IMPERIAL

Exposomics Analytics - Practical 4 Structural Equation Modelling

Contents

Overview of the Practical

In this practical you will be introduced to the counterfactual or potential outcomes framework, which defines causal effects based on what would happen to an individual or population under different scenarios. Causal effects are estimated by comparing the observed (factual) outcome with hypothetical unobserved outcomes, i.e. what would have happened under different circumstances (counterfactual). The counterfactual framework hence answers the question, what would the average treatment effect be if everyone had been exposed or not exposed. First, we compare causal effect estimation using Classical Regression, MSM combined with IPTW, and G-computation before applying these estimation approaches to mediation analysis.

Marginal Structural Models (MSM) combined with Inverse Probability Treatment Weighting (IPTW), as well as G computation aim to estimate these counterfactual outcomes via different approaches;

Inverse Probability Treatment Weighting (IPTW)

IPTW effectively aims to simulate a randomised experiment of our observed population, by balancing the exposed/treated and unexposed/untreated groups in terms of their confounders. It re-weights individuals based on their likelihood (propensity) of having been exposed/treated. This approach results in more comparable treatment and control groups, mimicking a creating a pseudo-population that mimics randomization.

Imagine a group of 1,000 patients who have been prescribed a new drug for diabetes. Some of the patients take the drug, while others do not. We want to know whether taking the drug reduces the risk of developing a heart condition. However, there's a problem: the patients who took the drug are not the same as those who didn't.
For example: Some people who took the drug might have healthier lifestyles. Others who didn't take the drug might already have complications that make them more likely to develop heart conditions regardless of the drug. This situation introduces confounding factors (other factors like lifestyle, age, existing health issues) that might affect both the decision to take the drug and the likelihood of developing a heart condition. If we just compare the two groups (those who took the drug and those who didn't), the results will be biased because these groups are not equivalent in terms of their underlying health conditions.

IPTW "re-balances" the groups to figure out the true effect of the drug. We do this by weighting the patients. Patients who are more likely to have taken the drug (based on their characteristics like age, lifestyle, health history) are given less weight in the analysis, while patients who are less likely to have taken the drug are given more weight.

The weights are based on how likely a person was to receive the treatment. This likelihood is called the propensity score. Imagine this as answering the question: "Given all their characteristics (age, lifestyle, etc.), how likely was this person to take the drug?"
If a patient had a high chance of taking the drug (e.g., younger, healthier patients), we downweight their contribution to the analysis because they are already over-represented in the treatment group.
If a patient had a low chance of taking the drug (e.g., older, sicker patients), we upweight them because they are under-represented in the treatment group. This way, we make the groups more balanced regarding their characteristics.
Once we've applied the weights, we can compare the outcomes (heart condition development) between the two groups. This comparison gives us the causal effect of the drug, because now we've adjusted for the confounding factors that originally made the groups different.

Relative to tradtional adjustment methods, IPTW has its advantages and disadvantages: IPTW allows for more adaptable adjustments through individual re-weighting rather than assuming specific relationships like traditional adjustment in regression. It is more robust in the presence of many confounders or complex interactions since it doesn't require modeling every interaction. It can also handle the issue of time-dependent confounding, i.e. the collider problem: In standard regression models, including time-dependent confounders as covariates can inadvertently introduce collider bias if these confounders are affected by prior treatments, as this can create new confounding pathways between the treatment and the outcome.
However, IPTW requires careful calculation of weights based on propensity scores, which can introduce complexity and lead to biased results if incorrectly estimated; it is sensitive to extreme weights, causing instability and necessitating potential trimming that may introduce further bias. Additionally, it relies on the assumption of positivity, where all individuals must have a chance of receiving treatment, and if violated, it can lead to biased estimates. IPTW is also dependent on the correct specification of the propensity score model, making it more sensitive to misspecification than traditional adjustment methods.

G-computation

G-computation directly estimates the potential outcomes under different exposure scenarios by modeling the outcome Y as a function of the exposure X and confounders Z. After fitting this model, we use it to predict the outcome for each individual as if they were treated and as if they were not treated, allowing us to simulate the counterfactual outcomes.

- Observed Outcome: Y for the actual exposure X.
- Counterfactual Outcome: G-computation models the relationship between X, Z, and Y to predict $Y(1)$ and $Y(0)$.
- The average treatment effect (ATE) is estimated by averaging these predicted outcomes;

  $$E[Y(1)] - E[Y(0)]$$

  which represents the population-level causal effect.

Comparison of G-computation and Marginal Structural Model IPTW effect estimation.

In the following section, we will apply both G-computation and Marginal Structural Models combined with Inverse Probability Treatment Weighting on a simulated example;

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
library(questionr)
library(dagitty)
library(ggdag)
```

```
##
## Attaching package: 'ggdag'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(ggplot2)

## MSM IPTW

# Simulated data: Assume X is the exposure, Y is
# the outcome, and confounder Z
n <- 1000
Z <- rnorm(n)  # Confounder
X <- ifelse(runif(n) < plogis(Z), 1, 0)  # X influenced by Z
Y <- 2 * X + 3 * Z + rnorm(n)  # Outcome influenced by both X and Z
data <- as.data.frame(cbind(X, Z, Y))

# Step 1: Estimate the propensity score (the
# probability of X given Z)
propensity_model <- glm(X ~ Z, family = binomial, data = data)
data$pscore <- predict(propensity_model, type = "response")

# Step 2: Calculate the inverse probability
# weights
data$weight <- ifelse(data$X == 1, 1/data$pscore, 1/(1 -
```

```
    data$pscore))

# Step 3: Fit the MSM using IPTW
msm_model <- glm(Y ~ X, weights = data$weight, data = data)
summary(msm_model)
```

```
##
## Call:
## glm(formula = Y ~ X, data = data, weights = data$weight)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.04764    0.13686  -0.348    0.728
## X            2.15349    0.19342  11.134   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 18.53923)
##
##     Null deviance: 20800  on 999  degrees of freedom
## Residual deviance: 18502  on 998  degrees of freedom
## AIC: 5178.5
##
## Number of Fisher Scoring iterations: 2
```

```
# Comparison to unadjusted and adjusted model
msm_model_unadjusted <- glm(Y ~ X, data = data)
msm_model_adjusted <- glm(Y ~ X + Z, data = data)
summary(msm_model_unadjusted)
```

```
##
## Call:
## glm(formula = Y ~ X, data = data)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.254      0.128  -9.798   <2e-16 ***
## X              4.633      0.181  25.598   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 8.190132)
##
##     Null deviance: 13540.2  on 999  degrees of freedom
## Residual deviance:  8173.8  on 998  degrees of freedom
## AIC: 4944.8
##
## Number of Fisher Scoring iterations: 2
```

```
summary(msm_model_adjusted)
```

```
##
## Call:
## glm(formula = Y ~ X + Z, data = data)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03510    0.04826   0.727    0.467
## X            1.98826    0.07207  27.586   <2e-16 ***
## Z            3.03517    0.03670  82.704   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.042978)
##
##     Null deviance: 13540.2  on 999  degrees of freedom
## Residual deviance:  1039.8  on 997  degrees of freedom
## AIC: 2885
##
## Number of Fisher Scoring iterations: 2
```

```
## G-Computation

# Fit a model for G-computation
model <- lm(Y ~ X + Z, data = data)

# Predict potential outcomes
data$Y1 <- predict(model, newdata = data.frame(X = 1,
    Z = data$Z))  # Y if treated
```

```r
data$Y0 <- predict(model, newdata = data.frame(X = 0,
    Z = data$Z))  # Y if not treated

# Calculate average treatment effect (ATE)
ATE <- mean(data$Y1) - mean(data$Y0)
print(ATE)
```

```
## [1] 1.988256
```

Mediation Analysis

Different methods of estimating the Controlled Direct Effect (CDE) (effect of X on Y for a fixed value of M)

- Y00 denotes the potential outcome Y if X=0 and M=0
- Y10 denotes the potential outcome Y if X=1 and M=0
- Y01 denotes the potential outcome Y if X=0 and M=1
- Y11 denotes the potential outcome Y if X=1 and M=1

We are interested in the CDE of X=1 vs X=0 for a controlled value of M = 0
(We have fixed M=1, the result may vary depending on whether M=0 or M=1 is fixed, especially if we account for the presence of an Exposure * Mediator interaction)

Classical Regression Analysis

```
rm(list=ls()) # command that clears all objects from memory

# data import
data.init <- read.csv("base ihpaf simulee.txt", sep=";")
summary(data.init)
```

```
##        id            sexe           scol_bin          csp
##  Min.   :   1   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1176   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:0.0000
##  Median :2350   Median :0.0000   Median :1.0000   Median :1.0000
##  Mean   :2350   Mean   :0.3938   Mean   :0.7972   Mean   :0.6155
##  3rd Qu.:3525   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :4700   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##     comport           pas
##  Min.   :0.0000   Min.   : 73.72
##  1st Qu.:0.0000   1st Qu.:111.92
##  Median :0.0000   Median :121.74
##  Mean   :0.1198   Mean   :121.72
##  3rd Qu.:0.0000   3rd Qu.:131.22
##  Max.   :1.0000   Max.   :183.10
```

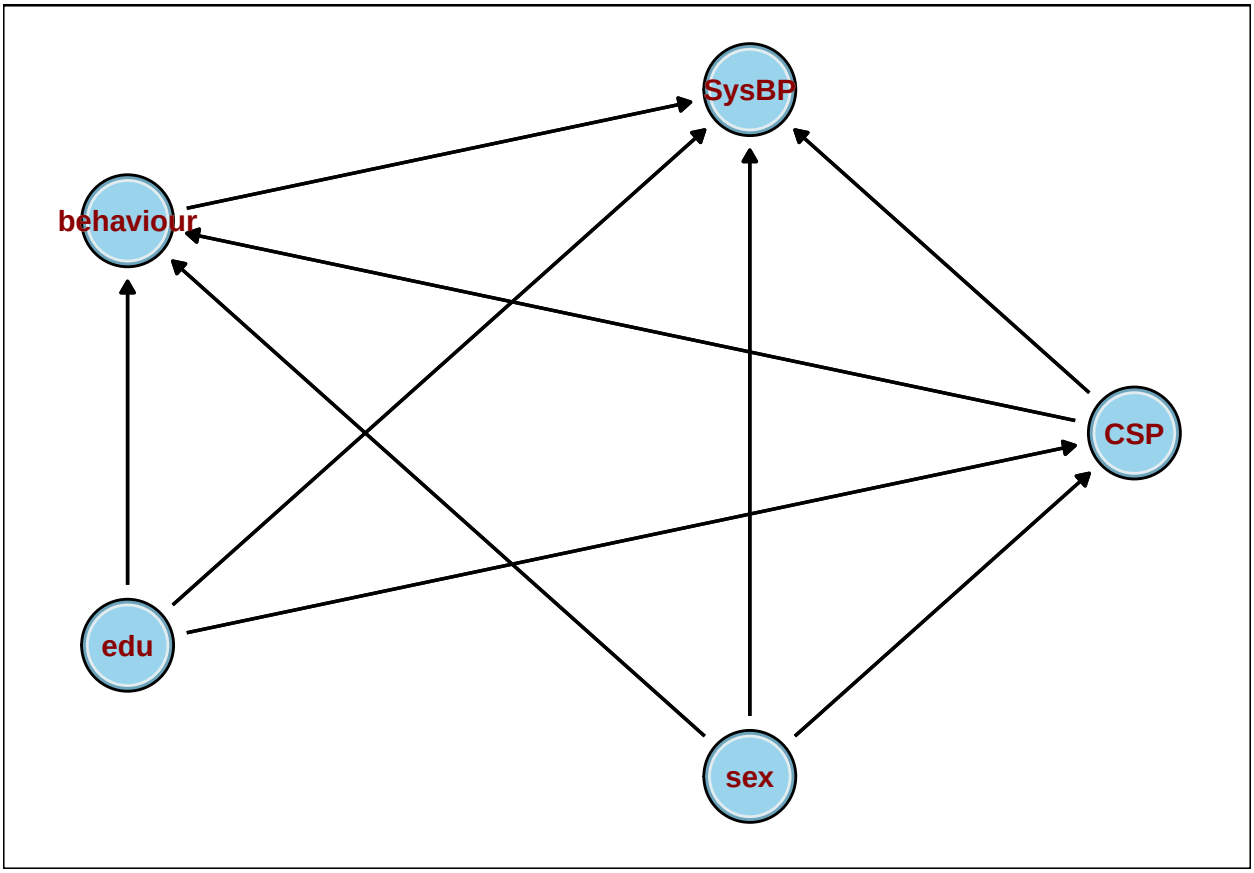
```
summary(data.init$pas)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   73.72  111.92  121.74  121.72  131.22  183.10
```

```r
# ############################################################################ #
#  I) description of causal relationships (if our hypotheses are correct)  ----
# ############################################################################ #

dag_social <- dagitty("dag {
  sex -> CSP
  edu -> CSP
  CSP -> behaviour
  sex -> behaviour
  edu -> behaviour
  behaviour -> SysBP
  CSP-> SysBP
  edu -> SysBP
  sex -> SysBP
}")

library(ggdag)

ggdag(dag_social, layout = "circle") +
  theme_dag_blank() +
  theme(
    plot.background = element_rect(fill = "white"), # Clean white background
    legend.position = "none"                        # Remove the legend
  ) +
  geom_dag_node(color = "skyblue", size = 15, alpha = 0.8) +     # Larger, softer-colored nodes
  geom_dag_edges(color = "gray50", size = 1) +                     # Softer edges
  geom_dag_text(color = "darkred", size = 4, fontface = "bold") # Improve text visibility
```

```
## Warning in geom_dag_edges_link(mapping, data = data_directed, arrow =
## arrow_directed, : Ignoring unknown parameters: `colour` and `size`
```

```
## Warning in geom_dag_edges_arc(mapping, data = data_bidirected, arrow =
## arrow_bidirected, : Ignoring unknown parameters: `colour` and `size`
```




```
######################################## #
## I.1) effect of sex on education ----
######################################## #
model.scol <- glm(scol_bin ~ sexe, family = "binomial", data = data.init)
model.scol$coefficients
```

```
## (Intercept)        sexe
##   1.4783340  -0.2647379
```

```
# to get the Odds Ratios (OR), we need to take the exponential
exp(model.scol$coefficients)
```

```
## (Intercept)        sexe
##   4.3856333   0.7674071
```

```
# (Intercept)        sexe
#   4.3856333   0.7674071

######################################################## #
## I.2) effect of sex and education on CSP (social class) ----
######################################################## #
```

```r
model.csp <- glm(csp ~ scol_bin + sexe, family = "binomial", data = data.init)
exp(model.csp$coefficients)
```

```
## (Intercept)    scol_bin        sexe
##   0.1665167  12.1711085   2.0905730
```

```r
# (Intercept)    scol_bin        sexe
#   0.1665167  12.1711085   2.0905730

############################################################################## #
## I.3) effect of sex, education, and CSP on behaviors ----
############################################################################## #
model.comport <- glm(comport ~ csp + scol_bin + sexe, family = "binomial", data = data.init)
exp(model.comport$coefficients)
```

```
## (Intercept)         csp    scol_bin        sexe
##  0.07626286  2.12017634  1.41218526  0.51840548
```

```r
# (Intercept)         csp    scol_bin        sexe
#  0.07626286  2.12017634  1.41218526  0.51840548

########################################################################################## #
## I.4) effect of sex, education, CSP, and behaviors on PAS (systolic blood pressure) ----
########################################################################################## #
model.pas <- glm(pas ~ comport + csp + scol_bin + sexe,
                 family = "gaussian",                  # here the outcome variable is quantitative = "g
                 data = data.init)
model.pas$coefficients  # no need for exponential transformation with a linear model
```

```
## (Intercept)     comport         csp    scol_bin        sexe
## 124.1425087   7.9701023   1.4502957   0.6700825 -12.1983451
```

```r
# (Intercept)     comport         csp    scol_bin        sexe
# 124.1425087   7.9701023   1.4502957   0.6700825 -12.1983451


# Note: this approach helps us roughly define the causal structure.
#       For each model, the estimate between the last explanatory variable
#       and the outcome variable can be interpreted as "causal effects."
#       However, the associations between the preceding explanatory variables
#       and the outcome variable are less clear.
```

```r
# ############################################################################ #
#  II) Mediation Analysis                                                  ----
# ############################################################################ #
#
# Total effect of education level on systolic blood pressure (PAS)
#
############################################################################### #
## II.1) Baron & Kenny method and the coefficient difference method           ----
############################################################################### #
# WARNING, THIS METHOD RELIES ON THE ASSUMPTION THAT SOCIOECONOMIC STATUS (CSP) IS NOT INFLUENCED BY ED
#                                                          => we assume that Figure 1a is correct.
#

########################################################### #
### II.1.A/ Applying the Baron & Kenny method ----
########################################################### #
## a) Model 1: X -> Y model to estimate the total effect
#
# propose a model for the total effect of education on PAS
#
# Total effect of X -> Y: average total effect = ATE = ?
model.effet.total <- glm(pas ~ scol_bin + sexe,
                         family = "gaussian",
                         data = data.init)
model.effet.total$coefficients
```

```
## (Intercept)    scol_bin        sexe
##  125.058943    1.957786  -12.441129
```

```r
model.effet.total$coefficients["scol_bin"]
```

```
## scol_bin
## 1.957786
```

```r
# average total effect, ATE = 1.957786 mmHg


## b) Model 2: The second model estimates the effect of the exposure on the mediator:
#
# Here, propose a model to estimate the effect of A on M
model.M.BaronKenny <- glm(comport ~ scol_bin + sexe, data = data.init, family = "binomial")
summary(model.M.BaronKenny)
```

```
##
```

```
## Call:
## glm(formula = comport ~ scol_bin + sexe, family = "binomial",
##     data = data.init)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.43158    0.13470 -18.052  < 2e-16 ***
## scol_bin     0.74766    0.13933   5.366 8.05e-08 ***
## sexe        -0.56059    0.09949  -5.635 1.75e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3445.1  on 4699  degrees of freedom
## Residual deviance: 3374.6  on 4697  degrees of freedom
## AIC: 3380.6
##
## Number of Fisher Scoring iterations: 5
```

```
exp(model.M.BaronKenny$coefficients["scol_bin"])
```

```
## scol_bin
##  2.11206
```

```
# scol_bin
# 2.11206
# Effect of A -> M: What measure of association? What estimate do you obtain?
# For the effect of A -> M: OR = 2.11, p-value < 0.001 => There is a significant effect of A -> M


## c) Model 3: Effect of education and behaviors on the mediator and on PAS, adjusted for confounders
# To properly evaluate the controlled direct effect under the assumption of Figure 1a,
# we need to use a model of PAS based on education and the mediator (behaviors),

# Propose a model for the outcome variable, adjusted for education (the first exposure), the mediator (
# and the appropriate confounders

model.Y.BaronKenny <- glm(pas ~ scol_bin + comport + sexe + csp, family = "gaussian",
                          data = data.init)
## The effect of M -> Y is significant (beta coefficient associated with behaviors = 7.97, p-value < 0.
summary(model.Y.BaronKenny)
```

```
##
```

```
## Call:
## glm(formula = pas ~ scol_bin + comport + sexe + csp, family = "gaussian",
##     data = data.init)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 124.1425     0.4549 272.910  < 2e-16 ***
## scol_bin      0.6701     0.5204   1.288 0.197909
## comport       7.9701     0.5825  13.683  < 2e-16 ***
## sexe        -12.1983     0.3893 -31.335  < 2e-16 ***
## csp           1.4503     0.4341   3.341 0.000842 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 164.1984)
##
##     Null deviance: 984255  on 4699  degrees of freedom
## Residual deviance: 770912  on 4695  degrees of freedom
## AIC: 37320
##
## Number of Fisher Scoring iterations: 2
```

```
# Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) 124.1425     0.4549 272.910  < 2e-16 ***
# scol_bin      0.6701     0.5204   1.288 0.197909      # p-value associated with education = 0.197909
# comport       7.9701     0.5825  13.683  < 2e-16 ***  # p-value associated with behaviors < 2e-16
# sexe        -12.1983     0.3893 -31.335  < 2e-16 ***
# csp           1.4503     0.4341   3.341 0.000842 ***
model.Y.BaronKenny$coefficients["comport"]
```

```
##  comport
## 7.970102
```

```
## beta_behaviors = 7.970102

# What is your conclusion regarding the effect of education on PAS and the mediation of this effect via

## The direct effect: EDC = 0.67, p-value = 0.20; we consider the effect to be "fully mediated" by alcol
model.Y.BaronKenny$coefficients["scol_bin"]
```

```
##  scol_bin
## 0.6700825
```

```
## beta_education = 0.6700825

# You can apply the coefficient difference to calculate the indirect effect:
# Indirect effect = total effect - direct effect
# = 1.957786 - 0.6700825 = 1.287704
model.effet.total$coefficients["scol_bin"] - model.Y.BaronKenny$coefficients["scol_bin"]
```

```
## scol_bin
## 1.287704
```

```
## 1.287704


# What are the values:
# - total effect: 1.957786
# - direct effect: 0.6700825
# - indirect effect: 1.287704


#########################################################################################################
### II.1.B/ Estimating a controlled direct effect using classical regression estimation (slide 49 of th
#########################################################################################################
# Create a model for the outcome variable as a function of the exposure and the mediator,
#                 adjusted for exposure-outcome confounders and mediator-outcome confounders
#                 (optionally, we can include an exposure*mediator interaction)

model.CDE.regress <- glm(pas ~ scol_bin + comport + sexe + csp + scol_bin*comport,
                         data = data.init,
                         family = "gaussian")
summary(model.CDE.regress)
```

```
##
## Call:
## glm(formula = pas ~ scol_bin + comport + sexe + csp + scol_bin *
##     comport, family = "gaussian", data = data.init)
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      124.2019     0.4657 266.710  < 2e-16 ***
## scol_bin           0.5928     0.5363   1.105 0.269071
## comport            7.0348     1.6713   4.209 2.61e-05 ***
## sexe             -12.1920     0.3895 -31.305  < 2e-16 ***
## csp                1.4480     0.4342   3.335 0.000859 ***
## scol_bin:comport   1.0635     1.7812   0.597 0.550503
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 164.2209)
##
##     Null deviance: 984255  on 4699  degrees of freedom
## Residual deviance: 770853  on 4694  degrees of freedom
## AIC: 37322
##
## Number of Fisher Scoring iterations: 2
```

```
# Coefficients:
#                 Estimate Std. Error t value Pr(>|t|)
# (Intercept)      124.2019     0.4657 266.710  < 2e-16 ***
# scol_bin           0.5928     0.5363   1.105 0.269071
# comport            7.0348     1.6713   4.209 2.61e-05 ***
# sexe             -12.1920     0.3895 -31.305  < 2e-16 ***
# csp                1.4480     0.4342   3.335 0.000859 ***
# scol_bin:comport   1.0635     1.7812   0.597 0.550503    # The interaction between exposure and media
##   The interaction between exposure and mediator is not significant => You can either remove it or ke
##   If you prefer to keep it:

## Controlled direct effect by setting M = 0 for the "behaviors" mediator:
(model.CDE.regress$coefficients["scol_bin"] + model.CDE.regress$coefficients["scol_bin:comport"]*0) * (
```

```
##  scol_bin
## 0.5927622
```

```
# CDE(m=0) = 0.5927622
# This corresponds to the direct effect of education on PAS, for a controlled value of behaviors where
# and no one is obese
# (This simulates a counterfactual scenario where a public health intervention eliminated excessive alc
#  obesity in the population)

## Controlled direct effect by setting M = 1 for the "behaviors" mediator:
(model.CDE.regress$coefficients["scol_bin"] + model.CDE.regress$coefficients["scol_bin:comport"]*1) * (
```

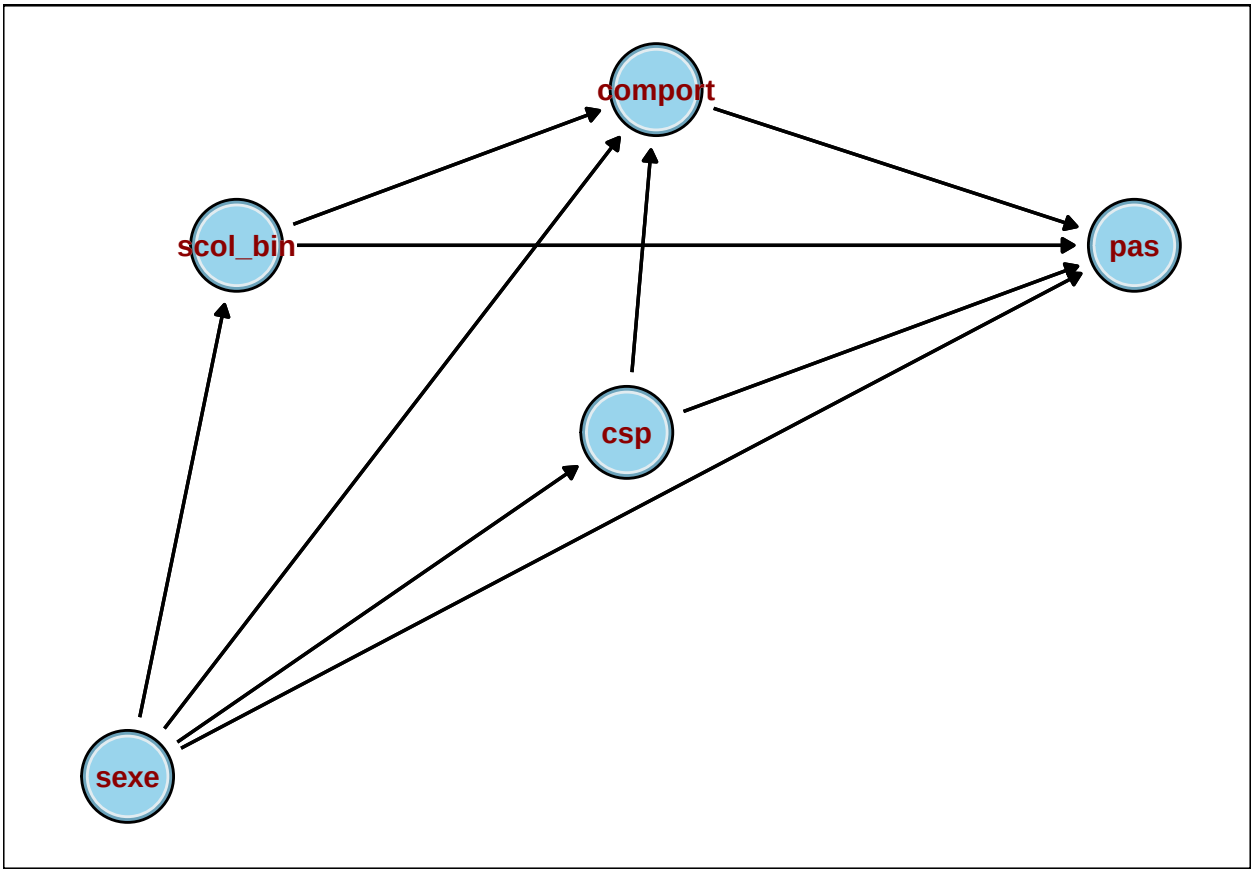
```
## scol_bin
## 1.656226
```

```
# CDE(m=1) = 1.656226
# This corresponds to the direct effect of education on PAS, for a controlled value of behaviors where
# to excessive alcohol consumption or obesity
# (This estimation is not very practical, as it's not an intervention we would want to implement)
```

```r
########################################################################################## #
## II.2 IN THE PRESENCE OF INTERMEDIATE CONFOUNDING INFLUENCED BY INITIAL EXPOSURE ----
########################################################################################## #
# => We assume that Figure 1b is correct +++
#    You can apply the following methods to estimate a controlled direct effect

# Using Dagitty, we could test if the assumption that "scol_bin" has no effect on "csp" is compatible w
## Dagitty under the assumption that there is no effect of scol_bin on csp
library(dagitty)
graph.hypot1 <- dagitty('dag {
  bb="0,0,1,1"
  comport [pos="0.419,0.735"]
  csp [pos="0.405,0.462"]
  pas [pos="0.647,0.611"]
  scol_bin [pos="0.219,0.611"]
  sexe [pos="0.167,0.188"]
  comport -> pas
  csp -> comport
  csp -> pas
  scol_bin -> comport
  scol_bin -> pas
  sexe -> comport
  sexe -> csp
  sexe -> pas
  sexe -> scol_bin
}')

ggdag(graph.hypot1, layout = "circle") +
  theme_dag_blank() +
  theme(
    plot.background = element_rect(fill = "white"), # Clean white background
    legend.position = "none"                        # Remove the legend
  ) +
  geom_dag_node(color = "skyblue", size = 15, alpha = 0.8) +     # Larger, softer-colored nodes
  geom_dag_edges(color = "gray50", size = 1) +                     # Softer edges
  geom_dag_text(color = "darkred", size = 4, fontface = "bold") # Improve text visibility
```

```
## Warning in geom_dag_edges_link(mapping, data = data_directed, arrow =
## arrow_directed, : Ignoring unknown parameters: `colour` and `size`
```

```
## Warning in geom_dag_edges_arc(mapping, data = data_bidirected, arrow =
## arrow_bidirected, : Ignoring unknown parameters: `colour` and `size`
```



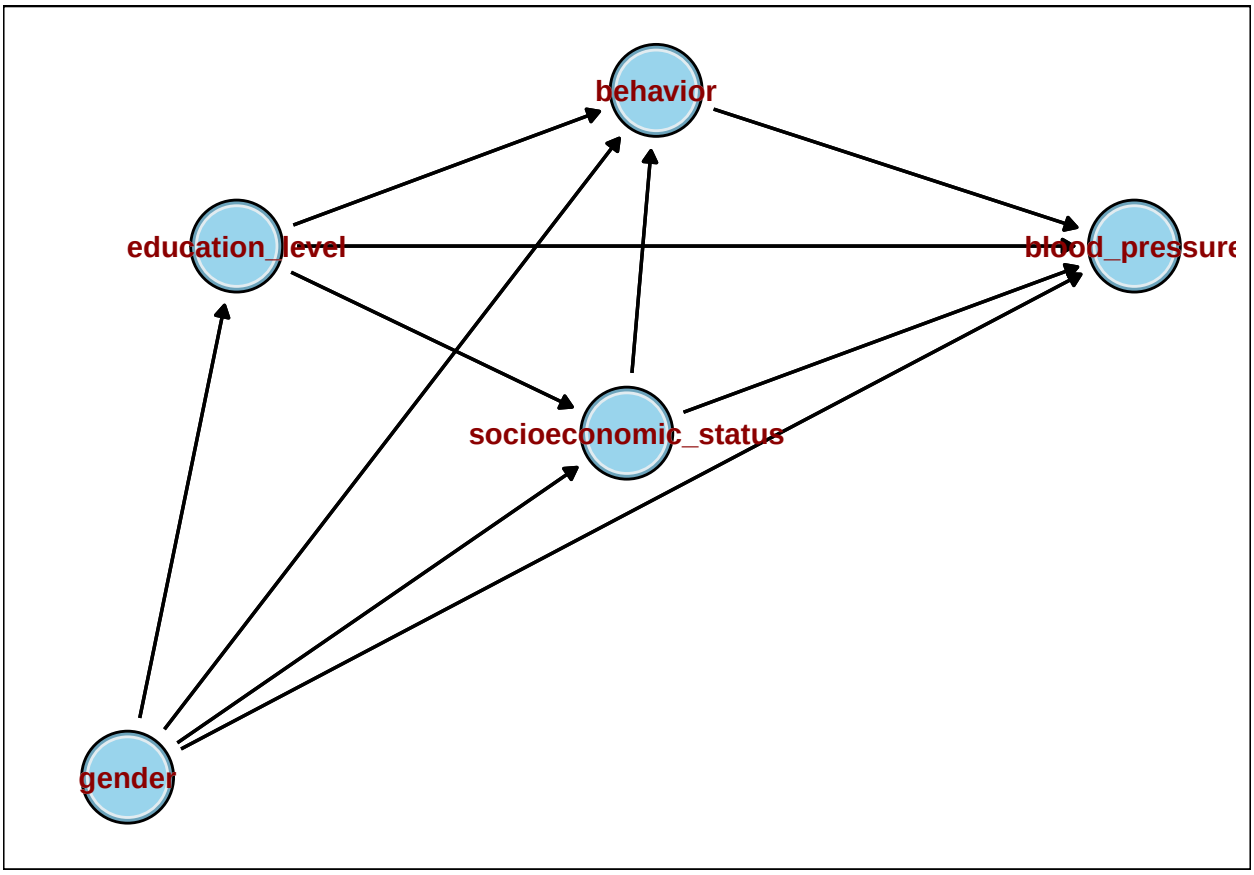

```
r <- localTests(graph.hypot1, data.init)
r
```

```
##                         estimate       p.value      2.5%     97.5%
## csp _||_ scl_ | sexe 0.4461556 3.530201e-237 0.4239253 0.4704176
```

```
#                         estimate       p.value      2.5%     97.5%
#   csp _||_ scl_ | sexe 0.4461556 3.530201e-237 0.4239253 0.4704176
# The test of independence between education and CSP (conditional on sex) is rejected (p < 0.0001)
# This assumption is not reasonable

graph.hypot2 <- dagitty('dag {
  bb="0,0,1,1"
  behavior [pos="0.419,0.735"]
  socioeconomic_status [pos="0.405,0.462"]
  blood_pressure [pos="0.647,0.611"]
  education_level [pos="0.219,0.611"]
  gender [pos="0.167,0.188"]
  behavior -> blood_pressure
  socioeconomic_status -> behavior
  socioeconomic_status -> blood_pressure
```

```
  education_level -> behavior
  education_level -> socioeconomic_status
  education_level -> blood_pressure
  gender -> behavior
  gender -> socioeconomic_status
  gender -> blood_pressure
  gender -> education_level
}')

ggdag(graph.hypot2, layout = "circle") +
  theme_dag_blank() +
  theme(
    plot.background = element_rect(fill = "white"), # Clean white background
    legend.position = "none"                        # Remove the legend
  ) +
  geom_dag_node(color = "skyblue", size = 15, alpha = 0.8) +     # Larger, softer-colored nodes
  geom_dag_edges(color = "gray50", size = 1) +                     # Softer edges
  geom_dag_text(color = "darkred", size = 4, fontface = "bold") # Improve text visibility
```

```
## Warning in geom_dag_edges_link(mapping, data = data_directed, arrow = arrow_directed, : Ignoring unkn
## Ignoring unknown parameters: `colour` and `size`
```



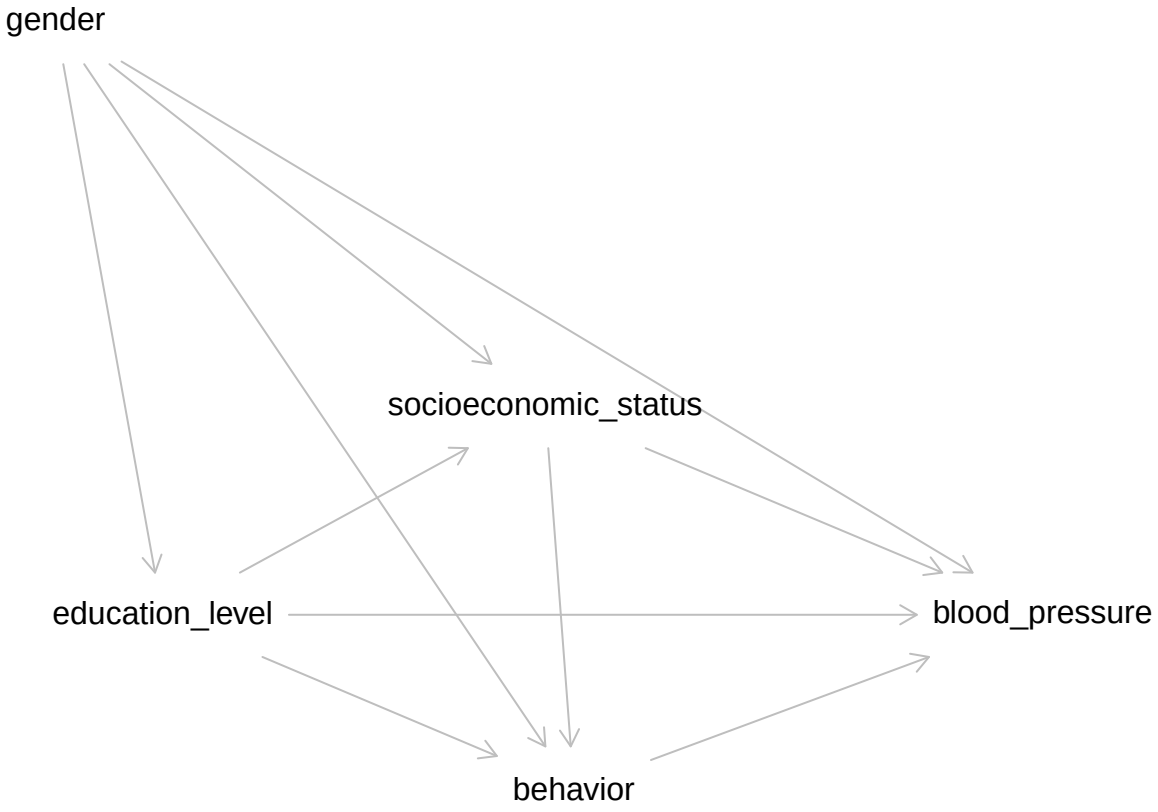
```r
r <- localTests(graph.hypot2, data.init)
r
```

```
## data frame with 0 columns and 0 rows
```

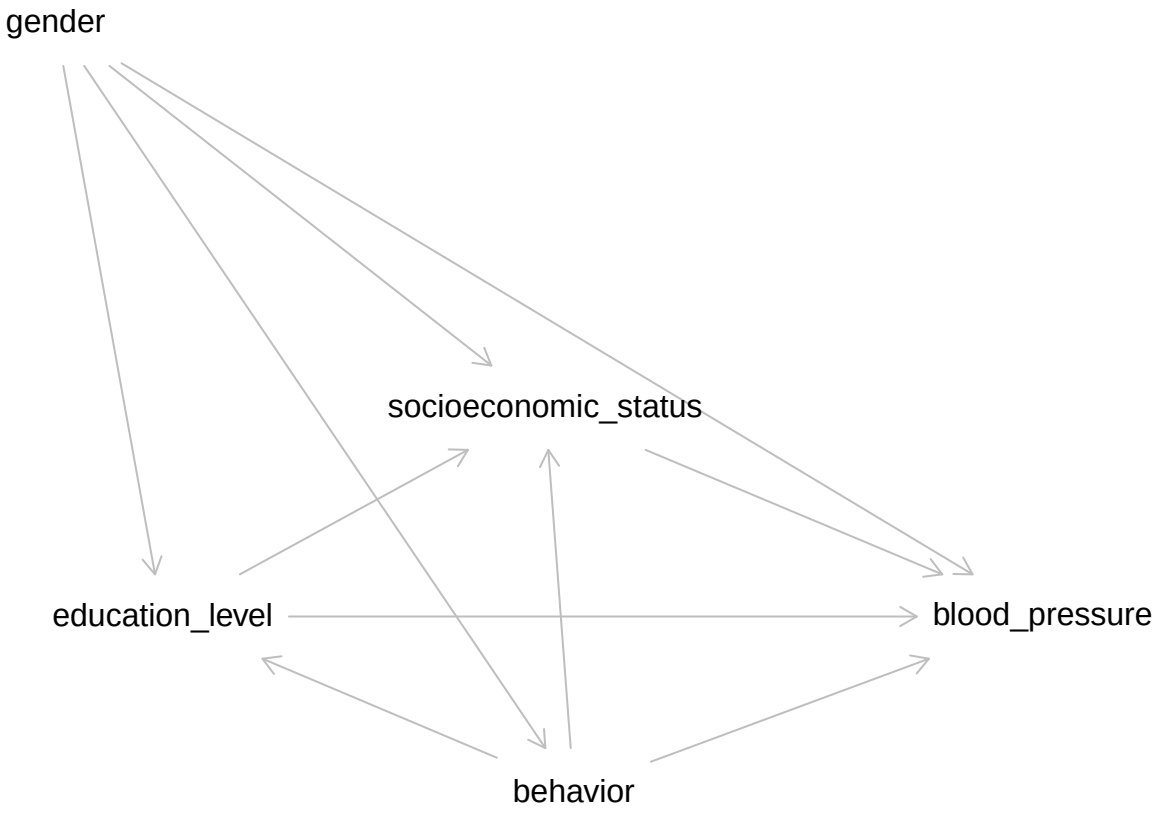
```r
# No testable implications (since no arrow is missing in this diagram, there is nothing to test)

length(equivalentDAGs(graph.hypot2))
```

```
## [1] 100
```

```r
# There are 100 equivalent DAGs !!!
plot(equivalentDAGs(graph.hypot2)[[1]]) ### this one is equivalent to
```




```r
plot(equivalentDAGs(graph.hypot2)[[21]]) ### we cannot distinguish from the data
```




```
### whether socioeconomic_status or behavior comes first in the mediation chain
```

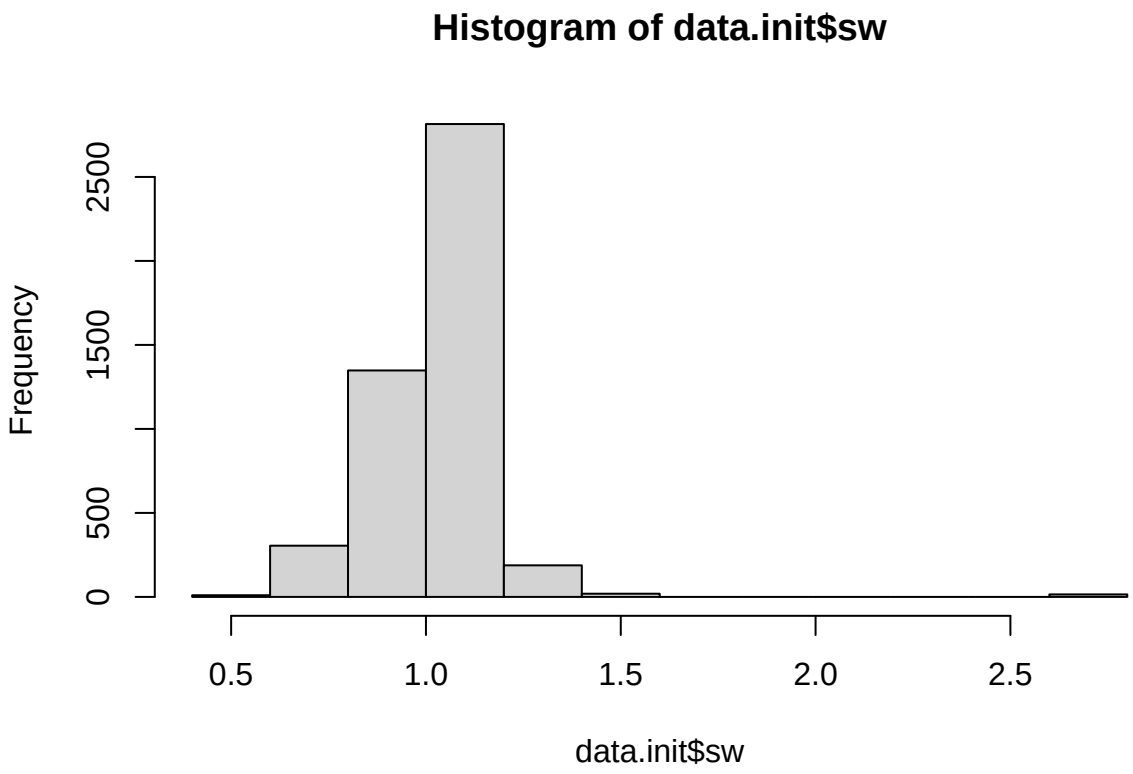
Marginal Structural Model IPTW

```
############################################################ #
### II.2.a) MSM by IPTW  ----
############################################################ #
## calculation of weights for education level
# numerator value
g.A.num <- glm(scol_bin ~ 1, family="binomial", data=data.init)
P.A1.num <- predict(g.A.num, type="response")
data.init$P.Aobs.num <- rep(NA, nrow(data.init))
data.init$P.Aobs.num[data.init$scol_bin == 1] <- P.A1.num[data.init$scol_bin == 1]
data.init$P.Aobs.num[data.init$scol_bin == 0] <- 1 - P.A1.num[data.init$scol_bin == 0]

# denominator value
g.AL.den <- glm(scol_bin ~ sexe, family="binomial", data=data.init)
P.A1.L.den <- predict(g.AL.den, type="response")
data.init$P.Aobs.den <- rep(NA, nrow(data.init))
data.init$P.Aobs.den[data.init$scol_bin == 1] <- P.A1.L.den[data.init$scol_bin == 1]
data.init$P.Aobs.den[data.init$scol_bin == 0] <- 1 - P.A1.L.den[data.init$scol_bin == 0]
```

```r
# stabilized weight (stabilized weight = numerator / denominator)
data.init$sw_scol <- data.init$P.Aobs.num / data.init$P.Aobs.den

## calculation of weights for the mediator (behavior)
# numerator
g.M.A.num <- glm(comport ~ scol_bin, family="binomial", data=data.init)
g.M1.A.num <- predict(g.M.A.num, type="response")
data.init$P.M.obs.num <- rep(NA, nrow(data.init))
data.init$P.M.obs.num[data.init$comport == 1] <- g.M1.A.num[data.init$comport == 1]
data.init$P.M.obs.num[data.init$comport == 0] <- 1 - g.M1.A.num[data.init$comport == 0]

# denominator
g.ML.den <- glm(comport ~ scol_bin + csp + sexe, family="binomial", data=data.init)
P.M1.AL.den <- predict(g.ML.den, type="response")
data.init$P.M.obs.den <- rep(NA, nrow(data.init))
data.init$P.M.obs.den[data.init$comport == 1] <- P.M1.AL.den[data.init$comport == 1]
data.init$P.M.obs.den[data.init$comport == 0] <- 1 - P.M1.AL.den[data.init$comport == 0]

data.init$sw_comport <- data.init$P.M.obs.num / data.init$P.M.obs.den

# individual weights
data.init$sw <- data.init$sw_scol * data.init$sw_comport
summary(data.init$sw)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5187  0.9397  1.0022  1.0002  1.0421  2.6096
```

```r
hist(data.init$sw)
```




```
## weighted regression to estimate the marginal structural model and controlled direct effect
# here, we force an interaction effect between the exposure and the mediator
msm_iptw <- glm(pas ~ scol_bin + comport + scol_bin * comport,
                weights = data.init$sw,                         # apply weighting (which ensures adjustm
                family = "gaussian", data = data.init)
EDC_m0 <- msm_iptw$coefficients["scol_bin"]
EDC_m0
```

```
## scol_bin
## 1.213183
```

```
# 1.213183
EDC_m1 <- msm_iptw$coefficients["scol_bin"] + msm_iptw$coefficients["scol_bin:comport"]
EDC_m1
```

```
## scol_bin
## 4.339364
```

```
# 1.213183 + 3.126181 = 4.339364

## effect of weights: creates a pseudo-population where confounding factors are balanced:
proportions(table(data.init$sexe, data.init$scol_bin), margin = 2)
```

```
##
```

```
##             0         1
##   0 0.5550892 0.6191620
##   1 0.4449108 0.3808380
```

```
proportions(wtd.table(x = data.init$sexe, y = data.init$scol_bin,
                      weights = data.init$sw),
            margin = 2)
```

```
##           0         1
## 0 0.6032955 0.6060539
## 1 0.3967045 0.3939461
```

```
proportions(table(data.init$sexe, data.init$comport), margin = 2)
```

```
##
##             0         1
##   0 0.5905245 0.7211368
##   1 0.4094755 0.2788632
```

```
proportions(wtd.table(x = data.init$sexe, y = data.init$comport,
                      weights = data.init$sw),
            margin = 2)
```

```
##           0         1
## 0 0.6060505 0.6013814
## 1 0.3939495 0.3986186
```

```
proportions(table(data.init$csp, data.init$comport), margin = 2)
```

```
##
##             0         1
##   0 0.4046410 0.2362345
##   1 0.5953590 0.7637655
```

```
proportions(wtd.table(x = data.init$csp, y = data.init$comport,
                      weights = data.init$sw),
            margin = 2)
```

```
##           0         1
## 0 0.3912617 0.3428173
## 1 0.6087383 0.6571827
```

```
proportions(table(data.init$csp[data.init$scol_bin == 0], data.init$comport[data.init$scol_bin == 0]), m
```

```
##
##             0         1
##   0 0.8123596 0.7142857
##   1 0.1876404 0.2857143
```

```
proportions(wtd.table(x = data.init$csp[data.init$scol_bin == 0], y = data.init$comport[data.init$scol_b
                      weights = data.init$sw[data.init$scol_bin == 0]),
            margin = 2)
```

```
##           0         1
## 0 0.8117744 0.8321495
## 1 0.1882256 0.1678505
```

```
proportions(table(data.init$csp[data.init$scol_bin == 1], data.init$comport[data.init$scol_bin == 1]), m
```

```
##
##             0         1
##   0 0.2928857 0.1760000
##   1 0.7071143 0.8240000
```

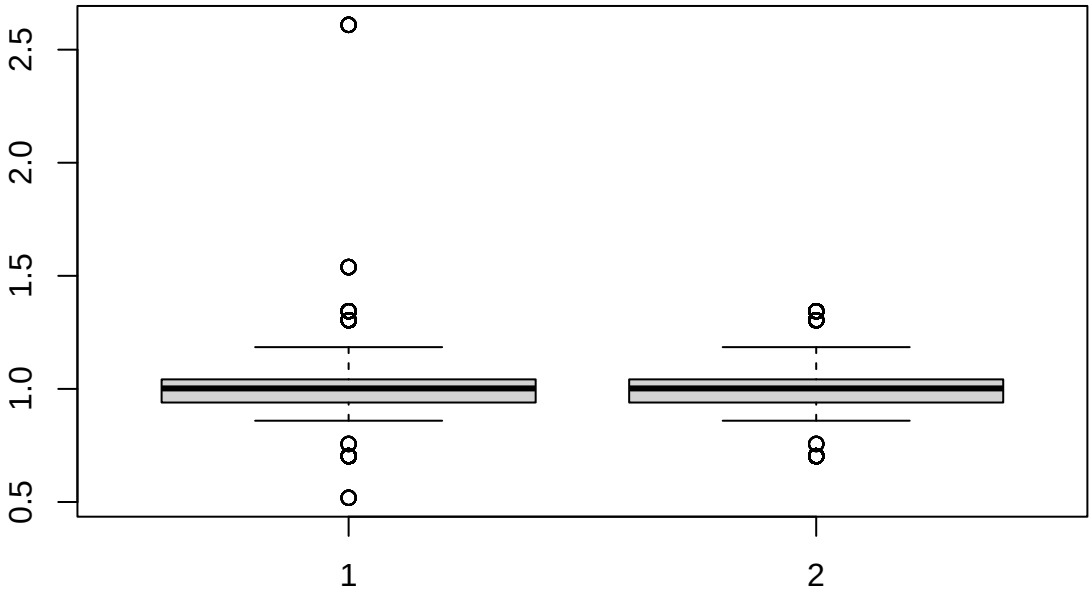
```
proportions(wtd.table(x = data.init$csp[data.init$scol_bin == 1], y = data.init$comport[data.init$scol_b
                      weights = data.init$sw[data.init$scol_bin == 1]),
            margin = 2)
```

```
##           0         1
## 0 0.2757819 0.2767400
## 1 0.7242181 0.7232600
```

```
# overall, there is better balance of comport based on csp
# in the subgroups defined by previous variables

############################################################ #
### II.2.b) MSM by IPTW with weight truncation ----
############################################################ #
p1_sw <- quantile(data.init$sw, probs = 0.01, na.rm = FALSE, type = 7)
p99_sw <- quantile(data.init$sw, probs = 0.99, na.rm = FALSE, type = 7)

data.init$sw199 <- data.init$sw
data.init$sw199[data.init$sw < p1_sw] <- p1_sw
data.init$sw199[data.init$sw > p99_sw] <- p99_sw

summary(data.init$sw199)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7028  0.9397  1.0022  0.9958  1.0421  1.3437
```

```r
boxplot(data.init$sw, data.init$sw199)
```




```r
# MSM with truncated weighting
msm_iptw_truncated <- glm(pas ~ scol_bin + comport + scol_bin * comport,
                          weights = data.init$sw199,                          # apply weighting (which en
                          family = "gaussian", data = data.init)
EDC_m0_tr <- msm_iptw_truncated$coefficients["scol_bin"]
EDC_m0_tr
```

```
## scol_bin
## 1.213183
```

```r
# 1.213183
EDC_m1_tr <- msm_iptw_truncated$coefficients["scol_bin"] + msm_iptw_truncated$coefficients["scol_bin:co
EDC_m1_tr
```

```
## scol_bin
## 4.094527
```

```r
# 1.213183 + 2.881344 = 4.094527
```

G-computation

```
########################################################## #
### II.3) g-computation ----
########################################################## #

########################################################################################################## #
##### IV.3.a) Parametric g-computation via Monte Carlo simulation ----
########################################################################################################## #

# set a seed for reproducibility of simulated results
set.seed(54321)

### 1) estimation of functional parameters
# model of intermediate confounding factors L(1)
model.csp <- glm(csp ~ scol_bin + sexe, family = "binomial", data = data.init)
# the model coefficients are accessible in the "coefficients" sub-object
model.csp$coefficients
```

```
## (Intercept)    scol_bin        sexe
##  -1.7926596   2.4990650   0.7374382
```

```
# (Intercept)    scol_bin        sexe
# -1.7926596   2.4990650   0.7374382

## model of the outcome Y (here, we force the interaction)
model.pas <- glm(pas ~ scol_bin + comport + csp + sexe + scol_bin*comport, family = "gaussian", data = d
# the model coefficients are accessible in the "coefficients" sub-object
model.pas$coefficients
```

```
##      (Intercept)         scol_bin          comport              csp
##      124.2019456        0.5927622        7.0348385        1.4480137
##             sexe scol_bin:comport
##      -12.1919762        1.0634640
```

```
# (Intercept)         scol_bin          comport              csp             sexe scol_bin:comport
# 124.2019456        0.5927622        7.0348385        1.4480137      -12.1919762        1.0634640

### 2.a) simulation of CSP for scol = 1
data.scol1 <- data.scol0 <- data.init
data.scol1$scol_bin <- rep(1, nrow(data.init))
data.scol0$scol_bin <- rep(0, nrow(data.init))

prob_csp1 <- predict(model.csp,
                     newdata = data.scol1, # simulate from a dataset where scol_bin = 1
                     type = "response")
```

```r
# this allows calculating the 4700 individual probabilities of having P(CSP = 1 | sexe, scol_bin == 1)
head(prob_csp1)
```

```
##         1         2         3         4         5         6
## 0.6696064 0.6696064 0.6696064 0.6696064 0.6696064 0.6696064
```

```r
tail(prob_csp1)
```

```
##      4695      4696      4697      4698      4699      4700
## 0.8090492 0.6696064 0.8090492 0.6696064 0.8090492 0.6696064
```

```r
csp1 <- rbinom(n = nrow(data.init), size = 1, prob = prob_csp1) # the rbinom function performs a draw a

### 2.b) simulation of CSP for scol = 0
prob_csp0 <- predict(model.csp,
                     newdata = data.scol0, # simulate from a dataset where scol_bin = 0
                     type = "response")
# this allows calculating the 4700 individual probabilities of having P(CSP = 1 | sexe, scol_bin == 0)
head(prob_csp0)
```

```
##        1        2        3        4        5        6
## 0.142747 0.142747 0.142747 0.142747 0.142747 0.142747
```

```r
tail(prob_csp0)
```

```
##      4695      4696      4697      4698      4699      4700
## 0.2582237 0.1427470 0.2582237 0.1427470 0.2582237 0.1427470
```

```r
csp0 <- rbinom(n = nrow(data.init), size = 1, prob = prob_csp0)

# preparing the databases for simulations under different counterfactual scenarios
data.00 <- data.01 <- data.10 <- data.11 <- data.init

# for scol = 1 and comport = 0
data.10$scol_bin <- rep(1, nrow(data.init))
data.10$csp <- csp1   # the expected value of CSP under the scenario with scol_bin == 1
data.10$comport <- rep(0, nrow(data.init))

# for scol = 1 and comport = 1
data.11$scol_bin <- rep(1, nrow(data.init))
data.11$csp <- csp1   # the expected value of CSP under the scenario with scol_bin == 1
data.11$comport <- rep(1, nrow(data.init))
```

```r
# for scol = 0 and comport = 0
data.00$scol_bin <- rep(0, nrow(data.init))
data.00$csp <- csp0   # the expected value of CSP under the scenario with scol_bin == 0
data.00$comport <- rep(0, nrow(data.init))

# for scol = 0 and comport = 1
data.01$scol_bin <- rep(0, nrow(data.init))
data.01$csp <- csp0   # the expected value of CSP under the scenario with scol_bin == 0
data.01$comport <- rep(1, nrow(data.init))

### 3.a) simulation of PAS for scol = 1 and comport = 0
pas10 <- predict(model.pas,
                 newdata = data.10,
                 type = "response")

### 3.b) simulation of PAS for scol = 1 and comport = 1
pas11 <- predict(model.pas,
                 newdata = data.11,
                 type = "response")

### 3.c) simulation of PAS for scol = 0 and comport = 0
pas00 <- predict(model.pas,
                 newdata = data.00,
                 type = "response")

### 3.d) simulation of PAS for scol = 0 and comport = 1
pas01 <- predict(model.pas,
                 newdata = data.01,
                 type = "response")

### 4) controlled direct effect:
E_pas00 <- mean(pas00)
E_pas00
```

```
## [1] 119.678
```

```r
# [1] 119.678

E_pas01 <- mean(pas01)
E_pas01
```

```
## [1] 126.7128
```

```r
# [1] 126.7128

E_pas10 <- mean(pas10)
E_pas10
```

```
## [1] 121.0487
```

```r
# [1] 121.0487

E_pas11 <- mean(pas11)
E_pas11
```

```
## [1] 129.147
```

```r
# [1] 129.147

#   * Controlled direct effect in g-computation fixing do(M=0)
ef_direct_gcomp_M0 = E_pas10 - E_pas00
ef_direct_gcomp_M0
```

```
## [1] 1.370684
```

```r
# [1] 1.370684

#   * Controlled direct effect in g-computation fixing do(M=1)
ef_direct_gcomp_M1 = E_pas11 - E_pas01
ef_direct_gcomp_M1
```

```
## [1] 2.434148
```

```r
# [1] 2.434148

############################################################ #
##### IV.3.b) g-computation using iterative regressions ----
############################################################ #
### 1) We start by creating a model of the PAS as a function of all previous factors
model_Y <- glm(pas ~ scol_bin + comport + csp + sexe + scol_bin*comport, data = data.init, family = "ga

# create datasets by fixing values A = (0 or 1) and M = (0 or 1) according to counterfactual scenarios
data.00 <- data.01 <- data.10 <- data.11 <- data.init

data.00$scol_bin <- data.00$comport <- rep(0, nrow(data.init))
```

```r
data.11$scol_bin <- data.11$comport <- rep(1, nrow(data.init))

data.10$scol_bin <- rep(1, nrow(data.init))
data.10$comport <- rep(0, nrow(data.init))

data.01$scol_bin <- rep(0, nrow(data.init))
data.01$comport <- rep(1, nrow(data.init))

### Fixing do(M=0)
# predict the average effect on the outcome if schooling = 0 and behavior = 0
Qbar_Y_scol_00 <- predict(model_Y,
                          newdata = data.00,
                          type = "response")

# predict the average effect on the outcome if schooling = 1 and behavior = 0
Qbar_Y_scol_10 <- predict(model_Y,
                          newdata = data.10,
                          type = "response")

### Fixing do(M=1)
# predict the average effect on the outcome if schooling = 0 and behavior = 1
Qbar_Y_scol_01 <- predict(model_Y,
                          newdata = data.01,
                          type = "response")

# predict the average effect on the outcome if schooling = 1 and behavior = 1
Qbar_Y_scol_11 <- predict(model_Y,
                          newdata = data.11,
                          type = "response")

### 2) Next step:
#   For each counterfactual intervention, we create a model of the predicted values as a function of th
#   for the counterfactual intervention do(A=0 & M=0): fix schooling = 0 (and M = 0)
model.L1.00 <- glm(Qbar_Y_scol_00 ~ scol_bin + sexe, data = data.init, family = "gaussian")
#   predict the average effect:
Qbar_L1_scol_00 <- predict(model.L1.00,
                           newdata = data.00,
                           type = "response")

# for the counterfactual intervention do(A=1 & M=0): fix schooling = 1 (and M = 0)
model.L1.10 <- glm(Qbar_Y_scol_10 ~ scol_bin + sexe, data = data.init, family = "gaussian")
#   predict the average effect
Qbar_L1_scol_10 <- predict(model.L1.10,
                           newdata = data.10,
```

```
                          type = "response")

# for the counterfactual intervention do(A=0 & M=1): fix schooling = 0 (and M = 1)
model.L1.01 <- glm(Qbar_Y_scol_01 ~ scol_bin + sexe, data = data.init, family = "gaussian")
#   predict the average effect
Qbar_L1_scol_01 <- predict(model.L1.01,
                          newdata = data.01,
                          type = "response")

# for the counterfactual intervention do(A=1 & M=1): fix schooling = 1 (and M = 1)
model.L1.11 <- glm(Qbar_Y_scol_11 ~ scol_bin + sexe, data = data.init, family = "gaussian")
#   predict the average effect
Qbar_L1_scol_11 <- predict(model.L1.11,
                          newdata = data.11,
                          type = "response")

### 3) Last step: estimate the controlled direct effect
E_pas00_iter <- mean(Qbar_L1_scol_00)
E_pas00_iter
```

```
## [1] 119.6715
```

```
# [1] 119.6715

E_pas10_iter <- mean(Qbar_L1_scol_10)
E_pas10_iter
```

```
## [1] 121.0422
```

```
# [1] 121.0422

E_pas01_iter <- mean(Qbar_L1_scol_01)
E_pas01_iter
```

```
## [1] 126.7064
```

```
# [1] 126.7064

E_pas11_iter <- mean(Qbar_L1_scol_11)
E_pas11_iter
```

```
## [1] 129.1405
```

```
# [1] 129.1405

# calculating controlled direct effects
ef_direct_gcomp_iter_M0 = E_pas10_iter - E_pas00_iter
ef_direct_gcomp_iter_M0
```

```
## [1] 1.370634
```

```
# [1] 1.370634

ef_direct_gcomp_iter_M1 = E_pas11_iter - E_pas01_iter
ef_direct_gcomp_iter_M1
```

```
## [1] 2.434098
```

```
# [1] 2.434098
```