0\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\right)\right]} - \frac{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A+\hat{\gamma}_M\right)\right]\left[1+\exp\left(\hat{\beta}_0\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\right)\right]} - \frac{\exp(\hat{\gamma}_A)\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M+\hat{\gamma}_{A*M}\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)\right]} + 1$$

```
### For a binary outcome, in the subgroup with L(0)=0 and L(1)=0
## The excess relative risk is 52.3% (calculated from the OR of the total effect)
1.523254 - 1
# 0.523254

## The excess relative risk is decomposed into 3 components:
## The component of the excess relative risk due to PNDE is:
comp_PNDE_death <- exp(gamma.A.d) * (1 + exp(beta.0 + gamma.M.d + gamma.AM.d)) /
  (1 + exp(beta.0 + gamma.M.d)) - 1
comp_PNDE_death
# 0.4480347

## The component of the excess relative risk due to PNIE is:
comp_PNIE_death <- (1 + exp(beta.0)) * (1 + exp(beta.0 + beta.A + gamma.M.d)) /
  ((1 + exp(beta.0 + beta.A)) * (1 + exp(beta.0 + gamma.M.d))) - 1
comp_PNIE_death
# 0.04835753

## The component of the excess relative risk due to the mediated interactive
## effect is:
comp_MIE_qol <- exp(gamma.A.d) *
  (1 + exp(beta.0 + beta.A + gamma.M.d + gamma.AM.d)) * (1 + exp(beta.0)) /
  ((1 + exp(beta.0 + gamma.M.d)) * (1 + exp(beta.0 + beta.A))) -
  (1 + exp(beta.0 + beta.A + gamma.M.d)) * (1 + exp(beta.0)) /
  ((1 + exp(beta.0 + gamma.M.d)) * (1 + exp(beta.0 + beta.A))) -
  exp(gamma.A.d) * (1 + exp(beta.0 + gamma.M.d + gamma.AM.d)) /
  (1 + exp(beta.0 + gamma.M.d)) + 1
comp_MIE_qol
# 0.02408674
```

In this example, the excess relative risk of the exposure to adverse childhood (ACE) exposure is $\approx 52.3\%$, and of this excess relative risk…:

- $\approx 44.8\%$ is attributable to the PNDE of ACE,
- $\approx 4.8\%$ is attributable to the PNIE of ACE through smoking,
- $\approx 2.4\%$ is attributable to the mediated interactive effect between ACE and smoking.

Note: in this simulated data, the probability of death is around 20%, so that the requirement of a rare outcome is not really fulfilled (usually, we would consider $< 10\%$ to be acceptable).

5.4 Four-way decomposition

The same models as for the two-way and three-way decomposition (equations (5.3), (5.5) and (5.4)) will be used in order to apply the four-way decomposition.

(VanderWeele 2014) defines:

- the $\text{CDE}_{M=0} = \mathbb{E}\left(Y_{1,0}\right) - \mathbb{E}\left(Y_{0,0}\right)$,
- the mediated interaction effect $\text{MIE} = \mathbb{E}\left(\left[Y_{1,1} - Y_{1,0} - Y_{0,1} - Y_{0,0}\right] \times [M_1 - M_0]\right)$,
- the reference interaction effect $\text{RIE} = \mathbb{E}\left(\left[Y_{1,1} - Y_{1,0} - Y_{0,1} - Y_{0,0}\right] \times M_0\right)$,
- and the $\text{PNIE} = \mathbb{E}\left(Y_{A=0,M_{A=1}}) - \mathbb{E}(Y_{A=0,M_{A=0}}\right)$.

The sum of these 4 components is equal to the Average total effect (ATE), and if the exposure affects the outcome, then at least one of these 4 components should be non-null.

With a continuous outcome and a binary mediator, the $\text{CDE}_{M=0}$ and PNIE can be estimated as for the two-way decomposition (sections 5.2.1 and 5.2.2), and the MIE can be estimated as for the three-way decomposition (section 5.3), using the linear regression of the outcome (equation (5.3)) and the logistic regression of the mediator (equation (5.4)).

$$\text{CDE}_{M=0} \mid (L(0), L(1)) = \hat{\gamma}_A + \hat{\gamma}_{A*M} \times 0$$

$$\text{PNIE} \mid L(0), L(1) = (\hat{\gamma}_M + \hat{\gamma}_{A*M} \times 0) \left[\frac{exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))}{1 + exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)} L(0) + \hat{\beta}'_{L(1)} L(1))} - \frac{exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)} L(}{1 + exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)} L} \right.$$

and

$$\text{MIE} \mid (L(0), L(1)) = \hat{\gamma}_{A*M} \Bigg[\frac{\exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)}L(0) + \hat{\beta}'_{L(1)}L(1))}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_A \times 1 + \hat{\beta}'_{L(0)}L(0) + \hat{\beta}'_{L(1)}L(1))} - \frac{\exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)}L}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)}}$$

The RIE can be estimated by:

$$\text{RIE} \mid (L(0), L(1)) = \hat{\gamma}_{A*M} \left[\frac{\exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)}L(0) + \hat{\beta}'_{L(1)}L(1))}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_A \times 0 + \hat{\beta}'_{L(0)}L(0) + \hat{\beta}'_{L(1)}L(1))} - 0 \right]$$

```
### For a continuous outcome, in the subgroup with L(0)=0 and L(1)=0
## The CDE_(M=0) is:
trad_CDE_qol_m0 <- gamma.A.q + gamma.AM.q * 0
trad_CDE_qol_m0
# -3.715265

## The PNIE is:
trad_PNIE_qol <- gamma.M.q *
  (exp(beta.0 + beta.A) /
     (1 + exp(beta.0 + beta.A)) - exp(beta.0) / (1 + exp(beta.0)))
trad_PNIE_qol
# -0.9095061

## The MIE is:
trad_MIE_qol <- gamma.AM.q *
  (exp(beta.0 + beta.A) / (1 + exp(beta.0 + beta.A)) -
     exp(beta.0) / (1 + exp(beta.0)))
trad_MIE_qol
# -0.591684

## The RIE is:
trad_RIE_qol <- gamma.AM.q * (exp(beta.0)) / (1 + exp(beta.0))
trad_RIE_qol
# -1.129824
```

With a binary outcome and a binary mediator, (VanderWeele 2014) suggested decomposing the *excess relative risk of the total effect* ($\text{RR}^{\text{TE}} - 1$) (as for the 3-way decomposition), which enables the expression of the MIE and the RIE on an additive scale.

If the outcome is rare,

The component of the excess relative risk due to the CDE is approximately equal to:

$$\frac{\mathbb{E}(Y_{0,0}|L(0),L(1))}{\mathbb{E}(Y_0|L(0),L(1))}\left(\frac{\mathbb{E}(Y_{1,0}|L(0),L(1))}{\mathbb{E}(Y_{0,0}|L(0),L(1))}-1\right) \approx \frac{\exp(\hat{\gamma}_A(1-0)+\hat{\gamma}_M\times 0+\hat{\gamma}_{A*M}\times 1\times 0)\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(0)}L(1)\right)\right]}{1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(0)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 0\right)} - \frac{\exp(\hat{\gamma}_M\times 0+\hat{\gamma}_{A*M}\times 0\times 0)\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(0)}L(1)\right)\right]}{1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(0)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 0\right)}$$

The component of the excess relative risk due to the PNIE is approximately equal to:

$$\mathrm{RR}^{\mathrm{PNIE}}-1 \approx \frac{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 1+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\right.\right.}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 1+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\right.\right.}$$

The component of the excess relative risk due to the mediated interactive effect is approximately equal to:

$$\mathrm{RERI}_{mediated} \approx \frac{\exp[\hat{\gamma}_A(1-0)]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 1+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 1\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 0\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 1+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]} - \frac{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 1+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 0\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 0\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 1+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]} - \frac{\exp[\hat{\gamma}_A(1-0)]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 1\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 0\right)\right]} + 1$$

and the component of the excess relative risk due to the reference interaction effect is approximately equal to:

$$\mathrm{RERI}_{\mathrm{ref}} \approx \frac{\exp[\hat{\gamma}_A(1-0)]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 1\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 0\right)\right]} - 1 - \frac{\exp[\hat{\gamma}_A(1-0)+\hat{\gamma}_M\times 0+\hat{\gamma}_{A*M}1\times 0]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]}{1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 0\right)} + \frac{\exp(\hat{\gamma}_M\times 0+\hat{\gamma}_{A*M}\times 0\times 0)\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)\right)\right]}{1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\times 0+\hat{\beta}'_{L(0)}L(0)+\hat{\beta}'_{L(1)}L(1)+\hat{\gamma}_M+\hat{\gamma}_{A*M}\times 0\right)}$$

Conditional on participants for which $L(0) = 0$ and $L(1) = 0$, these expressions are simplified:

The component of the excess relative risk due to the CDE is approximately equal to:

$$\approx \frac{\exp\left(\hat{\gamma}_A\right)\left[1+\exp\left(\hat{\beta}_0\right)\right]}{1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)} - \frac{\left[1+\exp\left(\hat{\beta}_0\right)\right]}{1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)}$$

The component of the excess relative risk due to the PNIE is approximately equal to:

$$\mathrm{RR}^{\mathrm{PNIE}} - 1 \approx \frac{\left[1+\exp\left(\hat{\beta}_0\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A+\hat{\gamma}_M\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)\right]} - 1$$

The component of the excess relative risk due to the mediated interactive effect is approximately equal to:

$$\text{RERI}_{mediated} \approx \frac{\exp(\hat{\gamma}_A)\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A+\hat{\gamma}_M+\hat{\gamma}_{A*M}\right)\right]\left[1+\exp\left(\hat{\beta}_0\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\right)\right]} - \frac{\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A+\hat{\gamma}_M\right)\right]\left[1+\exp\left(\hat{\beta}_0\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)\right]\left[1+\exp\left(\hat{\beta}_0+\hat{\beta}_A\right)\right]} - \frac{\exp(\hat{\gamma}_A)\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M+\hat{\gamma}_{A*M}\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)\right]} + 1$$

and the component of the excess relative risk due to the reference interaction effect is approximately equal to:

$$\text{RERI}_{\text{ref}} \approx \frac{\exp\left(\hat{\gamma}_A\right)\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M+\hat{\gamma}_{A*M}\right)\right]}{\left[1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)\right]} - 1 - \frac{\exp\left(\hat{\gamma}_A\right)\left[1+\exp\left(\hat{\beta}_0\right)\right]}{1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)} + \frac{\left[1+\exp\left(\hat{\beta}_0\right)\right]}{1+\exp\left(\hat{\beta}_0+\hat{\gamma}_M\right)}$$

```
### For a binary outcome, in the subgroup with L(0)=0 and L(1)=0
## The excess relative risk is 52.3% (calculated from the OR of the total effect)
1.523254 - 1
# 0.523254

## The excess relative risk is decomposed into 4 components:
## The component of the excess relative risk due to the CDE(M=0) is:
comp_CDE_death_m0 <- exp(gamma.A.d) * (1 + exp(beta.0)) /
  (1 + exp(beta.0 + gamma.M.d)) -
  (1 + exp(beta.0)) / (1 + exp(beta.0 + gamma.M.d))
comp_CDE_death_m0
# 0.402041

## The component of the excess relative risk due to the PNIE is:
comp_PNIE_death <- (1 + exp(beta.0)) * (1 + exp(beta.0 + beta.A + gamma.M.d)) /
  ((1 + exp(beta.0 + beta.A)) * (1 + exp(beta.0 + gamma.M.d))) - 1
comp_PNIE_death
# 0.04835753

## The component of the excess relative risk due to the MIE is:
comp_MIE_death <- exp(gamma.A.d) * (1 + exp(beta.0 + beta.A + gamma.M.d +
                                              gamma.AM.d)) * (1 + exp(beta.0)) /
  ((1 + exp(beta.0 + gamma.M.d)) * (1 + exp(beta.0 + beta.A))) -
  (1 + exp(beta.0 + beta.A + gamma.M.d)) * (1 + exp(beta.0)) /
  ((1 + exp(beta.0 + gamma.M.d)) * (1 + exp(beta.0 + beta.A))) -
  exp(gamma.A.d) * (1 + exp(beta.0 + gamma.M.d + gamma.AM.d)) /
  (1 + exp(beta.0 + gamma.M.d)) + 1
comp_MIE_death
# 0.02408674

## The component of the excess relative risk due to the RIE is:
comp_RIE_death <- exp(gamma.A.d) * (1 + exp(beta.0 + gamma.M.d + gamma.AM.d)) /
```

```
  (1 + exp(beta.0 + gamma.M.d)) - 1 -
  exp(gamma.A.d) * (1 + exp(beta.0)) / (1 + exp(beta.0 + gamma.M.d)) +
  (1 + exp(beta.0)) / (1 + exp(beta.0 + gamma.M.d))
comp_RIE_death
# 0.04599376
```

In this example, the excess relative risk of the exposure to adverse childhood (ACE) exposure is $\approx$ 52.3%, and of this excess relative risk…:

- $\approx$ 40.2% is attributable to the CDE of ACE,
- $\approx$ 4.8% is attributable to the PNIE of ACE through smoking,
- $\approx$ 2.4% is attributable to the mediated interactive effect between ACE and smoking.
- and $\approx$ 4.6% is attributable to the (ACE $*$ smoking) reference interactive effect.

Note: in this simulated data, the probability of death is around 20%, so that the requirement of a rare outcome is not really fulfilled (usually, we would consider $< 10\%$ to be acceptable).

R package for 3-way and 4-way decomposition

The `CMAverse` R package (a suite of functions for causal mediation analysis) can be used for 3-way and 4-way decomposition. Estimations of the CDE(M=0), PNIE, MIE and INTref presented above can be obtained as we show in the following example.

For continuous outcomes:

```
library(CMAverse)

### For the continuous outcome
## Closed-form parameter function estimation and delta method inferece
res_rb_param_delta <- cmest(data = df1_int,
                            model = "rb", # for "regression based" (rb) approach
                            outcome = "Y_qol",         # outcome variable
                            exposure = "A0_ace",       # exposure variable
                            mediator = "M_smoking",    # mediator
                            basec = c("L0_male",       # confounders
                                      "L0_parent_low_educ_lv",
                                      "L1"),
                            EMint = TRUE, # exposures*mediator interaction
                            mreg = list("logistic"), # model of the mediator
                            yreg = "linear",        # model of the outcome
                            astar = 0,
                            a = 1,
```

```
                            mval = list(0),
                            basecval = list(0,0,0),      # covariate level
                            estimation = "paramfunc", #  closed-form parameter
                                                      # function estimation
                            inference = "delta") # IC95% : delta method
summary(res_rb_param_delta)
# Closed-form parameter function estimation with
# delta method standard errors, confidence intervals and p-values
#
#                Estimate Std.error  95% CIL 95% CIU    P.val
#   cde          -3.71527   0.41600 -4.53061  -2.900  < 2e-16 ***   CDE(M=0)
#   pnde         -4.84509   0.35053 -5.53211  -4.158  < 2e-16 ***
#   tnde         -5.43677   0.34049 -6.10412  -4.769  < 2e-16 ***
#   pnie         -0.90951   0.12266 -1.14992  -0.669 1.22e-13 ***   PNIE
#   tnie         -1.50119   0.20822 -1.90929  -1.093 5.61e-13 ***
#   te           -6.34628   0.38788 -7.10652  -5.586  < 2e-16 ***
#   intref       -1.12982   0.13824 -1.40076  -0.859 2.22e-16 ***   INTref
#   intmed       -0.59168   0.10398 -0.79547  -0.388 1.27e-08 ***   MIE
#   cde(prop)     0.58542   0.04505  0.49712   0.674  < 2e-16 ***
#   intref(prop)  0.17803   0.02560  0.12786   0.228 3.51e-12 ***
#   intmed(prop)  0.09323   0.01586  0.06216   0.124 4.11e-09 ***
#   pnie(prop)    0.14331   0.01655  0.11087   0.176  < 2e-16 ***
#   pm            0.23655   0.02948  0.17877   0.294 1.11e-15 ***
#   int           0.27126   0.03780  0.19717   0.345 7.20e-13 ***
#   pe            0.41458   0.04505  0.32627   0.503  < 2e-16 ***
#   ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# for the 3-way decomposition, the PNDE, PNIE, and MIE are given by:
data.frame("Estimate" = res_rb_param_delta$effect.pe,
           "lower95CI" = res_rb_param_delta$effect.ci.low,
           "upper95CI" = res_rb_param_delta$effect.ci.high,
           "P.value" = res_rb_param_delta$effect.pval)[c("pnde","pnie", "intmed"),]
#          Estimate  lower95CI  upper95CI      P.value
# pnde   -4.8450888 -5.5321113 -4.1580663 0.000000e+00
# pnie   -0.9095061 -1.1499166 -0.6690957 1.219025e-13
# intmed -0.5916840 -0.7954739 -0.3878941 1.266206e-08

# for the 4-way decomposition, the CDE(M=0), Intref, MIE and PNIE are given by:
data.frame("Estimate" = res_rb_param_delta$effect.pe,
           "lower95CI" = res_rb_param_delta$effect.ci.low,
           "upper95CI" = res_rb_param_delta$effect.ci.high,
           "P.value" = res_rb_param_delta$effect.pval)[c("cde","intref","intmed","pnie
#          Estimate  lower95CI  upper95CI      P.value
# cde    -3.7152652 -4.5306145 -2.8999159 0.000000e+00
```

```
# intref -1.1298236 -1.4007600 -0.8588871 2.220446e-16
# intmed -0.5916840 -0.7954739 -0.3878941 1.266206e-08
# pnie   -0.9095061 -1.1499166 -0.6690957 1.219025e-13
```

For binary outcomes:

```
### For the binary outcome
## Closed-form parameter function estimation and delta method inferece
res_rb_param_delta <- cmest(data = df1_int,
                            model = "rb", # for "regression based" (rb) approach
                            outcome = "Y_death",        # outcome variable
                            exposure = "A0_ace",      # exposure variable
                            mediator = "M_smoking",   # mediator
                            basec = c("L0_male",      # confounders
                                      "L0_parent_low_educ_lv",
                                      "L1"),
                            EMint = TRUE, # exposures*mediator interaction
                            mreg = list("logistic"), # model of the mediator
                            yreg = "logistic",       # model of the outcome
                            astar = 0,
                            a = 1,
                            mval = list(0),
                            basecval = list(0,0,0),      # covariate level
                            estimation = "paramfunc", #  closed-form parameter
                            # function estimation
                            inference = "delta") # IC95% : delta method
summary(res_rb_param_delta)
# Closed-form parameter function estimation with
# delta method standard errors, confidence intervals and p-values
#
# Estimate Std.error  95% CIL 95% CIU    P.val
#   Rcde            1.44294   0.14115  1.19120   1.748 0.000178 ***
#   Rpnde           1.44803   0.11133  1.24547   1.684 1.47e-06 ***
#   Rtnde           1.45034   0.10585  1.25704   1.673 3.50e-07 ***
#   Rpnie           1.04836   0.00982  1.02929   1.068 4.62e-07 ***
#   Rtnie           1.05003   0.01879  1.01384   1.088 0.006368 **
#   Rte             1.52048   0.11148  1.31695   1.755 1.10e-08 ***
#   ERcde           0.40204   0.12699  0.15315   0.651 0.001546 **  CDE(M=0)
#   ERintref        0.04599   0.04821 -0.04850   0.140 0.340093     INTref
#   ERintmed        0.02409   0.02543 -0.02576   0.074 0.343607     MIE
#   ERpnie          0.04836   0.00982  0.02911   0.068 8.47e-07 *** PNIE
#   ERcde(prop)     0.77244   0.14282  0.49252   1.052 6.36e-08 ***
#   ERintref(prop)  0.08837   0.09311 -0.09413   0.271 0.342600
#   ERintmed(prop)  0.04628   0.04901 -0.04978   0.142 0.345058
#   ERpnie(prop)    0.09291   0.02602  0.04192   0.144 0.000356 ***
```

```
#   pm              0.13919   0.05498  0.03142   0.247 0.011359 *
#   int             0.13465   0.14186 -0.14340   0.413 0.342562
#   pe              0.22756   0.14282 -0.05237   0.507 0.111092
#   ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


# for the 3-way decomposition, the excess relative risk due to
# the PNDE, PNIE and MIE are given by:
data.frame("Estimate" = res_rb_param_delta$effect.pe - 1,
           "lower95CI" = res_rb_param_delta$effect.ci.low - 1,
           "upper95CI" = res_rb_param_delta$effect.ci.high - 1)[c("Rpnde"),]
#        Estimate lower95CI upper95CI
# Rpnde 0.4480347 0.2454699 0.6835449
# and
data.frame("Estimate" = res_rb_param_delta$effect.pe,
           "lower95CI" = res_rb_param_delta$effect.ci.low,
           "upper95CI" = res_rb_param_delta$effect.ci.high,
           "P.value" = res_rb_param_delta$effect.pval)[c("ERpnie", "ERintmed"),]
#            Estimate   lower95CI  upper95CI      P.value
# ERpnie   0.04835753  0.02910970 0.06760535 8.473169e-07
# ERintmed 0.02408674 -0.02576124 0.07393472 3.436070e-01




# for the 4-way decomposition, the CDE(M=0), the excess relative risk due to
# CDE(M=0), Intref, MIE and PNIE are given by:
data.frame("Estimate" = res_rb_param_delta$effect.pe,
           "lower95CI" = res_rb_param_delta$effect.ci.low,
           "upper95CI" = res_rb_param_delta$effect.ci.high,
           "P.value" = res_rb_param_delta$effect.pval)[c("ERcde","ERintref","ERintmed"
#            Estimate   lower95CI  upper95CI      P.value
# ERcde    0.40204098  0.15315072 0.65093124 1.545523e-03
# ERintref 0.04599376 -0.04850084 0.14048835 3.400930e-01
# ERintmed 0.02408674 -0.02576124 0.07393472 3.436070e-01
# ERpnie   0.04835753  0.02910970 0.06760535 8.473169e-07
```

Chapter 6

G-computation

If we make the assumption that the intermediate confounder $L(1)$ of the $M - Y$ relationship is affected by the exposure A (Causal model 2, Figure 3.2), it is necessary to use other methods than traditional regressions models. To illustrate g-computation estimators, we will use the `df2_int.csv` data set, which was generated from a system corresponding to this assumption. Moreover, we will assume that there is an $A \star M$ interaction effect on the outcome.

G-computation can be used for the estimation of the total effect and two-way decomposition (CDE, marginal and conditional randomized direct and indirect effects).

6.1 Estimation of the Average Total Effect (ATE)

The following steps describe the implementation of the g-computation estimator of the average total effect ATE $= \mathbb{E}(Y_{A=1}) - \mathbb{E}(Y_{A=0})$:

1. Fit a logistic or a linear regression to estimate $\overline{Q} = \mathbb{E}(Y \mid A, L(0))$
2. Use this estimate to predict an outcome for each subject $\hat{\overline{Q}}(A = 0)_i$ and $\hat{\overline{Q}}(A = 1)_i$, by evaluating the regression fit $\overline{Q}$ at $A = 0$ and $A = 1$ respectively
3. Plug the predicted outcomes in the g-formula and use the sample mean to estimate Ψ_{ATE}

$$\hat{\Psi}^{\text{ATE}}_{\text{gcomp}} = \frac{1}{n} \sum_{i=1}^{n} \left[\hat{\overline{Q}}(A = 1)_i - \hat{\overline{Q}}(A = 0)_i \right] \tag{6.1}$$

For continuous outcomes, $\overline{Q}(A = a)$ functions can be estimated using linear regressions. For binary outcomes, they can be estimated using logistic regressions.

```
## 1. Estimate Qbar
Q.tot.death <- glm(Y_death ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                   family = "binomial", data = df2_int)
Q.tot.qol <- glm(Y_qol ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                 family = "gaussian", data = df2_int)

## 2. Predict an outcome for each subject, setting A=0 and A=1
# prepare data sets used to predict the outcome under the counterfactual
# scenarios setting A=0 and A=1
data.A1 <- data.A0 <- df2_int
data.A1$A0_ace <- 1
data.A0$A0_ace <- 0

# predict values
Y1.death.pred <- predict(Q.tot.death, newdata = data.A1, type = "response")
Y0.death.pred <- predict(Q.tot.death, newdata = data.A0, type = "response")

Y1.qol.pred <- predict(Q.tot.qol, newdata = data.A1, type = "response")
Y0.qol.pred <- predict(Q.tot.qol, newdata = data.A0, type = "response")

## 3. Plug the predicted outcome in the gformula and use the sample mean
##    to estimate the ATE
ATE.death.gcomp <- mean(Y1.death.pred - Y0.death.pred)
ATE.death.gcomp
# [1] 0.08270821

ATE.qol.gcomp <- mean(Y1.qol.pred - Y0.qol.pred)
ATE.qol.gcomp
# [1] -8.360691
```

A 95% confidence interval can be estimated applying a bootstrap procedure. An example is given in the following code.

```
set.seed(1234)
B <- 2000
bootstrap.estimates <- data.frame(matrix(NA, nrow = B, ncol = 2))
colnames(bootstrap.estimates) <- c("boot.death.est", "boot.qol.est")
for (b in 1:B){
  # sample the indices 1 to n with replacement
  bootIndices <- sample(1:nrow(df2_int), replace=T)
  bootData <- df2_int[bootIndices,]
```

```
  if ( round(b/100, 0) == b/100 ) print(paste0("bootstrap number ",b))

  Q.tot.death <- glm(Y_death ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                     family = "binomial", data = bootData)
  Q.tot.qol <- glm(Y_qol ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                   family = "gaussian", data = bootData)

  boot.A.1 <- boot.A.0 <- bootData
  boot.A.1$A0_ace <- 1
  boot.A.0$A0_ace <- 0

  Y1.death.boot <- predict(Q.tot.death, newdata = boot.A.1, type = "response")
  Y0.death.boot <- predict(Q.tot.death, newdata = boot.A.0, type = "response")

  Y1.qol.boot <- predict(Q.tot.qol, newdata = boot.A.1, type = "response")
  Y0.qol.boot <- predict(Q.tot.qol, newdata = boot.A.0, type = "response")

  bootstrap.estimates[b,"boot.death.est"] <- mean(Y1.death.boot - Y0.death.boot)
  bootstrap.estimates[b,"boot.qol.est"] <- mean(Y1.qol.boot - Y0.qol.boot)
}

IC95.ATE.death <- c(ATE.death.gcomp -
                      qnorm(0.975)*sd(bootstrap.estimates[,"boot.death.est"]),
                    ATE.death.gcomp +
                      qnorm(0.975)*sd(bootstrap.estimates[,"boot.death.est"]) )
IC95.ATE.death
# [1] 0.05571017 0.10970624

IC95.ATE.qol <- c(ATE.qol.gcomp -
                    qnorm(0.975)*sd(bootstrap.estimates[,"boot.qol.est"]),
                  ATE.qol.gcomp +
                    qnorm(0.975)*sd(bootstrap.estimates[,"boot.qol.est"]) )
IC95.ATE.qol
# [1] -9.156051 -7.565331
```

6.2 Estimation of Controlled Direct Effects (CDE)

The controlled direct effect $\Psi^{\text{CDE}_m} = \mathbb{E}(Y_{A=1,M=m}) - \mathbb{E}(Y_{A=0,M=m})$ is the difference between the mean outcome had the whole population been exposed to ACE (setting $A = 1$), compared to the mean outcome had the whole population been unexposed (setting $A = 0$), while keeping the mediator equal to a constant given value ($M = m$) in both scenarios.

The g-formula for a CDE ($\mathbb{E}(Y_{A=a',M=m})$) is more complex than for the average total effect, and the simple substitution approach described previously is less convenient to apply:

$\mathbb{E}(Y_{A=a',M=m}) = \sum_{l(0),l(1)} \left[\mathbb{E}\left(Y \mid m, l(1), a', l(0)\right) \times P(\, L(1) = l(1) | a', l(0)\,)\right] \times P\left(L(0) = l(0)\right)$

In our simple example with a binary exposure A, a binary mediator M and a binary intermediate confounder $L(1)$, it is still possible to apply the substitution approach (corresponding to a non-parametric g-computation estimation) by estimating the following components of the g-formula:

- $\overline{Q}_Y(A, L(1), M) = \mathbb{E}\left(Y \mid L(0), A, L(1), M\right)$,
- and $\overline{Q}_{L(1)}(A) = P\left(L(1) = 1\right) \mid A, l(0))$

We can then generate predicted outcomes from these 3 models for each subject in the data set, and obtain a *non-parametric maximum likelihood estimator (NPMLE)* of the CDE using the empirical mean:

$$\begin{aligned}\Psi^{\text{CDE}_m}_{\text{NPMLE}} = \tfrac{1}{n}\sum \quad & \left[\hat{\overline{Q}}_Y(A=1, L(1)=1, M=m) \times \hat{\overline{Q}}_{L(1)}(A=1) + \hat{\overline{Q}}_Y(A=1, L(1)=0, M=m) \times (1 - \hat{\overline{Q}}_{L(1)}(A=1))\right] \\ & - \left[\hat{\overline{Q}}_Y(A=0, L(1)=1, M=m) \times \hat{\overline{Q}}_{L(1)}(A=0) + \hat{\overline{Q}}_Y(A=0, L(1)=0, M=m) \times (1 - \hat{\overline{Q}}_{L(1)}(A=0))\right]\end{aligned}$$

However NPMLE is tedious with high-dimensional intermediate confounders $L(1)$ or if mediators is repeated over time. In that case, parametric g-computation using a Monte Carlo algorithm, or g-computation by iterative conditional expectation are easier to apply.

Below, we describe three g-computation procedures for the estimation of a CDE:

- parametric g-computation, using Monte Carlo simulation
- g-computation by iterative conditional expectation
- sequential g-estimator

6.2.1 Parametric g-computation

Parametric g-computation by Monte Carlo simulation have been described by Robins (Robins 1986), Taubman *et al.* (Taubman et al. 2009), or Daniel *et al.* (Daniel et al. 2013).

1. Fit a parametric model to estimate the density of the intermediate confounder $L(1)$ conditional on its parents. If $L(1)$ is a set of several variables, it is necessary to fit a model for each variable conditional on its parents.

$$Q_{L(1)}(A) = P(L(1) = 1 \mid L(0), A) \tag{6.2}$$

2. Fit a model of the outcome Y conditional on its parents:

$$\overline{Q}_Y(A, L(1), M) = \mathbb{E}\left(Y \mid L(0), A, L(1), M\right) \tag{6.3}$$

3. Simulate individual values of $L(1)_a$ using the estimated density $\hat{Q}_{L(1)}(A = a)$ under the counterfactual scenarios setting $A = 0$ or $A = 1$

4. Estimate mean values of the outcome under the counterfactual scenarios setting $A = 0$ (or $A = 1$), $L(1) = l(1)_{A=0}$ (or $L(1) = l(1)_{A=1}$) and $M = m$, using $\hat{\overline{Q}}_Y(A = a, L(1) = l(1)_a, M = m)$

5. Estimate the controlled direct effect Ψ_{CDE_m} by the sample mean:

$$\hat{\Psi}^{\text{CDE}_m}_{\text{param.gcomp}} = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{\overline{Q}}_Y(A = 1, L(1) = l(1)_{A=1}, M = m)_i - \hat{\overline{Q}}_Y(A = 0, L(1) = l(1)_{A=0}, M = m)_i\right] \tag{6.4}$$

For continuous outcomes, $\overline{Q}_Y(A, L(1), M)$ functions can be estimated using linear regressions. For binary outcomes, they can be estimated using logistic regressions.

```r
## 1. Fit parametric models to estimate the density of intermediate confounders,
##    conditional on the parents of the intermediate confounders
L1.model <- glm(L1 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                family = "binomial", data = df2_int)

## 2. Fit parametric models for the outcome conditional on past
Y.death.model <- glm(Y_death ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1 +
                                 M_smoking + A0_ace:M_smoking,
                      family = "binomial", data = df2_int)
Y.qol.model <- glm(Y_qol ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1 +
                             M_smoking + A0_ace:M_smoking,
                    family = "gaussian", data = df2_int)

## 3. Simulate individual L1 values under the counterfactual scenarios setting A0=0 or A0=1
set.seed(54321)
data.A0  <- data.A1 <- df2_int
data.A0$A0_ace <- 0
data.A1$A0_ace <- 1
p.L1.A0 <- predict(L1.model, newdata = data.A0, type="response")
p.L1.A1 <- predict(L1.model, newdata = data.A1, type="response")
sim.L1.A0 <- rbinom(n = nrow(df2_int), size = 1, prob = p.L1.A0)
sim.L1.A1 <- rbinom(n = nrow(df2_int), size = 1, prob = p.L1.A1)

## 4. Estimate mean outcomes under the counterfactual scenarios setting different
##    levels of exposures for A and M:
```

```r
##    {A=0, M=0} or {A=1, M=0} or {A=0, M=1} or {A=1, M=1}

data.A0.M0 <- data.A0.M1 <- data.A0
data.A1.M0 <- data.A1.M1 <- data.A1

# L1 variable is replaced by the simulated values in step 3)
data.A0.M0$L1 <- sim.L1.A0
data.A0.M1$L1 <- sim.L1.A0
data.A1.M0$L1 <- sim.L1.A1
data.A1.M1$L1 <- sim.L1.A1

# set M to 0 or 1
data.A0.M0$M_smoking <- 0
data.A0.M1$M_smoking <- 1
data.A1.M0$M_smoking <- 0
data.A1.M1$M_smoking <- 1

# predict the probability of death
p.death.A0.M0 <- predict(Y.death.model, newdata = data.A0.M0, type="response")
p.death.A1.M0 <- predict(Y.death.model, newdata = data.A1.M0, type="response")
p.death.A0.M1 <- predict(Y.death.model, newdata = data.A0.M1, type="response")
p.death.A1.M1 <- predict(Y.death.model, newdata = data.A1.M1, type="response")

# predict the mean value of QoL
m.qol.A0.M0 <- predict(Y.qol.model, newdata = data.A0.M0, type="response")
m.qol.A1.M0 <- predict(Y.qol.model, newdata = data.A1.M0, type="response")
m.qol.A0.M1 <- predict(Y.qol.model, newdata = data.A0.M1, type="response")
m.qol.A1.M1 <- predict(Y.qol.model, newdata = data.A1.M1, type="response")

## 5. Estimate the CDE
# CDE setting M=0
CDE.death.m0.gcomp.param <- mean(p.death.A1.M0) - mean(p.death.A0.M0)
CDE.death.m0.gcomp.param
# [1] 0.06289087

CDE.qol.m0.gcomp.param <- mean(m.qol.A1.M0) - mean(m.qol.A0.M0)
CDE.qol.m0.gcomp.param
# [1] -4.838654

# CDE setting M=1
CDE.death.m1.gcomp.param <- mean(p.death.A1.M1) - mean(p.death.A0.M1)
CDE.death.m1.gcomp.param
# [1] 0.08751016

CDE.qol.m1.gcomp.param <- mean(m.qol.A1.M1) - mean(m.qol.A0.M1)
```

```r
CDE.qol.m1.gcomp.param
# [1] -10.35059
```

6.2.2 G-computation by iterative conditional expectation

The following steps describe the implementation of the g-computation estimator by iterative conditional expectation for the component $\mathbb{E}(Y_{A=a',M=m})$ used in the definition of CDE $\Psi^{\text{CDE}_m} = \mathbb{E}(Y_{A=1,M=m}) - \mathbb{E}(Y_{A=0,M=m})$. Interestingly, there is no need to estimate or simulate $L(1)$ density with this method.

1. Fit a logistic or a linear regression of the final outcome, conditional on the exposure A, the mediator M and all the parents of Y preceeding M, to estimate $\overline{Q}_Y = \mathbb{E}(Y \mid L(0), A, L(1), M)$;

2. Use this estimate to predict an outcome for each subject $\hat{\overline{Q}}_Y(M = m)_i$, by evaluating the regression fit $\overline{Q}_Y$ at the chosen value for the mediator $M = m$;

3. Fit a quasibinomial or a linear regression of the predicted values $\hat{\overline{Q}}_Y(M = m)_i$ conditional on the exposure A and baseline confounders $L(0)$ to estimate $\overline{Q}_{L(1)} = \mathbb{E}\left(\hat{\overline{Q}}_Y(M = m) \middle| L(0), A\right)$;

4. Use this estimate to predict the outcome $\hat{\overline{Q}}_{L(1)}(A = a')_i$ for each subject, by evaluating the regression fit $\overline{Q}_{L(1)}$ at $A = a'$;

5. Use the sample mean to estimate $\Psi^{\text{CDE}_m}_{\text{gcomp}}$

$$\hat{\Psi}^{\text{CDE}_m}_{\text{gcomp}} = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{\overline{Q}}_{L(1)}(A = 1)_i - \hat{\overline{Q}}_{L(1)}(A = 0)_i\right] \tag{6.5}$$

Note that G-computation by iterative expectation is preferable if the set of intermediate confounders $L(1)$ is high-dimensional as we only need to fit 1 model by counterfactual scenario (for a whole set of $L(1)$ variables) in the procedure described below, whereas at least 1 model by $L(1)$ variable and by counterfactual scenario are needed with parametric g-computation.

```r
## 1) Regress the outcome on L0, A, L1 and M (and the A*M interaction if appropriate)
Y.death.model <- glm(Y_death ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1 +
                              M_smoking + A0_ace:M_smoking,
                     family = "binomial", data = df2_int)
Y.qol.model <- glm(Y_qol ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1 +
                              M_smoking + A0_ace:M_smoking,
```

```
                    family = "gaussian", data = df2_int)

## 2) Generate predicted values by evaluating the regression setting the mediator
##    value to M=0 or to M=1
#    (Note: it is also possible to set A=0 or A=1 to evaluate the regression at
#     exposure history of interest: {A0=1,M=0},{A0=0,M=0},{A0=1,M=1},{A0=0,M=1})
data.Mis0 <- data.Mis1 <- df2_int
data.Mis0$M_smoking <- 0
data.Mis1$M_smoking <- 1

Q.Y.death.Mis0 <- predict(Y.death.model, newdata = data.Mis0, type="response")
Q.Y.death.Mis1 <- predict(Y.death.model, newdata = data.Mis1, type="response")

Q.Y.qol.Mis0 <- predict(Y.qol.model, newdata = data.Mis0, type="response")
Q.Y.qol.Mis1 <- predict(Y.qol.model, newdata = data.Mis1, type="response")


## 3) Regress the predicted values conditional on the exposure A
##    and baseline confounders L(0)
L1.death.Mis0.model <- glm(Q.Y.death.Mis0 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                           family = "quasibinomial", data = df2_int)
L1.death.Mis1.model <- glm(Q.Y.death.Mis1 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                           family = "quasibinomial", data = df2_int)

L1.qol.Mis0.model <- glm(Q.Y.qol.Mis0 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                         family = "gaussian", data = df2_int)
L1.qol.Mis1.model <- glm(Q.Y.qol.Mis1 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                         family = "gaussian", data = df2_int)

## 4) generate predicted values by evaluating the regression at exposure
##    of interest: {A=1} & {A=0}
data.Ais0 <- data.Ais1 <- df2_int
data.Ais0$A0_ace <- 0
data.Ais1$A0_ace <- 1

Q.L1.death.Ais0.Mis0 <- predict(L1.death.Mis0.model, newdata = data.Ais0, type="respons
Q.L1.death.Ais1.Mis0 <- predict(L1.death.Mis0.model, newdata = data.Ais1, type="respons
Q.L1.death.Ais0.Mis1 <- predict(L1.death.Mis1.model, newdata = data.Ais0, type="respons
Q.L1.death.Ais1.Mis1 <- predict(L1.death.Mis1.model, newdata = data.Ais1, type="respons

Q.L1.qol.Ais0.Mis0 <- predict(L1.qol.Mis0.model, newdata = data.Ais0, type="response")
Q.L1.qol.Ais1.Mis0 <- predict(L1.qol.Mis0.model, newdata = data.Ais1, type="response")
Q.L1.qol.Ais0.Mis1 <- predict(L1.qol.Mis1.model, newdata = data.Ais0, type="response")
Q.L1.qol.Ais1.Mis1 <- predict(L1.qol.Mis1.model, newdata = data.Ais1, type="response")
```

```
## 5) Take empirical mean of final predicted outcomes to estimate CDE
# CDE setting M=0
CDE.death.m0.gcomp.ice <- mean(Q.L1.death.Ais1.Mis0) - mean(Q.L1.death.Ais0.Mis0)
CDE.death.m0.gcomp.ice
# [1] 0.06341297

CDE.qol.m0.gcomp.ice <- mean(Q.L1.qol.Ais1.Mis0) - mean(Q.L1.qol.Ais0.Mis0)
CDE.qol.m0.gcomp.ice
# [1] -4.869509

# CDE setting M=1
CDE.death.m1.gcomp.ice <- mean(Q.L1.death.Ais1.Mis1) - mean(Q.L1.death.Ais0.Mis1)
CDE.death.m1.gcomp.ice
# [1] 0.08810508

CDE.qol.m1.gcomp.ice <- mean(Q.L1.qol.Ais1.Mis1) - mean(Q.L1.qol.Ais0.Mis1)
CDE.qol.m1.gcomp.ice
# [1] -10.38144
```

6.2.3 Sequential g-estimator

For quantitative outcomes, Vansteelandt et al. (Epidemiology 20(6);2009) described a sequential g-estimator for CDE. An extension for binary outcomes in case-control studies is also described using OR.

The following 2 steps are applied:

1. Fit a regression model for the outcome conditional on the exposure A, the mediator M, baseline and intermediate confounders $L(0)$ and $L(1)$, in order to estimate the regression coefficients $\hat{\gamma}_M$ and $\hat{\gamma}_{A*M}$ (in case of $(A*M)$ interaction effect).
$$\mathbb{E}(Y \mid L(0), A, L(1), M) = \gamma_0+\gamma_A A+\gamma_M M+\psi_{A*M}(A*M)+\gamma_{L(0)}L(0)+\gamma_{L(1)}L(1) \tag{6.6}$$

2. Remove the effect of mediator on the outcome, by evaluating the residual outcome:
$$Y_{res} = Y - \hat{\gamma}_M M - \hat{\psi}_{A*M} \times A \times M \tag{6.7}$$

and regress the residual outcome on the exposure A and baseline confounders $L(0)$:
$$\mathbb{E}(Y_{res} \mid A, L(0)) = \alpha_0 + \psi_A A + \beta_{L(0)} L(0) \tag{6.8}$$

The controlled direct effect CDE_m can then be estimated by:
$$\hat{\Psi}^{\text{CDE}_m}_{\text{seq.g.est}} = \hat{\psi}_A + \hat{\psi}_{A*M} \times m \tag{6.9}$$

```
## 1) Regress the outcome on past
Y.qol.model <- glm(Y_qol ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1 +
                             M_smoking + A0_ace:M_smoking,
                   family = "gaussian", data = df2_int)

## 2) Calculate a residual outcome Y - (coef.M * M_smoking) - (coef.A0:M * A0:M)
Y.res <- (df2_int$Y_qol -
            (Y2.qol.model$coefficients["M_smoking"] *  df2_int$M_smoking) -
            (Y2.qol.model$coefficients["A0_ace:M_smoking"] * df2_int$A0_ace
              * data.inter1$M_smoking) )

## 3) Regress the residual outcome on the exposure A and baseline confounders L(0)
Y.res.model <- glm(Y.res ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                   family = "gaussian", data = df2_int)

## 4) Use coefficients estimated from the 1st and 2nd regression to estimate CDE:
CDE.qol.m0.seq <- (Y.res.model$coefficients["A0_ace"] +
                     0*Y.qol.model$coefficients["A0_ace:M_smoking"])
CDE.qol.m0.seq
# -4.869509

CDE.qol.m1.seq <- (Y.res.model$coefficients["A0_ace"] +
                     1*Y.qol.model$coefficients["A0_ace:M_smoking"])
CDE.qol.m1.seq
# -10.38144
```

6.3 Estimation of Natural Direct (NDE) and Indirect Effects (NIE)

When Natural Direct Effects and Natural Indirect Effects are identifiable (i.e. making the assumption that the confounder $L(1)$ of the $M-Y$ relationship is NOT affected by the exposure A as in Causal model 1, in Figure 3.1), estimation are based on traditional regression models as described in chapter 5.

6.4 Estimation of "Marginal" Randomized/Interventional Natural Direct (NDE) and Indirect Effects (NIE)

When we assume that the intermediate confounder $L(1)$ of the $M-Y$ relationship is affected by the exposure A (Causal model 2, Figure 3.2), an "interventional analogue" of the Average Total Effect decomposition into a Natural

Direct and Indirect Effect has been suggested.Lin et al. (2017) For these effects, counterfactual scenarios are defined by setting different values for the exposure ($A = 0$ or $A = 1$) and random draw in the distribution $G_{A=a'|L(0)}$ of the mediator (conditional on baseline counfounders $L(0)$) under the counterfactual scenario setting $A = a'$.

An Overall (Total) Effect can be defined by the contrast OE $= \mathbb{E}\left[Y_{A=1,G_{A=1|L(0)}}\right] - \mathbb{E}\left[Y_{A=0,G_{A=0|L(0)}}\right]$.

This Overall Effect can be decomposed into the sum of:

- a Marginal Randomised (or Interventional) Direct Effect: MRDE $= \mathbb{E}\left[Y_{A=1,G_{A=0|L(0)}}\right] - \mathbb{E}\left[Y_{A=0,G_{A=0|L(0)}}\right]$
- a Marginal Randomised (or Interventional) Interect Effect: MRIE $= \mathbb{E}\left[Y_{A=1,G_{A=1|L(0)}}\right] - \mathbb{E}\left[Y_{A=1,G_{A=0|L(0)}}\right]$

For this 2-way decomposition, we have to estimate 3 causal quantities: MRIE $= \mathbb{E}\left[Y_{A=1,G_{A=1|L(0)}}\right]$, $\mathbb{E}\left[Y_{A=0,G_{A=0|L(0)}}\right]$ and $\mathbb{E}\left[Y_{A=1,G_{A=0|L(0)}}\right]$.

Under the identifiability conditions, in particular:

- no unmeasured exposure-outcome confounding
- no unmeasured mediator-outcome confounding
- and exposure-mediator confounding

the quantity of $\mathbb{E}\left[Y_{a,G_{a'|L(0)}}\right]$ can be estimated by the g-formula:

$$\begin{aligned}\mathbb{E}\left[Y_{a,G_{a'|L(0)}}\right] = \sum_{l(0),l(1),m} &\mathbb{E}\left(Y \mid m, l(1), A = a, l(0),\right) \times P[L(1) = l(1) \mid a, l(0)] \\ &\times P[M = m \mid a', l(0)] \times P(L(0) = l(0))\end{aligned}$$

These causal effects can be estimated by g-computation, IPTW, or TMLE. G-computation approaches are described below.

6.4.1 Parametric g-computation

The estimation using parametric g-computation is described in (Lin et al. 2017). The approach is described as an adaptation of the parametric g-computation presented for controlled direct effects, in order to estimate causal quantities $\mathbb{E}(Y_{a,G_{a'|L(0)}})$ corresponding to a counterfactual scenario where the exposures is set to $A = a$ for all individuals and M is a random draw from the distribution $G_{a'|L(0)}$ of the mediator (conditional on $L(0)$) had the exposure been set to $A = a'$.

Estimation of $\mathbb{E}(Y_{a,G_{a'|L(0)}})$ relies on the following steps:

1. Fit parametric models for the time-varying confounders $L(1)$, the mediator M and the outcome Y given the measured past;
2. Estimate the joint distribution of time-varying confounders ($L(1)_{A=1}$ and $L(1)_{A=0}$) and of the mediator ($M_{G_{A=0}}$ and $M_{G_{A=1}}$) under the counterfactual scenarios setting $A = 1$ or $A = 0$;
3. Simulate the outcomes $Y_{A=0,G_{A=0}}$, $Y_{A=1,G_{A=1}}$ and $Y_{A=1,G_{A=0}}$ in order to compute the randomized natural direct and indirect effects.

```
set.seed(54321)

# steps 1) to 3) will be repeated some fixed number k (for example k=25)
# we will save the k results in a matrix of k rows and 4 columns for the randomized
# direct and indirect effects on death (binary) and QoL (continuous) outcomes
est <- matrix(NA, nrow = 25, ncol = 4)
colnames(est) <- c("rNDE.death", "rNIE.death", "rNDE.qol", "rNIE.qol")

# repeat k=25 times the following steps 1) to 3)
for (k in 1:25) {
  ## 1) Fit parametric models for the time-varying confounders L(1), the mediator M
  ##    and the outcome Y
  ### 1a) fit parametric models of the confounders and mediators given the past
  L1.model <- glm(L1 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                  family = "binomial", data = df2_int)
  M.model <- glm(M_smoking ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1,
                 family = "binomial", data = df2_int)
  ### 1b) fit parametric models of the outcomes given the past
  Y.death.model <- glm(Y_death ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1 +
                         M_smoking + A0_ace:M_smoking,
                       family = "binomial", data = df2_int)
  Y.qol.model <- glm(Y_qol ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1 +
                       M_smoking + A0_ace:M_smoking,
                     family = "gaussian", data = df2_int)

  ## 2) Estimate the joint distribution of time-varying confounders and of the
  ##    mediator under the counterfactual scenarios setting A0_ace = 1 or 0

  # set the exposure A0_ace to 0 or 1 in two new counterfactual data sets
  data.A0  <- data.A1 <- df2_int
  data.A0$A0_ace <- 0
  data.A1$A0_ace <- 1
  # simulate L1 values under the counterfactual exposures A0_ace=0 or A0_ace=1
  p.L1.A0 <- predict(L1.model, newdata = data.A0, type="response")
  p.L1.A1 <- predict(L1.model, newdata = data.A1, type="response")
  sim.L1.A0 <- rbinom(n = nrow(df2_int), size = 1, prob = p.L1.A0)
```

```
sim.L1.A1 <- rbinom(n = nrow(df2_int), size = 1, prob = p.L1.A1)

# replace L(1) by their counterfactual values in the data under A=0 or A=1
data.A0.L <- data.A0
data.A1.L <- data.A1
data.A0.L$L1 <- sim.L1.A0
data.A1.L$L1 <- sim.L1.A1

# simulate M values under the counterfactual exposures A0_ace=0 or A0_ace=1
p.M.A0 <- predict(M.model, newdata = data.A0.L, type="response")
p.M.A1 <- predict(M.model, newdata = data.A1.L, type="response")
sim.M.A0 <- rbinom(n = nrow(df2_int), size = 1, prob = p.M.A0)
sim.M.A1 <- rbinom(n = nrow(df2_int), size = 1, prob = p.M.A1)
# permute the n values of the joint mediator to obtain the random distributions
# of the mediator: G_{A=0} and G_{A=1}
marg.M.A0 <- sample(sim.M.A0, replace = FALSE)
marg.M.A1 <- sample(sim.M.A1, replace = FALSE)

## 3) Simulate the outcomes Y_{A=0,G_{A=0}}
### 3a) use the previous permutation to replace the mediator
### in the counterfactual data sets for Y_{A=0,G_{A=0}}, Y_{A=1,G_{A=1}} and
### Y_{A=1,G_{A=0}}
data.A0.G0 <- data.A0.G1 <- data.A0.L
data.A1.G0 <- data.A1.G1 <- data.A1.L

data.A0.G0$M_smoking <- marg.M.A0
# data.A0.G1$M_smoking <- marg.M.A1 # note: this data set will not be useful

data.A1.G0$M_smoking <- marg.M.A0
data.A1.G1$M_smoking <- marg.M.A1

# simulate the average outcome using the models fitted at step 1)
p.death.A1.G1 <- predict(Y.death.model, newdata = data.A1.G1, type="response")
p.death.A1.G0 <- predict(Y.death.model, newdata = data.A1.G0, type="response")
p.death.A0.G0 <- predict(Y.death.model, newdata = data.A0.G0, type="response")

m.qol.A1.G1 <- predict(Y.qol.model, newdata = data.A1.G1, type="response")
m.qol.A1.G0 <- predict(Y.qol.model, newdata = data.A1.G0, type="response")
m.qol.A0.G0 <- predict(Y.qol.model, newdata = data.A0.G0, type="response")

## save the results in row k
# rNDE = E(Y_{A=1,G_{A=0}}) - E(Y_{A=0,G_{A=0}})
# rNIE = E(Y_{A=1,G_{A=1}}) - E(Y_{A=1,G_{A=0}})
est[k,"rNDE.death"]<- mean(p.death.A1.G0) - mean(p.death.A0.G0)
est[k,"rNIE.death"] <- mean(p.death.A1.G1) - mean(p.death.A1.G0)
```

```
  est[k,"rNDE.qol"] <- mean(m.qol.A1.G0) - mean(m.qol.A0.G0)
  est[k,"rNIE.qol"] <- mean(m.qol.A1.G1) - mean(m.qol.A1.G0)
}

# take empirical mean of final predicted outcomes
rNDE.death <- mean(est[,"rNDE.death"])
rNDE.death
# [1] 0.07118987
rNIE.death <- mean(est[,"rNIE.death"])
rNIE.death
# [1] 0.0110088

rNDE.qol <- mean(est[,"rNDE.qol"])
rNDE.qol
# [1] -6.649923
rNIE.qol <-  mean(est[,"rNIE.qol"])
rNIE.qol
# [1] -1.585373
```

In this example,

- the marginal "randomized" Natural Direct and Indirect effect on death are a $MRDE \approx +7.1\%$ and $MRIE \approx +1.1\%$;
- the marginal "randomized" Natural Direct and Indirect effect on quality of life are a $MRDE \approx -6.6$ and $MRIE \approx -1.6$;

95% confidence intervals can be calculated by repeating the algorithm in 500 bootstrap samples of the original data set.

6.4.2 G-computation by iterative conditional expectation

We describe below the g-computation algorithm which is used in the stremr package ("Streamlined Causal Inference for Static, Dynamic and Stochastic Regimes in Longitudinal Data").

Note that G-computation by iterative expectation is preferable if the set of intermediate confounders $L(1)$ is high-dimensional as we only need to fit 1 model by counterfactual scenario in the procedure described below (whatever the dimensionaly of the set $L(1)$), whereas at least 1 model by $L(1)$ variable and by counterfactual scenario are needed with parametric g-computation.

The following 4 steps are applied:

1. Fit a parametric model for the mediator conditional on A and $L(0)$. This model will be used to predict the probabilities $G_{A=0|L(0)} = P(M = 1|A =$

$0, L(0))$ and $G_{A=1|L(0)} = P(M=1|A=1, L(0))$ under the counterfactual scenarios setting $A=0$ and $A=1$.

2. Fit parametric models for the outcome Y given the past and generate predicted values $\bar{Q}_{L(2)}(M=0)$ and $\bar{Q}_{L(2)}(M=1)$ by evaluating the regression setting the mediator value to $M=0$ or to $M=1$.

Then calculate a weighted sum of the predicted $\bar{Q}_{L(2)}(M)$, with weights given by $G_{A=1|L(0)}$ or $G_{A=0|L(0)}$:

$$\bar{Q}_{L(2),G_{A=0|L(0)}} = \bar{Q}_{L(2)}(M=1) \times G_{A=0|L(0)} + \bar{Q}_{L(2)}(M=0) \times \left[1 - G_{A=0|L(0)}\right]$$
$$\bar{Q}_{L(2),G_{A=1|L(0)}} = \bar{Q}_{L(2)}(M=1) \times G_{A=1|L(0)} + \bar{Q}_{L(2)}(M=0) \times \left[1 - G_{A=1|L(0)}\right]$$

3. Fit parametric models for the predicted values $\bar{Q}_{L(2),G_{A=a|L(0)}}$ conditional on the exposure A and baseline confounders $L(0)$, and generate predicted values $\bar{Q}_{L(1),G_{A=0|L(0)}}(A=0)$, $\bar{Q}_{L(1),G_{A=0|L(0)}}(A=1)$ and $\bar{Q}_{L(1),G_{A=1|L(0)}}(A=1)$.

4. Estimate the marginal randomized natural direct and indirect effects, using the means of the $\bar{Q}_{L(1),G_{A=a'|L(0)}}(A=a)$ calculated at the previous step

$$\text{MRDE}_{\text{ICE.gcomp}} = \frac{1}{n}\sum_{i=1}^{n}\left[\bar{Q}_{L(1),G_{A=0|L(0)}}(A=1)\right] - \frac{1}{n}\sum_{i=1}^{n}\left[\bar{Q}_{L(1),G_{A=0|L(0)}}(A=0)\right]$$
$$\text{MRIE}_{\text{ICE.gcomp}} = \frac{1}{n}\sum_{i=1}^{n}\left[\bar{Q}_{L(1),G_{A=1|L(0)}}(A=1)\right] - \frac{1}{n}\sum_{i=1}^{n}\left[\bar{Q}_{L(1),G_{A=0|L(0)}}(A=1)\right]$$

```r
## 1) Fit a parametric model for the mediator conditional on A and L(0)
##    and generate predicted values by evaluating the regression setting the exposure
##    value to A=0 or A=1
### 1a) Fit parametric models for the mediator M, conditional on the exposure A and
###    baseline confounder Pr(M=1|A,L(0)) (but not conditional on L(1))
G.model <- glm(M_smoking ~ L0_male + L0_parent_low_educ_lv + A0_ace,
               family = "binomial", data = df2_int)

### 1b) generate predicted probabilites by evaluating the regression setting the
###     exposure value to A=0 or to A=1
# create datasets corresponding to the counterfactual scenarios setting A=0 and A=1
data.Ais0  <- data.Ais1 <- df2_int
data.Ais0$A0_ace <- 0
```

```r
data.Ais1$A0_ace <- 1

# estimate G_{A=0|L(0)} = Pr(M=1|A=0,L(0)) and G_{A=1|L(0)} = Pr(M=1|A=1,L(0))
G.Ais0.L0 <-predict(G.model, newdata = data.Ais0, type="response")
G.Ais1.L0 <-predict(G.model, newdata = data.Ais1, type="response")

## 2) Fit parametric models for the observed data for the outcome Y given the past
##    and generate predicted values by evaluating the regression setting the mediator
##    value to M=0 or to M=1
##    then calculate a weighted sum of the predicted Q.L2, with weights given by G
### 2a) fit parametric models of the outcomes given the past
Y.death.model <- glm(Y_death ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1 +
                       M_smoking + A0_ace:M_smoking,
                     family = "binomial", data = df2_int)
Y.qol.model <- glm(Y_qol ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1 +
                     M_smoking + A0_ace:M_smoking,
                   family = "gaussian", data = df2_int)

### 2b) generate predicted values by evaluating the regression setting the mediator
###     value to M=0 or to M=1
data.Mis0  <- data.Mis1 <- df2_int
data.Mis0$M_smoking <- 0
data.Mis1$M_smoking <- 1

Q.L2.death.Mis0 <- predict(Y.death.model, newdata = data.Mis0, type="response")
Q.L2.death.Mis1 <- predict(Y.death.model, newdata = data.Mis1, type="response")

Q.L2.qol.Mis0 <- predict(Y.qol.model, newdata = data.Mis0, type="response")
Q.L2.qol.Mis1 <- predict(Y.qol.model, newdata = data.Mis1, type="response")

### 2c) calculate a weighted sum of the predicted Q.L2, with weights given by the
###     predicted probabilities of the mediator G_{A=0|L(0)} or G_{A=1|L(0)}
# calculate barQ.L2_{A=0,G_{A=0|L(0)}}
Q.L2.death.A0.G0 <- Q.L2.death.Mis1 * G.Ais0.L0 + Q.L2.death.Mis0 * (1 - G.Ais0.L0)
Q.L2.qol.A0.G0 <- Q.L2.qol.Mis1 * G.Ais0.L0 + Q.L2.qol.Mis0 * (1 - G.Ais0.L0)

# calculate barQ.L2_{A=1,G_{A=0|L(0)}}
# note at this step, quantities are similar to barQ.L2_{A=0,G_{A=0|L(0)}}
Q.L2.death.A1.G0 <- Q.L2.death.Mis1 * G.Ais0.L0 + Q.L2.death.Mis0 * (1 - G.Ais0.L0)
Q.L2.qol.A1.G0 <- Q.L2.qol.Mis1 * G.Ais0.L0 + Q.L2.qol.Mis0 * (1 - G.Ais0.L0)

# calculate barQ.L2_{A=1,G_{A=1|L(0)}}
Q.L2.death.A1.G1 <- Q.L2.death.Mis1 * G.Ais1.L0 + Q.L2.death.Mis0 * (1 - G.Ais1.L0)
Q.L2.qol.A1.G1 <- Q.L2.qol.Mis1 * G.Ais1.L0 + Q.L2.qol.Mis0 * (1 - G.Ais1.L0)
```

```r
## 3) Fit parametric models for the predicted values barQ.L2 conditional on the
##    exposure A and baseline confounders L(0)
##    and generate predicted values by evaluating the regression setting the exposure
##    value to A=0 or to A=1
### 3a) Fit parametric models for the predicted values barQ.L2 conditional on the
###    exposure A and baseline confounders L(0)
L1.death.A0.G0.model <- glm(Q.L2.death.A0.G0 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                            family = "quasibinomial", data = df2_int)
L1.death.A1.G0.model <- glm(Q.L2.death.A1.G0 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                            family = "quasibinomial", data = df2_int)
L1.death.A1.G1.model <- glm(Q.L2.death.A1.G1 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                            family = "quasibinomial", data = df2_int)

L1.qol.A0.G0.model <- glm(Q.L2.qol.A0.G0 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                          family = "gaussian", data = df2_int)
L1.qol.A1.G0.model <- glm(Q.L2.qol.A1.G0 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                          family = "gaussian", data = df2_int)
L1.qol.A1.G1.model <- glm(Q.L2.qol.A1.G1 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                          family = "gaussian", data = df2_int)

### 3b) generate predicted values by evaluating the regression setting the exposure
###     value to A=0 or to A=1
Q.L1.death.A0.G0 <- predict(L1.death.A0.G0.model, newdata = data.Ais0, type="response")
Q.L1.death.A1.G0 <- predict(L1.death.A1.G0.model, newdata = data.Ais1, type="response")
Q.L1.death.A1.G1 <- predict(L1.death.A1.G1.model, newdata = data.Ais1, type="response")

Q.L1.qol.A0.G0 <- predict(L1.qol.A0.G0.model, newdata = data.Ais0, type="response")
Q.L1.qol.A1.G0 <- predict(L1.qol.A1.G0.model, newdata = data.Ais1, type="response")
Q.L1.qol.A1.G1 <- predict(L1.qol.A1.G1.model, newdata = data.Ais1, type="response")

## 4) Estimate the marginal randomized natural direct and indirect effects
### MRDE = E(Y_{A=1,G_{A=0|L(0)}}) - E(Y_{A=0,G_{A=0|L(0)}})
### MRIE = E(Y_{A=1,G_{A=1|L(0)}}) - E(Y_{A=1,G_{A=0|L(0)}})

### for deaths:
MRDE.death <- mean(Q.L1.death.A1.G0) - mean(Q.L1.death.A0.G0)
MRDE.death
# [1] 0.0714693
MRIE.death <- mean(Q.L1.death.A1.G1) - mean(Q.L1.death.A1.G0)
MRIE.death
# [1] 0.01130057

### for quality of life
MRDE.qol <- mean(Q.L1.qol.A1.G0) - mean(Q.L1.qol.A0.G0)
MRDE.qol
```

```
# [1] -6.719193
MRIE.qol <-  mean(Q.L1.qol.A1.G1) - mean(Q.L1.qol.A1.G0)
MRIE.qol
# [1] -1.624645
```

Results are close to the estimations obtained previously with parametric g-computation.

- the marginal "randomized" Natural Direct and Indirect effect on death are a $MRDE \approx +7.1\%$ and $MRIE \approx +1.1\%$;
- the marginal "randomized" Natural Direct and Indirect effect on quality of life are a $MRDE \approx -6.6$ and $MRIE \approx -1.6$;

95% confidence intervals can be calculated by bootstrap.

6.5 Using the `CMAverse` package for 2-way, 3-way and 4-way decomposition

The `CMAverse` package can be used to estimate the 2-way, 3-way and 4-way decompositions of a total effect by parametric g-computation, whether the intermediate confounder $L(1)$ of the $M-Y$ relationship is affected by the exposure A or not.

Here is an example with a continuous outcome using the `cmest` function. Note that :

- parametric g-computation is applied by specifying `model = "gformula"`. The `estimation` argument should be set to `imputation` (as the counterfactual values will be imputed).
- the presence of intermediate confounder $L(1)$ of the $M-Y$ relationship affected by the exposure A can be specified using the `postcreg` argument,
- the presence of an $A * M$ interaction effect on the outcome is indicated using the `EMint` argument,
- for the estimation of the Controlled direct effect, the fixed value set for the mediator is indicated using the `mval` argument.

The function returns the following results:

- fit of the Outcome regression $\overline{Q}_Y = \mathbb{E}(Y \mid L(0), A, L(1), M)$,
- fit of the Mediator regression $g_A = P(M = 1 \mid L(0), A, L(1))$,
- fit of the intermediate confounder regression $\overline{Q}_{L(1)} = P(L(1) = 1 \mid L(0), A)$,

- the 2-way, 3-way and 4-way decompositions.

```
library(CMAverse)

set.seed(1234)
res_gformula_Qol <- cmest(data = df2_int, #data.frame(df2_int[,c("L0_male",
                                                     #"L0_parent_low_educ_lv",
                                                     #"A0_ace")],
                                         #L1=as.factor(df2_int$L1),
                                         #df2_int[,c("M_smoking","Y_qol")]),
                         model = "gformula", # for parametric g-computation
                         outcome = "Y_qol", # outcome variable
                         exposure = "A0_ace", # exposure variable
                         mediator = "M_smoking", # mediator
                         basec = c("L0_male",
                                   "L0_parent_low_educ_lv"), # confounders
                         postc = "L1", # intermediate confounder (post-exposure)
                         EMint = TRUE, # exposures*mediator interaction
                         mreg = list("logistic"), # g(M=1|L1,A,L0)
                         yreg = "linear",# Qbar.L2 = P(Y=1|M,L1,A,L0)
                         postcreg = list("logistic"), # Qbar.L1 = P(L1=1|A,L0)
                         astar = 0,
                         a = 1,
                         mval = list(0), # do(M=0) to estimate CDE_m
                         estimation = "imputation", # if model= gformula
                         inference = "bootstrap",
                         boot.ci.type = "per", # forpercentile, other option: "bca"
                         nboot = 2) # we should use a large number of bootstrap samples
summary(res_gformula_Qol)

### 1) Estimation of Qbar.Y = P(Y=1|M,L1,A,L0) with A*M interaction,
### Outcome regression:
# Call:
#   glm(formula = Y_qol ~ A0_ace + M_smoking + A0_ace * M_smoking +
#         L0_male + L0_parent_low_educ_lv + L1, family = gaussian(),
#       data = getCall(x$reg.output$yreg)$data, weights = getCall(x$reg.output$yreg)$weights)
# Coefficients:
#                         Estimate Std. Error t value Pr(>|t|)
#   (Intercept)            74.8247     0.2133 350.823  < 2e-16 ***
#   A0_ace                 -3.7014     0.4295  -8.617  < 2e-16 ***
#   M_smoking              -8.6336     0.2331 -37.042  < 2e-16 ***
#   L0_male                -0.7280     0.2019  -3.605 0.000313 ***
#   L0_parent_low_educ_lv  -2.8828     0.2116 -13.621  < 2e-16 ***
#   L1                     -5.1668     0.2189 -23.608  < 2e-16 ***
#   A0_ace:M_smoking       -5.5119     0.6440  -8.559  < 2e-16 ***
```

```
### 2) Estimation of g(M=1|L1,A,L0), model of the mediator
### Mediator regressions:
# Call:
#   glm(formula = M_smoking ~ A0_ace + L0_male + L0_parent_low_educ_lv +
#         L1, family = binomial(), data = getCall(x$reg.output$mreg[[1L]])$data,
#       weights = getCall(x$reg.output$mreg[[1L]])$weights)
# Coefficients:
#                         Estimate Std. Error z value Pr(>|z|)
#   (Intercept)           -1.36249    0.04783 -28.488  < 2e-16 ***
#   A0_ace                 0.30994    0.06668   4.648 3.35e-06 ***
#   L0_male                0.24661    0.04369   5.644 1.66e-08 ***
#   L0_parent_low_educ_lv  0.30628    0.04650   6.587 4.50e-11 ***
#   L1                     0.86045    0.04493  19.152  < 2e-16 ***

### 3) Estimation of Qbar.L1 = P(L1=1|A,L0), model of intermediate confounder
### Regressions for mediator-outcome confounders affected by the exposure:
# Call:
#   glm(formula = L1 ~ A0_ace + L0_male + L0_parent_low_educ_lv,
#       family = binomial(), data = getCall(x$reg.output$postcreg[[1L]])$data,
#       weights = getCall(x$reg.output$postcreg[[1L]])$weights)
#
# Coefficients:
#                         Estimate Std. Error z value Pr(>|z|)
#   (Intercept)           -0.86983    0.04292 -20.267  < 2e-16 ***
#   A0_ace                 0.94354    0.06475  14.572  < 2e-16 ***
#   L0_male               -0.19827    0.04289  -4.622 3.80e-06 ***
#   L0_parent_low_educ_lv  0.32047    0.04556   7.034 2.01e-12 ***

### 4) Effect decomposition on the mean difference scale via the g-formula approach
#
# Direct counterfactual imputation estimation with
# bootstrap standard errors, percentile confidence intervals and p-values
#
#                  Estimate Std.error   95% CIL 95% CIU  P.val
#   cde           -5.863750  0.233488 -4.933234  -4.620 <2e-16 ***
#   rpnde         -7.565835  0.199867 -6.689581  -6.421 <2e-16 ***
#   rtnde         -8.463729  0.157383 -7.266085  -7.055 <2e-16 ***
#   rpnie         -1.406410  0.021876 -0.971101  -0.942 <2e-16 ***
#   rtnie         -2.304304  0.064359 -1.604682  -1.518 <2e-16 ***
#   te            -9.870139  0.135507 -8.207796  -8.026 <2e-16 ***
#   rintref       -1.702085  0.033622 -1.801518  -1.756 <2e-16 ***
#   rintmed       -0.897894  0.042484 -0.633581  -0.577 <2e-16 ***
#   cde(prop)      0.594090  0.018945  0.575575   0.601 <2e-16 ***
#   rintref(prop)  0.172448  0.007802  0.213991   0.224 <2e-16 ***
#   rintmed(prop)  0.090971  0.006479  0.070244   0.079 <2e-16 ***
```

```
#   rpnie(prop)    0.142491  0.004663  0.114737   0.121 <2e-16 ***
#   rpm            0.233462  0.011142  0.184981   0.200 <2e-16 ***
#   rint           0.263419  0.014282  0.284235   0.303 <2e-16 ***
#   rpe            0.405910  0.018945  0.398973   0.424 <2e-16 ***
#   ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# cde: controlled direct effect;
# rpnde: randomized analogue of pure natural direct effect;
# rtnde: randomized analogue of total natural direct effect;
# rpnie: randomized analogue of pure natural indirect effect;
# rtnie: randomized analogue of total natural indirect effect;
# te: total effect; rintref: randomized analogue of reference interaction;
# rintmed: randomized analogue of mediated interaction;
# cde(prop): proportion cde;
# rintref(prop): proportion rintref;
# rintmed(prop): proportion rintmed;
# rpnie(prop): proportion rpnie;
# rpm: randomized analogue of overall proportion mediated;
# rint: randomized analogue of overall proportion attributable to interaction;
# rpe: randomized analogue of overall proportion eliminated
```

Chapter 7

Inverse Probability of Treatment Weighting (IPTW)

7.1 Estimation of the Average total effect

7.1.1 IPTW for the ATE

If the average total effect (ATE) is identifiable, $\Psi_{ATE} = \mathbb{E}(Y_{A=1}) - \mathbb{E}(Y_{A=0})$ can be expressed using Inverse probability of treatment weighting (IPTW), denoting $\mathbb{P}(A = a \mid L(0)) = g(A = a \mid L(0))$:

$$\Psi_{ATE} = \mathbb{E}\left(\frac{\mathbb{I}(A=1)}{g(A=1 \mid L(0))}Y\right) - \mathbb{E}\left(\frac{\mathbb{I}(A=0)}{g(A=0 \mid L(0))}Y\right) \tag{7.1}$$

The following steps describe the implementation of the IPTW estimator

1. Estimate the treatment mechanism $g(A = 1 \mid L(0))$
2. Predict each individual's probability of being exposed to her own exposure
3. Apply weights corresponding to the inverse of the predicted probability $w_i = \frac{1}{\hat{g}(A=a_i \mid L(0)_i)}$
4. Use the empirical mean of the weighted outcome Y: $\hat{\mathbb{E}}(Y_a) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathbb{I}(A_i=a)}{\hat{g}(A=a_i \mid L(0)_i)} Y_i$

```
## 1. Estimate g
g.L <- glm(A0_ace ~ L0_male + L0_parent_low_educ_lv,
           family = "binomial", data = df2_int)

## 2. Predict each individual's probability of being exposed to her own exposure
# predict the probabilities P(A0_ace=1|L(0)) & P(A0_ace=0|L(0))
pred.g1.L <- predict(g.L, type="response")
pred.g0.L <- 1 - pred.g1.L
# the predicted probability of the observed treatment A=a_i is :
gA.L <- rep(NA, nrow(df2_int))
gA.L[df2_int$A0_ace == 1] <- pred.g1.L[df2_int$A0_ace == 1]
gA.L[df2_int$A0_ace == 0] <- pred.g0.L[df2_int$A0_ace == 0]

## 3. Apply weights corresponding to the inverse of the predicted probability
wt <- 1 / gA.L

## 4. Use the empirical mean of the weighted outcome
# point estimates:
IPTW.death <- mean(wt * as.numeric(df2_int$A0_ace == 1) * df2_int$Y_death) -
  mean(wt * as.numeric(df2_int$A0_ace == 0) * df2_int$Y_death)
IPTW.death
# [1] 0.08224947

IPTW.qol <- mean(wt * as.numeric(df2_int$A0_ace == 1) * df2_int$Y_qol) -
  mean(wt * as.numeric(df2_int$A0_ace == 0) * df2_int$Y_qol)
IPTW.qol
# [1] -8.436797
```

The ATE estimates using IPTW for death probability and mean quality of life are respectively +8.2% and -8.44.

7.1.2 Stabilized IPTW for the ATE

If the average total effect (ATE) is identifiable, Ψ_{ATE} can be estimated using a stabilized IPTW estimator:

$$\hat{\mathbb{E}}(Y_1) - \hat{\mathbb{E}}(Y_0) = \frac{\frac{1}{n}\sum_{i=1}^{n} \frac{\mathbb{I}(A_i=1)\hat{g}^*(A_i=1)}{\hat{g}(A_i=1|L(0)_i)} Y_i}{\frac{1}{n}\sum_{i=1}^{n} \frac{\mathbb{I}(A_i=1)\hat{g}^*(A_i=1)}{\hat{g}(A_i=1|L(0)_i)}} - \frac{\frac{1}{n}\sum_{i=1}^{n} \frac{\mathbb{I}(A_i=0)\hat{g}^*(A_i=0)}{\hat{g}(A_i=0|L(0)_i)} Y_i}{\frac{1}{n}\sum_{i=1}^{n} \frac{\mathbb{I}(A_i=0)\hat{g}^*(A_i=0)}{\hat{g}(A_i=0|L(0)_i)}} \tag{7.2}$$

The estimation algorithm is the same as for IPTW, but taking into account any non-null function of A ($g^*(A_i = a)$) in the denominator of the weight in step 3, and applying the stabilized estimator in step 4.

```
## 3. For example, applying g^*(A) = 1
## 4. Applying the stabilized estimator
# point estimates:
s.IPTW.death <- (mean(wt * as.numeric(df2_int$A0_ace == 1) * df2_int$Y_death) /
                   mean(wt * as.numeric(df2_int$A0_ace == 1))) -
  (mean(wt * as.numeric(df2_int$A0_ace == 0) * df2_int$Y_death) /
     mean(wt * as.numeric(df2_int$A0_ace == 0)))
s.IPTW.death
# [1] 0.08294185

s.IPTW.qol <- (mean(wt * as.numeric(df2_int$A0_ace == 1) * df2_int$Y_qol) /
                 mean(wt * as.numeric(df2_int$A0_ace == 1))) -
  (mean(wt * as.numeric(df2_int$A0_ace == 0) * df2_int$Y_qol) /
     mean(wt * as.numeric(df2_int$A0_ace == 0)))
s.IPTW.qol
# [1] -8.291992
```

The ATE estimates using stabilized IPTW for death probability and mean quality of life are respectively +8.3% and -8.29.

7.2 Estimation of the Controlled direct effect (CDE)

7.2.1 IPTW for the CDE

If the controlled direct effect (CDE) is identifiable, $\Psi^{\text{CDE}_m} = \mathbb{E}(Y_{A=1,M=m}) - \mathbb{E}(Y_{A=0,M=m})$ can be expressed by the basic Horvitz Thompson estimator (using Inverse probability of treatment weighting (IPTW)), denoting $\mathbb{P}\left(A = a \mid L(0)\right) = g(A = a \mid L(0))$ and $\mathbb{P}\left(M = m \mid L(0), A, L(1)\right) = g(M = m \mid L(0), A, L(1))$:

$$\Psi^{\text{CDE}_m} = \mathbb{E}\left[\frac{\mathbb{I}(A = 1 \cap M = m)}{g(A = 1 \mid L(0)) \times g(M = m \mid L(0), A, L(1))}Y\right] - \mathbb{E}\left[\frac{\mathbb{I}(A = 0 \cap M = m)}{g(A = 0 \mid L(0)) \times g(M = m \mid L(0), A, L(1))}Y\right] \tag{7.3}$$

The following steps describe the implementation of the IPTW estimator

1. Estimate the treatment mechanisms $g\left(A = 1 \mid L(0)\right)$ and $g\left(M = 1 \mid L(0), A, L(1)\right)$

2. Predict each individual's probability of being exposed to her own exposure

3. Apply weights corresponding to the inverse of the predicted probability $w_{A_i} = \frac{1}{\hat{g}(A=a_i|L(0)_i)}$ and $w_{M_i} = \frac{1}{\hat{g}(M=m_i|L(0)_i,A_i,L(1)_i)}$

4. Use the empirical mean of the weighted outcome Y: $\hat{\mathbb{E}}(Y_{a,m}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathbb{I}(A_i=a \cap M_i=m)}{\hat{g}(A=a_i|L(0)_i) \times \hat{g}(M=m_i|L(0)_i,A_i,L(1)_i)} Y_i$

```
## 1. Estimate gA and gM
gA.L <- glm(A0_ace ~ L0_male + L0_parent_low_educ_lv,
            family = "binomial", data = df2_int)
gM.L <- glm(M_smoking ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1,
            family = "binomial", data = df2_int)

## 2. Predict each individual's probability of being exposed to her own exposure
# predict the probabilities P(A0_ace=1|L(0)) & P(A0_ace=0|L(0))
pred.gA1.L <- predict(gA.L, type = "response")
pred.gA0.L <- 1 - pred.gA1.L
# the predicted probability of the observed treatment A_i=a is :
gAobs.L <- rep(NA, nrow(df2_int))
gAobs.L[df2_int$A0_ace == 1] <- pred.gA1.L[df2_int$A0_ace == 1]
gAobs.L[df2_int$A0_ace == 0] <- pred.gA0.L[df2_int$A0_ace == 0]

# predict the probabilities P(M=1|L(0),A,L(1)) & P(M=0|L(0),A,L(1))
pred.gM1.L <- predict(gM.L, type = "response")
pred.gM0.L <- 1 - pred.gM1.L
# the predicted probability of the observed treatment M_i=m is :
gMobs.L <- rep(NA, nrow(df2_int))
gMobs.L[df2_int$M_smoking == 1] <- pred.gM1.L[df2_int$M_smoking == 1]
gMobs.L[df2_int$M_smoking == 0] <- pred.gM0.L[df2_int$M_smoking == 0]

## 3. Apply weights corresponding to the inverse of the predicted probability
wt_A <- 1 / gAobs.L
wt_M <- 1 / gMobs.L
wt <- wt_A * wt_M

## 4. Use the empirical mean of the weighted outcome
# point estimates of CDE, setting M=0
CDE_IPTW_m0_death <- (mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                             df2_int$M_smoking == 0) *
                             df2_int$Y_death) -
                        mean(wt * as.numeric(df2_int$A0_ace==0 &
                                               df2_int$M_smoking == 0) *
                               df2_int$Y_death))
CDE_IPTW_m0_death
# [1] 0.05874684

CDE_IPTW_m0_qol <- (mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                           df2_int$M_smoking == 0) *
                           df2_int$Y_qol) -
                      mean(wt * as.numeric(df2_int$A0_ace==0 &
                                             df2_int$M_smoking == 0) *
                             df2_int$Y_qol))
```

```
CDE_IPTW_m0_qol
# [1] -5.341138

# point estimates of CDE, setting M=1
CDE_IPTW_m1_death <- (mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                             df2_int$M_smoking == 1) *
                             df2_int$Y_death) -
                        mean(wt * as.numeric(df2_int$A0_ace==0 &
                                               df2_int$M_smoking == 1) *
                               df2_int$Y_death))
CDE_IPTW_m1_death
# [1] 0.101733

CDE_IPTW_m1_qol <- (mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                           df2_int$M_smoking == 1) *
                           df2_int$Y_qol) -
                      mean(wt * as.numeric(df2_int$A0_ace==0 &
                                             df2_int$M_smoking == 1) *
                             df2_int$Y_qol))
CDE_IPTW_m1_qol
# [1] -8.185866
```

7.2.2 Stabilized IPTW for the CDE

If the controlled dired effect (CDE) is identifiable, Ψ^{CDE} can be estimated using a stabilized IPTW estimator (modified Horvitz Thompson estimator):

$$\hat{\mathbb{E}}(Y_{1,m})-\hat{\mathbb{E}}(Y_{0,m}) = \frac{\sum_{i=1}^{n} \frac{\mathbb{I}(A_i=1 \cap M_i=m)}{\hat{g}(A_i=1|L(0)_i)\times\hat{g}(M_i=m|L(0)_i,A_i,L(1)_i)} Y_i}{\sum_{i=1}^{n} \frac{\mathbb{I}(A_i=1 \cap M_i=m)}{\hat{g}(A_i=1|L(0)_i)\hat{g}(M_i=m|L(0)_i,A_i,L(1)_i)}} - \frac{\sum_{i=1}^{n} \frac{\mathbb{I}(A_i=0 \cap M_i=m)}{\hat{g}(A_i=0|L(0)_i)\times\hat{g}(M_i=m|L(0)_i,A_i,L(1)_i)} Y_i}{\sum_{i=1}^{n} \frac{\mathbb{I}(A_i=0 \cap M_i=m)}{\hat{g}(A_i=0|L(0)_i)\times\hat{g}(M_i=m|L(0)_i,A_i,L(1)_i)}} \tag{7.4}$$

The estimation algorithm is the same as for IPTW, but applying the stabilized estimator in step 4.

```
## 4. Applying the stabilized estimator
# point estimates of CDE, setting M=0:
CDE_sIPTW_m0_death <- (mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                              df2_int$M_smoking == 0) *
                              df2_int$Y_death) /
                         mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                                df2_int$M_smoking == 0))) -
                         (mean(wt * as.numeric(df2_int$A0_ace == 0 &
                                              df2_int$M_smoking == 0) *
                              df2_int$Y_death) /
                         mean(wt * as.numeric(df2_int$A0_ace == 0 &
```

```
                                                df2_int$M_smoking == 0)))
CDE_sIPTW_m0_death
# [1] 0.0601292

CDE_sIPTW_m0_qol <- (mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                              df2_int$M_smoking == 0) *
                              df2_int$Y_qol) /
                         mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                                df2_int$M_smoking == 0))) -
                         (mean(wt * as.numeric(df2_int$A0_ace == 0 &
                                              df2_int$M_smoking == 0) *
                              df2_int$Y_qol) /
                         mean(wt * as.numeric(df2_int$A0_ace == 0 &
                                                df2_int$M_smoking == 0)))
CDE_sIPTW_m0_qol
# [1] -4.966328

# point estimates of CDE, setting M=1:
CDE_sIPTW_m1_death <- (mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                              df2_int$M_smoking == 1) *
                              df2_int$Y_death) /
                         mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                                df2_int$M_smoking == 1))) -
                         (mean(wt * as.numeric(df2_int$A0_ace == 0 &
                                              df2_int$M_smoking == 1) *
                              df2_int$Y_death) /
                         mean(wt * as.numeric(df2_int$A0_ace == 0 &
                                                df2_int$M_smoking == 1)))
CDE_sIPTW_m1_death
# [1] 0.09030186

CDE_sIPTW_m1_qol <- (mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                              df2_int$M_smoking == 1) *
                              df2_int$Y_qol) /
                         mean(wt * as.numeric(df2_int$A0_ace == 1 &
                                                df2_int$M_smoking == 1))) -
                         (mean(wt * as.numeric(df2_int$A0_ace == 0 &
                                              df2_int$M_smoking == 1) *
                              df2_int$Y_qol) /
                         mean(wt * as.numeric(df2_int$A0_ace == 0 &
                                                df2_int$M_smoking == 1)))
CDE_sIPTW_m1_qol
# [1] -10.03045
```

Chapter 8

Marginal structural models

Marginal structural models (MSM) are parametric models that are used to summarize the relationship between the counterfactual outcome (Y_a or Y_{am} for example) and the exposure(s) A and mediators M. It also possible to summarize the relationship according to a subset of the baseline confounders if it is relevant for the scientific question.

To illustrate the application of MSMs, we will first use the data simulated from the Causal model 1 (with an $A * M$ interaction effect on the outcome) where the exposure A doesn't affect the counfonder $L(1)$ between the mediator and the outcome.

8.1 MSM for the Average Total Effect (ATE)

8.1.1 Expressing the ATE using coefficients of an MSM

For a continuous or binary outcome, we can use the following MSM to summarize the relationship between the counterfactual outcome (Y_a) and the exposure(s) A:

$$\mathbb{E}(Y_a) = \alpha_0 + \alpha_A a \tag{8.1}$$

The Average Total Effect ATE $= \mathbb{E}(Y_{A=1}) - \mathbb{E}(Y_{A=0})$ can then be expressed using the coefficients of this MSM (8.1):

$$\text{ATE} := (\alpha_0 + \alpha_A \times 1) - (\alpha_0 + \alpha_A \times 0) = \alpha_A$$

In this example, the coefficient α_A corresponds to the ATE.

Such a model is not very useful for a binary exposure. It would be much more useful for higher-dimensional exposures, for example with a continuous exposure, where the relationship between all the possible continuous values of the exposure $A = a$ and the corresponding outcomes Y_a is summarized (and arbitrarily simplified) by a single line and the slope coefficient α_A.

It is also possible to define MSMs adjusted for a subset V of the baseline confounders ($V \subset L(0)$). Such MSMs can be useful to estimate conditional effects. For example it is possible to define an average "conditional" total effect ATE$|L(0)$ (instead of the marginal ATE defined above), projecting the counterfactual outcomes on a parametric model such as the following:

$$\mathbb{E}(Y_a \mid L(0)) = \alpha_0 + \alpha_A a + \alpha_{\text{male}} L_{\text{male}}(0) + \alpha_{\text{low.educ.}} L_{\text{low.educ.}}(0) \tag{8.2}$$

so that ATE $\mid L(0) = \mathbb{E}(Y_{A=1} \mid L(0)) - \mathbb{E}(Y_{A=0} \mid L(0)) = \alpha_A$ using the coefficient from the MSM (8.2).

MSMs are also very useful to study interactions, or effect modification of the exposure A by a baseline confounder. For example, in order to study the average total effect according to sex, we can use the following MSM:

$$\mathbb{E}(Y_a \mid L_{\text{male}}(0)) = \alpha_0 + \alpha_A a + \alpha_{\text{male}} L_{\text{male}}(0) + \alpha_{A*L_{\text{male}}} (a \times L_{\text{male}}(0)) \tag{8.3}$$

and express the average total effect in each strata of sex using the coefficients of the MSM (8.3):

$$\{\text{ATE} \mid L_{\text{male}}(0) = 0\} := \mathbb{E}(Y_1 \mid L_{\text{male}}(0) = 0) - \mathbb{E}(Y_0 \mid L_{\text{male}}(0) = 0) = \alpha_A$$
$$\{\text{ATE} \mid L_{\text{male}}(0) = 1\} := \mathbb{E}(Y_1 \mid L_{\text{male}}(0) = 1) - \mathbb{E}(Y_0 \mid L_{\text{male}}(0) = 1) = \alpha_A + \alpha_{A*L_{\text{male}}}$$

Because MSMs are models of unobserved counterfactual outcomes, estimators of the MSM coefficients are necessary. We will describe two possible approaches : estimation by IPTW or by G-computation.

In both approaches, 95% confidence intervals can be computed by bootstrap.

8.1.2 Estimation of the MSM coefficients by IPTW

MSM coefficients can be easily estimated using an Inverse Probability of Treatment (IPTW) approach based on weighted regressions.

For example, in order to fit the MSM (8.3) described above, we can use a linear regression of the (observed) outcome Y on the exposure and sex, weigthed by individual weights w_i or sw_i:

$$\mathbb{E}\left[Y \mid L_{\text{male}}(0)\right] = \alpha_0 + \alpha_A a + \alpha_{\text{male}} L_{\text{male}}(0) + \alpha_{A*L_{\text{male}}} (a \times L_{\text{male}}(0)) \tag{8.4}$$

where $w_i = \frac{1}{P(A=a_i|L(0)=l(0)_i)}$ or $sw_i = \frac{P(A=a_i|L_{\text{male}}(0))}{P(A=a_i|L(0)=l(0)_i)}$.

As in chapter 7, the "no-unmeasured confounding" assumption is addressed by the application of weights w_i or sw_i, which balance confounders $L(0)$ relative to the exposure A.

```r
## 1. Denominator of the weight
# 1a. Estimate g(A=a_i|L(0)) (denominator of the weight)
g.A.L <- glm(A0_ace ~ L0_male + L0_parent_low_educ_lv,
           family = "binomial", data = df1_int)

# 1b. Predict each individual's probability of being exposed to her own exposure
# predict the probabilities P(A0_ace=1 & P(A0_ace=0)
pred.g1.L <- predict(g.A.L, type="response")
pred.g0.L <- 1 - pred.g1.L
# the predicted probability of the observed treatment P(A = a_i | L(0)) is :
gAi.L <- rep(NA, nrow(df1_int))
gAi.L[df1_int$A0_ace==1] <- pred.g1.L[df1_int$A0_ace==1]
gAi.L[df1_int$A0_ace==0] <- pred.g0.L[df1_int$A0_ace==0]

## 2. Numerator of the weight
# The numerator of the weight can be 1 for simple weights,
# or g(A=a_i|V) to obtain stabilized weights which put less weight to individuals
# with less observation. Stabilized weights enable a weaker positivity assumption.

# 2a. Estimate g(A=a_i | sex) (numerator of the stabilized weight)
g.A.sex <- glm(A0_ace ~ L0_male,
           family = "binomial", data = df1_int)

# 2b. Predict each individual's probability of being exposed to her own exposure
# predict the probabilities P(A0_ace=1 | sex) & P(A0_ace=0 | sex)
pred.g1.sex <- predict(g.A.sex, type="response")
pred.g0.sex <- 1 - pred.g1.sex
# the predicted probability of the observed treatment P(A = a_i | sex) is :
gAi.sex <- rep(NA, nrow(df1_int))
gAi.sex[df1_int$A0_ace==1] <- pred.g1.sex[df1_int$A0_ace==1]
gAi.sex[df1_int$A0_ace==0] <- pred.g0.sex[df1_int$A0_ace==0]

## 3. Define individual weights:
# We can use simple weights w = 1 / g(A=a_i | L(0))
w <- 1 / gAi.L
# Or alternatively, we can use stabilized weights :
# sw = g(A=a_i | sex) / g(A=a_i | L(0))
sw <- gAi.sex / gAi.L

# we can see that stabilized weights have less extreme values
```

```
par(mfcol = c(1,2))
boxplot(w ~ df1_int$A0_ace)
boxplot(sw ~ df1_int$A0_ace)

## 4. Estimate coefficients of the MSM using a weighted regression E(Y | A, sex)
# a GLM with gaussian family can be applied to estimate risk differences
# (for relative risk or rate ratios, we can apply a Poisson family;
#  for OR, we can apply a binomial family)
msm1 <- glm(Y_death ~ A0_ace + L0_male + A0_ace*L0_male,
           weights = w, # applying the simple weight
           family = "gaussian",
           data = df1_int)
coef(msm1)
# (Intercept)         A0_ace        L0_male A0_ace:L0_male
# 0.17573472     0.03589627     0.04598911     0.04136896

msm2 <- glm(Y_death ~ A0_ace + L0_male + A0_ace*L0_male,
            weights = sw, # applying the stabilized weight
            family = "gaussian",
            data = df1_int)
coef(msm2)
# (Intercept)         A0_ace        L0_male A0_ace:L0_male
# 0.17573472     0.03589627     0.04598911     0.04136896

## 5. Estimate the ATE stratified by sex
# According to the MSM1 (with simple weights)
ATE.msm1.male0 <- coef(msm1)["A0_ace"]
# 0.03589627
ATE.msm1.male1 <- coef(msm1)["A0_ace"] + coef(msm1)["A0_ace:L0_male"]
# 0.07726522

# According to the MSM2 (with stabilized weights)
ATE.msm2.male0 <- coef(msm2)["A0_ace"]
# 0.03589627
ATE.msm2.male1 <- coef(msm2)["A0_ace"] + coef(msm2)["A0_ace:L0_male"]
# 0.07726522
# The results are the same because there is no violation of the positivity assumption
# In case of positivity violation, stabilized weights would give more accurate estimat
```

The ATE estimates of death probability using an MSM estimated by IPTW are respectively +3.6% in women and +7.7% in men.

95% confidence intervals can be calculated by bootstrap.

Note: Using the true data generating model used to simulate the illustrative datasets, the "true" value of the ATE stratified by sex can be calculated:

- the "true" $(ATE \mid L_{\text{male}}(0) = 0) = 0.0688$ in women,
- the "true" $(ATE \mid L_{\text{male}}(0) = 1) = 0.0703$ in men.

8.1.3 Estimation of the MSM coefficients by G-computation (imputation)

We can also use a G-computation (sometimes described as an imputation) approach to estimate the coefficients of an MSM.

The following steps can be applied:

1. Fit a (logistic or a linear) regression to estimate $\overline{Q} = \mathbb{E}(Y \mid A, L(0))$
2. Use this estimate to predict an outcome for each subject under the counterfactual scenarios $\hat{\overline{Q}}(A = 0)_i$ and $\hat{\overline{Q}}(A = 1)_i$, by evaluating the regression fit $\overline{Q}$ at $A = 0$ and $A = 1$ respectively
3. Duplicate the initial dataset in a single long dataset in which:

- the first half of the long dataset corresponds to the first counterfactual scenario with $A = 0$ for all individuals and an additional column for the predicted counterfactual outcome $\hat{\overline{Q}}(A = 0)$;
- the second half of the long dataset corresponds to the second counterfactual scenario with $A = 1$ for all individuals and $\hat{\overline{Q}}(A = 1)$ for the predicted counterfactual column.

4. Fit the MSM $\mathbb{E}\left[Y_a \mid L_{\text{male}}(0)\right]$ using the long dataset.

```
## 1. Estimate Qbar
Q.tot.death <- glm(Y_death ~ A0_ace + L0_male + L0_parent_low_educ_lv,
                   family = "gaussian", data = df1_int)
# The final result would be sligthly different if we apply a binomial family.

## 2. Predict an outcome for each subject, in 2 counterfactual scenarios
##    setting A=0 and A=1
# prepare data sets used to predict the outcome under the counterfactual
# scenarios setting A=0 and A=1
data.A1 <- data.A0 <- df1_int
data.A1$A0_ace <- 1
data.A0$A0_ace <- 0

# predict values under the same name in the corresponding counterfactual dataset
data.A1$Ya.death.pred <- predict(Q.tot.death, newdata = data.A1, type = "response")
```

```r
data.A0$Ya.death.pred <- predict(Q.tot.death, newdata = data.A0, type = "response")

## 3. Append both counterfactual datasets in a single long dataset
# (the number of row is twice the initial number of row because there are 2
# counterfactual scenarios)
data.2scenarios <- rbind(data.A0, data.A1)

## 4. fit the MSM: E(Y_a|sex)
# a GLM with gaussian family can be applied to estimate risk differences
MSM.ATE.gcomp <- glm(Ya.death.pred ~ A0_ace + L0_male + A0_ace:L0_male,
                     family = "gaussian",
                     data = data.2scenarios)
coef(MSM.ATE.gcomp)
#  (Intercept)         A0_ace        L0_male A0_ace:L0_male
# 1.743994e-01   7.720726e-02   4.874750e-02   4.802530e-16

## 5. Estimate the ATE stratified by sex
# According to MSM.ATE.gcomp
ATE.MSM.gcomp.male0 <- coef(MSM.ATE.gcomp)["A0_ace"]
# 0.07720726
ATE.MSM.gcomp.male1 <- (coef(MSM.ATE.gcomp)["A0_ace"] +
                          coef(MSM.ATE.gcomp)["A0_ace:L0_male"])
# 0.07720726
# The results are the same in both strata, because in the first Qbar model,
# we did not include any (A * sex) interaction term

# Applying a binomial family for the first Qbar model would result in two
# different values of the ATE stratified by sex.
# => 0.06880798 in the L0_male = 0 strata
# => 0.08053575 in the L0_male = 1 strata

# Comments: For the estimation of the first Qbar model, applying a gaussian family
# (additive model) with no interaction terms implies the presence of some interaction
# terms in a multiplicative model (such as a glm with a binomial family).
# On the contrary, applying a binomial family (multiplicative model) with no
# interaction terms implies some interaction terms in an additive model (such as
# a glm with a gaussian family)
```

8.2 MSM for Controlled Direct Effects

8.2.1 Expressing the CDE using coefficients of an MSM

The controlled direct effect is defined as $\text{CDE}_m = \mathbb{E}(Y_{am}) - \mathbb{E}(Y_{a^*m})$.

Using the following MSM

$$\mathbb{E}(Y_{am}) = \alpha_0 + \alpha_A a + \alpha_M m + \alpha_{A*M} a \times m \tag{8.5}$$

the controlled direct effect (keeping the mediator constant to the value $M = m$) can be expressed using the coefficients of the MSM (8.5):

$$\text{CDE}_m = (\alpha_0 + \alpha_A a + \alpha_M m + \alpha_{A*M} a \times m) - (\alpha_0 + \alpha_A a^* + \alpha_M m + \alpha_{A*M} a^* \times m)$$
$$\text{CDE}_m = \alpha_A (a - a^*) + \alpha_{A*M} \times (a - a^*) \times m$$

For a binary exposure A, we have $\text{CDE}_m = \alpha_A + \alpha_{A*M} \times m$.

8.2.2 Estimation of the MSM coefficients by IPTW

MSM coefficients can be easily estimated using an Inverse Probability of Treatment (IPTW) approach based on weighted regressions.

In order to fit the MSM (8.5), we can use a linear regression of the (observed) outcome Y on the exposure and mediator, weighted by individual stabilized weights sw_i ((VanderWeele 2009)):

$$\mathbb{E}(Y \mid A, M) = \alpha_0 + \alpha_A a + \alpha_M m + \alpha_{A*M} a \times m \tag{8.6}$$

where sw_i is the product of two weights $sw_i = sw_{A,i} \times sw_{M,i}$,

$sw_{A,i} = \frac{P(A=a_i)}{P(A=a_i|L(0)=l(0)_i)}$ and $sw_{M,i} = \frac{P(M=m_i|A=a_i)}{P(M=m_i|A=a_i, L(0)=l(0)_i), L(1)=l(1)_i}$.

The "no-unmeasured confounding" assumption is addressed by the application of weights sw_i, which balance confounders $L(0)$ relative to the exposure-outcome $A - Y$ relationship, and balance the set of confounders $\{L(0), A, L(1)\}$ relative to the mediator-outcome $M - Y$ relationship.

Importantly, this approach for the estimation of the controlled direct effect CDE_m by IPTW is also valid if the exposure A affects the intermediate confounder $L(1)$ (as with the Causal model 2).

```r
### MSM of CDE, estimated by IPTW ---------------------------------------------
## 1. Stabilized weight for the exposure sw_{A,i}
# 1a. Estimate g(A=a_i|L(0)) (denominator of the weight)
g.A.L <- glm(A0_ace ~ L0_male + L0_parent_low_educ_lv,
             family = "binomial", data = df1_int)
# 1b. Predict each individual's probability of being exposed to her own exposure
# the predicted probability of the observed treatment g(A = a_i | L(0)) is :
gAi.L <- rep(NA, nrow(df1_int))
gAi.L[df1_int$A0_ace==1] <- predict(g.A.L, type="response")[df1_int$A0_ace==1]
gAi.L[df1_int$A0_ace==0] <- (1 - predict(g.A.L, type="response"))[df1_int$A0_ace==0]
```

```
# 1c. Estimate g(A=a_i) (numerator of the weight)
g.A <- glm(A0_ace ~ 1, family = "binomial", data = df1_int)
# 1d. Predict each individual's probability of being exposed to her own exposure
# the predicted probability of the observed treatment g(A = a_i) is :
gAi <- rep(NA, nrow(df1_int))
gAi[df1_int$A0_ace==1] <- predict(g.A, type="response")[df1_int$A0_ace==1]
gAi[df1_int$A0_ace==0] <- (1 - predict(g.A, type="response"))[df1_int$A0_ace==0]

# 1e. Calculate the weight for the exposure A: sw_{A,i}
sw_Ai <- gAi / gAi.L

## 2. Stabilized weight for the mediator sw_{M,i}
# 2a. Estimate g(M=m_i|L(0),A,L(1)) (denominator of the weight)
g.M.L <- glm(M_smoking ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1,
             family = "binomial", data = df1_int)
# 2b. Predict each individual's probability of being exposed to her own exposure
# the predicted probability of the observed treatment g(A = a_i | L(0)) is :
gMi.L <- rep(NA, nrow(df1_int))
gMi.L[df1_int$M_smoking==1] <- predict(g.M.L, type="response")[df1_int$M_smoking==1]
gMi.L[df1_int$M_smoking==0] <- (1 - predict(g.M.L, type="response"))[df1_int$M_smoking

# 2c. Estimate g(M=m_i|A) (numerator of the weight)
g.M.A <- glm(M_smoking ~ A0_ace, family = "binomial", data = df1_int)
# 2d. Predict each individual's probability of being exposed to her own exposure
# the predicted probability of the observed treatment g(M = m_i|A) is :
gMi.A <- rep(NA, nrow(df1_int))
gMi.A[df1_int$M_smoking==1] <- predict(g.M.A, type="response")[df1_int$M_smoking==1]
gMi.A[df1_int$M_smoking==0] <- (1 - predict(g.M.A, type="response"))[df1_int$M_smoking
# 2e. Calculate the weight for the mediator M: sw_{M,i}
sw_Mi <- gMi.A / gMi.L

## 3. Define the individual stabilized weight for the CDE_m
sw_cde <- sw_Ai * sw_Mi

## 4. Estimate coefficients of the MSM using a weighted regression E(Y | A, sex)
# a GLM with gaussian family can be applied to estimate risk differences
msm_cde <- glm(Y_death ~ A0_ace + M_smoking + A0_ace*M_smoking,
               weights = sw_cde,
               family = "gaussian",
               data = df1_int)
coef(msm_cde)
# (Intercept)           A0_ace        M_smoking A0_ace:M_smoking
#  0.17891689       0.06798282       0.06729724      -0.00495314

## 5. Estimate CDE for m=0 and for m=1 using the MSM's coefficients
```

```
CDE_mis0 <- coef(msm_cde)["A0_ace"]
# 0.06798282
CDE_mis1 <- coef(msm_cde)["A0_ace"] + coef(msm_cde)["A0_ace:M_smoking"]
# 0.06302968
```

In this example, our estimates of the controlled direct effects are $CDE_{M=0} = 6.8\%$ and $CDE_{M=1} = 6.3\%$. Confidence intervals can be calculated by bootstrap.

8.2.3 Estimation of the MSM coefficients by G-computation

As for the ATE, we can use G-computation to estimate the coefficients of the MSM to estimate Controlled Direct Effects.

Note that the algorithm described below is correct only if the exposure A doesn't affect the intermediate confounders $L(1)$ of the $M \rightarrow Y$ relationship (such as the data simulated from the Causal model 1 in our examples). In that case, the following steps can be applied:

1. Fit a (logistic or a linear) regression to estimate $\overline{Q} = \mathbb{E}(Y \mid A, M, L(0), L(1))$
2. Use this estimate to predict an outcome for each subject under the counterfactual scenarios $\hat{\overline{Q}}(A = 0, M = m, L(0), L(1))_i$ and $\hat{\overline{Q}}(A = 1, M = m, L(0), L(1))_i$, by evaluating the regression fit $\overline{Q}$ at $(A = 0, M = m)$ and $(A = 1, M = m)$ respectively. If we want to set the level of the mediator to $M = 0$ and $M = 1$, this would give 4 counterfactual scenarios do$(A = 0, M = 0)$, do$(A = 1, M = 0)$, do$(A = 0, M = 1)$ and do$(A = 1, M = 1)$.
3. Duplicate the initial dataset for each scenario in a single long dataset in which:

- the 1st part of the long dataset corresponds to the first counterfactual scenario with $(A = 0, M = 0)$ for all individuals and an additional column for the predicted counterfactual outcome $\mathbb{E}(Y_{A=0,M=0} \mid L(0), L(1)) = \hat{\overline{Q}}(A = 0, M = 0, L(0), L(1))$;
- the 2d part of the long dataset corresponds to the second counterfactual scenario with $(A = 1, M = 0)$ for all individuals and an additional column for the predicted counterfactual outcome $\mathbb{E}(Y_{A=1,M=0} \mid L(0), L(1)) = \hat{\overline{Q}}(A = 1, M = 0, L(0), L(1))$;
- the 3d part of the long dataset corresponds to the second counterfactual scenario with $(A = 0, M = 1)$ for all individuals and an additional column for the predicted counterfactual outcome $\mathbb{E}(Y_{A=0,M=1} \mid L(0), L(1)) = \hat{\overline{Q}}(A = 0, M = 1, L(0), L(1))$;

- the 4th part of the long dataset corresponds to the second counterfactual scenario with $(A = 1, M = 1)$ for all individuals and an additional column for the predicted counterfactual outcome $\mathbb{E}(Y_{A=1,M=1} \mid L(0), L(1)) = \hat{\overline{Q}}(A = 1, M = 1, L(0), L(1))$;

4. Fit the MSM $\mathbb{E}(Y_{am}) = \alpha_0 + \alpha_A a + \alpha_M m + \alpha_{A*M} a \times m$ using the long dataset.

```
### MSM of CDE, estimated by G-computation -------------------------------------
## 1. Estimate Qbar(A,M,L0,L1)
Q.cde.death <- glm(Y_death ~ A0_ace + M_smoking + A0_ace:M_smoking + L0_male +
                     L0_parent_low_educ_lv + L1,
                   family = "gaussian", data = df1_int)
# The final result would be sligthly different if we applied a binomial family
# The Gaussian family corresponds to the true generating model in this example.

## 2. Predict an outcome for each subject, in each counterfactual scenario
# Prepare data sets that will be used to predict the outcome under the counterfactual
# 4 counterfactual scenarios setting (A=0,M=0), (A=1,M=0), (A=0,M=1) and (A=1,M=1)
data.A0M0 <- data.A1M0 <- data.A0M1 <- data.A1M1 <- df1_int
data.A0M0$A0_ace <- 0
data.A0M0$M_smoking <- 0

data.A1M0$A0_ace <- 1
data.A1M0$M_smoking <- 0

data.A0M1$A0_ace <- 0
data.A0M1$M_smoking <- 1

data.A1M1$A0_ace <- 1
data.A1M1$M_smoking <- 1

# predict values under the same name in the corresponding counterfactual dataset
data.A0M0$Yam.death.pred <- predict(Q.cde.death, newdata = data.A0M0, type = "response"
data.A1M0$Yam.death.pred <- predict(Q.cde.death, newdata = data.A1M0, type = "response"
data.A0M1$Yam.death.pred <- predict(Q.cde.death, newdata = data.A0M1, type = "response"
data.A1M1$Yam.death.pred <- predict(Q.cde.death, newdata = data.A1M1, type = "response"

## 3. Append the 4 counterfactual datasets in a single long dataset
# number of row is 4 times the initial value (we have 4 counterfactual scenarios)
data.4scenarios <- rbind(data.A0M0, data.A1M0,data.A0M1,data.A1M1)

## 4. fit the MSM: E(Y_am) = alpha_0 + alpha_A a + alpha_M m + alpha_AM a:m
MSM.CDE.gcomp <- glm(Yam.death.pred ~ A0_ace +  M_smoking + A0_ace:M_smoking,
                     family = "gaussian", # gaussian family for risk differences
```

```
                     data = data.4scenarios)
coef(MSM.CDE.gcomp)
# (Intercept)           A0_ace        M_smoking A0_ace:M_smoking
#  0.17968603       0.06000138       0.06757214       0.01918153

## 5. Estimate the CDE(M=m)
# CDE(M=0) = E(Y_{A=1,M=0}) - E(Y_{A=0,M=0})
CDE_mis0_gcomp <- coef(MSM.CDE.gcomp)["A0_ace"]
# 0.06000138

# CDE(M=1) = E(Y_{A=1,M=1}) - E(Y_{A=0,M=1})
CDE_mis1_gcomp <- (coef(MSM.CDE.gcomp)["A0_ace"] +
                     coef(MSM.CDE.gcomp)["A0_ace:M_smoking"])
# 0.07918291

# Note: Applying a binomial family for the first Qbar model would result in two
# sligthly different values of the CDE(M=m)
# => 0.05934409 in setting M=0
# => 0.07537874 in setting M=1
```

If the exposure A affects the intermediate confounder $L(1)$, as in the Causal model 2, the steps 1) and 2) of the algorithm should follow the method described in paragraph 6.2.1 or 6.2.2.

Here is an example applying G-computation by iterative conditional expectation:

```
### MSM of CDE, estimated by G-computation (by ICE) ----------------------------
## 1a) Regress the outcome on L0, A, L1 and M (and the A*M interaction if appropriate)
Y.death.model <- glm(Y_death ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1 +
                       M_smoking + A0_ace:M_smoking,
                     family = "binomial", data = df2_int)

## 1b) Generate predicted values by evaluating the regression
##     under the 4 counterfactual scenarios
data.A0M0 <- data.A1M0 <- data.A0M1 <- data.A1M1 <- df2_int
data.A0M0$A0_ace <- 0
data.A0M0$M_smoking <- 0

data.A1M0$A0_ace <- 1
data.A1M0$M_smoking <- 0

data.A0M1$A0_ace <- 0
data.A0M1$M_smoking <- 1
```

```r
data.A1M1$A0_ace <- 1
data.A1M1$M_smoking <- 1

Q.Y.death.A0M0 <- predict(Y.death.model, newdata = data.A0M0, type = "response")
Q.Y.death.A1M0 <- predict(Y.death.model, newdata = data.A1M0, type = "response")
Q.Y.death.A0M1 <- predict(Y.death.model, newdata = data.A0M1, type = "response")
Q.Y.death.A1M1 <- predict(Y.death.model, newdata = data.A1M1, type = "response")

## 2a) Regress the predicted values conditional on the observed exposure A
##    and baseline confounders L(0)
L1.death.A0M0.model <- glm(Q.Y.death.A0M0 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                           family = "quasibinomial", data = df2_int)
L1.death.A1M0.model <- glm(Q.Y.death.A1M0 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                           family = "quasibinomial", data = df2_int)
L1.death.A0M1.model <- glm(Q.Y.death.A0M1 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                           family = "quasibinomial", data = df2_int)
L1.death.A1M1.model <- glm(Q.Y.death.A1M1 ~ L0_male + L0_parent_low_educ_lv + A0_ace,
                           family = "quasibinomial", data = df2_int)


## 2b) generate predicted values by evaluating the regression at exposure
##    of interest: {A=0,M=0}, {A=1,M=0}, {A=0,M=1}, {A=1,M=1}
data.A0M0$Yam.death.pred <- predict(L1.death.A0M0.model,
                                    newdata = data.A0M0, type = "response")
data.A1M0$Yam.death.pred <- predict(L1.death.A1M0.model,
                                    newdata = data.A1M0, type = "response")
data.A0M1$Yam.death.pred <- predict(L1.death.A0M1.model,
                                    newdata = data.A0M1, type = "response")
data.A1M1$Yam.death.pred <- predict(L1.death.A1M1.model,
                                    newdata = data.A1M1, type = "response")


## 3. Append the 4 counterfactual datasets in a single long dataset
# number of row is 4 times the initial value (we have 4 counterfactual scenarios)
data.4scenarios <- rbind(data.A0M0, data.A1M0,data.A0M1,data.A1M1)

## 4. fit the MSM: E(Y_am) = alpha_0 + alpha_A a + alpha_M m + alpha_AM a:m
MSM.CDE.gcomp <- glm(Yam.death.pred ~ A0_ace +  M_smoking + A0_ace:M_smoking,
                     family = "gaussian", # gaussian family for risk differences
                     data = data.4scenarios)
coef(MSM.CDE.gcomp)
# (Intercept)          A0_ace        M_smoking A0_ace:M_smoking
#  0.17974947      0.06342833       0.07366466       0.02469485

## 5. Estimate the CDE(M=m)
```

```
# CDE(M=0) = E(Y_{A=1,M=0}) - E(Y_{A=0,M=0})
CDE_mis0_gcomp_ice <- coef(MSM.CDE.gcomp)["A0_ace"]
# 0.06342833

# CDE(M=1) = E(Y_{A=1,M=1}) - E(Y_{A=0,M=1})
CDE_mis1_gcomp_ice <- (coef(MSM.CDE.gcomp)["A0_ace"] +
                         coef(MSM.CDE.gcomp)["A0_ace:M_smoking"])
# 0.08812318
```

The example above (MSM estimation using G-computation by ICE) corresponds to the algorithm applied by the `ltmle` package:

```
library(ltmle)
Qform <- c(L1="Q.kplus1 ~ L0_male + L0_parent_low_educ_lv + A0_ace",
           Y_death="Q.kplus1 ~ L0_male + L0_parent_low_educ_lv + L1 +
                    A0_ace * M_smoking")
gform <- c("A0_ace ~ L0_male + L0_parent_low_educ_lv",
           "M_smoking ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1")
# in this example of g-computation, the propensity scores 'gform' will not be used

data_binary <- subset(df2_int, select = c(L0_male, L0_parent_low_educ_lv,
                                          A0_ace, L1,
                                          M_smoking, Y_death))
CDE_ltmle_M0 <- ltmle(data = data_binary,
                      Anodes = c("A0_ace", "M_smoking"),
                      Lnodes = c("L1"), # intermediate confounders +/- baseline
                      Ynodes = c("Y_death"),
                      survivalOutcome = FALSE, # TRUE for time-to-event outcomes Y
                      Qform = Qform,
                      gform = gform,
                      abar = list(c(1,0), # counterfactual intervention do(A=1,M=0)
                                  c(0,0)), # counterfactual intervention do(A=0,M=0)
                      SL.library = NULL, # call glm() instead of SuperLearner
                      estimate.time = FALSE, # estimate computation time?
                      gcomp = TRUE, # to apply g-computation
                      variance.method = "ic")
# CDE with M=0
summary(CDE_ltmle_M0)$effect.measures$ATE$estimate
#   Parameter Estimate:  0.06342833

CDE_ltmle_M1 <- ltmle(data = data_binary,
                      Anodes = c("A0_ace", "M_smoking"),
                      Lnodes = c("L1"), # intermediate confounders +/- baseline
                      Ynodes = c("Y_death"),
                      survivalOutcome = FALSE, # TRUE for time-to-event outcomes Y
```

```
                      Qform = Qform,
                      gform = gform,
                      abar = list(c(1,1), # counterfactual intervention do(A=1,M=0)
                                  c(0,1)), # counterfactual intervention do(A=0,M=0)
                      SL.library = NULL, # call glm() instead of SuperLearner
                      estimate.time = FALSE, # estimate computation time?
                      gcomp = TRUE, # to apply g-computation
                      variance.method = "ic")
# CDE with M=1
summary(CDE_ltmle_M1)$effect.measures$ATE$estimate
#   Parameter Estimate:  0.08812318
```

8.3 MSM for Natural Direct and Indirect Effects

8.3.1 Expressing the NDE and NIE using coefficients of 2 MSMs

The (Pure) Natural Direct Effect is defined by PNDE $= \mathbb{E}(Y_{a,M_{a^*}}) - \mathbb{E}(Y_{a^*,M_{a^*}})$ and the (Total) Natural Indirect Effect is defined by TNIE $= \mathbb{E}(Y_{a,M_a}) - \mathbb{E}(Y_{a,M_{a^*}})$.

VanderWeele suggested using 2 MSMs conditional on baseline confounders $L(0)$ in order to estimate natural direct and indirect effects (VanderWeele 2009):

1) a model of the counterfactual values of the outcome $\mathbb{E}(Y_{a,m} \mid l(0)) = h^{-1}(a, m, l(0))$, where h is a link function. For example:

$$\mathbb{E}(Y_{a,m} \mid l(0)) = \alpha_0 + \alpha_A a + \alpha_M m + \alpha_{AM} a \times m + \alpha_{L(0)} l(0) \tag{8.7}$$

(where h is the identity function, so that the model can be used to express risk differences)

2) a model of the counterfactual values of the mediator $\mathbb{E}(M_a \mid L(0)) = g^{-1}(a, l(0))$, where g is a link function. For example with a binary mediator:

$$\mathbb{E}(M_a \mid l(0)) = g^{-1}\left[\beta_0 + \beta_A a + \beta_{L(0)} l(0)\right] \tag{8.8}$$

(where g is the logit function because the mediator is binary).

VanderWeele shows that if the function h is linear in m (no quadratic terms in m, nor transformations such as $\log(m)$ or $\sqrt{m}$, etc) and the exposure A does not affect the intermediate confounder $L(1)$, then

$$\mathbb{E}(Y_{a,M_{a^*}}) = h^{-1}\left[a, g^{-1}\left(a^*, l(0)\right), l(0)\right]$$

Using the 2 MSMs, we can express the Natural Direct and Indirect Effects conditional on baseline confounders $L(0)$. In our example:

$$\begin{aligned}
\text{PNDE} \mid L(0) &= \mathbb{E}(Y_{a,M_{a^*}} \mid L(0)) - \mathbb{E}(Y_{a^*,M_{a^*}} \mid L(0)) \\
&= \{\alpha_0 + \alpha_A a + [\alpha_M + \alpha_{AM} a] \times g^{-1}(a^*, l(0)) + \alpha_{L(0)} l(0)\} \\
&\quad - \{\alpha_0 + \alpha_A a^* + [\alpha_M + \alpha_{AM} a^*] \times g^{-1}(a^*, l(0)) + \alpha_{L(0)} l(0)\} \\
&= (a - a^*) \times [\alpha_A + \alpha_{AM} \times g^{-1}(a^*, l(0))]
\end{aligned}$$

$$\begin{aligned}
\text{TNIE} \mid L(0) &= \mathbb{E}(Y_{a,M_a} \mid L(0)) - \mathbb{E}(Y_{a,M_{a^*}} \mid L(0)) \\
&= \{\alpha_0 + \alpha_A a + [\alpha_M + \alpha_{AM} a] \times g^{-1}(a, l(0)) + \alpha_{L(0)} l(0)\} \\
&\quad - \{\alpha_0 + \alpha_A a + [\alpha_M + \alpha_{AM} a] \times g^{-1}(a^*, l(0)) + \alpha_{L(0)} l(0)\} \\
&= \left[g^{-1}(a, l(0)) - g^{-1}(a^*, l(0))\right] (\alpha_M + \alpha_{AM} a)
\end{aligned}$$

Marginal Natural Direct and Indirect effect can then be obtained:

$$\begin{aligned}
\text{PNDE} &= \sum_{l(0)} [\text{PNDE} \mid L(0) = l(0)] \times P(L(0) = l(0)) \\
\text{TNIE} &= \sum_{l(0)} [\text{TNIE} \mid L(0) = l(0)] \times P(L(0) = l(0))
\end{aligned}$$

8.3.2 Estimation of the 2 MSMs coefficients by IPTW for NDE and NIE

As previously, MSM coefficients can be estimated using an Inverse Probability of Treatment (IPTW) approach based on weighted regressions.

In order to fit the 1st MSM (8.7), we can use a linear regression of the (observed) outcome Y on the exposure and mediator, adjusted for $L(0)$, weighted by individual stabilized weights $sw_{msm1,i}$ (VanderWeele 2009):

$$\mathbb{E}\left(Y \mid A, M, L(0)\right) = \alpha_0 + \alpha_A a + \alpha_M m + \alpha_{AM} a \times m + \alpha_{L(0)} L(0)$$

where $sw_{msm1,i}$ is the product of two weights $sw_{msm1,i} = sw_{A,i} \times sw_{M,i}$,

$$\begin{aligned}
sw_{A,i} = &\frac{P(A = a_i)}{P(A = a_i \mid L(0) = l(0)_i)} \quad \text{or} \quad sw_{A,i} = \frac{P(A = a_i \mid L(0) = l(0)_i)}{P(A = a_i \mid L(0) = l(0)_i)} = 1 \\
sw_{M,i} = &\frac{P(M = m_i \mid A = a_i)}{P(M = m_i \mid A = a_i, L(0) = l(0)_i), L(1) = l(1)_i} \\
&\quad \text{or} \quad sw_{M,i} = \frac{P(M = m_i \mid A = a_i, L(0) = l(0)_i)}{P(M = m_i \mid A = a_i, L(0) = l(0)_i), L(1) = l(1)_i}
\end{aligned}$$

In order to fit the 2nd MSM (8.8), we can use a logistic regression of the (observed) mediator M on the exposure, adjusted for $L(0)$, weighted by individual stabilized weights $sw_{msm2,i}$:

$$\text{logit}\mathbb{E}(M \mid a, l(0)) = \beta_0 + \beta_A a + \beta_{L(0)} l(0)$$

$$\text{where} \quad sw_{msm2,i} = \frac{P(A = a_i)}{P(A = a_i \mid L(0) = l(0)_i)}$$

```
### MSM of NDE & NIE, estimated by IPTW ----------------------------------------
## 1. Stabilized weight for the MSM1
# 1a. sw_Ai = g(A=a_i | L(0)) / g(A=a_i | L(0)) = 1
sw_Ai <- rep(1, nrow(df1_int))

# 1b. sw_Mi = g(M=m_i | A,L(0)) / g(M=m_i | A,L(0),L(1))
g.M.AL0 <- glm(M_smoking ~ A0_ace + L0_male + L0_parent_low_educ_lv,
               family = "binomial", data = df1_int)
g.Mis1.AL0 <- predict(g.M.AL0, type = "response")
sw_M.num <- rep(NA, nrow(df1_int))
sw_M.num[df1_int$M_smoking==1] <- g.Mis1.AL0[df1_int$M_smoking==1]
sw_M.num[df1_int$M_smoking==0] <- (1 - g.Mis1.AL0[df1_int$M_smoking==0])

g.M.AL0L1 <- glm(M_smoking ~ A0_ace + L0_male + L0_parent_low_educ_lv + L1,
                 family = "binomial", data = df1_int)
g.Mis1.AL0L1 <- predict(g.M.AL0L1, type = "response")
sw_M.denom <- rep(NA, nrow(df1_int))
sw_M.denom[df1_int$M_smoking==1] <- g.Mis1.AL0L1[df1_int$M_smoking==1]
sw_M.denom[df1_int$M_smoking==0] <- (1 - g.Mis1.AL0L1[df1_int$M_smoking==0])

sw_msm1 <- sw_Ai * sw_M.num / sw_M.denom

## 2. Estimate coefficients of the MSM1
MSM1 <- glm(Y_death ~ A0_ace + M_smoking + A0_ace:M_smoking +
              L0_male + L0_parent_low_educ_lv,
            weights = sw_msm1,
            family = "gaussian",
            data = df1_int)
coef(MSM1)
# (Intercept)                A0_ace             M_smoking
#  0.12033221            0.06381257            0.06691712
#     L0_male L0_parent_low_educ_lv      A0_ace:M_smoking
#  0.04671886            0.05521263            0.01652446

## 3. Stabilized weight for the MSM2
# 3a. sw_A = g(A=a_i) / g(A=a_i | L(0))
```

```
# numerator
g.A <- glm(A0_ace ~ 1, family = "binomial", data = df1_int)
g.Ais1 <- predict(g.A, type = "response")
sw_msm2.num <- rep(NA, nrow(df1_int))
sw_msm2.num[df1_int$A0_ace==1] <- g.Ais1[df1_int$A0_ace==1]
sw_msm2.num[df1_int$A0_ace==0] <- (1 - g.Ais1[df1_int$A0_ace==0])

# denominator
g.A.L0 <- glm(A0_ace ~ L0_male + L0_parent_low_educ_lv,
              family = "binomial", data = df1_int)
g.Ais1.L0 <- predict(g.A.L0, type = "response")
sw_msm2.denom <- rep(NA, nrow(df1_int))
sw_msm2.denom[df1_int$A0_ace==1] <- g.Ais1.L0[df1_int$A0_ace==1]
sw_msm2.denom[df1_int$A0_ace==0] <- (1 - g.Ais1.L0[df1_int$A0_ace==0])

# stabilized weight
sw_msm2 <- sw_msm2.num / sw_msm2.denom

## 3. Estimate coefficients of the MSM2
MSM2 <- glm(M_smoking ~ A0_ace + L0_male + L0_parent_low_educ_lv,
            weights = sw_msm2,
            family = "binomial",
            data = df1_int)
coef(MSM2)
# (Intercept)                A0_ace               L0_male L0_parent_low_educ_lv
#  -1.2723106             0.5883720             0.2566129             0.3270087

## 4. Estimate PNDE conditional on L(0), and the marginal value of PNDE
# a = 1 and a* = 0
# PNDE|L(0) = (a - a*)[alpha_A + alpha_AM.g^-1(a^*,l(0))]
g.minus1.A0 <- plogis(coef(MSM2)["(Intercept)"] + coef(MSM2)["A0_ace"] * 0 +
                      coef(MSM2)["L0_male"] * df1_int$L0_male +
                      coef(MSM2)["L0_parent_low_educ_lv"] * df1_int$L0_parent_low_educ_lv)

# PNDE conditional on L(0)
PNDE_L0 <- (1 - 0) * (coef(MSM1)["A0_ace"] +
                        coef(MSM1)["A0_ace:M_smoking"] * g.minus1.A0)
# marginal PNDE
PNDE <- mean(PNDE_L0)
# [1] 0.06850657

## 4. Estimate TNIE conditional on L(0), and the marginal value of TNIE
# TNIE|L(0) = [g^-1(a,l(0)) - g^-1(a^*,l(0))] * (alpha_M + alpha_AM * a)
g.minus1.A1 <- plogis(coef(MSM2)["(Intercept)"] + coef(MSM2)["A0_ace"] * 1 +
                        coef(MSM2)["L0_male"] * df1_int$L0_male +
```

```
                        coef(MSM2)["L0_parent_low_educ_lv"] * df1_int$L0_parent_low_edu

# TNIE conditional on L(0)
TNIE_L0 <- (g.minus1.A1 - g.minus1.A0) * (coef(MSM1)["M_smoking"] +
                                            coef(MSM1)["A0_ace:M_smoking"] * 1)
# marginal PNDE
TNIE <- mean(TNIE_L0)
# [1] 0.01096799
```

In this example, the estimation of the PNDE is 6.9% and the estimation of the TNIE is 1.1%. Confidence intervals can be calculated by bootstrap.

Chapter 9

Targeted Maximum Likelihood Estimation (TMLE)

When estimating a mean counterfactual outcome using g-computation methods, we have to estimate some $\bar{Q}$ functions (functions of the outcome conditional on the exposures and confounders, $\bar{Q} = \mathbb{E}\left(Y \mid A, L(0)\right)$). For example, the Average Total Effect (ATE) is defined as a marginal effect, estimated using the empirical mean of such $\bar{Q}$ functions:

$$\hat{\Psi}^{\text{ATE}}_{\text{gcomp}} = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{\bar{Q}}(A=1)_i - \hat{\bar{Q}}(A=0)_i\right]$$

Unless the $\bar{Q}$ functions are not misspecified, its estimate is expected to be biased (and $\bar{Q}$ are expected to be misspecified, especially if the set of baseline confounders $L(0)$ is high dimensional, for example if it includes is a large number of variables or continuous variables). In order to improve the estimation of $\bar{Q}(A, L)$, it is possible to use data-adaptive methods (machine learning algorithms) in order to optimize the bias-variance trade-off. However, this bias-variance trade-off would be optimized for the $\bar{Q}$ functions, not for the ATE estimate $\hat{\Psi}^{\text{ATE}}_{\text{gcomp}}$. If the $\bar{Q}$ function is unknown and has to be estimated (preferably by data-adaptive methods), it can be shown that the g-computation estimate of Ψ^{ATE} is asymptotically biased.

The Targeted Maximum Likelihood Estimation (TMLE) method has been developed as an asymptotically linear estimator, so that the estimation of any target parameter in any semiparametric statistical model is unbiased and efficient. In order to estimate a parameter $\Psi(P_0)$, where P_0 is an unknown probability distribution among a set $\mathcal{M}$ of possible statistical models, the TMLE is described as a two-step procedure (Laan and Rose 2011):

- The first step is to obtain an initial estimate of the relevant part ($\bar{Q}_0$ in our applications) of the probability distribution P_0. Data adaptive methods (machine learning algorithms) can be used to optimize this first step.
- The second step is to update the initial fit in order to "target toward making an optimal bias-variance tradeoff for the parameter of interest" $\Psi(\bar{Q})$.

Several R packages have been developed in order to carry out TMLE estimation of causal effects. We will begin using the `ltmle` package, as it can be used to estimate ATE or CDE. More generally, this package can be used to estimate the counterfactual effects of repeated exposure in time-to-event settings. In the setting of mediation analysis, a controlled direct effect (CDE) corresponds to a sequence of counterfactual interventions on 2 "exposure variables": the initial exposure A and the mediator of interest M. The package can also be used in simpler settings with only one binary or continuous outcome, measure only once at the end a the study.

9.1 TMLE for the ATE

In order to illustrate the TMLE procedure, the estimation of a mean counterfactual outcome, denoted $\Psi(A=1) = \mathbb{E}\left[\bar{Q}(A=1, L(0))\right]$, will be described in detail, following the algorithm implemented in the `ltmle` package.

The basic steps of the procedure are the following (Laan and Rose 2011):

1. Estimate $\bar{Q}_0$. Data-adaptive methods can be used here, the `ltmle` package relies on the `SuperLearner` package to fit and predict $\hat{\bar{Q}}(A=1)$.

2. Estimate the treatment mechanism $g(A=1 \mid L(0))$. Once again, data-adaptive methods can be used to improve the estimation.

3. The initial estimator of $\bar{Q}_0(A=1)$ will be slightly modified using a parametric fluctuation model, in order to reduce the bias when estimating the ATE. For example, the following parametric model of $\bar{Q}_0(A=1)$ and a "clever covariate" $H = \frac{I(A=1)}{\hat{g}(A=1|L(0))}$ can be applied:

$$\text{logit} P(Y \mid \hat{\bar{Q}}, H) = \widehat{\text{logit}} \bar{Q} + \varepsilon H$$

   The parametric fluctuation model is chosen so that the derivative of its log-likelihood loss function is equal to the appropriate component of the efficient influence curve of the target parameter $\Psi(A=1)$.

4. Modify the initial estimator of $\bar{Q}_0(A=1)$ with the parametric fluctuation model (using the estimation $\hat{\varepsilon}$ from the previous step). We denote $\hat{\bar{Q}}^*(A=1)$ the updated value of $\hat{\bar{Q}}(A=1)$

5. Use the updated values $\hat{\bar{Q}}^*(A=1)$ in the substitution estimator to estimate the target parameter $\Psi(A=1)$:

$$\hat{\Psi}(A=1)_{\text{TMLE}} = \frac{1}{n}\sum_{i=1}^{n}\hat{\bar{Q}}^*(A=1, L(0))$$

6. Estimate the efficient influence curve $D^*(Q_0, g_0)$:

$$D^*(Q_0, g_0) = \frac{I(A=1)}{g_0(A=1 \mid L(0))}(Y - \bar{Q}_0(A, L(O))) + \bar{Q}_0(A=1, L(0)) + \Psi(A=1)$$

The variance of the target parameter can then be calculated using the variance of the efficient influence curve:

$$\text{var}\hat{\Psi}(A=1)_{\text{TMLE}} = \frac{\text{var}\hat{D}^*}{n}$$

```
## 1) Estimate Qbar and predict Qbar when A0_ace is set to 1
Q.fit <- glm(Y_death ~ A0_ace + L0_male + L0_parent_low_educ_lv,
             family = "binomial", data = df2_int)
data.A1 <- df2_int
data.A1$A0_ace <- 1

# predict the Qvar function when setting the exposure to A=1, on the logit scale
logit_Qbar_Ais1 <- predict(Q.fit, newdata = data.A1, type = "link")

## 2) Estimate the treatment mechanism
g.L <- glm(A0_ace ~ L0_male + L0_parent_low_educ_lv,
           family = "binomial", data = df2_int)
# predict the probabilities g(A=1 | L(0)) = P(A0_ace=1|L(0))
g1.L <- predict(g.L, type="response")

head(g1.L)
#          1          2          3          4          5          6
# 0.10989220 0.15629749 0.15629749 0.08894074 0.15629749 0.15629749

# It is useful to check the distribution of gA.L, as values close to 0 or 1 are
# indicators of near positivity violation and can result in large variance for the
# estimation.
# In case of near positivity violation, gA.L values can be truncated to decrease
# the variance (at the cost a increased bias).
```

```r
summary(g1.L)
#     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 0.06109 0.08894 0.10989 0.11240 0.15630 0.15630
# there is no positivity issues in this example.

## 3) Determine a parametric family of fluctuations of Qbar.
# The fluctuation model is a model of logitQbar and g(A=1|L(0))

# The clever covariate H(A,L(0)) depends on g(A=1|L(0)):
H <- (df2_int$A0_ace == 1) / g1.L

# Update the initial fit Qbar from step 1.
# This is achieved by holding Qbar fixed (as intercept) while estimating the
# coefficient epsilon for H

# for example we could use the following fluctuation model (from the "Targeted
# Learning" book)
update.fit <- glm(df2_int$Y_death ~ -1 + offset(logitQ) + H,
                  family = "quasibinomial")
# Coefficients:
#   H
# -0.0001756

# In the ltmle package, the fluctuation parametric model is slightly different
# (but with the same purpose). The "clever covariate" H is scaled and used as a
# weight in the parametric quasi-logistic regression
S1 <- rep(1, nrow(df2_int))
update.fit.ltmle <- glm(df2_int$Y_death ~ -1 + S1 + offset(logitQ),
                        family = "quasibinomial",
                        weights = scale(H, center = FALSE))
# Coefficients:
#   S1
# -0.001667

## 4) Update the initial estimate of Qbar using the fluctuation parametric model
Qstar.tmle <- predict(update.fit.ltmle,
                      data = data.frame(logitQ, H),
                      type = "response")

head(Qstar.tmle)
#         1         2         3         4         5         6
# 0.2872412 0.3441344 0.3441344 0.2591356 0.3441344 0.3441344

## 5) Obtain the substition estimator of Psi_Ais1
Psi_Ais1 <- mean(Qstar.tmle)
```

```
# [1] 0.2871408

## 5) Calculate standard errors based on the influence curve of the TMLE
IC <- H * (df2_int$Y_death - Qstar.tmle) + Qstar.tmle - Psi_Ais1
head(IC)
# 0.0001003559  4.2532581791  0.0569935644 -0.0280052148  0.0569935644  0.056993564

# Teh standard error of the target parameter Psi(A=1) can be estimated by :
sqrt(var(IC)/nrow(df2_int))
# [1] 0.01383821
```

We can see that we can get the same output using the `ltmle` package:

```
library(ltmle)
?ltmle

# The Qform and gform arguments are defined from the DAG
Qform <- c(Y_death="Q.kplus1 ~ L0_male + L0_parent_low_educ_lv + A0_ace")
gform <- c("A0_ace ~ L0_male + L0_parent_low_educ_lv")

# in the ltmle package, the data set should be formated so that the order of the
# columns corresponds to the time-ordering of the model
data_ltmle <- subset(df2_int, select = c(L0_male,L0_parent_low_educ_lv,
                                         A0_ace,
                                         Y_death))

# the counterfactual intervention is defined in the abar argument
abar <- 1

Psi_Ais1 <- ltmle(data = data_ltmle,
                  Anodes = "A0_ace",
                  Ynodes = "Y_death",
                  Qform = Qform,
                  gform = gform,
                  gbounds = c(0.01, 1), # by default, g function are truncated at 0.01
                  abar = abar,
                  SL.library = "glm",
                  variance.method = "ic")

# from the ltmle() function, we can get the point estimate, its standard error,
# 95% confidence interval and the p-value for the null hypothesis.
summary(Psi_Ais1, "tmle")
# Parameter Estimate:  0.28714
#  Estimated Std Err:  0.013838
#            p-value:  <2e-16
```

```
#  95% Conf Interval: (0.26002, 0.31426)

# The ltmle() function returns an object with several outputs.
# We can see that g functions are the same as in the previous manual calculation
head(Psi_Ais1$cum.g)
#            [,1]
# [1,] 0.10989220
# [2,] 0.15629749
# [3,] 0.15629749
# [4,] 0.08894074
# [5,] 0.15629749
# [6,] 0.15629749

# we can get the estimation of the epsilon parameter from the fluctuation model
Psi_Ais1$fit$Qstar
# Coefficients:
#   S1
# -0.001667
# Degrees of Freedom: 1124 Total (i.e. Null);  1123 Residual

# we can get the updated Qbar functions:
head(Psi_Ais1$Qstar)
# [1] 0.2872412 0.3441344 0.3441344 0.2591356 0.3441344 0.3441344

# we can get the influence curve
head(Psi_Ais1$IC$tmle)
# [1]  0.0001003559  4.2532581791  0.0569935644 -0.0280052148  0.0569935644  0.0569935
```

In practice, it is recommended to apply data-adaptive algorithms to estimate $\bar{Q}$ and g functions: the `ltmle` package relies on the `SuperLearner` package. As indicated in the Guide to SuperLearner, The `SuperLearner` is "an algorithm that uses cross-validation to estimate the performance of multiple machine learning models, or the same model with different settings. It then creates an optimal weighted average of those models (ensemble learning) using the test data performance."

Here is an example for our estimation of the Average Total Effect (ATE).

The `SuperLearner` package includes a set of algorithms with default parameters (showed by `listWrappers()`). Because the simulated data set only have 2 binary baseline variables, the set $\mathcal{M}$ of possible statistical models is limited. In order to estimate the ATE, we will include a library with:

- `SL.mean`, the null-model which only predict the marginal mean (it can be used as a reference for a bad model);
- `SL.glm`, a glm using the main terms from the `Qform` and `gform` argument;

- `SL.interaction.back`, a step-by-step backward GLM prodecure (based on the AIC), starting with all 2×2 interactions between main terms. Interaction terms might be useful to estimate the $\bar{Q}(A, L(0))$ function because the dataset was generated from and additive model, where as the function is estimated below using a logistic (multiplicative) model.
- `SL.xgboost.custom` a customized xgboost algorithm from the initial `SL.xgboost` algorithm, showing how we can modify some default arguments.

```
library(SuperLearner)
library(xgboost)
# Below, we use the same ltmle() function than previously,
# and specify a family of algorithms to be used with the SuperLearner

## we can change the default argument of the SL.xgboost algorithm and the
## SL.step.interaction algorithm

# We can check how arguments are used in the pre-specified algorithms
SL.step.interaction
# function (Y, X, newX, family, direction = "both", trace = 0,
#     k = 2, ...)
# {
#     fit.glm <- glm(Y ~ ., data = X, family = family)
#     fit.step <- step(fit.glm, scope = Y ~ .^2, direction = direction,
#         trace = trace, k = k)
#     pred <- predict(fit.step, newdata = newX, type = "response")
#     fit <- list(object = fit.step)
#     out <- list(pred = pred, fit = fit)
#     class(out$fit) <- c("SL.step")
#     return(out)
# }
# <bytecode: 0x000001b965ed0dc0>
# <environment: namespace:SuperLearner>

# The pre-specified algorithm can be easily modified to obtain a step-by-step backward
# selection.
SL.interaction.back = function(...) {
  SL.step.interaction(..., direction = "backward")
}

# The same principle can be applied to modify the SL.xgboost default algorithm
SL.xgboost
SL.xgboost.custom = function(...) {
  SL.xgboost(..., ntrees = 50)
}
```

```
## the algorithms we would like to use can be specified separately for the Q and
# g functions
SL.library <- list(Q=c("SL.mean","SL.glm","SL.interaction.back", "SL.xgboost.custom"),
                   g=c("SL.mean","SL.glm","SL.interaction.back", "SL.xgboost.custom"))

set.seed(42)
Psi_ATE_tmle <- ltmle(data = data_ltmle,
                      Anodes = "A0_ace",
                      Ynodes = "Y_death",
                      Qform = Qform,
                      gform = gform,
                      gbounds = c(0.01, 1),
                      abar = list(1,0), # vector of the counterfactual treatment
                      SL.library = SL.library,
                      variance.method = "ic")
summary(Psi_ATE_tmle, estimator = "tmle")
# The function give the ATE on the difference scale (as well, as RR and OR)
# Additive Treatment Effect:
   # Parameter Estimate:  0.081832
   #  Estimated Std Err:  0.014291
   #            p-value:  1.0275e-08
   #  95% Conf Interval: (0.053822, 0.10984)

## We can see how the SuperLearner used the algorithms for the g function
Psi_ATE_tmle$fit$g
# [[1]]$A0_ace
#                                Risk        Coef
# SL.mean_All             0.09976892 0.003545569 # risk is higher for the bad model
# SL.glm_All              0.09865424 0.416238369
# SL.interaction.back_All 0.09865424 0.000000000
# SL.xgboost.custom_All   0.09865550 0.580216062

# for the g function, the SuperLearner predicts the treatment mechanism
# base on a mix between the glm and the customized xgboost algorithm.

## We can see how the SuperLearner used the algorithms for the g function
Psi_ATE_tmle$fit$Q
#                               Risk       Coef
# SL.mean_All             0.1684737 0.02003166 # risk is higher for the bad model
# SL.glm_All              0.1662241 0.00000000
# SL.interaction.back_All 0.1662241 0.55956284
# SL.xgboost.custom_All   0.1662422 0.42040550

# The SuperLearner predicts both the treatment mechanism g and the Q function
```

```r
# from a mix between the backward interaction glm (or the main term glm) and the
# customized xgboost algorithm.
# However, the choice between the SL.glm and the SL.interaction.back
# procedure was arbitrary: as we can see the Risk is exactly the same for both
# algorithms. The final model from the step-by-step procedure was much probably
# a main term glm.


## The `ltmle` package can also be used to estimate the effect of binary exposures
## on continous outcomes
Qform <- c(Y_qol="Q.kplus1 ~ L0_male + L0_parent_low_educ_lv + A0_ace")
gform <- c("A0_ace ~ L0_male + L0_parent_low_educ_lv")

set.seed(42)
Psi_ATE_tmle_qol <- ltmle(data = subset(df2_int, select = c(L0_male,L0_parent_low_educ_lv,
                                        A0_ace,
                                        Y_qol)),
                      Anodes = "A0_ace",
                      Ynodes = "Y_qol",
                      Qform = Qform,
                      gform = gform,
                      gbounds = c(0.01, 1),
                      abar = list(1,0), # vector of the counterfactual treatment
                      SL.library = SL.library,
                      variance.method = "ic")
summary(Psi_ATE_tmle_qol, estimator = "tmle")
# Additive Treatment Effect:
#    Parameter Estimate:  -8.265
#     Estimated Std Err:  0.41008
#               p-value:  <2e-16
#     95% Conf Interval: (-9.0687, -7.4612)
```

The TMLE estimation of the ATE from the `ltmle` package for death probability and mean quality of life is +8.18% (95% CI=[+5.38%, +10.98%]) and -8.27 [-9.07, -7.46].

Note that the `ltmle` package can also be used to calculate the IPTW estimation of the ATE and the CDE.

```r
# using the output from the previous ltmle() procedure
summary(Psi_ATE_tmle, estimator = "iptw")
# Additive Treatment Effect:
#    Parameter Estimate:  0.082578
#     Estimated Std Err:  0.014415
#               p-value:  1.0135e-08
```

```
#     95% Conf Interval: (0.054325, 0.11083)

summary(Psi_ATE_tmle_qol, estimator = "iptw")
# Additive Treatment Effect:
#    Parameter Estimate:  -8.2887
#     Estimated Std Err:  0.41799
#               p-value:  <2e-16
#     95% Conf Interval: (-9.108, -7.4695)
```

The IPTW estimation of the ATE from the `ltmle` package for death probability and mean quality of life is +8.26% (95% CI=[+5.43%, +11.08%]) and -8.29 [-9.11, -7.47].

9.2 TMLE of the Controlled direct effect (CDE)

If the controlled direct effect (CDE) is identifiable, the `ltmle` package can be used to calculate a TMLE estimation of the CDE $\Psi^{\text{CDE}_m} = \mathbb{E}(Y_{A=1,M=m}) - \mathbb{E}(Y_{A=0,M=m})$.

Below, we show how to use the `ltmle()` function to estimate CDE by TMLE, with data generated from the causal model with the presence of confounders of the mediator-outcome relationship ($L(1)$) affected by the exposure A (Figure 3.2), and an $A * M$ interaction effect on the outcome.

As with the G-computation method by iterative conditional expectation, the TMLE procedure relies on the estimation of 2 $\bar{Q}$ functions:

- $\bar{Q}_Y = \mathbb{E}(Y \mid L(0), A, L(1), M)$
- and $\bar{Q}_{L(1)} = \mathbb{E}(\hat{\bar{Q}}_Y(M = m) \mid L(0), A)$;

And as with the IPTW method, the TMLE procedure relies also on the estimation of the 2 treatment mechanisms g:

- $g_A(L(0)) = P(A = 1 \mid L(0))$
- and $g_M(L(0), A, L(1)) = P(M = 1 \mid L(0), A, L(1))$.

Note: for continuous outcomes, the `ltmle` package transforms the outcome on a 0 to 1 continuous scale, $Y_{\text{transformed}} = \frac{Y - \min(Y)}{\max(Y) - \min(Y)}$, so that quasi-binomial parametric models can be used in the computation procedure. Mean predictions are then back-transformed on the original scale.

9.2.1 For binary outcomes

```
library(ltmle)
# Define the formulas for the estimation of the 2 barQ functions
# Note that it is possible to specify the A*M interaction, if we really want to
# take it into account.
# Another option is to indicate prediction algorithms well adapted to the estimation
# of interaction phenomena into the SuperLearner arguments.
Qform <- c(L1="Q.kplus1 ~ L0_male + L0_parent_low_educ_lv + A0_ace",
           Y_death="Q.kplus1 ~ L0_male + L0_parent_low_educ_lv + L1 +
                    A0_ace * M_smoking")

# Define the formulas for the estimation of the 2 g function
gform <- c("A0_ace ~ L0_male + L0_parent_low_educ_lv",
           "M_smoking ~ L0_male + L0_parent_low_educ_lv + A0_ace + L1")

# The data frame should follow the time-ordering of the nodes
data_binary <- subset(df2_int, select = c(L0_male, L0_parent_low_educ_lv,
                                          A0_ace, L1,
                                          M_smoking, Y_death))



# Choose a family of data-adaptive algorithms from the SuperLearner package
SL.library <- list(Q=c("SL.mean","SL.glm","SL.step.interaction","SL.xgboost"),
                   g=c("SL.mean","SL.glm","SL.step.interaction","SL.xgboost"))


## CDE, setting M=0
set.seed(42) # for reproducibility (xgboost algorithm relies on random procedures)
CDE_ltmle_M0_death <- ltmle(data = data_binary,
                            Anodes = c("A0_ace", "M_smoking"),
                            Lnodes = c("L1"), # intermediate confounders +/- baseline
                            Ynodes = c("Y_death"),
                            survivalOutcome = FALSE, # TRUE for time-to-event outcomes Y
                            Qform = Qform,
                            gform = gform,
                            abar = list(c(1,0), # counterfactual intervention do(A=1,M=0)
                                        c(0,0)), # counterfactual intervention do(A=0,M=0)
                            SL.library = SL.library,
                            estimate.time = FALSE, # estimate computation time?
                            gcomp = FALSE,
                            variance.method = "ic") # a more robust variance can
                                                    # be estimated with
                                                    # variance.method = "tmle"
```

```
summary(CDE_ltmle_M0_death)
# Parameter Estimate:  0.056766
#  Estimated Std Err:  0.018037
#            p-value:  0.0016488
#  95% Conf Interval: (0.021413, 0.092118)

## CDE, setting M=1
set.seed(42) # for reproducibility
CDE_ltmle_M1_death <- ltmle(data = data_binary,
                            Anodes = c("A0_ace", "M_smoking"),
                            Lnodes = c("L1"), # intermediate confounders +/- baseline
                            Ynodes = c("Y_death"),
                            survivalOutcome = FALSE, # TRUE for time-to-event outcomes
                            Qform = Qform,
                            gform = gform,
                            abar = list(c(1,1), # counterfactual intervention do(A=1,M
                                        c(0,1)), # counterfactual intervention do(A=0,
                            SL.library = SL.library,
                            estimate.time = FALSE, # estimate computation time?
                            gcomp = FALSE,
                            variance.method = "ic")
summary(CDE_ltmle_M1_death)
# Parameter Estimate:  0.094776
#  Estimated Std Err:  0.024
#            p-value:  7.8496e-05
#  95% Conf Interval: (0.047736, 0.14182)
```

The controlled direct effect of ACE on the probability of death, had the mediator been set to 0 for every participant is 5.68%, 95%CI=[2.14%, 9.21%].

The controlled direct effect of ACE on the probability of death, had the mediator been set to 1 for every participant is 9.48%, 95%CI=[4.77%, 14.18%].

9.2.2 For continuous outcomes

```
# Define the data set with the continuous outcome Y_qol
data_continuous <- subset(df2_int, select = c(L0_male, L0_parent_low_educ_lv,
                                              A0_ace, L1,
                                              M_smoking, Y_qol))

# Replace the Qbar function (the 2d formula should be named Y_qol instead of Y_death)
Qform <- c(L1="Q.kplus1 ~ L0_male + L0_parent_low_educ_lv + A0_ace",
           Y_qol="Q.kplus1 ~ L0_male + L0_parent_low_educ_lv + L1 +
```

```
                    A0_ace * M_smoking")

set.seed(42)
## CDE, setting M=0
CDE_ltmle_M0_qol <- ltmle(data = data_continuous,
                      Anodes = c("A0_ace", "M_smoking"),
                      Lnodes = c("L1"), # intermediate confounders +/- baseline confounders
                      Ynodes = c("Y_qol"),
                      survivalOutcome = FALSE, # TRUE for time-to-event outcomes Y
                      Qform = Qform,
                      gform = gform,
                      abar = list(c(1,0), # counterfactual intervention do(A=1,M=0)
                                  c(0,0)), # counterfactual intervention do(A=0,M=0)
                      SL.library = SL.library,
                      estimate.time = FALSE, # estimate computation time?
                      gcomp = FALSE,
                      variance.method = "ic")
summary(CDE_ltmle_M0_qol)
# Additive Treatment Effect:
#   Parameter Estimate:  -4.8023
#    Estimated Std Err:  0.43135
#              p-value:  <2e-16
#    95% Conf Interval: (-5.6477, -3.9569)

## CDE, setting M=1
set.seed(42)
CDE_ltmle_M1_qol <- ltmle(data = data_continuous,
                          Anodes = c("A0_ace", "M_smoking"),
                          Lnodes = c("L1"), # intermediate confounders +/- baseline
                          Ynodes = c("Y_qol"),
                          survivalOutcome = FALSE, # TRUE for time-to-event outcomes Y
                          Qform = Qform,
                          gform = gform,
                          abar = list(c(1,1), # counterfactual intervention do(A=1,M=1)
                                      c(0,1)), # counterfactual intervention do(A=0,M=1)
                          SL.library = SL.library,
                          estimate.time = FALSE, # estimate computation time?
                          gcomp = FALSE,
                          variance.method = "ic")
summary(CDE_ltmle_M1_qol)
# Additive Treatment Effect:
#   Parameter Estimate:  -10.219
#    Estimated Std Err:  0.544
#              p-value:  <2e-16
#    95% Conf Interval: (-11.285, -9.1523)
```

The controlled direct effect of ACE on the quality of life score, had the mediator been set to 0 for every participant is -4.8, 95%CI=[-5.6, -4.0].

The controlled direct effect of ACE on the quality of life score, had the mediator been set to 1 for every participant is -10.2, 95%CI=[-11.3, -9.2].

Chapter 10

Appendix A: Data generating mechanisms

The data generating mechanisms are characterized by a causal model and a statistical model that generate data given in example.

In the first causal model, the mediator-outcome confounder $L(1)$ is not affected by the exposure. In the second causal model, the mediator-outcome confounder $L(1)$ is affected by the exposure.

10.1 First causal model: Data generating mechanism without mediator-outcome confounder affected by the exposure

This data generating mechanism is defined by the following set of structural equations:

$$
\begin{array}{lcl}
P(L(0)_{male} = 1) & = & p_{L(0)_{male}} \\
P(L(0)_{parent} = 1) & = & p_{L(0)_{parent}} \\
P(A_{ACE} = 1) & = & \beta_A + \beta^A_{male} \times L(0)_{male} + \beta^A_{parent} \times L(0)_{parent} \\
P(L(1) = 1) & = & p_{L(1)} \\
P(M_{smoking} = 1) & = & \beta_M + \beta^M_{male} \times L(0)_{male} + \beta^M_{parent} \times L(0)_{parent} + \beta^M_{L(1)} \times L(1) + \beta^M_A \times A_{ACE} \\
P(Y_{death} = 1) & = & \beta_Y + \beta^Y_{male} \times L(0)_{male} + \beta^Y_{parent} \times L(0)_{parent} + \beta^Y_{L(1)} \times L(1) \\
 & & +\beta^Y_A \times A_{ACE} + \beta^Y_M \times M_{smoking} + \beta^Y_{A*M} \times A_{ACE} \times M_{smoking} \\
\mathbb{E}(Y_{Qol} = 1) & = & \gamma_Y + \gamma^Y_{male} \times L(0)_{male} + \gamma^Y_{parent} \times L(0)_{parent} + \gamma^Y_{L(1)} \times L(1) \\
 & & +\gamma^Y_A \times A_{ACE} + \gamma^Y_M \times M_{smoking} + \gamma^Y_{A*M} \times A_{ACE} \times M_{smoking} + \varepsilon_Y
\end{array}
$$

where $\varepsilon_Y \sim \mathcal{N}(0, \sigma_Y = 10)$.

One can set the parameters of these structural equations using the following function `param.causal.model.1()`:

```
param.causal.model.1 <- function(A.M.interaction = NULL) {
# L0
p_L0_male <- 0.5
p_L0_parent_low_educ_lv <- 0.65

# A: A0_ace <- rbinom( 0.05 + 0.04 * L0_male + 0.06 * L0_parent_low_educ_lv )
b_A <- 0.05   # reference prevalence is 5%
b_male_A <- 0.04  # + 0.04 for the effect of L0_male -> A0_ace
b_parent_educ_A <- 0.06  # +0.06 for the effect of L0_parent_low_educ_lv -> A0_ace

# L1: intermediate confounder between M and Y, not influenced by A
p_L1 <- 0.3

# M: M_smoking <- rbinom( 0.2 + 0.05 * L0_male + 0.06 * L0_parent_low_educ_lv + 0.07 *
#                         0.1 * A0_ace )
b_M <- 0.2 # reference prevalence is 20%
b_male_M <- 0.05 # +0.05 for the effect of L0_male -> M_smoking
b_parent_educ_M <- 0.06 # +0.06 for the effect of L0_parent_low_educ_lv -> M_smoking
b_L1_M <- 0.07 # +0.07 for the effect of L1 -> M_smoking
b_A_M <- 0.1 # +0.10 for the effect of A0_ace -> M_smoking

# Y binary: rbinom( 0.10 + 0.06 * L0_male + 0.04 * L0_parent_low_educ_lv + 0.05 * A0_a
#                   0.07 * L1 + 0.08 * M_smoking +
#                   0.03 * A0_ace * M_smoking * A.M.inter )
b_Y <- 0.1 # reference prevalence is 10%
b_male_Y <- 0.06 # +0.06 for the effect of L0_male -> Y
b_parent_educ_Y <- 0.04 # +0.04 for the effect of L0_parent_low_educ_lv -> Y
b_A_Y <- 0.05 # 0.05 for the effect of A0_ace -> Y
b_L1_Y <- 0.07 # +0.07 for the effect of L1 -> Y
b_M_Y <- 0.08 # 0.08 for the effect of M_smoking -> Y
b_AM_Y <- 0.03 # 0.03 for the interaction effect A0_ace * M_smoking -> Y

# Y continuous: (75 - 1 * L0_male - 3 * L0_parent_low_educ_lv - 4 * A0_ace -3.5 * L1 -
#                9 * M_smoking -5 * A0_ace * M_smoking * A.M.inter ) +
#                rnorm(N, mean = 0, sd = 10)
mu_Y <- 75 # reference mean for QoL
c_male_Y <- -1 # -1 for the effect of L0_male -> Y
c_parent_educ_Y <- -3 # -3 for the effect of L0_parent_low_educ_lv -> Y
c_A_Y <- -4 # -4 for the effect of A0_ace -> Y
c_L1_Y <- -3.5 # -3.5 for the effect of L1 -> Y
c_M_Y <- -9 # -9 for the effect of M_smoking -> Y
c_AM_Y <- -5  # - 5 for the interaction effect A0_ace * M_smoking  -> Y
sd_Y <- 10 # standard deviation of the residuals
```

```
# A*M interaction ?
A.M.inter <- A.M.interaction

coef <- c( p_L0_male = p_L0_male, p_L0_parent_low_educ_lv = p_L0_parent_low_educ_lv,
           b_A = b_A, b_male_A = b_male_A, b_parent_educ_A = b_parent_educ_A,
           p_L1 = p_L1,
           b_M = b_M, b_male_M = b_male_M, b_parent_educ_M = b_parent_educ_M,
            b_L1_M = b_L1_M, b_A_M = b_A_M,
           b_Y = b_Y, b_male_Y = b_male_Y, b_parent_educ_Y = b_parent_educ_Y,
            b_A_Y = b_A_Y, b_L1_Y = b_L1_Y, b_M_Y = b_M_Y, b_AM_Y = b_AM_Y,
           mu_Y = mu_Y, c_male_Y = c_male_Y, c_parent_educ_Y = c_parent_educ_Y,
            c_A_Y = c_A_Y, c_L1_Y = c_L1_Y, c_M_Y = c_M_Y, c_AM_Y = c_AM_Y,
           sd_Y = sd_Y, A.M.inter = A.M.inter)

return(coef)
}
```

10.2 Second causal model: Data generating mechanism with mediator-outcome confounder affected by the exposure

This data generating mechanism is defined by the following set of structural equations:

$$
\begin{array}{lcl}
P(L(0)_{male} = 1) & = & p_{L(0)_{male}} \\
P(L(0)_{parent} = 1) & = & p_{L(0)_{parent}} \\
P(A_{ACE} = 1) & = & \beta_A + \beta^A_{male} \times L(0)_{male} + \beta^A_{parent} \times L(0)_{parent} \\
P(L(1) = 1) & = & \beta_{L(1)} + \beta^{L(1)}_{male} \times L(0)_{male} + \beta^{L(1)}_{parent} \times L(0)_{parent} + \beta^{L(1)}_{A} \times A_{ACE} \\
P(M_{smoking} = 1) & = & \beta_M + \beta^M_{male} \times L(0)_{male} + \beta^M_{parent} \times L(0)_{parent} + \beta^M_{L(1)} \times L(1) + \beta^M_A \times A_{ACE} \\
P(Y_{death} = 1) & = & \beta_Y + \beta^Y_{male} \times L(0)_{male} + \beta^Y_{parent} \times L(0)_{parent} + \beta^Y_{L(1)} \times L(1) \\
 & & + \beta^Y_A \times A_{ACE} + \beta^Y_M \times M_{smoking} + \beta^Y_{A*M} \times A_{ACE} \times M_{smoking} \\
\mathbb{E}(Y_{Qol} = 1) & = & \gamma_Y + \gamma^Y_{male} \times L(0)_{male} + \gamma^Y_{parent} \times L(0)_{parent} + \gamma^Y_{L(1)} \times L(1) \\
 & & + \gamma^Y_A \times A_{ACE} + \gamma^Y_M \times M_{smoking} + \gamma^Y_{A*M} \times A_{ACE} \times M_{smoking} + \varepsilon_Y
\end{array}
$$

where $\varepsilon_Y \sim \mathcal{N}(0, \sigma_Y = 10)$.

One can set the parameters of these structural equations using the following function `param.causal.model.2()`:

```
param.causal.model.2 <- function(A.M.interaction = NULL) {
# L0
p_L0_male <- 0.5
```

```
p_L0_parent_low_educ_lv <- 0.65

# A: A0_ace <- rbinom( 0.05 + 0.04 * L0_male + 0.06 * L0_parent_low_educ_lv )
b_A <- 0.05   # reference prevalence is 5%
b_male_A <- 0.04  # + 0.04 for the effect of L0_male -> A0_ace
b_parent_educ_A <- 0.06  # +0.06 for the effect of L0_parent_low_educ_lv -> A0_ace

# L1: L1 <- rbinom( 0.30 - 0.05 * L0_male + 0.08 * L0_parent_low_educ_lv +
#                   0.2 * A0_ace )
b_L1 <- 0.30   # reference prevalence is 30%
b_male_L1 <- -0.05  # - 0.05 for the effect of L0_male -> L1
b_parent_L1 <- +0.08 # + 0.08 for the effect of L0_parent_low_educ_lv -> L1
b_A_L1 <- +0.2 # +0.2 for the effect of A0_ace -> L1

# M: M_smoking <- rbinom( 0.2 + 0.05 * L0_male + 0.06 * L0_parent_low_educ_lv +
#                         0.2 * L1 + 0.1 * A0_ace )
b_M <- 0.2 # reference prevalence is 20%
b_male_M <- 0.05 # +0.05 for the effect of L0_male -> M_smoking
b_parent_educ_M <- 0.06 # +0.06 for the effect of L0_parent_low_educ_lv -> M_smoking
b_A_M <- 0.1 # +0.10 for the effect of A0_ace -> M_smoking
b_L1_M <- 0.2 # +0.2 for the effect of L1 -> M_smoking

# Y binary: rbinom( 0.10 + 0.06 * L0_male + 0.04 * L0_parent_low_educ_lv +
#                   0.05 * A0_ace + 0.07 * L1 + 0.08 * M_smoking +
#                   0.03 * A0_ace * M_smoking * A.M.inter )
b_Y <- 0.1 # reference prevalence is 10%
b_male_Y <- 0.06 # +0.06 for the effect of L0_male -> Y
b_parent_educ_Y <- 0.04 # +0.04 for the effect of L0_parent_low_educ_lv -> Y
b_A_Y <- 0.05 # 0.05 for the effect of A0_ace -> Y
b_L1_Y <- 0.07 # +0.07 for the effect of L1 -> Y
b_M_Y <- 0.08 # 0.08 for the effect of M_smoking -> Y
b_AM_Y <- 0.03 # 0.03 for the interaction effect A0_ace * M_smoking -> Y

# Y continuous: (75 - 1 * L0_male - 3 * L0_parent_low_educ_lv - 4 * A0_ace +
#                -3.5 * L1 - 9 * M_smoking +
#                -5 * A0_ace * M_smoking * A.M.inter ) + rnorm(N, mean = 0, sd = 10)
mu_Y <- 75 # reference mean for QoL
c_male_Y <- -1 # -1 for the effect of L0_male -> Y
c_parent_educ_Y <- -3 # -3 for the effect of L0_parent_low_educ_lv -> Y
c_A_Y <- -4 # -4 for the effect of A0_ace -> Y
c_L1_Y <- -5 # -5 for the effect of L1 -> Y
c_M_Y <- -9 # -9 for the effect of M_smoking -> Y
c_AM_Y <- -5  # - 5 for the interaction effect A0_ace * M_smoking  -> Y
sd_Y <- 10 # standard deviation of the residuals
```

```
# A*M interaction ?
A.M.inter <- A.M.interaction

coef <- c( p_L0_male = p_L0_male, p_L0_parent_low_educ_lv = p_L0_parent_low_educ_lv,
           b_A = b_A, b_male_A = b_male_A, b_parent_educ_A = b_parent_educ_A,
           b_L1 = b_L1, b_male_L1 = b_male_L1, b_parent_L1 = b_parent_L1,
            b_A_L1 = b_A_L1,
           b_M = b_M, b_male_M = b_male_M, b_parent_educ_M = b_parent_educ_M,
            b_L1_M = b_L1_M, b_A_M = b_A_M,
           b_Y = b_Y, b_male_Y = b_male_Y, b_parent_educ_Y = b_parent_educ_Y,
            b_A_Y = b_A_Y, b_L1_Y = b_L1_Y, b_M_Y = b_M_Y, b_AM_Y = b_AM_Y,
           mu_Y = mu_Y, c_male_Y = c_male_Y, c_parent_educ_Y = c_parent_educ_Y,
            c_A_Y = c_A_Y, c_L1_Y = c_L1_Y, c_M_Y = c_M_Y, c_AM_Y = c_AM_Y,
              sd_Y = sd_Y, A.M.inter = A.M.inter)

  return(coef)
}
```

10.3 Simulation of the four data sets used in examples

10.3.1 Data sets generated from the causal model 1

The following function `gen.data.causal.model.1` can be used to simulate data sets using the parameters defined previously in the `param.causal.model.1` function.

```
gen.data.causal.model.1 <- function(N, A.M.inter) { # input parameters are the
  #   sample size N and the presence of A*M interaction with A.M.inter = 0 or 1

  b <- param.causal.model.1(A.M.interaction = A.M.inter)

  # baseline confounders: parent's educational level=L0_parent_low_educ_lv & sex=L0_male
  L0_male <- rbinom(N, size = 1, prob = b["p_L0_male"])
  L0_parent_low_educ_lv <- rbinom(N, size = 1, prob = b["p_L0_parent_low_educ_lv"])

  # exposure: A0_ace
  A0_ace <- rbinom(N, size = 1, prob =  b["b_A"] +
                     b["b_male_A"] * L0_male +
                     b["b_parent_educ_A"] * L0_parent_low_educ_lv )

  # intermediate confounder between M_smoking and Y, not affected by A0 L1
  L1 <- rbinom(N, size = 1, prob = b["p_L1"])
```

```r
  # mediator: M_smoking
  M_smoking <- rbinom(N, size = 1, prob = b["b_M"] +
                        b["b_male_M"] * L0_male +
                        b["b_parent_educ_M"] * L0_parent_low_educ_lv +
                        b["b_A_M"] * A0_ace +
                        b["b_L1_M"] * L1)

  # Y_death
  Y_death <- rbinom(N, size = 1, prob = b["b_Y"] +
                      b["b_male_Y"] * L0_male +
                      b["b_parent_educ_Y"] * L0_parent_low_educ_lv +
                      b["b_A_Y"] * A0_ace +
                      b["b_L1_Y"] * L1 +
                      b["b_M_Y"] * M_smoking +
                      b["b_AM_Y"] * A0_ace * M_smoking * A.M.inter )

  # Y_qol
  Y_qol <- ( b["mu_Y"] +
               b["c_male_Y"] * L0_male +
               b["c_parent_educ_Y"] * L0_parent_low_educ_lv +
               b["c_A_Y"] * A0_ace +
               b["c_L1_Y"] * L1 +
               b["c_M_Y"] * M_smoking +
               b["c_AM_Y"] * A0_ace * M_smoking * A.M.inter ) +
    rnorm(N, mean = 0, sd = b["sd_Y"])

  # data.frame
  data.sim <- data.frame(L0_male, L0_parent_low_educ_lv, A0_ace, L1, M_smoking,
                         Y_death, Y_qol)

  return( data.sim )
}
```

Applying a sample size N=10000, we generate the `df1.csv` and `df1_int.csv` data sets.

```r
set.seed(1234)
df1 <- gen.data.causal.model.1(N=10000, A.M.inter=0)
write.csv(df1, file = "data/df1.csv", row.names = FALSE)

set.seed(1234)
df1_int <- gen.data.causal.model.1(N=10000, A.M.inter=1)
write.csv(df1_int, file = "data/df1_int.csv", row.names = FALSE)
```

```
head(df1)
```

```
##   L0_male L0_parent_low_educ_lv A0_ace L1 M_smoking Y_death    Y_qol
## 1       0                     1      0  1         0       0 93.41819
## 2       1                     1      1  1         0       1 64.03221
## 3       1                     1      0  0         0       0 75.56249
## 4       1                     0      0  0         0       0 89.77055
## 5       1                     1      0  0         0       0 77.22353
## 6       1                     1      0  0         1       0 73.87975
```

```
head(df1_int)
```

```
##   L0_male L0_parent_low_educ_lv A0_ace L1 M_smoking Y_death    Y_qol
## 1       0                     1      0  1         0       0 93.41819
## 2       1                     1      1  1         0       1 64.03221
## 3       1                     1      0  0         0       0 75.56249
## 4       1                     0      0  0         0       0 89.77055
## 5       1                     1      0  0         0       0 77.22353
## 6       1                     1      0  0         1       0 73.87975
```

10.3.2 Data sets generated from the causal model 2

The following function `gen.data.causal.model.2` can be used to simulate data sets using the parameters defined previously in the `param.causal.model.2` function.

```
gen.data.causal.model.2 <- function(N, A.M.inter) { # input parameters are the
  #   sample size N and the presence of A*M interaction with A.M.inter = 0 or 1

  b <- param.causal.model.2(A.M.interaction = A.M.inter)

  # baseline confounders: parent's educational level=L0_parent_low_educ_lv & sex=L0_male
  L0_male <- rbinom(N, size = 1, prob = b["p_L0_male"])
  L0_parent_low_educ_lv <- rbinom(N, size = 1, prob = b["p_L0_parent_low_educ_lv"])

  # exposure: A0_ace
  A0_ace <- rbinom(N, size = 1, prob =  b["b_A"] +
                     b["b_male_A"] * L0_male +
                     b["b_parent_educ_A"] * L0_parent_low_educ_lv )

  # intermediate confounder between M_smoking and Y,
  L1 <- rbinom(N, size = 1, prob = b["b_L1"] +
                 b["b_male_L1"] * L0_male +
```

```r
                 b["b_parent_L1"] * L0_parent_low_educ_lv +
                 b["b_A_L1"]* A0_ace)

  # mediator: M_smoking
  M_smoking <- rbinom(N, size = 1, prob = b["b_M"] +
                        b["b_male_M"] * L0_male +
                        b["b_parent_educ_M"] * L0_parent_low_educ_lv +
                        b["b_A_M"] * A0_ace +
                        b["b_L1_M"] * L1)

  # Y_death
  Y_death <- rbinom(N, size = 1, prob = b["b_Y"] +
                      b["b_male_Y"] * L0_male +
                      b["b_parent_educ_Y"] * L0_parent_low_educ_lv +
                      b["b_A_Y"] * A0_ace +
                      b["b_L1_Y"] * L1 +
                      b["b_M_Y"] * M_smoking +
                      b["b_AM_Y"] * A0_ace * M_smoking * A.M.inter )

  # Y_qol
  Y_qol <- ( b["mu_Y"] +
               b["c_male_Y"] * L0_male +
               b["c_parent_educ_Y"] * L0_parent_low_educ_lv +
               b["c_A_Y"] * A0_ace +
               b["c_L1_Y"] * L1 +
               b["c_M_Y"] * M_smoking +
               b["c_AM_Y"] * A0_ace * M_smoking * A.M.inter ) +
    rnorm(N, mean = 0, sd = b["sd_Y"])

  # data.frame
  data.sim <- data.frame(L0_male, L0_parent_low_educ_lv, A0_ace, L1, M_smoking,
                         Y_death, Y_qol)

  return( data.sim )
}
```

Applying a sample size N=10000, we generate the `df2.csv` and `df2_int.csv` data sets.

```r
set.seed(1234)
df2 <- gen.data.causal.model.2(N=10000, A.M.inter=0)
write.csv(df2, file = "data/df2.csv", row.names = FALSE)

set.seed(1234)
df2_int <- gen.data.causal.model.2(N=10000, A.M.inter=1)
```

```r
write.csv(df2_int, file = "data/df2_int.csv", row.names = FALSE)
```

```r
tail(df2)
```

```
##       L0_male L0_parent_low_educ_lv A0_ace L1 M_smoking Y_death    Y_qol
## 9995        0                     1      0  1         1       0 53.25115
## 9996        0                     1      0  0         1       0 66.36484
## 9997        0                     1      1  1         1       0 74.20579
## 9998        1                     1      0  0         1       0 41.30248
## 9999        0                     0      0  1         0       0 85.60169
## 10000       1                     1      0  0         0       0 61.56969
```

```r
tail(df2_int)
```

```
##       L0_male L0_parent_low_educ_lv A0_ace L1 M_smoking Y_death    Y_qol
## 9995        0                     1      0  1         1       0 53.25115
## 9996        0                     1      0  0         1       0 66.36484
## 9997        0                     1      1  1         1       0 69.20579
## 9998        1                     1      0  0         1       0 41.30248
## 9999        0                     0      0  1         0       0 85.60169
## 10000       1                     1      0  0         0       0 61.56969
```

Chapter 11

Appendix B: Calculation of the true causal quantities

11.1 True causal quantities without mediator-ouctome confounder affected by the exposure

11.1.1 Average total effects (ATE)

The following function `true.ATE1` can be used to run the calculation for the average total effects (ATE).

```
true.ATE1 <- function(interaction = NULL) {
  b <- param.causal.model.1(A.M.interaction = interaction)

  # binary outcome (death)
  S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1), c(0,1)), rep(NA,n=2^4))
  colnames(S) <- list("male","parent_educ","L1","M","sum")
  for (n in 1:16) {
    S[n,"sum"] <- ( ( ( b["b_Y"] +
                      b["b_male_Y"] * S[n,"male"] +
                      b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                      b["b_A_Y"] * 1 +
                      b["b_L1_Y"] * S[n,"L1"] +
                      b["b_M_Y"] * S[n,"M"] +
                      b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
                    (( b["b_M"] +
                         b["b_male_M"] * S[n,"male"] +
```

```
                           b["b_parent_educ_M"] * S[n,"parent_educ"] +
                           b["b_L1_M"] * S[n,"L1"] +
                           b["b_A_M"] * 1 )^( S[n,"M"] )) *
                    (( 1 - (b["b_M"] +
                              b["b_male_M"] * S[n,"male"] +
                              b["b_parent_educ_M"] * S[n,"parent_educ"] +
                              b["b_L1_M"] * S[n,"L1"] +
                              b["b_A_M"] * 1) )^( 1 - S[n,"M"] )) ) -
                    ( ( b["b_Y"] +
                          b["b_male_Y"] * S[n,"male"] +
                          b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                          b["b_A_Y"] * 0 +
                          b["b_L1_Y"] * S[n,"L1"] +
                          b["b_M_Y"] * S[n,"M"] +
                          b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
                        (( b["b_M"] +
                             b["b_male_M"] * S[n,"male"] +
                             b["b_parent_educ_M"] * S[n,"parent_educ"] +
                             b["b_L1_M"] * S[n,"L1"] +
                             b["b_A_M"] * 0 )^( S[n,"M"] )) *
                        (( 1 - (b["b_M"] +
                                  b["b_male_M"] * S[n,"male"] +
                                  b["b_parent_educ_M"] * S[n,"parent_educ"] +
                                  b["b_L1_M"] * S[n,"L1"] +
                                  b["b_A_M"] * 0) )^( 1 - S[n,"M"] )) ) ) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
}

ATE.death <- sum(S[,"sum"])


# quantitative outcome (QoL)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1), c(0,1)), rep(NA,n=2^4))
colnames(S) <- list("male","parent_educ","L1","M","sum")
for (n in 1:16) {
  S[n,"sum"] <- ( ( ( b["mu_Y"] +
                        b["c_male_Y"] * S[n,"male"] +
                        b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                        b["c_A_Y"] * 1 +
                        b["c_L1_Y"] * S[n,"L1"] +
```

```
                        b["c_M_Y"] * S[n,"M"] +
                        b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
                      (( b["b_M"] +
                           b["b_male_M"] * S[n,"male"] +
                           b["b_parent_educ_M"] * S[n,"parent_educ"] +
                           b["b_L1_M"] * S[n,"L1"] +
                           b["b_A_M"] * 1 )^( S[n,"M"] )) *
                      (( 1 - (b["b_M"] +
                                b["b_male_M"] * S[n,"male"] +
                                b["b_parent_educ_M"] * S[n,"parent_educ"] +
                                b["b_L1_M"] * S[n,"L1"] +
                                b["b_A_M"] * 1) )^( 1 - S[n,"M"] )) ) -
                    ( ( b["mu_Y"] +
                          b["c_male_Y"] * S[n,"male"] +
                          b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                          b["c_A_Y"] * 0 +
                          b["c_L1_Y"] * S[n,"L1"] +
                          b["c_M_Y"] * S[n,"M"] +
                          b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
                        (( b["b_M"] +
                             b["b_male_M"] * S[n,"male"] +
                             b["b_parent_educ_M"] * S[n,"parent_educ"] +
                             b["b_L1_M"] * S[n,"L1"] +
                             b["b_A_M"] * 0 )^( S[n,"M"] )) *
                        (( 1 - (b["b_M"] +
                                  b["b_male_M"] * S[n,"male"] +
                                  b["b_parent_educ_M"] * S[n,"parent_educ"] +
                                  b["b_L1_M"] * S[n,"L1"] +
                                  b["b_A_M"] * 0) )^( 1 - S[n,"M"] )) ) ) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
}

ATE.qol <- sum(S[,"sum"])

return(list(ATE.death = ATE.death, ATE.qol = ATE.qol))

}
```

```
true.ATE1.no.inter <- true.ATE1(interaction = 0)
```

```
true.ATE1.with.inter <- true.ATE1(interaction = 1)
```

The average total effects ATE $= \mathbb{E}(Y_1) - \mathbb{E}(Y_0)$ are:

- 0.058 for death and -4.9 for quality of life without interaction;
- 0.06955 for death and -6.825 for quality of life with interaction.

11.1.2 Controlled direct effects (CDE)

The following function `true.CDE1` can be used to run the calculation for controlled direct effects (CDE).

```
true.CDE1 <- function(interaction = NULL) {
  b <- param.causal.model.1(A.M.interaction = interaction)

  # binary outcome (death)
  # we estimate both CDE, fixing do(M) = 0 et do(M) = 1 and
  # using the corresponding lines in the S matrix
  S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^3))
  colnames(S) <- list("male","parent_educ","L1","M","sum")
  for (n in 1:16) {
    S[n,"sum"] <- ( ( b["b_Y"] +
                      b["b_male_Y"] * S[n,"male"] +
                      b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                      b["b_A_Y"] * 1 +
                      b["b_L1_Y"] * S[n,"L1"] +
                      b["b_M_Y"] * S[n,"M"] +
                      b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) -
                    ( b["b_Y"] +
                      b["b_male_Y"] * S[n,"male"] +
                      b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                      b["b_A_Y"] * 0 +
                      b["b_L1_Y"] * S[n,"L1"] +
                      b["b_M_Y"] * S[n,"M"] +
                      b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) ) *
      ((b["p_L1"])^(S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }

  CDE.M0.death <- sum(S[1:8,"sum"])
```

```r
  CDE.M1.death <- sum(S[9:16,"sum"])

  # quantitative outcome (QoL)
  # we estimate both CDE, fixing do(M) = 0 et do(M) = 1 and using
  # the corresponding lines in the S matrix
  for (n in 1:16) {
    S[n,"sum"] <- ( ( b["mu_Y"] +
                        b["c_male_Y"] * S[n,"male"] +
                        b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                        b["c_A_Y"] * 1 +
                        b["c_L1_Y"] * S[n,"L1"] +
                        b["c_M_Y"] * S[n,"M"] +
                        b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) -
                      ( b["mu_Y"] +
                          b["c_male_Y"] * S[n,"male"] +
                          b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                          b["c_A_Y"] * 0 +
                          b["c_L1_Y"] * S[n,"L1"] +
                          b["c_M_Y"] * S[n,"M"] +
                          b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) ) *
      ((b["p_L1"])^(S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }

  CDE.M0.qol <- sum(S[1:8,"sum"])
  CDE.M1.qol <- sum(S[9:16,"sum"])

  return(list(CDE.M0.death = CDE.M0.death, CDE.M1.death = CDE.M1.death,
              CDE.M0.qol = CDE.M0.qol, CDE.M1.qol = CDE.M1.qol))
}
```

```r
true.CDE1.no.inter <- true.CDE1(interaction = 0)

true.CDE1.with.inter <- true.CDE1(interaction = 1)
```

Setting $do(M = 0)$, the controlled direct effects $\text{CDE}_{M=0} = \mathbb{E}\left(Y_{1,0}\right) - \mathbb{E}\left(Y_{0,0}\right)$ are:

- 0.05 for death and -4 for quality of life without interaction,
- 0.05 for death and -4 for quality of life with interaction.

Setting $do(M = 1)$, the controlled direct effects $\text{CDE}_{M=1} = \mathbb{E}\left(Y_{1,1}\right) - \mathbb{E}\left(Y_{0,1}\right)$ are:

- 0.05 for death and -4 for quality of life without interaction,
- 0.08 for death and -9 for quality of life with interaction.

11.1.3 Pure natural direct effect and Total natural indirect effect

The following function `true.PNDE.TNIE1` can be used to run the calculation for pure natural direct effects (PNDE) and total natural indirect effects (TNIE).

```
true.PNDE.TNIE1 <- function(interaction = NULL) {
  b <- param.causal.model.1(A.M.interaction = interaction)

  # binary outcome (death)
  S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4), rep(NA,n=2^4))
  colnames(S) <- list("male","parent_educ","L1","M","sum.pnde", "sum.tnie")
  for (n in 1:16) {
    # PNDE
    S[n,"sum.pnde"] <- ( ( b["b_Y"] +
                             b["b_male_Y"] * S[n,"male"] +
                             b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                             b["b_A_Y"] * 1 +
                             b["b_L1_Y"] * S[n,"L1"] +
                             b["b_M_Y"] * S[n,"M"] +
                             b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) -
                           ( b["b_Y"] +
                               b["b_male_Y"] * S[n,"male"] +
                               b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                               b["b_A_Y"] * 0 +
                               b["b_L1_Y"] * S[n,"L1"] +
                               b["b_M_Y"] * S[n,"M"] +
                               b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) ) *
      (( b["b_M"] +
           b["b_male_M"] * S[n,"male"] +
           b["b_parent_educ_M"] * S[n,"parent_educ"] +
           b["b_L1_M"] * S[n,"L1"] +
           b["b_A_M"] * 0 )^( S[n,"M"] )) *
      (( 1 - (b["b_M"] +
                b["b_male_M"] * S[n,"male"] +
                b["b_parent_educ_M"] * S[n,"parent_educ"] +
                b["b_L1_M"] * S[n,"L1"] +
                b["b_A_M"] * 0) )^( 1 - S[n,"M"] ))  *
```

```
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  # TNIE
  S[n,"sum.tnie"] <- ( b["b_Y"] +
                         b["b_male_Y"] * S[n,"male"] +
                         b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                         b["b_A_Y"] * 1 +
                         b["b_L1_Y"] * S[n,"L1"] +
                         b["b_M_Y"] * S[n,"M"] +
                         b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
    ( (( b["b_M"] +
           b["b_male_M"] * S[n,"male"] +
           b["b_parent_educ_M"] * S[n,"parent_educ"] +
           b["b_L1_M"] * S[n,"L1"] +
           b["b_A_M"] * 1 )^( S[n,"M"] )) +
        (( 1 - (b["b_M"] +
                  b["b_male_M"] * S[n,"male"] +
                  b["b_parent_educ_M"] * S[n,"parent_educ"] +
                  b["b_L1_M"] * S[n,"L1"] +
                  b["b_A_M"] * 1) )^( 1 - S[n,"M"] )) -
        (( b["b_M"] +
             b["b_male_M"] * S[n,"male"] +
             b["b_parent_educ_M"] * S[n,"parent_educ"] +
             b["b_L1_M"] * S[n,"L1"] +
             b["b_A_M"] * 0 )^( S[n,"M"] )) -
        (( 1 - (b["b_M"] +
                  b["b_male_M"] * S[n,"male"] +
                  b["b_parent_educ_M"] * S[n,"parent_educ"] +
                  b["b_L1_M"] * S[n,"L1"] +
                  b["b_A_M"] * 0) )^( 1 - S[n,"M"] )) ) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }

PNDE.death <- sum(S[,"sum.pnde"])
TNIE.death <- sum(S[,"sum.tnie"])
```

```
# quantitative outcome (QoL)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4), rep(NA,n=2^4))
colnames(S) <- list("male","parent_educ","L1","M","sum.pnde", "sum.tnie")
for (n in 1:16) {
  # PNDE
  S[n,"sum.pnde"] <- ( ( b["mu_Y"] +
                           b["c_male_Y"] * S[n,"male"] +
                           b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["c_A_Y"] * 1 +
                           b["c_L1_Y"] * S[n,"L1"] +
                           b["c_M_Y"] * S[n,"M"] +
                           b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) -
                         ( b["mu_Y"] +
                             b["c_male_Y"] * S[n,"male"] +
                             b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                             b["c_A_Y"] * 0 +
                             b["c_L1_Y"] * S[n,"L1"] +
                             b["c_M_Y"] * S[n,"M"] +
                             b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) ) *
    (( b["b_M"] +
         b["b_male_M"] * S[n,"male"] +
         b["b_parent_educ_M"] * S[n,"parent_educ"] +
         b["b_L1_M"] * S[n,"L1"] +
         b["b_A_M"] * 0 )^( S[n,"M"] )) *
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 0) )^( 1 - S[n,"M"] ))  *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  # TNIE
  S[n,"sum.tnie"] <- ( b["mu_Y"] +
                         b["c_male_Y"] * S[n,"male"] +
                         b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                         b["c_A_Y"] * 1 +
                         b["c_L1_Y"] * S[n,"L1"] +
                         b["c_M_Y"] * S[n,"M"] +
                         b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
```

```
      ( (( b["b_M"] +
             b["b_male_M"] * S[n,"male"] +
             b["b_parent_educ_M"] * S[n,"parent_educ"] +
             b["b_L1_M"] * S[n,"L1"] +
             b["b_A_M"] * 1 )^( S[n,"M"] )) +
          (( 1 - (b["b_M"] +
                    b["b_male_M"] * S[n,"male"] +
                    b["b_parent_educ_M"] * S[n,"parent_educ"] +
                    b["b_L1_M"] * S[n,"L1"] +
                    b["b_A_M"] * 1) )^( 1 - S[n,"M"] )) -
          (( b["b_M"] +
               b["b_male_M"] * S[n,"male"] +
               b["b_parent_educ_M"] * S[n,"parent_educ"] +
               b["b_L1_M"] * S[n,"L1"] +
               b["b_A_M"] * 0 )^( S[n,"M"] )) -
          (( 1 - (b["b_M"] +
                    b["b_male_M"] * S[n,"male"] +
                    b["b_parent_educ_M"] * S[n,"parent_educ"] +
                    b["b_L1_M"] * S[n,"L1"] +
                    b["b_A_M"] * 0) )^( 1 - S[n,"M"] )) ) *
      ((b["p_L1"])^(S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
    }

  PNDE.qol <- sum(S[,"sum.pnde"])
  TNIE.qol <- sum(S[,"sum.tnie"])

  return(list(PNDE.death = PNDE.death, TNIE.death = TNIE.death,
              PNDE.qol = PNDE.qol, TNIE.qol = TNIE.qol))
}
```

```
true.PNDE.TNIE.no.inter <- true.PNDE.TNIE1(interaction = 0)

true.PNDE.TNIE.with.inter <- true.PNDE.TNIE1(interaction = 1)
```

The PNDE $= \mathbb{E}\left(Y_{1,M_0}\right) - \mathbb{E}\left(Y_{0,M_0}\right)$ and TNIE $= \mathbb{E}\left(Y_{1,M_1}\right) - \mathbb{E}\left(Y_{1,M_0}\right)$ are respectively:

- 0.05 and 0.00800000000000001 for death without interaction,
- 0.05855 and 0.011 for death with interaction,
- -4 and -0.9 for quality of life without interaction,

- -5.425 and -1.4 for quality of life with interaction.

11.1.4 Total natural direct effect and Pure natural indirect effect

The following function `true.TNDE.PNIE1` can be used to run the calculation for total natural direct effects (TNDE) and pure natural indirect effects (PNIE).

```
true.TNDE.PNIE1 <- function(interaction = NULL) {
  b <- param.causal.model.1(A.M.interaction = interaction)

  # binary outcome (death)
  S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4), rep(NA,n=2^4))
  colnames(S) <- list("male","parent_educ","L1","M","sum.tnde", "sum.pnie")

  for (n in 1:16) {
    # TNDE
    S[n,"sum.tnde"] <- ( ( b["b_Y"] +
                             b["b_male_Y"] * S[n,"male"] +
                             b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                             b["b_A_Y"] * 1 +
                             b["b_L1_Y"] * S[n,"L1"] +
                             b["b_M_Y"] * S[n,"M"] +
                             b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) -
                           ( b["b_Y"] +
                               b["b_male_Y"] * S[n,"male"] +
                               b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                               b["b_A_Y"] * 0 +
                               b["b_L1_Y"] * S[n,"L1"] +
                               b["b_M_Y"] * S[n,"M"] +
                               b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) ) *
      (( b["b_M"] +
           b["b_male_M"] * S[n,"male"] +
           b["b_parent_educ_M"] * S[n,"parent_educ"] +
           b["b_L1_M"] * S[n,"L1"] +
           b["b_A_M"] * 1 )^( S[n,"M"] )) *
      (( 1 - (b["b_M"] +
                b["b_male_M"] * S[n,"male"] +
                b["b_parent_educ_M"] * S[n,"parent_educ"] +
                b["b_L1_M"] * S[n,"L1"] +
                b["b_A_M"] * 1) )^( 1 - S[n,"M"] ))  *
      ((b["p_L1"])^(S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
```

```
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

    # PNIE
    S[n,"sum.pnie"] <- ( b["b_Y"] +
                           b["b_male_Y"] * S[n,"male"] +
                           b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["b_A_Y"] * 0 +
                           b["b_L1_Y"] * S[n,"L1"] +
                           b["b_M_Y"] * S[n,"M"] +
                           b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
      ( (( b["b_M"] +
             b["b_male_M"] * S[n,"male"] +
             b["b_parent_educ_M"] * S[n,"parent_educ"] +
             b["b_L1_M"] * S[n,"L1"] +
             b["b_A_M"] * 1 )^( S[n,"M"] )) +
          (( 1 - (b["b_M"] +
                    b["b_male_M"] * S[n,"male"] +
                    b["b_parent_educ_M"] * S[n,"parent_educ"] +
                    b["b_L1_M"] * S[n,"L1"] +
                    b["b_A_M"] * 1) )^( 1 - S[n,"M"] )) -
          (( b["b_M"] +
               b["b_male_M"] * S[n,"male"] +
               b["b_parent_educ_M"] * S[n,"parent_educ"] +
               b["b_L1_M"] * S[n,"L1"] +
               b["b_A_M"] * 0 )^( S[n,"M"] )) -
          (( 1 - (b["b_M"] +
                    b["b_male_M"] * S[n,"male"] +
                    b["b_parent_educ_M"] * S[n,"parent_educ"] +
                    b["b_L1_M"] * S[n,"L1"] +
                    b["b_A_M"] * 0) )^( 1 - S[n,"M"] )) ) *
      ((b["p_L1"])^(S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
    }

  TNDE.death <- sum(S[,"sum.tnde"])
  PNIE.death <- sum(S[,"sum.pnie"])

  # quantitative outcome (QoL)
  S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4), rep(NA,n=2^4))
  colnames(S) <- list("male","parent_educ","L1","M","sum.tnde", "sum.pnie")
```

```
for (n in 1:16) {
  # TNDE
  S[n,"sum.tnde"] <- ( ( b["mu_Y"] +
                         b["c_male_Y"] * S[n,"male"] +
                         b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                         b["c_A_Y"] * 1 +
                         b["c_L1_Y"] * S[n,"L1"] +
                         b["c_M_Y"] * S[n,"M"] +
                         b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) -
                       ( b["mu_Y"] +
                           b["c_male_Y"] * S[n,"male"] +
                           b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["c_A_Y"] * 0 +
                           b["c_L1_Y"] * S[n,"L1"] +
                           b["c_M_Y"] * S[n,"M"] +
                           b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) ) *
    (( b["b_M"] +
         b["b_male_M"] * S[n,"male"] +
         b["b_parent_educ_M"] * S[n,"parent_educ"] +
         b["b_L1_M"] * S[n,"L1"] +
         b["b_A_M"] * 1 )^( S[n,"M"] )) *
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 1) )^( 1 - S[n,"M"] ))  *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  # PNIE
  S[n,"sum.pnie"] <- ( b["mu_Y"] +
                       b["c_male_Y"] * S[n,"male"] +
                       b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                       b["c_A_Y"] * 0 +
                       b["c_L1_Y"] * S[n,"L1"] +
                       b["c_M_Y"] * S[n,"M"] +
                       b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
    ( (( b["b_M"] +
           b["b_male_M"] * S[n,"male"] +
           b["b_parent_educ_M"] * S[n,"parent_educ"] +
           b["b_L1_M"] * S[n,"L1"] +
```

```
              b["b_A_M"] * 1 )^( S[n,"M"] )) +
           (( 1 - (b["b_M"] +
                     b["b_male_M"] * S[n,"male"] +
                     b["b_parent_educ_M"] * S[n,"parent_educ"] +
                     b["b_L1_M"] * S[n,"L1"] +
                     b["b_A_M"] * 1) )^( 1 - S[n,"M"] )) -
           (( b["b_M"] +
                b["b_male_M"] * S[n,"male"] +
                b["b_parent_educ_M"] * S[n,"parent_educ"] +
                b["b_L1_M"] * S[n,"L1"] +
                b["b_A_M"] * 0 )^( S[n,"M"] )) -
           (( 1 - (b["b_M"] +
                     b["b_male_M"] * S[n,"male"] +
                     b["b_parent_educ_M"] * S[n,"parent_educ"] +
                     b["b_L1_M"] * S[n,"L1"] +
                     b["b_A_M"] * 0) )^( 1 - S[n,"M"] )) ) *
       ((b["p_L1"])^(S[n,"L1"])) *
       ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
       ((b["p_L0_male"])^(S[n,"male"])) *
       ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
       ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
       ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
     }

  TNDE.qol <- sum(S[,"sum.tnde"])
  PNIE.qol <- sum(S[,"sum.pnie"])

  return(list(TNDE.death = TNDE.death, PNIE.death = PNIE.death,
              TNDE.qol = TNDE.qol, PNIE.qol = PNIE.qol))
}
```

```
true.TNDE.PNIE.no.inter <- true.TNDE.PNIE1(interaction = 0)

true.TNDE.PNIE.with.inter <- true.TNDE.PNIE1(interaction = 1)
```

The TNDE $= \mathbb{E}\left(Y_{1,M_1}\right) - \mathbb{E}\left(Y_{0,M_1}\right)$ and PNIE $= \mathbb{E}\left(Y_{0,M_1}\right) - \mathbb{E}\left(Y_{0,M_0}\right)$ are respectively:

- 0.05 and 0.00800000000000001 for death without interaction,
- 0.06155 and 0.00800000000000001 for death with interaction,
- -4 and -0.899999999999999 for quality of life without interaction,
- -5.925 and -0.899999999999999 for quality of life with interaction.

11.1.5 Vanderweele's 3-way decomposition

The following function `true.3way.decomp` can be used to run the calculation for the 3-way decomposition of the total effect into a "pure natural direct effect" (PNDE), a "pure natural indirect effect" (PNIE) and a "mediated interactive effect" (MIE).

```
true.3way.decomp <- function(interaction = NULL) {
  b <- param.causal.model.1(A.M.interaction = interaction)

  # binary outcome (death)
  S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4), rep(NA,n=2^4))
  colnames(S) <- list("male","parent_educ","L1","M","sum.pde", "sum.pie")
  for (n in 1:16) {
    # PDE
    S[n,"sum.pde"] <- ( ( b["b_Y"] +
                           b["b_male_Y"] * S[n,"male"] +
                           b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["b_A_Y"] * 1 +
                           b["b_L1_Y"] * S[n,"L1"] +
                           b["b_M_Y"] * S[n,"M"] +
                           b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) -
                         ( b["b_Y"] +
                             b["b_male_Y"] * S[n,"male"] +
                             b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                             b["b_A_Y"] * 0 +
                             b["b_L1_Y"] * S[n,"L1"] +
                             b["b_M_Y"] * S[n,"M"] +
                             b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) ) *
    (( b["b_M"] +
         b["b_male_M"] * S[n,"male"] +
         b["b_parent_educ_M"] * S[n,"parent_educ"] +
         b["b_L1_M"] * S[n,"L1"] +
         b["b_A_M"] * 0 )^( S[n,"M"] )) *
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 0) )^( 1 - S[n,"M"] ))  *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
```

```
  # PIE
  S[n,"sum.pie"] <- ( b["b_Y"] +
                        b["b_male_Y"] * S[n,"male"] +
                        b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                        b["b_A_Y"] * 0 +
                        b["b_L1_Y"] * S[n,"L1"] +
                        b["b_M_Y"] * S[n,"M"] +
                        b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
    ( (( b["b_M"] +
           b["b_male_M"] * S[n,"male"] +
           b["b_parent_educ_M"] * S[n,"parent_educ"] +
           b["b_L1_M"] * S[n,"L1"] +
           b["b_A_M"] * 1 )^( S[n,"M"] )) +
        (( 1 - (b["b_M"] +
                  b["b_male_M"] * S[n,"male"] +
                  b["b_parent_educ_M"] * S[n,"parent_educ"] +
                  b["b_L1_M"] * S[n,"L1"] +
                  b["b_A_M"] * 1) )^( 1 - S[n,"M"] )) -
        (( b["b_M"] +
             b["b_male_M"] * S[n,"male"] +
             b["b_parent_educ_M"] * S[n,"parent_educ"] +
             b["b_L1_M"] * S[n,"L1"] +
             b["b_A_M"] * 0 )^( S[n,"M"] )) -
        (( 1 - (b["b_M"] +
                  b["b_male_M"] * S[n,"male"] +
                  b["b_parent_educ_M"] * S[n,"parent_educ"] +
                  b["b_L1_M"] * S[n,"L1"] +
                  b["b_A_M"] * 0) )^( 1 - S[n,"M"] )) ) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }

# MI
S.MI <- cbind(expand.grid(c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4))
colnames(S.MI) <- list("male","parent_educ","L1", "sum.mi")
for (n in 1:8) {
  S.MI[n,"sum.mi"] <- ( ( b["b_Y"] +
                            b["b_male_Y"] * S.MI[n,"male"] +
                            b["b_parent_educ_Y"] * S.MI[n,"parent_educ"] +
                            b["b_A_Y"] * 1 +
                            b["b_L1_Y"] * S.MI[n,"L1"] +
```

```r
                              b["b_M_Y"] * 1 +
                              b["b_AM_Y"] * 1 * 1 * b["A.M.inter"] ) -
                            ( b["b_Y"] +
                                b["b_male_Y"] * S.MI[n,"male"] +
                                b["b_parent_educ_Y"] * S.MI[n,"parent_educ"] +
                                b["b_A_Y"] * 1 +
                                b["b_L1_Y"] * S.MI[n,"L1"] +
                                b["b_M_Y"] * 0 +
                                b["b_AM_Y"] * 1 * 0 * b["A.M.inter"] ) -
                            ( b["b_Y"] +
                                b["b_male_Y"] * S.MI[n,"male"] +
                                b["b_parent_educ_Y"] * S.MI[n,"parent_educ"] +
                                b["b_A_Y"] * 0 +
                                b["b_L1_Y"] * S.MI[n,"L1"] +
                                b["b_M_Y"] * 1 +
                                b["b_AM_Y"] * 0 * 1 * b["A.M.inter"] ) +
                            ( b["b_Y"] +
                                b["b_male_Y"] * S.MI[n,"male"] +
                                b["b_parent_educ_Y"] * S.MI[n,"parent_educ"] +
                                b["b_A_Y"] * 0 +
                                b["b_L1_Y"] * S.MI[n,"L1"] +
                                b["b_M_Y"] * 0 +
                                b["b_AM_Y"] * 0 * 0 * b["A.M.inter"] )) *
    ( ( b["b_M"] +
          b["b_male_M"] * S.MI[n,"male"] +
          b["b_parent_educ_M"] * S.MI[n,"parent_educ"] +
          b["b_L1_M"] * S[n,"L1"] +
          b["b_A_M"] * 1 ) -
        ( b["b_M"] +
            b["b_male_M"] * S.MI[n,"male"] +
            b["b_parent_educ_M"] * S.MI[n,"parent_educ"] +
            b["b_L1_M"] * S[n,"L1"] +
            b["b_A_M"] * 0 )) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S.MI[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S.MI[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S.MI[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S.MI[n,"parent_educ"]))
  }

PDE.death <- sum(S[,"sum.pde"])
PIE.death <- sum(S[,"sum.pie"])
MI.death <- sum(S.MI[,"sum.mi"])
```

```
# quantitative outcome (QoL)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4), rep(NA,n=2^4))
colnames(S) <- list("male","parent_educ","L1","M","sum.pde", "sum.pie")
for (n in 1:16) {
  # PDE
  S[n,"sum.pde"] <- ( ( b["mu_Y"] +
                        b["c_male_Y"] * S[n,"male"] +
                        b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                        b["c_A_Y"] * 1 +
                        b["c_L1_Y"] * S[n,"L1"] +
                        b["c_M_Y"] * S[n,"M"] +
                        b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) -
                      ( b["mu_Y"] +
                          b["c_male_Y"] * S[n,"male"] +
                          b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                          b["c_A_Y"] * 0 +
                          b["c_L1_Y"] * S[n,"L1"] +
                          b["c_M_Y"] * S[n,"M"] +
                          b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) ) *
    (( b["b_M"] +
         b["b_male_M"] * S[n,"male"] +
         b["b_parent_educ_M"] * S[n,"parent_educ"] +
         b["b_L1_M"] * S[n,"L1"] +
         b["b_A_M"] * 0 )^( S[n,"M"] )) *
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 0) )^( 1 - S[n,"M"] ))  *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  # PIE
  S[n,"sum.pie"] <- ( b["mu_Y"] +
                      b["c_male_Y"] * S[n,"male"] +
                      b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                      b["c_A_Y"] * 0 +
                      b["c_L1_Y"] * S[n,"L1"] +
                      b["c_M_Y"] * S[n,"M"] +
                      b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
    ( (( b["b_M"] +
```

```
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 1 )^( S[n,"M"] )) +
           (( 1 - (b["b_M"] +
                     b["b_male_M"] * S[n,"male"] +
                     b["b_parent_educ_M"] * S[n,"parent_educ"] +
                     b["b_L1_M"] * S[n,"L1"] +
                     b["b_A_M"] * 1) )^( 1 - S[n,"M"] )) -
           (( b["b_M"] +
                b["b_male_M"] * S[n,"male"] +
                b["b_parent_educ_M"] * S[n,"parent_educ"] +
                b["b_L1_M"] * S[n,"L1"] +
                b["b_A_M"] * 0 )^( S[n,"M"] )) -
           (( 1 - (b["b_M"] +
                     b["b_male_M"] * S[n,"male"] +
                     b["b_parent_educ_M"] * S[n,"parent_educ"] +
                     b["b_L1_M"] * S[n,"L1"] +
                     b["b_A_M"] * 0) )^( 1 - S[n,"M"] )) ) *
       ((b["p_L1"])^(S[n,"L1"])) *
       ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
       ((b["p_L0_male"])^(S[n,"male"])) *
       ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
       ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
       ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }

# MI
S.MI <- cbind(expand.grid(c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4))
colnames(S.MI) <- list("male","parent_educ","L1","sum.mi")
for (n in 1:8) {
  S.MI[n,"sum.mi"] <- ( ( b["mu_Y"] +
                          b["c_male_Y"] * S.MI[n,"male"] +
                          b["c_parent_educ_Y"] * S.MI[n,"parent_educ"] +
                          b["c_A_Y"] * 1 +
                          b["c_L1_Y"] * S.MI[n,"L1"] +
                          b["c_M_Y"] * 1 +
                          b["c_AM_Y"] * 1 * 1 * b["A.M.inter"] ) -
                        ( b["mu_Y"] +
                            b["c_male_Y"] * S.MI[n,"male"] +
                            b["c_parent_educ_Y"] * S.MI[n,"parent_educ"] +
                            b["c_A_Y"] * 1 +
                            b["c_L1_Y"] * S.MI[n,"L1"] +
                            b["c_M_Y"] * 0 +
                            b["c_AM_Y"] * 1 * 0 * b["A.M.inter"] ) -
```

```
                            ( b["mu_Y"] +
                                b["c_male_Y"] * S.MI[n,"male"] +
                                b["c_parent_educ_Y"] * S.MI[n,"parent_educ"] +
                                b["c_A_Y"] * 0 +
                                b["c_L1_Y"] * S.MI[n,"L1"] +
                                b["c_M_Y"] * 1 +
                                b["c_AM_Y"] * 0 * 1 * b["A.M.inter"] ) +
                            ( b["mu_Y"] +
                                b["c_male_Y"] * S.MI[n,"male"] +
                                b["c_parent_educ_Y"] * S.MI[n,"parent_educ"] +
                                b["c_A_Y"] * 0 +
                                b["c_L1_Y"] * S.MI[n,"L1"] +
                                b["c_M_Y"] * 0 +
                                b["c_AM_Y"] * 0 * 0 * b["A.M.inter"] )) *
      ( ( b["b_M"] +
            b["b_male_M"] * S.MI[n,"male"] +
            b["b_parent_educ_M"] * S.MI[n,"parent_educ"] +
            b["b_L1_M"] * S[n,"L1"] +
            b["b_A_M"] * 1 ) -
          ( b["b_M"] +
              b["b_male_M"] * S.MI[n,"male"] +
              b["b_parent_educ_M"] * S.MI[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 0 )) *
      ((b["p_L1"])^(S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
      ((b["p_L0_male"])^(S.MI[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S.MI[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S.MI[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S.MI[n,"parent_educ"]))
    }

  PDE.qol <- sum(S[,"sum.pde"])
  PIE.qol <- sum(S[,"sum.pie"])
  MI.qol <- sum(S.MI[,"sum.mi"])

  return(list(PDE.death = PDE.death, PIE.death = PIE.death, MI.death = MI.death,
              PDE.qol = PDE.qol, PIE.qol = PIE.qol, MI.qol = MI.qol))
}
```

```
true.3way.no.inter <- true.3way.decomp(interaction = 0)

true.3way.with.inter <- true.3way.decomp(interaction = 1)
```

The PNDE $= \mathbb{E}\left(Y_{1,M_0}\right) - \mathbb{E}\left(Y_{0,M_0}\right)$, the PNIE $= \mathbb{E}\left(Y_{0,M_1}\right) - \mathbb{E}\left(Y_{0,M_0}\right)$ and the MIE $= \mathbb{E}\left((Y_{1,1} - Y_{1,0} - Y_{0,1} + Y_{0,0}) \times (M_1 - M_0)\right)$ are respectively:

- 0.05, 0.00800000000000001 and 0.000 for death without interaction,
- 0.05855, 0.00800000000000001 and 0.003 for death with interaction,
- -4, -0.899999999999999 and 0 for quality of life without interaction,
- -5.425, -0.899999999999999 and -0.5 for quality of life with interaction.

11.1.6 Vanderweele's 4-way decomposition

The following function `true.4way.decomp` can be used to run the calculation for the 4-way decomposition of the total effect into a "controlled direct effect" (CDE), a "reference interaction effect" (RIE), a "mediated interaction effect" (MIE) and a "pure natural indirect effect" (PNIE).

```
true.4way.decomp <- function(interaction = NULL) {
  b <- param.causal.model.1(A.M.interaction = interaction)

  # binary outcome (death)
  S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1)), rep(NA,n=2^3), rep(NA,n=2^3),
             rep(NA,n=2^3), rep(NA,n=2^3))
  colnames(S) <- list("male","parent_educ", "L1", "sum.cde", "sum.intref",
                      "sum.intmed", "sum.pie")
  for (n in 1:8) {
    # CDE
    S[n,"sum.cde"] <- ( ( b["b_Y"] +
                            b["b_male_Y"] * S[n,"male"] +
                            b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                            b["b_A_Y"] * 1 +
                            b["b_L1_Y"] * S[n,"L1"] +
                            b["b_M_Y"] * 0 +
                            b["b_AM_Y"] * 1 * 0 * b["A.M.inter"] ) -
                          ( b["b_Y"] +
                              b["b_male_Y"] * S[n,"male"] +
                              b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                              b["b_A_Y"] * 0 +
                              b["b_L1_Y"] * S[n,"L1"] +
                              b["b_M_Y"] * 0 +
                              b["b_AM_Y"] * 0 * 0 * b["A.M.inter"] ) ) *
      ((b["p_L1"])^(S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
```

```
# INTref
S[n,"sum.intref"] <- ( ( b["b_Y"] +
                           b["b_male_Y"] * S[n,"male"] +
                           b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["b_A_Y"] * 1 +
                           b["b_L1_Y"] * S[n,"L1"] +
                           b["b_M_Y"] * 1 +
                           b["b_AM_Y"] * 1 * 1 * b["A.M.inter"] ) -
                         ( b["b_Y"] +
                             b["b_male_Y"] * S[n,"male"] +
                             b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                             b["b_A_Y"] * 1 +
                             b["b_L1_Y"] * S[n,"L1"] +
                             b["b_M_Y"] * 0 +
                             b["b_AM_Y"] * 1 * 0 * b["A.M.inter"] ) -
                         ( b["b_Y"] +
                             b["b_male_Y"] * S[n,"male"] +
                             b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                             b["b_A_Y"] * 0 +
                             b["b_L1_Y"] * S[n,"L1"] +
                             b["b_M_Y"] * 1 +
                             b["b_AM_Y"] * 0 * 1 * b["A.M.inter"] ) +
                         ( b["b_Y"] +
                             b["b_male_Y"] * S[n,"male"] +
                             b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                             b["b_A_Y"] * 0 +
                             b["b_L1_Y"] * S[n,"L1"] +
                             b["b_M_Y"] * 0 +
                             b["b_AM_Y"] * 0 * 0 * b["A.M.inter"] )) *
  ( b["b_M"] +
      b["b_male_M"] * S[n,"male"] +
      b["b_parent_educ_M"] * S[n,"parent_educ"] +
      b["b_L1_M"] * S[n,"L1"] +
      b["b_A_M"] * 0 ) *
  ((b["p_L1"])^(S[n,"L1"])) *
  ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
  ((b["p_L0_male"])^(S[n,"male"])) *
  ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
  ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
  ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

# INTmed
S[n,"sum.intmed"] <- ( ( b["b_Y"] +
                           b["b_male_Y"] * S[n,"male"] +
                           b["b_parent_educ_Y"] * S[n,"parent_educ"] +
```

```
                              b["b_A_Y"] * 1 +
                              b["b_L1_Y"] * S[n,"L1"] +
                              b["b_M_Y"] * 1 +
                              b["b_AM_Y"] * 1 * 1 * b["A.M.inter"] ) -
                            ( b["b_Y"] +
                                b["b_male_Y"] * S[n,"male"] +
                                b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                                b["b_A_Y"] * 1 +
                                b["b_L1_Y"] * S[n,"L1"] +
                                b["b_M_Y"] * 0 +
                                b["b_AM_Y"] * 1 * 0 * b["A.M.inter"] ) -
                            ( b["b_Y"] +
                                b["b_male_Y"] * S[n,"male"] +
                                b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                                b["b_A_Y"] * 0 +
                                b["b_L1_Y"] * S[n,"L1"] +
                                b["b_M_Y"] * 1 +
                                b["b_AM_Y"] * 0 * 1 * b["A.M.inter"] ) +
                            ( b["b_Y"] +
                                b["b_male_Y"] * S[n,"male"] +
                                b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                                b["b_A_Y"] * 0 +
                                b["b_L1_Y"] * S[n,"L1"] +
                                b["b_M_Y"] * 0 +
                                b["b_AM_Y"] * 0 * 0 * b["A.M.inter"] )) *
  ( ( b["b_M"] +
        b["b_male_M"] * S[n,"male"] +
        b["b_parent_educ_M"] * S[n,"parent_educ"] +
        b["b_L1_M"] * S[n,"L1"] +
        b["b_A_M"] * 1 ) -
      ( b["b_M"] +
          b["b_male_M"] * S[n,"male"] +
          b["b_parent_educ_M"] * S[n,"parent_educ"] +
          b["b_L1_M"] * S[n,"L1"] +
          b["b_A_M"] * 0 )) *
  ((b["p_L1"])^(S[n,"L1"])) *
  ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
  ((b["p_L0_male"])^(S[n,"male"])) *
  ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
  ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
  ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

# PIE
S[n,"sum.pie"] <- ( ( b["b_Y"] +
                        b["b_male_Y"] * S[n,"male"] +
```

```
                            b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                            b["b_A_Y"] * 0 +
                            b["b_L1_Y"] * S[n,"L1"] +
                            b["b_M_Y"] * 1 +
                            b["b_AM_Y"] * 0 * 1 * b["A.M.inter"] ) -
                          ( b["b_Y"] +
                              b["b_male_Y"] * S[n,"male"] +
                              b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                              b["b_A_Y"] * 0 +
                              b["b_L1_Y"] * S[n,"L1"] +
                              b["b_M_Y"] * 0 +
                              b["b_AM_Y"] * 0 * 0 * b["A.M.inter"] ) ) *
    ( ( b["b_M"] +
          b["b_male_M"] * S[n,"male"] +
          b["b_parent_educ_M"] * S[n,"parent_educ"] +
          b["b_L1_M"] * S[n,"L1"] +
          b["b_A_M"] * 1 ) -
        ( b["b_M"] +
            b["b_male_M"] * S[n,"male"] +
            b["b_parent_educ_M"] * S[n,"parent_educ"] +
            b["b_L1_M"] * S[n,"L1"] +
            b["b_A_M"] * 0 )) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }

CDE.death <- sum(S[,"sum.cde"])
INTref.death <- sum(S[,"sum.intref"])
INTmed.death <- sum(S[,"sum.intmed"])
PIE.death <- sum(S[,"sum.pie"])

# quantitative outcome (QoL)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1)), rep(NA,n=2^3), rep(NA,n=2^3),
           rep(NA,n=2^3), rep(NA,n=2^3))
colnames(S) <- list("male","parent_educ", "L1", "sum.cde", "sum.intref",
                    "sum.intmed", "sum.pie")
for (n in 1:8) {
  # CDE
  S[n,"sum.cde"] <- ( ( b["mu_Y"] +
                            b["c_male_Y"] * S[n,"male"] +
                            b["c_parent_educ_Y"] * S[n,"parent_educ"] +
```

```
                             b["c_A_Y"] * 1 +
                             b["c_L1_Y"] * S[n,"L1"] +
                             b["c_M_Y"] * 0 +
                             b["c_AM_Y"] * 1 * 0 * b["A.M.inter"] ) -
                           ( b["mu_Y"] +
                               b["c_male_Y"] * S[n,"male"] +
                               b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                               b["c_A_Y"] * 0 +
                               b["c_L1_Y"] * S[n,"L1"] +
                               b["c_M_Y"] * 0 +
                               b["c_AM_Y"] * 0 * 0 * b["A.M.inter"] ) ) *
  ((b["p_L1"])^(S[n,"L1"])) *
  ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
  ((b["p_L0_male"])^(S[n,"male"])) *
  ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
  ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
  ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

# INTref
S[n,"sum.intref"] <- ( ( b["mu_Y"] +
                               b["c_male_Y"] * S[n,"male"] +
                               b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                               b["c_A_Y"] * 1 +
                               b["c_L1_Y"] * S[n,"L1"] +
                               b["c_M_Y"] * 1 +
                               b["c_AM_Y"] * 1 * 1 * b["A.M.inter"] ) -
                             ( b["mu_Y"] +
                                 b["c_male_Y"] * S[n,"male"] +
                                 b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                                 b["c_A_Y"] * 1 +
                                 b["c_L1_Y"] * S[n,"L1"] +
                                 b["c_M_Y"] * 0 +
                                 b["c_AM_Y"] * 1 * 0 * b["A.M.inter"] ) -
                             ( b["mu_Y"] +
                                 b["c_male_Y"] * S[n,"male"] +
                                 b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                                 b["c_A_Y"] * 0 +
                                 b["c_L1_Y"] * S[n,"L1"] +
                                 b["c_M_Y"] * 1 +
                                 b["c_AM_Y"] * 0 * 1 * b["A.M.inter"] ) +
                             ( b["mu_Y"] +
                                 b["c_male_Y"] * S[n,"male"] +
                                 b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                                 b["c_A_Y"] * 0 +
                                 b["c_L1_Y"] * S[n,"L1"] +
```

```
                                b["c_M_Y"] * 0 +
                                b["c_AM_Y"] * 0 * 0 * b["A.M.inter"] )) *
    ( b["b_M"] +
        b["b_male_M"] * S[n,"male"] +
        b["b_parent_educ_M"] * S[n,"parent_educ"] +
        b["b_L1_M"] * S[n,"L1"] +
        b["b_A_M"] * 0 ) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  # INTmed
  S[n,"sum.intmed"] <- ( ( b["mu_Y"] +
                              b["c_male_Y"] * S[n,"male"] +
                              b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                              b["c_A_Y"] * 1 +
                              b["c_L1_Y"] * S[n,"L1"] +
                              b["c_M_Y"] * 1 +
                              b["c_AM_Y"] * 1 * 1 * b["A.M.inter"] ) -
                            ( b["mu_Y"] +
                                b["c_male_Y"] * S[n,"male"] +
                                b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                                b["c_A_Y"] * 1 +
                                b["c_L1_Y"] * S[n,"L1"] +
                                b["c_M_Y"] * 0 +
                                b["c_AM_Y"] * 1 * 0 * b["A.M.inter"] ) -
                            ( b["mu_Y"] +
                                b["c_male_Y"] * S[n,"male"] +
                                b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                                b["c_A_Y"] * 0 +
                                b["c_L1_Y"] * S[n,"L1"] +
                                b["c_M_Y"] * 1 +
                                b["c_AM_Y"] * 0 * 1 * b["A.M.inter"] ) +
                            ( b["mu_Y"] +
                                b["c_male_Y"] * S[n,"male"] +
                                b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                                b["c_A_Y"] * 0 +
                                b["c_L1_Y"] * S[n,"L1"] +
                                b["c_M_Y"] * 0 +
                                b["c_AM_Y"] * 0 * 0 * b["A.M.inter"] )) *
    ( ( b["b_M"] +
          b["b_male_M"] * S[n,"male"] +
```

```
          b["b_parent_educ_M"] * S[n,"parent_educ"] +
          b["b_L1_M"] * S[n,"L1"] +
          b["b_A_M"] * 1 ) -
        ( b["b_M"] +
            b["b_male_M"] * S[n,"male"] +
            b["b_parent_educ_M"] * S[n,"parent_educ"] +
            b["b_L1_M"] * S[n,"L1"] +
            b["b_A_M"] * 0 )) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  # PIE
  S[n,"sum.pie"] <- ( ( b["mu_Y"] +
                          b["c_male_Y"] * S[n,"male"] +
                          b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                          b["c_A_Y"] * 0 +
                          b["c_L1_Y"] * S[n,"L1"] +
                          b["c_M_Y"] * 1 +
                          b["c_AM_Y"] * 0 * 1 * b["A.M.inter"] ) -
                        ( b["mu_Y"] +
                            b["c_male_Y"] * S[n,"male"] +
                            b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                            b["c_A_Y"] * 0 +
                            b["c_L1_Y"] * S[n,"L1"] +
                            b["c_M_Y"] * 0 +
                            b["c_AM_Y"] * 0 * 0 * b["A.M.inter"] ) ) *
    ( ( b["b_M"] +
          b["b_male_M"] * S[n,"male"] +
          b["b_parent_educ_M"] * S[n,"parent_educ"] +
          b["b_L1_M"] * S[n,"L1"] +
          b["b_A_M"] * 1 ) -
        ( b["b_M"] +
            b["b_male_M"] * S[n,"male"] +
            b["b_parent_educ_M"] * S[n,"parent_educ"] +
            b["b_L1_M"] * S[n,"L1"] +
            b["b_A_M"] * 0 )) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
```

```
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
    }

  CDE.qol <- sum(S[,"sum.cde"])
  INTref.qol <- sum(S[,"sum.intref"])
  INTmed.qol <- sum(S[,"sum.intmed"])
  PIE.qol <- sum(S[,"sum.pie"])

  return(list(CDE.death = CDE.death, INTref.death = INTref.death,
              INTmed.death = INTmed.death, PIE.death = PIE.death,
              CDE.qol = CDE.qol, INTref.qol = INTref.qol,
              INTmed.qol = INTmed.qol, PIE.qol = PIE.qol))
}
```

```
true.4way.no.inter <- true.4way.decomp(interaction = 0)

true.4way.with.inter <- true.4way.decomp(interaction = 1)
```

The CDE $= \mathbb{E}\left(Y_{1,0}\right) - \mathbb{E}\left(Y_{0,0}\right)$, RIE $= \left((Y_{1,1} - Y_{1,0} - Y_{0,1} + Y_{0,0}) \times M_0\right)$, MIE $= \mathbb{E}\left((Y_{1,1} - Y_{1,0} - Y_{0,1} + Y_{0,0}) \times (M_1 - M_0)\right)$ and PNIE $= \mathbb{E}\left(Y_{0,M_1}\right) - \mathbb{E}\left(Y_{0,M_0}\right)$ are respectively:

- 0.05, 0.000, 0.000 and 0.008 for death without interaction,
- 0.05, 0.00855, 0.003 and 0.008 for death with interaction,
- -4, 0, 0 and -0.9 for quality of life without interaction,
- -4, -1.425, -0.5 and -0.9 for quality of life with interaction.

11.1.7 Marginal randomized direct and indirect effects

The following function `true.marg.random` can be used to run the calculation for the marginal randomized natural direct (marginal MRDE) and indirect effects (marginal MRIE).

```
true.marg.random <- function(interaction = NULL) {
  b <- param.causal.model.1(A.M.interaction = interaction)

  # marginal distribution of M
  M.S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^5))
  colnames(M.S) <- list("male","parent_educ","L1","M","A","sum")

  for (n in 1:32) {
    M.S[n,"sum"] <- (( b["b_M"] +
                         b["b_male_M"] * M.S[n,"male"] +
```

```
                       b["b_parent_educ_M"] * M.S[n,"parent_educ"] +
                       b["b_L1_M"] * M.S[n,"L1"] +
                       b["b_A_M"] * M.S[n,"A"])^( M.S[n,"M"] )) *
      (( 1 - (b["b_M"] +
                 b["b_male_M"] * M.S[n,"male"] +
                 b["b_parent_educ_M"] * M.S[n,"parent_educ"] +
                 b["b_L1_M"] * M.S[n,"L1"] +
                 b["b_A_M"] * M.S[n,"A"]) )^( 1 - M.S[n,"M"] ))
    }

  M0.A0.L0_00.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                          M.S[,"parent_educ"]==0 & M.S[,"L1"]==0,"sum"]
  M0.A0.L0_01.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                          M.S[,"parent_educ"]==1 & M.S[,"L1"]==0,"sum"]
  M0.A0.L0_10.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                          M.S[,"parent_educ"]==0 & M.S[,"L1"]==0,"sum"]
  M0.A0.L0_11.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                          M.S[,"parent_educ"]==1 & M.S[,"L1"]==0,"sum"]
  M0.A0.L0_00.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                          M.S[,"parent_educ"]==0 & M.S[,"L1"]==1,"sum"]
  M0.A0.L0_01.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                          M.S[,"parent_educ"]==1 & M.S[,"L1"]==1,"sum"]
  M0.A0.L0_10.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                          M.S[,"parent_educ"]==0 & M.S[,"L1"]==1,"sum"]
  M0.A0.L0_11.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                          M.S[,"parent_educ"]==1 & M.S[,"L1"]==1,"sum"]

  M1.A0.L0_00.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                          M.S[,"parent_educ"]==0 & M.S[,"L1"]==0,"sum"]
  M1.A0.L0_01.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                          M.S[,"parent_educ"]==1 & M.S[,"L1"]==0,"sum"]
  M1.A0.L0_10.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                          M.S[,"parent_educ"]==0 & M.S[,"L1"]==0,"sum"]
  M1.A0.L0_11.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                          M.S[,"parent_educ"]==1 & M.S[,"L1"]==0,"sum"]
  M1.A0.L0_00.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                          M.S[,"parent_educ"]==0 & M.S[,"L1"]==1,"sum"]
  M1.A0.L0_01.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                          M.S[,"parent_educ"]==1 & M.S[,"L1"]==1,"sum"]
  M1.A0.L0_10.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                          M.S[,"parent_educ"]==0 & M.S[,"L1"]==1,"sum"]
  M1.A0.L0_11.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                          M.S[,"parent_educ"]==1 & M.S[,"L1"]==1,"sum"]

  M0.A1.L0_00.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==0 &
```

```
                         M.S[,"parent_educ"]==0 & M.S[,"L1"]==0,"sum"]
M0.A1.L0_01.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                         M.S[,"parent_educ"]==1 & M.S[,"L1"]==0,"sum"]
M0.A1.L0_10.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                         M.S[,"parent_educ"]==0 & M.S[,"L1"]==0,"sum"]
M0.A1.L0_11.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                         M.S[,"parent_educ"]==1 & M.S[,"L1"]==0,"sum"]
M0.A1.L0_00.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                         M.S[,"parent_educ"]==0 & M.S[,"L1"]==1,"sum"]
M0.A1.L0_01.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                         M.S[,"parent_educ"]==1 & M.S[,"L1"]==1,"sum"]
M0.A1.L0_10.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                         M.S[,"parent_educ"]==0 & M.S[,"L1"]==1,"sum"]
M0.A1.L0_11.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                         M.S[,"parent_educ"]==1 & M.S[,"L1"]==1,"sum"]

M1.A1.L0_00.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                         M.S[,"parent_educ"]==0 & M.S[,"L1"]==0,"sum"]
M1.A1.L0_01.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                         M.S[,"parent_educ"]==1 & M.S[,"L1"]==0,"sum"]
M1.A1.L0_10.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                         M.S[,"parent_educ"]==0 & M.S[,"L1"]==0,"sum"]
M1.A1.L0_11.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                         M.S[,"parent_educ"]==1 & M.S[,"L1"]==0,"sum"]
M1.A1.L0_00.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                         M.S[,"parent_educ"]==0 & M.S[,"L1"]==1,"sum"]
M1.A1.L0_01.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                         M.S[,"parent_educ"]==1 & M.S[,"L1"]==1,"sum"]
M1.A1.L0_10.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                         M.S[,"parent_educ"]==0 & M.S[,"L1"]==1,"sum"]
M1.A1.L0_11.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                         M.S[,"parent_educ"]==1 & M.S[,"L1"]==1,"sum"]

# binary outcome (death)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4), rep(NA,n=2^4),
           rep(NA,n=2^4))
colnames(S) <- list("male","parent_educ","L1","M","sum.psi11", "sum.psi10", "sum.psi00")
for (n in 1:16) {
  S[n,"sum.psi11"] <-  ( b["b_Y"] +                                          # A=1
                          b["b_male_Y"] * S[n,"male"] +
                          b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                          b["b_A_Y"] * 1 +
                          b["b_L1_Y"] * S[n,"L1"] +
                          b["b_M_Y"] * S[n,"M"] +
                          b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
```

```
   ((M1.A1.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) + # A
       M1.A1.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
       M1.A1.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
       M1.A1.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
       M1.A1.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
       M1.A1.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
       M1.A1.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
       M1.A1.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
       (S[n,"L1"]==1) )^( S[n,"M"] )) *
   ((M0.A1.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
       M0.A1.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
       M0.A1.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
       M0.A1.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
       M0.A1.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
       M0.A1.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
       M0.A1.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
       M0.A1.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
       (S[n,"L1"]==1) )^( 1 - S[n,"M"] )) *
   ((b["p_L1"])^(M.S[n,"L1"])) *
   ((1 - b["p_L1"])^(1 - M.S[n,"L1"])) *
   ((b["p_L0_male"])^(S[n,"male"])) *
   ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
   ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
   ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

 S[n,"sum.psi10"] <-  ( b["b_Y"] +                                              # A=1
                         b["b_male_Y"] * S[n,"male"] +
                         b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                         b["b_A_Y"] * 1 +
                         b["b_L1_Y"] * S[n,"L1"] +
                         b["b_M_Y"] * S[n,"M"] +
                         b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
   ((M1.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) + # A
       M1.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
       M1.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
       M1.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
       M1.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
       M1.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
       M1.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
       M1.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
       (S[n,"L1"]==1) )^( S[n,"M"] )) *
   ((M0.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
       M0.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
       M0.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
       M0.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
```

```
        M0.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
        M0.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
        M0.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
        M0.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
        (S[n,"L1"]==1) )^( 1 - S[n,"M"] )) *
    ((b["p_L1"])^(M.S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - M.S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  S[n,"sum.psi00"] <-  ( b["b_Y"] +                                           # A=0
                         b["b_male_Y"] * S[n,"male"] +
                         b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                         b["b_A_Y"] * 0 +
                         b["b_L1_Y"] * S[n,"L1"] +
                         b["b_M_Y"] * S[n,"M"] +
                         b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
    ((M1.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) + # A'=0
        M1.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
        M1.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
        M1.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
        M1.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
        M1.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
        M1.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
        M1.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
        (S[n,"L1"]==1) )^( S[n,"M"] )) *
    ((M0.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
        M0.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
        M0.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
        M0.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
        M0.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
        M0.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
        M0.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
        M0.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
        (S[n,"L1"]==1) )^( 1 - S[n,"M"] )) *
    ((b["p_L1"])^(M.S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - M.S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }
```

```
mrNDE.death <- sum(S[,"sum.psi10"]) - sum(S[,"sum.psi00"])
mrNIE.death <- sum(S[,"sum.psi11"]) - sum(S[,"sum.psi10"])

# quantitative outcome (QoL)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4),
           rep(NA,n=2^4), rep(NA,n=2^4))
colnames(S) <- list("male","parent_educ","L1","M","sum.psi11", "sum.psi10",
                    "sum.psi00")
for (n in 1:16) {
  S[n,"sum.psi11"] <-  ( b["mu_Y"] +                                     # A=1
                           b["c_male_Y"] * S[n,"male"] +
                           b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["c_A_Y"] * 1 +
                           b["c_L1_Y"] * S[n,"L1"] +
                           b["c_M_Y"] * S[n,"M"] +
                           b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
    ((M1.A1.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) + # A
        M1.A1.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
        M1.A1.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
        M1.A1.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
        M1.A1.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
        M1.A1.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
        M1.A1.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
        M1.A1.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
        (S[n,"L1"]==1) )^( S[n,"M"] )) *
    ((M0.A1.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
        M0.A1.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
        M0.A1.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
        M0.A1.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
        M0.A1.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
        M0.A1.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
        M0.A1.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
        M0.A1.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
        (S[n,"L1"]==1) )^( 1 - S[n,"M"] )) *
    ((b["p_L1"])^(M.S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - M.S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  S[n,"sum.psi10"] <-  ( b["mu_Y"] +                                     # A=1
                           b["c_male_Y"] * S[n,"male"] +
                           b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["c_A_Y"] * 1 +
```

```
                              b["c_L1_Y"] * S[n,"L1"] +
                              b["c_M_Y"] * S[n,"M"] +
                              b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
      ((M1.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) + # A'=0
          M1.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
          M1.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
          M1.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
          M1.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
          M1.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
          M1.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
          M1.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
          (S[n,"L1"]==1) )^( S[n,"M"] )) *
      ((M0.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
          M0.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
          M0.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
          M0.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
          M0.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
          M0.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
          M0.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
          M0.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
          (S[n,"L1"]==1) )^( 1 - S[n,"M"] )) *
      ((b["p_L1"])^(M.S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - M.S[n,"L1"])) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

    S[n,"sum.psi00"] <-  ( b["mu_Y"] +                                              # A=0
                              b["c_male_Y"] * S[n,"male"] +
                              b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                              b["c_A_Y"] * 0 +
                              b["c_L1_Y"] * S[n,"L1"] +
                              b["c_M_Y"] * S[n,"M"] +
                              b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
      ((M1.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) + # A'=0
          M1.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
          M1.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
          M1.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
          M1.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
          M1.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
          M1.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
          M1.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
          (S[n,"L1"]==1) )^( S[n,"M"] )) *
      ((M0.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
```

```
          M0.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
          M0.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==0) +
          M0.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==0) +
          M0.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
          M0.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"parent_educ"]==1)*(S[n,"L1"]==1) +
          M0.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==0)*(S[n,"L1"]==1) +
          M0.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"parent_educ"]==1)*
          (S[n,"L1"]==1) )^( 1 - S[n,"M"] )) *
      ((b["p_L1"])^(M.S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - M.S[n,"L1"])) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
    }

  mrNDE.qol <- sum(S[,"sum.psi10"]) - sum(S[,"sum.psi00"])
  mrNIE.qol <- sum(S[,"sum.psi11"]) - sum(S[,"sum.psi10"])

  return(list(mrNDE.death = mrNDE.death, mrNIE.death = mrNIE.death,
              mrNDE.qol = mrNDE.qol, mrNIE.qol = mrNIE.qol))
}
```

```
true.marg.random.no.inter <- true.marg.random(interaction = 0)

true.marg.random.with.inter <- true.marg.random(interaction = 1)
```

The marginal randomized direct effect MRDE $= \mathbb{E}\left(Y_{1,G_{0|L(0)}}\right) - \mathbb{E}\left(Y_{0,G_{0|L(0)}}\right)$ and the marginal randomized indirect effect MRIE $= \mathbb{E}\left(Y_{1,G_{1|L(0)}}\right) - \mathbb{E}\left(Y_{1,G_{0|L(0)}}\right)$ are respectively:

- 0.05 and 0.00800000000000003 for death without interaction,
- 0.05855 and 0.011 for death with interaction,
- -3.99999999999999 and -0.900000000000006 for quality of life without interaction,
- -5.42499999999999 and -1.40000000000001 for quality of life with interaction.

11.1.8 Conditional randomized direct and indirect effects

The following function `true.cond.random` can be used to run the calculation for the conditional randomized natural direct (CRDE) and indirect effects (CRIE).

```r
true.cond.random <- function(interaction = NULL) {
  b <- param.causal.model.1(A.M.interaction = interaction)

  # binary outcome (death)
  S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4),
             rep(NA,n=2^4), rep(NA,n=2^4))
  colnames(S) <- list("male","parent_educ","L1","M","sum.psi11", "sum.psi10",
                      "sum.psi00")
  for (n in 1:16) {
    S[n,"sum.psi11"] <-  ( b["b_Y"] +                                      # A=1
                           b["b_male_Y"] * S[n,"male"] +
                           b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["b_A_Y"] * 1 +
                           b["b_L1_Y"] * S[n,"L1"] +
                           b["b_M_Y"] * S[n,"M"] +
                           b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
      (( b["b_M"] +                                                        # A'=1
           b["b_male_M"] * S[n,"male"] +
           b["b_parent_educ_M"] * S[n,"parent_educ"] +
           b["b_L1_M"] * S[n,"L1"] +
           b["b_A_M"] * 1 )^( S[n,"M"] )) *
      (( 1 - (b["b_M"] +
                b["b_male_M"] * S[n,"male"] +
                b["b_parent_educ_M"] * S[n,"parent_educ"] +
                b["b_L1_M"] * S[n,"L1"] +
                b["b_A_M"] * 1 ) )^( 1 - S[n,"M"] )) *
      ((b["p_L1"])^(S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

    S[n,"sum.psi10"] <-  ( b["b_Y"] +                                      # A=1
                           b["b_male_Y"] * S[n,"male"] +
                           b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["b_A_Y"] * 1 +
                           b["b_L1_Y"] * S[n,"L1"] +
                           b["b_M_Y"] * S[n,"M"] +
                           b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
      (( b["b_M"] +                                                        # A'=0
           b["b_male_M"] * S[n,"male"] +
           b["b_parent_educ_M"] * S[n,"parent_educ"] +
           b["b_L1_M"] * S[n,"L1"] +
           b["b_A_M"] * 0 )^( S[n,"M"] )) *
```

```
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 0 ) )^( 1 - S[n,"M"] )) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  S[n,"sum.psi00"] <-  ( b["b_Y"] +                                     # A=0
                         b["b_male_Y"] * S[n,"male"] +
                         b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                         b["b_A_Y"] * 0 +
                         b["b_L1_Y"] * S[n,"L1"] +
                         b["b_M_Y"] * S[n,"M"] +
                         b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
    (( b["b_M"] +                                                        # A'=0
         b["b_male_M"] * S[n,"male"] +
         b["b_parent_educ_M"] * S[n,"parent_educ"] +
         b["b_L1_M"] * S[n,"L1"] +
         b["b_A_M"] * 0 )^( S[n,"M"] )) *
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 0 ) )^( 1 - S[n,"M"] )) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }

crNDE.death <- sum(S[,"sum.psi10"]) - sum(S[,"sum.psi00"])
crNIE.death <- sum(S[,"sum.psi11"]) - sum(S[,"sum.psi10"])

# quantitative outcome (QoL)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4),
           rep(NA,n=2^4), rep(NA,n=2^4))
colnames(S) <- list("male","parent_educ","L1","M","sum.psi11", "sum.psi10",
                    "sum.psi00")
```

```
for (n in 1:16) {
  S[n,"sum.psi11"] <-  ( b["mu_Y"] +                                          # A=1
                            b["c_male_Y"] * S[n,"male"] +
                            b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                            b["c_A_Y"] * 1 +
                            b["c_L1_Y"] * S[n,"L1"] +
                            b["c_M_Y"] * S[n,"M"] +
                            b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
    (( b["b_M"] +                                                             # A'=1
         b["b_male_M"] * S[n,"male"] +
         b["b_parent_educ_M"] * S[n,"parent_educ"] +
         b["b_L1_M"] * S[n,"L1"] +
         b["b_A_M"] * 1 )^( S[n,"M"] )) *
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 1 ) )^( 1 - S[n,"M"] )) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  S[n,"sum.psi10"] <-  ( b["mu_Y"] +                                          # A=1
                            b["c_male_Y"] * S[n,"male"] +
                            b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                            b["c_A_Y"] * 1 +
                            b["c_L1_Y"] * S[n,"L1"] +
                            b["c_M_Y"] * S[n,"M"] +
                            b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
    (( b["b_M"] +                                                             # A'=0
         b["b_male_M"] * S[n,"male"] +
         b["b_parent_educ_M"] * S[n,"parent_educ"] +
         b["b_L1_M"] * S[n,"L1"] +
         b["b_A_M"] * 0 )^( S[n,"M"] )) *
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 0 ) )^( 1 - S[n,"M"] )) *
    ((b["p_L1"])^(S[n,"L1"])) *
    ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
    ((b["p_L0_male"])^(S[n,"male"])) *
```

```
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

    S[n,"sum.psi00"] <-  ( b["mu_Y"] +                                 # A=0
                            b["c_male_Y"] * S[n,"male"] +
                            b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                            b["c_A_Y"] * 0 +
                            b["c_L1_Y"] * S[n,"L1"] +
                            b["c_M_Y"] * S[n,"M"] +
                            b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
      (( b["b_M"] +                                                    # A'=0
           b["b_male_M"] * S[n,"male"] +
           b["b_parent_educ_M"] * S[n,"parent_educ"] +
           b["b_L1_M"] * S[n,"L1"] +
           b["b_A_M"] * 0 )^( S[n,"M"] )) *
      (( 1 - (b["b_M"] +
                b["b_male_M"] * S[n,"male"] +
                b["b_parent_educ_M"] * S[n,"parent_educ"] +
                b["b_L1_M"] * S[n,"L1"] +
                b["b_A_M"] * 0 ) )^( 1 - S[n,"M"] )) *
      ((b["p_L1"])^(S[n,"L1"])) *
      ((1 - b["p_L1"])^(1 - S[n,"L1"])) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
    }

  crNDE.qol <- sum(S[,"sum.psi10"]) - sum(S[,"sum.psi00"])
  crNIE.qol <- sum(S[,"sum.psi11"]) - sum(S[,"sum.psi10"])

  return(list(crNDE.death = crNDE.death, crNIE.death = crNIE.death,
              crNDE.qol = crNDE.qol, crNIE.qol = crNIE.qol))
}
```

```
true.cond.random.no.inter <- true.cond.random(interaction = 0)

true.cond.random.with.inter <- true.cond.random(interaction = 1)
```

The conditional randomized direct effect CRDE $= \mathbb{E}\left(Y_{1,\Gamma_{0|L(0),L(1)}}\right) - \mathbb{E}\left(Y_{0,\Gamma_{0|L(0),L(1)}}\right)$ and conditional randomized indirect effect CRIE $= \mathbb{E}\left(Y_{1,\Gamma_{1|L(0),L(1)}}\right) - \mathbb{E}\left(Y_{1,\Gamma_{0|L(0),L(1)}}\right)$ are respectively:

- 0.05 and 0.00800000000000003 for death without interaction,
- 0.05855 and 0.011 for death with interaction,
- -3.99999999999999 and -0.900000000000006 for quality of life without interaction,
- -5.42499999999999 and -1.40000000000001 for quality of life with interaction.

Table 11.1: True values without time varying confounders

Effects	Without $A * M$ interaction	with $A * M$ interaction
Binary outcome		
Average total effect (ATE)	0.058	0.06955
Controlled direct effect (CDE)		
- CDE, setting do(M=0)	0.05	0.05
- CDE, setting do(M=1)	0.05	0.08
Pure NDE and Total NIE		
- PNDE	0.05	0.05855
- TNIE	0.00800000000000001	0.011
Total NDE and Pure NIE		
- TNDE	0.05	0.06155
- PNIE	0.00800000000000001	0.00800000000000001
3-way decomposition		
- PDE	0.05	0.05855
- PIE	0.00800000000000001	0.00800000000000001
- MI	0.000	0.003
4-way decomposition		
- CDE	0.05	0.05
- INTref	0.000	0.00855
- INTmed	0.000	0.003

Effects	Without $A * M$ interaction	with $A * M$ interaction
- PIE	0.008	0.008
Marginal randomized		
- marginal rNDE	0.05	0.05855
- marginal rNIE	0.00800000000000003	0.011
Conditional randomized		
- conditional rNDE	0.05	0.05855
- conditional rNIE	0.00800000000000003	0.011
Quantitative outcome		
Average total effect (ATE)	-4.9	-6.825
Controlled direct effect (CDE)		
- CDE, setting do(M=0)	-4	-4
- CDE, setting do(M=1)	-4	-9
Pure NDE and Total NIE		
- PNDE	-4	-5.425
- TNIE	-0.9	-1.4
Total NDE and Pure NIE		
- TNDE	-4	-5.925
- PNIE	-0.899999999999999	-0.899999999999999
3-way decomposition		
- PDE	-4	-5.425
- PIE	-0.899999999999999	-0.899999999999999
- MI	0	-0.5

Effects	Without $A * M$ interaction	with $A * M$ interaction
4-way decomposition		
- CDE	-4	-4
- INTref	0.000	-1.425
- INTmed	0.000	-0.5
- PIE	-0.9	-0.9
Marginal randomized		
- marginal rNDE	-3.99999999999999	-5.42499999999999
- marginal rNIE	-0.900000000000006	-1.40000000000001
Conditional randomized		
- conditional rNDE	-3.99999999999999	-5.42499999999999
- conditional rNIE	-0.900000000000006	-1.40000000000001

11.2 True causal quantities with mediator-ouctome confounder affected by the exposure

11.2.1 Average total effects (ATE)

The following function `true.ATE.tv.conf` can be used to run the calculation for the average total effects (ATE).

```
true.ATE.time.var.conf <- function(interaction = NULL) {
  b <- param.causal.model.2(A.M.interaction = interaction)

  # binary outcome (death)
  S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1), c(0,1)), rep(NA,n=2^4))
  colnames(S) <- list("male","parent_educ","L1","M","sum")
  for (n in 1:16) {
    S[n,"sum"] <- ( ( ( b["b_Y"] +
                        b["b_male_Y"] * S[n,"male"] +
                        b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                        b["b_A_Y"] * 1 +
                        b["b_L1_Y"] * S[n,"L1"] +
                        b["b_M_Y"] * S[n,"M"] +
```

```
                      b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
                    (( b["b_M"] +
                           b["b_male_M"] * S[n,"male"] +
                           b["b_parent_educ_M"] * S[n,"parent_educ"] +
                           b["b_L1_M"] * S[n,"L1"] +
                           b["b_A_M"] * 1 )^( S[n,"M"] )) *
                    (( 1 - (b["b_M"] +
                              b["b_male_M"] * S[n,"male"] +
                              b["b_parent_educ_M"] * S[n,"parent_educ"] +
                              b["b_L1_M"] * S[n,"L1"] +
                              b["b_A_M"] * 1) )^( 1 - S[n,"M"] )) *
                        (( b["b_L1"] +
                             b["b_male_L1"] * S[n,"male"] +
                             b["b_parent_L1"] * S[n,"parent_educ"] +
                             b["b_A_L1"] * 1)^( S[n,"L1"] )) *
                        (( 1 - ( b["b_L1"] +
                                   b["b_male_L1"] * S[n,"male"] +
                                   b["b_parent_L1"] * S[n,"parent_educ"] +
                                   b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) ) -
                    ( ( b["b_Y"] +
                          b["b_male_Y"] * S[n,"male"] +
                          b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                          b["b_A_Y"] * 0 +
                          b["b_L1_Y"] * S[n,"L1"] +
                          b["b_M_Y"] * S[n,"M"] +
                          b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
                        (( b["b_M"] +
                             b["b_male_M"] * S[n,"male"] +
                             b["b_parent_educ_M"] * S[n,"parent_educ"] +
                             b["b_L1_M"] * S[n,"L1"] +
                             b["b_A_M"] * 0 )^( S[n,"M"] )) *
                        (( 1 - (b["b_M"] +
                                  b["b_male_M"] * S[n,"male"] +
                                  b["b_parent_educ_M"] * S[n,"parent_educ"] +
                                  b["b_L1_M"] * S[n,"L1"] +
                                  b["b_A_M"] * 0) )^( 1 - S[n,"M"] )) *
                        (( b["b_L1"] +
                             b["b_male_L1"] * S[n,"male"] +
                             b["b_parent_L1"] * S[n,"parent_educ"] +
                             b["b_A_L1"] * 0)^( S[n,"L1"] )) *
                        (( 1 - ( b["b_L1"] +
                                   b["b_male_L1"] * S[n,"male"] +
                                   b["b_parent_L1"] * S[n,"parent_educ"] +
                                   b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) ) ) *
       ((b["p_L0_male"])^(S[n,"male"])) *
```

```
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }

ATE.death <- sum(S[,"sum"])

# quantitative outcome (QoL)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1), c(0,1)), rep(NA,n=2^4))
colnames(S) <- list("male","parent_educ","L1","M","sum")
for (n in 1:16) {
  S[n,"sum"] <- ( ( ( b["mu_Y"] +
                        b["c_male_Y"] * S[n,"male"] +
                        b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                        b["c_A_Y"] * 1 +
                        b["c_L1_Y"] * S[n,"L1"] +
                        b["c_M_Y"] * S[n,"M"] +
                        b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
                      (( b["b_M"] +
                           b["b_male_M"] * S[n,"male"] +
                           b["b_parent_educ_M"] * S[n,"parent_educ"] +
                           b["b_L1_M"] * S[n,"L1"] +
                           b["b_A_M"] * 1 )^( S[n,"M"] )) *
                      (( 1 - (b["b_M"] +
                                b["b_male_M"] * S[n,"male"] +
                                b["b_parent_educ_M"] * S[n,"parent_educ"] +
                                b["b_L1_M"] * S[n,"L1"] +
                                b["b_A_M"] * 1) )^( 1 - S[n,"M"] )) *
                      (( b["b_L1"] +
                           b["b_male_L1"] * S[n,"male"] +
                           b["b_parent_L1"] * S[n,"parent_educ"] +
                           b["b_A_L1"] * 1)^( S[n,"L1"] )) *
                      (( 1 - ( b["b_L1"] +
                                 b["b_male_L1"] * S[n,"male"] +
                                 b["b_parent_L1"] * S[n,"parent_educ"] +
                                 b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) ) -
                    ( ( b["mu_Y"] +
                          b["c_male_Y"] * S[n,"male"] +
                          b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                          b["c_A_Y"] * 0 +
                          b["c_L1_Y"] * S[n,"L1"] +
                          b["c_M_Y"] * S[n,"M"] +
                          b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
                        (( b["b_M"] +
                             b["b_male_M"] * S[n,"male"] +
```

```
                              b["b_parent_educ_M"] * S[n,"parent_educ"] +
                              b["b_L1_M"] * S[n,"L1"] +
                              b["b_A_M"] * 0 )^( S[n,"M"] )) *
                         (( 1 - (b["b_M"] +
                                   b["b_male_M"] * S[n,"male"] +
                                   b["b_parent_educ_M"] * S[n,"parent_educ"] +
                                   b["b_L1_M"] * S[n,"L1"] +
                                   b["b_A_M"] * 0) )^( 1 - S[n,"M"] )) ) *
                       (( b["b_L1"] +
                            b["b_male_L1"] * S[n,"male"] +
                            b["b_parent_L1"] * S[n,"parent_educ"] +
                            b["b_A_L1"] * 0)^( S[n,"L1"] )) *
                       (( 1 - ( b["b_L1"] +
                                  b["b_male_L1"] * S[n,"male"] +
                                  b["b_parent_L1"] * S[n,"parent_educ"] +
                                  b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) ) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
    }

  ATE.qol <- sum(S[,"sum"])

  return(list(ATE.death = ATE.death, ATE.qol = ATE.qol))
}
```

```
true.ATE2.no.inter <- true.ATE.time.var.conf(interaction = 0)

true.ATE2.with.inter <- true.ATE.time.var.conf(interaction = 1)
```

The average total effects $\text{ATE} = \mathbb{E}(Y_1) - \mathbb{E}(Y_0)$ are:

- 0.0752 for death and -6.26 for quality of life without interaction
- 0.089282 for death and -8.607 for quality of life with interaction

11.2.2 Controlled direct effects (CDE)

The following function `true.CDE.time.var` can be used to run the calculation for controlled direct effects (CDE).

```
true.CDE.time.var <- function(interaction = NULL) {
  b <- param.causal.model.2(A.M.interaction = interaction)
```

```
# binary outcome (death)
# we estimate both CDE, fixing do(M) = 0 et do(M) = 1 and using the
# corresponding lines in the S matrix
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^3))
colnames(S) <- list("male","parent_educ","L1","M","sum")
for (n in 1:16) {
  S[n,"sum"] <- ( (( b["b_Y"] +
                      b["b_male_Y"] * S[n,"male"] +
                      b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                      b["b_A_Y"] * 1 +
                      b["b_L1_Y"] * S[n,"L1"] +
                      b["b_M_Y"] * S[n,"M"] +
                      b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
                    (( b["b_L1"] +
                         b["b_male_L1"] * S[n,"male"] +
                         b["b_parent_L1"] * S[n,"parent_educ"] +
                         b["b_A_L1"] * 1)^( S[n,"L1"] )) *
                    (( 1 - ( b["b_L1"] +
                               b["b_male_L1"] * S[n,"male"] +
                               b["b_parent_L1"] * S[n,"parent_educ"] +
                               b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) ) -
                    (( b["b_Y"] +
                        b["b_male_Y"] * S[n,"male"] +
                        b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                        b["b_A_Y"] * 0 +
                        b["b_L1_Y"] * S[n,"L1"] +
                        b["b_M_Y"] * S[n,"M"] +
                        b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
                    (( b["b_L1"] +
                         b["b_male_L1"] * S[n,"male"] +
                         b["b_parent_L1"] * S[n,"parent_educ"] +
                         b["b_A_L1"] * 0)^( S[n,"L1"] )) *
                    (( 1 - ( b["b_L1"] +
                               b["b_male_L1"] * S[n,"male"] +
                               b["b_parent_L1"] * S[n,"parent_educ"] +
                               b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) ) ) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }

CDE.M0.death <- sum(S[1:8,"sum"])
CDE.M1.death <- sum(S[9:16,"sum"])
```

```r
# quantitative outcome (QoL)
# we estimate both CDE, fixing do(M) = 0 et do(M) = 1 and using the
# corresponding lines in the S matrix
for (n in 1:16) {
  S[n,"sum"] <- ( (( b["mu_Y"] +
                      b["c_male_Y"] * S[n,"male"] +
                      b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                      b["c_A_Y"] * 1 +
                      b["c_L1_Y"] * S[n,"L1"] +
                      b["c_M_Y"] * S[n,"M"] +
                      b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
                    (( b["b_L1"] +
                         b["b_male_L1"] * S[n,"male"] +
                         b["b_parent_L1"] * S[n,"parent_educ"] +
                         b["b_A_L1"] * 1)^( S[n,"L1"] )) *
                    (( 1 - ( b["b_L1"] +
                               b["b_male_L1"] * S[n,"male"] +
                               b["b_parent_L1"] * S[n,"parent_educ"] +
                               b["b_A_L1"] * 1))^( 1 - S[n,"L1"] ))) -
                    (( b["mu_Y"] +
                        b["c_male_Y"] * S[n,"male"] +
                        b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                        b["c_A_Y"] * 0 +
                        b["c_L1_Y"] * S[n,"L1"] +
                        b["c_M_Y"] * S[n,"M"] +
                        b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
                    (( b["b_L1"] +
                         b["b_male_L1"] * S[n,"male"] +
                         b["b_parent_L1"] * S[n,"parent_educ"] +
                         b["b_A_L1"] * 0)^( S[n,"L1"] )) *
                    (( 1 - ( b["b_L1"] +
                               b["b_male_L1"] * S[n,"male"] +
                               b["b_parent_L1"] * S[n,"parent_educ"] +
                               b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) ) ) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }

CDE.M0.qol <- sum(S[1:8,"sum"])
CDE.M1.qol <- sum(S[9:16,"sum"])

return(list(CDE.M0.death = CDE.M0.death, CDE.M1.death = CDE.M1.death,
            CDE.M0.qol = CDE.M0.qol, CDE.M1.qol = CDE.M1.qol))
```

```
}
```

```
true.CDE2.no.inter <- true.CDE.time.var(interaction = 0)

true.CDE2.with.inter <- true.CDE.time.var(interaction = 1)
```

Setting $do(M = 0)$, the controlled direct effects $\text{CDE}_{M=0} = \mathbb{E}\left(Y_{1,0}\right) - \mathbb{E}\left(Y_{0,0}\right)$ are:

- 0.064 for death and -5 for quality of life without interaction
- 0.064 for death and -5 for quality of life with interaction

Setting $do(M = 1)$, the controlled direct effects $\text{CDE}_{M=1} = \mathbb{E}\left(Y_{1,1}\right) - \mathbb{E}\left(Y_{0,1}\right)$ are:

- 0.064 for death and -5 for quality of life without interaction
- 0.094 for death and -10 for quality of life with interaction

11.2.3 Marginal randomized direct and indirect effects

The following function `true.marg.random.time.var` can be used to run the calculation for the marginal randomized natural direct (marginal MRDE) and indirect effects (marginal MRIE).

```
true.marg.random.time.var <- function(interaction = NULL) {
  b <- param.causal.model.2(A.M.interaction = interaction)

  # marginal distribution of M
  M.S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^5))
  colnames(M.S) <- list("male","parent_educ","L1","M","A","sum")

  for (n in 1:32) {
    M.S[n,"sum"] <- (( b["b_M"] +
                        b["b_male_M"] * M.S[n,"male"] +
                        b["b_parent_educ_M"] * M.S[n,"parent_educ"] +
                        b["b_L1_M"] * M.S[n,"L1"] +
                        b["b_A_M"] * M.S[n,"A"])^( M.S[n,"M"] )) *
      (( 1 - (b["b_M"] +
                b["b_male_M"] * M.S[n,"male"] +
                b["b_parent_educ_M"] * M.S[n,"parent_educ"] +
                b["b_L1_M"] * M.S[n,"L1"] +
                b["b_A_M"] * M.S[n,"A"]) )^( 1 - M.S[n,"M"] )) *
      (( b["b_L1"] +
```

```
          b["b_male_L1"] * M.S[n,"male"] +
          b["b_parent_L1"] * M.S[n,"parent_educ"] +
          b["b_A_L1"] * M.S[n,"A"])^( M.S[n,"L1"] )) *
     (( 1 - ( b["b_L1"] +
                b["b_male_L1"] * M.S[n,"male"] +
                b["b_parent_L1"] * M.S[n,"parent_educ"] +
                b["b_A_L1"] * M.S[n,"A"]))^( 1 - M.S[n,"L1"] ))
   }

 M0.A0.L00 <- sum(M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                       M.S[,"parent_educ"]==0,"sum"])
 M0.A0.L01 <- sum(M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                       M.S[,"parent_educ"]==1,"sum"])
 M0.A0.L10 <- sum(M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                       M.S[,"parent_educ"]==0,"sum"])
 M0.A0.L11 <- sum(M.S[M.S[,"M"]==0 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                       M.S[,"parent_educ"]==1,"sum"])

 M1.A0.L00 <- sum(M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                       M.S[,"parent_educ"]==0,"sum"])
 M1.A0.L01 <- sum(M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==0 &
                       M.S[,"parent_educ"]==1,"sum"])
 M1.A0.L10 <- sum(M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                       M.S[,"parent_educ"]==0,"sum"])
 M1.A0.L11 <- sum(M.S[M.S[,"M"]==1 & M.S[,"A"]==0 & M.S[,"male"]==1 &
                       M.S[,"parent_educ"]==1,"sum"])

 M0.A1.L00 <- sum(M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                       M.S[,"parent_educ"]==0,"sum"])
 M0.A1.L01 <- sum(M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                       M.S[,"parent_educ"]==1,"sum"])
 M0.A1.L10 <- sum(M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                       M.S[,"parent_educ"]==0,"sum"])
 M0.A1.L11 <- sum(M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                       M.S[,"parent_educ"]==1,"sum"])

 M1.A1.L00 <- sum(M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                       M.S[,"parent_educ"]==0,"sum"])
 M1.A1.L01 <- sum(M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==0 &
                       M.S[,"parent_educ"]==1,"sum"])
 M1.A1.L10 <- sum(M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                       M.S[,"parent_educ"]==0,"sum"])
 M1.A1.L11 <- sum(M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==1 &
                       M.S[,"parent_educ"]==1,"sum"])
```

```
# binary outcome (death)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4),
           rep(NA,n=2^4), rep(NA,n=2^4))
colnames(S) <- list("male","parent_educ","L1","M","sum.psi11",
                    "sum.psi10", "sum.psi00")
for (n in 1:16) {
  S[n,"sum.psi11"] <-  ( b["b_Y"] +                                      # A=1
                           b["b_male_Y"] * S[n,"male"] +
                           b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["b_A_Y"] * 1 +
                           b["b_L1_Y"] * S[n,"L1"] +
                           b["b_M_Y"] * S[n,"M"] +
                           b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
        ((M1.A1.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +             # A'=1
          M1.A1.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
          M1.A1.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
          M1.A1.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( S[n,"M"] )) *
    ((M0.A1.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +
      M0.A1.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
      M0.A1.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
      M0.A1.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( 1 - S[n,"M"] )) *
    (( b["b_L1"] +                                                       # A=1
       b["b_male_L1"] * S[n,"male"] +
       b["b_parent_L1"] * S[n,"parent_educ"] +
       b["b_A_L1"] * 1)^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
             b["b_male_L1"] * S[n,"male"] +
             b["b_parent_L1"] * S[n,"parent_educ"] +
             b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  S[n,"sum.psi10"] <-  ( b["b_Y"] +                                      # A=1
                           b["b_male_Y"] * S[n,"male"] +
                           b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["b_A_Y"] * 1 +
                           b["b_L1_Y"] * S[n,"L1"] +
                           b["b_M_Y"] * S[n,"M"] +
                           b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
    ((M1.A0.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +                 # A'=0
      M1.A0.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
      M1.A0.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
      M1.A0.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( S[n,"M"] )) *
```

```
    ((M0.A0.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +
        M0.A0.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
        M0.A0.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
        M0.A0.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( 1 - S[n,"M"] )) *
    (( b["b_L1"] +                                                               # A=1
         b["b_male_L1"] * S[n,"male"] +
         b["b_parent_L1"] * S[n,"parent_educ"] +
         b["b_A_L1"] * 1)^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
               b["b_male_L1"] * S[n,"male"] +
               b["b_parent_L1"] * S[n,"parent_educ"] +
               b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  S[n,"sum.psi00"] <-  ( b["b_Y"] +                                              # A=0
                           b["b_male_Y"] * S[n,"male"] +
                           b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["b_A_Y"] * 0 +
                           b["b_L1_Y"] * S[n,"L1"] +
                           b["b_M_Y"] * S[n,"M"] +
                           b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
    ((M1.A0.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +                     # A'=0
        M1.A0.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
        M1.A0.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
        M1.A0.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( S[n,"M"] )) *
    ((M0.A0.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +
        M0.A0.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
        M0.A0.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
        M0.A0.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( 1 - S[n,"M"] )) *
    (( b["b_L1"] +                                                               # A=0
         b["b_male_L1"] * S[n,"male"] +
         b["b_parent_L1"] * S[n,"parent_educ"] +
         b["b_A_L1"] * 0)^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
               b["b_male_L1"] * S[n,"male"] +
               b["b_parent_L1"] * S[n,"parent_educ"] +
               b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
  }
```

```
mrNDE.death <- sum(S[,"sum.psi10"]) - sum(S[,"sum.psi00"])
mrNIE.death <- sum(S[,"sum.psi11"]) - sum(S[,"sum.psi10"])

# quantitative outcome (QoL)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4),
           rep(NA,n=2^4), rep(NA,n=2^4))
colnames(S) <- list("male","parent_educ","L1","M","sum.psi11", "sum.psi10",
                    "sum.psi00")
for (n in 1:16) {
  S[n,"sum.psi11"] <-  ( b["mu_Y"] +                                     # A=1
                           b["c_male_Y"] * S[n,"male"] +
                           b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["c_A_Y"] * 1 +
                           b["c_L1_Y"] * S[n,"L1"] +
                           b["c_M_Y"] * S[n,"M"] +
                           b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
    ((M1.A1.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +              # A'=1
        M1.A1.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
        M1.A1.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
        M1.A1.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( S[n,"M"] )) *
    ((M0.A1.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +
        M0.A1.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
        M0.A1.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
        M0.A1.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( 1 - S[n,"M"] )) *
    (( b["b_L1"] +                                                       # A=1
         b["b_male_L1"] * S[n,"male"] +
         b["b_parent_L1"] * S[n,"parent_educ"] +
         b["b_A_L1"] * 1)^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
               b["b_male_L1"] * S[n,"male"] +
               b["b_parent_L1"] * S[n,"parent_educ"] +
               b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  S[n,"sum.psi10"] <-  ( b["mu_Y"] +                                     # A=1
                           b["c_male_Y"] * S[n,"male"] +
                           b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["c_A_Y"] * 1 +
                           b["c_L1_Y"] * S[n,"L1"] +
                           b["c_M_Y"] * S[n,"M"] +
                           b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
    ((M1.A0.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +              # A'=0
```

```
          M1.A0.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
          M1.A0.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
          M1.A0.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( S[n,"M"] )) *
    ((M0.A0.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +
          M0.A0.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
          M0.A0.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
          M0.A0.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( 1 - S[n,"M"] )) *
    (( b["b_L1"] +                                                          # A=1
         b["b_male_L1"] * S[n,"male"] +
         b["b_parent_L1"] * S[n,"parent_educ"] +
         b["b_A_L1"] * 1)^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
               b["b_male_L1"] * S[n,"male"] +
               b["b_parent_L1"] * S[n,"parent_educ"] +
               b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  S[n,"sum.psi00"] <-  ( b["mu_Y"] +                                       # A=0
                           b["c_male_Y"] * S[n,"male"] +
                           b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["c_A_Y"] * 0 +
                           b["c_L1_Y"] * S[n,"L1"] +
                           b["c_M_Y"] * S[n,"M"] +
                           b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
    ((M1.A0.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +                  # A'=0
          M1.A0.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
          M1.A0.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
          M1.A0.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( S[n,"M"] )) *
    ((M0.A0.L00*(S[n,"male"]==0)*(S[n,"parent_educ"]==0) +
          M0.A0.L01*(S[n,"male"]==0)*(S[n,"parent_educ"]==1) +
          M0.A0.L10*(S[n,"male"]==1)*(S[n,"parent_educ"]==0) +
          M0.A0.L11*(S[n,"male"]==1)*(S[n,"parent_educ"]==1) )^( 1 - S[n,"M"] )) *
    (( b["b_L1"] +                                                          # A=0
         b["b_male_L1"] * S[n,"male"] +
         b["b_parent_L1"] * S[n,"parent_educ"] +
         b["b_A_L1"] * 0)^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
               b["b_male_L1"] * S[n,"male"] +
               b["b_parent_L1"] * S[n,"parent_educ"] +
               b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
```

```
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
    }

  mrNDE.qol <- sum(S[,"sum.psi10"]) - sum(S[,"sum.psi00"])
  mrNIE.qol <- sum(S[,"sum.psi11"]) - sum(S[,"sum.psi10"])

  return(list(mrNDE.death = mrNDE.death, mrNIE.death = mrNIE.death,
              mrNDE.qol = mrNDE.qol, mrNIE.qol = mrNIE.qol))
}
```

```
true.marg.random2.no.inter <- true.marg.random.time.var(interaction = 0)

true.marg.random2.with.inter <- true.marg.random.time.var(interaction = 1)
```

The marginal randomized direct effect MRDE $= \mathbb{E}\left(Y_{1,G_{0|L(0)}}\right) - \mathbb{E}\left(Y_{0,G_{0|L(0)}}\right)$ and the marginal randomized indirect effect MRIE $= \mathbb{E}\left(Y_{1,G_{1|L(0)}}\right) - \mathbb{E}\left(Y_{1,G_{0|L(0)}}\right)$ are respectively:

- 0.064 and 0.0112 for death without interaction
- 0.073882 and 0.0154000000000001 for death with interaction
- -4.99999999999999 and -1.26 for quality of life without interaction
- -6.64699999999998 and -1.96 for quality of life with interaction

11.2.4 Conditional randomized direct and indirect effects

The following function `true.cond.random.time.var` can be used to run the calculation for the conditional randomized natural direct (CRDE) and the conditional randomized indirect effects (CRIE).

```
true.cond.random.time.var <- function(interaction = NULL) {
  b <- param.causal.model.2(A.M.interaction = interaction)

  # binary outcome (death)
  S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4),
             rep(NA,n=2^4), rep(NA,n=2^4))
  colnames(S) <- list("male","parent_educ","L1","M","sum.psi11", "sum.psi10",
                      "sum.psi00")
  for (n in 1:16) {
    S[n,"sum.psi11"] <-  ( b["b_Y"] +                                # A=1
                             b["b_male_Y"] * S[n,"male"] +
                             b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                             b["b_A_Y"] * 1 +
```

```
                        b["b_L1_Y"] * S[n,"L1"] +
                        b["b_M_Y"] * S[n,"M"] +
                        b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
  (( b["b_M"] +                                                        # A'=1
      b["b_male_M"] * S[n,"male"] +
      b["b_parent_educ_M"] * S[n,"parent_educ"] +
      b["b_L1_M"] * S[n,"L1"] +
      b["b_A_M"] * 1 )^( S[n,"M"] )) *
  (( 1 - (b["b_M"] +
          b["b_male_M"] * S[n,"male"] +
          b["b_parent_educ_M"] * S[n,"parent_educ"] +
          b["b_L1_M"] * S[n,"L1"] +
          b["b_A_M"] * 1 ) )^( 1 - S[n,"M"] )) *
  (( b["b_L1"] +                                                       # A=1
      b["b_male_L1"] * S[n,"male"] +
      b["b_parent_L1"] * S[n,"parent_educ"] +
      b["b_A_L1"] * 1)^( S[n,"L1"] )) *
  (( 1 - ( b["b_L1"] +
           b["b_male_L1"] * S[n,"male"] +
           b["b_parent_L1"] * S[n,"parent_educ"] +
           b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) *
  ((b["p_L0_male"])^(S[n,"male"])) *
  ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
  ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
  ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

S[n,"sum.psi10"] <-  ( b["b_Y"] +                                      # A=1
                        b["b_male_Y"] * S[n,"male"] +
                        b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                        b["b_A_Y"] * 1 +
                        b["b_L1_Y"] * S[n,"L1"] +
                        b["b_M_Y"] * S[n,"M"] +
                        b["b_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
  (( b["b_M"] +                                                        # A'=0
      b["b_male_M"] * S[n,"male"] +
      b["b_parent_educ_M"] * S[n,"parent_educ"] +
      b["b_L1_M"] * S[n,"L1"] +
      b["b_A_M"] * 0 )^( S[n,"M"] )) *
  (( 1 - (b["b_M"] +
          b["b_male_M"] * S[n,"male"] +
          b["b_parent_educ_M"] * S[n,"parent_educ"] +
          b["b_L1_M"] * S[n,"L1"] +
          b["b_A_M"] * 0 ) )^( 1 - S[n,"M"] )) *
  (( b["b_L1"] +                                                       # A=1
      b["b_male_L1"] * S[n,"male"] +
```

```
          b["b_parent_L1"] * S[n,"parent_educ"] +
          b["b_A_L1"] * 1)^( S[n,"L1"] )) *
     (( 1 - ( b["b_L1"] +
                b["b_male_L1"] * S[n,"male"] +
                b["b_parent_L1"] * S[n,"parent_educ"] +
                b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) *
     ((b["p_L0_male"])^(S[n,"male"])) *
     ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
     ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
     ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

   S[n,"sum.psi00"] <-  ( b["b_Y"] +                                        # A=0
                          b["b_male_Y"] * S[n,"male"] +
                          b["b_parent_educ_Y"] * S[n,"parent_educ"] +
                          b["b_A_Y"] * 0 +
                          b["b_L1_Y"] * S[n,"L1"] +
                          b["b_M_Y"] * S[n,"M"] +
                          b["b_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
     (( b["b_M"] +                                                          # A'=0
          b["b_male_M"] * S[n,"male"] +
          b["b_parent_educ_M"] * S[n,"parent_educ"] +
          b["b_L1_M"] * S[n,"L1"] +
          b["b_A_M"] * 0 )^( S[n,"M"] )) *
     (( 1 - (b["b_M"] +
               b["b_male_M"] * S[n,"male"] +
               b["b_parent_educ_M"] * S[n,"parent_educ"] +
               b["b_L1_M"] * S[n,"L1"] +
               b["b_A_M"] * 0 ) )^( 1 - S[n,"M"] )) *
     (( b["b_L1"] +                                                         # A=0
          b["b_male_L1"] * S[n,"male"] +
          b["b_parent_L1"] * S[n,"parent_educ"] +
          b["b_A_L1"] * 0)^( S[n,"L1"] )) *
     (( 1 - ( b["b_L1"] +
                b["b_male_L1"] * S[n,"male"] +
                b["b_parent_L1"] * S[n,"parent_educ"] +
                b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) *
     ((b["p_L0_male"])^(S[n,"male"])) *
     ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
     ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
     ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
   }

 crNDE.death <- sum(S[,"sum.psi10"]) - sum(S[,"sum.psi00"])
 crNIE.death <- sum(S[,"sum.psi11"]) - sum(S[,"sum.psi10"])
```

```
# quantitative outcome (QoL)
S <- cbind(expand.grid(c(0,1),c(0,1),c(0,1),c(0,1)), rep(NA,n=2^4),
           rep(NA,n=2^4), rep(NA,n=2^4))
colnames(S) <- list("male","parent_educ","L1","M","sum.psi11", "sum.psi10",
                    "sum.psi00")
for (n in 1:16) {
  S[n,"sum.psi11"] <-  ( b["mu_Y"] +                                          # A=1
                           b["c_male_Y"] * S[n,"male"] +
                           b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["c_A_Y"] * 1 +
                           b["c_L1_Y"] * S[n,"L1"] +
                           b["c_M_Y"] * S[n,"M"] +
                           b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
    (( b["b_M"] +                                                               # A'=1
         b["b_male_M"] * S[n,"male"] +
         b["b_parent_educ_M"] * S[n,"parent_educ"] +
         b["b_L1_M"] * S[n,"L1"] +
         b["b_A_M"] * 1 )^( S[n,"M"] )) *
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 1 ) )^( 1 - S[n,"M"] )) *
    (( b["b_L1"] +                                                              # A=1
         b["b_male_L1"] * S[n,"male"] +
         b["b_parent_L1"] * S[n,"parent_educ"] +
         b["b_A_L1"] * 1)^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
               b["b_male_L1"] * S[n,"male"] +
               b["b_parent_L1"] * S[n,"parent_educ"] +
               b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  S[n,"sum.psi10"] <-  ( b["mu_Y"] +                                          # A=1
                           b["c_male_Y"] * S[n,"male"] +
                           b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["c_A_Y"] * 1 +
                           b["c_L1_Y"] * S[n,"L1"] +
                           b["c_M_Y"] * S[n,"M"] +
                           b["c_AM_Y"] * 1 * S[n,"M"] * b["A.M.inter"] ) *
    (( b["b_M"] +                                                               # A'=0
         b["b_male_M"] * S[n,"male"] +
```

```
         b["b_parent_educ_M"] * S[n,"parent_educ"] +
         b["b_L1_M"] * S[n,"L1"] +
         b["b_A_M"] * 0 )^( S[n,"M"] )) *
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 0 ) )^( 1 - S[n,"M"] )) *
    (( b["b_L1"] +                                                      # A=1
         b["b_male_L1"] * S[n,"male"] +
         b["b_parent_L1"] * S[n,"parent_educ"] +
         b["b_A_L1"] * 1)^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
               b["b_male_L1"] * S[n,"male"] +
               b["b_parent_L1"] * S[n,"parent_educ"] +
               b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) *
    ((b["p_L0_male"])^(S[n,"male"])) *
    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
    ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))

  S[n,"sum.psi00"] <-  ( b["mu_Y"] +                                    # A=0
                           b["c_male_Y"] * S[n,"male"] +
                           b["c_parent_educ_Y"] * S[n,"parent_educ"] +
                           b["c_A_Y"] * 0 +
                           b["c_L1_Y"] * S[n,"L1"] +
                           b["c_M_Y"] * S[n,"M"] +
                           b["c_AM_Y"] * 0 * S[n,"M"] * b["A.M.inter"] ) *
    (( b["b_M"] +                                                       # A'=0
         b["b_male_M"] * S[n,"male"] +
         b["b_parent_educ_M"] * S[n,"parent_educ"] +
         b["b_L1_M"] * S[n,"L1"] +
         b["b_A_M"] * 0 )^( S[n,"M"] )) *
    (( 1 - (b["b_M"] +
              b["b_male_M"] * S[n,"male"] +
              b["b_parent_educ_M"] * S[n,"parent_educ"] +
              b["b_L1_M"] * S[n,"L1"] +
              b["b_A_M"] * 0 ) )^( 1 - S[n,"M"] )) *
    (( b["b_L1"] +                                                      # A=0
         b["b_male_L1"] * S[n,"male"] +
         b["b_parent_L1"] * S[n,"parent_educ"] +
         b["b_A_L1"] * 0)^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
               b["b_male_L1"] * S[n,"male"] +
               b["b_parent_L1"] * S[n,"parent_educ"] +
```

```
                 b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) *
      ((b["p_L0_male"])^(S[n,"male"])) *
      ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
      ((b["p_L0_parent_low_educ_lv"])^(S[n,"parent_educ"])) *
      ((1 - b["p_L0_parent_low_educ_lv"])^(1 - S[n,"parent_educ"]))
    }

  crNDE.qol <- sum(S[,"sum.psi10"]) - sum(S[,"sum.psi00"])
  crNIE.qol <- sum(S[,"sum.psi11"]) - sum(S[,"sum.psi10"])

  return(list(crNDE.death = crNDE.death, crNIE.death = crNIE.death,
              crNDE.qol = crNDE.qol, crNIE.qol = crNIE.qol))
}
```

```
true.cond.random2.no.inter <- true.cond.random.time.var(interaction = 0)

true.cond.random2.with.inter <- true.cond.random.time.var(interaction = 1)
```

The conditional randomized direct effect CRDE $= \mathbb{E}\left(Y_{1,\Gamma_{0|L(0),L(1)}}\right) - \mathbb{E}\left(Y_{0,\Gamma_{0|L(0),L(1)}}\right)$ and conditional randomized indirect effect CRIE $= \mathbb{E}\left(Y_{1,\Gamma_{1|L(0),L(1)}}\right) - \mathbb{E}\left(Y_{1,\Gamma_{0|L(0),L(1)}}\right)$ are respectively:

- 0.0672 and 0.00800000000000001 for death without interaction,
- 0.078282 and 0.011 for death with interaction,
- -5.36 and -0.900000000000006 for quality of life without interaction,
- -7.207 and -1.40000000000001 for quality of life with interaction.

Table 11.2: True values with time varying confounders

Effects	Without $A * M$ interaction	with $A * M$ interaction
Binary outcome		
Average total effect (ATE)	0.0752	0.089282
Controlled direct effect (CDE)		
- CDE, setting do(M=0)	0.064	0.064
- CDE, setting do(M=1)	0.064	0.094

Effects	Without $A * M$ interaction	with $A * M$ interaction
Marginal randomized		
- marginal rNDE	0.064	0.073882
- marginal rNIE	0.0112	0.0154000000000001
Conditional randomized		
- conditional rNDE	0.0672	0.078282
- conditional rNIE	0.00800000000000001	0.011
Quantitative outcome		
Average total effect (ATE)	-6.26	-8.607
Controlled direct effect (CDE)		
- CDE, setting do(M=0)	-5	-5
- CDE, setting do(M=1)	-5	-10
Marginal randomized		
- marginal rNDE	-4.99999999999999	-6.64699999999998
- marginal rNIE	-1.26	-1.96
Conditional randomized		
- conditional rNDE	-5.36	-7.207
- conditional rNIE	-0.900000000000006	-1.40000000000001

Daniel, R M, S N Cousens, B L De Stavola, M G Kenward, and J A C Sterne. 2013. "Methods for Dealing with Time-Dependent Confounding." *Statistics in Medicine* 32 (9): 1584–1618.

Laan, Mark J. van der, and Sherri Rose. 2011. *Targeted Learning: Causal*

Inference for Observational and Experimental Data. 1st ed. Springer Series in Statistics. New York, NY: Springer.

Lin, Sheng-Hsuan, Jessica Young, Roger Logan, Eric J. Tchetgen Tchetgen, and Tyler J VanderWeele. 2017. "Parametric Mediational g-Formula Approach to Mediation Analysis with Time-Varying Exposures, Mediators, and Confounders." *Epidemiology* 28 (2): 266–74.

Naimi, Ashley I, and Brian W Whitcomb. 2020. "Estimating Risk Ratios and Risk Differences Using Regression." *American Journal of Epidemiology* 189 (6): 508–10.

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.

Taubman, Sarah L, James M Robins, Murray A Mittleman, and Miguel A Hernán. 2009. "Intervening on Risk Factors for Coronary Heart Disease: An Application of the Parametric g-Formula." *International Journal of Epidemiology* 38 (6): 1599–1611.

Valeri, Linda, and Tyler J VanderWeele. 2013. "Mediation Analysis Allowing for Exposure-Mediator Interactions and Causal Interpretation: Theoretical Assumptions and Implementation with SAS and SPSS Macros." *Psycol Methods* 18 (2): 137–50.

VanderWeele, Tyler J. 2009. "Marginal Structural Models for the Estimation of Direct and Indirect Effects." *Epidemiology* 20: 18–26.

———. 2013. "A Three-Way Decomposition of a Total Effect into Direct, Indirect, and Interactive Effects." *Epidemiology* 13: 224–32.

———. 2014. "A Unification of Mediation and Interaction. A 4-Way Decomposition." *Epidemiology* 25: 749–61.

VanderWeele, Tyler J, and Eric J Tchetgen Tchetgen. 2017. "Mediation Analysis with Time Varying Exposures and Mediators." *J R Stat Soc Series B Stat Methodol* 79 (3): 917–38.