First steps into CMAverse

Cécile Proust-Lima (Inserm, Univ. Bordeaux)

October 15, 2025

Contents

This is an introduction to the R package CMAverse. See the website with various vignettes: https://bs1125.github.io/CMAverse/

1 Installation of the R package

It is for now on github only. For the installation, install remotes R package if necessary and run:

```
library(remotes)
install_github("BS1125/CMAverse")
```

```
library(CMAverse)
```

2 Working dataset

We create a dataset that includes: 2 confounders at baseline C1 and C2, a binary treatment A, a binary mediator M1, a continuous mediator M2, and a continuous outcome Y

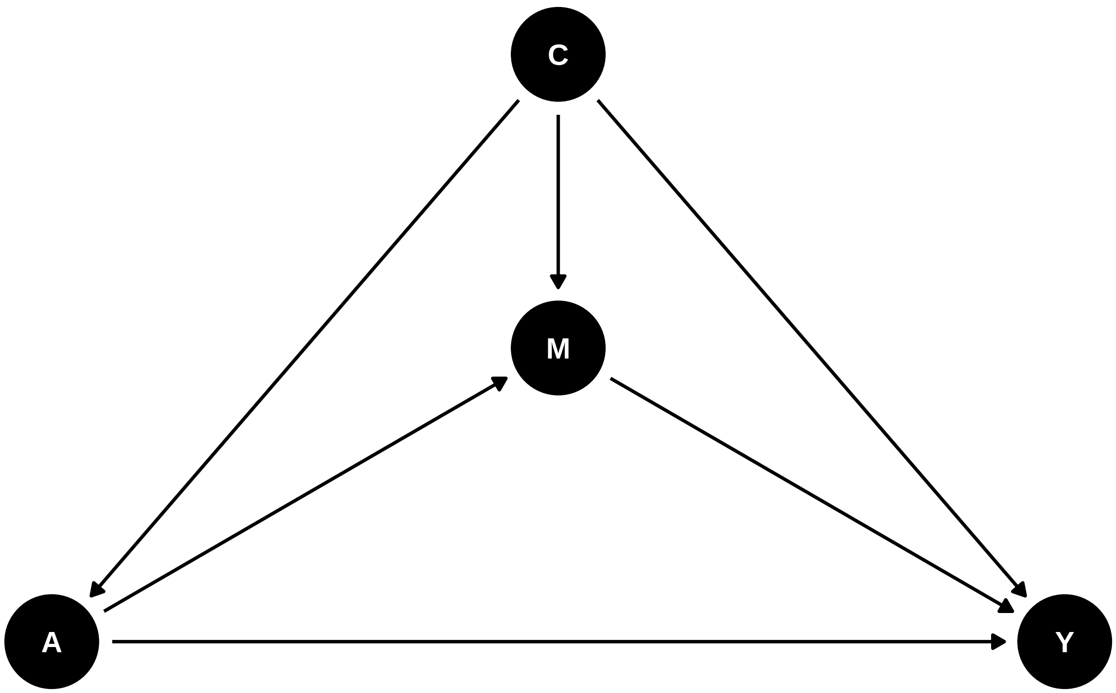
```
set.seed(1)
n <- 100
C1 <- rnorm(n, mean = 1, sd = 1)
C2 <- rbinom(n, 1, 0.6)
pa <- exp(0.2 - 0.5*C1 + 0.1*C2)/(1 + exp(0.2 - 0.5*C1 + 0.1*C2))
A <- rbinom(n, 1, pa)
```

```r
pm <- exp(1 + 0.5*A - 1.5*C1 + 0.5*C2)/ (1 + exp(1 + 0.5*A - 1.5*C1 + 0.5*C2))
M1 <- rbinom(n, 1, pm)
M2 <- rnorm(n, 2 + 0.8*A - M1 + 0.5*C1 + 2*C2, 1)
Y <- rnorm(n, mean = 0.5 + 0.4*A + 0.5*M1 + 0.6*M2 + 0.3*A*M1 + 0.5*A*M2 - 0.3*C1 + 2*C2, sd = 1)
dataSim <- data.frame(A, M1, M2, Y, C1, C2)
#save(dataSim, file="dataSim_for_Session3.Rdata")
```

3 Study of A - M2 - Y (neglecting M1)

3.1 We define the DAG

```r
cmdag(outcome = "Y", exposure = "A", mediator = c("M2"),
      basec = c("C1", "C2"))
```



A (exposure): A
M (mediator): M2
Y (outcome): Y
C (confounders not affected by the exposure): C1, C2

We could use classical regression tools to compute the causal effects as shown in the slides.

3.2 Step by step using glm and posterior computations

Estimation of the two regression models

```r
estRegY <- glm(formula = Y ~ A + M2 + A * M2 + C1 + C2, family = gaussian(), data = dataSim)
estRegM <- glm(formula = M2 ~ A + C1 + C2, family = gaussian(), data = dataSim)
summary(estRegY)
```

```
##
## Call:
## glm(formula = Y ~ A + M2 + A * M2 + C1 + C2, family = gaussian(),
```

```
##     data = dataSim)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1806     0.3977   0.454   0.6508
## A             1.5580     0.6507   2.394   0.0186 *
## M2            0.5485     0.1292   4.244 5.15e-05 ***
## C1           -0.2672     0.1398  -1.911   0.0590 .
## C2            2.8463     0.3351   8.495 2.90e-13 ***
## A:M2          0.2973     0.1515   1.962   0.0527 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.195193)
##
##     Null deviance: 796.20  on 99  degrees of freedom
## Residual deviance: 112.35  on 94  degrees of freedom
## AIC: 309.43
##
## Number of Fisher Scoring iterations: 2
```

```
summary(estRegM)
```

```
##
## Call:
## glm(formula = M2 ~ A + C1 + C2, family = gaussian(), data = dataSim)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6403     0.2564   6.397 5.75e-09 ***
## A             0.2901     0.2170   1.337    0.184
## C1            0.6208     0.1189   5.219 1.04e-06 ***
## C2            2.0833     0.2366   8.806 5.44e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.110528)
##
##     Null deviance: 225.08  on 99  degrees of freedom
## Residual deviance: 106.61  on 96  degrees of freedom
## AIC: 300.19
##
## Number of Fisher Scoring iterations: 2
```

Posterior computation of the regression-based natural effects:

```
theta <- coef(estRegY)
beta <- coef(estRegM)
EC1 <- mean(dataSim$C1)
EC2 <- mean(dataSim$C2)
a <- 1
astar <- 0
m <- 1
CDE <- (theta[2]+ theta[6]*m)*(a-astar)
NDE <- theta[2] + theta[6] * (beta[1] + beta[2] * astar + beta[3]*EC1 + beta[4]*EC2)*(a - astar)
NIE <- theta[3]*beta[2] + theta[6]*beta[2]*a
```

```r
TE <- NDE + NIE
data.frame(CDE, NDE, NIE, TE)
```

```
##        CDE      NDE       NIE       TE
## A 1.855251 2.696107 0.2453954 2.941502
```

3.3 Regressions using cmest function

We specify the model with cmest function using model="rb" (regression based). We have to specify the values for a and astar, and for the mediator (for CDE). To obtain the closed form estimates, we need to add estimation = "paramfunc" and inference = "delta" to obtain variance estimates using the Delta-Method.

```r
estRB <- cmest(data = dataSim, model = "rb",
               outcome = "Y",
               exposure = "A",
               mediator = c("M2"),
               basec = c("C1", "C2"), EMint = TRUE,
               mreg = list("linear"), yreg = "linear",
               astar = 0, a = 1, mval = list(1),
               estimation = "paramfunc", inference = "delta")
summary(estRB)
```

```
## Causal Mediation Analysis
##
## # Outcome regression:
##
## Call:
## glm(formula = Y ~ A + M2 + A * M2 + C1 + C2, 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)   0.1806     0.3977   0.454   0.6508
## A             1.5580     0.6507   2.394   0.0186 *
## M2            0.5485     0.1292   4.244 5.15e-05 ***
## C1           -0.2672     0.1398  -1.911   0.0590 .
## C2            2.8463     0.3351   8.495 2.90e-13 ***
## A:M2          0.2973     0.1515   1.962   0.0527 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.195193)
##
##     Null deviance: 796.20  on 99  degrees of freedom
## Residual deviance: 112.35  on 94  degrees of freedom
## AIC: 309.43
##
## Number of Fisher Scoring iterations: 2
##
##
## # Mediator regressions:
##
## Call:
## glm(formula = M2 ~ A + C1 + C2, family = gaussian(), data = getCall(x$reg.output$mreg[[1L]])$data,
##     weights = getCall(x$reg.output$mreg[[1L]])$weights)
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6403     0.2564   6.397 5.75e-09 ***
## A             0.2901     0.2170   1.337    0.184
## C1            0.6208     0.1189   5.219 1.04e-06 ***
## C2            2.0833     0.2366   8.806 5.44e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.110528)
##
##     Null deviance: 225.08  on 99  degrees of freedom
## Residual deviance: 106.61  on 96  degrees of freedom
## AIC: 300.19
##
## Number of Fisher Scoring iterations: 2
##
##
## # Effect decomposition on the mean difference scale via the regression-based approach
##
## 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           1.855251  0.511489  0.852752   2.858 0.000287 ***
## pnde          2.696107  0.232822  2.239784   3.152  < 2e-16 ***
## tnde          2.782354  0.232782  2.326109   3.239  < 2e-16 ***
## pnie          0.159148  0.124779 -0.085414   0.404 0.202153
## tnie          0.245395  0.187409 -0.121919   0.613 0.190394
## te            2.941502  0.275977  2.400598   3.482  < 2e-16 ***
## intref        0.840856  0.430513 -0.002933   1.685 0.050802 .
## intmed        0.086247  0.078048 -0.066725   0.239 0.269138
## cde(prop)     0.630715  0.160823  0.315509   0.946 8.79e-05 ***
## intref(prop)  0.285859  0.150207 -0.008541   0.580 0.057027 .
## intmed(prop)  0.029321  0.025405 -0.020471   0.079 0.248439
## pnie(prop)    0.054104  0.039881 -0.024060   0.132 0.174889
## pm            0.083425  0.059751 -0.033684   0.201 0.162647
## int           0.315180  0.163410 -0.005097   0.635 0.053759 .
## pe            0.369285  0.160823  0.054078   0.684 0.021663 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (cde: controlled direct effect; pnde: pure natural direct effect; tnde: total natural direct effect;
##
## Relevant variable values:
## $a
## [1] 1
##
## $astar
## [1] 0
##
## $mval
## $mval[[1]]
```

```
## [1] 1
##
##
## $basecval
## $basecval[[1]]
## [1] 1.108887
##
## $basecval[[2]]
## [1] 0.72
```

We find exactly the same effects as those computed step by step from glm estimates. Inded, the program performs exactly the same computations as above: two glms and derived closed-form solutions for the causal effects. With the add-in of correct standard errors thnaks to the Delta-Method.

There are many effects computed in the output. Here we focus on CDE, NDE (PNDE / TNDE), NIE (PNDE / TNDE), TE and PM (= TNIE / TE) only.

We can remove the others to not get confused with full=FALSE:

```
estRB2 <- cmest(data = dataSim, model = "rb", full=FALSE,
               outcome = "Y",
               exposure = "A",
               mediator = c("M2"),
               basec = c("C1", "C2"), EMint = TRUE,
               mreg = list("linear"), yreg = "linear",
               astar = 0, a = 1, mval = list(1),
               estimation = "paramfunc", inference = "delta")
summary(estRB2)
```

```
## Causal Mediation Analysis
##
## # Outcome regression:
##
## Call:
## glm(formula = Y ~ A + M2 + A * M2 + C1 + C2, 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)   0.1806     0.3977   0.454   0.6508
## A             1.5580     0.6507   2.394   0.0186 *
## M2            0.5485     0.1292   4.244 5.15e-05 ***
## C1           -0.2672     0.1398  -1.911   0.0590 .
## C2            2.8463     0.3351   8.495 2.90e-13 ***
## A:M2          0.2973     0.1515   1.962   0.0527 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.195193)
##
##     Null deviance: 796.20  on 99  degrees of freedom
## Residual deviance: 112.35  on 94  degrees of freedom
## AIC: 309.43
##
## Number of Fisher Scoring iterations: 2
##
```

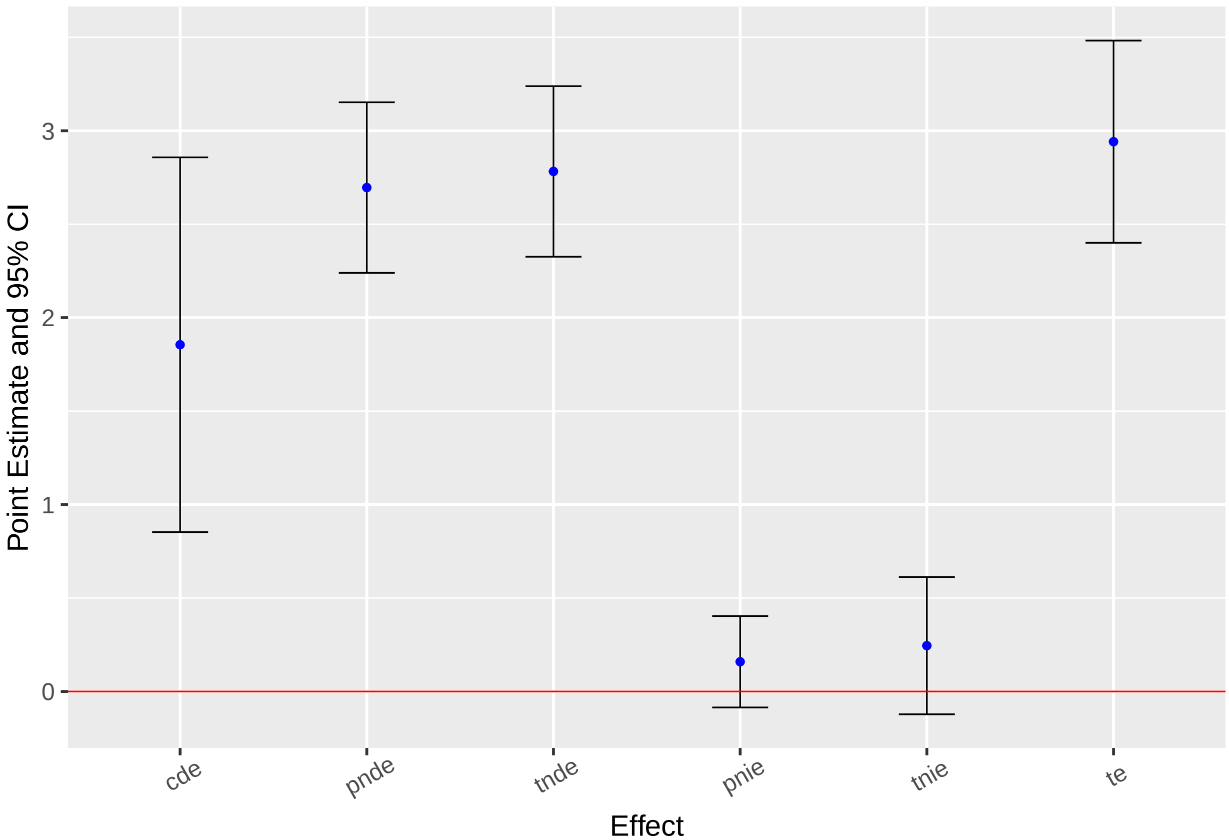
```
##
## # Mediator regressions:
##
## Call:
## glm(formula = M2 ~ A + C1 + C2, family = gaussian(), data = getCall(x$reg.output$mreg[[1L]])$data,
##     weights = getCall(x$reg.output$mreg[[1L]])$weights)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6403     0.2564   6.397 5.75e-09 ***
## A             0.2901     0.2170   1.337    0.184
## C1            0.6208     0.1189   5.219 1.04e-06 ***
## C2            2.0833     0.2366   8.806 5.44e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.110528)
##
##     Null deviance: 225.08  on 99  degrees of freedom
## Residual deviance: 106.61  on 96  degrees of freedom
## AIC: 300.19
##
## Number of Fisher Scoring iterations: 2
##
##
## # Effect decomposition on the mean difference scale via the regression-based approach
##
## 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   1.85525   0.51149  0.85275   2.858 0.000287 ***
## pnde  2.69611   0.23282  2.23978   3.152  < 2e-16 ***
## tnde  2.78235   0.23278  2.32611   3.239  < 2e-16 ***
## pnie  0.15915   0.12478 -0.08541   0.404 0.202153
## tnie  0.24540   0.18741 -0.12192   0.613 0.190394
## te    2.94150   0.27598  2.40060   3.482  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (cde: controlled direct effect; pnde: pure natural direct effect; tnde: total natural direct effect;
##
## Relevant variable values:
## $a
## [1] 1
##
## $astar
## [1] 0
##
## $mval
## $mval[[1]]
## [1] 1
##
##
```

```
## $basecval
## $basecval[[1]]
## [1] 1.108887
##
## $basecval[[2]]
## [1] 0.72
```

We can plot the results:

```
ggcmest(estRB2) + ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 30, vjust = 0.8))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## i The deprecated feature was likely used in the CMAverse package.
##   Please report the issue at <https://github.com/BS1125/CMAverse/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



3.4 What about other techniques?

We could use alternative techniques to estimate these effects

3.4.1 With natural effect (NE) model

The dataset is expanded using imputation technique for the "missing" counterfactual. Here, no need for a mediator model.

```r
estNE <- cmest(data = dataSim, model = "ne", full=FALSE,
               outcome = "Y",
               exposure = "A",
               mediator = c("M2"),
               basec = c("C1", "C2"), EMint = TRUE,
               yreg = "linear",
               astar = 0, a = 1, mval = list(1))
```

```
##   |                                                                              |
```

```r
summary(estNE)
```

```
## Causal Mediation Analysis
##
## # Outcome regression:
##
## Call:
## glm(formula = Y ~ A + M2 + A * M2 + C1 + C2, 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)   0.1806     0.3977   0.454   0.6508
## A             1.5580     0.6507   2.394   0.0186 *
## M2            0.5485     0.1292   4.244 5.15e-05 ***
## C1           -0.2672     0.1398  -1.911   0.0590 .
## C2            2.8463     0.3351   8.495 2.90e-13 ***
## A:M2          0.2973     0.1515   1.962   0.0527 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.195193)
##
##     Null deviance: 796.20  on 99  degrees of freedom
## Residual deviance: 112.35  on 94  degrees of freedom
## AIC: 309.43
##
## Number of Fisher Scoring iterations: 2
##
##
##
## # Effect decomposition on the mean difference scale via the natural effect 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   1.85525   0.61167  0.71554   3.015   0.01 **
## pnde  2.71902   0.24629  2.20536   3.155 <2e-16 ***
## tnde  2.75071   0.23884  2.30403   3.184 <2e-16 ***
## pnie  0.17456   0.14737 -0.08323   0.484   0.20
## tnie  0.20624   0.19853 -0.15665   0.596   0.28
## te    2.92527   0.30376  2.35618   3.505 <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (cde: controlled direct effect; pnde: pure natural direct effect; tnde: total natural direct effect;
##
## Relevant variable values:
## $a
## [1] 1
##
## $astar
## [1] 0
##
## $mval
## $mval[[1]]
## [1] 1
```

3.4.2 With the G-formula

We can use the G-formula.

```
estGform <- cmest(data = dataSim, model = "gformula", full=FALSE,
                outcome = "Y",
                exposure = "A",
                mediator = c("M2"),
                basec = c("C1", "C2"), EMint = TRUE,
                yreg = "linear", mreg=list("linear"),
                astar = 0, a = 1, mval = list(1))
```

```
##   |                                                                              |
```

This method uses the same regression models as for rb technique. But the estimates are numerically computed with Monte-Carlo and bootstrap (with 200 samples by default):

```
summary(estGform)
```

```
## Causal Mediation Analysis
##
## # Outcome regression:
##
## Call:
## glm(formula = Y ~ A + M2 + A * M2 + C1 + C2, 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)   0.1806     0.3977   0.454   0.6508
## A             1.5580     0.6507   2.394   0.0186 *
## M2            0.5485     0.1292   4.244 5.15e-05 ***
## C1           -0.2672     0.1398  -1.911   0.0590 .
## C2            2.8463     0.3351   8.495 2.90e-13 ***
## A:M2          0.2973     0.1515   1.962   0.0527 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.195193)
##
##     Null deviance: 796.20  on 99  degrees of freedom
```

```
## Residual deviance: 112.35  on 94  degrees of freedom
## AIC: 309.43
##
## Number of Fisher Scoring iterations: 2
##
##
## # Mediator regressions:
##
## Call:
## glm(formula = M2 ~ A + C1 + C2, family = gaussian(), data = getCall(x$reg.output$mreg[[1L]])$data,
##     weights = getCall(x$reg.output$mreg[[1L]])$weights)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6403     0.2564   6.397 5.75e-09 ***
## A             0.2901     0.2170   1.337    0.184
## C1            0.6208     0.1189   5.219 1.04e-06 ***
## C2            2.0833     0.2366   8.806 5.44e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.110528)
##
##     Null deviance: 225.08  on 99  degrees of freedom
## Residual deviance: 106.61  on 96  degrees of freedom
## AIC: 300.19
##
## Number of Fisher Scoring iterations: 2
##
##
## # 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   1.85525   0.54062  0.82050   2.765 <2e-16 ***
## pnde  2.67102   0.23699  2.26749   3.164 <2e-16 ***
## tnde  2.75726   0.22449  2.37546   3.207 <2e-16 ***
## pnie  0.15915   0.13805 -0.03655   0.488   0.13
## tnie  0.24540   0.18406 -0.05238   0.633   0.13
## te    2.91641   0.26086  2.47716   3.480 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (cde: controlled direct effect; pnde: pure natural direct effect; tnde: total natural direct effect;
##
## Relevant variable values:
## $a
## [1] 1
##
## $astar
## [1] 0
##
```

```
## $mval
## $mval[[1]]
## [1] 1
```

In this specific setting, the call with model = “rb”, and inference=“bootstrap” is equivalent to model = “gformula”. This is not always the case: the g-formula also handles exposure-affected confounders for M-Y relation.

This can be checked by setting the seed for the Bootstrap:

```
set.seed(1)
estRBBoot <- cmest(data = dataSim, model = "rb", full=FALSE,
                   outcome = "Y",
                   exposure = "A",
                   mediator = c("M2"),
                   basec = c("C1", "C2"), EMint = TRUE,
                   mreg = list("linear"), yreg = "linear",
                   astar = 0, a = 1, mval = list(1), inference = "bootstrap")
```

```
##   |                                                                              |
```

```
set.seed(1)
estGformBoot <- cmest(data = dataSim, model = "gformula", full=FALSE,
                   outcome = "Y",
                   exposure = "A",
                   mediator = c("M2"),
                   basec = c("C1", "C2"), EMint = TRUE,
                   mreg = list("linear"), yreg = "linear",
                   astar = 0, a = 1, mval = list(1))
```

```
##   |                                                                              |
```

```
Compar <- cbind(estRBBoot$effect.pe, estRBBoot$effect.se, estGformBoot$effect.pe, estGformBoot$effect.se
colnames(Compar) <- c("rb-boot","SE rb-boot","Gform","SE Gform")
Compar
```

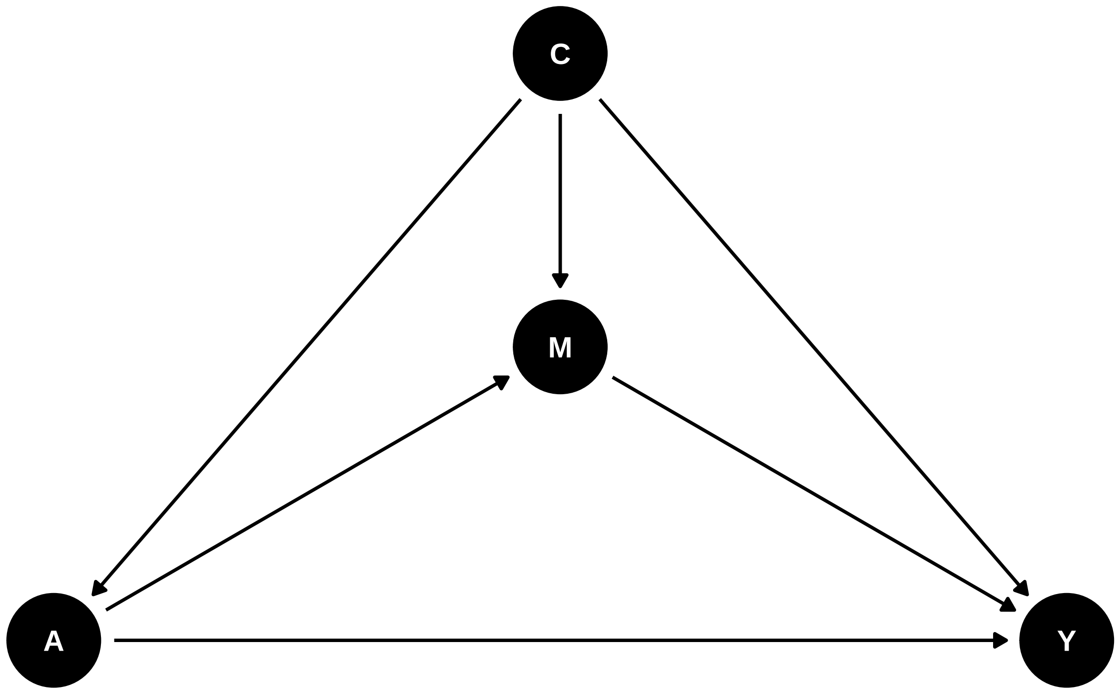
```
##        rb-boot SE rb-boot     Gform  SE Gform
## cde  1.8552508  0.6102194 1.8552508 0.6102194
## pnde 2.7302173  0.2442600 2.7302173 0.2442600
## tnde 2.8164644  0.2392261 2.8164644 0.2392261
## pnie 0.1591483  0.1383247 0.1591483 0.1383247
## tnie 0.2453954  0.1876259 0.2453954 0.1876259
## te   2.9756127  0.3024615 2.9756127 0.3024615
```

4 Study of A - M1 & M2 - Y

Let’s now consider we have two intermediate mediators with M1 impacting M2. We can consider two settings:

```
 - the joint mediating effect of M1 and M2
```

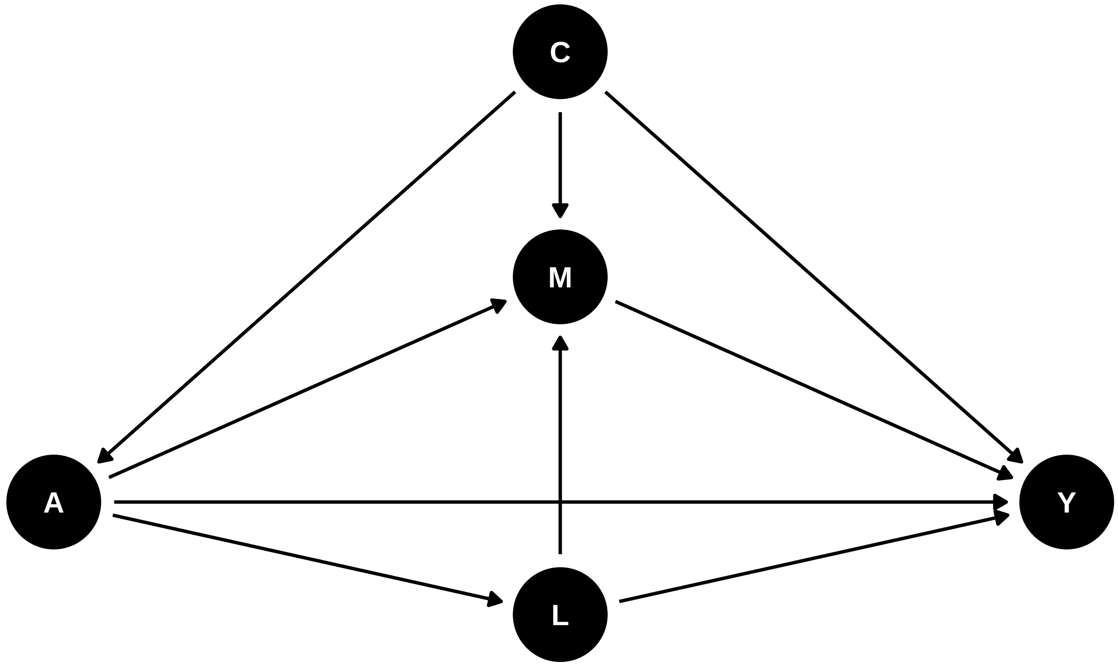
```
cmdag(outcome = "Y", exposure = "A", mediator = c("M1","M2"),
      basec = c("C1", "C2"))
```



A (exposure): A
M (mediator): M1, M2
Y (outcome): Y
C (confounders not affected by the exposure): C1, C2

```
- the focus on M2 considering M1 as a confounder
```

```r
cmdag(outcome = "Y", exposure = "A", mediator = c("M2"), postc = c("M1"),
      basec = c("C1", "C2"))
```



A (exposure): A
M (mediator): M2
Y (outcome): Y
C (confounders not affected by the exposure): C1, C2
L (confounders affected by the exposure): M1

In this setting, we have a confounder of M2-Y that is affected by A. Only G-formula will be possible here.

4.1 Joint mediating effect of M1 and M2

The function cmest handles multiple mediators. We can use the regression-based technique here with the Bootstrap technique for the uncertainty.

```
estJointRB <- cmest(data = dataSim, model = "rb", outcome = "Y", exposure = "A", full=FALSE,
mediator = c("M1", "M2"), basec = c("C1", "C2"), EMint = TRUE,
mreg = list("logistic", "linear"), yreg = "linear",
astar = 0, a = 1, mval = list(1, 1))
```

```
##   |                                                                                |
```

```
summary(estJointRB)
```

```
## Causal Mediation Analysis
##
## # Outcome regression:
##
## Call:
## glm(formula = Y ~ A + M1 + M2 + A * M1 + A * M2 + C1 + C2, 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)  -0.4293     0.4467  -0.961   0.3390
## A             1.3052     0.7518   1.736   0.0859 .
## M1            0.7622     0.3108   2.452   0.0161 *
## M2            0.6919     0.1320   5.240 1.01e-06 ***
```

```
## C1          -0.1973     0.1369  -1.442   0.1528
## C2           2.3898     0.3554   6.724 1.46e-09 ***
## A:M1         0.1279     0.4665   0.274   0.7846
## A:M2         0.2984     0.1512   1.974   0.0514 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.100923)
##
##     Null deviance: 796.20  on 99  degrees of freedom
## Residual deviance: 101.28  on 92  degrees of freedom
## AIC: 303.06
##
## Number of Fisher Scoring iterations: 2
##
##
## # Mediator regressions:
##
## Call:
## glm(formula = M1 ~ A + C1 + C2, 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)  0.003446   0.547959   0.006 0.994982
## A            0.828694   0.456732   1.814 0.069616 .
## C1          -0.984454   0.292880  -3.361 0.000776 ***
## C2           0.892199   0.511832   1.743 0.081308 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 138.47  on 99  degrees of freedom
## Residual deviance: 115.67  on 96  degrees of freedom
## AIC: 123.67
##
## Number of Fisher Scoring iterations: 3
##
##
##
## Call:
## glm(formula = M2 ~ A + C1 + C2, family = gaussian(), data = getCall(x$reg.output$mreg[[2L]])$data,
##     weights = getCall(x$reg.output$mreg[[2L]])$weights)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6403     0.2564   6.397 5.75e-09 ***
## A             0.2901     0.2170   1.337    0.184
## C1            0.6208     0.1189   5.219 1.04e-06 ***
## C2            2.0833     0.2366   8.806 5.44e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for gaussian family taken to be 1.110528)
##
##     Null deviance: 225.08  on 99  degrees of freedom
## Residual deviance: 106.61  on 96  degrees of freedom
## AIC: 300.19
##
## Number of Fisher Scoring iterations: 2
##
##
## # Effect decomposition on the mean difference scale via the regression-based 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   1.73144   0.55398 0.50032   2.601   0.01 **
## pnde  2.50222   0.25297 2.01963   3.014 <2e-16 ***
## tnde  2.60670   0.25588 2.10292   3.090 <2e-16 ***
## pnie  0.30744   0.20093 0.06504   0.806   0.02 *
## tnie  0.41191   0.23033 0.08034   0.959   0.01 **
## te    2.91414   0.29591 2.38294   3.537 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (cde: controlled direct effect; pnde: pure natural direct effect; tnde: total natural direct effect;
##
## Relevant variable values:
## $a
## [1] 1
##
## $astar
## [1] 0
##
## $mval
## $mval[[1]]
## [1] 1
##
## $mval[[2]]
## [1] 1
```

Here, we do not consider the impact of M1 on M2. We directly look at the joint effect.

4.2 The stochastic/randomized analogues to NDE/NIE for M2

The function cmest handles post-exposure confounders with the gformula method, and postc and postcreg arguments.

```
estLGForm <- cmest(data = dataSim, model = "gformula", outcome = "Y", exposure = "A", full=FALSE,
mediator = c("M2"), basec = c("C1", "C2"), EMint = TRUE, postc = "M1", postcreg = list("logistic"),
mreg = list("linear"), yreg = "linear",
astar = 0, a = 1, mval = list(1))
```

```
##   |                                                                                   |
```

The procedure now includes three regressions: one for Y, one for M2 and one for L. It then estimates the direct and indirect effects under randomized intervention.

```r
summary(estLGForm)
```

```
## Causal Mediation Analysis
##
## # Outcome regression:
##
## Call:
## glm(formula = Y ~ A + M2 + A * M2 + C1 + C2 + M1, 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)  -0.4602     0.4301  -1.070  0.28745
## A             1.4193     0.6229   2.278  0.02499 *
## M2            0.6912     0.1314   5.262 9.11e-07 ***
## C1           -0.1936     0.1355  -1.429  0.15646
## C2            2.4059     0.3488   6.898 6.29e-10 ***
## M1            0.8103     0.2553   3.174  0.00204 **
## A:M2          0.2871     0.1447   1.984  0.05021 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.089975)
##
##     Null deviance: 796.20  on 99  degrees of freedom
## Residual deviance: 101.37  on 93  degrees of freedom
## AIC: 301.15
##
## Number of Fisher Scoring iterations: 2
##
##
## # Mediator regressions:
##
## Call:
## glm(formula = M2 ~ A + C1 + C2 + M1, family = gaussian(), data = getCall(x$reg.output$mreg[[1L]])$da
##     weights = getCall(x$reg.output$mreg[[1L]])$weights)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0904     0.2602   8.033 2.56e-12 ***
## A             0.4478     0.2040   2.195 0.030577 *
## C1            0.4412     0.1180   3.739 0.000316 ***
## C2            2.2523     0.2223  10.133  < 2e-16 ***
## M1           -0.9142     0.2187  -4.179 6.50e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9479316)
##
##     Null deviance: 225.084  on 99  degrees of freedom
## Residual deviance:  90.053  on 95  degrees of freedom
## AIC: 285.31
##
## Number of Fisher Scoring iterations: 2
```

```
##
##
## # Regressions for mediator-outcome confounders affected by the exposure:
##
## Call:
## glm(formula = M1 ~ A + C1 + C2, 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)  0.003446   0.547959   0.006 0.994982
## A            0.828694   0.456732   1.814 0.069616 .
## C1          -0.984454   0.292880  -3.361 0.000776 ***
## C2           0.892199   0.511832   1.743 0.081308 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 138.47  on 99  degrees of freedom
## Residual deviance: 115.67  on 96  degrees of freedom
## AIC: 123.67
##
## Number of Fisher Scoring iterations: 3
##
##
## # 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    1.86032   0.60080  0.57187   2.920   0.01 **
## rpnde  2.63348   0.26015  2.13280   3.122 <2e-16 ***
## rtnde  2.71217   0.22581  2.29804   3.175 <2e-16 ***
## rpnie  0.18943   0.16772 -0.09267   0.561   0.18
## rtnie  0.26811   0.23906 -0.13024   0.785   0.18
## te     2.90160   0.25276  2.45455   3.398 <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: rand
##
## Relevant variable values:
## $a
## [1] 1
##
## $astar
## [1] 0
##
## $mval
## $mval[[1]]
## [1] 1
```

5 sensitivity analyses

Let's go back to the firt example with M2 only. cmsens provides tools to assess two issues in the data: unmeasured confounding (sens="uc") and measurement error (sens="me").

5.1 Unmeasured confounding

We can check for the sensitivity of the results under the assumption of a unmeasured confounder

```
cmsens(object = estRB2, sens = "uc")
```

```
## Confidence interval crosses the true value, so its E-value is 1.
## Confidence interval crosses the true value, so its E-value is 1.
```

```
## Sensitivity Analysis For Unmeasured Confounding
##
## Evalues on the risk or rate ratio scale:
##         estRR   lowerRR  upperRR Evalue.estRR Evalue.lowerRR Evalue.upperRR
## cde  1.813612 1.3155801 2.500180     3.028344       1.959918             NA
## pnde 2.375333 2.0523872 2.749094     4.182782       3.522049             NA
## tnde 2.441989 2.1100335 2.826168     4.318507       3.640461             NA
## pnie 1.052395 0.9731170 1.138131     1.287213       1.000000             NA
## tnie 1.081927 0.9618603 1.216981     1.379649       1.000000             NA
## te   2.569936 2.1611933 3.055983     4.578576       3.745353             NA
```

5.2 Measurement error

Let's assume that C1 is measured with error (this is not true, so we do not expect a change in estimates). For continuous variables, cmsens implements two techniques, the regression calibration and the SIMEX approach. This is usable with regression technique and g-formula.

Here is with regression calibration considering masurement error with standard deviation of 0.1,0.2, 0.3:

```
me1 <- cmsens(object = estRB2, sens = "me", MEmethod = "rc",
MEvariable = "C1", MEvartype = "con", MEerror = c(0.1, 0.2, 0.3))
```

```
summary(me1)
```

```
## Sensitivity Analysis For Measurement Error
##
## The variable measured with error: C1
## Type of the variable measured with error: continuous
##
## # Measurement error 1:
## [1] 0.1
##
## ## Error-corrected regressions for measurement error 1:
##
## ### Outcome regression:
## Call:
## rcreg(reg = getCall(x$sens[[1L]]$reg.output$yreg)$reg, formula = Y ~
##     A + M2 + A * M2 + C1 + C2, data = getCall(x$sens[[1L]]$reg.output$yreg)$data,
##     MEvariable = "C1", MEerror = 0.1, variance = TRUE, nboot = 400,
##     weights = getCall(x$sens[[1L]]$reg.output$yreg)$weights)
##
## Naive coefficient estimates:
## (Intercept)           A          M2          C1          C2        A:M2
```

```
##   0.1806018   1.5579866   0.5485291  -0.2672158   2.8463247   0.2972642
##
## Naive var-cov estimates:
##              (Intercept)            A           M2            C1           C2
## (Intercept)  0.158163633 -0.160930937 -0.036124509 -0.0067450850  0.011213946
## A           -0.160930937  0.423419225  0.041945997  0.0042419412 -0.035422845
## M2          -0.036124509  0.041945997  0.016705298 -0.0071204548 -0.027668152
## C1          -0.006745085  0.004241941 -0.007120455  0.0195512640  0.015052202
## C2           0.011213946 -0.035422845 -0.027668152  0.0150522017  0.112273323
## A:M2         0.036247463 -0.092374183 -0.011229253  0.0003283333  0.008813631
##                     A:M2
## (Intercept)  0.0362474633
## A           -0.0923741831
## M2          -0.0112292528
## C1           0.0003283333
## C2           0.0088136313
## A:M2         0.0229496384
##
## Variable measured with error:
## C1
## Measurement error:
## 0.1
## Error-corrected results:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1822     0.4058   0.449   0.6544
## A             1.5567     0.7260   2.144   0.0346 *
## M2            0.5501     0.1341   4.103 8.67e-05 ***
## C1           -0.2716     0.1219  -2.228   0.0283 *
## C2            2.8430     0.3476   8.179 1.34e-12 ***
## A:M2          0.2973     0.1574   1.889   0.0620 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ### Mediator regressions:
## Call:
## rcreg(reg = getCall(x$sens[[1L]]$reg.output$mreg[[1L]])$reg,
##     formula = M2 ~ A + C1 + C2, data = getCall(x$sens[[1L]]$reg.output$mreg[[1L]])$data,
##     MEvariable = "C1", MEerror = 0.1, variance = TRUE, nboot = 400,
##     weights = getCall(x$sens[[1L]]$reg.output$mreg[[1L]])$weights)
##
## Naive coefficient estimates:
## (Intercept)           A          C1          C2
##   1.6402566   0.2901364   0.6208107   2.0833115
##
## Naive var-cov estimates:
##             (Intercept)            A            C1            C2
## (Intercept)  0.06573913 -0.018924346 -0.0173562931 -0.0381103530
## A           -0.01892435  0.047073002  0.0032931500 -0.0062472874
## C1          -0.01735629  0.003293150  0.0141472682  0.0003964488
## C2          -0.03811035 -0.006247287  0.0003964488  0.0559647177
##
## Variable measured with error:
## C1
## Measurement error:
```

```
## 0.1
## Error-corrected results:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6305     0.2276   7.164 1.60e-10 ***
## A             0.2920     0.2118   1.379    0.171
## C1            0.6287     0.1336   4.707 8.46e-06 ***
## C2            2.0835     0.1900  10.968  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## ## Error-corrected causal effects on the mean difference scale for measurement error 1:
##      Estimate Std.error  95% CIL 95% CIU  P.val
## cde   1.85400   0.57656  0.72396   2.984 0.0013 **
## pnde  2.69462   0.22671  2.25029   3.139 <2e-16 ***
## tnde  2.78142   0.21400  2.36198   3.201 <2e-16 ***
## pnie  0.16062   0.12290 -0.08027   0.402 0.1913
## tnie  0.24741   0.18326 -0.11176   0.607 0.1770
## te    2.94204   0.26200  2.42854   3.456 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ----------------------------------------------------------------
##
## # Measurement error 2:
## [1] 0.2
##
## ## Error-corrected regressions for measurement error 2:
##
## ### Outcome regression:
## Call:
## rcreg(reg = getCall(x$sens[[2L]]$reg.output$yreg)$reg, formula = Y ~
##     A + M2 + A * M2 + C1 + C2, data = getCall(x$sens[[2L]]$reg.output$yreg)$data,
##     MEvariable = "C1", MEerror = 0.2, variance = TRUE, nboot = 400,
##     weights = getCall(x$sens[[2L]]$reg.output$yreg)$weights)
##
## Naive coefficient estimates:
## (Intercept)           A          M2          C1          C2        A:M2
##   0.1806018   1.5579866   0.5485291  -0.2672158   2.8463247   0.2972642
##
## Naive var-cov estimates:
##              (Intercept)            A           M2            C1           C2
## (Intercept)  0.158163633 -0.160930937 -0.036124509 -0.0067450850  0.011213946
## A           -0.160930937  0.423419225  0.041945997  0.0042419412 -0.035422845
## M2          -0.036124509  0.041945997  0.016705298 -0.0071204548 -0.027668152
## C1          -0.006745085  0.004241941 -0.007120455  0.0195512640  0.015052202
## C2           0.011213946 -0.035422845 -0.027668152  0.0150522017  0.112273323
## A:M2         0.036247463 -0.092374183 -0.011229253  0.0003283333  0.008813631
##                     A:M2
## (Intercept)  0.0362474633
## A           -0.0923741831
## M2          -0.0112292528
## C1           0.0003283333
## C2           0.0088136313
## A:M2         0.0229496384
```

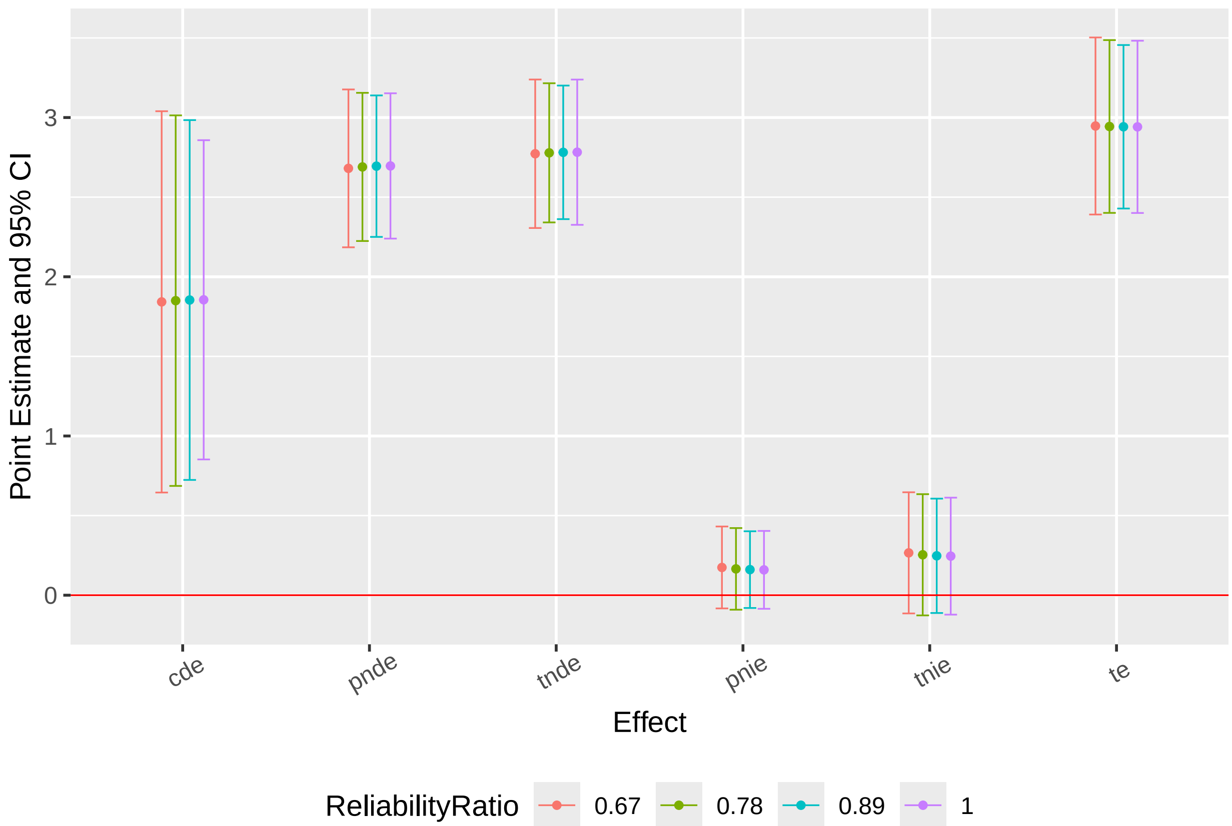
```
##
## Variable measured with error:
## C1
## Measurement error:
## 0.2
## Error-corrected results:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1875     0.4160   0.451 0.653256
## A             1.5527     0.7454   2.083 0.039967 *
## M2            0.5551     0.1388   3.999 0.000127 ***
## C1           -0.2857     0.1270  -2.250 0.026770 *
## C2            2.8322     0.3412   8.300 7.46e-13 ***
## A:M2          0.2973     0.1599   1.859 0.066134 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ### Mediator regressions:
## Call:
## rcreg(reg = getCall(x$sens[[2L]]$reg.output$mreg[[1L]])$reg,
##     formula = M2 ~ A + C1 + C2, data = getCall(x$sens[[2L]]$reg.output$mreg[[1L]])$data,
##     MEvariable = "C1", MEerror = 0.2, variance = TRUE, nboot = 400,
##     weights = getCall(x$sens[[2L]]$reg.output$mreg[[1L]])$weights)
##
## Naive coefficient estimates:
## (Intercept)           A          C1          C2
##   1.6402566   0.2901364   0.6208107   2.0833115
##
## Naive var-cov estimates:
##             (Intercept)            A            C1            C2
## (Intercept)  0.06573913 -0.018924346 -0.0173562931 -0.0381103530
## A           -0.01892435  0.047073002  0.0032931500 -0.0062472874
## C1          -0.01735629  0.003293150  0.0141472682  0.0003964488
## C2          -0.03811035 -0.006247287  0.0003964488  0.0559647177
##
## Variable measured with error:
## C1
## Measurement error:
## 0.1
## Error-corrected results:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6305     0.2276   7.164 1.60e-10 ***
## A             0.2920     0.2118   1.379    0.171
## C1            0.6287     0.1336   4.707 8.46e-06 ***
## C2            2.0835     0.1900  10.968  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## ## Error-corrected causal effects on the mean difference scale for measurement error 2:
##      Estimate Std.error 95% CIL 95% CIU   P.val
## cde    1.8500    0.5937  0.6863   3.014 0.00183 **
## pnde   2.6899    0.2374  2.2245   3.155 < 2e-16 ***
## tnde   2.7784    0.2230  2.3414   3.215 < 2e-16 ***
## pnie   0.1653    0.1308 -0.0911   0.422 0.20635
```

```
## tnie   0.2539    0.1943 -0.1269   0.635 0.19128
## te     2.9437    0.2769  2.4011   3.486 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ----------------------------------------------------------------
##
## # Measurement error 3:
## [1] 0.3
##
## ## Error-corrected regressions for measurement error 3:
##
## ### Outcome regression:
## Call:
## rcreg(reg = getCall(x$sens[[3L]]$reg.output$yreg)$reg, formula = Y ~
##     A + M2 + A * M2 + C1 + C2, data = getCall(x$sens[[3L]]$reg.output$yreg)$data,
##     MEvariable = "C1", MEerror = 0.3, variance = TRUE, nboot = 400,
##     weights = getCall(x$sens[[3L]]$reg.output$yreg)$weights)
##
## Naive coefficient estimates:
## (Intercept)           A          M2          C1          C2        A:M2
##   0.1806018   1.5579866   0.5485291  -0.2672158   2.8463247   0.2972642
##
## Naive var-cov estimates:
##              (Intercept)            A           M2            C1           C2
## (Intercept)  0.158163633 -0.160930937 -0.036124509 -0.0067450850  0.011213946
## A           -0.160930937  0.423419225  0.041945997  0.0042419412 -0.035422845
## M2          -0.036124509  0.041945997  0.016705298 -0.0071204548 -0.027668152
## C1          -0.006745085  0.004241941 -0.007120455  0.0195512640  0.015052202
## C2           0.011213946 -0.035422845 -0.027668152  0.0150522017  0.112273323
## A:M2         0.036247463 -0.092374183 -0.011229253  0.0003283333  0.008813631
##                      A:M2
## (Intercept)  0.0362474633
## A           -0.0923741831
## M2          -0.0112292528
## C1           0.0003283333
## C2           0.0088136313
## A:M2         0.0229496384
##
## Variable measured with error:
## C1
## Measurement error:
## 0.3
## Error-corrected results:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1975     0.4188   0.472 0.638265
## A             1.5450     0.7668   2.015 0.046781 *
## M2            0.5648     0.1409   4.008 0.000122 ***
## C1           -0.3128     0.1476  -2.120 0.036668 *
## C2            2.8115     0.3491   8.053 2.47e-12 ***
## A:M2          0.2973     0.1654   1.797 0.075493 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ### Mediator regressions:
```

```
## Call:
## rcreg(reg = getCall(x$sens[[3L]]$reg.output$mreg[[1L]])$reg,
##     formula = M2 ~ A + C1 + C2, data = getCall(x$sens[[3L]]$reg.output$mreg[[1L]])$data,
##     MEvariable = "C1", MEerror = 0.3, variance = TRUE, nboot = 400,
##     weights = getCall(x$sens[[3L]]$reg.output$mreg[[1L]])$weights)
##
## Naive coefficient estimates:
## (Intercept)           A          C1          C2
##   1.6402566   0.2901364   0.6208107   2.0833115
##
## Naive var-cov estimates:
##             (Intercept)            A            C1            C2
## (Intercept)  0.06573913 -0.018924346 -0.0173562931 -0.0381103530
## A           -0.01892435  0.047073002  0.0032931500 -0.0062472874
## C1          -0.01735629  0.003293150  0.0141472682  0.0003964488
## C2          -0.03811035 -0.006247287  0.0003964488  0.0559647177
##
## Variable measured with error:
## C1
## Measurement error:
## 0.1
## Error-corrected results:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6305     0.2276   7.164 1.60e-10 ***
## A             0.2920     0.2118   1.379    0.171
## C1            0.6287     0.1336   4.707 8.46e-06 ***
## C2            2.0835     0.1900  10.968  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## ## Error-corrected causal effects on the mean difference scale for measurement error 3:
##      Estimate Std.error  95% CIL 95% CIU   P.val
## cde   1.84228   0.61090  0.64493   3.040 0.00256 **
## pnde  2.68082   0.25292  2.18512   3.177 < 2e-16 ***
## tnde  2.77257   0.23789  2.30631   3.239 < 2e-16 ***
## pnie  0.17431   0.13125 -0.08295   0.432 0.18417
## tnie  0.26605   0.19413 -0.11443   0.647 0.17053
## te    2.94688   0.28365  2.39094   3.503 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ----------------------------------------------------------------
##
## (cde: controlled direct effect; pnde: pure natural direct effect; tnde: total natural direct effect;
##
## Relevant variable values:
## $a
## [1] 1
##
## $astar
## [1] 0
##
## $mval
## $mval[[1]]
```

```
## [1] 1
##
##
## $basecval
## $basecval[[1]]
## [1] 1.108887
##
## $basecval[[2]]
## [1] 0.72
```

```
ggcmsens(me1) +
ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 30, vjust = 0.8))
```



Here is with SIMEX considering masurement error with standard deviation of 0.1,0.2, 0.3:

```
me1simex <- cmsens(object = estRB2, sens = "me", MEmethod = "simex",
MEvariable = "C1", MEvartype = "con", MEerror = c(0.1, 0.2, 0.3))
```

```
summary(me1simex)
```

```
## Sensitivity Analysis For Measurement Error
##
## The variable measured with error: C1
## Type of the variable measured with error: continuous
##
## # Measurement error 1:
## [1] 0.1
##
## ## Error-corrected regressions for measurement error 1:
```

```
##
## ### Outcome regression:
## Call:
## simexreg(reg = getCall(x$sens[[1L]]$reg.output$yreg)$reg, formula = Y ~
##     A + M2 + A * M2 + C1 + C2, data = getCall(x$sens[[1L]]$reg.output$yreg)$data,
##     MEvariable = "C1", MEvartype = "continuous", MEerror = 0.1,
##     variance = TRUE, lambda = c(0.5, 1, 1.5, 2), B = 200, weights = getCall(x$sens[[1L]]$reg.output$y
##
## Naive coefficient estimates:
## (Intercept)           A          M2          C1          C2        A:M2
##   0.1806018   1.5579866   0.5485291  -0.2672158   2.8463247   0.2972642
##
## Naive var-cov estimates:
##              (Intercept)            A           M2            C1           C2
## (Intercept)  0.158163633 -0.160930937 -0.036124509 -0.0067450850  0.011213946
## A           -0.160930937  0.423419225  0.041945997  0.0042419412 -0.035422845
## M2          -0.036124509  0.041945997  0.016705298 -0.0071204548 -0.027668152
## C1          -0.006745085  0.004241941 -0.007120455  0.0195512640  0.015052202
## C2           0.011213946 -0.035422845 -0.027668152  0.0150522017  0.112273323
## A:M2         0.036247463 -0.092374183 -0.011229253  0.0003283333  0.008813631
##                      A:M2
## (Intercept)  0.0362474633
## A           -0.0923741831
## M2          -0.0112292528
## C1           0.0003283333
## C2           0.0088136313
## A:M2         0.0229496384
##
## Variable measured with error:
## C1
## Measurement error:
## [1] 0.1
##
## Error-corrected results:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1808     0.3978   0.455   0.6505
## A             1.5591     0.6509   2.396   0.0186 *
## M2            0.5488     0.1294   4.241 5.20e-05 ***
## C1           -0.2686     0.1411  -1.904   0.0600 .
## C2            2.8445     0.3355   8.477 3.15e-13 ***
## A:M2          0.2974     0.1515   1.963   0.0526 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ### Mediator regressions:
## Call:
## simexreg(reg = getCall(x$sens[[1L]]$reg.output$mreg[[1L]])$reg,
##     formula = M2 ~ A + C1 + C2, data = getCall(x$sens[[1L]]$reg.output$mreg[[1L]])$data,
##     MEvariable = "C1", MEvartype = "continuous", MEerror = 0.1,
##     variance = TRUE, lambda = c(0.5, 1, 1.5, 2), B = 200, weights = getCall(x$sens[[1L]]$reg.output$m
##
## Naive coefficient estimates:
## (Intercept)           A          C1          C2
##   1.6402566   0.2901364   0.6208107   2.0833115
```

```
##
## Naive var-cov estimates:
##             (Intercept)            A           C1            C2
## (Intercept)  0.06573913 -0.018924346 -0.0173562931 -0.0381103530
## A           -0.01892435  0.047073002  0.0032931500 -0.0062472874
## C1          -0.01735629  0.003293150  0.0141472682  0.0003964488
## C2          -0.03811035 -0.006247287  0.0003964488  0.0559647177
##
## Variable measured with error:
## C1
## Measurement error:
## [1] 0.1
##
## Error-corrected results:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6228     0.2578   6.295 9.17e-09 ***
## A             0.2938     0.2166   1.356    0.178
## C1            0.6327     0.1207   5.244 9.36e-07 ***
## C2            2.0839     0.2362   8.822 5.05e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## ## Error-corrected causal effects on the mean difference scale for measurement error 1:
##      Estimate Std.error  95% CIL 95% CIU    P.val
## cde   1.85654   0.51158  0.85385   2.859 0.000285 ***
## pnde  2.69668   0.23283  2.24034   3.153  < 2e-16 ***
## tnde  2.78405   0.23270  2.32796   3.240  < 2e-16 ***
## pnie  0.16123   0.12482 -0.08341   0.406 0.196464
## tnie  0.24860   0.18733 -0.11856   0.616 0.184483
## te    2.94527   0.27585  2.40461   3.486  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ----------------------------------------------------------------
##
## # Measurement error 2:
## [1] 0.2
##
## ## Error-corrected regressions for measurement error 2:
##
## ### Outcome regression:
## Call:
## simexreg(reg = getCall(x$sens[[2L]]$reg.output$yreg)$reg, formula = Y ~
##     A + M2 + A * M2 + C1 + C2, data = getCall(x$sens[[2L]]$reg.output$yreg)$data,
##     MEvariable = "C1", MEvartype = "continuous", MEerror = 0.2,
##     variance = TRUE, lambda = c(0.5, 1, 1.5, 2), B = 200, weights = getCall(x$sens[[2L]]$reg.output$
##
## Naive coefficient estimates:
## (Intercept)           A          M2          C1          C2        A:M2
##   0.1806018   1.5579866   0.5485291  -0.2672158   2.8463247   0.2972642
##
## Naive var-cov estimates:
##              (Intercept)            A           M2            C1           C2
## (Intercept)  0.158163633 -0.160930937 -0.036124509 -0.0067450850  0.011213946
```

```
## A           -0.160930937  0.423419225  0.041945997  0.0042419412 -0.035422845
## M2          -0.036124509  0.041945997  0.016705298 -0.0071204548 -0.027668152
## C1          -0.006745085  0.004241941 -0.007120455  0.0195512640  0.015052202
## C2           0.011213946 -0.035422845 -0.027668152  0.0150522017  0.112273323
## A:M2         0.036247463 -0.092374183 -0.011229253  0.0003283333  0.008813631
##                      A:M2
## (Intercept)  0.0362474633
## A           -0.0923741831
## M2          -0.0112292528
## C1           0.0003283333
## C2           0.0088136313
## A:M2         0.0229496384
##
## Variable measured with error:
## C1
## Measurement error:
## [1] 0.2
##
## Error-corrected results:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1902     0.3981   0.478   0.6339
## A             1.5507     0.6506   2.383   0.0192 *
## M2            0.5560     0.1303   4.268 4.70e-05 ***
## C1           -0.2896     0.1487  -1.947   0.0545 .
## C2            2.8276     0.3373   8.384 4.97e-13 ***
## A:M2          0.2976     0.1514   1.966   0.0523 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ### Mediator regressions:
## Call:
## simexreg(reg = getCall(x$sens[[2L]]$reg.output$mreg[[1L]])$reg,
##     formula = M2 ~ A + C1 + C2, data = getCall(x$sens[[2L]]$reg.output$mreg[[1L]])$data,
##     MEvariable = "C1", MEvartype = "continuous", MEerror = 0.2,
##     variance = TRUE, lambda = c(0.5, 1, 1.5, 2), B = 200, weights = getCall(x$sens[[2L]]$reg.output$m
##
## Naive coefficient estimates:
## (Intercept)           A          C1          C2
##   1.6402566   0.2901364   0.6208107   2.0833115
##
## Naive var-cov estimates:
##             (Intercept)            A           C1            C2
## (Intercept)  0.06573913 -0.018924346 -0.0173562931 -0.0381103530
## A           -0.01892435  0.047073002  0.0032931500 -0.0062472874
## C1          -0.01735629  0.003293150  0.0141472682  0.0003964488
## C2          -0.03811035 -0.006247287  0.0003964488  0.0559647177
##
## Variable measured with error:
## C1
## Measurement error:
## [1] 0.1
##
## Error-corrected results:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   1.6228     0.2578   6.295 9.17e-09 ***
## A             0.2938     0.2166   1.356    0.178
## C1            0.6327     0.1207   5.244 9.36e-07 ***
## C2            2.0839     0.2362   8.822 5.05e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## ## Error-corrected causal effects on the mean difference scale for measurement error 2:
##      Estimate Std.error  95% CIL 95% CIU    P.val
## cde   1.84833   0.51151  0.84579   2.851 0.000302 ***
## pnde  2.68816   0.23333  2.23084   3.145  < 2e-16 ***
## tnde  2.77779   0.23315  2.32083   3.235  < 2e-16 ***
## pnie  0.16744   0.12665 -0.08079   0.416 0.186149
## tnie  0.25707   0.18912 -0.11360   0.628 0.174052
## te    2.94523   0.27680  2.40271   3.488  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ----------------------------------------------------------------
##
## # Measurement error 3:
## [1] 0.3
##
## ## Error-corrected regressions for measurement error 3:
##
## ### Outcome regression:
## Call:
## simexreg(reg = getCall(x$sens[[3L]]$reg.output$yreg)$reg, formula = Y ~
##     A + M2 + A * M2 + C1 + C2, data = getCall(x$sens[[3L]]$reg.output$yreg)$data,
##     MEvariable = "C1", MEvartype = "continuous", MEerror = 0.3,
##     variance = TRUE, lambda = c(0.5, 1, 1.5, 2), B = 200, weights = getCall(x$sens[[3L]]$reg.output$y
##
## Naive coefficient estimates:
## (Intercept)           A          M2          C1          C2        A:M2
##   0.1806018   1.5579866   0.5485291  -0.2672158   2.8463247   0.2972642
##
## Naive var-cov estimates:
##              (Intercept)            A           M2            C1           C2
## (Intercept)  0.158163633 -0.160930937 -0.036124509 -0.0067450850  0.011213946
## A           -0.160930937  0.423419225  0.041945997  0.0042419412 -0.035422845
## M2          -0.036124509  0.041945997  0.016705298 -0.0071204548 -0.027668152
## C1          -0.006745085  0.004241941 -0.007120455  0.0195512640  0.015052202
## C2           0.011213946 -0.035422845 -0.027668152  0.0150522017  0.112273323
## A:M2         0.036247463 -0.092374183 -0.011229253  0.0003283333  0.008813631
##                     A:M2
## (Intercept)  0.0362474633
## A           -0.0923741831
## M2          -0.0112292528
## C1           0.0003283333
## C2           0.0088136313
## A:M2         0.0229496384
##
## Variable measured with error:
## C1
```

```
## Measurement error:
## [1] 0.3
##
## Error-corrected results:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1887     0.3997   0.472   0.6379
## A             1.5523     0.6531   2.377   0.0195 *
## M2            0.5566     0.1326   4.198 6.12e-05 ***
## C1           -0.2865     0.1589  -1.803   0.0746 .
## C2            2.8259     0.3407   8.294 7.68e-13 ***
## A:M2          0.2976     0.1520   1.958   0.0532 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ### Mediator regressions:
## Call:
## simexreg(reg = getCall(x$sens[[3L]]$reg.output$mreg[[1L]])$reg,
##     formula = M2 ~ A + C1 + C2, data = getCall(x$sens[[3L]]$reg.output$mreg[[1L]])$data,
##     MEvariable = "C1", MEvartype = "continuous", MEerror = 0.3,
##     variance = TRUE, lambda = c(0.5, 1, 1.5, 2), B = 200, weights = getCall(x$sens[[3L]]$reg.output$m
##
## Naive coefficient estimates:
## (Intercept)           A          C1          C2
##   1.6402566   0.2901364   0.6208107   2.0833115
##
## Naive var-cov estimates:
##             (Intercept)            A            C1            C2
## (Intercept)  0.06573913 -0.018924346 -0.0173562931 -0.0381103530
## A           -0.01892435  0.047073002  0.0032931500 -0.0062472874
## C1          -0.01735629  0.003293150  0.0141472682  0.0003964488
## C2          -0.03811035 -0.006247287  0.0003964488  0.0559647177
##
## Variable measured with error:
## C1
## Measurement error:
## [1] 0.1
##
## Error-corrected results:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6228     0.2578   6.295 9.17e-09 ***
## A             0.2938     0.2166   1.356    0.178
## C1            0.6327     0.1207   5.244 9.36e-07 ***
## C2            2.0839     0.2362   8.822 5.05e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## ## Error-corrected causal effects on the mean difference scale for measurement error 3:
##       Estimate Std.error  95% CIL 95% CIU    P.val
## cde    1.84994   0.51344  0.84361   2.856 0.000315 ***
## pnde  2.69069   0.23457  2.23095   3.150  < 2e-16 ***
## tnde  2.78033   0.23456  2.32059   3.240  < 2e-16 ***
## pnie  0.16761   0.12731 -0.08192   0.417 0.187998
## tnie  0.25725   0.18989 -0.11494   0.629 0.175516
```

```
## te    2.94794   0.27797  2.40312   3.493  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## ----------------------------------------------------------------
##
## (cde: controlled direct effect; pnde: pure natural direct effect; tnde: total natural direct effect;
##
## Relevant variable values:
## $a
## [1] 1
##
## $astar
## [1] 0
##
## $mval
## $mval[[1]]
## [1] 1
##
##
## $basecval
## $basecval[[1]]
## [1] 1.108887
##
## $basecval[[2]]
## [1] 0.72
```

```
ggcmsens(me1simex) +
ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 30, vjust = 0.8))
```