

Mediation Workshop

Benoît Lepage

2024-10-28

Contents

1	Introduction	7
2	Software	9
3	Data sets	11
3.1	General presentation of the data used in our examples	11
3.2	Data generating mechanisms	12
4	Baron and Kenny, structural equation models	15
4.1	Baron and Kenny approach	15
4.2	Path analysis and Structural Equation Modeling	19
5	Traditional regression models	31
5.1	Estimation of the Average Total Effect (ATE)	31
5.2	Two-way decomposition	34
5.3	Three-way decomposition	42
5.4	Four-way decomposition	45
6	G-computation	55
6.1	Estimation of the Average Total Effect (ATE)	55
6.2	Estimation of Controlled Direct Effects (CDE)	58
6.3	Estimation of Natural Direct (NDE) and Indirect Effects (NIE) .	68
6.4	Estimation of “Marginal” Randomized/Interventional Natural Direct (MRDE) and Indirect Effects (MRIE)	77

6.5	Using the CMAverse package for 2-way, 3-way and 4-way decomposition	85
6.6	Estimation of “Conditional” Randomized/Interventional Natural Direct (CRDE) and Indirect Effects (CRIE)	93
7	Inverse Probability of Treatment Weighting (IPTW)	99
7.1	Estimation of the Average total effect	99
7.2	Estimation of the Controlled direct effect (CDE)	101
7.3	Estimation of the PNDE and TNIE by Inverse Odds Ratio Weighing	105
7.4	Estimation of “Conditional” Randomized/Interventional Natural Direct (CRDE) and Indirect Effects (CRIE)	109
8	Marginal structural models	113
8.1	MSM for the Average Total Effect (ATE)	113
8.2	MSM for Controlled Direct Effects	119
8.3	MSM for Natural Direct and Indirect Effects	132
8.4	MSM estimated by the CMAverse package	136
8.5	“Unified” approach for estimating Natural Direct and Indirect effects	146
9	Targeted Maximum Likelihood Estimation (TMLE)	165
9.1	TMLE for the ATE	166
9.2	TMLE of the Controlled direct effect (CDE)	174
9.3	One-step and TMLE estimator of the Marginal Randomized/Interventional Direct and Indirect Effects	178
10	Appendix A: Data generating mechanisms	185
10.1	First causal model: Data generating mechanism without mediator-outcome confounder affected by the exposure	185
10.2	Second causal model: Data generating mechanism with mediator-outcome confounder affected by the exposure	187
10.3	Simulation of the four data sets used in examples	189

<i>CONTENTS</i>	5
11 Appendix B: Calculation of the true causal quantities	195
11.1 True causal quantities without mediator-outcome confounder affected by the exposure	195
11.2 True causal quantities with mediator-outcome confounder affected by the exposure	235

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 4.3.3).

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

- `lavaan` package
- `semPlot` package
- `CMAverse` package
- `sandwich` package
- `boot` package
- `regmedint` package
- `mediation` package
- `ltmle` package
- `intmed` package
- `medflex` package
- `questionr` package
- `geepack` package
- `SuperLearner` package
- `xgboost` package
- `tidyverse` package

- `s13` package
- `hal9001` package
- `medoutcon` package
- `glmnet` package

Chapter 3

Data sets

3.1 General presentation of the data used in our examples

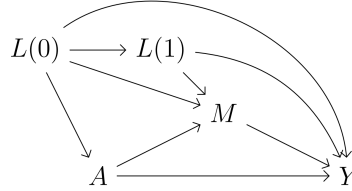
Four data sets have been simulated, each containing 7 variables:

- 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_soc_env`, a binary variable indicating if the participant has been exposed to a deprived social environment);
- 1 exposure of interest (denoted A in the DAGs):
 - `A0_PM2.5`, a binary variable indicating if the participants had been exposed to a high level of $PM_{2.5}$;
- 1 confounder of the mediator-outcome relationship (denoted $L(1)$ in the DAGs):
 - `L1`, a binary variable indicating if the participant is overweight (1 for being overweight, 0 for not being overweight);
- 1 mediator of interest (denoted M in the DAGs):
 - `M_diabetes`, a binary variable indicating if the participant has type 2 diabetes;
- 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 .



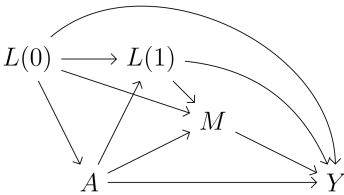
DAG of models \mathcal{M}_1 and \mathcal{M}_{1*}

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).

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.



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

Figure 3.2: Causal model 2

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:
 $\beta_A \times \gamma_M$.

```
## Import data
rm(list=ls())
df1 <- read.csv(file = "./data/df1.csv")

## Model 1 to estimate the total effect:
model.tot.A.QoL <- lm(Y_qol ~ A0_PM2.5 + L0_male + L0_soc_env,
                      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_PM2.5      -5.0961     0.3486  -14.617 < 2e-16 *** <- Total effect
# L0_male       -1.1486     0.2194   -5.235 1.68e-07 ***
# L0_soc_env    -3.4441     0.2295  -15.005 < 2e-16 ***

# The total effect of being exposed to high levels of PM2.5 on Quality of life
# is approximately equal to an average decrease of -5.1 on the QoL scale,
# given by the A0_PM2.5 coefficient:
model.tot.A.QoL$coefficients["A0_PM2.5"]
# -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 regression
## for example
logit.model.A.M <- glm(M_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env,
                      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 ***
# AO_PM2.5     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_soc_env   0.32683    0.04731   6.908 4.91e-12 ***
exp(coefficients(logit.model.A.M)["AO_PM2.5"])
# Odds ratio = 1.753609 for the effect of being exposed to high levels of PM_2.5
# on the mediator (probability of type 2 diabetes)

## 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 ~ AO_PM2.5 + M_diabetes + L1 + L0_male + L0_soc_env,
                    data = df1)
summary(model.A.M.QoL)
# Coefficients:
#               Estimate Std. Error t value Pr(>|t|)
# (Intercept)  74.7858    0.2130 351.178 < 2e-16 ***
# AO_PM2.5     -3.9650    0.3212 -12.345 < 2e-16 *** <- Direct effect
# M_diabetes   -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_soc_env   -2.8876    0.2112 -13.674 < 2e-16 ***

# The direct effect of PM_2.5 is approximately -4.0 given by the AO_PM2.5 coefficient
model.A.M.QoL$coefficients["AO_PM2.5"]
# -3.965038

## Following the Baron & Kenny Steps, we would conclude that :
# - There is a significant total effect of PM_2.5 on Quality of Life (Model 1)
# - There is a significant effect of PM_2.5 on the mediator (diabetes) (Model 2)
# - There is a significant effect of the mediator (diabetes) on QoL (model 3)
# - The direct effect is significantly non-null
# => Conclusion: Diabetes partially mediates the relationship between PM_2.5 and QoL

### Estimation of the indirect effect:
### We can apply the difference in coefficient method to estimate the indirect effect:
### subtract the direct effect from the Total effect:
ind.effect.dif.meth <- (model.tot.A.QoL$coefficients["AO_PM2.5"] -
                       model.A.M.QoL$coefficients["AO_PM2.5"])
# -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_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env,
                      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_PM2.5     0.127180    0.014446   8.804  < 2e-16 *** <- effect of A on M
# L0_male      0.052363    0.009091   5.760 8.67e-09 ***
# L0_soc_env   0.065806    0.009511   6.919 4.83e-12 ***

## product of coefficient method:
ind.effect.prod.meth <- (linear.model.A.M$coefficients["A0_PM2.5"] *
                        model.A.M.QoL$coefficients["M_diabetes"])

# -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_PM2.5 + L0_male + L0_soc_env,
                      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_PM2.5     0.060247    0.012796   4.708 2.53e-06 *** <- Total effect
# L0_male      0.050285    0.008053   6.244 4.43e-10 ***
# L0_soc_env   0.059565    0.008425   7.070 1.65e-12 ***
# On a risk difference scale the total effect of being exposed to high levels of
# PM2.5 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_PM2.5 + M_diabetes + L1 + L0_male + L0_soc_env,
                      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_PM2.5     0.051509   0.012759   4.037 5.45e-05 *** <- Direct effect
# M_diabetes   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_soc_env   0.055676   0.008389   6.637 3.36e-11 ***

# The direct effect is approximately +5.2% given by the A0_PM2.5 coefficient
model.A.M.death$coefficients["A0_PM2.5"]
# 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$coefficients["A0_PM2.5"] - model.A.M.death$coefficients["A0_PM2.5"]
# 0.008737889, i.e. approximately 0.9%

# or the product of coefficients using the previous model 2bis and model 3:
linear.model.A.M$coefficients["A0_PM2.5"] * model.A.M.death$coefficients["M_diabetes"]
# 0.008234978, i.e. approximately 0.8%
```

4.2 Path analysis and Structural Equation Modeling

4.2.1 First application, without intermediate confounding affected by the exposure

Path analyses can be considered as a “generalization” of the product of coefficients method. They combine graphical representation of causal structures, a set of linear equations and assumptions concerning the covariance structure of residuals.

When the causal model includes latent variables (represented by a measurement model), we refer to the analysis as structural equation modeling.

Variables which does not receive causal inputs from any other variable in the diagram are called exogenous variables. Unless the assumption is explicitly made that two exogenous variables are uncorrelated (because they do not have

any common causal factor), it is preferable to consider them as not independent (connecting them by double arrows).

In the `df1.csv` data set, there are 3 exogenous variables: `L0_male`, `L0_soc_env` and `L1`.

We will use the `lavaan` R package to run path analyses.

- The first step is to write the model syntax
- The second step is to analyze the model with the dataset

For a complete overview of the analyses that can be performed using `lavaan`, you can read:

- the `lavaan` tutorial
- A `lavaan` Compendium for Structural Equation Modeling in Educational Research, (Jak, Suzanne and Jorgensen, Terrence D. 2024)
- R Cookbook for Structural Equation Modeling, (Jiang, Ge 2023)

First step. Let us write the model syntax to analyze the effect of being exposed to high levels of $PM_{2.5}$ on quality of life, mediated by type-2 diabetes.

We have to define:

- regression models for the endogenous variables: exposure to $PM_{2.5}$ (`A0_PM2.5`), type-2 diabetes (`M_diabetes`) and quality of life (`Y_qol`)
- covariance assumptions for the exogenous variables
- we can also define additional parameters of interest (in our case, direct, indirect and total effects)

```
rm(list=ls())
df1 <- read.csv(file = "./data/df1.csv")

library(lavaan)

sem.QoL <- "                                     # models are written between quotes
## Regression models
## we can also add the names of some path coefficients
  A0_PM2.5 ~ L0_male + L0_soc_env
  M_diabetes ~ L0_male + L0_soc_env + b.A * A0_PM2.5 + L1    # label b.A path coef
  Y_qol ~ L0_male + L0_soc_env + c.A*A0_PM2.5 + L1 + c.M*M_diabetes # label c.A and c.M
## Covariances
# Assuming a non-null covariance between confounders
# covariances are represented with a double tilde ~~ (double arrow)
```

```
# (note: in the data-generating system, the null assumptions were true)
L0_male ~~ L0_soc_env
L0_male ~~ L1
L0_soc_env ~~ L1

## We can define other parameters to estimate, using the := syntax
## we want the direct, indirect and total effects:
direct := c.A
indirect := b.A * c.M
total := (b.A * c.M) + c.A
"
```

Second step. Let us analyze the model with the `df1.csv` dataset using the `sem` function from the `lavaan` package.

Binary exogenous variables (`L0_male`, `L0_soc_env` and `L1`) can be coded as simple dummy (0/1) variables, there is no need to declare them as factors (`ordered`).

Note that in the following example, we will not declare the other binary variables (`A0_PM2.5`, `M_diabetes`) as factors and let them as simple dummy variables.

Running the `sem` function with the `df1.csv` dataset, we obtain:

- information about the estimator
- estimates of the regression models
- estimates of the covariance between exogenous variables
- estimates of the residual variances
- estimates of the additional parameters (direct, indirect and total effects)

For datasets with only continuous variables, `lavaan` would estimate the covariance structure by maximum likelihood estimation. For categorical data, weighted least square estimators (WLS) are typically used. Applying WLS to continuous data is referred to as asymptotical distribution-free estimation (ADF) in the SEM literature. ADF estimation requires large sample sizes (> 2000 or 5000). (Jak, Suzanne and Jorgensen, Terrence D. 2024)

In order to avoid underestimating standard errors and confidence intervals, robust estimators (“sandwich” and bootstrap) are usually applied if the data are not multivariate normal.

```
set.seed(1234)
fit.qol <- lavaan::sem(model = sem.QoL,
  fixed.x = FALSE, # ask for covariance of exogenous variables
                  # to be freely estimated
  # estimator = "DWLS", # for WLS estimation
```

```

se = "boot", # estimation of SE by bootstrap
# se = "robust.sem", # for sandwich-type SE
bootstrap = 100, # better with 1000 bootstrap samples or more
data = df1)
summary(fit.qol,
        ci = TRUE) # add 95%CI in the output
# You will get 2 warning messages:
# 1) we did not standardize the outcome, its variance is very large variance compared
#    to the other binary variables (not a problem)
# 2) 3 bootstrap runs failed or did not converge.
#    => The estimations rely on the 97 other bootstrap samples

# lavaan 0.6-18 ended normally after 23 iterations
#
#      Estimator                      ML
#      Optimization method          NLMINB
#      Number of model parameters      20
#
#      Number of observations          10000
#
# Model Test User Model:
#      Test statistic                  0.211
#      Degrees of freedom              1
#      P-value (Chi-square)            0.646
#
# Parameter Estimates:
#      Standard errors                  Bootstrap
#      Number of requested bootstrap draws      100
#      Number of successful bootstrap draws      97
#
# Regressions:
#
#               Estimate Std.Err  z-value  P(>|z|)  ci.lower ci.upper
# AO_PM2.5 ~
#   LO_male          0.040   0.005   7.310   0.000   0.029   0.050
#   LO_sc_nv          0.058   0.006   9.243   0.000   0.046   0.071
# M_diabetes ~
#   LO_male          0.053   0.009   5.626   0.000   0.033   0.074
#   LO_sc_nv          0.066   0.010   6.948   0.000   0.047   0.088
# AO_PM2.5 (b.A)      0.127   0.015   8.475   0.000   0.100   0.156
#   L1               0.070   0.011   6.391   0.000   0.046   0.088
# Y_qol ~
#   LO_male          -0.719   0.190  -3.777   0.000  -1.081  -0.314
#   LO_sc_nv          -2.888   0.210 -13.760   0.000  -3.262  -2.408
# AO_PM2.5 (c.A)      -3.965   0.331 -11.977   0.000  -4.631  -3.262
#   L1               -3.425   0.189 -18.136   0.000  -3.808  -3.043

```

```

#   M_diabts (c.M)   -8.714    0.211  -41.394    0.000   -9.082   -8.280
#
# Covariances:
#               Estimate Std.Err  z-value  P(>|z|)  ci.lower ci.upper
# L0_male ~~
#   L0_soc_env      -0.003    0.003   -1.055    0.291   -0.008    0.004
#   L1              -0.002    0.002   -0.887    0.375   -0.006    0.002
# L0_soc_env ~~
#   L1              -0.001    0.002   -0.452    0.651   -0.006    0.003
#
# Variances:
#               Estimate Std.Err  z-value  P(>|z|)  ci.lower ci.upper
# .A0_PM2.5         0.099    0.002   43.273    0.000    0.094    0.103
# .M_diabetes        0.205    0.002  114.604    0.000    0.200    0.208
# .Y_gol            100.882    1.409   71.573    0.000   98.185  103.725
#   L0_male          0.250    0.000 9358.475    0.000    0.250    0.250
#   L0_soc_env        0.229    0.001  166.244    0.000    0.226    0.232
#   L1                0.207    0.002  117.109    0.000    0.204    0.210
#
# Defined Parameters:
#               Estimate Std.Err  z-value  P(>|z|)  ci.lower ci.upper
# direct         -3.965    0.333  -11.915    0.000   -4.631   -3.262
# indirect       -1.104    0.129   -8.586    0.000   -1.359   -0.866
# total         -5.069    0.358  -14.170    0.000   -5.843   -4.343
#
# ?lavaan::parameterEstimates for more options on bootstrap SE estimations.
parameterEstimates(fit.qol,
  level = 0.95,
  boot.ci.type = "bca.simple") # "bca.simple" to correct for bias,
                                # but not acceleration
#               lhs op               rhs   label   est   se      z pvalue ci.lower ci.upper
# 1   A0_PM2.5 ~      L0_male          0.040 0.005    7.310 0.000    0.031 0.051
# 2   A0_PM2.5 ~      L0_soc_env        0.058 0.006    9.243 0.000    0.045 0.070
# ...
# 21   direct :=          c.A   direct  -3.965 0.333  -11.915 0.000   -4.544 -3.190
# 22   indirect :=        b.A*c.M indirect -1.104 0.129   -8.586 0.000   -1.401 -0.897
# 23   total := (b.A*c.M)+c.A   total  -5.069 0.358  -14.170 0.000   -5.658 -4.201

```

We recall that the results obtained using Baron & Kenny approach were:

- direct effect = -3.965
- indirect effect = -1.131 (using the “difference in coefficients” method) or
-1.108213 (using the “product of coefficients” method)
- total effect = -5.096

The path analysis results from `lavaan` are very close to the Baron & Kenny results.

We can plot the estimated model using the `semPlot` package (Figure 4.1):

```
library(semPlot)
semPaths(fit.qol,
  what = "est",
  layout = "tree2", # tree, tree2, spring
  rotation = 2, #exogenous on the left, endogenous on the right
  sizeMan = 10, # font size of manifest variable names
  nCharNodes = 0,
  nCharEdges = 0, # don't limit variable name lengths
  edge.label.cex = 0.6,
  curvePivot = TRUE,
  fade = FALSE)
```

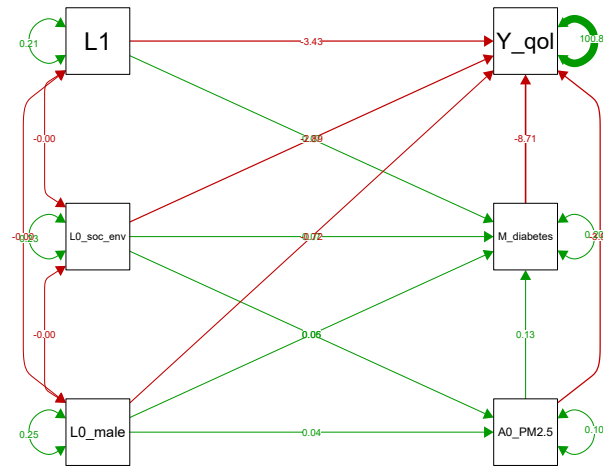


Figure 4.1: Plot of the SEM

4.2.2 Dealing with categorical endogenous variables

In the dataset, we can declare the binary endogenous variables as `ordered` variables, so that the `sem` function will detect them automatically as categorical variables, and apply a Weighted Least Square estimation.

When endogenous variables are binary or ordered categorical variables, the SEM-method will assume the existence of latent normal variables that underlies the observed categorical variables (corresponding to *latent response variables* (LRV)). SEM will then fit the model to the LRV instead of the categorical variables. The distribution of a categorical variable is linked to the LRV using “thresholds”, defined from the cumulative distribution function of a standard normal distribution to match the probabilities of observed levels (similarly to probit regressions). (Jak, Suzanne and Jorgensen, Terrence D. 2024)

```
## We first create a data.frame, setting the exposure and mediator to ordered variables
df1.cat <- data.frame(L0_male = df1$L0_male,
                      L0_soc_env = df1$L0_soc_env,
                      A0_PM2.5 = ordered(df1$A0_PM2.5),
                      L1 = df1$L1,
                      M_diabetes = ordered(df1$M_diabetes),
                      Y_qol = df1$Y_qol)

## We analyze the model with the new data.frame (df1.cat)
## In this example, we estimate standwich-type standard errors
fit.qol <- lavaan::sem(model = sem.QoL,
                      se = "robust.sem", # for sandwich-type SE
                      fixed.x = FALSE,
                      data = df1.cat)

# You will get 2 warnings:
# 1) A generalized inverse for A11 submatrix was used to solve a trouble
#    constructing W matrix
# 2) The variance-covariance matrix of the vcov does not appear to be positive definite.
#    Probably because the mean and variance of binary variable are collinear.
#    This message might be quite common and can be ignored
summary(fit.qol,
        ci = TRUE)

# lavaan 0.6-18 ended normally after 76 iterations
#
# Estimator                      DWLS # apply WLS estimator
# Optimization method           NLMINB
# Number of model parameters     24
#
# Number of observations         10000
#
# Model Test User Model:
#   Standard      Scaled
# Test Statistic      0.209      0.214
# Degrees of freedom      1      1
# P-value (Chi-square)    0.648      0.644
# Scaling correction factor      0.977
# Shift parameter          -0.000
```

```

# simple second-order correction
#
# Parameter Estimates:
#   Parameterization              Delta
#   Standard errors              Robust.sem
#   Information                  Expected
#   Information saturated (h1) model Unstructured
#
# Regressions:
#           Estimate Std.Err  z-value  P(>|z|)  ci.lower ci.upper
# A0_PM2.5 ~
#   L0_male          0.209    0.033    6.394    0.000    0.145    0.273
#   L0_soc_env        0.324    0.035    9.129    0.000    0.254    0.393
# M_diabetes ~
#   L0_male          0.129    0.027    4.846    0.000    0.077    0.181
#   L0_soc_env        0.156    0.029    5.463    0.000    0.100    0.212
#   A0_PM2.5 (b.A)    0.182    0.021    8.498    0.000    0.140    0.224
#   L1                0.199    0.028    7.083    0.000    0.144    0.254
# Y_gol ~
#   L0_male          -0.193    0.214   -0.899    0.369   -0.613    0.227
#   L0_soc_env       -2.130    0.232   -9.195    0.000   -2.585   -1.676
#   A0_PM2.5 (c.A)   -1.754    0.169  -10.351    0.000   -2.086   -1.422
#   L1               -3.070    0.229  -13.411    0.000   -3.519   -2.621
#   M_diabts (c.M)   -4.927    0.142  -34.802    0.000   -5.204   -4.649
#
# Covariances:
#   Estimate Std.Err  z-value  P(>|z|)  ci.lower ci.upper
# L0_male ~~
#   L0_soc_env      -0.003    0.002   -1.167    0.243   -0.007    0.002
#   L1              -0.002    0.002   -0.801    0.423   -0.006    0.003
# L0_soc_env ~~
#   L1              -0.001    0.002   -0.425    0.671   -0.005    0.003
#
# Intercepts:
#           Estimate Std.Err  z-value  P(>|z|)  ci.lower ci.upper
# .Y_gol       72.816    0.228   319.590    0.000   72.370   73.263
#   L0_male      0.501    0.005   100.197    0.000    0.491    0.511
#   L0_soc_env    0.644    0.003   187.422    0.000    0.637    0.651
#   L1            0.293    0.002   128.425    0.000    0.289    0.298
#
# Thresholds:
#           Estimate Std.Err  z-value  P(>|z|)  ci.lower ci.upper
# A0_PM2.5/t1    1.527    0.031   48.583    0.000    1.465    1.589
# M_diabetes/t1   0.810    0.026   30.789    0.000    0.758    0.861
#

```

```

# Variances:
#           Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
# .A0_PM2.5      0.965          0.965      0.965
# .M_diabetes     0.975          0.975      0.975
# .Y_gol         88.521      1.538    57.551    0.000    85.506    91.535
# L0_male        0.250      0.000 12499.500    0.000    0.250    0.250
# L0_soc_env     0.229      0.002   106.346    0.000    0.225    0.234
# L1             0.207      0.002    91.060    0.000    0.203    0.212
#
# Defined Parameters:
#           Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
# direct      -1.754      0.169   -10.351    0.000    -2.086    -1.422
# indirect     -0.896      0.099    -9.041    0.000    -1.091    -0.702
# total       -2.651      0.177   -14.942    0.000    -2.998    -2.303

```

The results of the direct, indirect and total effects are quite different from the Baron & Kenny estimations and the previous SEM (where categorical exogenous variables were treated as simple dummy variables).

This difference can probably be explained by the fact that the data generating system (used to simulate the data) does not imply latent normal variables underlying the categorical variables. Binary variables were directly simulated using binomial distributions.

If the exposure, mediator and outcome variables are a mix of continuous and categorical variables, (Iacobucci 2008) recommends to treat them as simple dummy variables (and as continuous variables when fitting the models).

4.2.3 Application with intermediate confounding affected by the exposure

Under the causal model where the intermediate confounder $L(1)$ of the $M - Y$ relationship is affected by the exposure A (Causal model 2, Figure 3.2), we can describe 4 directed paths connecting the exposure to the outcome :

- (i) the path $A \rightarrow Y$,
- (ii) the path $A \rightarrow L(1) \rightarrow Y$
- (iii) the path $A \rightarrow M \rightarrow Y$
- (iv) and the path $A \rightarrow L(1) \rightarrow M \rightarrow Y$

In the causal inference literature,

- the *Marginal Randomized (or Interventional) Indirect Effect* (MRIE), described by (VanderWeele and Tchetgen Tchetgen 2017), can be considered

as an analogue of the set of directed paths from A to Y going through the mediator M (i.e. paths (iii) $A \rightarrow M \rightarrow Y$ and (iv) $A \rightarrow L(1) \rightarrow M \rightarrow Y$)

- and the *Marginal Randomized (or Interventional) Direct Effect* (MRDE), corresponds to the other paths (i.e. paths (i) $A \rightarrow Y$ and (ii) $A \rightarrow L(1) \rightarrow Y$).

Regarding the *conditional effects* described by (Zheng and van der Laan 2017),

- the *Conditional Randomized (or Interventional) Indirect Effect* (CRIE), described by Lin et al. (2017), can be considered as an analogue of the specific paths from A to Y going only through the mediator M (i.e. path (iii) $A \rightarrow M \rightarrow Y$)
- and the *Conditional Randomized (or Interventional) Direct Effect* (CRDE), corresponds to the other paths (i.e. paths (i) $A \rightarrow Y$ and (ii) $A \rightarrow L(1) \rightarrow Y$ and (iv) $A \rightarrow L(1) \rightarrow M \rightarrow Y$).

We can estimate the 4 paths by SEM analysis. Using the `df2.csv` dataset (without $A * M$ interaction effect on the outcome), we will explore the effect of the exposure to high levels of $PM_{2.5}$ on quality of life (Y), through type-2 diabetes (mediator of interest M), where overweight $L(1)$ is affected by the exposure.

```
df2 <- read.csv(file = "data/df2.csv")
head(df2)
#   L0_male L0_soc_env AO_PM2.5 L1 M_diabetes Y_death   Y_qol
# 1      0      1      0  1      0      0 91.91819
# 2      1      1      1  0      0      1 67.53221
# 3      1      1      0  0      0      0 75.56249
# 4      1      0      0  0      0      0 89.77055
# 5      1      1      0  1      1     1 63.22353
# 6      1      1      0  1      0      0 77.87975

## quality of life ----
## First step: write the model syntax
sem.QoL.df2 <- "
  ## SEM for quantitative outcome (QoL)
  # Regression models
  AO_PM2.5 ~ a.L01 * L0_male + a.L02 * L0_soc_env
  L1 ~ b.L01 * L0_male + b.L02 * L0_soc_env + b.A * AO_PM2.5
  M_diabetes ~ c.L01 * L0_male + c.L02 * L0_soc_env + c.A * AO_PM2.5 + c.L1 * L1
  Y_qol ~ d.01 * L0_male + d.02 * L0_soc_env + d.A * AO_PM2.5 + d.L1 * L1 + d.M * M_diabetes

  # Assuming the possibility of non-null covariance between confounders
  # (note: in the data-generating system, the null assumptions were true)
  L0_male ~~ L0_soc_env
```

```

# define other parameters: specific paths, direct, indirect and total effects
path.A_Y := d.A
path.A_L1_Y := b.A * d.L1
path.A_M_Y := c.A * d.M
path.A_L1_M_Y := b.A * c.L1 * d.M
MRDE := d.A + (b.A * d.L1)
MRIE := (c.A * d.M) + (b.A * c.L1 * d.M)
CRDE := d.A + (b.A * d.L1) + (b.A * c.L1 * d.M)
CRIE := (c.A * d.M)
total := d.A + (b.A * d.L1) + (c.A * d.M) + (b.A * c.L1 * d.M)
"

## Second step: estimate the model with the df2 dataset
fit.qol.df2 <- lavaan::sem(model = sem.QoL.df2,
                           se = "robust.sem", # for sandwich-type SE
                           fixed.x = FALSE,
                           data = df2)

summary(fit.qol.df2,
        ci = TRUE)
# lavaan 0.6-18 ended normally after 18 iterations
#
# Estimator                               ML
# Optimization method                     NLMINB
# Number of model parameters              21
#
# Number of observations                   10000
#
# Model Test User Model:
#   Test statistic                        0.000
#   Degrees of freedom                    0
#
# Parameter Estimates:
#   Standard errors                      Robust.sem
#   Information                          Expected
#   Information saturated (h1) model     Structured
#
# Regressions:
#               Estimate Std.Err z-value P(>|z|) ci.lower ci.upper
# AO_PM2.5 ~
#   L0_mal (a.L01)    0.040   0.006   6.339   0.000   0.027   0.052
#   L0_sc_ (a.L02)    0.058   0.006   9.527   0.000   0.046   0.070
# L1 ~
#   L0_mal (b.L01)   -0.043   0.009   -4.638   0.000   -0.062   -0.025
#   L0_sc_ (b.L02)    0.069   0.010    7.166   0.000    0.050    0.088
#   AO_PM2 (b.A)      0.226   0.016   14.373   0.000    0.195    0.257

```

```

# M_diabetes ~
#   LO_mal (c.L01)    0.052    0.009    5.662    0.000    0.034    0.070
#   LO_sc_ (c.L02)    0.064    0.010    6.692    0.000    0.045    0.082
#   AO_PM2  (c.A)     0.072    0.016    4.567    0.000    0.041    0.103
#   L1      (c.L1)     0.194    0.010    18.962   0.000    0.174    0.215
# Y_gol ~
#   LO_mal (d.01)    -0.725    0.202    -3.591    0.000    -1.121   -0.329
#   LO_sc_ (d.02)    -2.881    0.210   -13.722   0.000    -3.292   -2.469
#   AO_PM2 (d.A)     -3.926    0.329   -11.945   0.000    -4.570   -3.282
#   L1      (d.L1)    -5.165    0.217   -23.794   0.000    -5.591   -4.740
#   M_dbts  (d.M)     -8.698    0.217   -40.020   0.000    -9.124   -8.272
#
# Covariances:
#               Estimate Std.Err  z-value  P(>|z|) ci.lower ci.upper
# LO_male ~~
#   LO_soc_env    -0.003    0.002    -1.167    0.243    -0.007    0.002
#
# Variances:
#               Estimate Std.Err  z-value  P(>|z|) ci.lower ci.upper
#   .AO_PM2.5      0.099    0.002    41.100    0.000    0.094    0.103
#   .L1             0.219    0.002   135.196    0.000    0.216    0.222
#   .M_diabetes     0.212    0.002   123.351    0.000    0.208    0.215
#   .Y_gol          100.878    1.409    71.598    0.000    98.117   103.640
#   LO_male         0.250    0.000  24999.950    0.000    0.250    0.250
#   LO_soc_env      0.229    0.001   166.381    0.000    0.227    0.232
#
# Defined Parameters:
#               Estimate Std.Err  z-value  P(>|z|) ci.lower ci.upper
# path.A_Y      -3.926    0.329   -11.945    0.000    -4.570   -3.282
# path.A_L1_Y   -1.168    0.095   -12.322    0.000    -1.354   -0.982
# path.A_M_Y    -0.625    0.138    -4.536    0.000    -0.895   -0.355
# path.A_L1_M_Y -0.382    0.035   -10.999    0.000    -0.451   -0.314
# MRDE          -5.093    0.336   -15.170    0.000    -5.751   -4.435
# MRIE          -1.007    0.139    -7.228    0.000    -1.280   -0.734
# CRDE          -5.476    0.343   -15.942    0.000    -6.149   -4.803
# CRIE          -0.625    0.138    -4.536    0.000    -0.895   -0.355
# total         -6.101    0.370   -16.500    0.000    -6.825   -5.376

```

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_PM2.5:M_diabetes` 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 high levels of $PM_{2.5}$, compared to the mean outcome had the whole population been unexposed: $ATE = \mathbb{E}(Y_{A=1}) - \mathbb{E}(Y_{A=0})$.

For the quantitative outcome, the ATE of the exposure to high levels of $PM_{2.5}$ ($A = 1$ versus $A = 0$) on the quality of life score Y can be estimated using a traditional linear regression of `Y_qol` on `A0_PM2.5`, 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.

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

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

```
## Import data
rm(list=ls())
df1_int <- read.csv(file = "./data/df1_int.csv")

## For quantitative outcomes, apply a linear regression of Y on A (A0_PM2.5),
## adjusted for the baseline confounders L(0):
trad_ATE_qol <- lm(Y_qol ~ A0_PM2.5 + L0_male + L0_soc_env,
                  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_PM2.5 + L0_male + L0_soc_env,
                    family = gaussian("identity"),
                    data = df1_int)

## Use the regression coefficient of the exposure (A0_PM2.5) to estimate the ATE
ATE_trad_qol <- coefficients(trad_ATE_qol)["A0_PM2.5"]
# -7.210089

ATE_trad_death <- coefficients(trad_ATE_death)["A0_PM2.5"]
# 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)
?sandwich
## the sandwich() function returns a sandwich covariance matrix estimate
## for the Quality of Life outcome
ATE_trad_qol <- list(ATE = coef(trad_ATE_qol)["A0_PM2.5"],
                  lo = coef(trad_ATE_qol)["A0_PM2.5"] - qnorm(0.975) *
                    sqrt(sandwich(trad_ATE_qol)["A0_PM2.5", "A0_PM2.5"]),
```



```

hi = coef(trad_ATE_qol)["A0_PM2.5"] + qnorm(0.975) *
      sqrt(sandwich(trad_ATE_qol)["A0_PM2.5", "A0_PM2.5"]))

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

## for death outcome
ATE_trad_death <- list(ATE = coef(trad_ATE_death)["A0_PM2.5"],
  lo = coef(trad_ATE_death)["A0_PM2.5"] - qnorm(0.975) *
    sqrt(sandwich(trad_ATE_death)["A0_PM2.5", "A0_PM2.5"]),
  hi = coef(trad_ATE_death)["A0_PM2.5"] + qnorm(0.975) *
    sqrt(sandwich(trad_ATE_death)["A0_PM2.5", "A0_PM2.5"]))

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_PM2.5 + L0_male + L0_soc_env,
    data = boot_dat)
  mod.death <- glm(Y_death ~ A0_PM2.5 + L0_male + L0_soc_env,
    family = gaussian("identity"),
    data = boot_dat)
  est <- c(coef(mod.qol)["A0_PM2.5"],
    coef(mod.death)["A0_PM2.5"])
  return(est)
}

set.seed(1234)
boot_est <- boot(df1_int, bootfunc, R = 1000)

## the 95% CI for the estimation of the ATE of ACE on QoL is:
boot.ci(boot_est, index = 1, type = "norm")
# (-7.955, -6.445 )

## the 95% CI for the estimation of the ATE of ACE on death is:
boot.ci(boot_est, index = 2, type = "norm")
# ( 0.0501, 0.1046 )

```

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).

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

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

```
TE_death_model <- glm(Y_death ~ A0_PM2.5 + L0_male + L0_soc_env,
  family = "binomial",
  data = df1_int)
res_TE_death <- summary(TE_death_model)
tot.effect.death.OR <- list(OR = exp(coef(res_TE_death)["A0_PM2.5", "Estimate"]),
  lo = exp(coef(res_TE_death)["A0_PM2.5", "Estimate"] -
    qnorm(0.975) *
    coef(res_TE_death)["A0_PM2.5", "Std. Error"]),
  hi = exp(coef(res_TE_death)["A0_PM2.5", "Estimate"] +
    qnorm(0.975) *
    coef(res_TE_death)["A0_PM2.5", "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)

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

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

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

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

```
trad_qol_am <- lm(Y_qol ~ A0_PM2.5 + M_diabetes + A0_PM2.5:M_diabetes +
  L0_male + L0_soc_env + L1,
  data = df1_int)
gamma.A.q <- coef(trad_qol_am)["A0_PM2.5"]
```

```

gamma.M.q <- coef(trad_qol_am)["M_diabetes"]
gamma.AM.q <- coef(trad_qol_am)["A0_PM2.5:M_diabetes"]

trad_m <- glm(M_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env + L1,
              family = "binomial",
              data = df1_int)
beta.0 <- coef(trad_m)["(Intercept)"]
beta.A <- coef(trad_m)["A0_PM2.5"]

trad_death_am <- glm(Y_death ~ A0_PM2.5 + M_diabetes + A0_PM2.5:M_diabetes +
                     L0_male + L0_soc_env + L1,
                     family = "binomial",
                     data = df1_int)
gamma.A.d <- coef(trad_death_am)["A0_PM2.5"]
gamma.M.d <- coef(trad_death_am)["M_diabetes"]
gamma.AM.d <- coef(trad_death_am)["A0_PM2.5:M_diabetes"]

```

5.2.1 Controlled Direct Effect

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

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

$$CDE_m = \hat{\gamma}_A + \hat{\gamma}_{A*M} \times m$$

```

### 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:

$$OR^{CDE_m} = \exp(\hat{\gamma}_A + \hat{\gamma}_{A*M} \times m)$$

```

### For a binary outcome
## setting the mediator to M=0
trad_OR_CDE_death_m0 <- exp(gamma.A.d + gamma.AM.d * 0)

```

```

trad_OR_CDE_death_m0
# OR_CDE_{M=0} = 1.442942

## setting the mediator to M=1
trad_OR_CDE_death_m1 <- exp(gamma.A.d + gamma.AM.d * 1)
trad_OR_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:

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

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

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

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)):

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

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

The alternative TNDE and PNIE can be estimated by:

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

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

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:

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

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

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

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

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

and

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

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

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

and

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

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

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

and

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

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

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

and

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

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

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

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

```
### 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_qo1",          # outcome variable
  avar = "A0_PM2.5",       # exposure
  mvar = "M_diabetes",     # mediator
  cvar = c("L0_male",     # confounders
            "L0_soc_env",
            "L1"),
  eventvar = "event",      # only for survival outcome
  ## Values at which effects are evaluated
  a0 = 0,
  a1 = 1,
  m_cde = 0,
  c_cond = c(0,0,0),      # covariate level
  ## Model types
  mreg = "logistic",
  yreg = "linear",
  ## Additional specification
  interaction = TRUE,     # presence of A:M interaction term
                          # in the outcome model
  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
# tnle -1.5011902 0.20821830 -7.209694 5.608847e-13 -1.9092905 -1.0930898
# tnle -5.4367728 0.34049175 -15.967414 0.000000e+00 -6.1041244 -4.7694213
# pnle -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 + tnle) = (tnde + pnle)
#       pm = proportion mediated = tnle / te
```

For binary outcomes:

```
regmedint_bin <- regmedint(data = df1_int,
  ## Variables
  yvar = "Y_death", # outcome variable
  avar = "A0_PM2.5", # exposure
  mvar = "M_diabetes", # mediator
  cvar = c("LO_male", # confounders
           "LO_soc_env",
           "L1"),
  #eventvar = "event", # only for survival outcome
  ## Values at which effects are evaluated
  a0 = 0,
  a1 = 1,
  m_cde = 0,
  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", "tnle", "tnde", "pnle", "te"),
  c("est", "lower", "upper")])

#### Mediation analysis
#      est      lower      upper
# cde  1.442942  1.191195  1.747893
# pnle  1.448035  1.245470  1.683545
# tnle  1.050029  1.013842  1.087509
# tnle  1.450344  1.257042  1.673371
# pnle  1.048358  1.029285  1.067783
# te   1.520479  1.316954  1.755457
```

The `regmedint` package gives the same results as those calculated manually using the regression coefficients.

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

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:

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

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

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

```
### 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_qo1 <- gamma.AM.q *
  (exp(beta.0 + beta.A) / (1 + exp(beta.0 + beta.A)) -
   exp(beta.0) / (1 + exp(beta.0)))
trad_MIE_qo1
# -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* ($RR^{TE} - 1$), 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:

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

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

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

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:

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

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

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

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

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

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

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

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

and

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

```
### 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_qo1 <- 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_qo1
# 0.02408674
```

In this example, the excess relative risk of the exposure to high levels of $PM_{2.5}$ is $\approx 52.3\%$, and of this excess relative risk...:

- $\approx 44.8\%$ is attributable to the PNDE of $PM_{2.5}$,
- $\approx 4.8\%$ is attributable to the PNIE of $PM_{2.5}$ through type-2 diabetes,
- $\approx 2.4\%$ is attributable to the mediated interactive effect between $PM_{2.5}$ and type-2 diabetes

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

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 $CDE_{M=0}$ 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)).

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

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

and

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

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* ($RR^{TE} - 1$) (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:

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

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

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

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

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

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

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

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:

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

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

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

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

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

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

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

```
### 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 high levels of $PM_{2.5}$ is $\approx 52.3\%$, and of this excess relative risk...:

- $\approx 40.2\%$ is attributable to the CDE of $PM_{2.5}$,
- $\approx 4.8\%$ is attributable to the PNIE of $PM_{2.5}$ through type-2 diabetes,
- $\approx 2.4\%$ is attributable to the mediated interactive effect between $PM_{2.5}$ and type-2 diabetes
- and $\approx 4.6\%$ is attributable to the $(PM_{2.5} * \text{type-2 diabetes})$ 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 2-way, 3-way and 4-way decompositions. Estimations of the CDE($M=0$), PNIE, MIE and INTref presented above can be obtained as we show in the following example.

Standard errors and confidence intervals can be computed by delta-method or by bootstrap.

Note that the results from regression based methods (applying the `estimation = "paramfunc"` argument) are not marginal expectations, but conditional expectations of direct, indirect, total, and interaction effects.

For continuous outcomes:

```

library(CMAverse)
?CMAverse::cmest

### For the continuous outcome

```

```
## Closed-form parameter function estimation and delta method inference
res_rb_param_delta <- ctest(data = df1_int,
  model = "rb", # for "regression based" (rb) approach
  outcome = "Y_qol", # outcome variable
  exposure = "A0_PM2.5", # exposure variable
  mediator = "M_diabetes", # mediator
  basec = c("L0_male", # confounders
    "L0_soc_env",
    "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" or "bootstrap"

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 ***
# pnle     -0.90951   0.12266 -1.14992 -0.669 1.22e-13 *** PNIE
# tnle     -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 ***
# pnle(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
# pnies  -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
# pnies  -0.9095061 -1.1499166 -0.6690957 1.219025e-13

```

For binary outcomes:

```

### For the binary outcome
## Closed-form parameter function estimation and delta method inference
res_rb_param_delta <- cmest(data = df1_int,
                           model = "rb", # for "regression based" (rb) approach
                           outcome = "Y_death", # outcome variable
                           exposure = "AO_PM2.5", # exposure variable
                           mediator = "M_diabetes", # mediator
                           basec = c("LO_male", # confounders
                                     "LO_soc_env",
                                     "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$effect.ci.high - 1)[c("Rpnde", "Rpnie"),]
# Estimate lower95CI upper95CI
# Rpnde 0.44803473 0.24546995 0.68354491
# Rpnie 0.04835753 0.02928532 0.06778313
# and the excess relative risk due to MIE is given by:
data.frame("Estimate" = res_rb_param_delta$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("ERintmed"),]
# Estimate lower95CI upper95CI P.value
# 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","ERpnie"),]
```

#	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). Analogs of the 3-way and 4-way decompositions are also given by the `CMAverse` package.

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 $\bar{Q} = \mathbb{E}(Y \mid A, L(0))$
2. Use this estimate to predict an outcome for each subject $\hat{\bar{Q}}(A = 0)_i$ and $\hat{\bar{Q}}(A = 1)_i$, by evaluating the regression fit \bar{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}

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

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

```
## 0. Import data
rm(list=ls())
df2_int <- read.csv(file = "./data/df2_int.csv")

## 1. Estimate Qbar
Q.tot.death <- glm(Y_death ~ A0_PM2.5 + L0_male + L0_soc_env,
                  family = "binomial", data = df2_int)
Q.tot.qol <- glm(Y_qol ~ A0_PM2.5 + L0_male + L0_soc_env,
                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$A0_PM2.5 <- 1
data.A0$A0_PM2.5 <- 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 <- 1000
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_PM2.5 + L0_male + L0_soc_env,
                    family = "binomial", data = bootData)
  Q.tot.qol <- glm(Y_qol ~ A0_PM2.5 + L0_male + L0_soc_env,
                 family = "gaussian", data = bootData)

  boot.A.1 <- boot.A.0 <- bootData
  boot.A.1$A0_PM2.5 <- 1
  boot.A.0$A0_PM2.5 <- 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] 0.05612907 0.10928734

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] -9.157856 -7.563526

```

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 high levels of $\text{PM}_{2.5}$ (setting $A = 1$), compared to the mean outcome had the whole population been unexposed (setting $A = 0$), while keeping the mediator (type-2 diabetes) 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:

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

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:

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

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:

$$\Psi_{\text{NPMLE}}^{\text{CDE}_m} = \frac{1}{n} \sum \left[\hat{\bar{Q}}_Y(A=1, L(1)=1, M=m) \times \hat{\bar{Q}}_{L(1)}(A=1) + \hat{\bar{Q}}_Y(A=1, L(1)=0, M=m) \times (1 - \hat{\bar{Q}}_{L(1)}(A=1)) \right] \\ - \left[\hat{\bar{Q}}_Y(A=0, L(1)=1, M=m) \times \hat{\bar{Q}}_{L(1)}(A=0) + \hat{\bar{Q}}_Y(A=0, L(1)=0, M=m) \times (1 - \hat{\bar{Q}}_{L(1)}(A=0)) \right]$$

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 (where the $L(1)$ variables are put in an arbitrary order).

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

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

$$\bar{Q}_Y(A, L(1), M) = \mathbb{E}(Y \mid L(0), A, L(1), M) \quad (6.3)$$

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 $\hat{Q}_Y(A = a, L(1) = l(1)_a, M = m)$
5. Estimate the controlled direct effect Ψ_{CDE_m} by the sample mean:

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

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

```
rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")

## 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_soc_env + A0_PM2.5,
                family = "binomial", data = df2_int)

## 2. Fit parametric models for the outcome conditional on past
Y.death.model <- glm(Y_death ~ L0_male + L0_soc_env + A0_PM2.5 + L1 +
                    M_diabetes + A0_PM2.5:M_diabetes,
                    family = "binomial", data = df2_int)

Y.qol.model <- glm(Y_qol ~ L0_male + L0_soc_env + A0_PM2.5 + L1 +
                  M_diabetes + A0_PM2.5:M_diabetes,
                  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$A0_PM2.5 <- 0
data.A1$A0_PM2.5 <- 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:
## {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$M_diabetes <- 0
data.A0.M1$M_diabetes <- 1
data.A1.M0$M_diabetes <- 0
data.A1.M1$M_diabetes <- 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)
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 preceding M , to estimate $\bar{Q}_Y = \mathbb{E}(Y \mid L(0), A, L(1), M)$;
2. Use this estimate to predict an outcome for each subject $\hat{\bar{Q}}_Y(A = a', M = m)_i$, by evaluating the regression fit \bar{Q}_Y at the chosen value for the exposure $A = a'$ and the mediator $M = m$;
3. Fit a quasibinomial or a linear regression of the predicted values $\hat{\bar{Q}}_Y(M = m)_i$ conditional on the exposure A and baseline confounders $L(0)$ to estimate $\bar{Q}_{L(1)} = \mathbb{E}(\hat{\bar{Q}}_Y(A = a', M = m) \mid L(0), A)$;
4. Use this estimate to predict the outcome $\hat{\bar{Q}}_{L(1)}(A = a')_i$ for each subject, by evaluating the regression fit $\bar{Q}_{L(1)}$ at $A = a'$;
5. Use the sample mean to estimate $\Psi_{\text{gcomp}}^{\text{CDE}_m}$

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

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.

```
rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")

## 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_soc_env + A0_PM2.5 + L1 +
                     M_diabetes + A0_PM2.5:M_diabetes,
                     family = "binomial", data = df2_int)
Y.qol.model <- glm(Y_qol ~ L0_male + L0_soc_env + A0_PM2.5 + L1 +
                  M_diabetes + A0_PM2.5:M_diabetes,
                  family = "gaussian", data = df2_int)

# 2) Generate predicted values by evaluating the regression setting the exposure
#    and the mediator at exposure history of interest:
#    {A=1,M=0}, {A=0,M=0}, {A=1,M=1}, {A=0,M=1}
data.Ais0.Mis0 <- data.Ais0.Mis1 <- df2_int
data.Ais1.Mis0 <- data.Ais1.Mis1 <- df2_int

data.Ais0.Mis0$A0_PM2.5 <- 0
data.Ais0.Mis0$M_diabetes <- 0

data.Ais0.Mis1$A0_PM2.5 <- 0
data.Ais0.Mis1$M_diabetes <- 1

data.Ais1.Mis0$A0_PM2.5 <- 1
data.Ais1.Mis0$M_diabetes <- 0

data.Ais1.Mis1$A0_PM2.5 <- 1
data.Ais1.Mis1$M_diabetes <- 1

Q.L2.death.A0M0 <- predict(Y.death.model, newdata = data.Ais0.Mis0, type="response")
Q.L2.death.A0M1 <- predict(Y.death.model, newdata = data.Ais0.Mis1, type="response")
Q.L2.death.A1M0 <- predict(Y.death.model, newdata = data.Ais1.Mis0, type="response")
Q.L2.death.A1M1 <- predict(Y.death.model, newdata = data.Ais1.Mis1, type="response")

Q.L2.qol.A0M0 <- predict(Y.qol.model, newdata = data.Ais0.Mis0, type="response")
Q.L2.qol.A0M1 <- predict(Y.qol.model, newdata = data.Ais0.Mis1, type="response")
Q.L2.qol.A1M0 <- predict(Y.qol.model, newdata = data.Ais1.Mis0, type="response")
Q.L2.qol.A1M1 <- predict(Y.qol.model, newdata = data.Ais1.Mis1, type="response")

## 3) Regress the predicted values conditional on the exposure A
##    and baseline confounders L(0)
L1.death.A0M0.model <- glm(Q.L2.death.A0M0 ~ L0_male + L0_soc_env + A0_PM2.5,
```

```

        family = "quasibinomial", data = df2_int)
L1.death.AOM1.model <- glm(Q.L2.death.AOM1 ~ L0_male + L0_soc_env + A0_PM2.5,
        family = "quasibinomial", data = df2_int)
L1.death.A1M0.model <- glm(Q.L2.death.A1M0 ~ L0_male + L0_soc_env + A0_PM2.5,
        family = "quasibinomial", data = df2_int)
L1.death.A1M1.model <- glm(Q.L2.death.A1M1 ~ L0_male + L0_soc_env + A0_PM2.5,
        family = "quasibinomial", data = df2_int)

L1.qol.AOM0.model <- glm(Q.L2.qol.AOM0 ~ L0_male + L0_soc_env + A0_PM2.5,
        family = "gaussian", data = df2_int)
L1.qol.AOM1.model <- glm(Q.L2.qol.AOM1 ~ L0_male + L0_soc_env + A0_PM2.5,
        family = "gaussian", data = df2_int)
L1.qol.A1M0.model <- glm(Q.L2.qol.A1M0 ~ L0_male + L0_soc_env + A0_PM2.5,
        family = "gaussian", data = df2_int)
L1.qol.A1M1.model <- glm(Q.L2.qol.A1M1 ~ L0_male + L0_soc_env + A0_PM2.5,
        family = "gaussian", data = df2_int)

## 4) generate predicted values by evaluating the regression at exposure
## of interest: {A=1} & {A=0}
Q.L1.death.AOM0 <- predict(L1.death.AOM0.model, newdata = data.Ais0.Mis0, type="response")
Q.L1.death.AOM1 <- predict(L1.death.AOM1.model, newdata = data.Ais0.Mis1, type="response")
Q.L1.death.A1M0 <- predict(L1.death.A1M0.model, newdata = data.Ais1.Mis0, type="response")
Q.L1.death.A1M1 <- predict(L1.death.A1M1.model, newdata = data.Ais1.Mis1, type="response")

Q.L1.qol.AOM0 <- predict(L1.qol.AOM0.model, newdata = data.Ais0.Mis0, type="response")
Q.L1.qol.AOM1 <- predict(L1.qol.AOM1.model, newdata = data.Ais0.Mis1, type="response")
Q.L1.qol.A1M0 <- predict(L1.qol.A1M0.model, newdata = data.Ais1.Mis0, type="response")
Q.L1.qol.A1M1 <- predict(L1.qol.A1M1.model, newdata = data.Ais1.Mis1, 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.A1M0) - mean(Q.L1.death.AOM0)
CDE.death.m0.gcomp.ice
# [1] 0.06342833

CDE.qol.m0.gcomp.ice <- mean(Q.L1.qol.A1M0) - mean(Q.L1.qol.AOM0)
CDE.qol.m0.gcomp.ice
# [1] -4.869509

# CDE setting M=1
CDE.death.m1.gcomp.ice <- mean(Q.L1.death.A1M1) - mean(Q.L1.death.AOM1)
CDE.death.m1.gcomp.ice
# [1] 0.08812318

CDE.qol.m1.gcomp.ice <- mean(Q.L1.qol.A1M1) - mean(Q.L1.qol.AOM1)

```

```
CDE.qol.m1.gcomp.ice
# [1] -10.38144
```

6.2.2.1 G-computation by ICE using the ltmle package

The ltmle package can be used to estimate Controlled Direct Effects by g-computation.

An application is shown below.

```
library(ltmle)
rm(list=ls())
df2_int <- read.csv(file = "./data/df2_int.csv")

# the data set should be composed of continuous or binary variables,
# ordered following the cause-effect sequence of each variables.
# Note that within a set of exposures or intermediate confounders measured at a
# single discrete time t, any causal sequence can be applied (for example,
# with several L1 variable, it can be {L1.1, L1.2, L1.3} or {L1.2,L1.3,L1.1},
# without any consequences on the estimation.
df.death <- subset(df2_int, select = -Y_qol)
df.qol <- subset(df2_int, select = -Y_death)

## 1) Define Q formulas (Qbar_L1 and Qbar_Y functions)
Q_formulas.death <- c(L1 = "Q.kplus1 ~ L0_male + L0_soc_env + A0_PM2.5",
  Y_death = "Q.kplus1 ~ L0_male + L0_soc_env + L1 +
    A0_PM2.5 * M_diabetes") # add interaction
Q_formulas.qol <- c(L1 = "Q.kplus1 ~ L0_male + L0_soc_env + A0_PM2.5",
  Y_qol = "Q.kplus1 ~ L0_male + L0_soc_env + L1 +
    A0_PM2.5 * M_diabetes") # add interaction

## 2) Define g formulas (needed for the ltmle package) but they are not used
# with the g-computation estimator
g_formulas <- c("A0_PM2.5 ~ L0_male + L0_soc_env",
  "M_diabetes ~ L0_male + L0_soc_env + A0_PM2.5 + L1")

## 3) Use the ltmle() function
# arguments:
# - Anodes: indicate the exposure and the mediator variables
# - Lnodes: indicate the intermediate confounders (+/- baseline confounders)
# - Cnodes: censoring nodes, useless in our example
# - Ynodes: outcome variable
# - survivalOutcome = FALSE in our example
# - abar: list of the two values used to define counterfactual outcomes
# for the contrast of interest. For example, setting M=0,
# CDE(M=0) = E(Y_{A=1,M=0}) - E(Y_{A=0,M=0})
```



```

# - rule: to define dynamic rules (useless in our example)
# - gbounds = c(0.01, 1) by default. This parameter is not used with g-computation
# - Yrange = NULL, can be used to define range (min,max) for continuous outcomes
# - SL.library = "glm", will apply main terms glm models.
#       The argument can be used to specify SuperLearner libraries.
#       However, simple glm models might be preferable as data.adaptive
#       algorithms rely on cross-validation, which is difficult and long to
#       implement with the bootstrap procedure needed for 95% confidence
#       intervals
# - stratify = FALSE by default. If TRUE, glm estimations are stratified for
#       each counterfactual scenario defined in abar.
# - estimate.time = FALSE. If TRUE, print a rough estimate of computation time
# - iptw.only = FALSE, useless with g-computation
# - variance.method = "ic", computation is faster than with "tmle" which
#       is useless with g-comp: variance estimates rely on
#       influence curves which cannot be used with g-comp because
#       g-computation is not a asymptotically efficient estimator.
# - observation.weights = NULL, can be used to specify individual weights

## With a binary outcome,  $CDE(M=1) = P(Y_{\{A=1,M=0\}} = 1) - P(Y_{\{A=0,M=0\}} = 1)$ 
ltmle.gcomp.CDE.MO <- ltmle(data = df.death,
  Anodes = c("AO_PM2.5", "M_diabetes"),
  Lnodes = c("L1"),
  Ynodes = c("Y_death"), # binary outcome
  survivalOutcome = FALSE,
  Qform = Q_formulas.death, # Q formulas
  gform = g_formulas, # g formulas
  abar = list(c(1,0),c(0,0)), #  $Y_{\{A=1,M=0\}}$  vs  $Y_{\{A=0,M=0\}}$ 
  rule = NULL,
  gbounds = c(0.01, 1), # by default
  Yrange = NULL,
  deterministic.g.function = NULL,
  stratify = FALSE,
  SL.library = "glm",
  SL.cvControl = list(),
  estimate.time = FALSE,
  gcomp = TRUE, # should be TRUE for g-computation
  iptw.only = FALSE,
  deterministic.Q.function = NULL,
  variance.method = "ic",
  observation.weights = NULL,
  id = NULL)

summary(ltmle.gcomp.CDE.MO)
# Additive Treatment Effect:
# Parameter Estimate: 0.063428 # same as manual computation

```

```

# Estimated Std Err: 0.018159
# p-value: 0.00047789
# 95% Conf Interval: (0.027836, 0.09902) # those 95%CI should not be used
# => apply a bootstrap computation instead

## With a continuous outcome,  $CDE(M=1) = E(Y_{\{A=1,M=1\}}) - E(Y_{\{A=0,M=1\}})$ 
ltmle.gcomp.CDE.M1 <- ltmle(data = df.qol,
  Anodes = c("A0_PM2.5", "M_diabetes"),
  Lnodes = c("L1"),
  Ynodes = c("Y_qol"), # continous outcome
  survivalOutcome = FALSE,
  Qform = Q_formulas.qol, # Q formulas
  gform = g_formulas, # g formulas
  abar = list(c(1,1),c(0,1)), #  $Y_{\{A=1,M=1\}}$  vs  $Y_{\{A=0,M=1\}}$ 
  rule = NULL,
  gbounds = c(0.01, 1), # by default
  Yrange = NULL,
  deterministic.g.function = NULL,
  stratify = FALSE,
  SL.library = "glm",
  SL.cvControl = list(),
  estimate.time = FALSE,
  gcomp = TRUE, # should be TRUE for g-computation
  iptw.only = FALSE,
  deterministic.Q.function = NULL,
  variance.method = "ic",
  observation.weights = NULL,
  id = NULL)

summary(ltmle.gcomp.CDE.M1)
# Additive Treatment Effect:
# Parameter Estimate: -10.432
# Estimated Std Err: 0.55975
# p-value: <2e-16
# 95% Conf Interval: (-11.529, -9.335) those 95%CI should not be used
# => apply a bootstrap computation instead

# For quantitative outcomes, the outcome is first transformed into a continuous variab
# with [0;1] range:  $Y' = (Y - \min(Y)) / (\max(Y) - \min(Y))$  to run a quasi-binomial
# regression, and then estimations are back-transformed on the original scale.

```

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).

$$\mathbb{E}(Y \mid L(0), A, L(1), M) = \gamma_0 + \gamma_A A + \gamma_M M + \psi_{A*M}(A*M) + \gamma_{L(0)} L(0) + \gamma_{L(1)} L(1) \quad (6.6)$$

2. Remove the effect of mediator on the outcome, by evaluating the residual outcome:

$$Y_{res} = Y - \hat{\gamma}_M M - \hat{\psi}_{A*M} \times A \times M \quad (6.7)$$

and regress the residual outcome on the exposure A and baseline confounders $L(0)$:

$$\mathbb{E}(Y_{res} \mid A, L(0)) = \alpha_0 + \psi_A A + \beta_{L(0)} L(0) \quad (6.8)$$

The controlled direct effect CDE_m can then be estimated by:

$$\hat{\Psi}_{seq.g.est}^{CDE_m} = \hat{\psi}_A + \hat{\psi}_{A*M} \times m \quad (6.9)$$

```
rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")

## 1) Regress the outcome on past
Y.qol.model <- glm(Y_qol ~ L0_male + L0_soc_env + A0_PM2.5 + L1 +
  M_diabetes + A0_PM2.5:M_diabetes,
  family = "gaussian", data = df2_int)

## 2) Calculate a residual outcome Y - (coef.M * M_diabetes) - (coef.A0:M * A0:M)
Y.res <- (df2_int$Y_qol -
  (Y.qol.model$coefficients["M_diabetes"] * df2_int$M_diabetes) -
  (Y.qol.model$coefficients["A0_PM2.5:M_diabetes"] * df2_int$A0_PM2.5 *
  df2_int$M_diabetes) )

## 3) Regress the residual outcome on the exposure A and baseline confounders L(0)
Y.res.model <- glm(Y.res ~ L0_male + L0_soc_env + A0_PM2.5,
  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$coefficients["A0_PM2.5"] +
  0 * Y.qol.model$coefficients["A0_PM2.5:M_diabetes"])
CDE.qol.m0.seq
```

```
# -4.869509

CDE.qol.m1.seq <- (Y.res.model$coefficients["A0_PM2.5"] +
                  1 * Y.qol.model$coefficients["A0_PM2.5:M_diabetes"])
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), estimations are based on traditional regression models as described in chapter 5.

The g-formulas for the PNDE and TNIE are:

$$\begin{aligned}\Psi^{\text{PNDE}} &= \sum_{l(0), l(1)} \sum_m [\mathbb{E}(Y \mid A = 1, m, l(0), l(1)) - (Y \mid A = 0, m, l(0), l(1))] \\ &\quad \times \mathbb{P}(M = m \mid A = 0, l(0), l(1)) \times \mathbb{P}(l(0), l(1)) \\ \Psi^{\text{TNIE}} &= \sum_{l(0), l(1)} \sum_m \mathbb{E}(Y \mid A = 1, m, l(0), l(1)) \\ &\quad \times [\mathbb{P}(M = m \mid A = 1, l(0), l(1)) - \mathbb{P}(M = m \mid A = 0, l(0), l(1))] \\ &\quad \times \mathbb{P}(l(0), l(1))\end{aligned}$$

6.3.1 Simple “plug-in” estimator

```
## For the example, we will use the df1_int.csv data set (with an A*M interaction
## effect on the outcome, but no intermediate confounder affected by the exposure)
rm(list=ls())
df1_int <- read.csv(file = "./data/df1_int.csv")

## 1) Regress a model of the mediator and models of the outcomes (for binary and
##      continuous outcomes)
# Estimate a model of the mediator (logistic regression)
trad_m <- glm(M_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env + L1,
              family = "binomial",
              data = df1_int)
```

6.3. ESTIMATION OF NATURAL DIRECT (NDE) AND INDIRECT EFFECTS (NIE) 69

```
# Estimate models of the outcome (for continuous and binary outcomes)
trad_qol_am <- lm(Y_qol ~ A0_PM2.5 + M_diabetes + A0_PM2.5:M_diabetes +
  L0_male + L0_soc_env + L1,
  data = df1_int)
trad_death_am <- glm(Y_death ~ A0_PM2.5 + M_diabetes + A0_PM2.5:M_diabetes +
  L0_male + L0_soc_env + L1,
  family = "binomial",
  data = df1_int)

## 2) Generate predicted values for every combination of A={0,1} and M={0,1}
## 2.a) Predict counterfactual probabilities of the mediator
data.Ais0 <- data.Ais1 <- df1_int
data.Ais0$A0_PM2.5 <- 0
data.Ais1$A0_PM2.5 <- 1

# Predict the counterfactual probabilities
#  $P(M_{\{A=0\}} | l(0), l(1))$  and  $P(M_{\{A=1\}} | l(0), l(1))$ 
P.M.Ais0 <- predict(trad_m, newdata = data.Ais0, type = "response")
P.M.Ais1 <- predict(trad_m, newdata = data.Ais1, type = "response")

## 2.b) Predict counterfactual expected values of the outcomes
data.Ais0.Mis0 <- data.Ais0.Mis1 <- df1_int
data.Ais1.Mis0 <- data.Ais1.Mis1 <- df1_int

data.Ais0.Mis0$A0_PM2.5 <- 0
data.Ais0.Mis0$M_diabetes <- 0

data.Ais0.Mis1$A0_PM2.5 <- 0
data.Ais0.Mis1$M_diabetes <- 1

data.Ais1.Mis0$A0_PM2.5 <- 1
data.Ais1.Mis0$M_diabetes <- 0

data.Ais1.Mis1$A0_PM2.5 <- 1
data.Ais1.Mis1$M_diabetes <- 1

# Predict  $E(Y_{\{am\}} | l(0), l(1))$ 
Q.death.AOM0 <- predict(trad_death_am, newdata = data.Ais0.Mis0, type="response")
Q.death.AOM1 <- predict(trad_death_am, newdata = data.Ais0.Mis1, type="response")
Q.death.A1M0 <- predict(trad_death_am, newdata = data.Ais1.Mis0, type="response")
Q.death.A1M1 <- predict(trad_death_am, newdata = data.Ais1.Mis1, type="response")

Q.qol.AOM0 <- predict(trad_qol_am, newdata = data.Ais0.Mis0, type="response")
Q.qol.AOM1 <- predict(trad_qol_am, newdata = data.Ais0.Mis1, type="response")
Q.qol.A1M0 <- predict(trad_qol_am, newdata = data.Ais1.Mis0, type="response")
```

```

Q.qol.A1M1 <- predict(trad_qol_am, newdata = data.Ais1.Mis1, type="response")

## 3) Plug-in the predicted values in the g-formulas
## and estimate the population means
PNDE.death.gcomp <- mean((Q.death.A1M0 - Q.death.A0M0) * (1 - P.M.Ais0) +
                        (Q.death.A1M1 - Q.death.A0M1) * P.M.Ais0)
# [1] 0.0638596
TNIE.death.gcomp <- mean(Q.death.A1M0 * ((1 - P.M.Ais1) - (1 - P.M.Ais0)) +
                        Q.death.A1M1 * (P.M.Ais1 - P.M.Ais0) )
# [1] 0.01044539

PNDE.qol.gcomp <- mean((Q.qol.A1M0 - Q.qol.A0M0) * (1 - P.M.Ais0) +
                        (Q.qol.A1M1 - Q.qol.A0M1) * P.M.Ais0)
# [1] -5.310172
TNIE.qol.gcomp <- mean(Q.qol.A1M0 * ((1 - P.M.Ais1) - (1 - P.M.Ais0)) +
                        Q.qol.A1M1 * (P.M.Ais1 - P.M.Ais0) )
# [1] -1.774295

```

6.3.2 Using the CMAverse R package

```

library(CMAverse)
set.seed(1234)
res_rb_param_delta <- cmest(data = df1_int,
                             model = "rb", # for "regression based" (rb) approach
                             outcome = "Y_qol", # outcome variable
                             exposure = "AO_PM2.5", # exposure variable
                             mediator = "M_diabetes", # mediator
                             basec = c("LO_male", # confounders
                                         "LO_soc_env",
                                         "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),
                             estimation = "imputation", # closed-form parameter
                             # function estimation
                             inference = "bootstrap") # IC95% : "delta" or "bootstrap"
summary(res_rb_param_delta)
#           Estimate Std.error 95% CIL 95% CIU P.val
# cde          -3.71527  0.42805 -4.54643 -2.944 <2e-16 ***

```

6.3. ESTIMATION OF NATURAL DIRECT (NDE) AND INDIRECT EFFECTS (NIE) 71

```
#      pnde      -5.10339    0.34582 -6.02478   -4.719 <2e-16 *** vs -5.310172
#      tnde      -5.65988    0.35217 -6.76655   -5.342 <2e-16 ***
#      pnle      -0.85540    0.14386 -1.35738   -0.827 <2e-16 ***
#      tnle      -1.41189    0.23667 -2.19506   -1.370 <2e-16 *** vs -1.774295
#      te       -6.51528    0.41465 -8.05718   -6.394 <2e-16 ***
#      intref    -1.38812    0.18813 -1.92391   -1.226 <2e-16 ***
#      intmed    -0.55649    0.11655 -0.91444   -0.473 <2e-16 ***
#      cde(prop)  0.57024    0.04454  0.43771    0.608 <2e-16 ***
#      intref(prop) 0.21306    0.03102  0.16866    0.284 <2e-16 ***
#      intmed(prop) 0.08541    0.01511  0.06924    0.131 <2e-16 ***
#      pnle(prop)  0.13129    0.01686  0.12098    0.186 <2e-16 ***
#      pm        0.21670    0.02821  0.19441    0.305 <2e-16 ***
#      int       0.29847    0.04113  0.24601    0.400 <2e-16 ***
#      pe        0.42976    0.04454  0.39163    0.562 <2e-16 ***

set.seed(1234)
res_rb_param_delta <- ctest(data = df1_int,
                             model = "rb", # for "regression based" (rb) approach
                             outcome = "Y_death", # outcome variable
                             exposure = "AO_PM2.5", # exposure variable
                             mediator = "M_diabetes", # mediator
                             basec = c("LO_male", # confounders
                                       "LO_soc_env",
                                       "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),
                             estimation = "imputation", # closed-form parameter
                             # function estimation
                             inference = "bootstrap") # IC95% : "delta" or "bootstrap"

summary(res_rb_param_delta)
# Estimate Std.error 95% CIL 95% CIU P.val
#      cde      0.060001 0.016971 0.029741 0.096 <2e-16 ***
#      pnle      0.064743 0.013036 0.040779 0.094 <2e-16 *** vs 0.0638596
#      tnle      0.066644 0.012808 0.041684 0.095 <2e-16 ***
#      pnle      0.006696 0.001666 0.005635 0.012 <2e-16 ***
#      tnle      0.008597 0.003576 0.003856 0.018 0.02 * vs 0.01044539
#      te       0.073340 0.012905 0.049302 0.102 <2e-16 ***
#      intref    0.004742 0.008584 -0.011243 0.021 0.51
#      intmed    0.001901 0.003764 -0.005370 0.009 0.51
#      cde(prop) 0.818122 0.160328 0.501113 1.108 <2e-16 ***
```

```
#   intref(prop)  0.064653  0.115196 -0.151420  0.270  0.51
#   intmed(prop)  0.025919  0.050149 -0.069991  0.121  0.51
#   pnle(prop)   0.091306  0.029324  0.067935  0.178 <2e-16 ***
#   pm           0.117225  0.052331  0.053291  0.245  0.02 *
#   int          0.090572  0.164830 -0.223164  0.400  0.51
#   pe           0.181878  0.160328 -0.108220  0.499  0.15
```

6.3.3 Using the mediation R package

The `mediation` R package can be used to estimate Natural Direct and Indirect Effects for causal models where the intermediate confounders are not affected by the exposure (as in Figure 3.1).

The estimation relies on the model of the mediator and the model of the outcome. Then a quasi-bayesian Monte Carlo method is applied, simulating counterfactual distributions of the mediator. The approach is rather similar to the parametric g-computation to estimate “Marginal” Randomized/Interventional Direct and Indirect Effects, described in the next paragraph.

Note that the `mediation` package can also be used for multilevel data.

```
## Using the "mediation" package : R Package for Causal Mediation Analysis
library(mediation)
?mediation::mediate

## We will use the previous model of the mediator (logistic regression)
trad_m
# Call:  glm(formula = M_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env +
#           L1, family = "binomial", data = df1_int)
# Coefficients:
#   (Intercept)      A0_PM2.5      L0_male  L0_soc_env          L1
#      -1.3788         0.5626         0.2586         0.3305         0.3346

## We will use the previous models of the outcomes (continuous and binary outcomes)
trad_qol_am
# Call:
#   lm(formula = Y_qol ~ A0_PM2.5 + M_diabetes + A0_PM2.5:M_diabetes +
#       L0_male + L0_soc_env + L1, data = df1_int)
# Coefficients:
#   (Intercept)  A0_PM2.5  M_diabetes  L0_male  L0_soc_env      L1  A0_PM2.5:M_diabetes
#      74.7669   -3.7153    -8.6317   -0.7235   -2.8899  -3.4280    -5.6154

trad_death_am
# Call:  glm(formula = Y_death ~ A0_PM2.5 + M_diabetes + A0_PM2.5:M_diabetes +
#           L0_male + L0_soc_env + L1, family = "binomial", data = df1_int)
```


6.3. ESTIMATION OF NATURAL DIRECT (NDE) AND INDIRECT EFFECTS (NIE)73

```
# Coefficients:
# (Intercept) A0_PM2.5 M_diabetes L0_male L0_soc_env L1 A0_PM2.5:M_diabetes
# -2.06294 0.36668 0.40921 0.29249 0.36360 0.44716 0.01275

## Use the mediate() function to estimate the TNDE, PNIE
## Estimations relies on quasi-bayesian Monte Carlo method (especially for continuous
## mediators) => set seed for reproducibility
set.seed(2024)

## For the quantitative outcome
mediation.res.qol <- mediate(trad_m, # model of the mediator
                           trad_qol_am, # model of the outcome
                           treat = "A0_PM2.5", # exposure
                           mediator = "M_diabetes", # mediator
                           robustSE = TRUE, # estimate sandwich SEs
                           sims = 100) # better to use >= 1000

summary(mediation.res.qol)
#
```

	Estimate	95% CI Lower	95% CI Upper	p-value
ACME (control)	-1.074	-1.356	-0.82	<2e-16 *** PNIE
ACME (treated)	-1.778	-2.239	-1.38	<2e-16 *** TNIE
ADE (control)	-5.301	-5.950	-4.63	<2e-16 *** PNDE
ADE (treated)	-6.005	-6.646	-5.27	<2e-16 *** TNDE
Total Effect	-7.079	-7.768	-6.24	<2e-16 ***
Prop. Mediated (control)	0.151	0.118	0.19	<2e-16 ***
Prop. Mediated (treated)	0.251	0.197	0.31	<2e-16 ***
ACME (average)	-1.426	-1.811	-1.11	<2e-16 ***
ADE (average)	-5.653	-6.267	-4.95	<2e-16 ***
Prop. Mediated (average)	0.201	0.160	0.25	<2e-16 ***

In the output,

- ACME (control) (Average Causal Mediated Effect) corresponds to the Pure Natural Indirect Effect,
- ACME (treated) corresponds to the Total Natural Indirect Effect,
- ADE (control) (Average Direct Effect) corresponds to the Pure Natural Direct Effect,
- ADE (treated) corresponds to the Total Natural Direct Effect,

So the sum of ACME (control) + ADE (treated) is equal to the total effect. Similarly, the sum of ACME (treated) + ADE (control) is equal to the total effect.

We can plot the results. For example the results for the decomposition of the effect of exposure to high-levels of PM_{2.5} on quality of life, through type-2 diabetes is described in the Figure

```
## Plot the estimations
plot(mediation.res.qol)
```

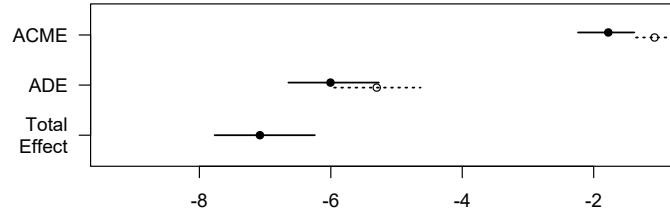


Figure 6.1: Plot of the 2-way decomposition (mediation package)

For the binary outcome:

```
mediation.res.death <- mediate(trad_m,                # model of the mediator
                              trad_death_am,         # model of the outcome)
                              treat = "AO_PM2.5",    # exposure
                              mediator = "M_diabetes", # mediator
                              robustSE = TRUE,        # estimate sandwich SEs
                              sims = 100)            # better to use >= 1000

summary(mediation.res.death)
```

#	Estimate	95% CI Lower	95% CI Upper	p-value
# ACME (control)	0.00860	0.00569	0.01	<2e-16 *** PNIE
# ACME (treated)	0.00979	0.00377	0.02	<2e-16 *** TNIE
# ADE (control)	0.06543	0.03658	0.09	<2e-16 *** PNDE
# ADE (treated)	0.06662	0.03946	0.09	<2e-16 *** TNDE
# Total Effect	0.07522	0.04800	0.10	<2e-16 ***
# Prop. Mediated (control)	0.11172	0.07323	0.18	<2e-16 ***
# Prop. Mediated (treated)	0.13066	0.05714	0.22	<2e-16 ***
# ACME (average)	0.00920	0.00567	0.01	<2e-16 ***
# ADE (average)	0.06603	0.03803	0.09	<2e-16 ***
# Prop. Mediated (average)	0.12119	0.07330	0.20	<2e-16 ***

Interestingly, the `mediation` package enable to carry-out a sensitivity analysis to test if unmeasured confounders of the mediator-outcome relationship could offset the estimated direct or indirect effects. This sensitivity analysis relies on assumptions on the mediator-outcome correlation coefficient. I.e., it can be used to test for unmeasured variables within the $L(1)$ set in the causal structure of Figure 3.1.

6.3. ESTIMATION OF NATURAL DIRECT (NDE) AND INDIRECT EFFECTS (NIE) 75

If there exist unobserved (baseline) confounders of the M-Y relationship, we expect that ρ is no longer zero. The sensitivity analysis is conducted by varying the value of ρ and examining how the estimated ACME and ADE change.

Note 1: this sensitivity analysis cannot be used to test the assumption that no intermediate confounder $L(1)$ of the mediator-outcome relationship is affected by the exposure (like in Figure 3.2).

Note 2: Sensitivity analysis from the `mediation` package cannot be applied when the mediator and the outcome are both binary. We will only use it for the “Quality of Life” outcome.

```
## Sensitivity analysis to test sequential ignorability (assess the possible
## existence of unobserved (baseline) confounders of the M-Y relationship)

## Estimate a probit model of the mediator
trad_m <- glm(M_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env + L1,
              family = binomial("probit"), # sensitivity analysis works only for
                                           # probit models if M is binary
              data = df1_int)
mediation.res.qol <- mediate(trad_m, # model of the mediator
                           trad_qol_am, # model of the outcome)
                           treat = "A0_PM2.5", # exposure
                           mediator = "M_diabetes", # mediator
                           robustSE = TRUE, # estimate sandwich SEs
                           sims = 100) # better to use >= 1000

summary(mediation.res.qol)
#
```

	Estimate	95% CI Lower	95% CI Upper	p-value
# ACME (control)	-1.071	-1.416	-0.79	<2e-16 ***
# ACME (treated)	-1.784	-2.333	-1.31	<2e-16 ***
# ADE (control)	-5.266	-5.809	-4.56	<2e-16 ***
# ADE (treated)	-5.979	-6.515	-5.31	<2e-16 ***
# Total Effect	-7.050	-7.754	-6.34	<2e-16 ***
# Prop. Mediated (control)	0.151	0.119	0.19	<2e-16 ***
# Prop. Mediated (treated)	0.254	0.194	0.32	<2e-16 ***
# ACME (average)	-1.427	-1.889	-1.06	<2e-16 ***
# ADE (average)	-5.623	-6.152	-4.97	<2e-16 ***
# Prop. Mediated (average)	0.203	0.159	0.26	<2e-16 ***

```
sensitiviv <- medsens(mediation.res.qol,
                      rho.by = 0.1, # sensitivity parameter = correlation between
                      # the residuals of the mediator & outcome regressions
                      # here, rho varies from -0.9 to +0.9 by 0.1 increments
                      effect.type = "both", # "direct", "indirect" or "both"
                      sims = 100) # better to use >= 1000

summary(sensitiviv)
# Mediation Sensitivity Analysis: Average Mediation Effect
```

```

# Sensitivity Region: ACME for Control Group
#      Rho ACME(control) 95% CI Lower 95% CI Upper R^2_M*R^2_Y* R^2_M~R^2_Y~
#      [1,] -0.6          0.0474      -0.0093      0.1229          0.36          0.2647
#
# Rho at which ACME for Control Group = 0: -0.6
# R^2_M*R^2_Y* at which ACME for Control Group = 0: 0.36
# R^2_M~R^2_Y~ at which ACME for Control Group = 0: 0.2647
#
# Rho at which ACME for Treatment Group = 0: -0.9
# R^2_M*R^2_Y* at which ACME for Treatment Group = 0: 0.81
# R^2_M~R^2_Y~ at which ACME for Treatment Group = 0: 0.5956
#
# Mediation Sensitivity Analysis: Average Direct Effect
# Rho at which ADE for Control Group = 0: 0.8
# R^2_M*R^2_Y* at which ADE for Control Group = 0: 0.64
# R^2_M~R^2_Y~ at which ADE for Control Group = 0: 0.4706
#
# Rho at which ADE for Treatment Group = 0: 0.8
# R^2_M*R^2_Y* at which ADE for Treatment Group = 0: 0.64
# R^2_M~R^2_Y~ at which ADE for Treatment Group = 0: 0.4706

par(mfrow = c(2,2))
plot(sensisitivity)
par(mfrow = c(1,1))

```

Here, for the Pure Natural Indirect Effect:

- the confidence interval of the ACME (PNIE) contains zero when $\rho = -0.6$;
- when the product of the residual variance explained by the omitted confounding is 0.36, the point estimate of PNIE = 0;
- when the product of the total variance explained by the omitted confounding is 0.27, the point estimate of PNIE = 0;

For the TNDE and PNDE,

- the confidence interval of the ADE (TNDE and PNDE) contain zero when $\rho > +0.8$;
- when the product of the residual variance explained by the omitted confounding is 0.64, the point estimate of TNDE = 0;
- when the product of the total variance explained by the omitted confounding is 0.47, the point estimate of TNDE = 0;

We can conclude that the possibility of the PNIE and the TNIE cancelling out because of unmeasured M-Y confounding is unlikely in our example (it requires rather strong correlations).

6.4. ESTIMATION OF “MARGINAL” RANDOMIZED/INTERVENTIONAL NATURAL DIRECT (MRDE) AND

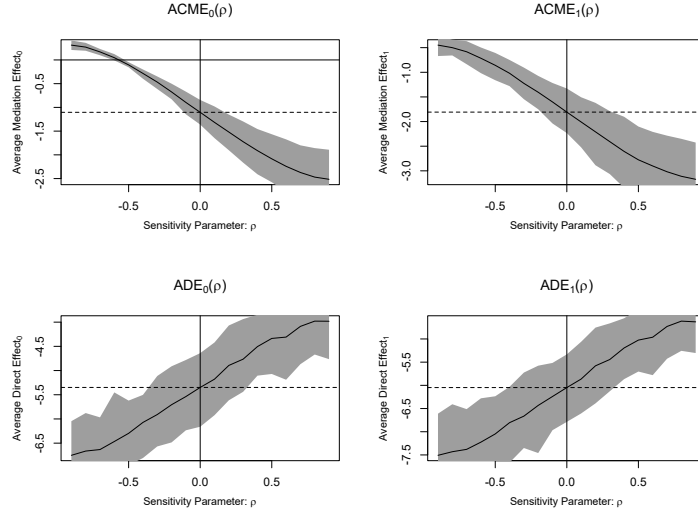


Figure 6.2: Sensitivity analysis testing unmeasured confounding of the M-Y relationship

6.4 Estimation of “Marginal” Randomized/Interventional Natural Direct (MRDE) and Indirect Effects (MRIE)

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. (VanderWeele and Tchetgen Tchetgen 2017) (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 confounders $L(0)$) under the counterfactual scenario setting $A = a'$.

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

This Overall Effect can be decomposed into the sum of:

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

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

Under the identifiability conditions, in particular:

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

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

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

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.

```
rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")
set.seed(54321)
```

6.4. ESTIMATION OF “MARGINAL” RANDOMIZED/INTERVENTIONAL NATURAL DIRECT (MRDE) AND

```

# 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_soc_env + A0_PM2.5,
    family = "binomial", data = df2_int)
  M.model <- glm(M_diabetes ~ L0_male + L0_soc_env + A0_PM2.5 + 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_soc_env + A0_PM2.5 + L1 +
    M_diabetes + A0_PM2.5:M_diabetes,
    family = "binomial", data = df2_int)
  Y.qol.model <- glm(Y_qol ~ L0_male + L0_soc_env + A0_PM2.5 + L1 +
    M_diabetes + A0_PM2.5:M_diabetes,
    family = "gaussian", data = df2_int)

  ## 2) Estimate the joint distribution of time-varying confounders and of the
  ##    mediator under the counterfactual scenarios setting A0_PM2.5 = 1 or 0

  # set the exposure A0_PM2.5 to 0 or 1 in two new counterfactual data sets
  data.A0 <- data.A1 <- df2_int
  data.A0$A0_PM2.5 <- 0
  data.A1$A0_PM2.5 <- 1
  # simulate L1 values under the counterfactual exposures A0_PM2.5=0 or A0_PM2.5=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_PM2.5=0 or A0_PM2.5=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$M_diabetes <- marg.M.A0
# data.A0.G1$M_diabetes <- marg.M.A1 # note: this data set will not be useful

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

```


6.4. ESTIMATION OF “MARGINAL” RANDOMIZED/INTERVENTIONAL NATURAL DIRECT (MRDE) AND

```
# [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 dimensionality 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 $G_{A=0|L(0)}$:

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

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

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

```
rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")

## 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_diabetes ~ L0_male + L0_soc_env + A0_PM2.5,
               family = "binomial", data = df2_int)

### 1b) generate predicted probabilities 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$A0_PM2.5 <- 0
data.Ais1$A0_PM2.5 <- 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")
```

6.4. ESTIMATION OF “MARGINAL” RANDOMIZED/INTERVENTIONAL NATURAL DIRECT (MRDE) AND

```
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_soc_env + A0_PM2.5 + L1 +
                     M_diabetes + A0_PM2.5:M_diabetes,
                     family = "binomial", data = df2_int)
Y.qol.model <- glm(Y_qol ~ L0_male + L0_soc_env + A0_PM2.5 + L1 +
                  M_diabetes + A0_PM2.5:M_diabetes,
                  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$M_diabetes <- 0
data.Mis1$M_diabetes <- 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  $\bar{Q}.L2_{\{A=0, G_{\{A=0/L(0)\}}\}}$ 
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  $\bar{Q}.L2_{\{A=1, G_{\{A=0/L(0)\}}\}}$ 
# note at this step, quantities are similar to  $\bar{Q}.L2_{\{A=0, G_{\{A=0/L(0)\}}\}}$ 
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  $\bar{Q}.L2_{\{A=1, G_{\{A=1/L(0)\}}\}}$ 
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)

## 3) Fit parametric models for the predicted values  $\bar{Q}.L2$  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_soc_env + A0_PM2.5,
                           family = "quasibinomial", data = df2_int)
L1.death.A1.G0.model <- glm(Q.L2.death.A1.G0 ~ L0_male + L0_soc_env + A0_PM2.5,
                           family = "quasibinomial", data = df2_int)
L1.death.A1.G1.model <- glm(Q.L2.death.A1.G1 ~ L0_male + L0_soc_env + A0_PM2.5,
                           family = "quasibinomial", data = df2_int)

L1.qol.A0.G0.model <- glm(Q.L2.qol.A0.G0 ~ L0_male + L0_soc_env + A0_PM2.5,
                         family = "gaussian", data = df2_int)
L1.qol.A1.G0.model <- glm(Q.L2.qol.A1.G0 ~ L0_male + L0_soc_env + A0_PM2.5,
                         family = "gaussian", data = df2_int)
L1.qol.A1.G1.model <- glm(Q.L2.qol.A1.G1 ~ L0_male + L0_soc_env + A0_PM2.5,
                         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 $\bar{Q}_Y = \mathbb{E}(Y \mid L(0), A, L(1), M)$,
- fit of the Mediator regression $g_A = P(M = 1 \mid L(0), A, L(1))$,
- fit of the intermediate confounder regression $\bar{Q}_{L(1)} = P(L(1) = 1 \mid L(0), A)$,
- the 2-way, 3-way and 4-way decompositions.

```
library(CMAverse)
rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")
```

```

set.seed(1234)
res_gformula_Qol_M0 <- cmest(data = df2_int,
                             model = "gformula", # for parametric g-computation
                             outcome = "Y_qol", # outcome variable
                             exposure = "A0_PM2.5", # exposure variable
                             mediator = "M_diabetes", # mediator
                             basec = c("L0_male", # confounders
                                         "L0_soc_env"),
                             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", # for percentile, other option: "bca"
                             nboot = 2) # we should use a large number of bootstrap sample.

summary(res_gformula_Qol_M0)

### 1) Estimation of Qbar.Y = P(Y=1/M,L1,A,L0) with A*M interaction,
### Outcome regression:
# Call:
# glm(formula = Y_qol ~ A0_PM2.5 + M_diabetes + A0_PM2.5 * M_diabetes +
#       L0_male + L0_soc_env + L1, family = gaussian(),
#       data = getCall(x$reg.output$yreg)$data, weights = getCall(x$reg.output$yreg)$w
# Coefficients:
#               Estimate Std. Error t value Pr(>|t|)
# (Intercept)      74.8247    0.2133 350.823 < 2e-16 ***
# A0_PM2.5         -3.7014    0.4295  -8.617 < 2e-16 ***
# M_diabetes       -8.6336    0.2331 -37.042 < 2e-16 ***
# L0_male          -0.7280    0.2019  -3.605 0.000313 ***
# L0_soc_env       -2.8828    0.2116 -13.621 < 2e-16 ***
# L1               -5.1668    0.2189 -23.608 < 2e-16 ***
# A0_PM2.5:M_diabetes -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_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env +
#       L1, family = binomial(), data = getCall(x$reg.output$mreg[[1L]])$data,
#       weights = getCall(x$reg.output$mreg[[1L]])$weights)
# Coefficients:

```

6.5. USING THE CMAVERSE PACKAGE FOR 2-WAY, 3-WAY AND 4-WAY DECOMPOSITION⁸⁷

```
#
#           Estimate Std. Error z value Pr(>|z|)
# (Intercept)      -1.36249    0.04783 -28.488 < 2e-16 ***
# AO_PM2.5         0.30994    0.06668   4.648 3.35e-06 ***
# LO_male          0.24661    0.04369   5.644 1.66e-08 ***
# LO_soc_env       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,LO), model of intermediate confounder
### Regressions for mediator-outcome confounders affected by the exposure:
# Call:
# glm(formula = L1 ~ AO_PM2.5 + LO_male + LO_soc_env,
#      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 ***
# AO_PM2.5         0.94354    0.06475  14.572 < 2e-16 ***
# LO_male          -0.19827    0.04289  -4.622 3.80e-06 ***
# LO_soc_env       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

```

Using the **CMAverse** package,

- the total effect (overall effect) on the QoL quantitative outcome is estimated to be $TE = \mathbb{E}(Y_{A=1, G_{A=1|L(0)}}) - \mathbb{E}(Y_{A=0, G_{A=0|L(0)}}) \approx -9.87$,
- the controlled direct effect ($CDE_{M=0}$), setting $M = 0$ is $CDE_{M=0} = \mathbb{E}(Y_{A=1, M=0}) - \mathbb{E}(Y_{A=0, M=0}) \approx -5.86$,
- the randomized pure natural direct effect is $rPNDE = \mathbb{E}(Y_{A=1, G_{A=0|L(0)}}) - \mathbb{E}(Y_{A=0, G_{A=0|L(0)}}) \approx -7.57$
- the randomized total natural indirect effect is $rTNIE = \mathbb{E}(Y_{A=1, G_{A=1|L(0)}}) - \mathbb{E}(Y_{A=1, G_{A=0|L(0)}}) \approx -2.30$

(VanderWeele 2014) defined the 3-way or 4-way decomposition in causal structures with no intermediate confounder $L(1)$ of the $M - Y$ relationship affected by the exposure A . In causal structures with such intermediate confounders (Causal model 2, Figure 3.2), analogues of the 3-way and 4-way decomposition can still be defined. Those analogues are estimated by the **CMAverse** package. They can be calculated from the controlled direct effect (setting $M = 0$) and the pure Natural Direct and Indirect Effects, using the following relationships:

- We can start by estimating the Total Effect (Overall effect), the CDE (setting $M = 0$), the randomized Pure Natural Direct effect and the randomized Pure Natural Indirect Effect:

$$\begin{aligned}
TE &= \mathbb{E}(Y_{A=1, G_{A=1|L(0)}}) - \mathbb{E}(Y_{A=0, G_{A=0|L(0)}}) \\
CDE_{M=0} &= \mathbb{E}(Y_{A=1, M=0}) - \mathbb{E}(Y_{A=0, M=0}) \\
rPNDE &= \mathbb{E}(Y_{A=1, G_{A=0|L(0)}}) - \mathbb{E}(Y_{A=0, G_{A=0|L(0)}}) \\
rPNIE &= \mathbb{E}(Y_{A=0, G_{A=1|L(0)}}) - \mathbb{E}(Y_{A=0, G_{A=0|L(0)}})
\end{aligned}$$

- the Mediated Interaction can then be obtained by :

$$\begin{aligned}
 MIE &= rTNDE - rPNDE = rTNIE - rPNIE \\
 MIE &= \left[\mathbb{E}(Y_{A=1, G_{A=1|L(0)}}) - \mathbb{E}(Y_{A=0, G_{A=1|L(0)}}) \right] \\
 &\quad - \left[\mathbb{E}(Y_{A=1, G_{A=0|L(0)}}) - \mathbb{E}(Y_{A=0, G_{A=0|L(0)}}) \right] \\
 &= \left[\mathbb{E}(Y_{A=1, G_{A=1|L(0)}}) - \mathbb{E}(Y_{A=1, G_{A=0|L(0)}}) \right] \\
 &\quad - \left[\mathbb{E}(Y_{A=0, G_{A=1|L(0)}}) - \mathbb{E}(Y_{A=0, G_{A=0|L(0)}}) \right]
 \end{aligned}$$

- and the Reference Interaction can be obtained by :

$$\begin{aligned}
 RIE &= rPNDE - CDE_{M=0} \\
 RIE &= \left[\mathbb{E}(Y_{A=1, G_{A=0|L(0)}}) - \mathbb{E}(Y_{A=0, G_{A=0|L(0)}}) \right] - \left[\mathbb{E}(Y_{A=1, M=0}) - \mathbb{E}(Y_{A=0, M=0}) \right]
 \end{aligned}$$

```
## Using the previous results, we can check those equalities for the analogues
## of the 3-way and 4-way decomposition
res_gformula_Qol_M0$effect.pe["te"]
# TE = -9.870139
res_gformula_Qol_M0$effect.pe["cde"]
# CDE(M=0) = -5.86375
res_gformula_Qol_M0$effect.pe["rpnde"]
# rPNDE = -7.565835
res_gformula_Qol_M0$effect.pe["rpnie"]
# rPNIE = -1.40641

## Check that MI = rTNIE - rPNIE = rTNDE - rPNDE
(res_gformula_Qol_M0$effect.pe["rtnie"] - res_gformula_Qol_M0$effect.pe["rpnie"])
# -0.897894
(res_gformula_Qol_M0$effect.pe["rtnde"] - res_gformula_Qol_M0$effect.pe["rpnde"])
# -0.897894
res_gformula_Qol_M0$effect.pe["rintmed"]
# -0.897894 # we have MI = rTNIE - rPNIE = rTNDE - rPNDE

## Check that RE = PNDE - CDE_{M=0}
res_gformula_Qol_M0$effect.pe["rpnde"] - res_gformula_Qol_M0$effect.pe["cde"]
# -1.702085
res_gformula_Qol_M0$effect.pe["rintref"]
# -1.702085 # we have RE = PNDE - CDE_{M=0}
```

With a binary outcome, we can use the same `cmest` function as previously, replacing the `yreg = "linear"` argument by `yreg = "logistic"`. The results will be given on the Odds Ratio scale:

```

set.seed(1234)
res_gformula_OR_MO <- cmest(data = df2_int,
                             model = "gformula",
                             outcome = "Y_death",
                             exposure = "A0_PM2.5",
                             mediator = "M_diabetes",
                             basec = c("L0_male", "L0_soc_env"),
                             postc = "L1",
                             EMint = TRUE,
                             mreg = list("logistic"), # g(M=1/L1,A,L0)
                             yreg = "logistic", # 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", # parametric g-comp if model= gformul
                             inference = "bootstrap",
                             boot.ci.type = "per", # forpercentile, other option: "bca"
                             nboot = 2) # we should use a large number of bootstrap samples

summary(res_gformula_OR_MO)

### 1) Estimation of Qbar.Y = P(Y=1/M,L1,A,L0) with A*M interaction,
###   by logistic regression
# Outcome regression:
# Call:
# glm(formula = Y_death ~ A0_PM2.5 + M_diabetes + A0_PM2.5 * M_diabetes +
#      L0_male + L0_soc_env + L1, family = binomial(),
#      data = getCall(x$reg.output$yreg)$data, weights = getCall(x$reg.output$yreg)$wei
#
# Coefficients:
#
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept)    -2.04033    0.05855  -34.849  < 2e-16 ***
# A0_PM2.5         0.28922    0.10123   2.857  0.00428 **
# M_diabetes       0.44406    0.05569   7.974  1.53e-15 ***
# L0_male          0.26913    0.04996   5.387  7.15e-08 ***
# L0_soc_env       0.34603    0.05432   6.370  1.89e-10 ***
# L1               0.42894    0.05195   8.257  < 2e-16 ***
# A0_PM2.5:M_diabetes 0.04387    0.14311   0.307  0.75919

# etc

### 4) Effect decomposition on the odds ratio scale via the g-formula approach
#
#               Estimate Std.error 95% CIL 95% CIU P.val
# Rcde          1.592115 0.121242 1.401748 1.565 <2e-16 ***
# rRpnde        1.610141 0.064880 1.458433 1.546 <2e-16 ***
# rRtnde        1.621371 0.046720 1.479227 1.542 <2e-16 ***

```

6.5. USING THE CMAVERSE PACKAGE FOR 2-WAY, 3-WAY AND 4-WAY DECOMPOSITION⁹¹

```
# rRpnie      1.076471  0.001714  1.036683   1.039 <2e-16 ***
# rRtnie      1.083979  0.014539  1.034267   1.054 <2e-16 ***
# Rte         1.745359  0.045898  1.536897   1.599 <2e-16 ***
# ERcde       0.392219  0.073776  0.277167   0.376 <2e-16 ***
# rERintref   0.217921  0.008895  0.169383   0.181 <2e-16 ***
# rERintmed   0.058747  0.017268  0.016243   0.039 <2e-16 ***
# rERpnie     0.076471  0.001714  0.036684   0.039 <2e-16 ***
# ERcde(prop) 0.526216  0.083694  0.515858   0.628 <2e-16 ***
# rERintref(prop) 0.292371  0.040769  0.283120   0.338 <2e-16 ***
# rERintmed(prop) 0.078817  0.034491  0.027264   0.074 <2e-16 ***
# rERpnie(prop) 0.102596  0.008434  0.061314   0.073 <2e-16 ***
# rpm         0.181413  0.042925  0.088578   0.146 <2e-16 ***
# rint        0.371188  0.075260  0.310384   0.411 <2e-16 ***
# rpe         0.473784  0.083694  0.371698   0.484 <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Rcde: controlled direct effect odds ratio;
# rRpnde: randomized analogue of pure natural direct effect odds ratio;
# rRtnde: randomized analogue of total natural direct effect odds ratio;
# rRpnie: randomized analogue of pure natural indirect effect odds ratio;
# rRtnie: randomized analogue of total natural indirect effect odds ratio;
# Rte: total effect odds ratio;
# ERcde: excess relative risk due to controlled direct effect;
# rERintref: randomized analogue of excess relative risk due to reference interaction;
# rERintmed: randomized analogue of excess relative risk due to mediated interaction;
# rERpnie: randomized analogue of excess relative risk due to pure natural indirect effect;
# ERcde(prop): proportion ERcde;
# rERintref(prop): proportion rERintref;
# rERintmed(prop): proportion rERintmed;
# rERpnie(prop): proportion rERpnie;
# rpm: randomized analogue of overall proportion mediated;
# rint: randomized analogue of overall proportion attributable to interaction;
# rpe: randomized analogue of overall proportion eliminated)
```

If we prefer to estimate results for the binary outcome on a risk difference scale, it is still possible, by applying a linear regression for the outcome model, using the argument `yreg = "linear"`.

```
set.seed(1234)
res_gformula_RD_MO <- cmest(data = df2_int,
                             model = "gformula",
                             outcome = "Y_death",
                             exposure = "A0_PM2.5",
                             mediator = "M_diabetes",
```

```

basec = c("L0_male", "L0_soc_env"),
postc = "L1",
EMint = TRUE,
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", # parametric g-comp if model= gformul
inference = "bootstrap",
boot.ci.type = "per", # forpercentile, other option: "bca"
nboot = 2) # we should use a large number of bootstrap samples
summary(res_gformula_RD_M0)
## 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
#
# Direct counterfactual imputation estimation with
# bootstrap standard errors, percentile confidence intervals and p-values
#
#           Estimate Std.error   95% CIL 95% CIU  P.val
# cde      0.0638914 0.0158641 0.0467659 0.068 <2e-16 ***
# rpnde    0.0734700 0.0307357 0.0598528 0.101 <2e-16 ***
# rtnde    0.0771555 0.0333264 0.0647151 0.109 <2e-16 ***
# rpnie    0.0090890 0.0033498 0.0050769 0.010 <2e-16 ***
# rtnie    0.0127744 0.0007591 0.0134198 0.014 <2e-16 ***
# te       0.0862445 0.0299766 0.0742924 0.115 <2e-16 ***
# rintref  0.0095786 0.0148716 0.0130869 0.033 <2e-16 ***
# rintmed  0.0036855 0.0025907 0.0048623 0.008 <2e-16 ***
# cde(prop) 0.7408176 0.0263715 0.5945733 0.630 <2e-16 ***
# rintref(prop) 0.1110634 0.0841501 0.1744983 0.288 <2e-16 ***
# rintmed(prop) 0.0427328 0.0055167 0.0653396 0.073 <2e-16 ***
# rpnie(prop) 0.1053862 0.0632953 0.0451212 0.130 <2e-16 ***
# rpm      0.1481190 0.0577786 0.1178726 0.195 <2e-16 ***
# rint     0.1537963 0.0896668 0.2398378 0.360 <2e-16 ***
# rpe      0.2591824 0.0263715 0.3699965 0.405 <2e-16 ***

```

Using the `CMAverse` package,

- the total effect (overall effect) on the probability of death (binary outcome) is estimated to be $TE = P(Y_{A=1, G_{A=1|L(0)}} = 1) - P(Y_{A=0, G_{A=0|L(0)}} = 1) \approx 0.086$,
- the controlled direct effect ($CDE_{M=0}$), setting $M = 0$ is $CDE_{M=0} = P(Y_{A=1, M=0} = 1) - P(Y_{A=0, M=0} = 1) \approx 0.064$,

6.6. ESTIMATION OF “CONDITIONAL” RANDOMIZED/INTERVENTIONAL NATURAL DIRECT (CRDE) AND

- the randomized pure natural direct effect is $rPNDE = P(Y_{A=1, G_{A=0|L(0)}} = 1) - P(Y_{A=0, G_{A=0|L(0)}} = 1) \approx 0.073$
- the randomized total natural indirect effect is $rTNIE = P(Y_{A=1, G_{A=1|L(0)}} = 1) - P(Y_{A=1, G_{A=0|L(0)}} = 1) \approx 0.013$

6.6 Estimation of “Conditional” Randomized/Interventional Natural Direct (CRDE) and Indirect Effects (CRIE)

Zheng and van der Laan (Zheng and van der Laan 2017) described “Conditional” Randomized Direct and Indirect Effects, using random draws from the counterfactual distribution of the mediator, conditional on both $L(0)$ and $L(1)$. The Average Total Effect (ATE) can be decomposed into the sum of :

- a Conditional Randomized Natural Direct Effect (CRDE): $CRDE = \mathbb{E}(Y_{1, \Gamma_0|L(0), L(1)}) - \mathbb{E}(Y_{0, \Gamma_0|L(0), L(1)})$
- and Conditional Randomized Natural Indirect Effect (CRIE) $CRIE = \mathbb{E}(Y_{1, \Gamma_1|L(0), L(1)}) - \mathbb{E}(Y_{1, \Gamma_0|L(0), L(1)})$

For this 2-way decomposition, we have to estimate 3 causal quantities: $\mathbb{E}(Y_{1, \Gamma_1|L(0), L(1)})$, $\mathbb{E}(Y_{1, \Gamma_0|L(0), L(1)})$ and $\mathbb{E}(Y_{0, \Gamma_0|L(0), L(1)})$.

Under the identifiability conditions, in particular:

- no unmeasured confounding between A and Y , given $L(0)$
- no unmeasured confounding between M and Y , given A , $L(0)$ and $L(1)$
- no unmeasured confounding between A and M , given $L(0)$
- and no unmeasured confounding between A and $L(1)$, given $L(0)$

the quantity of $\mathbb{E}(Y_{a, \Gamma_{a'}|L(0), L(1)})$ can be estimated by the g-formula:

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

These causal effects can be estimated by g-computation, IPTW, or TMLE. The G-computation approach is described below.

6.6.1 G-computation by iterative conditional expectation (simple substitution estimator)

We describe below the g-computation algorithm (“non targeted substitution estimator”) which is described in (Zheng and van der Laan 2017).

The following steps are applied to estimate the quantity $\Psi^{a,a'} = \mathbb{E} [Y_{a,\Gamma_{a'}} | L(0), L(1)]$:

1. Define $\bar{Q}_{R_2}^{a,a'} = Y$

2. Calculate $\bar{Q}_{L_1}^{a,a'}$, $\bar{Q}_{M_1}^{a,a'}$ and $\bar{Q}_{R_1}^{a,a'}$

(a) Regress $\bar{Q}_{R_2}^{a,a'}$ on the observed values of $\{M, L(1), A, L(0)\}$,

$$\mathbb{E} (\bar{Q}_{R_2}^{a,a'} | M, L(1), A, L(0)) = \mathbb{E} (Y | M, L(1), A, L(0))$$

Evaluate the fitted function at the observed mediator and covariates histories $\{L(1), M, L(0)\}$ and the intervened exposure $A = a$. This gives the estimates

$$\hat{\bar{Q}}_{L_1}^{a,a'}(L(1), M, L(0)) = \mathbb{E} (Y | A = a, L(1), M, L(0))$$

(b) Regress the predicted value $\hat{\bar{Q}}_{L_1}^{a,a'}(L(1), M, L(0))$ on the observed values of $\{L(1), A, L(0)\}$

$$\mathbb{E} \left(\hat{\bar{Q}}_{L_1}^{a,a'}(L(1), M, L(0)) | L(1), A, L(0) \right)$$

Then, evaluate the fitted function at the observed covariates histories $\{L(1), L(0)\}$ and the intervened exposure $A = a'$. This gives the estimates

$$\hat{\bar{Q}}_{M_1}^{a,a'}(L(1), L(0)) = \mathbb{E} \left(\hat{\bar{Q}}_{L_1}^{a,a'}(L(1), M, L(0)) | A = a', L(1), L(0) \right)$$

(c) Regress the predicted value $\hat{\bar{Q}}_{M_1}^{a,a'}(L(1), L(0))$ on the observed values of $\{A, L(0)\}$

$$\mathbb{E} \left(\hat{\bar{Q}}_{M_1}^{a,a'}(L(1), L(0)) | A, L(0) \right)$$

Then, evaluate the fitted function at the observed covariates histories $\{L(0)\}$ and the intervened exposure $A = a$. This gives the estimates

$$\hat{\bar{Q}}_{R_1}^{a,a'}(L(0)) = \mathbb{E} \left(\hat{\bar{Q}}_{M_1}^{a,a'}(L(1), L(0)) | A = a, L(0) \right)$$

6.6. ESTIMATION OF “CONDITIONAL” RANDOMIZED/INTERVENTIONAL NATURAL DIRECT (CRDE) AND

3. We have now estimated $\hat{Q}_{R_1}^{a,a'}(L(0))$ for each individual.

$$\hat{\Psi}^{a,a'} = \frac{1}{n} \sum_{i=1}^n \hat{Q}_{R_1}^{a,a'}(L(0))$$

$$\text{CRDE} = \hat{\Psi}^{1,0} - \hat{\Psi}^{0,0}$$

$$\text{CRIE} = \hat{\Psi}^{1,1} - \hat{\Psi}^{1,0}$$

```
rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")

## 1)  $Q^{a,a'}_{R2} = Y$ 
Q.death_R2 <- df2_int$Y_death
Q.qol_R2 <- df2_int$Y_qol

## 2) Obtain  $Q^{a,a'}_{L1}$ ,  $Q^{a,a'}_{M1}$ ,  $Q^{a,a'}_{R1}$ 
## 2.a) Obtain  $Q^{a,a'}_{L1}$ 
# Regress  $Q^{a,a'}_{R2}$  on observed values  $(L(0), A, L(1), M)$ 
L1.model.death <- glm(Q.death_R2 ~ L0_male + L0_soc_env + A0_PM2.5 + L1 +
  M_diabetes + A0_PM2.5:M_diabetes,
  family = "binomial", data = df2_int)
L1.model.qol <- glm(Q.qol_R2 ~ L0_male + L0_soc_env + A0_PM2.5 + L1 +
  M_diabetes + A0_PM2.5:M_diabetes,
  family = "gaussian", data = df2_int)

# Evaluate the fitted function at the observed mediator M and covariate history L(1), L(0)
# and the intervened exposure A = a
data.Ais0 <- data.Ais1 <- df2_int
data.Ais0$A0_PM2.5 <- 0
data.Ais1$A0_PM2.5 <- 1

# We will need 3 counterfactual quantities  $Q^{a,a'}$ :  $Q^{1,1}$ ,  $Q^{1,0}$  and  $Q^{0,0}$ 
Q11.death.L1 <- predict(L1.model.death, newdata = data.Ais1, type = "response")
Q10.death.L1 <- predict(L1.model.death, newdata = data.Ais1, type = "response")
Q00.death.L1 <- predict(L1.model.death, newdata = data.Ais0, type = "response")

Q11.qol.L1 <- predict(L1.model.qol, newdata = data.Ais1, type = "response")
Q10.qol.L1 <- predict(L1.model.qol, newdata = data.Ais1, type = "response")
Q00.qol.L1 <- predict(L1.model.qol, newdata = data.Ais0, type = "response")

## 2.b) Obtain  $Q^{a,a'}_{M1}$ 
# Regress  $Q^{a,a'}_{L1}$  on observed values  $(L(0), A, L(1))$ 
M1.model.death.11 <- glm(Q11.death.L1 ~ L0_male + L0_soc_env + A0_PM2.5 + L1,
  family = "quasibinomial", data = df2_int)
```

```

M1.model.death.10 <- glm(Q10.death.L1 ~ L0_male + L0_soc_env + A0_PM2.5 + L1,
  family = "quasibinomial", data = df2_int)
M1.model.death.00 <- glm(Q00.death.L1 ~ L0_male + L0_soc_env + A0_PM2.5 + L1,
  family = "quasibinomial", data = df2_int)

M1.model.qol.11 <- glm(Q11.qol.L1 ~ L0_male + L0_soc_env + A0_PM2.5 + L1,
  family = "gaussian", data = df2_int)
M1.model.qol.10 <- glm(Q10.qol.L1 ~ L0_male + L0_soc_env + A0_PM2.5 + L1,
  family = "gaussian", data = df2_int)
M1.model.qol.00 <- glm(Q00.qol.L1 ~ L0_male + L0_soc_env + A0_PM2.5 + L1,
  family = "gaussian", data = df2_int)

# Evaluate the fitted function at the observed covariate history L(1),L(0)
# and the intervened exposure A = a'
Q11.death.M1 <- predict(M1.model.death.11, newdata = data.Ais1, type = "response")
Q10.death.M1 <- predict(M1.model.death.10, newdata = data.Ais0, type = "response")
Q00.death.M1 <- predict(M1.model.death.00, newdata = data.Ais0, type = "response")

Q11.qol.M1 <- predict(M1.model.qol.11, newdata = data.Ais1, type = "response")
Q10.qol.M1 <- predict(M1.model.qol.10, newdata = data.Ais0, type = "response")
Q00.qol.M1 <- predict(M1.model.qol.00, newdata = data.Ais0, type = "response")

## 2.c) Obtain  $Q^{a,a'}_{R1}$ 
# Regress  $Q^{a,a'}_{M1}$  on observed values  $(L(0),A)$ 
R1.model.death.11 <- glm(Q11.death.M1 ~ L0_male + L0_soc_env + A0_PM2.5,
  family = "quasibinomial", data = df2_int)
R1.model.death.10 <- glm(Q10.death.M1 ~ L0_male + L0_soc_env + A0_PM2.5,
  family = "quasibinomial", data = df2_int)
R1.model.death.00 <- glm(Q00.death.M1 ~ L0_male + L0_soc_env + A0_PM2.5,
  family = "quasibinomial", data = df2_int)

R1.model.qol.11 <- glm(Q11.qol.M1 ~ L0_male + L0_soc_env + A0_PM2.5,
  family = "gaussian", data = df2_int)
R1.model.qol.10 <- glm(Q10.qol.M1 ~ L0_male + L0_soc_env + A0_PM2.5,
  family = "gaussian", data = df2_int)
R1.model.qol.00 <- glm(Q00.qol.M1 ~ L0_male + L0_soc_env + A0_PM2.5,
  family = "gaussian", data = df2_int)

# Evaluate the fitted function at the observed covariate history L(0)
# and the intervened exposure A = a
Q11.death.R1 <- predict(R1.model.death.11, newdata = data.Ais1, type = "response")
Q10.death.R1 <- predict(R1.model.death.10, newdata = data.Ais1, type = "response")
Q00.death.R1 <- predict(R1.model.death.00, newdata = data.Ais0, type = "response")

Q11.qol.R1 <- predict(R1.model.qol.11, newdata = data.Ais1, type = "response")

```


6.6. ESTIMATION OF “CONDITIONAL” RANDOMIZED/INTERVENTIONAL NATURAL DIRECT (CRDE) AND NATURAL INDIRECT (CRIE) EFFECTS

```
Q10.qol.R1 <- predict(R1.model.qol.10, newdata = data.Ais1, type = "response")
Q00.qol.R1 <- predict(R1.model.qol.00, newdata = data.Ais0, type = "response")

## 4) Estimate the Conditional Randomized Direct and Indirect Effects
## For death
CRDE.death <- mean(Q10.death.R1) - mean(Q00.death.R1)
CRDE.death
# [1] 0.07585836
CRIE.death <- mean(Q11.death.R1) - mean(Q10.death.R1)
CRIE.death
# [1] 0.006802907

## For quality of life
CRDE.qol <- mean(Q10.qol.R1) - mean(Q00.qol.R1)
CRDE.qol
# [1] -7.278295
CRIE.qol <- mean(Q11.qol.R1) - mean(Q10.qol.R1)
CRIE.qol
# [1] -1.015901
```

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

- the conditional “randomized” Natural Indirect effect is $\text{CRIE} \approx +0.7\%$ on death and $\text{CRIE} \approx -1.0$ on quality of life. This indirect effect corresponds to the specific path $A \rightarrow M \rightarrow Y$ (and can also contain the mediated interactive effect of due to the $A * M$ interaction on Y).
- the conditional “randomized” Natural Direct effect is $\text{CRDE} \approx +7.6\%$ on death and $\text{CRDE} \approx -7.3$ on quality of life. This direct effect corresponds to the combination of the paths $A \rightarrow Y$, $A \rightarrow L(1) \rightarrow Y$ and $A \rightarrow L(1) \rightarrow M \rightarrow Y$.

95% confidence intervals can be calculated by bootstrap.

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 $\mathbb{P}(A = a \mid L(0)) = g(A = a \mid L(0))$:

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

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 : $\hat{\mathbb{E}}(Y_a) = \frac{1}{n} \sum_{i=1}^n \frac{\mathbb{I}(A_i=a)}{\hat{g}(A=a_i \mid L(0)_i)} Y_i$

```

rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")

## 1. Estimate g
g.L <- glm(A0_PM2.5 ~ L0_male + L0_soc_env,
           family = "binomial", data = df2_int)

## 2. Predict each individual's probability of being exposed to her own exposure
# predict the probabilities P(A0_PM2.5=1|L(0)) & P(A0_PM2.5=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$A0_PM2.5 == 1] <- pred.g1.L[df2_int$A0_PM2.5 == 1]
gA.L[df2_int$A0_PM2.5 == 0] <- pred.g0.L[df2_int$A0_PM2.5 == 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$A0_PM2.5 == 1) * df2_int$Y_death) -
  mean(wt * as.numeric(df2_int$A0_PM2.5 == 0) * df2_int$Y_death)
IPTW.death
# [1] 0.08224947

IPTW.qol <- mean(wt * as.numeric(df2_int$A0_PM2.5 == 1) * df2_int$Y_qol) -
  mean(wt * as.numeric(df2_int$A0_PM2.5 == 0) * df2_int$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:

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

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$A0_PM2.5 == 1) * df2_int$Y_death) /
                 mean(wt * as.numeric(df2_int$A0_PM2.5 == 1))) -
  (mean(wt * as.numeric(df2_int$A0_PM2.5 == 0) * df2_int$Y_death) /
   mean(wt * as.numeric(df2_int$A0_PM2.5 == 0)))
s.IPTW.death
# [1] 0.08294185

s.IPTW.qol <- (mean(wt * as.numeric(df2_int$A0_PM2.5 == 1) * df2_int$Y_qol) /
               mean(wt * as.numeric(df2_int$A0_PM2.5 == 1))) -
  (mean(wt * as.numeric(df2_int$A0_PM2.5 == 0) * df2_int$Y_qol) /
   mean(wt * as.numeric(df2_int$A0_PM2.5 == 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 $\mathbb{P}(A = a | L(0)) = g(A = a | L(0))$ and $\mathbb{P}(M = m | L(0), A, L(1)) = g(M = m | L(0), A, L(1))$:

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

The following steps describe the implementation of the IPTW estimator

1. Estimate the treatment mechanisms $g(A = 1 | L(0))$ and $g(M = 1 | L(0), A, L(1))$
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 : $\hat{\mathbb{E}}(Y_{a,m}) = \frac{1}{n} \sum_{i=1}^n \frac{\mathbb{I}(A_i=a \cap M_i=m)}{\hat{g}(A=a_i|L(0)_i) \times \hat{g}(M=m_i|L(0)_i, A_i, L(1)_i)} Y_i$

```

rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")

## 1. Estimate gA and gM
gA.L <- glm(A0_PM2.5 ~ L0_male + L0_soc_env,
            family = "binomial", data = df2_int)
gM.L <- glm(M_diabetes ~ L0_male + L0_soc_env + A0_PM2.5 + L1,
            family = "binomial", data = df2_int)

## 2. Predict each individual's probability of being exposed to her own exposure
# predict the probabilities  $P(A0\_PM2.5=1|L(0))$  &  $P(A0\_PM2.5=0|L(0))$ 
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$A0_PM2.5 == 1] <- pred.gA1.L[df2_int$A0_PM2.5 == 1]
gAobs.L[df2_int$A0_PM2.5 == 0] <- pred.gA0.L[df2_int$A0_PM2.5 == 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$M_diabetes == 1] <- pred.gM1.L[df2_int$M_diabetes == 1]
gMobs.L[df2_int$M_diabetes == 0] <- pred.gM0.L[df2_int$M_diabetes == 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$A0_PM2.5 == 1 &
                                           df2_int$M_diabetes == 0) *
                           df2_int$Y_death) -
                      mean(wt * as.numeric(df2_int$A0_PM2.5 == 0 &
                                           df2_int$M_diabetes == 0) *
                           df2_int$Y_death))
CDE_IPTW_m0_death
# [1] 0.05874684

CDE_IPTW_m0_qol <- (mean(wt * as.numeric(df2_int$A0_PM2.5 == 1 &
                                           df2_int$M_diabetes == 0) *
                           df2_int$Y_qol) -

```

```

        mean(wt * as.numeric(df2_int$A0_PM2.5==0 &
                             df2_int$M_diabetes == 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$A0_PM2.5 == 1 &
                                             df2_int$M_diabetes == 1) *
                             df2_int$Y_death) -
                      mean(wt * as.numeric(df2_int$A0_PM2.5==0 &
                                             df2_int$M_diabetes == 1) *
                             df2_int$Y_death))
CDE_IPTW_m1_death
# [1] 0.101733

CDE_IPTW_m1_qol <- (mean(wt * as.numeric(df2_int$A0_PM2.5 == 1 &
                                             df2_int$M_diabetes == 1) *
                             df2_int$Y_qol) -
                    mean(wt * as.numeric(df2_int$A0_PM2.5==0 &
                                             df2_int$M_diabetes == 1) *
                             df2_int$Y_qol))
CDE_IPTW_m1_qol
# [1] -8.185866

```

7.2.2 Stabilized IPTW for the CDE

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

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

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$A0_PM2.5 == 1 &
                                             df2_int$M_diabetes == 0) *
                             df2_int$Y_death) /
                      mean(wt * as.numeric(df2_int$A0_PM2.5 == 1 &
                                             df2_int$M_diabetes == 0))) -
                      (mean(wt * as.numeric(df2_int$A0_PM2.5 == 0 &

```

```

df2_int$M_diabetes == 0) *
df2_int$Y_death) /
mean(wt * as.numeric(df2_int$A0_PM2.5 == 0 &
df2_int$M_diabetes == 0)))
CDE_sIPTW_m0_death
# [1] 0.0601292

CDE_sIPTW_m0_qol <- (mean(wt * as.numeric(df2_int$A0_PM2.5 == 1 &
df2_int$M_diabetes == 0) *
df2_int$Y_qol) /
mean(wt * as.numeric(df2_int$A0_PM2.5 == 1 &
df2_int$M_diabetes == 0))) -
(mean(wt * as.numeric(df2_int$A0_PM2.5 == 0 &
df2_int$M_diabetes == 0) *
df2_int$Y_qol) /
mean(wt * as.numeric(df2_int$A0_PM2.5 == 0 &
df2_int$M_diabetes == 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$A0_PM2.5 == 1 &
df2_int$M_diabetes == 1) *
df2_int$Y_death) /
mean(wt * as.numeric(df2_int$A0_PM2.5 == 1 &
df2_int$M_diabetes == 1))) -
(mean(wt * as.numeric(df2_int$A0_PM2.5 == 0 &
df2_int$M_diabetes == 1) *
df2_int$Y_death) /
mean(wt * as.numeric(df2_int$A0_PM2.5 == 0 &
df2_int$M_diabetes == 1)))
CDE_sIPTW_m1_death
# [1] 0.09030186

CDE_sIPTW_m1_qol <- (mean(wt * as.numeric(df2_int$A0_PM2.5 == 1 &
df2_int$M_diabetes == 1) *
df2_int$Y_qol) /
mean(wt * as.numeric(df2_int$A0_PM2.5 == 1 &
df2_int$M_diabetes == 1))) -
(mean(wt * as.numeric(df2_int$A0_PM2.5 == 0 &
df2_int$M_diabetes == 1) *
df2_int$Y_qol) /
mean(wt * as.numeric(df2_int$A0_PM2.5 == 0 &
df2_int$M_diabetes == 1)))
CDE_sIPTW_m1_qol

```



```
# [1] -10.03045
```

7.3 Estimation of the PNDE and TNIE by Inverse Odds Ratio Weighing

(Tchetgen Tchetgen 2013) described the estimation of the Pure Natural Direct Effect (PNDE) and Total Natural Indirect Effect (TNIE) using Inverse Odds Ratio Weighting (IORW), when the the two effects are identifiable (Causal model 1, Figure 3.1).

A practical guidance is also given in (Nguyen et al. 2015).

The approach is particularly useful with multiple mediators, as we don't need to estimate a model for each mediator of interest. We also don't need to make any assumptions regarding the exposure-mediator interaction effects on the outcome.

The IORW approach relies on :

- a single model of the exposure conditional on the set of mediators and baseline confounders,
- and 2 models of the outcome, circonvventing the need to specify the possible interactions between the exposure and mediators

The analysis relies on the following steps:

1. Fit a standard multiple logistic regression model for the exposure A , conditional on mediators M and baseline confounders $L(0)$ and $L(1)$
2. Compute an IORW weight by taking the inverse of the predicted odds ratio from step 1 for each observation in the exposed group (the unexposed group weight equals 1)
3. Estimate the direct effect of the exposure via a glm of the regression of the outcome Y on the exposure A and baseline confounders $L(0)$ and $L(1)$, with the appropriate link function and the weights from step 2
4. Estimate the total effect of the exposure A on the outcome Y using a standard glm with the appropriate link function (adjusted for baseline confounders)
5. Calculate indirect effects by substracting the direct effect from the total effect
6. Bootstrap effect estimates to get 95% confidence intervals

Note: Examples of sensitivity analysis to test the unmeasured confounding assumptions are also described in (Nguyen et al. 2015) (not shown below).

```
rm(list=ls())
df1_int <- read.csv(file = "data/df1_int.csv")

## 1) Fit a standard multiple logistic regression model for the exposure A,
##    conditional on mediators M and baseline confounders L(0) and L(1)
g.A.L0 <- glm(A0_PM2.5 ~ M_diabetes + L0_male + L0_soc_env + L1,
              family = "binomial",
              data = df1_int)
summary(g.A.L0)
# Coefficients:
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept) -2.883345    0.080069 -36.011 < 2e-16 ***
# M_diabetes   0.562935    0.065539   8.589 < 2e-16 ***
# L0_male      0.374904    0.064828   5.783 7.34e-09 ***
# L0_soc_env   0.600956    0.073872   8.135 4.12e-16 ***
# L1          -0.007768    0.070141  -0.111 0.912

## 2) Compute an IORW weight by taking the inverse of the predicted odds ratio
##    from step 1 for each observation in the exposed group (the unexposed
##    group weight equals 1)
p <- predict(g.A.L0, type = "response")

iorw <- rep(NA, nrow(df1_int))
iorw[df1_int$A0_PM2.5 == 0] <- 1
iorw[df1_int$A0_PM2.5 == 1] <- ((1 - p[df1_int$A0_PM2.5 == 1]) /
                                p[df1_int$A0_PM2.5 == 1])

## 3) Estimate the direct effect of the exposure via a glm of the regression of
##    the outcome Y on the exposure A and baseline confounders L(0) and L(1),
##    with the appropriate link function and the weights from step 2
Dir.Y.model.death <- glm(Y_death ~ A0_PM2.5 + L0_male + L0_soc_env + L1,
                        weights = iorw,
                        family = "gaussian", # to get risk differences
                        data = df1_int)
summary(Dir.Y.model.death)
#               Estimate Std. Error t value Pr(>|t|)
# (Intercept)  0.102447    0.009481  10.805 < 2e-16 ***
# A0_PM2.5     0.063862    0.008313   7.682 1.71e-14 ***
# L0_male      0.066841    0.008318   8.035 1.04e-15 ***
# L0_soc_env   0.056381    0.008625   6.537 6.57e-11 ***
# L1           0.091272    0.009103  10.026 < 2e-16 ***

PNDE <- coef(Dir.Y.model.death)["A0_PM2.5"]
# 0.06386221
```

7.3. ESTIMATION OF THE PNDE AND TNIE BY INVERSE ODDS RATIO WEIGHTING 107

```
## 4) Estimate the total effect of the exposure A on the outcome Y using a standard
##    glm with the appropriate link function (adjusted for baseline confounders)
Tot.Y.model.death <- glm(Y_death ~ AO_PM2.5 + LO_male + LO_soc_env + L1,
                        family = "gaussian", # to get risk differences
                        data = df1_int)
summary(Tot.Y.model.death)
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) 0.111516   0.008328  13.390 < 2e-16 ***
# AO_PM2.5    0.076673   0.012777   6.001 2.03e-09 ***
# LO_male     0.050049   0.008042   6.224 5.05e-10 ***
# LO_soc_env  0.060178   0.008413   7.153 9.07e-13 ***
# L1          0.080243   0.008813   9.106 < 2e-16 ***

# the total effect is 0.076673

## 5) Calculate indirect effects by subtracting the direct effect from the total
##    effect
TNIE <- coef(Tot.Y.model.death)["AO_PM2.5"] - PNDE
# 0.01281078

## 6) bootstrap effect estimates to get 95% confidence intervals
```

We can use the `CMAverse` package to obtain those estimations by IORW.

```
rm(list=ls())
df1_int <- read.csv(file = "data/df1_int.csv")
library(CMAverse)

## Using the CMAverse to estimate PNDE and TNIE by IORW
res_msm_df1.death <- cmest(data = df1_int,
                          model = "iorw",
                          outcome = "Y_death",
                          exposure = "AO_PM2.5",
                          mediator = "M_diabetes",
                          basec = c("LO_male", "LO_soc_env", "L1"),
                          postc = NULL,
                          # EMint = TRUE, # not needed for IORW
                          ereg = "logistic", # exposure regression model g(A=1|L(0))
                          yreg = "linear", # to get risk difference
                          # mreg = list("logistic"), # not needed for IORW
                          # wmmomreg = list("logistic"), # not needed for IORW
                          # wmdenomreg = list("logistic"), # not needed for IORW
                          astar = 0, #E(Y_{A=0,M=1})
                          a = 1, #E(Y_{A=1,M=1})
```

```

# mval = list(), # not needed for IORW
estimation = "imputation",
inference = "bootstrap",
nboot = 2)

summary(res_msm_df1.death)
# Causal Mediation Analysis
#
# # Outcome regression for the total effect:
# Call:
# glm(formula = Y_death ~ A0_PM2.5 + L0_male + L0_soc_env + L1,
#      family = gaussian(), data = getCall(x$reg.output$yregTot)$data,
#      weights = getCall(x$reg.output$yregTot)$weights)
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  0.111516   0.008328  13.390 < 2e-16 ***
# A0_PM2.5     0.076673   0.012777   6.001 2.03e-09 ***
# L0_male      0.050049   0.008042   6.224 5.05e-10 ***
# L0_soc_env   0.060178   0.008413   7.153 9.07e-13 ***
# L1           0.080243   0.008813   9.106 < 2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# # Outcome regression for the direct effect:
# Call:
# glm(formula = Y_death ~ A0_PM2.5 + L0_male + L0_soc_env + L1,
#      family = gaussian(), data = getCall(x$reg.output$yregDir)$data,
#      weights = getCall(x$reg.output$yregDir)$weights)
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  0.102447   0.009481  10.805 < 2e-16 ***
# A0_PM2.5     0.063862   0.008313   7.682 1.71e-14 ***
# L0_male      0.066841   0.008318   8.035 1.04e-15 ***
# L0_soc_env   0.056381   0.008625   6.537 6.57e-11 ***
# L1           0.091272   0.009103  10.026 < 2e-16 ***
#
# # Exposure regression for weighting:
# Call:
# glm(formula = A0_PM2.5 ~ M_diabetes + L0_male + L0_soc_env +
#      L1, family = binomial(), data = getCall(x$reg.output$ereg)$data,
#      weights = getCall(x$reg.output$ereg)$weights)
#
# Coefficients:
#              Estimate Std. Error z value Pr(>|z|)
# (Intercept) -2.883345   0.080069 -36.011 < 2e-16 ***
# M_diabetes   0.562935   0.065539   8.589 < 2e-16 ***

```

7.4. ESTIMATION OF “CONDITIONAL” RANDOMIZED/INTERVENTIONAL NATURAL DIRECT (CRDE) AND INDIRECT EFFECTS (CRIE)

```
# L0_male      0.374904    0.064828    5.783 7.34e-09 ***
# L0_soc_env   0.600956    0.073872    8.135 4.12e-16 ***
# L1           -0.007768    0.070141   -0.111    0.912
#
# # Effect decomposition on the mean difference scale via the inverse odds ratio weighting approach
#
# Direct counterfactual imputation estimation with
# bootstrap standard errors, percentile confidence intervals and p-values
#
#      Estimate Std.error 95% CIL 95% CIU P.val
# te    0.076673  0.009812 0.072033  0.085 <2e-16 ***
# pnde   0.063862  0.003280 0.056996  0.061 <2e-16 ***
# tnies  0.012811  0.006532 0.015038  0.024 <2e-16 ***
# pm     0.167083  0.052652 0.208409  0.279 <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# (te: total effect;
# pnde: pure natural direct effect;
# tnies: total natural indirect effect;
# pm: proportion mediated)
```

7.4 Estimation of “Conditional” Randomized/Interventional Natural Direct (CRDE) and Indirect Effects (CRIE)

7.4.1 IPTW for the CRDE and CRIE

Zheng and van der Laan (Zheng and van der Laan 2017) described “Conditional” Randomized Direct and Indirect Effects, using random draws from the counterfactual distribution of the mediator, conditional on both $L(0)$ and $L(1)$. The Average Total Effect (ATE) can be decomposed into the sum of :

- a Conditional Randomized Natural Direct Effect (CRDE): $CRDE = \mathbb{E}(Y_{1,\Gamma_0|L(0),L(1)}) - \mathbb{E}(Y_{0,\Gamma_0|L(0),L(1)})$,
- and Conditional Randomized Natural Indirect Effect (CRIE) $CRIE = \mathbb{E}(Y_{1,\Gamma_1|L(0),L(1)}) - \mathbb{E}(Y_{1,\Gamma_0|L(0),L(1)})$.

Under the identifiability conditions, the quantity of $\mathbb{E}(Y_{a,\Gamma_{a'}|L(0),L(1)})$ can be estimated by IPTW:

$$\Psi_{IPTW}^{a,a'} = \mathbb{E} \left[Y_{a,\Gamma_{a'}|L(0),L(1)} \right] = \frac{1}{n} \sum_{i=1}^n D_{IPTW}^{a,a'}$$

where

$$D_{\text{IPTW}}^{a,a'} = Y \frac{I(A=a)}{p_A(A=a | L(0))} \frac{p_M(M | A=a', L(1), L(0))}{p_M(M | A=a, L(1), L(0))}$$

and the variance of the estimator $\Psi_{\text{IPTW}}^{a,a'}$ is $\text{var}(\Psi_{\text{IPTW}}^{a,a'}) = \frac{\text{var}(D_{\text{IPTW}}^{a,a'})}{n}$, which can be used to estimate 90% confidence intervals.

```
rm(list=ls())
df2_int <- read.csv(file = "./data/df2_int.csv")

## In order to estimate Psi^{a,a'}
## 1. Estimate gA and gM
gA <- glm(A0_PM2.5 ~ L0_male + L0_soc_env,
          family = "binomial", data = df2_int)
gM <- glm(M_diabetes ~ L0_male + L0_soc_env + A0_PM2.5 + L1,
          family = "binomial", data = df2_int)

## 2. Predict each individual's probability of being exposed to her own exposure
# predict the probabilities P(A0_PM2.5=1/L(0)) & P(A0_PM2.5=0/L(0))
pred.gA1 <- predict(gA, type = "response")
pred.gA0 <- 1 - pred.gA1
# the predicted probability of the observed treatment A_i=a is :
gAobs <- rep(NA, nrow(df2_int))
gAobs[df2_int$A0_PM2.5 == 1] <- pred.gA1[df2_int$A0_PM2.5 == 1]
gAobs[df2_int$A0_PM2.5 == 0] <- pred.gA0[df2_int$A0_PM2.5 == 0]

# predict the probabilities
# P(M=1/L(0), A=1, L(1)) & P(M=0/L(0), A=1, L(1)) setting A = 1
# and P(M=1/L(0), A=0, L(1)) & P(M=0/L(0), A=0, L(1)) setting A = 0
data.Ais0 <- data.Ais1 <- df2_int
data.Ais0$A0_PM2.5 <- 0
data.Ais1$A0_PM2.5 <- 1

pred.gM1.Ais1 <- predict(gM, newdata = data.Ais1, type = "response")
pred.gM0.Ais1 <- 1 - pred.gM1.Ais1
# the predicted probability of the observed treatment M_i=m is :
gMobs.Ais1 <- rep(NA, nrow(df2_int))
gMobs.Ais1[df2_int$M_diabetes == 1] <- pred.gM1.Ais1[df2_int$M_diabetes == 1]
gMobs.Ais1[df2_int$M_diabetes == 0] <- pred.gM0.Ais1[df2_int$M_diabetes == 0]

pred.gM1.Ais0 <- predict(gM, newdata = data.Ais0, type = "response")
pred.gM0.Ais0 <- 1 - pred.gM1.Ais0
# the predicted probability of the observed treatment M_i=m is :
gMobs.Ais0 <- rep(NA, nrow(df2_int))
gMobs.Ais0[df2_int$M_diabetes == 1] <- pred.gM1.Ais0[df2_int$M_diabetes == 1]
gMobs.Ais0[df2_int$M_diabetes == 0] <- pred.gM0.Ais0[df2_int$M_diabetes == 0]
```

7.4. ESTIMATION OF “CONDITIONAL” RANDOMIZED/INTERVENTIONAL NATURAL DIRECT (CRDE) AND

```
## 3. Calculate  $D^{\{a,a'\}}$  - influence curve of the IPTW estimator
D.death.11 <- (df2_int$Y_death * (I(df2_int$A0_PM2.5 == 1) / gAobs) *
              (gMobs.Ais1 / gMobs.Ais1))
D.death.10 <- (df2_int$Y_death * (I(df2_int$A0_PM2.5 == 1) / gAobs) *
              (gMobs.Ais0 / gMobs.Ais1))
D.death.00 <- (df2_int$Y_death * (I(df2_int$A0_PM2.5 == 0) / gAobs) *
              (gMobs.Ais0 / gMobs.Ais0))

D.qol.11 <- (df2_int$Y_qol * (I(df2_int$A0_PM2.5 == 1) / gAobs) *
            (gMobs.Ais1 / gMobs.Ais1))
D.qol.10 <- (df2_int$Y_qol * (I(df2_int$A0_PM2.5 == 1) / gAobs) *
            (gMobs.Ais0 / gMobs.Ais1))
D.qol.00 <- (df2_int$Y_qol * (I(df2_int$A0_PM2.5 == 0) / gAobs) *
            (gMobs.Ais0 / gMobs.Ais0))

## 4. Calculate CRDE and CRIE
## For death
CRDE.death <- mean(D.death.10) - mean(D.death.00)
CRDE.death
# [1] 0.07405068
CRIE.death <- mean(D.death.11) - mean(D.death.10)
CRIE.death
# [1] 0.00819879

## For quality of life
CRDE.qol <- mean(D.qol.10) - mean(D.qol.00)
CRDE.qol
# [1] -7.563116
CRIE.qol <- mean(D.qol.11) - mean(D.qol.10)
CRIE.qol
# [1] -0.8736813

## 5. Calculate 95% CI based on the influence curve  $D^{\{a,a'\}}$ 
# the variance of the estimator  $\Psi^{\{a,a'\}} = \text{mean}(D^{\{a,a'\}})$  is  $\text{var}(D^{\{a,a'\}}) / n$ 
se.CRDE.death <- sqrt(var(D.death.10 - D.death.00) / nrow(df2_int))
c(CRDE.death - qnorm(0.975) * se.CRDE.death,
  CRDE.death + qnorm(0.975) * se.CRDE.death)
# [1] 0.04141535 0.1068602
se.CRIE.death <- sqrt(var(D.death.11 - D.death.10) / nrow(df2_int))
c(CRIE.death - qnorm(0.975) * se.CRIE.death,
  CRIE.death + qnorm(0.975) * se.CRIE.death)
# [1] 0.003380238 0.013017342

se.CRDE.qol <- sqrt(var(D.qol.10 - D.qol.00) / nrow(df2_int))
c(CRDE.qol - qnorm(0.975) * se.CRDE.qol,
```

```

CRDE.qol + qnorm(0.975) * se.CRDE.qol)
# [1] -11.748095 -3.378137
se.CRIE.qol <- sqrt(var(D.qol.11 - D.qol.10) / nrow(df2_int))
c(CRIE.qol - qnorm(0.975) * se.CRIE.qol,
  CRIE.qol + qnorm(0.975) * se.CRIE.qol)
# [1] -1.4111404 -0.3362223

```

Results are close to the estimations obtained by g-computation. IPTW estimations are known to be more sensitive to positivity issues, with larger confidence intervals (they can be too conservative with more than 95% coverage).

- the conditional “randomized” Natural Indirect effect is $\text{CRIE} \approx +0.8\%$, $95\%CI = [0.3\%, 1.3\%]$ on death and $\text{CRIE} \approx -0.9$, $95\%CI = [-1.4, -0.3]$ on quality of life. This indirect effect corresponds to the specific path $A \rightarrow M \rightarrow Y$ (and can also contain the mediated interactive effect of due to the $A * M$ interaction on Y).
- the conditional “randomized” Natural Direct effect is $\text{CRDE} \approx +7.4$, $95\%CI = [4.1\%, 10.7\%]$ on death and $\text{CRDE} \approx -7.6$, $95\%CI = [-11.7, -3.4]$ on quality of life. This direct effect corresponds to the combination of the paths $A \rightarrow Y$, $A \rightarrow L(1) \rightarrow Y$ and $A \rightarrow L(1) \rightarrow M \rightarrow Y$.

95% confidence intervals can be calculated by bootstrap.

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 is 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 confounder $L(1)$ 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 :

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

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):

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

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 $\text{ATE}|L(0)$ (instead of the marginal ATE defined above), projecting the counterfactual outcomes on a parametric model such as the following:

$$\mathbb{E}(Y_a | L(0)) = \alpha_0 + \alpha_A a + \alpha_{\text{male}} L_{\text{male}}(0) + \alpha_{\text{soc.env.}} L_{\text{soc.env.}}(0) \quad (8.2)$$

so that $\text{ATE} | L(0) = \mathbb{E}(Y_{A=1} | L(0)) - \mathbb{E}(Y_{A=0} | L(0)) = \alpha_A$ 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:

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

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

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

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, weighted by individual weights w_i or sw_i :

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

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 .

```
rm(list=ls())
df1_int <- read.csv(file = "data/df1_int.csv")

## 1. Denominator of the weight
# 1a. Estimate  $g(A=a_i|L(0))$  (denominator of the weight)
g.A.L <- glm(A0_PM2.5 ~ L0_male + L0_soc_env,
             family = "binomial", data = df1_int)

# 1b. Predict each individual's probability of being exposed to her own exposure
# predict the probabilities  $P(A0\_PM2.5 = 1)$  &  $P(A0\_PM2.5 = 0)$ 
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$A0_PM2.5==1] <- pred.g1.L[df1_int$A0_PM2.5==1]
gAi.L[df1_int$A0_PM2.5==0] <- pred.g0.L[df1_int$A0_PM2.5==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_PM2.5 ~ L0_male,
              family = "binomial", data = df1_int)

# 2b. Predict each individual's probability of being exposed to her own exposure
# predict the probabilities  $P(A0\_PM2.5 = 1 | sex)$  &  $P(A0\_PM2.5 = 0 | sex)$ 
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$A0_PM2.5==1] <- pred.g1.sex[df1_int$A0_PM2.5==1]
gAi.sex[df1_int$A0_PM2.5==0] <- pred.g0.sex[df1_int$A0_PM2.5==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$A0_PM2.5)
boxplot(sw ~ df1_int$A0_PM2.5)
par(mfcol = c(1,1))

## applying these weights creates a pseudo-population were the baseline
## confounders are balanced, relative to the exposure:
## before applying weights to the individuals:
table(df1_int$L0_male, df1_int$A0_PM2.5, deparse.level = 2)
#
#           df1_int$A0_PM2.5
# df1_int$L0_male    0      1
#           0 4527  463
#           1 4349  661
prop.table(table(df1_int$L0_male, df1_int$A0_PM2.5, deparse.level = 2),
            margin = 2)
#
#           df1_int$A0_PM2.5
# df1_int$L0_male    0      1
#           0 0.5100270 0.4119217
#           1 0.4899730 0.5880783 # sex is unbalanced: 49% versus 59%

## after applying weights to the individuals:
library(questionr) # The questionr package enables to describe weighted populations
wtd.table(x = df1_int$L0_male, y = df1_int$A0_PM2.5,
           weights = w)
#
#           0      1
# 0 4989.425 4918.462
# 1 5010.862 5057.676
prop.table(wtd.table(x = df1_int$L0_male, y = df1_int$A0_PM2.5,
                     weights = w), margin = 2)
#
#           0      1
# 0 0.4989282 0.4930227
# 1 0.5010718 0.5069773 # sex is balanced in the weighted population

## 4. Estimate coefficients of the MSM using a weighted regression  $E(Y | A, \text{sex})$ 
# 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_PM2.5 + L0_male + A0_PM2.5*L0_male,
            weights = w,
            family = "gaussian",
            data = df1_int)
coef(msm1)

```

```

# (Intercept)      A0_PM2.5      L0_male A0_PM2.5:L0_male
# 0.17573472      0.05883228      0.04598911      0.03077614

msm2 <- glm(Y_death ~ A0_PM2.5 + L0_male + A0_PM2.5*L0_male,
            weights = sw,
            family = "gaussian",
            data = df1_int)
coef(msm2)
# (Intercept)      A0_PM2.5      L0_male A0_PM2.5:L0_male
# 0.17573472      0.05883228      0.04598911      0.03077614

## 5. Estimate the ATE stratified by sex
# According to MSM1 (with simple weights)
ATE.msm1.male0 <- coef(msm1)["A0_PM2.5"]
# 0.05883228
ATE.msm1.male1 <- coef(msm1)["A0_PM2.5"] + coef(msm1)["A0_PM2.5:L0_male"]
# 0.08960842

# According to the MSM2 (with stabilized weights)
ATE.msm2.male0 <- coef(msm2)["A0_PM2.5"]
# 0.05883228
ATE.msm2.male1 <- coef(msm2)["A0_PM2.5"] + coef(msm2)["A0_PM2.5:L0_male"]
# 0.08960842
# 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 estimates

```

The ATE estimates of death probability using an MSM estimated by IPTW are respectively +5.9% in women and +9.0% 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” ($ATE | L_{\text{male}}(0) = 0$) = 0.0688 in women,
- the “true” ($ATE | L_{\text{male}}(0) = 1$) = 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 $\bar{Q} = \mathbb{E}(Y \mid A, L(0))$. Don't forget to add an $A * L_{\text{male}}(0)$ interaction term, because we aim to estimate the effect of the exposure A , modified by sex.
2. Use this estimate to predict an outcome for each subject under the counterfactual scenarios $\hat{\bar{Q}}(A = 0)_i$ and $\hat{\bar{Q}}(A = 1)_i$, by evaluating the regression fit \bar{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 $\hat{\bar{Q}}(A = 0)$;
 - the second half of the long dataset corresponds to the second counterfactual scenario with $A = 1$ for all individuals and $\hat{\bar{Q}}(A = 1)$ for the predicted counterfactual column.
4. Fit the MSM $\mathbb{E}[Y_a \mid L_{\text{male}}(0)]$ using the long dataset.

```
rm(list=ls())
df1_int <- read.csv(file = "data/df1_int.csv")

## 1. Estimate Qbar
Q.tot.death <- glm(Y_death ~ AO_PM2.5 + LO_male + LO_soc_env +
                  AO_PM2.5:LO_male, # don't forget the interaction term
                  family = "binomial", data = df1_int)
# The final result would be slightly 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$AO_PM2.5 <- 1
data.A0$AO_PM2.5 <- 0

# predict values under the same name in the corresponding counterfactual dataset
data.A1$Ya.death.pred <- predict(Q.tot.death, newdata = data.A1, type = "response")
data.A0$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_PM2.5 + L0_male + A0_PM2.5:L0_male,
                     family = "gaussian",
                     data = data.2scenarios)

coef(MSM.ATE.gcomp)
# (Intercept)          A0_PM2.5          L0_male A0_PM2.5:L0_male
# 0.17571658      0.06108129      0.04602303      0.02516386

## 5. Estimate the ATE stratified by sex
# According to MSM.ATE.gcomp
ATE.MSM.gcomp.male0 <- coef(MSM.ATE.gcomp) ["A0_PM2.5"]
# 0.06108129
ATE.MSM.gcomp.male1 <- (coef(MSM.ATE.gcomp) ["A0_PM2.5"] +
                        coef(MSM.ATE.gcomp) ["A0_PM2.5:L0_male"])
# 0.08624515
```

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 $CDE_m = \mathbb{E}(Y_{am}) - \mathbb{E}(Y_{a^*m})$.

Using the following MSM

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

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

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

$$CDE_m = \alpha_A (a - a^*) + \alpha_{A*M} \times (a - a^*) \times m$$

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

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)):

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

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

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

The “no-unmeasured confounding” assumption is addressed by the application of weights sw_i , which balances 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.

8.2.2.1 Example 1, with no intermediate confounder affected by the exposure

```
### MSM of CDE, estimated by IPTW -----
rm(list=ls())
df1_int <- read.csv(file = "data/df1_int.csv")

## 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_PM2.5 ~ L0_male + L0_soc_env,
             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$A0_PM2.5==1] <- predict(g.A.L, type="response")[df1_int$A0_PM2.5==1]
gAi.L[df1_int$A0_PM2.5==0] <- (1 - predict(g.A.L, type="response"))[df1_int$A0_PM2.5==0]

# 1c. Estimate g(A=a_i) (numerator of the weight)
g.A <- glm(A0_PM2.5 ~ 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$A0_PM2.5==1] <- predict(g.A, type="response")[df1_int$A0_PM2.5==1]
gAi[df1_int$A0_PM2.5==0] <- (1 - predict(g.A, type="response"))[df1_int$A0_PM2.5==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_diabetes ~ L0_male + L0_soc_env + A0_PM2.5 + 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$M_diabetes==1] <- predict(g.M.L, type="response")[df1_int$M_diabetes==1]
```



```

gMi.L[df1_int$M_diabetes==0] <- (1 - predict(g.M.L, type="response"))[df1_int$M_diabetes==0]

# 2c. Estimate  $g(M=m_i/A)$  (numerator of the weight)
g.M.A <- glm(M_diabetes ~ A0_PM2.5, 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$M_diabetes==1] <- predict(g.M.A, type="response")[df1_int$M_diabetes==1]
gMi.A[df1_int$M_diabetes==0] <- (1 - predict(g.M.A, type="response"))[df1_int$M_diabetes==0]
# 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  $E(Y | A, \text{sex})$ 
# a GLM with gaussian family can be applied to estimate risk differences
msm_cde <- glm(Y_death ~ A0_PM2.5 + M_diabetes + A0_PM2.5*M_diabetes,
               weights = sw_cde,
               family = "gaussian",
               data = df1_int)

coef(msm_cde)
# (Intercept)          A0_PM2.5          M_diabetes A0_PM2.5:M_diabetes
# 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_PM2.5"]
# 0.06798282
CDE_mis1 <- coef(msm_cde)["A0_PM2.5"] + coef(msm_cde)["A0_PM2.5:M_diabetes"]
# 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.

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).

8.2.2.2 Example 2, with intermediate confounder affected by the exposure

Below, we estimate the CDE by hand, and using the `ltmle` package and the `CMAverse` package

```

### MSM of CDE, estimated by IPTW
rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")

## 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_PM2.5 ~ L0_male + L0_soc_env,
             family = "binomial", data = df2_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(df2_int))
gAi.L[df2_int$A0_PM2.5==1] <- predict(g.A.L, type="response")[df2_int$A0_PM2.5==1]
gAi.L[df2_int$A0_PM2.5==0] <- (1 - predict(g.A.L, type="response"))[df2_int$A0_PM2.5==0]

# 1c. Estimate  $g(A=a_i)$  (numerator of the weight)
g.A <- glm(A0_PM2.5 ~ 1, family = "binomial", data = df2_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(df2_int))
gAi[df2_int$A0_PM2.5==1] <- predict(g.A, type="response")[df2_int$A0_PM2.5==1]
gAi[df2_int$A0_PM2.5==0] <- (1 - predict(g.A, type="response"))[df2_int$A0_PM2.5==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_diabetes ~ L0_male + L0_soc_env + A0_PM2.5 + L1,
             family = "binomial", data = df2_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(df2_int))
gMi.L[df2_int$M_diabetes==1] <- predict(g.M.L, type="response")[df2_int$M_diabetes==1]
gMi.L[df2_int$M_diabetes==0] <- (1 - predict(g.M.L, type="response"))[df2_int$M_diabetes==0]

# 2c. Estimate  $g(M=m_i/A)$  (numerator of the weight)
g.M.A <- glm(M_diabetes ~ A0_PM2.5, family = "binomial", data = df2_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(df2_int))
gMi.A[df2_int$M_diabetes==1] <- predict(g.M.A, type="response")[df2_int$M_diabetes==1]
gMi.A[df2_int$M_diabetes==0] <- (1 - predict(g.M.A, type="response"))[df2_int$M_diabetes==0]
# 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  $E(Y | A, \text{sex})$ 
# a GLM with gaussian family can be applied to estimate risk differences
msm_cde <- glm(Y_death ~ A0_PM2.5 + M_diabetes + A0_PM2.5*M_diabetes,
               weights = sw_cde,
               family = "gaussian",
               data = df2_int)
coef(msm_cde)
# (Intercept)          A0_PM2.5          M_diabetes A0_PM2.5:M_diabetes
# 0.17932146         0.06012920         0.07239661         0.03017266

## 5. Estimate CDE for m=0 and for m=1 using the MSM's coefficients
CDE_mis0 <- coef(msm_cde)["A0_PM2.5"]
# 0.0601292
CDE_mis1 <- coef(msm_cde)["A0_PM2.5"] + coef(msm_cde)["A0_PM2.5:M_diabetes"]
# 0.09030186

### Estimation by IPTW using ltmle package
library(ltmle)
Qform <- c(L1="Q.kplus1 ~ L0_male + L0_soc_env + A0_PM2.5",
          Y_death="Q.kplus1 ~ L0_male + L0_soc_env + L1 +
                  A0_PM2.5 * M_diabetes")
gform <- c("A0_PM2.5 ~ L0_male + L0_soc_env",
          "M_diabetes ~ L0_male + L0_soc_env + A0_PM2.5 + L1")
data_binary <- subset(df2_int, select = c(L0_male, L0_soc_env,
                                         A0_PM2.5, L1,
                                         M_diabetes, Y_death))
CDE_ltmle_M0_death <- ltmle(data = data_binary,
                           Anodes = c("A0_PM2.5", "M_diabetes"),
                           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,
                           estimate.time = FALSE, # estimate computation time?
                           iptw.only = TRUE, # for only IPTW estimations (no TMLE)
                           gcomp = FALSE,
                           variance.method = "iptw") # IPTW influence curve

# CDE with M=0

```

```

summary(CDE_ltmle_M0_death, estimator = "iptw")$effect.measures$ATE
# $estimate
# [1] 0.0601292
# $CI
#           2.5%      97.5%
# [1,] 0.02424819 0.09601021

CDE_ltmle_M1_death <- ltmle(data = data_binary,
                           Anodes = c("AO_PM2.5", "M_diabetes"),
                           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,M
                           SL.library = NULL,
                           estimate.time = FALSE, # estimate computation time?
                           iptw.only = TRUE, # for only IPTW estimations (no TMLE)
                           gcomp = FALSE,
                           variance.method = "iptw") # IPTW influence curve

# CDE with M=1
summary(CDE_ltmle_M1_death, estimator = "iptw")$effect.measures$ATE
# $estimate
# [1] 0.09030186
# $CI
#           2.5%      97.5%
# [1,] 0.04084792 0.1397558

## we obtain the same results as the IPTW estimation by hand

### Estimation by IPTW using ltmle package
library(CMAverse)
rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")
cmdag(outcome = "Y_death", exposure = "AO_PM2.5", mediator = "M_diabetes",
      basec = c("LO_male", "LO_soc_env"), postc = "L1", node = TRUE, text_col = "white")
# In this setting, we can use the marginal structural model and the $g$-formula approach

## The Marginal Structural Model
res_msm_RD <- cmest(data = df2_int,
                   model = "msm",
                   outcome = "Y_death",
                   exposure = "AO_PM2.5",

```

```

mediator = "M_diabetes",
basec = c("L0_male", "L0_soc_env"),
postc = "L1",
EMint = TRUE, # E*M interaction
ereg = "logistic",
yreg = "linear", # MSM is a linear regression (to get RD)
mreg = list("logistic"),
wmnomreg = list("logistic"),
wmnomreg = list("logistic"),
astar = 0, #E(Y_{A=0,M=1})
a = 1, #E(Y_{A=1,M=1})
mval = list(1), # for the CDE, set mediator to M=1
estimation = "imputation",
inference = "bootstrap",
nboot = 2)

summary(res_msm_RD)
# Causal Mediation Analysis
# # Outcome regression:
# Call:
# glm(formula = Y_death ~ A0_PM2.5 + M_diabetes + A0_PM2.5 * M_diabetes,
#      family = gaussian(), data = getCall(x$reg.output$yreg)$data,
#      weights = getCall(x$reg.output$yreg)$weights)
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)      # MSM coef
# (Intercept)   0.179321   0.005242  34.210 < 2e-16 *** # are the
# A0_PM2.5      0.060129   0.017253   3.485 0.000494 *** # same as
# M_diabetes    0.072397   0.009242   7.834 5.21e-15 *** # calculation
# A0_PM2.5:M_diabetes 0.030173   0.025895   1.165 0.243960      # by hand
#
# # Mediator regressions:
# Call:
# glm(formula = M_diabetes ~ A0_PM2.5, 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) -0.73432   0.02268 -32.384 < 2e-16 ***
# A0_PM2.5     0.51531   0.06422   8.024 1.02e-15 ***
#
# # Mediator regressions for weighting (denominator):
# Call:
# glm(formula = M_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env +
#      L1, family = binomial(), data = getCall(x$reg.output$umdenomreg[[1L]]$data,
#      weights = getCall(x$reg.output$umdenomreg[[1L]]$weights)
# Coefficients:

```

```

#               Estimate Std. Error z value Pr(>|z|)
# (Intercept) -1.36249    0.04783 -28.488 < 2e-16 ***
# A0_PM2.5     0.30994    0.06668  4.648 3.35e-06 ***
# L0_male      0.24661    0.04369  5.644 1.66e-08 ***
# L0_soc_env   0.30628    0.04650  6.587 4.50e-11 ***
# L1           0.86045    0.04493 19.152 < 2e-16 ***
#
# # Mediator regressions for weighting (nominator):
# Call:
# glm(formula = M_diabetes ~ A0_PM2.5, family = binomial(), data = getCall(x$reg.out)$data,
#      weights = getCall(x$reg.output$umnomreg[[1L]])$weights)
# Coefficients:
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept) -0.74205    0.02271 -32.680 <2e-16 ***
# A0_PM2.5     0.55288    0.06408  8.628 <2e-16 ***
#
# # Exposure regression for weighting:
# Call:
# glm(formula = A0_PM2.5 ~ L0_male + L0_soc_env, family = binomial(),
#      data = getCall(x$reg.output$ereg)$data, weights = getCall(x$reg.output$ereg)$weights)
# Coefficients:
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept) -2.73244    0.07425 -36.799 < 2e-16 ***
# L0_male      0.40580    0.06447  6.294 3.09e-10 ***
# L0_soc_env   0.64060    0.07350  8.716 < 2e-16 ***
#
# # Effect decomposition on the mean difference scale via the marginal structural model
# Direct counterfactual imputation estimation with
# bootstrap standard errors, percentile confidence intervals and p-values
#               Estimate Std.error 95% CIL 95% CIU P.val
# cde           9.030e-02 1.601e-02 6.094e-02 0.082 <2e-16 *** # same result
# rpnde         7.003e-02 1.600e-02 5.838e-02 0.080 <2e-16 ***
# rtnde         7.374e-02 1.597e-02 5.888e-02 0.080 <2e-16 ***
# rpnie         8.903e-03 9.159e-04 8.284e-03 0.010 <2e-16 ***
# rtnie         1.261e-02 9.523e-04 8.738e-03 0.010 <2e-16 ***
# te            8.265e-02 1.505e-02 6.839e-02 0.089 <2e-16 ***
# rintref       -2.027e-02 7.341e-06 -2.571e-03 -0.003 <2e-16 ***
# rintmed       3.710e-03 3.634e-05 4.542e-04 0.001 <2e-16 ***
# cde(prop)     1.093e+00 2.940e-02 8.907e-01 0.930 <2e-16 ***
# rintref(prop) -2.452e-01 6.289e-03 -3.752e-02 -0.029 <2e-16 ***
# rintmed(prop) 4.489e-02 1.662e-03 5.140e-03 0.007 <2e-16 ***
# rpnie(prop)   1.077e-01 3.403e-02 9.376e-02 0.139 <2e-16 ***
# rpm           1.526e-01 3.569e-02 9.890e-02 0.147 <2e-16 ***
# rint          -2.004e-01 4.627e-03 -3.014e-02 -0.024 <2e-16 ***
# rpe           -9.263e-02 2.940e-02 6.983e-02 0.109 <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)

```

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 $\bar{Q} = \mathbb{E}(Y \mid A, M, L(0), L(1))$
2. Use this estimate to predict an outcome for each subject under the counterfactual scenarios $\hat{\bar{Q}}(A = 0, M = m, L(0), L(1))_i$ and $\hat{\bar{Q}}(A = 1, M = m, L(0), L(1))_i$, by evaluating the regression fit \bar{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 $\text{do}(A = 0, M = 0)$, $\text{do}(A = 1, M = 0)$, $\text{do}(A = 0, M = 1)$ and $\text{do}(A = 1, M = 1)$.
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 $\mathbb{E}(Y_{A=0,M=0} \mid L(0), L(1)) = \hat{Q}(A = 0, M = 0, L(0), L(1))$;

- 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 $\mathbb{E}(Y_{A=1,M=0} \mid L(0), L(1)) = \hat{Q}(A = 1, M = 0, L(0), L(1))$;
- 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 $\mathbb{E}(Y_{A=0,M=1} \mid L(0), L(1)) = \hat{Q}(A = 0, M = 1, L(0), L(1))$;
- 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 $\mathbb{E}(Y_{A=1,M=1} \mid L(0), L(1)) = \hat{Q}(A = 1, M = 1, L(0), L(1))$;

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.

```
rm(list=ls())
df1_int <- read.csv(file = "data/df1_int.csv")

### MSM of CDE, estimated by G-computation -----
## 1. Estimate Qbar(A,M,L0,L1)
Q.cde.death <- glm(Y_death ~ A0_PM2.5 + M_diabetes + A0_PM2.5:M_diabetes +
                  L0_male + L0_soc_env + L1,
                  family = "gaussian", data = df1_int)
# The final result would be slightly 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.AOM0 <- data.A1M0 <- data.AOM1 <- data.A1M1 <- df1_int
data.AOM0$A0_PM2.5 <- 0
data.AOM0$M_diabetes <- 0

data.A1M0$A0_PM2.5 <- 1
data.A1M0$M_diabetes <- 0

data.AOM1$A0_PM2.5 <- 0
data.AOM1$M_diabetes <- 1

data.A1M1$A0_PM2.5 <- 1
data.A1M1$M_diabetes <- 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_PM2.5 + M_diabetes + A0_PM2.5:M_diabetes,
  family = "gaussian", # gaussian family for risk differences
  data = data.4scenarios)

coef(MSM.CDE.gcomp)
# (Intercept)          A0_PM2.5          M_diabetes A0_PM2.5:M_diabetes
# 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_PM2.5"]
# 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_PM2.5"] +
  coef(MSM.CDE.gcomp) ["A0_PM2.5:M_diabetes"])
# 0.07918291

# Note: Applying a binomial family for the first Qbar model would result in two
# slightly 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:

```

rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")

### MSM of CDE, estimated by G-computation (by ICE) -----
## 1a) Regress the outcome on  $L_0$ ,  $A$ ,  $L_1$  and  $M$  (and the  $A*M$  interaction if appropriate)

```

```

Y.death.model <- glm(Y_death ~ L0_male + L0_soc_env + A0_PM2.5 + L1 +
                     M_diabetes + A0_PM2.5:M_diabetes,
                     family = "binomial", data = df2_int)

## 1b) Generate predicted values by evaluating the regression
##      under the 4 counterfactual scenarios
data.AOM0 <- data.A1M0 <- data.AOM1 <- data.A1M1 <- df2_int
data.AOM0$A0_PM2.5 <- 0
data.AOM0$M_diabetes <- 0

data.A1M0$A0_PM2.5 <- 1
data.A1M0$M_diabetes <- 0

data.AOM1$A0_PM2.5 <- 0
data.AOM1$M_diabetes <- 1

data.A1M1$A0_PM2.5 <- 1
data.A1M1$M_diabetes <- 1

Q.Y.death.AOM0 <- predict(Y.death.model, newdata = data.AOM0, type = "response")
Q.Y.death.A1M0 <- predict(Y.death.model, newdata = data.A1M0, type = "response")
Q.Y.death.AOM1 <- predict(Y.death.model, newdata = data.AOM1, 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.AOM0.model <- glm(Q.Y.death.AOM0 ~ L0_male + L0_soc_env + A0_PM2.5,
                           family = "quasibinomial", data = df2_int)
L1.death.A1M0.model <- glm(Q.Y.death.A1M0 ~ L0_male + L0_soc_env + A0_PM2.5,
                           family = "quasibinomial", data = df2_int)
L1.death.AOM1.model <- glm(Q.Y.death.AOM1 ~ L0_male + L0_soc_env + A0_PM2.5,
                           family = "quasibinomial", data = df2_int)
L1.death.A1M1.model <- glm(Q.Y.death.A1M1 ~ L0_male + L0_soc_env + A0_PM2.5,
                           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.AOM0$Yam.death.pred <- predict(L1.death.AOM0.model,
                                    newdata = data.AOM0, type = "response")
data.A1M0$Yam.death.pred <- predict(L1.death.A1M0.model,
                                    newdata = data.A1M0, type = "response")
data.AOM1$Yam.death.pred <- predict(L1.death.AOM1.model,
                                    newdata = data.AOM1, type = "response")
data.A1M1$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_PM2.5 + M_diabetes + A0_PM2.5:M_diabetes,
  family = "gaussian", # gaussian family for risk differences
  data = data.4scenarios)
coef(MSM.CDE.gcomp)
# (Intercept)          A0_PM2.5          M_diabetes A0_PM2.5:M_diabetes
# 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_PM2.5"]
# 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_PM2.5"] +
  coef(MSM.CDE.gcomp)["A0_PM2.5:M_diabetes"])
# 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_soc_env + A0_PM2.5",
  Y_death="Q.kplus1 ~ L0_male + L0_soc_env + L1 +
  A0_PM2.5 * M_diabetes")
gform <- c("A0_PM2.5 ~ L0_male + L0_soc_env",
  "M_diabetes ~ L0_male + L0_soc_env + A0_PM2.5 + 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_soc_env,
  A0_PM2.5, L1,
  M_diabetes, Y_death))
CDE_ltmle_M0 <- ltmle(data = data_binary,
  Anodes = c("A0_PM2.5", "M_diabetes"),
  Lnodes = c("L1"), # intermediate confounders +/- baseline
  Ynodes = c("Y_death"),
  survivalOutcome = FALSE, # TRUE for time-to-event outcomes Y
  Qform = Qform,

```

```

gform = gform,
abarc = list(c(1,0), # counterfactual intervention do(A=1,M=0)
            c(0,0)), # counterfactual intervention do(A=0,M=0)
SL.library = NULL, # calls 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_PM2.5", "M_diabetes"),
  Lnodes = c("L1"), # intermediate confounders +/- baseline
  Ynodes = c("Y_death"),
  survivalOutcome = FALSE, # TRUE for time-to-event outcomes Y
  Qform = Qform,
  gform = gform,
  abarc = list(c(1,1), # counterfactual intervention do(A=1,M=0)
              c(0,1)), # counterfactual intervention do(A=0,M=0)
  SL.library = NULL, # calls 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 $\text{PNDE} = \mathbb{E}(Y_{a,M_{a^*}}) - \mathbb{E}(Y_{a^*,M_{a^*}})$ and the (Total) Natural Indirect Effect is defined by $\text{TNIE} = \mathbb{E}(Y_{a,M_a}) - \mathbb{E}(Y_{a,M_{a^*}})$.

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

$$\mathbb{E}(Y_{a,m} \mid l(0), l(1)) = \alpha_0 + \alpha_A a + \alpha_M m + \alpha_{AM} a \times m + \alpha_{L(0)} l(0) + \alpha_{L(1)} l(1) \quad (8.7)$$

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

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

(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

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

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

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

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

Marginal Natural Direct and Indirect effect can then be obtained:

$$\begin{aligned} \text{PNDE} &= \sum_{l(0), l(1)} [\text{PNDE} \mid L(0) = l(0), L(1) = l(1)] \times P(L(0) = l(0), L(1) = l(1)) \\ \text{TNIE} &= \sum_{l(0), l(1)} [\text{TNIE} \mid L(0) = l(0), L(1) = l(1)] \times P(L(0) = l(0), L(1) = l(1)) \end{aligned}$$

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):

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

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

$$sw_{A,i} = \frac{P(A = a_i)}{P(A = a_i \mid L(0) = l(0)_i)} \quad \text{or} \quad sw_{A,i} = \frac{P(A = a_i \mid L(0) = l(0)_i)}{P(A = a_i \mid L(0) = l(0)_i)} = 1$$

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

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

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_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env,
               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$M_diabetes==1] <- g.Mis1.AL0[df1_int$M_diabetes==1]
sw_M.num[df1_int$M_diabetes==0] <- (1 - g.Mis1.AL0[df1_int$M_diabetes==0])

g.M.AL0L1 <- glm(M_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env + 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$M_diabetes==1] <- g.Mis1.AL0L1[df1_int$M_diabetes==1]
sw_M.denom[df1_int$M_diabetes==0] <- (1 - g.Mis1.AL0L1[df1_int$M_diabetes==0])
```

```

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

## 2. Estimate coefficients of the MSM1
MSM1 <- glm(Y_death ~ A0_PM2.5 + M_diabetes + A0_PM2.5:M_diabetes +
            L0_male + L0_soc_env,
            weights = sw_msm1,
            family = "gaussian",
            data = df1_int)
coef(MSM1)
# (Intercept)      A0_PM2.5  M_diabetes    L0_male L0_soc_env  A0_PM2.5:M_diabetes
# 0.12033221  0.06381257  0.06691712  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_PM2.5 ~ 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$A0_PM2.5==1] <- g.Ais1[df1_int$A0_PM2.5==1]
sw_msm2.num[df1_int$A0_PM2.5==0] <- (1 - g.Ais1[df1_int$A0_PM2.5==0])

# denominator
g.A.L0 <- glm(A0_PM2.5 ~ L0_male + L0_soc_env,
              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$A0_PM2.5==1] <- g.Ais1.L0[df1_int$A0_PM2.5==1]
sw_msm2.denom[df1_int$A0_PM2.5==0] <- (1 - g.Ais1.L0[df1_int$A0_PM2.5==0])

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

## 3. Estimate coefficients of the MSM2
MSM2 <- glm(M_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env,
            weights = sw_msm2,
            family = "binomial",
            data = df1_int)
coef(MSM2)
# (Intercept)      A0_PM2.5    L0_male  L0_soc_env
# -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_PM2.5"] * 0 +

```

```

coef(MSM2)["L0_male"] * df1_int$L0_male +
coef(MSM2)["L0_soc_env"] * df1_int$L0_soc_env)

# PNDE conditional on L(0)
PNDE_L0 <- (1 - 0) * (coef(MSM1)["A0_PM2.5"] +
                      coef(MSM1)["A0_PM2.5:M_diabetes"] * 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_PM2.5"] * 1 +
                      coef(MSM2)["L0_male"] * df1_int$L0_male +
                      coef(MSM2)["L0_soc_env"] * df1_int$L0_soc_env)

# TNIE conditional on L(0)
TNIE_L0 <- (g.minus1.A1 - g.minus1.A0) * (coef(MSM1)["M_diabetes"] +
                                           coef(MSM1)["A0_PM2.5:M_diabetes"] * 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.

8.4 MSM estimated by the CMAverse package

8.4.1 Calculation by hand, using the “mediation formula”

The counterfactual quantity $\mathbb{E}(Y_{a,M_{a'}})$ used to defined the Natural Direct and Indirect effects can be expressed by the *mediation formula* (conditional on $\{L(0), L(1)\}$, or the population average):

$$\mathbb{E}(Y_{a,M_{a'}} | L(0), L(1)) = \sum_m \mathbb{E}(Y_{a,m} | L(0), L(1)) \times \mathbb{P}(M_{a'} = m | L(0), L(1))$$

or

$$\mathbb{E}(Y_{a,M_{a'}}) = \sum_m \mathbb{E}(Y_{a,m}) \times \mathbb{P}(M_{a'} = m)$$

In order to estimate the $\text{PNDE} = \mathbb{E}(Y_{1,M_0}) - \mathbb{E}(Y_{0,M_0})$ and the $\text{TNIE} = \mathbb{E}(Y_{1,M_1}) - \mathbb{E}(Y_{1,M_0})$ we can estimate 2 Marginal Structural Models:

- a model of the counterfactual mediator (where g is a link function), for example:

$$g[\mathbb{P}(M_{a'} = 1)] = \beta_0 + \beta_A a'$$

- a model of the counterfactual outcome (where h is a link function), for example:

$$h[\mathbb{E}(Y_{a,m})] = \alpha_0 + \alpha_A a + \alpha_M m + \alpha_{A*M} a \times m$$

Below, we will estimate the coefficients of the 2 MSMs by Inverse Probability of Treatment Weighting (IPTW).

```
rm(list=ls())
df1_int <- read.csv(file = "data/df1_int.csv")

## 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_PM2.5 ~ L0_male + L0_soc_env + L1,
             family = "binomial", data = df1_int)
summary(g.A.L)
# Coefficients:
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept) -2.74236     0.07729  -35.484 < 2e-16 ***
# L0_male      0.40610     0.06448   6.298 3.01e-10 ***
# L0_soc_env   0.64079     0.07350   8.718 < 2e-16 ***
# L1           0.03257     0.06968   0.467  0.64

# 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$A0_PM2.5==1] <- predict(g.A.L, type="response")[df1_int$A0_PM2.5==1]
gAi.L[df1_int$A0_PM2.5==0] <- (1 - predict(g.A.L, type="response"))[df1_int$A0_PM2.5==0]

# 1c. Estimate  $g(A=a_i)$  (numerator of the weight)
g.A <- glm(A0_PM2.5 ~ 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$A0_PM2.5==1] <- predict(g.A, type="response")[df1_int$A0_PM2.5==1]
gAi[df1_int$A0_PM2.5==0] <- (1 - predict(g.A, type="response"))[df1_int$A0_PM2.5==0]

# 1e. Calculate  $sw_{\{A,i\}}$ 
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_diabetes ~ L0_male + L0_soc_env + A0_PM2.5 + L1,
             family = "binomial", data = df1_int)
summary(g.M.L)
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept) -1.37880    0.04872 -28.303 < 2e-16 ***
# L0_male      0.25861    0.04437   5.829 5.57e-09 ***
# L0_soc_env   0.33050    0.04744   6.967 3.23e-12 ***
# A0_PM2.5     0.56260    0.06555   8.583 < 2e-16 ***
# L1           0.33462    0.04744   7.054 1.74e-12 ***

# 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$M_diabetes==1] <- predict(g.M.L, type="response")[df1_int$M_diabetes==1]
gMi.L[df1_int$M_diabetes==0] <- (1 - predict(g.M.L, type="response"))[df1_int$M_diabetes==0]

# 2c. Estimate  $g(M=m_i/A)$  (numerator of the weight)
g.M.A <- glm(M_diabetes ~ A0_PM2.5, family = "binomial", data = df1_int)
summary(g.M.A)
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept) -0.93236    0.02358 -39.544 <2e-16 ***
# A0_PM2.5     0.62388    0.06481   9.627 <2e-16 ***

# 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$M_diabetes==1] <- predict(g.M.A, type="response")[df1_int$M_diabetes==1]
gMi.A[df1_int$M_diabetes==0] <- (1 - predict(g.M.A, type="response"))[df1_int$M_diabetes==0]
# 2e. Calculate  $sw_{\{M,i\}}$ 
sw_Mi <- gMi.A / gMi.L

## 3. Estimate marginal model for the counterfactual outcome  $Y_{\{am\}}$ 
model.Yam.death <- glm(Y_death ~ A0_PM2.5 * M_diabetes,
                      weights = sw_Ai * sw_Mi,
                      family = "gaussian",
                      data = df1_int)
summary(model.Yam.death)
# Coefficients:
#               Estimate Std. Error t value Pr(>|t|)
# (Intercept)    0.178967    0.005049  35.445 < 2e-16 ***
# A0_PM2.5       0.067151    0.016679   4.026 5.72e-05 ***
# M_diabetes     0.067321    0.009508   7.080 1.54e-12 ***
# A0_PM2.5:M_diabetes -0.004491    0.026027  -0.173  0.863

## 4) Estimate marginal model for the counterfactual mediator  $P(M_{a*} = 1)$ 

```

```

model.Ma <- glm(M_diabetes ~ A0_PM2.5,
               weights = sw_Ai,
               family = "binomial",
               data = df1_int)
summary(model.Ma)
# Coefficients:
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept) -0.92405     0.02353 -39.264  <2e-16 ***
# A0_PM2.5     0.58413     0.06501   8.986  <2e-16 ***

## 5) Estimate population average counterfactuals using the "mediation formula"
##  $E(Y_{\{a,M_a^*\}}) = \sum_m E(Y_{\{am\}}) * P(M_a^* = m)$ 
E.Y0M0 <- ((coef(model.Yam.death)["(Intercept)"] + #  $\sum_m E(Y_{\{a0\}}) * P(M_a^* = 0)$ 
           coef(model.Yam.death)["A0_PM2.5"] * 0 +
           coef(model.Yam.death)["M_diabetes"] * 0 +
           coef(model.Yam.death)["A0_PM2.5:M_diabetes"] * 0 * 0) *
           (1 - plogis(coef(model.Ma)["(Intercept)"] +
                       coef(model.Ma)["A0_PM2.5"] * 0))) +
           ((coef(model.Yam.death)["(Intercept)"] + #  $\sum_m E(Y_{\{a1\}}) * P(M_a^* = 1)$ 
           coef(model.Yam.death)["A0_PM2.5"] * 0 +
           coef(model.Yam.death)["M_diabetes"] * 1 +
           coef(model.Yam.death)["A0_PM2.5:M_diabetes"] * 0 * 1) *
           (plogis(coef(model.Ma)["(Intercept)"] +
                    coef(model.Ma)["A0_PM2.5"] * 0)))

E.Y1M0 <- ((coef(model.Yam.death)["(Intercept)"] + #  $\sum_m E(Y_{\{a,M=0\}}) * P(M_0 = 0)$ 
           coef(model.Yam.death)["A0_PM2.5"] * 1 +
           coef(model.Yam.death)["M_diabetes"] * 0 +
           coef(model.Yam.death)["A0_PM2.5:M_diabetes"] * 1 * 0) *
           (1 - plogis(coef(model.Ma)["(Intercept)"] +
                       coef(model.Ma)["A0_PM2.5"] * 0))) +
           ((coef(model.Yam.death)["(Intercept)"] + #  $\sum_m E(Y_{\{a,M=1\}}) * P(M_0 = 1)$ 
           coef(model.Yam.death)["A0_PM2.5"] * 1 +
           coef(model.Yam.death)["M_diabetes"] * 1 +
           coef(model.Yam.death)["A0_PM2.5:M_diabetes"] * 1 * 1) *
           (plogis(coef(model.Ma)["(Intercept)"] +
                    coef(model.Ma)["A0_PM2.5"] * 0)))

E.Y1M1 <- ((coef(model.Yam.death)["(Intercept)"] + #  $E(Y_{\{a,M=0\}}) * P(M_1 = 0)$ 
           coef(model.Yam.death)["A0_PM2.5"] * 1 +
           coef(model.Yam.death)["M_diabetes"] * 0 +
           coef(model.Yam.death)["A0_PM2.5:M_diabetes"] * 1 * 0) *
           (1 - plogis(coef(model.Ma)["(Intercept)"] +
                       coef(model.Ma)["A0_PM2.5"] * 1))) +
           ((coef(model.Yam.death)["(Intercept)"] + #  $E(Y_{\{a,M=1\}}) * P(M_1 = 1)$ 

```

```

coef(model.Yam.death)["A0_PM2.5"] * 1 +
coef(model.Yam.death)["M_diabetes"] * 1 +
coef(model.Yam.death)["A0_PM2.5:M_diabetes"] * 1 * 1) *
(plogis(coef(model.Ma)["(Intercept)"] +
        coef(model.Ma)["A0_PM2.5"] * 1)))

PNDE.death <- E.Y1M0 - E.Y0M0
# 0.06587481
TNIE.death <- E.Y1M1 - E.Y1M0
# 0.008274351

```

In the example above, we directly used the estimated coefficients of the MSMs to calculate the quantities $\mathbb{E}(Y_{a,M_{a'}})$ for the PNDE and the TNIE. When using the CMAverse, individual values for the counterfactual mediator are simulated as shown below. This strategy can be useful if the mediator is continuous of high-dimensional.

```

## 6) Estimate population average counterfactuals using simulated
##      values of the mediator
PNDE.death.sim <- rep(NA,5)
TNIE.death.sim <- rep(NA,5)

set.seed(1234)
for(k in 1:15){
  M0 <- rbinom(n = nrow(df1_int), size = 1,
               prob = plogis(coef(model.Ma)["(Intercept)"] +
                             coef(model.Ma)["A0_PM2.5"] * 0))
  M1 <- rbinom(n = nrow(df1_int), size = 1,
               prob = plogis(coef(model.Ma)["(Intercept)"] +
                             coef(model.Ma)["A0_PM2.5"] * 1))
  E.Y0M0 <- mean((coef(model.Yam.death)["(Intercept)"] +
                  coef(model.Yam.death)["A0_PM2.5"] * 0 +
                  coef(model.Yam.death)["M_diabetes"] * M0 +
                  coef(model.Yam.death)["A0_PM2.5:M_diabetes"] * 0 * M0))
  E.Y1M0 <- mean((coef(model.Yam.death)["(Intercept)"] +
                  coef(model.Yam.death)["A0_PM2.5"] * 1 +
                  coef(model.Yam.death)["M_diabetes"] * M0 +
                  coef(model.Yam.death)["A0_PM2.5:M_diabetes"] * 1 * M0))
  E.Y1M1 <- mean((coef(model.Yam.death)["(Intercept)"] +
                  coef(model.Yam.death)["A0_PM2.5"] * 1 +
                  coef(model.Yam.death)["M_diabetes"] * M1 +
                  coef(model.Yam.death)["A0_PM2.5:M_diabetes"] * 1 * M1))
  PNDE.death.sim[k] <- E.Y1M0 - E.Y0M0
  TNIE.death.sim[k] <- E.Y1M1 - E.Y1M0
}

```

```
mean(PNDE.death.sim)
# [1] 0.06587304
mean(TNIE.death.sim)
# [1] 0.00824998
```

8.4.2 Estimation of NDE and NIE using CMAverse (by IPTW or gcomputation)

We can use the `CMAverse` package to estimate the coefficients of the 2 MSMs required to estimate Natural Direct and Indirect Effects.

For estimation by IPTW, we have to specify the following arguments `model = msm` (to estimate the MSM of the outcome and the MSM of the mediator), and that the counterfactual mediator values are simulated (`estimation = "imputation"`). Because the coefficients of the MSMs are estimated by IPTW, we have to specify

- the link functions for the MSM of the outcome (`yreg`) and the MSM of the mediator (`mreg`)
- and the link function to calculate the weights `ereg` for $\mathbb{P}(A = 1|L(0))$, `wmnomreg` for the numerator of the mediator's weight and `wmdenomreg` for the denominator of the mediator's weight.

```
## Using the CMAverse to estimate MSM estimated by IPTW
set.seed(1234) # note: there is randomness, even with estimation by IPTW
#           => counterfactuals are imputed (estimation = "imputation")
res_msm_df1 <- cmest(data = df1_int,
  model = "msm",
  outcome = "Y_death",
  exposure = "A0_PM2.5",
  mediator = "M_diabetes",
  basec = c("L0_male", "L0_soc_env", "L1"),
  postc = NULL,
  EMint = TRUE, # E*M interaction
  ereg = "logistic", # exposure regression model g(A=1/L(0))
  yreg = "linear", # to get risk difference
  mreg = list("logistic"), # mediation model g(M=1/L1,A,L0)
  wmnomreg = list("logistic"), # g(M=1/A) wgt nominator
  wmdenomreg = list("logistic"), # g(M=1/L1,A,L(0)) wgt denom
  astar = 0, # E(Y_{A=0,M=1})
  a = 1, # E(Y_{A=1,M=1})
  mval = list(0), # mediator value at which the variable is controlled
  estimation = "imputation",
  inference = "bootstrap",
```

```

nboot = 2)
summary(res_msm_df1)
# Causal Mediation Analysis
#
# # Outcome regression:
# Call:
# glm(formula = Y_death ~ A0_PM2.5 + M_diabetes + A0_PM2.5 * M_diabetes,
#      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)    0.178967   0.005049  35.445 < 2e-16 ***
# A0_PM2.5       0.067151   0.016679   4.026 5.72e-05 ***
# M_diabetes     0.067321   0.009508   7.080 1.54e-12 ***
# A0_PM2.5:M_diabetes -0.004491  0.026027  -0.173  0.863
#
# # Mediator regressions:
# Call:
# glm(formula = M_diabetes ~ A0_PM2.5, 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) -0.92405    0.02353 -39.264 <2e-16 ***
# A0_PM2.5     0.58413    0.06501   8.986 <2e-16 ***
#
# # Mediator regressions for weighting (denominator):
# Call:
# glm(formula = M_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env +
#      L1, family = binomial(), data = getCall(x$reg.output$wmdenomreg[[1L]])$data,
#      weights = getCall(x$reg.output$wmdenomreg[[1L]])$weights)
# Coefficients:
#
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept) -1.37880    0.04872 -28.303 < 2e-16 ***
# A0_PM2.5     0.56260    0.06555   8.583 < 2e-16 ***
# L0_male      0.25861    0.04437   5.829 5.57e-09 ***
# L0_soc_env   0.33050    0.04744   6.967 3.23e-12 ***
# L1           0.33462    0.04744   7.054 1.74e-12 ***
#
# # Mediator regressions for weighting (nominator):
# Call:
# glm(formula = M_diabetes ~ A0_PM2.5, family = binomial(),
#      data = getCall(x$reg.output$wnomreg[[1L]])$data,
#      weights = getCall(x$reg.output$wnomreg[[1L]])$weights)
# Coefficients:

```

```

#           Estimate Std. Error z value Pr(>|z|)
# (Intercept) -0.93236    0.02358 -39.544  <2e-16 ***
# A0_PM2.5     0.62388    0.06481   9.627  <2e-16 ***
#
# # Exposure regression for weighting:
# Call:
# glm(formula = A0_PM2.5 ~ L0_male + L0_soc_env + L1, family = binomial(),
#      data = getCall(x$reg.output$ereg)$data,
#      weights = getCall(x$reg.output$ereg)$weights)
# Coefficients:
#           Estimate Std. Error z value Pr(>|z|)
# (Intercept) -2.74236    0.07729 -35.484  < 2e-16 ***
# L0_male      0.40610    0.06448   6.298 3.01e-10 ***
# L0_soc_env   0.64079    0.07350   8.718  < 2e-16 ***
# L1           0.03257    0.06968   0.467    0.64
#
# # Effect decomposition on the mean difference scale via the marginal structural model
# Direct counterfactual imputation estimation with
# bootstrap standard errors, percentile confidence intervals and p-values
#
#           Estimate Std.error    95% CIL 95% CIU P.val
# cde      0.0671509  0.0157808  0.0761499  0.097 <2e-16 ***
# pnnde    0.0658727  0.0169713  0.0643983  0.087 <2e-16 *** <- PNDE
# tnnde    0.0652889  0.0172698  0.0594343  0.083 <2e-16 ***
# pnnde    0.0087514  0.0013389  0.0070189  0.009 <2e-16 ***
# tnnde    0.0081675  0.0010405  0.0024559  0.004 <2e-16 *** <- TNDE
# te       0.0740402  0.0159309  0.0682520  0.090 <2e-16 ***
# intref   -0.0012782  0.0011906 -0.0117517 -0.010 <2e-16 ***
# intmed   -0.0005839  0.0002984 -0.0049640 -0.005 <2e-16 ***
# cde(prop) 0.9069522  0.0222813  1.0860362  1.116 <2e-16 ***
# intref(prop) -0.0172642  0.0439631 -0.1726807 -0.114 <2e-16 ***
# intmed(prop) -0.0078856  0.0162851 -0.0729153 -0.051 <2e-16 ***
# pnnde(prop) 0.1181976  0.0379668  0.0786163  0.130 <2e-16 ***
# pm       0.1103120  0.0216818  0.0275800  0.057 <2e-16 ***
# int      -0.0251498  0.0602482 -0.2455960 -0.165 <2e-16 ***
# pe       0.0930478  0.0222813 -0.1159712 -0.086 <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# (cde: controlled direct effect;
# pnnde: pure natural direct effect;
# tnnde: total natural direct effect;
# pnnde: pure natural indirect effect;
# tnnde: total natural indirect effect;
# te: total effect;

```

```
# intref: reference interaction;
# intmed: mediated interaction;
# cde(prop): proportion cde;
# intref(prop): proportion intref;
# intmed(prop): proportion intmed;
# pnle(prop): proportion pnle;
# pm: overall proportion mediated;
# int: overall proportion attributable to interaction;
# pe: overall proportion eliminated)
```

The CMAverse can also estimate the coefficients of the 2 MSM by (parametric) g-computation. For this estimation, we have to specify the following arguments `model = gformula`, and that the counterfactual outcome and mediator values are simulated (`estimation = "imputation"`). We don't need to specify the `ereg`, `wmnomreg` and `wmdenomreg` arguments which were used to calculate weights in the previous approach.

Note that the estimated models of the outcome and the mediator are conditional on baseline confounders $L(0)$ and $L(1)$.

```
## Using the CMAverse to estimate MSM estimated by g-comp
set.seed(1234) # note: there is randomness, even with estimation by IPTW
#           => counterfactuals are imputed (estimation = "imputation")
res_msm_df1.qol <- cmest(data = df1_int,
  model = "gformula",
  outcome = "Y_qol",
  exposure = "A0_PM2.5",
  mediator = "M_diabetes",
  basec = c("L0_male", "L0_soc_env", "L1"),
  postc = NULL,
  EMint = TRUE, # E*M interaction
  # ereg = "logistic", # # not needed with gcomp
  yreg = "linear", # to get risk difference
  mreg = list("logistic"), # mediation model g(M=1/L1,A,L0)
  # wmnomreg = list("logistic"), # not needed with gcomp
  # wmdenomreg = list("logistic"), ## not needed with gcomp
  astar = 0, #E(Y_{A=0,M=1})
  a = 1, #E(Y_{A=1,M=1})
  mval = list(0), # mediator value at which the variable is control
  estimation = "imputation",
  inference = "bootstrap",
  nboot = 2)
summary(res_msm_df1.qol)
# Causal Mediation Analysis
#
```



```

# # Outcome regression:
# Call:
#   glm(formula = Y_qol ~ A0_PM2.5 + M_diabetes + A0_PM2.5 * M_diabetes +
#     L0_male + L0_soc_env + 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)    74.7669    0.2139  349.557 < 2e-16 ***
# A0_PM2.5       -3.7153    0.4160  -8.931 < 2e-16 ***
# M_diabetes      -8.6317    0.2385 -36.197 < 2e-16 ***
# L0_male         -0.7235    0.2018  -3.586 0.000337 ***
# L0_soc_env      -2.8899    0.2112 -13.684 < 2e-16 ***
# L1              -3.4280    0.2212 -15.494 < 2e-16 ***
# A0_PM2.5:M_diabetes -5.6154    0.6514  -8.621 < 2e-16 ***
#
# # Mediator regressions:
# Call:
#   glm(formula = M_diabetes ~ A0_PM2.5 + L0_male + L0_soc_env +
#     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) -1.37880    0.04872 -28.303 < 2e-16 ***
# A0_PM2.5     0.56260    0.06555   8.583 < 2e-16 ***
# L0_male      0.25861    0.04437   5.829 5.57e-09 ***
# L0_soc_env   0.33050    0.04744   6.967 3.23e-12 ***
# L1           0.33462    0.04744   7.054 1.74e-12 ***
#
# # 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          -3.715265  0.114707 -3.084527 -2.930 <2e-16 ***
# pnnde         -5.103390  0.008919 -4.820137 -4.808 <2e-16 ***
# tnnde         -5.659875  0.006095 -5.579563 -5.571 <2e-16 ***
# pnne          -0.855400  0.040149 -1.006846 -0.953 <2e-16 ***
# tnne          -1.411886  0.025135 -1.758083 -1.724 <2e-16 ***
# te            -6.515276  0.034054 -6.578220 -6.532 <2e-16 ***
# intref        -1.388125  0.105788 -1.877736 -1.736 <2e-16 ***
# intmed        -0.556485  0.015014 -0.771408 -0.751 <2e-16 ***
# cde(prop)      0.570239  0.015115  0.448589  0.469 <2e-16 ***
# intref(prop)   0.213057  0.017570  0.263846  0.287 <2e-16 ***
# intmed(prop)   0.085412  0.002894  0.114201  0.118 <2e-16 ***

```

```

#   pnie(prop)      0.131292  0.005348  0.145871  0.153 <2e-16 ***
#   pm              0.216704  0.002454  0.263960  0.267 <2e-16 ***
#   int             0.298469  0.020463  0.378048  0.406 <2e-16 ***
#   pe              0.429761  0.015115  0.531104  0.551 <2e-16 ***
#   ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# (cde: controlled direct effect;
# pnde: pure natural direct effect;
# tn timer: total natural direct effect;
# pnie: pure natural indirect effect;
# tn timer: total natural indirect effect;
# te: total effect; intref:
# reference interaction;
# intmed: mediated interaction;
# cde(prop): proportion cde;
# intref(prop): proportion intref;
# intmed(prop): proportion intmed;
# pnie(prop): proportion pnie;
# pm: overall proportion mediated;
# int: overall proportion attributable to interaction;
# pe: overall proportion eliminated)

```

8.5 “Unified” approach for estimating Natural Direct and Indirect effects

(Lange, Vansteelandt, and Bekaert 2012) suggested to use a “unified” approach for estimating Natural Direct and Indirect effects (when they are identifiable), based on the following Marginal Structural Model:

$$h \left[\mathbb{E}(Y_{a, M_{a^*}}) \right] = \gamma_0 + \gamma_1 a + \gamma_2 a^* + \gamma_3 (a \times a^*) \text{ where } h \text{ is a link function}$$

So that we can express the PNDE as:

$$\begin{aligned}
 \text{PNDE} &= \mathbb{E}(Y_{1, M_0}) - \mathbb{E}(Y_{0, M_0}) \\
 \text{PNDE} &= h^{-1}(\gamma_0 + \gamma_1) - h^{-1}(\gamma_0) \\
 \text{PNDE} &= \gamma_1 \text{ if the link function is the identity function}
 \end{aligned}$$

and we can express the TNIE as:

$$\begin{aligned}
 \text{TNIE} &= \mathbb{E}(Y_{1, M_1}) - \mathbb{E}(Y_{1, M_0}) \\
 \text{TNIE} &= h^{-1}(\gamma_0 + \gamma_1 + \gamma_2 + \gamma_3) - h^{-1}(\gamma_0 + \gamma_1) \\
 \text{TNIE} &= \gamma_2 + \gamma_3 \text{ if the link function is the identity function}
 \end{aligned}$$

Note that this model can also be extended to study effect modifications by baseline confounders.

8.5.1 Estimation of the MSM by IPTW

8.5.1.1 Manual calculation

The estimation of the “unified” MSM for Natural Direct and Indirect effects relies on the following steps:

1. Estimate suitable models for the exposure conditional on baseline confounders ($L(0)$ and $L(1)$), using the original data set, in order to compute weights for the exposure A
2. Estimate a suitable model for the mediator conditional on baseline confounders ($L(0)$ and $L(1)$), using the original data set, in order to compute weights for the mediator M
3. Construct a new data set by repeating the original data set twice, and including an additional variable A^* , equal to the original exposure in the 1st data set, and equal to $1 - A$ in the 2nd data set (for multicategorical exposures, see supplementary material of (Lange, Vansteelandt, and Bekaert 2012)).
4. Compute weights by applying the fitted models from steps 1 and 2

$$w_i = \frac{1}{P(A = A_i \mid L(0) = L(0)_i, L(1) = L(1)_i)} \times \frac{P(M = M_i \mid A = A_i^*, L(0) = L(0)_i, L(1) = L(1)_i)}{P(M = M_i \mid A = A_i, L(0) = L(0)_i, L(1) = L(1)_i)}$$

$$\text{or } sw_i = \frac{P(A = A_i)}{P(A = A_i \mid L(0) = L(0)_i, L(1) = L(1)_i)} \times \frac{P(M = M_i \mid A = A_i^*, L(0) = L(0)_i, L(1) = L(1)_i)}{P(M = M_i \mid A = A_i, L(0) = L(0)_i, L(1) = L(1)_i)}$$

5. Estimate the unified MSM as a weighted model of the outcome including only A and A^* (and their interaction), weighted by simple weights or stabilized weights. Use the estimated coefficients of the MSM to calculate the PNDE and TNIE

```
rm(list=ls())
df1_int <- read.csv(file = "data/df1_int.csv")

## Step 1. Estimate a suitable model for the exposure conditional on confounders,
##           using the original data set
# for A weight numerator
g.A <- glm(A0_PM2.5 ~ 1,
           family = "binomial", data = df1_int)

# for A weight denominator
```

```

g.A.L0 <- glm(A0_PM2.5 ~ L0_male + L0_soc_env + L1, # L1 not necessary in our example
              family = "binomial", data = df1_int)

## Step 2. Estimate a suitable model for the mediator conditional on confounders,
##          using the original data set
df1_int$A0_PM2.5_temp <- df1_int$A0_PM2.5
g.M.AL0 <- glm(M_diabetes ~ L0_male + L0_soc_env + A0_PM2.5_temp + L1,
              family = "binomial", data = df1_int)

## Step 3. Construct a new data set by repeating the original data set twice,
##          and including an additional variable A*, equal to the original exposure
##          in the 1st data set, and equal to 1-A in the 2nd data set
##          (note for multicategorical exposure, see supplementary material of Lange et

# create identifier
df1_int$id <- 1:nrow(df1_int)

# duplicate the original data set
df1_int.1 <- df1_int.2 <- df1_int

# Add an A* variable
df1_int.1$A0_PM2.5_star <- df1_int$A0_PM2.5 # A* = A in the 1st data set
df1_int.2$A0_PM2.5_star <- 1 - df1_int$A0_PM2.5 # A* = 1 - A in the 2d data set

# stack the 2 new datasets
df1_int.new <- rbind(df1_int.1, df1_int.2)
# sort the data set by the id-variable to use geeglm
# (necessary to use geepack and the geeglm() later)
df1_int.new <- df1_int.new[order(df1_int.new$id), ]

head(df1_int.new, 12)
#           L0_male L0_soc_env A0_PM2.5 L1 M_diabetes Y_death Y_qol A0_PM2.5_temp id A0_PM
# 1           0         1         0 1         0         0 93.4         0 1
# 10001        0         1         0 1         0         0 93.4         0 1
# 2           1         1         1 1         0         1 64.0         1 2
# 10002        1         1         1 1         0         1 64.0         1 2
# 3           1         1         0 0         0         0 75.6         0 3
# 10003        1         1         0 0         0         0 75.6         0 3
# 4           1         0         0 0         0         0 89.8         0 4
# 10004        1         0         0 0         0         0 89.8         0 4
# 5           1         1         0 0         0         0 77.2         0 5
# 10005        1         1         0 0         0         0 77.2         0 5
# 6           1         1         0 0         1         0 73.9         0 6
# 10006        1         1         0 0         1         0 73.9         0 6

```

```

## Step 4. Compute weights by applying the fitted models from steps 1 and 2
# weights for A
g.Ais1 <- predict(g.A, newdata = df1_int.new, type = "response")
sw_A.num <- rep(NA, nrow(df1_int.new))
sw_A.num[df1_int.new$A0_PM2.5==1] <- g.Ais1[df1_int.new$A0_PM2.5==1]
sw_A.num[df1_int.new$A0_PM2.5==0] <- (1 - g.Ais1[df1_int.new$A0_PM2.5==0])

g.Ais1.L0 <- predict(g.A.L0, newdata = df1_int.new, type = "response")
sw_A.denom <- rep(NA, nrow(df1_int.new))
sw_A.denom[df1_int.new$A0_PM2.5==1] <- g.Ais1.L0[df1_int.new$A0_PM2.5==1]
sw_A.denom[df1_int.new$A0_PM2.5==0] <- (1 - g.Ais1.L0[df1_int.new$A0_PM2.5==0])

w_A <- 1 / sw_A.denom
sw_A <- sw_A.num / sw_A.denom

# weights for M/A*
df1_int.new$A0_PM2.5_temp <- df1_int.new$A0_PM2.5_star
g.Mis1.Astar <- predict(g.M.AL0, newdata = df1_int.new, type = "response")
sw_M.Astar <- rep(NA, nrow(df1_int.new))
sw_M.Astar[df1_int.new$M_diabetes==1] <- g.Mis1.Astar[df1_int.new$M_diabetes==1]
sw_M.Astar[df1_int.new$M_diabetes==0] <- 1 - g.Mis1.Astar[df1_int.new$M_diabetes==0]

# weight for M/A
df1_int.new$A0_PM2.5_temp <- df1_int.new$A0_PM2.5
g.Mis1.A <- predict(g.M.AL0, newdata = df1_int.new, type = "response")
sw_M.A <- rep(NA, nrow(df1_int.new))
sw_M.A[df1_int.new$M_diabetes==1] <- g.Mis1.A[df1_int.new$M_diabetes==1]
sw_M.A[df1_int.new$M_diabetes==0] <- 1 - g.Mis1.A[df1_int.new$M_diabetes==0]

# non-stabilized weight
w <- w_A * (sw_M.Astar / sw_M.A)
# stabilized weight
sw <- sw_A * (sw_M.Astar / sw_M.A)

boxplot(data.frame(w, sw))

## Step 5. Fit a suitable model to the outcome including only A and A* (and their
##          interaction) using a weighted regression

# we will use a GEE model in order to get "robust" standard errors
library(geepack)
# note: be careful that individuals in the "df1_int.new" data set
#       and in the "w" vector have the same order

## for a risk difference in death

```

```

MSM.model.death <- geeglm(Y_death ~ A0_PM2.5 * A0_PM2.5_star,
                          weights = sw,
                          family = "gaussian",
                          id = df1_int.new$id,
                          data = df1_int.new,
                          scale.fix = TRUE)

summary(MSM.model.death)
# Coefficients:
#
#               Estimate Std.err Wald Pr(>|W|)
# (Intercept)    0.198840  0.004251 2187.61 < 2e-16 ***
# A0_PM2.5       0.065950  0.014539   20.58 5.7e-06 ***
# A0_PM2.5_star  0.008534  0.001242    47.25 6.2e-12 ***
# A0_PM2.5:A0_PM2.5_star -0.000406  0.003762    0.01 0.91

## Calculate PNDE and PNIE using the MSM's coefficients
unif.PNDE.death <- coef(MSM.model.death)["A0_PM2.5"]
# 0.066
unif.TNIE.death <- (coef(MSM.model.death)["A0_PM2.5_star"] +
                   coef(MSM.model.death)["A0_PM2.5:A0_PM2.5_star"])
# 0.00813

## For Quality of life
MSM.model.qol <- geeglm(Y_qol ~ A0_PM2.5 * A0_PM2.5_star,
                       weights = sw,
                       family = "gaussian",
                       id = df1_int.new$id,
                       data = df1_int.new,
                       scale.fix = TRUE)

summary(MSM.model.qol)
# Coefficients:
#
#               Estimate Std.err Wald Pr(>|W|)
# (Intercept)    69.0853  0.1174 346240.4 < 2e-16 ***
# A0_PM2.5      -5.3771  0.4010   179.8 < 2e-16 ***
# A0_PM2.5_star -1.0769  0.0311  1197.2 < 2e-16 ***
# A0_PM2.5:A0_PM2.5_star -0.6947  0.0909    58.4 2.2e-14 ***

## Calculate PNDE and PNIE using the MSM's coefficients
unif.PNDE.qol <- coef(MSM.model.qol)["A0_PM2.5"]
# -5.38
unif.TNIE.qol <- (coef(MSM.model.qol)["A0_PM2.5_star"] +
                 coef(MSM.model.qol)["A0_PM2.5:A0_PM2.5_star"])
# -1.77

# 95% confidence intervals can also be computed by bootstrap

```

8.5.1.2 Using the medflex package

We can use the `medflex` package to apply this approach, as described below.

The “unified” MSM can be defined as a model of the expected counterfactual outcome $\mathbb{E}(Y_{a,M_{a^*}} \mid L(0), L(1))$ conditional on the baseline confounders $L(0)$ and $L(1)$, or as a model of the marginal expected counterfactual outcome $\mathbb{E}(Y_{a,M_{a^*}})$.

For estimation by IPTW, the data is expanded using the `neWeight` function, and the MSM is estimated using the `neModel` function. Population average estimations are obtained using a weighted regression, where the weights are calculated using a model of the exposure, stated at the `xFit` argument inside the `neModel` function.

Standard errors can be computed by bootstrap or as robust “Sandwich” standard errors.

```
library(medflex)
# this package can used to estimate Natural direct and indirect effects when they are
# identifiable

### estimate PNDE and TNDE, conditional on L0 and L1
# ----- #
# get back to the original data set
df1_int <- subset(df1_int, select = -c(A0_PM2.5_temp, id))

## 1) Expand the dataset and calculate weights using the neWeight() function
g.M <- glm(M_diabetes ~ factor(A0_PM2.5) + # the 1st variable should be the exposure
          L0_male + L0_soc_env + L1, # then add baseline confounders
          family = "binomial", data = df1_int)

# expend the data set and add a second column A* = 1 - A
# and calculate the weights = sw_M.Astar / sw_M.A
exp.Data <- neWeight(g.M)

# it is also possible to use the neWeight function directly:
# exp.Data <- neWeight(M_diabetes ~ factor(A0_PM2.5) + # the 1st variable should be the exposure
#               L0_male + L0_soc_env + L1, # then add baseline confounders
#               family = "binomial", data = df1_int)

class(exp.Data)
# [1] "data.frame" "expData"      "weightData"
head(exp.Data)
#   id A0_PM2.50 A0_PM2.51 L0_male L0_soc_env L1 M_diabetes Y_death Y_qol
# 1  1         0         0         0         1  1         0         0  93.4
# 2  1         0         1         0         1  1         0         0  93.4
# 3  2         1         1         1         1  1         0         1  64.0
```

```

# 4 2      1      0      1      1 1      0      1 64.0
# 5 3      0      0      1      1 0      0      0 75.6
# 6 3      0      1      1      1 0      0      0 75.6

# the new variables A0_PM2.50 and A0_PM2.51 correspond to A and A*

# check the weights:
w <- weights(exp.Data)
boxplot(w)

# Note that we estimated the same weights by hand previously:
head(data.frame(w, sw_M.Astar / sw_M.A), 6)
#      w sw_M.Astar.sw_M.A
# 1 1.000      1.000
# 2 0.801      0.801
# 3 1.000      1.000
# 4 1.293      1.293
# 5 1.000      1.000
# 6 0.809      0.809
plot(w, sw_M.Astar / sw_M.A)
abline(0,1)

## 2) Fit the natural effect model using the neModel() function
##      (adjusted for baseline confounders)

## for death
set.seed(1234)
neMod.death <- neModel(Y_death ~ A0_PM2.50 * A0_PM2.51 + L0_male + L0_soc_env + L1,
  family = "gaussian", # to estimate risk difference
  expData = exp.Data,
  se = c("bootstrap"), # or "robust"
  nBoot = 10, # use >= 1000 samples, if se = bootstrap
)

summary(neMod.death)
#      Estimate Std. Error z value Pr(>|z|)
# (Intercept)      0.10982    0.00807   13.61 < 2e-16 ***
# A0_PM2.501        0.06576    0.01642    4.01 6.2e-05 ***
# A0_PM2.511        0.00836    0.00214    3.91 9.1e-05 ***
# L0_male           0.05041    0.00741    6.80 1.0e-11 ***
# L0_soc_env        0.06113    0.00609   10.03 < 2e-16 ***
# L1                0.08341    0.01281    6.51 7.5e-11 ***
# A0_PM2.501:A0_PM2.511 0.00238    0.00446    0.53 0.59

## estimate the conditional PNDE and TNDE, given L(0)=0 and L(1)=0
medflex.PNDE.death.L0 <- coef(neMod.death)["A0_PM2.501"]

```


8.5. “UNIFIED” APPROACH FOR ESTIMATING NATURAL DIRECT AND INDIRECT EFFECTS 153

```
# 0.0658
medflex.TNIE.death.L0 <- (coef(neMod.death)["A0_PM2.511"] +
                          coef(neMod.death)["A0_PM2.501:A0_PM2.511"])

# 0.0107

# 95% CI of the TNIE (which combines 2 coefficients) ?
# cov(b1,b2) = var(b1) + var(b2) + 2 * cov(b1,b2)
c(medflex.TNIE.death.L0 -
  qnorm(0.975) * sqrt(var(neMod.death$bootRes$t[,3]) + # column of A0_PM2.511
                      var(neMod.death$bootRes$t[,7]) + # column of A0_PM2.501:A0_PM2.511
                      2 * var(neMod.death$bootRes$t[,c(3,7)])[1,2]),
  medflex.TNIE.death.L0 +
  qnorm(0.975) * sqrt(var(neMod.death$bootRes$t[,3]) +
                      var(neMod.death$bootRes$t[,7]) +
                      2 * var(neMod.death$bootRes$t[,c(3,7)])[1,2]))

# A0_PM2.511 A0_PM2.511
#      0.0027      0.0188

## for quality of life
neMod.qol <- neModel(Y_qol ~ A0_PM2.50 * A0_PM2.51 + L0_male + L0_soc_env + L1,
                    family = "gaussian", # to estimate risk difference
                    expData = exp.Data,
                    se = c("robust") # or "bootstrap"
                    # nBoot = 1000, # if se = bootstrap
                    )

summary(neMod.qol)
#
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept)      73.187      0.226   324.09 < 2e-16 ***
# A0_PM2.501       -5.336      0.338   -15.77 < 2e-16 ***
# A0_PM2.511       -1.069      0.136    -7.87 3.4e-15 ***
# L0_male          -1.231      0.223    -5.53 3.3e-08 ***
# L0_soc_env       -3.546      0.230   -15.41 < 2e-16 ***
# L1               -4.101      0.243   -16.87 < 2e-16 ***
# A0_PM2.501:A0_PM2.511 -0.765      0.126    -6.09 1.1e-09 ***

## estimate the conditional PNDE and TNDE, given L(0)=0 and L(1)=0
medflex.PNDE.qol.L0 <- coef(neMod.qol)["A0_PM2.501"]
# -5.34
medflex.TNIE.qol.L0 <- (coef(neMod.qol)["A0_PM2.511"] +
                      coef(neMod.qol)["A0_PM2.501:A0_PM2.511"])
# -1.83

# 95% CI of the TNIE (which combines 2 coefficients) ?
# cov(b1,b2) = var(b1) + var(b2) + 2cov(b1,b2)
c(medflex.TNIE.qol.L0 -
```

```

qnorm(0.975) * sqrt(neMod.qol$vcov["A0_PM2.511", "A0_PM2.511"] +
                    neMod.qol$vcov["A0_PM2.501:A0_PM2.511", "A0_PM2.501:A0_PM2.511"] +
                    2 * neMod.qol$vcov["A0_PM2.511", "A0_PM2.501:A0_PM2.511"])),
medflex.TNIE.qol.L0 +
qnorm(0.975) * sqrt(neMod.qol$vcov["A0_PM2.511", "A0_PM2.511"] +
                    neMod.qol$vcov["A0_PM2.501:A0_PM2.511", "A0_PM2.501:A0_PM2.511"] +
                    2 * neMod.qol$vcov["A0_PM2.511", "A0_PM2.501:A0_PM2.511"])))
# A0_PM2.511 A0_PM2.511
#      -2.30      -1.37

# plot results
plot(neMod.qol)

### estimate marginal PNDE and TNDE (population average)
# ----- #

## 1) Expand the dataset and calculate weights using the neWeight() function

## Estimate a model of the exposure
g.A.L0 <- glm(A0_PM2.5 ~ L0_male + L0_soc_env + L1, # will no work without L(1) !
              family = "binomial", data = df1_int)

## Estimate a model of the mediator
g.M <- glm(M_diabetes ~ factor(A0_PM2.5) + # the 1st variable should be the exposure
           L0_male + L0_soc_env + L1, # then add baseline confounders
           family = "binomial", data = df1_int)

# expend the data set and add a second column A* = 1 - A
# and calculate the weights = sw_M.Astar / sw_M.A
exp.Data <- neWeight(g.M)

## 2) Fit the natural effect model using the neModel() function
## for death
neMod.death.pop <- neModel(Y_death ~ A0_PM2.50 * A0_PM2.51, # no need for L(0), L(1)
                          family = "gaussian", # to estimate risk difference
                          expData = exp.Data,
                          xFit = g.A.L0, # model of the exposure
                          se = c("robust")) # or "bootstrap"
summary(neMod.death.pop)
# Parameter estimates:
# Estimate Std. Error z value Pr(>|z|)
# (Intercept)      0.198840  0.004247  46.82 < 2e-16 ***
# A0_PM2.501      0.065950  0.014346   4.60 4.3e-06 ***
# A0_PM2.511      0.008534  0.001619   5.27 1.4e-07 ***
# A0_PM2.501:A0_PM2.511 -0.000406  0.003744  -0.11  0.91

```

```
## estimate the population average PNDE and TNDE,
medflex.PNDE.death <- coef(neMod.death.pop)["A0_PM2.501"]
# 0.066
medflex.TNIE.death <- (coef(neMod.death.pop)["A0_PM2.511"] +
                      coef(neMod.death.pop)["A0_PM2.501:A0_PM2.511"])
# 0.00813
# the results are the same than results calculated by hand

## for quality of life
neMod.qol.pop <- neModel(Y_qol ~ A0_PM2.50 * A0_PM2.51, # no need for L(0),L(1)
                        family = "gaussian", # to estimate risk difference
                        expData = exp.Data,
                        xFit = g.A.L0, # model of the exposure
                        se = c("robust")) # or "bootstrap"

summary(neMod.qol.pop)
# Parameter estimates:
#
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept)      69.085      0.117  590.36 < 2e-16 ***
# A0_PM2.501       -5.377      0.350  -15.38 < 2e-16 ***
# A0_PM2.511       -1.077      0.137   -7.88 3.3e-15 ***
# A0_PM2.501:A0_PM2.511 -0.695      0.120   -5.80 6.7e-09 ***

## estimate the population average PNDE and TNDE,
medflex.PNDE.qol <- coef(neMod.qol.pop)["A0_PM2.501"]
# -5.38
medflex.TNIE.qol <- (coef(neMod.qol.pop)["A0_PM2.511"] +
                    coef(neMod.qol.pop)["A0_PM2.501:A0_PM2.511"])
# -1.77
# the results are the same than results calculated by hand
```

8.5.2 Estimation of the MSM by g-computation

8.5.2.1 Manual calculation

The “unified” MSM can also be estimated by g-computation. Note that for the estimation of the population average effects, calculation relies on a weighted regression, as with the IPTW approach.

We can apply the following steps:

1. Estimate a suitable model for the outcome conditional on the exposure, mediator and confounders, using the original data set
2. Construct a new data set by repeating the original data set twice, and including an additional variable A^* , equal to the original exposure in both

data sets, and change the value of A from A to $1 - A$ in the 2nd data set. Then impute counterfactuals $\mathbb{E}(Y_{a,M_{a^*}}|L(0), L(1))$

3. Estimate the unified MSM as a model of the outcome including A and A^* (and their interaction), adjusted for baseline confounders $L(0)$ and $L(1)$. In order to obtain population average model (unconditional on baseline confounders), coefficients can be estimated using a weighted regression (with simple weights or stabilized weights of the exposure). Use the estimated coefficients of the MSM to calculate the PNDE and TNIE.

```
rm(list=ls())
df1_int <- read.csv(file = "data/df1_int.csv")

## Step 1. Estimate a suitable model for the outcome conditional on the exposure,
##           mediator and confounders, using the original data set
Q.Y.death <- glm(Y_death ~ L0_male + L0_soc_env + L1 + A0_PM2.5 * M_diabetes,
                 family = "binomial",
                 data = df1_int)

Q.Y.qol <- glm(Y_qol ~ L0_male + L0_soc_env + L1 + A0_PM2.5 * M_diabetes,
              family = "gaussian",
              data = df1_int)

## Step 2. Expand data and impute counterfactuals  $E(Y_{\{a,M_{a^*}\}}|L(0),L(1))$ 
## Expand data
# create identifier
df1_int$id <- 1:nrow(df1_int)

# duplicate the original data set
df1_int.1 <- df1_int.2 <- df1_int

# Add an  $A^*$  variable where  $A^* = A$ 
df1_int.1$A0_PM2.5_star <- df1_int$A0_PM2.5
df1_int.2$A0_PM2.5_star <- df1_int$A0_PM2.5

# for the 2nd data set, change the exposure  $A = 1 - A$ 
df1_int.2$A0_PM2.5 <- 1 - df1_int$A0_PM2.5

# stack the 2 new datasets
df1_int.new <- rbind(df1_int.1, df1_int.2)
# sort the data set by the id-variable to use geeglm
# (necessary to use geepack and the geeglm() later)
df1_int.new <- df1_int.new[order(df1_int.new$id), ]

head(df1_int.new, 12)
#           L0_male L0_soc_env A0_PM2.5 L1 M_diabetes Y_death Y_qol A0_PM2.5_temp id A0_PM2.5_star
```

8.5. “UNIFIED” APPROACH FOR ESTIMATING NATURAL DIRECT AND INDIRECT EFFECTS 157

```
# 1      0      1      0 1      0      0 93.4      0 1      0
# 10001  0      1      1 1      0      0 93.4      0 1      0
# 2      1      1      1 1      0      1 64.0      1 2      1
# 10002  1      1      0 1      0      1 64.0      1 2      1
# 3      1      1      0 0      0      0 75.6      0 3      0
# 10003  1      1      1 0      0      0 75.6      0 3      0
# 4      1      0      0 0      0      0 89.8      0 4      0
# 10004  1      0      1 0      0      0 89.8      0 4      0
# 5      1      1      0 0      0      0 77.2      0 5      0
# 10005  1      1      1 0      0      0 77.2      0 5      0
# 6      1      1      0 0      1      0 73.9      0 6      0
# 10006  1      1      1 0      1      0 73.9      0 6      0

## Impute counterfactuals
# death - we predict the probability  $P(Y=1|A,M,L(0),L(1))$ 
Y.pred.death <- predict(Q.Y.death, newdata = df1_int.new, type = "response")

# qol - we can predict the mean  $E(Y|A,M,L(0),L(1))$ 
Y.pred.qol <- predict(Q.Y.qol, newdata = df1_int.new, type = "response")

## Step 3. Fit a suitable model to the outcome including only A and A* (and their
## interaction)
library(geepack)
# for death (conditional)
MSM.model.death.1 <- geeglm(Y.pred.death ~ A0_PM2.5 * A0_PM2.5_star +
  L0_male + L0_soc_env + L1,
  family = "gaussian",
  id = df1_int.new$id,
  data = df1_int.new,
  scale.fix = TRUE)

summary(MSM.model.death.1)
#               Estimate Std.err   Wald Pr(>|W|)
# (Intercept)    0.104153 0.000611  29104 < 2e-16 ***
# A0_PM2.5       0.063543 0.000138 210638 < 2e-16 ***
# A0_PM2.5_star  0.007032 0.001112    40 2.5e-10 ***
# L0_male        0.054291 0.000706   5922 < 2e-16 ***
# L0_soc_env     0.065692 0.000692   9004 < 2e-16 ***
# L1             0.086446 0.000860  10100 < 2e-16 ***
# A0_PM2.5:A0_PM2.5_star 0.004918 0.000397   154 < 2e-16 ***

## estimate the conditional PNDE and TNDE, given  $L(0)=0$  and  $L(1)=0$ 
PNDE.death.L0 <- coef(MSM.model.death.1)["A0_PM2.5"]
# 0.0635
TNIE.death.L0 <- (coef(MSM.model.death.1)["A0_PM2.5_star"] +
  coef(MSM.model.death.1)["A0_PM2.5:A0_PM2.5_star"])
```

```

# 0.012

# For quality of life (conditional)
MSM.model.qol.1 <- geeglm(Y.pred.qol ~ A0_PM2.5 * A0_PM2.5_star +
  L0_male + L0_soc_env + L1,
  family = "gaussian",
  id = df1_int.new$id,
  data = df1_int.new,
  scale.fix = TRUE)

summary(MSM.model.qol.1)
#
# (Intercept)      Estimate Std. err      Wald Pr(>|W|)
# A0_PM2.5          -5.3013  0.0268   39033.2 < 2e-16 ***
# A0_PM2.5_star     -1.0535  0.1353    60.6 6.9e-15 ***
# L0_male           -1.3289  0.1038    164.0 < 2e-16 ***
# L0_soc_env        -3.6467  0.1061   1180.6 < 2e-16 ***
# L1                -4.2325  0.1168   1312.8 < 2e-16 ***
# A0_PM2.5:A0_PM2.5_star -0.7920  0.0870    82.9 < 2e-16 ***

## estimate the conditional PNDE and TNDE, given L(0)=0 and L(1)=0
PNDE.qol.L0 <- coef(MSM.model.qol.1)["A0_PM2.5"]
# -5.3
TNIE.qol.L0 <- (coef(MSM.model.qol.1)["A0_PM2.5_star"] +
  coef(MSM.model.qol.1)["A0_PM2.5:A0_PM2.5_star"])
# -1.85

## For population average estimations
## Estimate a model of the exposure
g.A.L0 <- glm(A0_PM2.5 ~ L0_male + L0_soc_env + L1, # will no work without L(1) !
  family = "binomial", data = df1_int)
g.Ais1.L0 <- predict(g.A.L0, newdata = df1_int.new, type = "response")

w <- rep(NA, nrow(df1_int.new))
w[df1_int.new$A0_PM2.5_star == 1] <- 1 / g.Ais1.L0[df1_int.new$A0_PM2.5_star == 1]
w[df1_int.new$A0_PM2.5_star == 0] <- 1 / (1 - g.Ais1.L0[df1_int.new$A0_PM2.5_star == 0])

# for death (population average), we use the weighted regression
MSM.model.death.2 <- geeglm(Y.pred.death ~ A0_PM2.5 * A0_PM2.5_star,
  family = "gaussian",
  weights = w,
  id = df1_int.new$id,
  data = df1_int.new,
  scale.fix = TRUE)

summary(MSM.model.death.2)
#
# Estimate Std. err      Wald Pr(>|W|)

```

```

# (Intercept)          0.198882 0.000644 95250.3 < 2e-16 ***
# A0_PM2.5             0.063867 0.000138 213552.3 < 2e-16 ***
# A0_PM2.5_star        0.009213 0.002009    21.0 4.5e-06 ***
# A0_PM2.5:A0_PM2.5_star 0.002330 0.000445    27.4 1.7e-07 ***

## estimate the conditional PNDE and TNDE, given L(0)=0 and L(1)=0
PNDE.death <- coef(MSM.model.death.2)["A0_PM2.5"]
# 0.0639
TNIE.death <- (coef(MSM.model.death.2)["A0_PM2.5_star"] +
               coef(MSM.model.death.2)["A0_PM2.5:A0_PM2.5_star"])
# 0.0115

# For quality of life (population average), we use the weighted regression
MSM.model.qol.2 <- geeglm(Y.pred.qol ~ A0_PM2.5 * A0_PM2.5_star,
                          family = "gaussian",
                          weights = w,
                          id = df1_int.new$id,
                          data = df1_int.new,
                          scale.fix = TRUE)

summary(MSM.model.qol.2)
#               Estimate Std. err      Wald Pr(>|W|)
# (Intercept)      69.0849   0.0492 1.97e+06 < 2e-16 ***
# A0_PM2.5         -5.3108   0.0269 3.88e+04 < 2e-16 ***
# A0_PM2.5_star    -1.1690   0.1624 5.18e+01 6.1e-13 ***
# A0_PM2.5:A0_PM2.5_star -0.7395   0.0906 6.66e+01 3.3e-16 ***

## estimate the conditional PNDE and TNDE, given L(0)=0 and L(1)=0
PNDE.qol.L0 <- coef(MSM.model.qol.2)["A0_PM2.5"]
# -5.31
TNIE.qol.L0 <- (coef(MSM.model.qol.2)["A0_PM2.5_star"] +
                coef(MSM.model.qol.2)["A0_PM2.5:A0_PM2.5_star"])
# -1.91

```

8.5.2.2 Using the medflex package

We can use the `medflex` package to apply this approach, as described below.

The “unified” MSM can be defined as a model of the expected counterfactual outcome $\mathbb{E}(Y_{a,M_{a^*}} \mid L(0), L(1))$ conditional on the baseline confounders $L(0)$ and $L(1)$, or as a model the marginal expected counterfactual outcome $\mathbb{E}(Y_{a,M_{a^*}})$.

For estimation by g-computation, the data is expanded using the `neImpute` function, and the MSM is estimated using the `neModel` function. Population average estimations are obtained using a weighted regression, where the weights are calculated using a model of the exposure, stated at the `xFit` argument inside the `neModel` function.

Standard errors can be computed by bootstrap or as robust “Sandwich” standard errors.

```
library(medflex)

### estimate PNDE and TNDE, conditional on L0 and L1
# ----- #
# get back to the original data set
df1_int <- subset(df1_int, select = -c(id))
df1_int$A0_PM2.5 <- factor(df1_int$A0_PM2.5)

## 1) Fit a model for the outcome
Q.Y.death <- glm(Y_death ~ A0_PM2.5 + # start with the exposure
                M_diabetes + A0_PM2.5:M_diabetes + # then the mediator
                L0_male + L0_soc_env + L1, # then baseline confounders
                family = "binomial",
                data = df1_int)

Q.Y.qol <- glm(Y_qol ~ A0_PM2.5 + # start with the exposure
              M_diabetes + A0_PM2.5:M_diabetes + # then the mediator
              L0_male + L0_soc_env + L1, # then baseline confounders
              family = "gaussian",
              data = df1_int)

## 2) expend the data set and add a second column A* = 1 - A
exp.Data.death <- neImpute(Q.Y.death)
head(exp.Data.death)
#   id A0_PM2.50 A0_PM2.51 L0_male L0_soc_env L1 M_diabetes Y_death Y_qol
# 1  1         0         0         0         1  1         0   0.222  93.4
# 2  1         1         0         0         1  1         0   0.292  93.4
# 3  2         1         1         1         1  1         0   0.356  64.0
# 4  2         0         1         1         1  1         0   0.277  64.0
# 5  3         0         0         1         1  0         0   0.197  75.6
# 6  3         1         0         1         1  0         0   0.261  75.6

exp.Data.qol <- neImpute(Q.Y.qol)
head(exp.Data.qol)
#   id A0_PM2.50 A0_PM2.51 L0_male L0_soc_env L1 M_diabetes Y_death Y_qol
# 1  1         0         0         0         1  1         0     0  68.4
# 2  1         1         0         0         1  1         0     0  64.7
# 3  2         1         1         1         1  1         0     1  64.0
# 4  2         0         1         1         1  1         0     1  67.7
# 5  3         0         0         1         1  0         0     0  71.2
# 6  3         1         0         1         1  0         0     0  67.4
```


8.5. "UNIFIED" APPROACH FOR ESTIMATING NATURAL DIRECT AND INDIRECT EFFECTS 161

```
# the predicted outcomes by medflex are the same than our previous predictions
plot(exp.Data.death$Y_death, Y.pred.death)
abline(0,1)

plot(exp.Data.qol$Y_qol, Y.pred.qol)
abline(0,1)

## 3) Fit the natural effect model using the neModel() function
##      (adjusted for baseline confounders)

## for death
neMod.death <- neModel(Y_death ~ A0_PM2.50 * A0_PM2.51 + L0_male + L0_soc_env + L1,
  family = "gaussian", # to estimate risk difference
  expData = exp.Data.death,
  se = c("robust") # or "bootstrap"
)

summary(neMod.death)
#
#      Estimate Std. Error z value Pr(>|z|)
# (Intercept)      0.10415    0.00829   12.56 < 2e-16 ***
# A0_PM2.501        0.06354    0.01373    4.63 3.7e-06 ***
# A0_PM2.511        0.00703    0.00175    4.01 6.1e-05 ***
# L0_male           0.05429    0.00867    6.26 3.8e-10 ***
# L0_soc_env        0.06569    0.00882    7.45 9.3e-14 ***
# L1                0.08645    0.00999    8.66 < 2e-16 ***
# A0_PM2.501:A0_PM2.511 0.00492    0.00391    1.26 0.21

## estimate the conditional PNDE and TNDE, given L(0)=0 and L(1)=0
medflex.PNDE.death.L0 <- coef(neMod.death)["A0_PM2.501"]
# 0.0635
medflex.TNIE.death.L0 <- (coef(neMod.death)["A0_PM2.511"] +
  coef(neMod.death)["A0_PM2.501:A0_PM2.511"])
# 0.012

## for quality of life
neMod.qol <- neModel(Y_qol ~ A0_PM2.50 * A0_PM2.51 + L0_male + L0_soc_env + L1,
  family = "gaussian", # to estimate risk difference
  expData = exp.Data.qol,
  se = c("robust")) # or "bootstrap"

summary(neMod.qol)
#
#      Estimate Std. Error z value Pr(>|z|)
# (Intercept)      73.337    0.230  319.17 < 2e-16 ***
# A0_PM2.501       -5.301    0.340  -15.61 < 2e-16 ***
# A0_PM2.511       -1.054    0.139   -7.58 3.6e-14 ***
# L0_male          -1.329    0.227   -5.85 5.0e-09 ***
# L0_soc_env       -3.647    0.235  -15.54 < 2e-16 ***
```

```

#      L1                -4.233      0.249  -17.01  < 2e-16 ***
#      A0_PM2.501:A0_PM2.511  -0.792      0.127   -6.21  5.1e-10 ***

## estimate the conditional PNDE and TNDE, given L(0)=0 and L(1)=0
medflex.PNDE.qol.L0 <- coef(neMod.qol)["A0_PM2.501"]
# -5.3
medflex.TNIE.qol.L0 <- (coef(neMod.qol)["A0_PM2.511"] +
                        coef(neMod.qol)["A0_PM2.501:A0_PM2.511"])
# -1.85

### estimate marginal PNDE and TNDE (population average)
# ----- #
## Estimate a model of the exposure
g.A.L0 <- glm(A0_PM2.5 ~ L0_male + L0_soc_env + L1, # will no work without L(1) !
              family = "binomial", data = df1_int)

## 3) Fit the natural effect model using the neModel() function
## for death
neMod.death.pop <- neModel(Y_death ~ A0_PM2.50 * A0_PM2.51, # no need for L(0),L(1)
                           family = "gaussian", # to estimate risk difference
                           expData = exp.Data.death,
                           xFit = g.A.L0, # model of the exposure
                           se = c("robust")) # or "bootstrap"

summary(neMod.death.pop)
# Parameter estimates:
#
#      Estimate Std. Error z value Pr(>|z|)
#      (Intercept)      0.19888    0.00425   46.83 < 2e-16 ***
#      A0_PM2.501        0.06387    0.01378    4.64 3.6e-06 ***
#      A0_PM2.511        0.00921    0.00169    5.45 5.1e-08 ***
#      A0_PM2.501:A0_PM2.511 0.00233    0.00367    0.64 0.53

## estimate the population average PNDE and TNDE,
medflex.PNDE.death <- coef(neMod.death.pop)["A0_PM2.501"]
# 0.0639
medflex.TNIE.death <- (coef(neMod.death.pop)["A0_PM2.511"] +
                      coef(neMod.death.pop)["A0_PM2.501:A0_PM2.511"])
# 0.0115
# the results are the same than results calculated by hand

## for quality of life
neMod.qol.pop <- neModel(Y_qol ~ A0_PM2.50 * A0_PM2.51, # no need for L(0),L(1)
                         family = "gaussian", # to estimate risk difference
                         expData = exp.Data.qol,
                         xFit = g.A.L0, # model of the exposure

```

```

                                se = c("robust")) # or "bootstrap"
summary(neMod.qol.pop)
# Parameter estimates:
#
#           Estimate Std. Error z value Pr(>|z|)
# (Intercept)      69.085      0.117  590.50 < 2e-16 ***
# A0_PM2.501       -5.311      0.339  -15.65 < 2e-16 ***
# A0_PM2.511       -1.169      0.145   -8.09 6.1e-16 ***
# A0_PM2.501:A0_PM2.511 -0.740      0.125   -5.92 3.2e-09 ***

## estimate the population average PNDE and TNDE,
medflex.PNDE.qol <- coef(neMod.qol.pop)["A0_PM2.501"]
# -5.31
medflex.TNIE.qol <- (coef(neMod.qol.pop)["A0_PM2.511"] +
                     coef(neMod.qol.pop)["A0_PM2.501:A0_PM2.511"])
# -1.91
# the results are the same than results calculated by hand

```


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, $\bar{Q} = \mathbb{E}(Y \mid A, L(0))$). For example, the Average Total Effect (ATE) is defined as a marginal effect, estimated using the empirical mean of such \bar{Q} functions:

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

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 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 $\hat{\Psi}_{\text{gcomp}}^{\text{ATE}}$. 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 dis-

tribution 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 of the study.

9.1 TMLE for the ATE

In order to illustrate the TMLE procedure, the estimation of a mean counterfactual outcome, denoted $\Psi(A = 1) = \mathbb{E}[\bar{Q}(A = 1, L(0))]$, 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 $g(A = 1 | L(0))$. 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” $H = \frac{I(A=1)}{\bar{g}(A=1|L(0))}$ can be applied:

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

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 $\bar{Q}(A=1)$
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 $D^*(Q_0, g_0)$:

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

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

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

```
rm(list=ls())
df2_int <- read.csv(file = "./data/df2_int.csv")

## 1) Estimate Qbar and predict Qbar when A0_PM2.5 is set to 1
Q.fit <- glm(Y_death ~ A0_PM2.5 + L0_male + L0_soc_env,
             family = "binomial", data = df2_int)
data.A1 <- df2_int
data.A1$A0_PM2.5 <- 1

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

## 2) Estimate the treatment mechanism
g.L <- glm(A0_PM2.5 ~ L0_male + L0_soc_env,
           family = "binomial", data = df2_int)
# predict the probabilities g(A=1 | L(0)) = P(A0_PM2.5=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).
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_PM2.5 == 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
Qstar <- predict(update.fit, data = data.frame(logitQ, H), type = "response")

# 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.tmlle <- predict(update.fit.ltmle,
                      data = data.frame(logitQ, H),
                      type = "response")

head(Qstar.tmlle)

```



```
#           1           2           3           4           5           6
# 0.2872412 0.3441344 0.3441344 0.2591356 0.3441344 0.3441344

## 5) Obtain the substitution 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

# The 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:

```
rm(list=ls())
df2_int <- read.csv(file = "./data/df2_int.csv")

library(ltmle)
?ltmle

# The Qform and gform arguments are defined from the DAG
Qform <- c(Y_death="Q.kplus1 ~ L0_male + L0_soc_env + A0_PM2.5")
gform <- c("A0_PM2.5 ~ L0_male + L0_soc_env")

# in the ltmle package, the data set should be formatted 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_soc_env, A0_PM2.5, Y_death))

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

Psi_Ais1 <- ltmle(data_ltmle,
                  Anodes = "A0_PM2.5",
                  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 procedure (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 an additive model, whereas 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 = "AO_PM2.5",
  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]]$AO_PM2.5
#
# 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
# 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 continuous outcomes
Qform <- c(Y_qol="Q.kplus1 ~ L0_male + L0_soc_env + A0_PM2.5")
gform <- c("A0_PM2.5 ~ L0_male + L0_soc_env")

set.seed(42)
Psi_ATE_tmle_qol <- ltmle(data = subset(df2_int,
                                         select = c(L0_male, L0_soc_env,
                                                    A0_PM2.5,
                                                    Y_qol)),
                          Anodes = "A0_PM2.5",
                          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.

```
# 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

```
rm(list=ls())
df2_int <- read.csv(file = "./data/df2_int.csv")

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_soc_env + A0_PM2.5",
           Y_death="Q.kplus1 ~ L0_male + L0_soc_env + L1 +
                   A0_PM2.5 * M_diabetes")

# Define the formulas for the estimation of the 2 g function
gform <- c("A0_PM2.5 ~ L0_male + L0_soc_env",
           "M_diabetes ~ L0_male + L0_soc_env + A0_PM2.5 + L1")

# The data frame should follow the time-ordering of the nodes
data_binary <- subset(df2_int, select = c(L0_male, L0_soc_env,
                                         A0_PM2.5, L1,
                                         M_diabetes, 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_PM2.5", "M_diabetes"),
                           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
              c(0,0)), # counterfactual intervention do(A=0,M
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)
# Additive Treatment Effect:
# 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("AO_PM2.5", "M_diabetes"),
                             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,M
SL.library = SL.library,
estimate.time = FALSE, # estimate computation time?
gcomp = FALSE,
variance.method = "ic")

summary(CDE_ltmle_M1_death)
# Additive Treatment Effect:
# 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_soc_env,
                                             AO_PM2.5, L1,
                                             M_diabetes, 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_soc_env + AO_PM2.5",
          Y_qol="Q.kplus1 ~ L0_male + L0_soc_env + L1 +
                AO_PM2.5 * M_diabetes")

set.seed(42)
## CDE, setting M=0
CDE_ltmle_M0_qol <- ltmle(data = data_continuous,
                          Anodes = c("AO_PM2.5", "M_diabetes"),
                          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("AO_PM2.5", "M_diabetes"),
                          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].

9.3 One-step and TMLE estimator of the Marginal Randomized/Interventional Direct and Indirect Effects

The `medoutcon` package (Hejazi, Rudolph, and Díaz 2022),(Díaz et al. 2020) enables the estimation of Marginal Randomized/Interventional Direct and Indirect Effects (analogues of the Natural direct and indirect effects) by one-step estimation or TMLE. The one-step estimator relies on “cross-fitting” and the TMLE relies on cross-validation.

More practical details can be found in the Materials for the workshop “Modern Causal Mediation Analysis” at the 2024 Society for Epidemiologic Research (SER) annual meeting in Austin, TX.

This package is associated with the `tlverse` ecosystem which has been developed for applying Targeted Learning methodology in practice. To use the `medoutcon` package, we will need:

- the `s13` package (to implement SuperLearning)
- and the Highly-adaptive lasso `hal9001` package to estimate some functions of interest.

Note that the `medoutcon` package can deal with multiple mediators, but only a single binary intermediate confounder $L(1)$. For causal structures with multiple mediators and multiple intermediate confounders, the `HDmediation` package could be used instead.

9.3. ONE-STEP AND TMLE ESTIMATOR OF THE MARGINAL RANDOMIZED/INTERVENTIONAL DIRECT

Below, we apply the one-step estimator to the data set with an intermediate confounder $L(1)$ affected by the exposure, and the presence of an $A * M$ interaction effect on the outcome.

```
# remotes::install_github("nhejazi/medoutcon")
# remotes::install_github("nhejazi/medoutcon", INSTALL_opts=c("--no-multiarch"))
# https://arxiv.org/pdf/1912.09936
rm(list=ls())
df2_int <- read.csv(file = "data/df2_int.csv")

library(tidyverse)
library(sl3)
library(medoutcon)
library(hal9001)

?medoutcon::medoutcon

## 1) binary outcome
# ----- #
### compute one-step estimate of the interventional direct effect
set.seed(1234)
os_de <- medoutcon(W = df2_int[,c("L0_male", "L0_soc_env")], #matrix of baseline L(0)
                  A = df2_int$A0_PM2.5, # numeric vector of the exposure
                  Z = df2_int$L1, # numeric vector L(1) (only 1 variable)
                  M = df2_int$M_diabetes, # numeric vector or matrix
                  Y = df2_int$Y_death, # numeric vector
                  effect = "direct",
                  b_learners = sl3::Lrn_r_hal9001$new(), # outcome regression
                  estimator = "onestep")

os_de
# Interventional Direct Effect
# Estimator: onestep
# Estimate: 0.068
# Std. Error: 0.015
# 95% CI: [0.039, 0.096]

os_de$theta
# [1] 0.06763031

sqrt(os_de$var)
# [1] 0.01455125

set.seed(1234)
os_ie <- medoutcon(W = df2_int[,c("L0_male", "L0_soc_env")], #matrix of baseline L(0)
                  A = df2_int$A0_PM2.5, # numeric vector of the exposure
                  Z = df2_int$L1, # numeric vector L(1) (only 1 variable)
```

```

M = df2_int$M_diabetes, # numeric vector or matrix
Y = df2_int$Y_death, # numeric vector
effect = "indirect",
b_learners = sl3::Lrn_r_hal9001$new(), # outcome regression
estimator = "onestep")

os_ie
# Interventional Indirect Effect
# Estimator: onestep
# Estimate: 0.015
# Std. Error: 0.004
# 95% CI: [0.008, 0.022]

os_ie$theta
# [1] 0.01502601

### Estimate counterfactuals  $E(Y_{\{a,G_{\{a^*\}}\}})$ 
set.seed(1234)
EY_1M0 <- medoutcon(W = df2_int[,c("L0_male", "L0_soc_env")],
  A = df2_int$A0_PM2.5,
  Z = df2_int$L1,
  M = df2_int$M_diabetes,
  Y = df2_int$Y_death,
  contrast = c(1,0),
  b_learners = sl3::Lrn_r_hal9001$new(), # outcome regression
  estimator = "onestep")

EY_1M0
# Counterfactual TSM
# Contrast:  $A = 1, M(A = 0)$ 
# Estimator: onestep
# Estimate: 0.273
# Std. Error: 0.014
# 95% CI: [0.246, 0.301]

EY_0M0 <- medoutcon(W = df2_int[,c("L0_male", "L0_soc_env")],
  A = df2_int$A0_PM2.5,
  Z = df2_int$L1,
  M = df2_int$M_diabetes,
  Y = df2_int$Y_death,
  contrast = c(0,0),
  b_learners = sl3::Lrn_r_hal9001$new(), # outcome regression
  estimator = "onestep")

EY_0M0
# Counterfactual TSM
# Contrast:  $A = 0, M(A = 0)$ 
# Estimator: onestep

```

9.3. ONE-STEP AND TMLE ESTIMATOR OF THE MARGINAL RANDOMIZED/INTERVENTIONAL DIRECT

```
# Estimate: 0.203
# Std. Error: 0.004
# 95% CI: [0.195, 0.212]

## randomized Natural Direct Effect =
rNDE <- EY_1M0$theta - EY_0M0$theta
# [1] 0.06964436

## se
se_rNDE <- sqrt(var(EY_1M0$eif - EY_0M0$eif) / nrow(df2_int))
# [1] 0.01459493

## 95%CI
c(rNDE - qnorm(0.975) * se_rNDE,
  rNDE + qnorm(0.975) * se_rNDE)
# [1] 0.04103882 0.09824989

## 2) continuous outcome
# ----- #
### compute one-step estimate of the interventional direct effect
set.seed(1234)
os_de <- medoutcon(W = df2_int[,c("L0_male", "L0_soc_env")], #matrix of baseline L(0)
  A = df2_int$A0_PM2.5, # numeric vector of the exposure
  Z = df2_int$L1, # numeric vector L(1) (only 1 variable)
  M = df2_int$M_diabetes, # numeric vector or matrix
  Y = df2_int$Y_qol, # numeric vector
  effect = "direct",
  b_learners = sl3::Lrn_r_hal9001$new(), # outcome regression
  estimator = "onestep")

os_de
# Interventional Direct Effect
# Estimator: onestep
# Estimate: -6.597
# Std. Error: 0.346
# 95% CI: [-7.276, -5.918]

set.seed(1234)
os_ie <- medoutcon(W = df2_int[,c("L0_male", "L0_soc_env")], #matrix of baseline L(0)
  A = df2_int$A0_PM2.5, # numeric vector of the exposure
  Z = df2_int$L1, # numeric vector L(1) (only 1 variable)
  M = df2_int$M_diabetes, # numeric vector or matrix
  Y = df2_int$Y_qol, # numeric vector
  effect = "indirect",
  b_learners = sl3::Lrn_r_hal9001$new(), # outcome regression
  estimator = "onestep")
```

```

os_ie
# Interventional Indirect Effect
# Estimator: onestep
# Estimate: -1.703
# Std. Error: 0.239
# 95% CI: [-2.172, -1.234]

```

Below, we apply the TMLE estimator to the same data set. Note that the estimation of the direct effect seems to be biased (confirmed on simulations): maybe the arguments given in the `medoutcon` function are not correct?

```

## 1) binary outcome
# ----- #
### compute tmle estimate of the interventional direct effect & indirect effects
set.seed(1234)
tmle_de <- medoutcon(W = df2_int[,c("L0_male", "L0_soc_env")], #matrix of baseline L(0)
                    A = df2_int$A0_PM2.5, # numeric vector of the exposure
                    Z = df2_int$L1, # numeric vector L(1) (only 1 variable)
                    M = df2_int$M_diabetes, # numeric vector or matrix
                    Y = df2_int$Y_death, # numeric vector
                    effect = "direct",
                    estimator = "tmle",
                    b_learners = sl3::Lrn_r_ha19001$new(), # outcome regression
                    # estimator_args = list(cv_folds = 10L, # instead of 5L
                    #                               max_iter = 1L, # instead of 5L
                    #                               tiltmod_tol = 1) # instead of 5
                    )

tmle_de
# Interventional Direct Effect
# Estimator: tmle
# Estimate: 0.042          <- ?! biased (confirmed on simulations)
# Std. Error: 0.015
# 95% CI: [0.014, 0.071]

tmle_de$theta
# [1] 0.04222315

sqrt(tmle_de$var)
# [1] 0.01455125

set.seed(1234)
tmle_ie <- medoutcon(W = df2_int[,c("L0_male", "L0_soc_env")], #matrix of baseline L(0)
                    A = df2_int$A0_PM2.5, # numeric vector of the exposure
                    Z = df2_int$L1, # numeric vector L(1) (only 1 variable)
                    M = df2_int$M_diabetes, # numeric vector or matrix

```

9.3. ONE-STEP AND TMLE ESTIMATOR OF THE MARGINAL RANDOMIZED/INTERVENTIONAL DIRECT

```

        Y = df2_int$Y_death, # numeric vector
        effect = "indirect",
        b_learners = sl3::Lrn_r_hal9001$new(), # outcome regression
        estimator = "tmle")

tmle_ie
# Interventional Indirect Effect
# Estimator: tmle
# Estimate: 0.013
# Std. Error: 0.004
# 95% CI: [0.006, 0.021]

## 2) continuous outcome
# ----- #
### compute tmle estimate of the interventional direct effect & indirect effects
set.seed(1234)
tmle_de <- medoutcon(W = df2_int[,c("L0_male", "L0_soc_env")], #matrix of baseline L(0)
                    A = df2_int$A0_PM2.5, # numeric vector of the exposure
                    Z = df2_int$L1, # numeric vector L(1) (only 1 variable)
                    M = df2_int$M_diabetes, # numeric vector or matrix
                    Y = df2_int$Y_qol, # numeric vector
                    effect = "direct",
                    b_learners = sl3::Lrn_r_hal9001$new(), # outcome regression
                    estimator = "tmle")

tmle_de
# Interventional Direct Effect
# Estimator: tmle
# Estimate: -4.808          <- also biased ?
# Std. Error: 0.346
# 95% CI: [-5.487, -4.129]

set.seed(1234)
tmle_ie <- medoutcon(W = df2_int[,c("L0_male", "L0_soc_env")], #matrix of baseline L(0)
                    A = df2_int$A0_PM2.5, # numeric vector of the exposure
                    Z = df2_int$L1, # numeric vector L(1) (only 1 variable)
                    M = df2_int$M_diabetes, # numeric vector or matrix
                    Y = df2_int$Y_qol, # numeric vector
                    effect = "indirect",
                    b_learners = sl3::Lrn_r_hal9001$new(), # outcome regression
                    estimator = "tmle")

tmle_ie
# Interventional Indirect Effect
# Estimator: tmle
# Estimate: -1.738
# Std. Error: 0.239
# 95% CI: [-2.207, -1.269]

```


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:

$$\begin{aligned}
 P(L(0)_{male} = 1) &= p_{L(0)_{male}} \\
 P(L(0)_{soc.env} = 1) &= p_{L(0)_{soc.env}} \\
 P(A_{PM_{2.5}} = 1) &= \beta_A + \beta_{male}^A \times L(0)_{male} + \beta_{soc.env}^A \times L(0)_{soc.env} \\
 P(L(1) = 1) &= p_{L(1)} \\
 P(M_{smoking} = 1) &= \beta_M + \beta_{male}^M \times L(0)_{male} + \beta_{soc.env}^M \times L(0)_{soc.env} + \beta_{L(1)}^M \times L(1) + \beta_A^M \times A_{PM_{2.5}} \\
 P(Y_{death} = 1) &= \beta_Y + \beta_{male}^Y \times L(0)_{male} + \beta_{soc.env}^Y \times L(0)_{soc.env} + \beta_{L(1)}^Y \times L(1) \\
 &\quad + \beta_A^Y \times A_{PM_{2.5}} + \beta_M^Y \times M_{diabetes} + \beta_{A*M}^Y \times A_{PM_{2.5}} \times M_{diabetes} \\
 \mathbb{E}(Y_{Qol} = 1) &= \gamma_Y + \gamma_{male}^Y \times L(0)_{male} + \gamma_{soc.env}^Y \times L(0)_{soc.env} + \gamma_{L(1)}^Y \times L(1) \\
 &\quad + \gamma_A^Y \times A_{PM_{2.5}} + \gamma_M^Y \times M_{diabetes} + \gamma_{A*M}^Y \times A_{PM_{2.5}} \times M_{diabetes} + \varepsilon_Y
 \end{aligned}$$

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_soc_env <- 0.65

  # A: A0_PM2.5 <- rbinom( 0.05 + 0.04 * L0_male + 0.06 * L0_soc_env )
  b_A <- 0.05 # reference prevalence is 5%
  b_male_A <- 0.04 # + 0.04 for the effect of L0_male -> A0_PM2.5
  b_soc_env_A <- 0.06 # +0.06 for the effect of L0_soc_env -> A0_PM2.5

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

  # M: M_diabetes <- rbinom( 0.2 + 0.05 * L0_male + 0.06 * L0_soc_env + 0.07 * L1 +
  #                          0.1 * A0_PM2.5 )
  b_M <- 0.2 # reference prevalence is 20%
  b_male_M <- 0.05 # +0.05 for the effect of L0_male -> M_diabetes
  b_soc_env_M <- 0.06 # +0.06 for the effect of L0_soc_env -> M_diabetes
  b_L1_M <- 0.07 # +0.07 for the effect of L1 -> M_diabetes
  b_A_M <- 0.1 # +0.10 for the effect of A0_PM2.5 -> M_diabetes

  # Y binary: rbinom( 0.10 + 0.06 * L0_male + 0.04 * L0_soc_env + 0.05 * A0_PM2.5 +
  #                  0.07 * L1 + 0.08 * M_diabetes +
  #                  0.03 * A0_PM2.5 * M_diabetes * 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_soc_env_Y <- 0.04 # +0.04 for the effect of L0_soc_env -> Y
  b_A_Y <- 0.05 # 0.05 for the effect of A0_PM2.5 -> 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_diabetes -> Y
  b_AM_Y <- 0.03 # 0.03 for the interaction effect A0_PM2.5 * M_diabetes -> Y

  # Y continuous: (75 - 1 * L0_male - 3 * L0_soc_env - 4 * A0_PM2.5 -3.5 * L1 -
  #                9 * M_diabetes -5 * A0_PM2.5 * M_diabetes * 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_soc_env_Y <- -3 # -3 for the effect of L0_soc_env -> Y
  c_A_Y <- -4 # -4 for the effect of A0_PM2.5 -> Y
  c_L1_Y <- -3.5 # -3.5 for the effect of L1 -> Y
  c_M_Y <- -9 # -9 for the effect of M_diabetes -> Y
  c_AM_Y <- -5 # - 5 for the interaction effect A0_PM2.5 * M_diabetes -> 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_soc_env = p_L0_soc_env,
           b_A = b_A, b_male_A = b_male_A, b_soc_env_A = b_soc_env_A,
           p_L1 = p_L1,
           b_M = b_M, b_male_M = b_male_M, b_soc_env_M = b_soc_env_M,
           b_L1_M = b_L1_M, b_A_M = b_A_M,
           b_Y = b_Y, b_male_Y = b_male_Y, b_soc_env_Y = b_soc_env_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_soc_env_Y = c_soc_env_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:

$$\begin{aligned}
P(L(0)_{male} = 1) &= p_{L(0)_{male}} \\
P(L(0)_{soc.env} = 1) &= p_{L(0)_{soc.env}} \\
P(A_{PM_{2.5}} = 1) &= \beta_A + \beta_{male}^A \times L(0)_{male} + \beta_{soc.env}^A \times L(0)_{soc.env} \\
P(L(1) = 1) &= \beta_{L(1)} + \beta_{male}^{L(1)} \times L(0)_{male} + \beta_{soc.env}^{L(1)} \times L(0)_{soc.env} + \beta_A^{L(1)} \times A_{PM_{2.5}} \\
P(M_{diabetes} = 1) &= \beta_M + \beta_{male}^M \times L(0)_{male} + \beta_{soc.env}^M \times L(0)_{soc.env} + \beta_{L(1)}^M \times L(1) + \beta_A^M \times A_{PM_{2.5}} \\
P(Y_{death} = 1) &= \beta_Y + \beta_{male}^Y \times L(0)_{male} + \beta_{soc.env}^Y \times L(0)_{soc.env} + \beta_{L(1)}^Y \times L(1) \\
&\quad + \beta_A^Y \times A_{PM_{2.5}} + \beta_M^Y \times M_{diabetes} + \beta_{A*M}^Y \times A_{PM_{2.5}} \times M_{diabetes} \\
\mathbb{E}(Y_{Qol} = 1) &= \gamma_Y + \gamma_{male}^Y \times L(0)_{male} + \gamma_{soc.env}^Y \times L(0)_{soc.env} + \gamma_{L(1)}^Y \times L(1) \\
&\quad + \gamma_A^Y \times A_{PM_{2.5}} + \gamma_M^Y \times M_{diabetes} + \gamma_{A*M}^Y \times A_{PM_{2.5}} \times M_{diabetes} + \varepsilon_Y
\end{aligned}$$

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_soc_env <- 0.65

# A: A0_PM2.5 <- rbinom( 0.05 + 0.04 * L0_male + 0.06 * L0_soc_env )
b_A <- 0.05 # reference prevalence is 5%
b_male_A <- 0.04 # + 0.04 for the effect of L0_male -> A0_PM2.5
b_soc_env_A <- 0.06 # +0.06 for the effect of L0_soc_env -> A0_PM2.5

# L1: L1 <- rbinom( 0.30 - 0.05 * L0_male + 0.08 * L0_soc_env +
#                 0.2 * A0_PM2.5 )
b_L1 <- 0.30 # reference prevalence is 30%
b_male_L1 <- -0.05 # - 0.05 for the effect of L0_male -> L1
b_soc_env_L1 <- +0.08 # + 0.08 for the effect of L0_soc_env -> L1
b_A_L1 <- +0.2 # +0.2 for the effect of A0_PM2.5 -> L1

# M: M_diabetes <- rbinom( 0.2 + 0.05 * L0_male + 0.06 * L0_soc_env +
#                         0.2 * L1 + 0.1 * A0_PM2.5 )
b_M <- 0.2 # reference prevalence is 20%
b_male_M <- 0.05 # +0.05 for the effect of L0_male -> M_diabetes
b_soc_env_M <- 0.06 # +0.06 for the effect of L0_soc_env -> M_diabetes
b_A_M <- 0.1 # +0.10 for the effect of A0_PM2.5 -> M_diabetes
b_L1_M <- 0.2 # +0.2 for the effect of L1 -> M_diabetes

# Y binary: rbinom( 0.10 + 0.06 * L0_male + 0.04 * L0_soc_env +
#                  0.05 * A0_PM2.5 + 0.07 * L1 + 0.08 * M_diabetes +
#                  0.03 * A0_PM2.5 * M_diabetes * 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_soc_env_Y <- 0.04 # +0.04 for the effect of L0_soc_env -> Y
b_A_Y <- 0.05 # 0.05 for the effect of A0_PM2.5 -> 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_diabetes -> Y
b_AM_Y <- 0.03 # 0.03 for the interaction effect A0_PM2.5 * M_diabetes -> Y

# Y continuous: (75 - 1 * L0_male - 3 * L0_soc_env - 4 * A0_PM2.5 +
#                -3.5 * L1 - 9 * M_diabetes +
#                -5 * A0_PM2.5 * M_diabetes * 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_soc_env_Y <- -3 # -3 for the effect of L0_soc_env -> Y
c_A_Y <- -4 # -4 for the effect of A0_PM2.5 -> Y
c_L1_Y <- -5 # -5 for the effect of L1 -> Y
c_M_Y <- -9 # -9 for the effect of M_diabetes -> Y
c_AM_Y <- -5 # - 5 for the interaction effect A0_PM2.5 * M_diabetes -> 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_soc_env = p_L0_soc_env,
           b_A = b_A, b_male_A = b_male_A, b_soc_env_A = b_soc_env_A,
           b_L1 = b_L1, b_male_L1 = b_male_L1, b_soc_env_L1 = b_soc_env_L1,
           b_A_L1 = b_A_L1,
           b_M = b_M, b_male_M = b_male_M, b_soc_env_M = b_soc_env_M,
           b_L1_M = b_L1_M, b_A_M = b_A_M,
           b_Y = b_Y, b_male_Y = b_male_Y, b_soc_env_Y = b_soc_env_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_soc_env_Y = c_soc_env_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_soc_env & sex=L0_male
  L0_male <- rbinom(N, size = 1, prob = b["p_L0_male"])
  L0_soc_env <- rbinom(N, size = 1, prob = b["p_L0_soc_env"])

  # exposure: AO_PM2.5
  AO_PM2.5 <- rbinom(N, size = 1, prob = b["b_A"] +
                    b["b_male_A"] * L0_male +
                    b["b_soc_env_A"] * L0_soc_env )

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

```

```

# mediator: M_diabetes
M_diabetes <- rbinom(N, size = 1, prob = b["b_M"] +
                    b["b_male_M"] * L0_male +
                    b["b_soc_env_M"] * L0_soc_env +
                    b["b_A_M"] * A0_PM2.5 +
                    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_soc_env_Y"] * L0_soc_env +
                    b["b_A_Y"] * A0_PM2.5 +
                    b["b_L1_Y"] * L1 +
                    b["b_M_Y"] * M_diabetes +
                    b["b_AM_Y"] * A0_PM2.5 * M_diabetes * A.M.inter )

# Y_qol
Y_qol <- ( b["mu_Y"] +
            b["c_male_Y"] * L0_male +
            b["c_soc_env_Y"] * L0_soc_env +
            b["c_A_Y"] * A0_PM2.5 +
            b["c_L1_Y"] * L1 +
            b["c_M_Y"] * M_diabetes +
            b["c_AM_Y"] * A0_PM2.5 * M_diabetes * A.M.inter ) +
  rnorm(N, mean = 0, sd = b["sd_Y"])

# data.frame
data.sim <- data.frame(L0_male, L0_soc_env, A0_PM2.5, L1, M_diabetes,
                      Y_death, Y_qol)

return( data.sim )
}

```

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

```

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)

```

10.3. SIMULATION OF THE FOUR DATA SETS USED IN EXAMPLES 191

```
head(df1)
```

```
##   L0_male L0_soc_env A0_PM2.5 L1 M_diabetes 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_soc_env A0_PM2.5 L1 M_diabetes 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_soc_env & sex=L0_male
  L0_male <- rbinom(N, size = 1, prob = b["p_L0_male"])
  L0_soc_env <- rbinom(N, size = 1, prob = b["p_L0_soc_env"])

  # exposure: A0_PM2.5
  A0_PM2.5 <- rbinom(N, size = 1, prob = b["b_A"] +
    b["b_male_A"] * L0_male +
    b["b_soc_env_A"] * L0_soc_env )

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

```

        b["b_soc_env_L1"] * L0_soc_env +
        b["b_A_L1"] * A0_PM2.5)

# mediator: M_diabetes
M_diabetes <- rbinom(N, size = 1, prob = b["b_M"] +
        b["b_male_M"] * L0_male +
        b["b_soc_env_M"] * L0_soc_env +
        b["b_A_M"] * A0_PM2.5 +
        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_soc_env_Y"] * L0_soc_env +
        b["b_A_Y"] * A0_PM2.5 +
        b["b_L1_Y"] * L1 +
        b["b_M_Y"] * M_diabetes +
        b["b_AM_Y"] * A0_PM2.5 * M_diabetes * A.M.inter )

# Y_qol
Y_qol <- ( b["mu_Y"] +
        b["c_male_Y"] * L0_male +
        b["c_soc_env_Y"] * L0_soc_env +
        b["c_A_Y"] * A0_PM2.5 +
        b["c_L1_Y"] * L1 +
        b["c_M_Y"] * M_diabetes +
        b["c_AM_Y"] * A0_PM2.5 * M_diabetes * A.M.inter ) +
        rnorm(N, mean = 0, sd = b["sd_Y"])

# data.frame
data.sim <- data.frame(L0_male, L0_soc_env, A0_PM2.5, L1, M_diabetes,
        Y_death, Y_qol)

return( data.sim )
}

```

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

```

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)

```


10.3. SIMULATION OF THE FOUR DATA SETS USED IN EXAMPLES193

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

```
tail(df2)
```

##	L0_male	L0_soc_env	A0_PM2.5	L1	M_diabetes	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

```
tail(df2_int)
```

##	L0_male	L0_soc_env	A0_PM2.5	L1	M_diabetes	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-outcome 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","soc_env","L1","M","sum")  
  for (n in 1:16) {  
    S[n,"sum"] <- ( ( b["b_Y"] +  
                     b["b_male_Y"] * S[n,"male"] +  
                     b["b_soc_env_Y"] * S[n,"soc_env"] +  
                     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_soc_env_M"] * S[n,"soc_env"] +
        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_soc_env_M"] * S[n,"soc_env"] +
        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_soc_env_Y"] * S[n,"soc_env"] +
        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_soc_env_M"] * S[n,"soc_env"] +
        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_soc_env_M"] * S[n,"soc_env"] +
        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"])^S[n,"L1"]))) *
((b["p_L0_male"])^S[n,"male"]))) *
((1 - b["p_L0_male"])^S[n,"male"]))) *
((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^S[n,"soc_env"])))
}

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","soc_env","L1","M","sum")
for (n in 1:16) {
  S[n,"sum"] <- ( ( ( b["mu_Y"] +
        b["c_male_Y"] * S[n,"male"] +
        b["c_soc_env_Y"] * S[n,"soc_env"] +
        b["c_A_Y"] * 1 +
        b["c_L1_Y"] * S[n,"L1"] +

```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
        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_soc_env_M"] * S[n,"soc_env"] +
        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_soc_env"]^(S[n,"soc_env"]))) *
    ((1 - b["p_L0_soc_env"]^(1 - S[n,"soc_env"])))
  }

  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 $ATE = E(Y_1) - E(Y_0)$ 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","soc_env","L1","M","sum")
  for (n in 1:16) {
    S[n,"sum"] <- ( ( b["b_Y"] +
                      b["b_male_Y"] * S[n,"male"] +
                      b["b_soc_env_Y"] * S[n,"soc_env"] +
                      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_soc_env_Y"] * S[n,"soc_env"] +
                     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_soc_env"])^(S[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))
  }

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

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env"])^S[n,"soc_env"]))) *
    ((1 - b["p_L0_soc_env"])^((1 - S[n,"soc_env"])))
}

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

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

```

Setting $do(M = 0)$, the controlled direct effects $CDE_{M=0} = \mathbb{E}(Y_{1,0}) - \mathbb{E}(Y_{0,0})$ 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 $CDE_{M=1} = \mathbb{E}(Y_{1,1}) - \mathbb{E}(Y_{0,1})$ 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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
                          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_soc_env_Y"] * S[n,"soc_env"] +
        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_soc_env_M"] * S[n,"soc_env"] +
        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_soc_env_M"] * S[n,"soc_env"] +
        b["b_L1_M"] * S[n,"L1"] +
        b["b_A_M"] * 0 ) ) ^ ( 1 - S[n,"M"] )) *

```


11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

((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_soc_env"])^S[n,"soc_env"])) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"]))

# TNIE
S[n,"sum.tnie"] <- ( b["b_Y"] +
                    b["b_male_Y"] * S[n,"male"] +
                    b["b_soc_env_Y"] * S[n,"soc_env"] +
                    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env"])^S[n,"soc_env"])) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"]))
}

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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env"])^(S[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"])))

  # TNIE
  S[n,"sum.tnie"] <- ( b["mu_Y"] +
    b["c_male_Y"] * S[n,"male"] +
    b["c_soc_env_Y"] * S[n,"soc_env"] +
    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"] ) *

```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

( (( b["b_M"] +
    b["b_male_M"] * S[n,"male"] +
    b["b_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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"])^S[n,"L1"]))) *
  ((b["p_L0_male"])^S[n,"male"]))) *
  ((1 - b["p_L0_male"])^S[n,"male"]))) *
  ((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
  ((1 - b["p_L0_soc_env"])^S[n,"soc_env"])))
}

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

- 0.05 and 0.008000000000000001 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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
                        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_soc_env_Y"] * S[n,"soc_env"] +
                        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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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"])) ) }
```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"])))

# PNIE
S[n,"sum.pnie"] <- ( b["b_Y"] +
                    b["b_male_Y"] * S[n,"male"] +
                    b["b_soc_env_Y"] * S[n,"soc_env"] +
                    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env"])^S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"])))
}

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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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"])^S[n,"L1"]))) *
    ((b["p_L0_male"])^S[n,"male"]))) *
    ((1 - b["p_L0_male"])^S[n,"male"]))) *
    ((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
    ((1 - b["p_L0_soc_env"])^S[n,"soc_env"])))

  # PNIE
  S[n,"sum.pnie"] <- ( b["mu_Y"] +
    b["c_male_Y"] * S[n,"male"] +
    b["c_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
      b["b_L1_M"] * S[n,"L1"] +

```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

      b["b_A_M"] * 1 )^( S[n,"M"] )) +
    (( 1 - (b["b_M"] +
      b["b_male_M"] * S[n,"male"] +
      b["b_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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"])^S[n,"L1"]))) *
    ((b["p_L0_male"])^S[n,"male"]))) *
    ((1 - b["p_L0_male"])^S[n,"male"]))) *
    ((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
    ((1 - b["p_L0_soc_env"])^S[n,"soc_env"])))
  }

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

- 0.05 and 0.008000000000000001 for death without interaction,
- 0.06155 and 0.008000000000000001 for death with interaction,
- -4 and -0.8999999999999999 for quality of life without interaction,
- -5.925 and -0.8999999999999999 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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
                        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_soc_env_Y"] * S[n,"soc_env"] +
                        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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_LO_male"] )^(S[n,"male"])) *
    ((1 - b["p_LO_male"] )^(1 - S[n,"male"])) *
    ((b["p_LO_soc_env"] )^(S[n,"soc_env"])) *
    ((1 - b["p_LO_soc_env"] )^(1 - S[n,"soc_env"]))
```


11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```
# PIE
S[n,"sum.pie"] <- ( b["b_Y"] +
  b["b_male_Y"] * S[n,"male"] +
  b["b_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env_M"] * S[n,"soc_env"] +
  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_soc_env_M"] * S[n,"soc_env"] +
  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_soc_env_M"] * S[n,"soc_env"] +
  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_soc_env_M"] * S[n,"soc_env"] +
  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_soc_env"])^S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"])))
}

# 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","soc_env","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_soc_env_Y"] * S.MI[n,"soc_env"] +
    b["b_A_Y"] * 1 +
    b["b_L1_Y"] * S.MI[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.MI[n,"male"] +
        b["b_soc_env_Y"] * S.MI[n,"soc_env"] +
        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_soc_env_Y"] * S.MI[n,"soc_env"] +
        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_soc_env_Y"] * S.MI[n,"soc_env"] +
        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_soc_env_M"] * S.MI[n,"soc_env"] +
        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_soc_env_M"] * S.MI[n,"soc_env"] +
            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_soc_env"])^ (S.MI[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])^ (1 - S.MI[n,"soc_env"]))
}

PDE.death <- sum(S[, "sum.pde"])
PIE.death <- sum(S[, "sum.pie"])
MI.death <- sum(S.MI[, "sum.mi"])

```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

# 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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env"])^(S[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"])))

  # PIE
  S[n,"sum.pie"] <- ( b["mu_Y"] +
    b["c_male_Y"] * S[n,"male"] +
    b["c_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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"])^S[n,"L1"]))) *
    ((b["p_L0_male"])^S[n,"male"]))) *
    ((1 - b["p_L0_male"])^S[n,"male"]))) *
    ((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
    ((1 - b["p_L0_soc_env"])^S[n,"soc_env"])))
  }

# 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","soc_env","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_soc_env_Y"] * S.MI[n,"soc_env"] +
    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_soc_env_Y"] * S.MI[n,"soc_env"] +
    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"] ) -

```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

      ( b["mu_Y"] +
        b["c_male_Y"] * S.MI[n,"male"] +
        b["c_soc_env_Y"] * S.MI[n,"soc_env"] +
        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_soc_env_Y"] * S.MI[n,"soc_env"] +
        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_soc_env_M"] * S.MI[n,"soc_env"] +
      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_soc_env_M"] * S.MI[n,"soc_env"] +
        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_soc_env"])^(S.MI[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])^(1 - S.MI[n,"soc_env"]))
  }

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 = $\mathbb{E}(Y_{1,M_0}) - \mathbb{E}(Y_{0,M_0})$, the PNIE = $\mathbb{E}(Y_{0,M_1}) - \mathbb{E}(Y_{0,M_0})$ and

the MIE = $\mathbb{E}((Y_{1,1} - Y_{1,0} - Y_{0,1} + Y_{0,0}) \times (M_1 - M_0))$ are respectively:

- 0.05, 0.008000000000000001 and 0.000 for death without interaction,
- 0.05855, 0.008000000000000001 and 0.003 for death with interaction,
- -4, -0.8999999999999999 and 0 for quality of life without interaction,
- -5.425, -0.8999999999999999 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","soc_env", "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_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_Y"] * S[n,"soc_env"] +
        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_soc_env"])^(S[n,"soc_env"])) *
      ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"])))
  }
```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

# INTref
S[n,"sum.intref"] <- ( ( b["b_Y"] +
  b["b_male_Y"] * S[n,"male"] +
  b["b_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env"])^(S[n,"soc_env"])) *
  ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))

# INTmed
S[n,"sum.intmed"] <- ( ( b["b_Y"] +
  b["b_male_Y"] * S[n,"male"] +
  b["b_soc_env_Y"] * S[n,"soc_env"] +

```

```

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_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env_M"] * S[n,"soc_env"] +
  b["b_L1_M"] * S[n,"L1"] +
  b["b_A_M"] * 1 ) -
  ( b["b_M"] +
    b["b_male_M"] * S[n,"male"] +
    b["b_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env"])^(S[n,"soc_env"])) *
((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))

# PIE
S[n,"sum.pie"] <- ( ( b["b_Y"] +
  b["b_male_Y"] * S[n,"male"] +

```



```

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_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env"])^(S[n,"soc_env"])) *
((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))

# INTref
S[n,"sum.intref"] <- ( ( b["mu_Y"] +
  b["c_male_Y"] * S[n,"male"] +
  b["c_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env_Y"] * S[n,"soc_env"] +
  b["c_A_Y"] * 0 +
  b["c_L1_Y"] * S[n,"L1"] +

```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

        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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env"])(S[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])(1 - S[n,"soc_env"]))

# INTmed
S[n,"sum.intmed"] <- ( ( b["mu_Y"] +
  b["c_male_Y"] * S[n,"male"] +
  b["c_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
      b["b_L1_M"] * S[n,"L1"] +
      b["b_A_M"] * 1 ) -
    ( b["b_M"] +
      b["b_male_M"] * S[n,"male"] +
      b["b_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env"])^(S[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))

# PIE
S[n,"sum.pie"] <- ( ( b["mu_Y"] +
      b["c_male_Y"] * S[n,"male"] +
      b["c_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      b["b_L1_M"] * S[n,"L1"] +
      b["b_A_M"] * 1 ) -
    ( b["b_M"] +
      b["b_male_M"] * S[n,"male"] +
      b["b_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env"])^(S[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))

```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```
((1 - b["p_L0_soc_env"])^ (1 - S[n, "soc_env"]))
}

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 = $\mathbb{E}(Y_{1,0}) - \mathbb{E}(Y_{0,0})$, RIE = $((Y_{1,1} - Y_{1,0} - Y_{0,1} + Y_{0,0}) \times M_0)$, MIE = $\mathbb{E}((Y_{1,1} - Y_{1,0} - Y_{0,1} + Y_{0,0}) \times (M_1 - M_0))$ and PNIE = $\mathbb{E}(Y_{0,M_1}) - \mathbb{E}(Y_{0,M_0})$ 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","soc_env","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_soc_env_M"] * M.S[n, "soc_env"] +
        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_soc_env_M"] * M.S[n, "soc_env"] +
        b["b_L1_M"] * M.S[n, "L1"] +
        b["b_A_M"] * M.S[n, "A"])))^( 1 - M.S[n, "M"] ))
}

M0.A0.LO_00.L1_0 <- M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==0 &
    M.S[, "soc_env"]==0 & M.S[, "L1"]==0, "sum"]
M0.A0.LO_01.L1_0 <- M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==0 &
    M.S[, "soc_env"]==1 & M.S[, "L1"]==0, "sum"]
M0.A0.LO_10.L1_0 <- M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==1 &
    M.S[, "soc_env"]==0 & M.S[, "L1"]==0, "sum"]
M0.A0.LO_11.L1_0 <- M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==1 &
    M.S[, "soc_env"]==1 & M.S[, "L1"]==0, "sum"]
M0.A0.LO_00.L1_1 <- M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==0 &
    M.S[, "soc_env"]==0 & M.S[, "L1"]==1, "sum"]
M0.A0.LO_01.L1_1 <- M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==0 &
    M.S[, "soc_env"]==1 & M.S[, "L1"]==1, "sum"]
M0.A0.LO_10.L1_1 <- M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==1 &
    M.S[, "soc_env"]==0 & M.S[, "L1"]==1, "sum"]
M0.A0.LO_11.L1_1 <- M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==1 &
    M.S[, "soc_env"]==1 & M.S[, "L1"]==1, "sum"]

M1.A0.LO_00.L1_0 <- M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==0 &
    M.S[, "soc_env"]==0 & M.S[, "L1"]==0, "sum"]
M1.A0.LO_01.L1_0 <- M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==0 &
    M.S[, "soc_env"]==1 & M.S[, "L1"]==0, "sum"]
M1.A0.LO_10.L1_0 <- M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==1 &
    M.S[, "soc_env"]==0 & M.S[, "L1"]==0, "sum"]
M1.A0.LO_11.L1_0 <- M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==1 &
    M.S[, "soc_env"]==1 & M.S[, "L1"]==0, "sum"]
M1.A0.LO_00.L1_1 <- M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==0 &
    M.S[, "soc_env"]==0 & M.S[, "L1"]==1, "sum"]
M1.A0.LO_01.L1_1 <- M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==0 &
    M.S[, "soc_env"]==1 & M.S[, "L1"]==1, "sum"]
M1.A0.LO_10.L1_1 <- M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==1 &
    M.S[, "soc_env"]==0 & M.S[, "L1"]==1, "sum"]
M1.A0.LO_11.L1_1 <- M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==1 &
    M.S[, "soc_env"]==1 & M.S[, "L1"]==1, "sum"]

M0.A1.LO_00.L1_0 <- M.S[M.S[, "M"]==0 & M.S[, "A"]==1 & M.S[, "male"]==0 &

```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

M.S[,"soc_env"]==0 & M.S[,"L1"]==0,"sum"]
M0.A1.LO_01.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==0 &
M.S[,"soc_env"]==1 & M.S[,"L1"]==0,"sum"]
M0.A1.LO_10.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==1 &
M.S[,"soc_env"]==0 & M.S[,"L1"]==0,"sum"]
M0.A1.LO_11.L1_0 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==1 &
M.S[,"soc_env"]==1 & M.S[,"L1"]==0,"sum"]
M0.A1.LO_00.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==0 &
M.S[,"soc_env"]==0 & M.S[,"L1"]==1,"sum"]
M0.A1.LO_01.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==0 &
M.S[,"soc_env"]==1 & M.S[,"L1"]==1,"sum"]
M0.A1.LO_10.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==1 &
M.S[,"soc_env"]==0 & M.S[,"L1"]==1,"sum"]
M0.A1.LO_11.L1_1 <- M.S[M.S[,"M"]==0 & M.S[,"A"]==1 & M.S[,"male"]==1 &
M.S[,"soc_env"]==1 & M.S[,"L1"]==1,"sum"]

M1.A1.LO_00.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==0 &
M.S[,"soc_env"]==0 & M.S[,"L1"]==0,"sum"]
M1.A1.LO_01.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==0 &
M.S[,"soc_env"]==1 & M.S[,"L1"]==0,"sum"]
M1.A1.LO_10.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==1 &
M.S[,"soc_env"]==0 & M.S[,"L1"]==0,"sum"]
M1.A1.LO_11.L1_0 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==1 &
M.S[,"soc_env"]==1 & M.S[,"L1"]==0,"sum"]
M1.A1.LO_00.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==0 &
M.S[,"soc_env"]==0 & M.S[,"L1"]==1,"sum"]
M1.A1.LO_01.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==0 &
M.S[,"soc_env"]==1 & M.S[,"L1"]==1,"sum"]
M1.A1.LO_10.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==1 &
M.S[,"soc_env"]==0 & M.S[,"L1"]==1,"sum"]
M1.A1.LO_11.L1_1 <- M.S[M.S[,"M"]==1 & M.S[,"A"]==1 & M.S[,"male"]==1 &
M.S[,"soc_env"]==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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
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,"soc_env"]==0)*(S[n,"L1"]==0) + # A'=1
  M1.A1.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M1.A1.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
  M1.A1.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M1.A1.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M1.A1.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
  M1.A1.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M1.A1.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*
  (S[n,"L1"]==1) )^( S[n,"M"] )) *
((M0.A1.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
  M0.A1.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M0.A1.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
  M0.A1.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M0.A1.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M0.A1.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
  M0.A1.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M0.A1.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==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_soc_env"])^ (S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"])))

S[n,"sum.psi10"] <- ( b["b_Y"] + # A=1
  b["b_male_Y"] * S[n,"male"] +
  b["b_soc_env_Y"] * S[n,"soc_env"] +
  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,"soc_env"]==0)*(S[n,"L1"]==0) + # A'=0
  M1.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M1.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
  M1.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M1.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M1.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
  M1.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M1.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*
  (S[n,"L1"]==1) )^( S[n,"M"] )) *
((M0.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
  M0.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M0.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
  M0.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M0.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M0.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
  M0.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M0.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1)*
  (S[n,"L1"]==1) )^( 1 - S[n,"M"] )) *

```


11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

M0.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
M0.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
M0.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
M0.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==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_soc_env"])^ (S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"])))

S[n,"sum.psi00"] <- ( b["b_Y"] + # A=0
b["b_male_Y"] * S[n,"male"] +
b["b_soc_env_Y"] * S[n,"soc_env"] +
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,"soc_env"]==0)*(S[n,"L1"]==0) + # A'=0
M1.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
M1.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
M1.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
M1.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
M1.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
M1.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
M1.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*
(S[n,"L1"]==1) )^( S[n,"M"] ) ) *
((M0.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
M0.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
M0.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
M0.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
M0.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
M0.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
M0.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
M0.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==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_soc_env"])^ (S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"])))
}

```

```

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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
    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,"soc_env"]==0)*(S[n,"L1"]==0) + # A'=1
    M1.A1.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
    M1.A1.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
    M1.A1.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
    M1.A1.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
    M1.A1.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
    M1.A1.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
    M1.A1.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*
    (S[n,"L1"]==1) )^( S[n,"M"] )) *
  ((M0.A1.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
    M0.A1.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
    M0.A1.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
    M0.A1.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
    M0.A1.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
    M0.A1.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
    M0.A1.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
    M0.A1.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==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_soc_env"])^(S[n,"soc_env"])) *
  ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"])))

  S[n,"sum.psi10"] <- ( b["mu_Y"] + # A=1
    b["c_male_Y"] * S[n,"male"] +
    b["c_soc_env_Y"] * S[n,"soc_env"] +
    b["c_A_Y"] * 1 +

```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

        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,"soc_env"]==0)*(S[n,"L1"]==0) + # A'=0
  M1.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M1.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
  M1.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M1.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M1.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
  M1.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M1.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*
  (S[n,"L1"]==1) )^( S[n,"M"] )) *
((M0.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
  M0.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M0.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
  M0.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M0.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M0.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
  M0.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M0.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==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_soc_env"])^ (S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"])))

S[n,"sum.psi00"] <- ( b["mu_Y"] + # A=0
  b["c_male_Y"] * S[n,"male"] +
  b["c_soc_env_Y"] * S[n,"soc_env"] +
  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,"soc_env"]==0)*(S[n,"L1"]==0) + # A'=0
  M1.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M1.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
  M1.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
  M1.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M1.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
  M1.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
  M1.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*
  (S[n,"L1"]==1) )^( S[n,"M"] )) *
((M0.A0.L0_00.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +

```

```

M0.A0.L0_01.L1_0*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
M0.A0.L0_10.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==0) +
M0.A0.L0_11.L1_0*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==0) +
M0.A0.L0_00.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
M0.A0.L0_01.L1_1*(S[n,"male"]==0)*(S[n,"soc_env"]==1)*(S[n,"L1"]==1) +
M0.A0.L0_10.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==0)*(S[n,"L1"]==1) +
M0.A0.L0_11.L1_1*(S[n,"male"]==1)*(S[n,"soc_env"]==1)*(S[n,"L1"]==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_soc_env"])^ (S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"])))
}

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 $MRDE = \mathbb{E}(Y_{1,G_{0|L(0)}}) - \mathbb{E}(Y_{0,G_{0|L(0)}})$ and the marginal randomized indirect effect $MRIE = \mathbb{E}(Y_{1,G_{1|L(0)}}) - \mathbb{E}(Y_{1,G_{0|L(0)}})$ are respectively:

- 0.05 and 0.00800000000000003 for death without interaction,
- 0.05855 and 0.011 for death with interaction,
- -3.999999999999999 and -0.9000000000000006 for quality of life without interaction,
- -5.424999999999999 and -1.400000000000001 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).

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
                        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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env"] ) ^ (S[n,"soc_env"]))) *
    ((1 - b["p_L0_soc_env"] ) ^ (1 - S[n,"soc_env"])))

    S[n,"sum.psi10"] <- ( b["b_Y"] + # A=1
                        b["b_male_Y"] * S[n,"male"] +
                        b["b_soc_env_Y"] * S[n,"soc_env"] +
                        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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env"])^(S[n,"soc_env"])) *
((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))

S[n,"sum.psi00"] <- ( b["b_Y"] +                                     # A=0
                    b["b_male_Y"] * S[n,"male"] +
                    b["b_soc_env_Y"] * S[n,"soc_env"] +
                    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_soc_env_M"] * S[n,"soc_env"] +
  b["b_L1_M"] * S[n,"L1"] +
  b["b_A_M"] * 0 ) ^ ( S[n,"M"] )) *                                     # A'=0
(( 1 - (b["b_M"] +
      b["b_male_M"] * S[n,"male"] +
      b["b_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env"])^(S[n,"soc_env"])) *
((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))
}

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","soc_env","L1","M","sum.psi11", "sum.psi10",
                  "sum.psi00")

```

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

```

for (n in 1:16) {
  S[n,"sum.psi11"] <- ( b["mu_Y"] +                                     # A=1
    b["c_male_Y"] * S[n,"male"] +
    b["c_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env"]^(S[n,"soc_env"]))) *
  ((1 - b["p_L0_soc_env"]^(1 - S[n,"soc_env"])))

  S[n,"sum.psi10"] <- ( b["mu_Y"] +                                     # A=1
    b["c_male_Y"] * S[n,"male"] +
    b["c_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env"])^ (S[n,"soc_env"])) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"]))

S[n,"sum.psi00"] <- ( b["mu_Y"] +                                     # A=0
                     b["c_male_Y"] * S[n,"male"] +
                     b["c_soc_env_Y"] * S[n,"soc_env"] +
                     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_soc_env_M"] * S[n,"soc_env"] +
   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_soc_env_M"] * S[n,"soc_env"] +
        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_soc_env"])^ (S[n,"soc_env"])) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"]))
}

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 $CRDE = \mathbb{E}(Y_{1,\Gamma_{0|L(0),L(1)}}) - \mathbb{E}(Y_{0,\Gamma_{0|L(0),L(1)}})$ and conditional randomized indirect effect $CRIE = \mathbb{E}(Y_{1,\Gamma_{1|L(0),L(1)}}) - \mathbb{E}(Y_{1,\Gamma_{0|L(0),L(1)}})$ are respectively:

11.1. TRUE CAUSAL QUANTITIES WITHOUT MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE

- 0.05 and 0.008000000000000003 for death without interaction,
- 0.05855 and 0.011 for death with interaction,
- -3.999999999999999 and -0.9000000000000006 for quality of life without interaction,
- -5.424999999999999 and -1.400000000000001 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	0.008000000000000001	0.011
Total NDE and Pure NIE		
- TNDE	0.05	0.06155
- PNIE	0.008000000000000001	0.008000000000000001
3-way decomposition		
- PDE	0.05	0.05855
- PIE	0.008000000000000001	0.008000000000000001
- 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	0.008000000000000003	0.011
Conditional randomized		
- conditional rNDE	0.05	0.05855
- conditional rNIE	0.008000000000000003	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	-0.8999999999999999	-0.8999999999999999
3-way decomposition		
- PDE	-4	-5.425
- PIE	-0.8999999999999999	-0.8999999999999999
- MI	0	-0.5

11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXPOSURE

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	-3.999999999999999	-5.424999999999999
- marginal rNIE	-0.9000000000000006	-1.4000000000000001
Conditional randomized		
- conditional rNDE	-3.999999999999999	-5.424999999999999
- conditional rNIE	-0.9000000000000006	-1.4000000000000001

11.2 True causal quantities with mediator-outcome 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","soc_env","L1","M","sum")
  for (n in 1:16) {
    S[n,"sum"] <- ( ( b["b_Y"] +
                      b["b_male_Y"] * S[n,"male"] +
                      b["b_soc_env_Y"] * S[n,"soc_env"] +
                      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_soc_env_M"] * S[n,"soc_env"] +
  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_soc_env_M"] * S[n,"soc_env"] +
  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_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 1)^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
  b["b_male_L1"] * S[n,"male"] +
  b["b_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) ) -
( ( b["b_Y"] +
  b["b_male_Y"] * S[n,"male"] +
  b["b_soc_env_Y"] * S[n,"soc_env"] +
  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_soc_env_M"] * S[n,"soc_env"] +
  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_soc_env_M"] * S[n,"soc_env"] +
  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_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 0)^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
  b["b_male_L1"] * S[n,"male"] +
  b["b_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) ) ) *
((b["p_L0_male"])^S[n,"male"]))) *

```

11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXPOSURE

```

    ((1 - b["p_L0_male"])^(1 - S[n,"male"])) *
    ((b["p_L0_soc_env"])^(S[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))
  }

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","soc_env","L1","M","sum")
for (n in 1:16) {
  S[n,"sum"] <- ( ( ( b["mu_Y"] +
    b["c_male_Y"] * S[n,"male"] +
    b["c_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_L1"] * S[n,"soc_env"] +
      b["b_A_L1"] * 1 )^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
      b["b_male_L1"] * S[n,"male"] +
      b["b_soc_env_L1"] * S[n,"soc_env"] +
      b["b_A_L1"] * 1 ) )^( 1 - S[n,"L1"] )) ) -
    ( ( b["mu_Y"] +
      b["c_male_Y"] * S[n,"male"] +
      b["c_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
        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_soc_env_M"] * S[n,"soc_env"] +
        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_soc_env_L1"] * S[n,"soc_env"] +
        b["b_A_L1"] * 0)^( S[n,"L1"] )) *
      (( 1 - ( b["b_L1"] +
        b["b_male_L1"] * S[n,"male"] +
        b["b_soc_env_L1"] * S[n,"soc_env"] +
        b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) ) *
      ((b["p_L0_male"])^S[n,"male"]))) *
      ((1 - b["p_L0_male"])^S[n,"male"]))) *
      ((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
      ((1 - b["p_L0_soc_env"])^S[n,"soc_env"])))
    }

    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 $ATE = E(Y_1) - E(Y_0)$ 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)

```

11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXPOSURE

```

# 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","soc_env","L1","M","sum")
for (n in 1:16) {
  S[n,"sum"] <- ( ( b["b_Y"] +
    b["b_male_Y"] * S[n,"male"] +
    b["b_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_L1"] * S[n,"soc_env"] +
      b["b_A_L1"] * 1 )^( S[n,"L1"] ) ) *
    ( ( 1 - ( b["b_L1"] +
      b["b_male_L1"] * S[n,"male"] +
      b["b_soc_env_L1"] * S[n,"soc_env"] +
      b["b_A_L1"] * 1 )^( 1 - S[n,"L1"] ) ) ) -
    ( ( b["b_Y"] +
      b["b_male_Y"] * S[n,"male"] +
      b["b_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_L1"] * S[n,"soc_env"] +
        b["b_A_L1"] * 0 )^( S[n,"L1"] ) ) *
      ( ( 1 - ( b["b_L1"] +
        b["b_male_L1"] * S[n,"male"] +
        b["b_soc_env_L1"] * S[n,"soc_env"] +
        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_soc_env"]))^(S[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"]))^(1 - S[n,"soc_env"]))
}

CDE.M0.death <- sum(S[1:8,"sum"])
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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_L1"] * S[n,"soc_env"] +
      b["b_A_L1"] * 1 ) ^ ( S[n,"L1"] ) ) *
    ( ( 1 - ( b["b_L1"] +
      b["b_male_L1"] * S[n,"male"] +
      b["b_soc_env_L1"] * S[n,"soc_env"] +
      b["b_A_L1"] * 1 ) ) ^ ( 1 - S[n,"L1"] ) ) ) ) *
    ( ( b["mu_Y"] +
      b["c_male_Y"] * S[n,"male"] +
      b["c_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_L1"] * S[n,"soc_env"] +
      b["b_A_L1"] * 0 ) ^ ( S[n,"L1"] ) ) *
    ( ( 1 - ( b["b_L1"] +
      b["b_male_L1"] * S[n,"male"] +
      b["b_soc_env_L1"] * S[n,"soc_env"] +
      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_soc_env"])^(S[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))
  }

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))

```


11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXP

```
}
```

```
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 $CDE_{M=0} = \mathbb{E}(Y_{1,0}) - \mathbb{E}(Y_{0,0})$ 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 $CDE_{M=1} = \mathbb{E}(Y_{1,1}) - \mathbb{E}(Y_{0,1})$ 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","soc_env","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_soc_env_M"] * M.S[n,"soc_env"] +
                      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_soc_env_M"] * M.S[n,"soc_env"] +
            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_soc_env_L1"] * M.S[n,"soc_env"] +
      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_soc_env_L1"] * M.S[n,"soc_env"] +
      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[, "soc_env"]==0, "sum")
M0.A0.L01 <- sum(M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==0 &
  M.S[, "soc_env"]==1, "sum")
M0.A0.L10 <- sum(M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==1 &
  M.S[, "soc_env"]==0, "sum")
M0.A0.L11 <- sum(M.S[M.S[, "M"]==0 & M.S[, "A"]==0 & M.S[, "male"]==1 &
  M.S[, "soc_env"]==1, "sum")

M1.A0.L00 <- sum(M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==0 &
  M.S[, "soc_env"]==0, "sum")
M1.A0.L01 <- sum(M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==0 &
  M.S[, "soc_env"]==1, "sum")
M1.A0.L10 <- sum(M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==1 &
  M.S[, "soc_env"]==0, "sum")
M1.A0.L11 <- sum(M.S[M.S[, "M"]==1 & M.S[, "A"]==0 & M.S[, "male"]==1 &
  M.S[, "soc_env"]==1, "sum")

M0.A1.L00 <- sum(M.S[M.S[, "M"]==0 & M.S[, "A"]==1 & M.S[, "male"]==0 &
  M.S[, "soc_env"]==0, "sum")
M0.A1.L01 <- sum(M.S[M.S[, "M"]==0 & M.S[, "A"]==1 & M.S[, "male"]==0 &
  M.S[, "soc_env"]==1, "sum")
M0.A1.L10 <- sum(M.S[M.S[, "M"]==0 & M.S[, "A"]==1 & M.S[, "male"]==1 &
  M.S[, "soc_env"]==0, "sum")
M0.A1.L11 <- sum(M.S[M.S[, "M"]==0 & M.S[, "A"]==1 & M.S[, "male"]==1 &
  M.S[, "soc_env"]==1, "sum")

M1.A1.L00 <- sum(M.S[M.S[, "M"]==1 & M.S[, "A"]==1 & M.S[, "male"]==0 &
  M.S[, "soc_env"]==0, "sum")
M1.A1.L01 <- sum(M.S[M.S[, "M"]==1 & M.S[, "A"]==1 & M.S[, "male"]==0 &
  M.S[, "soc_env"]==1, "sum")
M1.A1.L10 <- sum(M.S[M.S[, "M"]==1 & M.S[, "A"]==1 & M.S[, "male"]==1 &
  M.S[, "soc_env"]==0, "sum")
M1.A1.L11 <- sum(M.S[M.S[, "M"]==1 & M.S[, "A"]==1 & M.S[, "male"]==1 &
  M.S[, "soc_env"]==1, "sum")

```

11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXPOSURE

```

# 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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
                       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,"soc_env"]==0) + # A'=1
     M1.A1.L01*(S[n,"male"]==0)*(S[n,"soc_env"]==1) +
     M1.A1.L10*(S[n,"male"]==1)*(S[n,"soc_env"]==0) +
     M1.A1.L11*(S[n,"male"]==1)*(S[n,"soc_env"]==1) )^( S[n,"M"] )) *
    ((M0.A1.L00*(S[n,"male"]==0)*(S[n,"soc_env"]==0) +
     M0.A1.L01*(S[n,"male"]==0)*(S[n,"soc_env"]==1) +
     M0.A1.L10*(S[n,"male"]==1)*(S[n,"soc_env"]==0) +
     M0.A1.L11*(S[n,"male"]==1)*(S[n,"soc_env"]==1) )^( 1 - S[n,"M"] )) *
    (( b["b_L1"] + # A=1
      b["b_male_L1"] * S[n,"male"] +
      b["b_soc_env_L1"] * S[n,"soc_env"] +
      b["b_A_L1"] * 1)^( S[n,"L1"] )) *
    (( 1 - ( b["b_L1"] +
      b["b_male_L1"] * S[n,"male"] +
      b["b_soc_env_L1"] * S[n,"soc_env"] +
      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_soc_env"])^(S[n,"soc_env"])) *
    ((1 - b["p_L0_soc_env"])^(1 - S[n,"soc_env"]))

  S[n,"sum.psi10"] <- ( b["b_Y"] + # A=1
                       b["b_male_Y"] * S[n,"male"] +
                       b["b_soc_env_Y"] * S[n,"soc_env"] +
                       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,"soc_env"]==0) + # A'=0
     M1.A0.L01*(S[n,"male"]==0)*(S[n,"soc_env"]==1) +
     M1.A0.L10*(S[n,"male"]==1)*(S[n,"soc_env"]==0) +
     M1.A0.L11*(S[n,"male"]==1)*(S[n,"soc_env"]==1) )^( S[n,"M"] )) *

```

```

((M0.A0.L00*(S[n,"male"]==0)*(S[n,"soc_env"]==0) +
  M0.A0.L01*(S[n,"male"]==0)*(S[n,"soc_env"]==1) +
  M0.A0.L10*(S[n,"male"]==1)*(S[n,"soc_env"]==0) +
  M0.A0.L11*(S[n,"male"]==1)*(S[n,"soc_env"]==1) )^( 1 - S[n,"M"] )) *
(( b["b_L1"] +
  b["b_male_L1"] * S[n,"male"] +
  b["b_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 1)^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
  b["b_male_L1"] * S[n,"male"] +
  b["b_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) *
((b["p_L0_male"])^S[n,"male"]))) *
((1 - b["p_L0_male"])^S[n,"male"]))) *
((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^S[n,"soc_env"])))

S[n,"sum.psi00"] <- ( b["b_Y"] +
  b["b_male_Y"] * S[n,"male"] +
  b["b_soc_env_Y"] * S[n,"soc_env"] +
  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,"soc_env"]==0) +
  M1.A0.L01*(S[n,"male"]==0)*(S[n,"soc_env"]==1) +
  M1.A0.L10*(S[n,"male"]==1)*(S[n,"soc_env"]==0) +
  M1.A0.L11*(S[n,"male"]==1)*(S[n,"soc_env"]==1) )^( S[n,"M"] )) *
((M0.A0.L00*(S[n,"male"]==0)*(S[n,"soc_env"]==0) +
  M0.A0.L01*(S[n,"male"]==0)*(S[n,"soc_env"]==1) +
  M0.A0.L10*(S[n,"male"]==1)*(S[n,"soc_env"]==0) +
  M0.A0.L11*(S[n,"male"]==1)*(S[n,"soc_env"]==1) )^( 1 - S[n,"M"] )) *
(( b["b_L1"] +
  b["b_male_L1"] * S[n,"male"] +
  b["b_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 0)^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
  b["b_male_L1"] * S[n,"male"] +
  b["b_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) *
((b["p_L0_male"])^S[n,"male"]))) *
((1 - b["p_L0_male"])^S[n,"male"]))) *
((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^S[n,"soc_env"])))
}

```

11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXPOSURE

```

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", "soc_env", "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_soc_env_Y"] * S[n, "soc_env"] +
    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, "soc_env"]==0) +                # A'=1
    M1.A1.L01*(S[n, "male"]==0)*(S[n, "soc_env"]==1) +
    M1.A1.L10*(S[n, "male"]==1)*(S[n, "soc_env"]==0) +
    M1.A1.L11*(S[n, "male"]==1)*(S[n, "soc_env"]==1) )^( S[n, "M"] )) *
  ((M0.A1.L00*(S[n, "male"]==0)*(S[n, "soc_env"]==0) +
    M0.A1.L01*(S[n, "male"]==0)*(S[n, "soc_env"]==1) +
    M0.A1.L10*(S[n, "male"]==1)*(S[n, "soc_env"]==0) +
    M0.A1.L11*(S[n, "male"]==1)*(S[n, "soc_env"]==1) )^( 1 - S[n, "M"] )) *
  (( b["b_L1"] +                                                     # A=1
    b["b_male_L1"] * S[n, "male"] +
    b["b_soc_env_L1"] * S[n, "soc_env"] +
    b["b_A_L1"] * 1)^( S[n, "L1"] )) *
  (( 1 - ( b["b_L1"] +
    b["b_male_L1"] * S[n, "male"] +
    b["b_soc_env_L1"] * S[n, "soc_env"] +
    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_soc_env"] )^(S[n, "soc_env"]))) *
  ((1 - b["p_L0_soc_env"] )^(1 - S[n, "soc_env"])))

  S[n, "sum.psi10"] <- ( b["mu_Y"] +                                     # A=1
    b["c_male_Y"] * S[n, "male"] +
    b["c_soc_env_Y"] * S[n, "soc_env"] +
    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, "soc_env"]==0) +                # A'=0

```

```

M1.A0.L01*(S[n,"male"]==0)*(S[n,"soc_env"]==1) +
M1.A0.L10*(S[n,"male"]==1)*(S[n,"soc_env"]==0) +
M1.A0.L11*(S[n,"male"]==1)*(S[n,"soc_env"]==1) )^( S[n,"M"] )) *
((M0.A0.L00*(S[n,"male"]==0)*(S[n,"soc_env"]==0) +
M0.A0.L01*(S[n,"male"]==0)*(S[n,"soc_env"]==1) +
M0.A0.L10*(S[n,"male"]==1)*(S[n,"soc_env"]==0) +
M0.A0.L11*(S[n,"male"]==1)*(S[n,"soc_env"]==1) )^( 1 - S[n,"M"] )) *
(( b["b_L1"] +
  b["b_male_L1"] * S[n,"male"] +
  b["b_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 1)^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
  b["b_male_L1"] * S[n,"male"] +
  b["b_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 1))^( 1 - S[n,"L1"] )) *
((b["p_L0_male"])^S[n,"male"]))) *
((1 - b["p_L0_male"])^S[n,"male"]))) *
((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^S[n,"soc_env"])))

S[n,"sum.psi00"] <- ( b["mu_Y"] +
  b["c_male_Y"] * S[n,"male"] +
  b["c_soc_env_Y"] * S[n,"soc_env"] +
  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,"soc_env"]==0) +
M1.A0.L01*(S[n,"male"]==0)*(S[n,"soc_env"]==1) +
M1.A0.L10*(S[n,"male"]==1)*(S[n,"soc_env"]==0) +
M1.A0.L11*(S[n,"male"]==1)*(S[n,"soc_env"]==1) )^( S[n,"M"] )) *
((M0.A0.L00*(S[n,"male"]==0)*(S[n,"soc_env"]==0) +
M0.A0.L01*(S[n,"male"]==0)*(S[n,"soc_env"]==1) +
M0.A0.L10*(S[n,"male"]==1)*(S[n,"soc_env"]==0) +
M0.A0.L11*(S[n,"male"]==1)*(S[n,"soc_env"]==1) )^( 1 - S[n,"M"] )) *
(( b["b_L1"] +
  b["b_male_L1"] * S[n,"male"] +
  b["b_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 0)^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
  b["b_male_L1"] * S[n,"male"] +
  b["b_soc_env_L1"] * S[n,"soc_env"] +
  b["b_A_L1"] * 0))^( 1 - S[n,"L1"] )) *
((b["p_L0_male"])^S[n,"male"]))) *
((1 - b["p_L0_male"])^S[n,"male"])))

```

11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXPOSURE

```

      ((b["p_L0_soc_env"])^S[n,"soc_env"])) *
      ((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"])))
    }

    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 $MRDE = \mathbb{E}(Y_{1,G_{0|L(0)}}) - \mathbb{E}(Y_{0,G_{0|L(0)}})$ and the marginal randomized indirect effect $MRIE = \mathbb{E}(Y_{1,G_{1|L(0)}}) - \mathbb{E}(Y_{1,G_{0|L(0)}})$ are respectively:

- 0.064 and 0.0112 for death without interaction
- 0.073882 and 0.0154000000000001 for death with interaction
- -4.999999999999999 and -1.26 for quality of life without interaction
- -6.646999999999998 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", "soc_env", "L1", "M", "sum.psi11", "sum.psi10",
                     "sum.psi00")

  for (n in 1:16) {
    S[n, "sum.psi11"] <- ( b["b_Y"] +
                          b["b_male_Y"] * S[n, "male"] +
                          b["b_soc_env_Y"] * S[n, "soc_env"] +
                          b["b_A_Y"] * 1 +

```

A=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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_L1"] * S[n,"soc_env"] +
    b["b_A_L1"] * 1 )^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
    b["b_male_L1"] * S[n,"male"] +
    b["b_soc_env_L1"] * S[n,"soc_env"] +
    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_soc_env"])^ (S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"])^ (1 - S[n,"soc_env"])))

S[n,"sum.psi10"] <- ( b["b_Y"] +
    b["b_male_Y"] * S[n,"male"] +
    b["b_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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"] +

```

A'=1

A=1

A=1

A'=0

A=1

11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXPOSURE

```

      b["b_soc_env_L1"] * S[n,"soc_env"] +
      b["b_A_L1"] * 1)^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
      b["b_male_L1"] * S[n,"male"] +
      b["b_soc_env_L1"] * S[n,"soc_env"] +
      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_soc_env"])^S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"]^(1 - S[n,"soc_env"])))

S[n,"sum.psi00"] <- ( b["b_Y"] +                                     # A=0
      b["b_male_Y"] * S[n,"male"] +
      b["b_soc_env_Y"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_M"] * S[n,"soc_env"] +
      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_soc_env_L1"] * S[n,"soc_env"] +
      b["b_A_L1"] * 0)^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
      b["b_male_L1"] * S[n,"male"] +
      b["b_soc_env_L1"] * S[n,"soc_env"] +
      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_soc_env"])^S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"]^(1 - S[n,"soc_env"])))
}

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","soc_env","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_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_L1"] * S[n,"soc_env"] +
    b["b_A_L1"] * 1 )^( S[n,"L1"] )) *

  (( 1 - ( b["b_L1"] +
    b["b_male_L1"] * S[n,"male"] +
    b["b_soc_env_L1"] * S[n,"soc_env"] +
    b["b_A_L1"] * 1 ) )^( 1 - S[n,"L1"] )) ) *

  ((b["p_L0_male"])^S[n,"male"]))) *
  ((1 - b["p_L0_male"])^S[n,"male"]))) *
  ((b["p_L0_soc_env"])^S[n,"soc_env"]))) *
  ((1 - b["p_L0_soc_env"])^S[n,"soc_env"]))) *

  S[n,"sum.psi10"] <- ( b["mu_Y"] +                                     # A=1
    b["c_male_Y"] * S[n,"male"] +
    b["c_soc_env_Y"] * S[n,"soc_env"] +
    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"] +

```

11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXP

```

    b["b_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_L1"] * S[n,"soc_env"] +
    b["b_A_L1"] * 1 )^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
    b["b_male_L1"] * S[n,"male"] +
    b["b_soc_env_L1"] * S[n,"soc_env"] +
    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_soc_env"] )^(S[n,"soc_env"]))) *
((1 - b["p_L0_soc_env"] )^(1 - S[n,"soc_env"])))

S[n,"sum.psi00"] <- ( b["mu_Y"] +
    b["c_male_Y"] * S[n,"male"] +
    b["c_soc_env_Y"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_M"] * S[n,"soc_env"] +
    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_soc_env_L1"] * S[n,"soc_env"] +
    b["b_A_L1"] * 0 )^( S[n,"L1"] )) *
(( 1 - ( b["b_L1"] +
    b["b_male_L1"] * S[n,"male"] +
    b["b_soc_env_L1"] * S[n,"soc_env"] +

```

A=1

A=0

A'=0

A=0

```

      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_soc_env"]^(S[n,"soc_env"]))) *
      ((1 - b["p_L0_soc_env"]^(1 - S[n,"soc_env"])))
    }

    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 $CRDE = \mathbb{E}(Y_{1,\Gamma_{0|L(0),L(1)}}) - \mathbb{E}(Y_{0,\Gamma_{0|L(0),L(1)}})$ and conditional randomized indirect effect $CRIE = \mathbb{E}(Y_{1,\Gamma_{1|L(0),L(1)}}) - \mathbb{E}(Y_{1,\Gamma_{0|L(0),L(1)}})$ are respectively:

- 0.0672 and 0.008000000000000001 for death without interaction,
- 0.078282 and 0.011 for death with interaction,
- -5.36 and -0.9000000000000006 for quality of life without interaction,
- -7.207 and -1.4000000000000001 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

11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXPOSURE

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	0.008000000000000001	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	-4.999999999999999	-6.646999999999998
- marginal rNIE	-1.26	-1.96
Conditional randomized		
- conditional rNDE	-5.36	-7.207
- conditional rNIE	-0.9000000000000006	-1.4000000000000001

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.

- Díaz, I, N S Hejazi, K E Rudolph, and M J van Der Laan. 2020. “Nonparametric efficient causal mediation with intermediate confounders.” *Biometrika* 108 (3): 627–41. <https://doi.org/10.1093/biomet/asaa085>.
- Hejazi, Nima S., Kara E. Rudolph, and Iván Díaz. 2022. “‘Medoutcon’: Non-parametric Efficient Causal Mediation Analysis with Machine Learning in ‘r’.” *Journal of Open Source Software* 7 (69): 3979. <https://doi.org/10.21105/joss.03979>.
- Iacobucci, Dawn. 2008. *Mediation Analysis*. Thousand Oaks, California: Sage.
- Jak, Suzanne and Jorgensen, Terrence D. 2024. “A ‘Lavaan’ Compendium for Structural Equation Modeling in Educational Research.” <https://tdjorgensen.github.io/SEM-in-Ed-compendium/>.
- Jiang, Ge. 2023. “R Cookbook for Structural Equation Modeling.” <https://gabriellajg.github.io/EPsy-579-R-Cookbook-for-SEM/>.
- 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.
- Lange, Theis, Stijn Vansteelandt, and Maarten Bekaert. 2012. “A Simple Unified Approach for Estimating Natural Direct and Indirect Effects.” *American Journal of Epidemiology* 176 (3): 190–95.
- Lin, Sheng-Hsuan, Jessica Young, Roger Logan, Eric J. Tchetgen Tchetgen, and Tyler J VanderWeele. 2017. “Parametric Mediation 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.
- Nguyen, Quynh C., Theresa L. Osypuk, Nicole M. Schmidt, M. Maria Glymour, and Eric J. Tchetgen Tchetgen. 2015. “Practical Guidance for Conducting Mediation Analysis With Multiple Mediators Using Inverse Odds Ratio Weighting.” *American Journal of Epidemiology* 181 (5): 349–56. <https://doi.org/10.1093/aje/kwu278>.
- 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.
- Tchetgen Tchetgen, Eric J. 2013. “Inverse Odds Ratio-Weighted Estimation for Causal Mediation Analysis.” *Statistics in Medicine* 32 (26): 4567–80. <https://doi.org/https://doi.org/10.1002/sim.5864>.
- 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.” *Psychol Methods* 18 (2): 137–50.
- VanderWeele, Tyler J. 2009. “Marginal Structural Models for the Estimation

11.2. TRUE CAUSAL QUANTITIES WITH MEDIATOR-OUTCOME CONFOUNDER AFFECTED BY THE EXPOSURE

- 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.
- Zheng, Wenjing, and Mark van der Laan. 2017. “Longitudinal Mediation Analysis with Time-Varying Mediators and Exposures, with Application to Survival Outcomes.” *Journal of Causal Inference* 5 (2).