SOCI 40258

Causal Mediation Analysis

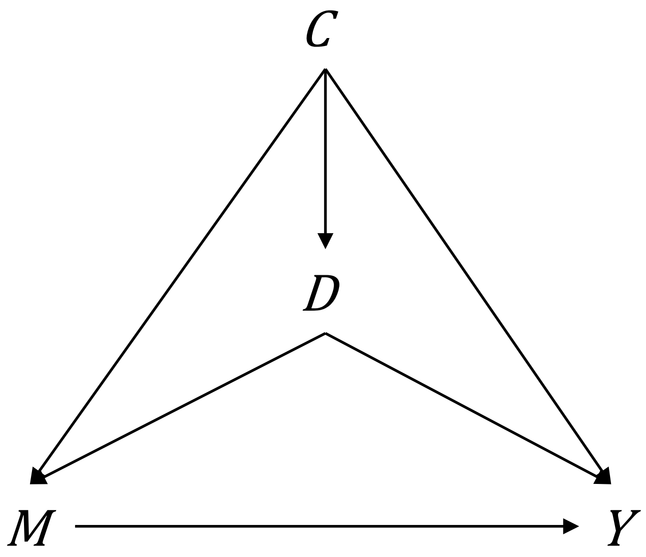
Week 4: Estimating Natural Effects

Outline

- Review of nonparametric identification and estimation
- Limitations of nonparametric estimation
- Parametric estimation with linear models
- Parametric estimation via simulation
- Parametric estimation via weighting

A model with baseline confounding

- In this model, D affects Y directly and indirectly through M
- A set of covariates, denoted by C, jointly affects the exposure, mediator, and outcome of interest
- Thus, C confounds the exposure-outcome, exposure-mediator, and mediator-outcome relationships
- We will refer to variables like C as baseline confounders



A model with baseline confounding

- The methods covered this week are appropriate for data arising from a causal process resembling the graphical model depicted previously

- My presentation of these methods is tailored for models that allow general patterns of baseline confounding

- These methods are also appropriate for settings without any confounding or with exposure-outcome, mediator-outcome, or exposure mediator confounding only

The natural effects decomposition

- The average total effect of the exposure on the outcome can be decomposed into direct and indirect components as follows:

$$ATE(d, d^*) = E\big(Y(d) - Y(d^*)\big)$$

$$= E\left(Y\big(d, M(d)\big) - Y\big(d^*, M(d^*)\big)\right)$$

$$= E\left(Y\big(d, M(d^*)\big) - Y\big(d^*, M(d^*)\big)\right) + E\left(Y\big(d, M(d)\big) - Y\big(d, M(d^*)\big)\right)$$

The natural effects decomposition

- The average total effect of the exposure on the outcome can be decomposed into direct and indirect components as follows:

$$
\begin{aligned}
ATE(d,d^*) &= E\big(Y(d) - Y(d^*)\big) \\
&= E\Big(Y\big(d,M(d)\big) - Y\big(d^*,M(d^*)\big)\Big) \\
&= \underbrace{E\Big(Y\big(d,M(d^*)\big) - Y\big(d^*,M(d^*)\big)\Big)}_{\text{natural direct effect}} + \underbrace{E\Big(Y\big(d,M(d)\big) - Y\big(d,M(d^*)\big)\Big)}_{\text{natural indirect effect}}
\end{aligned}
$$

Nonparametric identification

- Natural direct and indirect effects can be nonparametrically identified if the following conditions are met:

Assumption NE.1: $Y(d,m) \perp D|C$

Assumption NE.2: $Y(d,m) \perp M|C,D$

Assumption NE.3: $M(d) \perp D|C$

Assumption NE.4: $Y(d,m) \perp M(d^*)|C$

Assumption NE.5: $P(d,m|c) > 0$

Assumption NE.6: $Y = Y(D) = Y\big(D, M(D)\big) = Y(D,M)$

Identification formulas

- Nonparametric identification formula for the natural direct effect:

$$NDE(d,d^*) = E\left(Y\big(d, M(d^*)\big) - Y\big(d^*, M(d^*)\big)\right)$$

$$= \sum_c \sum_m [E(Y|c,d,m) - E(Y|c,d^*,m)]P(m|c,d^*)P(c)$$

- Nonparametric identification formula for the natural indirect effect:

$$NIE(d,d^*) = E\left(Y\big(d, M(d)\big) - Y\big(d, M(d^*)\big)\right)$$

$$= \sum_c \sum_m E(Y|c,d,m)[P(m|c,d) - P(m|c,d^*)]P(c)$$

Nonparametric estimation

- Nonparametric estimation just involves plugging in sample analogs for the population quantities in the nonparametric identification formulas outlined previously

- However, this approach to estimation is often difficult, impractical, or impossible to implement owing to:
  - Sparsity
  - The curse of dimensionality
  - Excessive sampling variability

Parametric estimation

- A parametric estimator is based on a parametric model or a set of parametric models (i.e., models that impose constraints on the joint distribution of the data)

- Parametric estimators can mitigate the problems of sparsity, dimensionality, and variability by sharing, borrowing, and filling in information that is otherwise not available from the sample data

    - With parametric estimation, the positivity condition required for nonparametric identification and estimation is supplanted by an alternative assumption about correct model specification

    - Parametric estimators are only consistent if their underlying models are correctly specified; otherwise, they are biased and inconsistent

Estimation with linear models

- Consider the following set of linear and additive models:

$$E(M|c,d) = \beta_0 + \beta_1^T c + \beta_2 d$$

$$E(Y|c,d,m) = \gamma_0 + \gamma_1^T c + \gamma_2 d + \gamma_3 m$$

- Under these models, the natural effects of interest are given by:

$$NDE(d,d^*) = \gamma_2(d - d^*)$$

$$NIE(d,d^*) = \beta_2\gamma_3(d - d^*)$$

- To compute effect estimates, fit these models by OLS and plug the parameter estimates into the expressions above

Estimation with linear models

- These estimators just come from substituting our linear models into the nonparametric identification formulae and simplifying as follows:

$$
\begin{aligned}
NDE(d, d^*) &= E\left(Y\left(d, M(d^*)\right) - Y\left(d^*, M(d^*)\right)\right) \\
&= \sum_c \sum_m [E(Y|c, d, m) - E(Y|c, d^*, m)] P(m|c, d^*) P(c) \\
&= \sum_c \sum_m [\gamma_2 d - \gamma_2 d^*] P(m|c, d^*) P(c) \\
&= \gamma_2 (d - d^*) \sum_c \sum_m P(m|c, d^*) P(c) \\
&= \gamma_2 (d - d^*)
\end{aligned}
$$

Estimation with linear models

- These estimators just come from substituting our linear models into the nonparametric identification formulae and simplifying as follows:

$$NIE(d,d^*) = E\left(Y\big(d,M(d)\big) - Y\big(d,M(d^*)\big)\right)$$

$$= \sum_c \sum_m E(Y|c,d,m)[P(m|c,d) - P(m|c,d^*)]P(c)$$

$$= \sum_c \sum_m (\gamma_0 + \gamma_1^T c + \gamma_2 d + \gamma_3 m)[P(m|c,d) - P(m|c,d^*)]P(c)$$

$$= \sum_c [\gamma_3 \sum_m mP(m|c,d) - \gamma_3 \sum_m mP(m|c,d^*)]P(c)$$

$$= \sum_c [\gamma_3 E(M|c,d) - \gamma_3 E(M|c,d^*)]P(c)$$

$$= \sum_c [\gamma_3(\beta_0 + \beta_1^T c + \beta_2 d) - \gamma_3(\beta_0 + \beta_1^T c + \beta_2 d^*)]P(c)$$

$$= \beta_2\gamma_3(d - d^*)$$

Treatment-mediator interactions

- Now consider the following set of linear models:

$$E(M|c,d) = \beta_0 + \beta_1^T c^{\perp} + \beta_2 d$$

$$E(Y|c,d,m) = \gamma_0 + \gamma_1^T c^{\perp} + \gamma_2 d + m(\gamma_3 + \gamma_4 d),$$

where $c^{\perp} = c - \bar{C}$

- Under these models, the natural effects of interest are given by:

$$NDE(d,d^*) = \big(\gamma_2 + \gamma_4(\beta_0 + \beta_2 d^*)\big)(d - d^*)$$

$$NIE(d,d^*) = \beta_2(\gamma_3 + \gamma_4 d)(d - d^*)$$

Covariate interactions

- Now consider the following set of linear models:

$$E(M|c,d) = \beta_0 + \beta_1^T c^{\perp} + d(\beta_2 + \beta_3^T c^{\perp})$$

$$E(Y|c,d,m) = \gamma_0 + \gamma_1^T c^{\perp} + \gamma_2 d + m(\gamma_3 + \gamma_4 d) + c^{\perp}\left(\gamma_5^T d + m(\gamma_6^T + \gamma_7^T d)\right),$$

where $c^{\perp} = c - \bar{C}$

- Under these models, the natural effects of interest are given by:

$$\left.\begin{aligned} NDE(d,d^*) &= \left(\gamma_2 + \gamma_4(\beta_0 + \beta_2 d^*)\right)(d - d^*) \\ NIE(d,d^*) &= \beta_2(\gamma_3 + \gamma_4 d)(d - d^*) \end{aligned}\right\} \text{Exactly the same as before!}$$

Summary

- Natural direct and indirect effects can be estimated using linear models for the mediator and outcome fit to sample data by the method of least squares

- These estimators are consistent provided that…
  - the assumptions required for identification are satisfied
  - the models for the mediator and outcome are correctly specified

- Linear models are somewhat restrictive, but they can easily accommodate exposure-mediator interaction and effect moderation across levels of the baseline confounders

Example: NLSY79

- 1979 National Longitudinal Study of Youth
  - Exposure (D)
    - sample member attended college before age 22
  - Outcome (Y):
    - standardized scores on the CES-D at age 40
  - Covariates (C):
    - Race, gender, parental education, occupation, and income, household size, AFQT scores
  - A potential mediator (M)
    - unemployment between age 35-40

Example: NLSY79

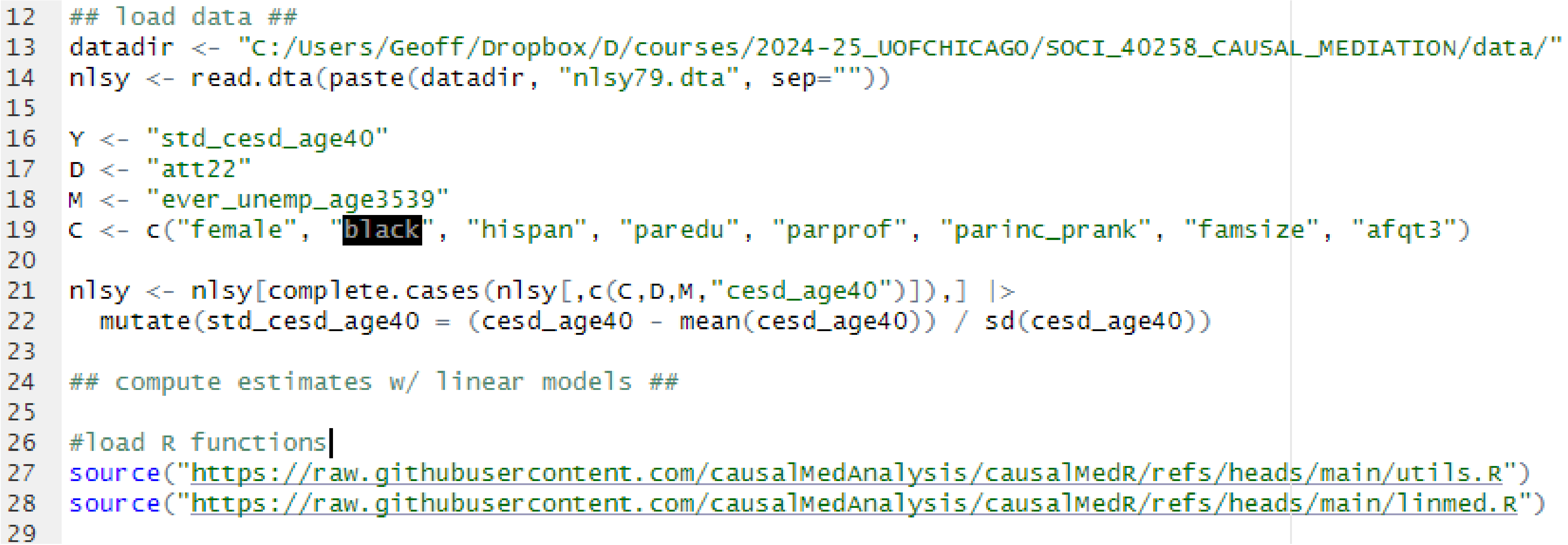
- Many studies have documented that going to college seems to reduce the likelihood of becoming depressed later in life, but how does this effect come about?

- One possibility is that a more advanced education reduces depression by protecting its recipients from financially strenuous and mentally taxing spells of unemployment

- Does unemployment mediate the effect of college attendance on depression?

Example: NLSY79

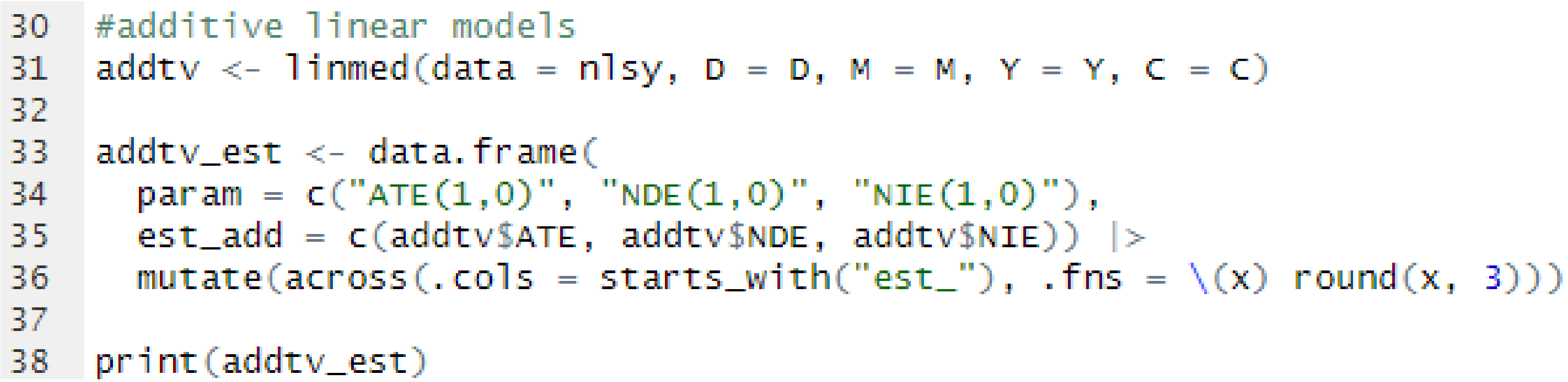
- Compute estimates of the natural effects decomposition using additive linear models



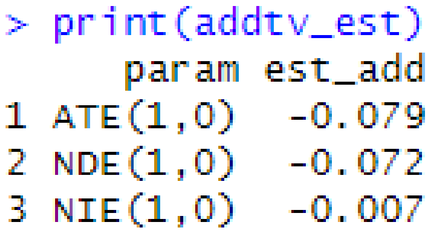
```
## load data ##
datadir <- "C:/Users/Geoff/Dropbox/D/courses/2024-25_UOFCHICAGO/SOCI_40258_CAUSAL_MEDIATION/data/"
nlsy <- read.dta(paste(datadir, "nlsy79.dta", sep=""))

Y <- "std_cesd_age40"
D <- "att22"
M <- "ever_unemp_age3539"
C <- c("female", "black", "hispan", "paredu", "parprof", "parinc_prank", "famsize", "afqt3")

nlsy <- nlsy[complete.cases(nlsy[,c(C,D,M,"cesd_age40")]),] |>
  mutate(std_cesd_age40 = (cesd_age40 - mean(cesd_age40)) / sd(cesd_age40))

## compute estimates w/ linear models ##

#load R functions
source("https://raw.githubusercontent.com/causalMedAnalysis/causalMedR/refs/heads/main/utils.R")
source("https://raw.githubusercontent.com/causalMedAnalysis/causalMedR/refs/heads/main/linmed.R")

```

Example: NLSY79

- Compute estimates of the natural effects decomposition using additive linear models



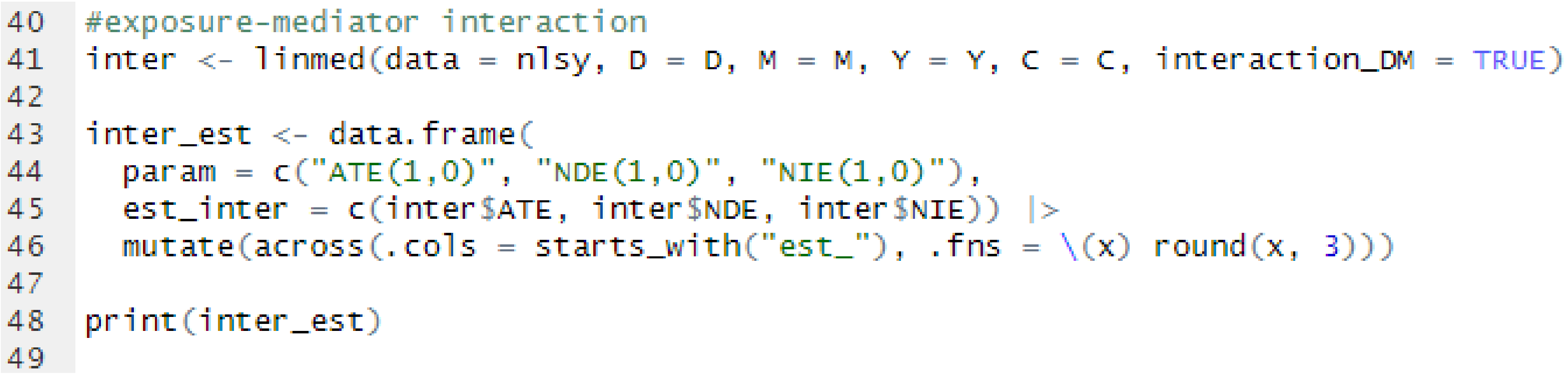
```r
#additive linear models
addtv <- linmed(data = nlsy, D = D, M = M, Y = Y, C = C)

addtv_est <- data.frame(
  param = c("ATE(1,0)", "NDE(1,0)", "NIE(1,0)"),
  est_add = c(addtv$ATE, addtv$NDE, addtv$NIE)) |>
  mutate(across(.cols = starts_with("est_"), .fns = \(x) round(x, 3)))

print(addtv_est)
```



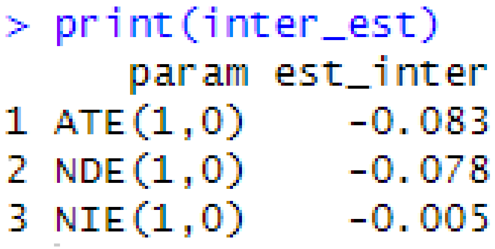
```
> print(addtv_est)
     param est_add
1 ATE(1,0)  -0.079
2 NDE(1,0)  -0.072
3 NIE(1,0)  -0.007
```

Example: NLSY79

- Compute estimates using linear models with an exposure-mediator interaction



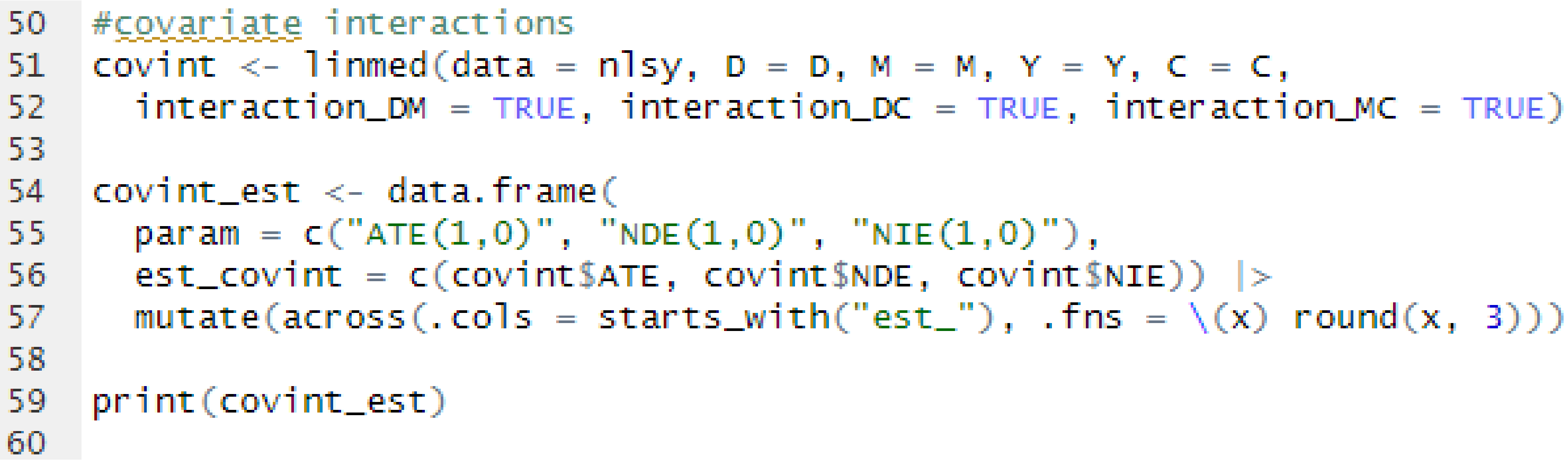
```r
#exposure-mediator interaction
inter <- linmed(data = nlsy, D = D, M = M, Y = Y, C = C, interaction_DM = TRUE)

inter_est <- data.frame(
  param = c("ATE(1,0)", "NDE(1,0)", "NIE(1,0)"),
  est_inter = c(inter$ATE, inter$NDE, inter$NIE)) |>
  mutate(across(.cols = starts_with("est_"), .fns = \(x) round(x, 3)))

print(inter_est)

```



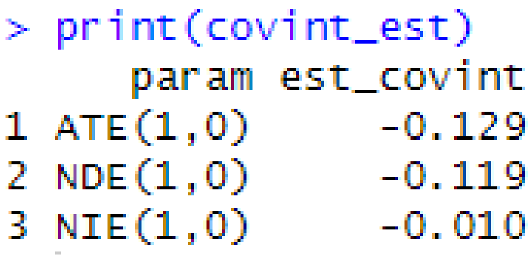
```
> print(inter_est)
     param est_inter
1 ATE(1,0)    -0.083
2 NDE(1,0)    -0.078
3 NIE(1,0)    -0.005
```

Example: NLSY79

- Compute estimates using linear models with covariate interactions



```r
#covariate interactions
covint <- linmed(data = nlsy, D = D, M = M, Y = Y, C = C,
  interaction_DM = TRUE, interaction_DC = TRUE, interaction_MC = TRUE)

covint_est <- data.frame(
  param = c("ATE(1,0)", "NDE(1,0)", "NIE(1,0)"),
  est_covint = c(covint$ATE, covint$NDE, covint$NIE)) |>
  mutate(across(.cols = starts_with("est_"), .fns = \(x) round(x, 3)))

print(covint_est)

```



```
> print(covint_est)
     param est_covint
1 ATE(1,0)     -0.129
2 NDE(1,0)     -0.119
3 NIE(1,0)     -0.010
```

Limitations

- Models that are linear in the parameters may not perform very well when the mediator or outcome is binary, ordinal, or a count
  - In such applications, any linear model is likely incorrect and may provide poor estimates for the true but unknown conditional expected values, leading to misspecification bias

- Consequently, this approach is best suited for applications in which both the mediator and outcome are unbounded and possess equal-interval scaling

- Nevertheless, there are some situations where a linear model still provides a reasonable approximation for the conditional expected value of a binary, ordinal, or count variable

Estimation via simulation

- Natural direct and indirect effects can also be estimated using a simulation approach that is implemented with generalized linear models (GLMs)

- The class of GLMs is broad and subsumes normal linear regression as a special case, but it also includes many nonlinear models, such as logit, probit, and Poisson regression, among others

- This approach to estimation is therefore extremely general and can be used in a wide variety of different applications (i.e., with continuous, binary, ordinal, nominal, or count variables)

Estimation via simulation

- The simulation estimator is implemented through the following series of steps:

  1. Fit models for the mediator and outcome
  2. Simulate potential values for the mediator
  3. Simulate potential outcomes using the simulated values for the mediator
  4. Compute effect estimates using the simulated outcomes

Estimation via simulation

- Step 1: fit models for the mediator and outcome
  - Fit a GLM for the mediator given the baseline confounders and the exposure, denoted by $g(M|C,D)$, and then fit another GLM for the outcome given the baseline confounders, the exposure, and the mediator, denoted by $h(Y|C,D,M)$
    - For example:

$$g(M|c,d) = Bern\left(p = \text{logit}^{-1}(\beta_0 + \beta_1^T c + \beta_2 d)\right)$$

$$h(Y|c,d,m) = Pois\left(\lambda = \exp\left(\gamma_0 + \gamma_1^T c^{\perp} + \gamma_2 d + m(\gamma_3 + \gamma_4 d)\right)\right)$$

  - Let $\hat{g}(M|C,D)$ and $\hat{h}(Y|C,D,M)$ denote these models with their parameters estimated by maximum likelihood

Estimation via simulation

- Step 2: simulate potential values for the mediator
  - For every individual in the sample…
    - simulate $10^3 \leq J \leq 10^4$ copies of $M(d^*)$ from $\hat{g}(M|C, d^*)$ and then…
    - simulate another $10^3 \leq J \leq 10^4$ copies of $M(d)$ from $\hat{g}(M|C, d)$
  - Let $\widetilde{M}_j(d^*)$ and $\widetilde{M}_j(d)$ denote the simulated values (i.e., Monte Carlo draws) of the mediator for each simulation $j = 1, 2, \ldots, J$

Estimation via simulation

- Step 3: simulate potential outcomes
  - For every individual in the sample and for each simulated value of the mediator…
    - simulate one copy of $Y\big(d, M(d)\big)$ from $\hat{h}(Y|C, d, \widetilde{M}_j(d))$ and then…
    - simulate one copy of $Y\big(d^*, M(d^*)\big)$ from $\hat{h}(Y|C, d^*, \widetilde{M}_j(d^*))$ and then…
    - simulate one copy of $Y\big(d, M(d^*)\big)$ from $\hat{h}(Y|C, d, \widetilde{M}_j(d^*))$
  - Let $\tilde{Y}_j\big(d, M(d)\big)$, $\tilde{Y}_j\big(d^*, M(d^*)\big)$, and $\tilde{Y}_j\big(d, M(d^*)\big)$ denote the simulated values (i.e., Monte Carlo draws) of the outcome for each simulation $j = 1, 2, \ldots, J$

Estimation via simulation

- Step 4: compute effect estimates
  - Average the difference between simulated outcomes over simulations and over sample members as follows…

$$\widehat{NDE}(d,d^*)=\frac{1}{nJ}\sum\sum_j\left[\tilde{Y}_j\big(d,M(d^*)\big)-\tilde{Y}_j\big(d^*,M(d^*)\big)\right]$$

$$\widehat{NIE}(d,d^*)=\frac{1}{nJ}\sum\sum_j\left[\tilde{Y}_j\big(d,M(d)\big)-\tilde{Y}_j\big(d,M(d^*)\big)\right]$$

$$\widehat{ATE}(d,d^*)=\frac{1}{nJ}\sum\sum_j\left[\tilde{Y}_j\big(d,M(d)\big)-\tilde{Y}_j\big(d^*,M(d^*)\big)\right]$$

Model specification

- This approach can easily accommodate treatment-mediator interactions, covariate interactions, and nonlinear terms, as well as many different link functions and distribution models

- The steps for implementing the simulation approach are exactly the same as outlined previously, regardless of the particular form of the GLMs used for the mediator and outcome

- The simulation estimator is therefore highly flexible

Summary

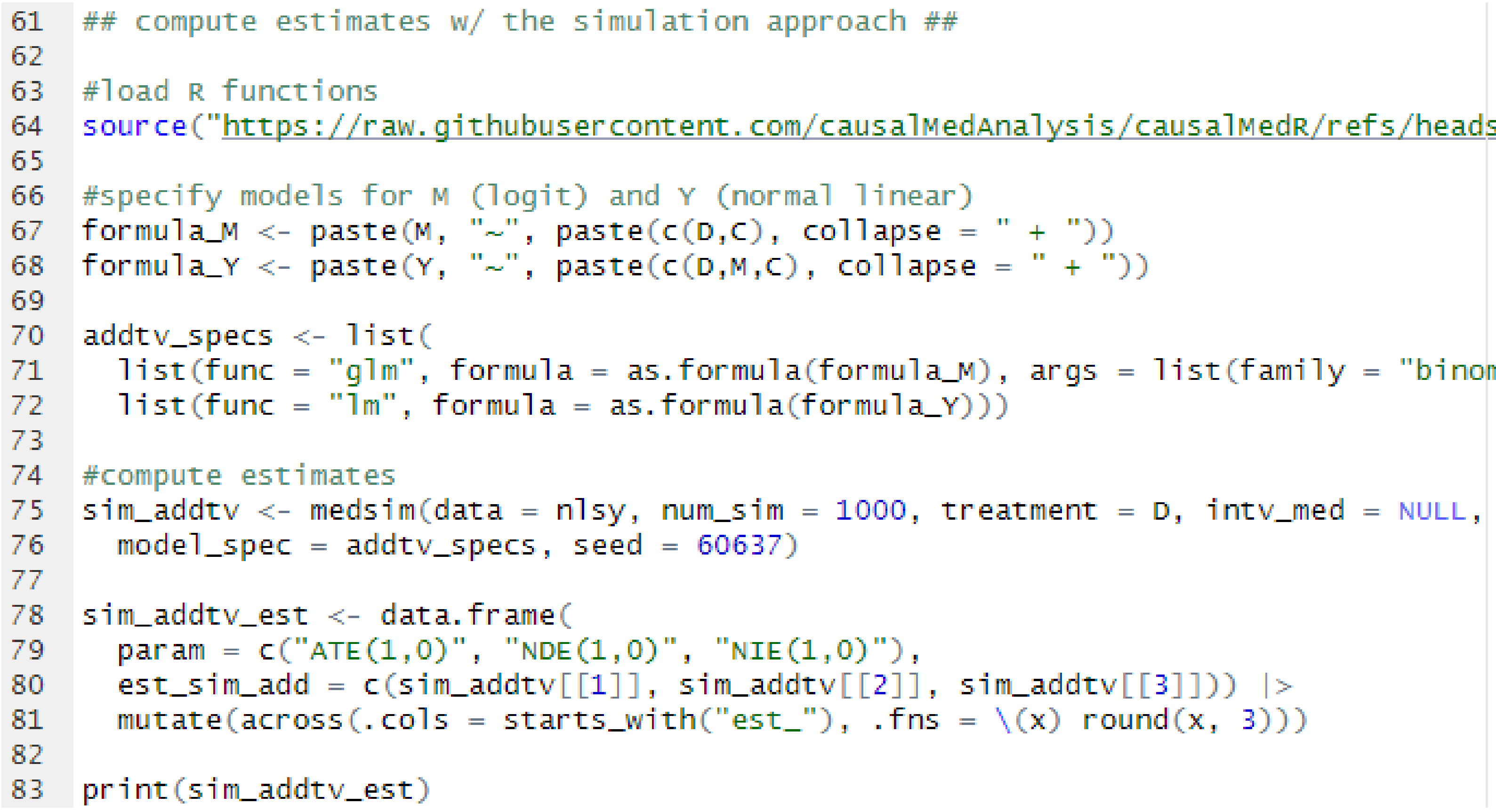
- Natural direct and indirect effects can be estimated via simulation with a broad class of GLMs fit to the sample data by the method of maximum likelihood

- The simulation estimator is consistent provided that…
  - the assumptions required for identification are satisfied
  - the models for the mediator and outcome are correctly specified

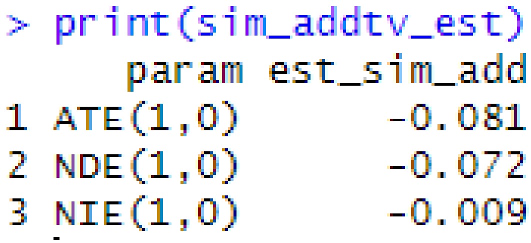
- The class of GLMs is broad, and these models can also easily incorporate exposure-mediator interaction, effect moderation, and nonlinear terms

Example: NLSY79

- Compute natural effects estimates via simulation using a logit model for the mediator and a normal linear model for the outcome



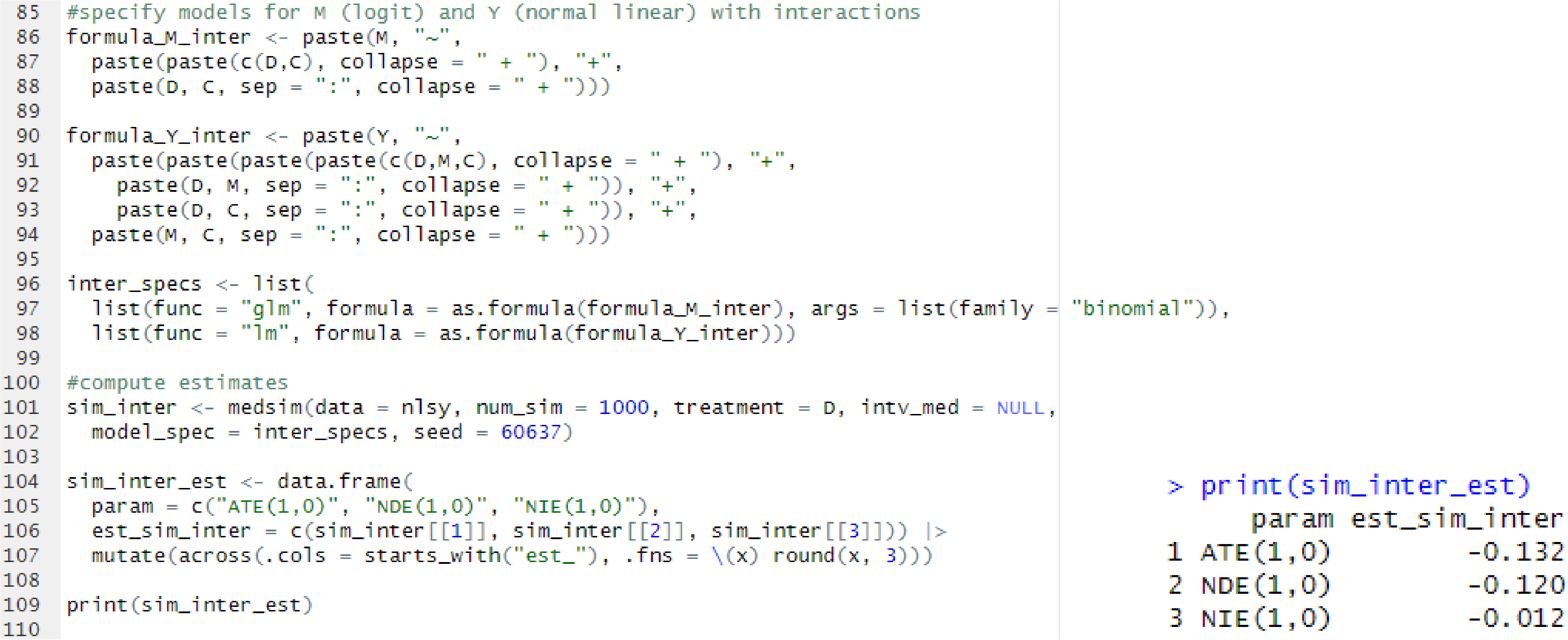
```r
## compute estimates w/ the simulation approach ##

#load R functions
source("https://raw.githubusercontent.com/causalMedAnalysis/causalMedR/refs/heads/main/medsim.R")

#specify models for M (logit) and Y (normal linear)
formula_M <- paste(M, "~", paste(c(D,C), collapse = " + "))
formula_Y <- paste(Y, "~", paste(c(D,M,C), collapse = " + "))

addtv_specs <- list(
  list(func = "glm", formula = as.formula(formula_M), args = list(family = "binomial")),
  list(func = "lm", formula = as.formula(formula_Y)))

#compute estimates
sim_addtv <- medsim(data = nlsy, num_sim = 1000, treatment = D, intv_med = NULL,
  model_spec = addtv_specs, seed = 60637)

sim_addtv_est <- data.frame(
  param = c("ATE(1,0)", "NDE(1,0)", "NIE(1,0)"),
  est_sim_add = c(sim_addtv[[1]], sim_addtv[[2]], sim_addtv[[3]])) |>
  mutate(across(.cols = starts_with("est_"), .fns = \(x) round(x, 3)))

print(sim_addtv_est)
```



```
> print(sim_addtv_est)
     param est_sim_add
1 ATE(1,0)      -0.081
2 NDE(1,0)      -0.072
3 NIE(1,0)      -0.009
```

Example: NLSY79

- Compute estimates via simulation using a logit model for the mediator and a normal linear model for the outcome, both with interactions



```r
#specify models for M (logit) and Y (normal linear) with interactions
formula_M_inter <- paste(M, "~",
  paste(paste(c(D,C), collapse = " + "), "+",
  paste(D, C, sep = ":", collapse = " + ")))

formula_Y_inter <- paste(Y, "~",
  paste(paste(paste(paste(c(D,M,C), collapse = " + "), "+",
    paste(D, M, sep = ":", collapse = " + ")), "+",
    paste(D, C, sep = ":", collapse = " + ")), "+",
  paste(M, C, sep = ":", collapse = " + ")))

inter_specs <- list(
  list(func = "glm", formula = as.formula(formula_M_inter), args = list(family = "binomial")),
  list(func = "lm", formula = as.formula(formula_Y_inter)))

#compute estimates
sim_inter <- medsim(data = nlsy, num_sim = 1000, treatment = D, intv_med = NULL,
  model_spec = inter_specs, seed = 60637)

sim_inter_est <- data.frame(
  param = c("ATE(1,0)", "NDE(1,0)", "NIE(1,0)"),
  est_sim_inter = c(sim_inter[[1]], sim_inter[[2]], sim_inter[[3]])) |>
  mutate(across(.cols = starts_with("est_"), .fns = \(x) round(x, 3)))

print(sim_inter_est)
```

```
> print(sim_inter_est)
     param est_sim_inter
1 ATE(1,0)        -0.132
2 NDE(1,0)        -0.120
3 NIE(1,0)        -0.012
```

Estimation via weighting

- In contrast to the regression and simulation estimators, which are based on models for the mediator and outcome, weighting estimators are implemented with models for the exposure

- These models are used to construct a set of weights that transform the empirical distribution of the sample data in ways that emulate different hypothetical experiments

- The effects of interest are estimated by comparing the mean of the outcome across differently weighted samples

Estimation via weighting

- The weighting estimator is implemented through the following series of steps:

  1. Fit two different models for the exposure

  2. Compute predicted probabilities of exposure from each model

  3. Use the exposure probabilities to construct a set of inverse probability weights (IPWs)

  4. Compute effect estimates by comparing weighted means of the outcome

Estimation via weighting

- Step 1: fit models for the exposure
  - Fit a GLM for the exposure given the baseline confounders, denoted by $f(D|C)$
  - Next, fit another GLM for the exposure given the baseline confounders and the mediator, denoted by $s(D|C,M)$; if, for example, the exposure is binary, then $f(D|C)$ and $s(D|C,M)$ might be logit or probit models
  - Let $\hat{f}(D|C)$ and $\hat{s}(D|C,M)$ denote these models with their parameters estimated by maximum likelihood

Estimation via weighting

- Step 2: compute predicted probabilities of exposure
  - For each sample member, use $\hat{f}(D|C)$ to predict…
    - the probability of exposure to d given the baseline confounders, denoted by $\hat{P}(d|C)$
    - the probability of exposure to d^* given the baseline confounders, denoted by $\hat{P}(d^*|C)$
  - For each sample member, use $\hat{s}(D|C,M)$ to predict…
    - the probability of exposure to d given the baseline confounders and mediator, denoted by $\hat{P}(d|C,M)$
    - the probability of exposure to d^* given the baseline confounders and mediator, denoted by $\hat{P}(d^*|C,M)$

Estimation via weighting

- Step 3: construct IPWs
  - Among sample members with $D = d^*$, compute…
    - $\hat{w}_1 = \frac{1}{\hat{P}(d^*|C)}$
  - Among sample members with $D = d$, compute…
    - $\hat{w}_2 = \frac{1}{\hat{P}(d|C)}$
    - $\hat{w}_3 = \frac{\hat{P}(d^*|C,M)}{\hat{P}(d|C,M)\hat{P}(d^*|C)}$

Estimation via weighting

- Step 4: compute effect estimates
  - Compute differences between weighted means of the observed outcome as follows…

$$\widehat{NDE}(d,d^*) = \frac{\sum I(D=d)\widehat{w}_3 Y}{\sum I(D=d)\widehat{w}_3} - \frac{\sum I(D=d^*)\widehat{w}_1 Y}{\sum I(D=d^*)\widehat{w}_1}$$

$$\widehat{NIE}(d,d^*) = \frac{\sum I(D=d)\widehat{w}_2 Y}{\sum I(D=d)\widehat{w}_2} - \frac{\sum I(D=d)\widehat{w}_3 Y}{\sum I(D=d)\widehat{w}_3}$$

$$\widehat{ATE}(d,d^*) = \frac{\sum I(D=d)\widehat{w}_2 Y}{\sum I(D=d)\widehat{w}_2} - \frac{\sum I(D=d^*)\widehat{w}_1 Y}{\sum I(D=d^*)\widehat{w}_1}$$

Estimation via weighting

- Step 4: compute effect estimates
  - Compute differences between weighted means of the observed outcome as follows…

$$\widehat{NDE}(d,d^*) = \frac{\sum I(D=d)\hat{w}_3 Y}{\sum I(D=d)\hat{w}_3} - \frac{\sum I(D=d^*)\hat{w}_1 Y}{\sum I(D=d^*)\hat{w}_1} = \hat{E}\left(Y\left(d, M(d^*)\right)\right) - \hat{E}\left(Y\left(d^*, M(d^*)\right)\right)$$

$$\widehat{NIE}(d,d^*) = \frac{\sum I(D=d)\hat{w}_2 Y}{\sum I(D=d)\hat{w}_2} - \frac{\sum I(D=d)\hat{w}_3 Y}{\sum I(D=d)\hat{w}_3} = \hat{E}\left(Y\left(d, M(d)\right)\right) - \hat{E}\left(Y\left(d, M(d^*)\right)\right)$$

$$\widehat{ATE}(d,d^*) = \frac{\sum I(D=d)\hat{w}_2 Y}{\sum I(D=d)\hat{w}_2} - \frac{\sum I(D=d^*)\hat{w}_1 Y}{\sum I(D=d^*)\hat{w}_1} = \hat{E}\left(Y\left(d, M(d)\right)\right) - \hat{E}\left(Y\left(d^*, M(d^*)\right)\right)$$

Weighted pseudo-samples

- Weighting sample members with $D = d^*$ by $\widehat{w}_1 = \frac{1}{\hat{P}(d^*|C)}$ transforms the distribution of C among this group to resemble the total sample

- Weighting sample members with $D = d$ by $\widehat{w}_2 = \frac{1}{\hat{P}(d|C)}$ transforms the distribution of C among this group to resemble the total sample

- Weighting sample members with $D = d$ by $\widehat{w}_3 = \frac{\hat{P}(d^*|C, M)}{\hat{P}(d|C, M)\hat{P}(d^*|C)}$ transforms…
  - the distribution of C among this group to resemble the total sample
  - the distribution of M to resemble that observed among those with $D = d^*$

Stabilized weights

- The inverse probability weights defined previously can yield imprecise and unstable estimates in finite samples

- This is because the weights involve inverse probabilities, which can be very large when the probability of exposure is very small, distorting the effect estimates

- The challenges stemming from extreme weights can be partly mitigated by using stabilized versions of the inverse probabilities, which involves scaling them down to have a mean equal to 1 and lower variance

Stabilized weights

- Stabilized versions of the IPWs can be expressed as follows:
  - $\widehat{sw}_1 = \frac{\hat{P}(d^*)}{\hat{P}(d^*|C)}$
  - $\widehat{sw}_2 = \frac{\hat{P}(d)}{\hat{P}(d|C)}$
  - $\widehat{sw}_3 = \frac{\hat{P}(d^*|C,M)\hat{P}(d^*)}{\hat{P}(d|C,M)\hat{P}(d^*|C)}$

Censored weights

- The performance of weighting estimators can usually be improved even further by censoring the weights

- Censoring the weights involves top and bottom coding very large and very small weights, respectively, to reduce to the influence of outliers, and by extension, to improve the precision of effect estimates

- In general, the greater the degree of censoring, the more stable are the weights, and consequently, also the effect estimates based thereon, but this improved stability comes at the cost of greater systematic bias

Ratio of mediator probability weighting

- The third inverse probability weight, $w_3 = \frac{P(d^*|C,M)}{P(d|C,M)P(d^*|C)}$, can also be expressed as follows:

$$w_3^{\text{alt}} = \frac{P(M|C,d^*)}{P(M|C,d)P(d|C)}$$

- This equivalence points toward an alternative weighting estimator for natural direct and indirect effects based on distribution models for the exposure and mediator

- Known as ratio of mediator probability weighting, due to Hong (2015)

Summary

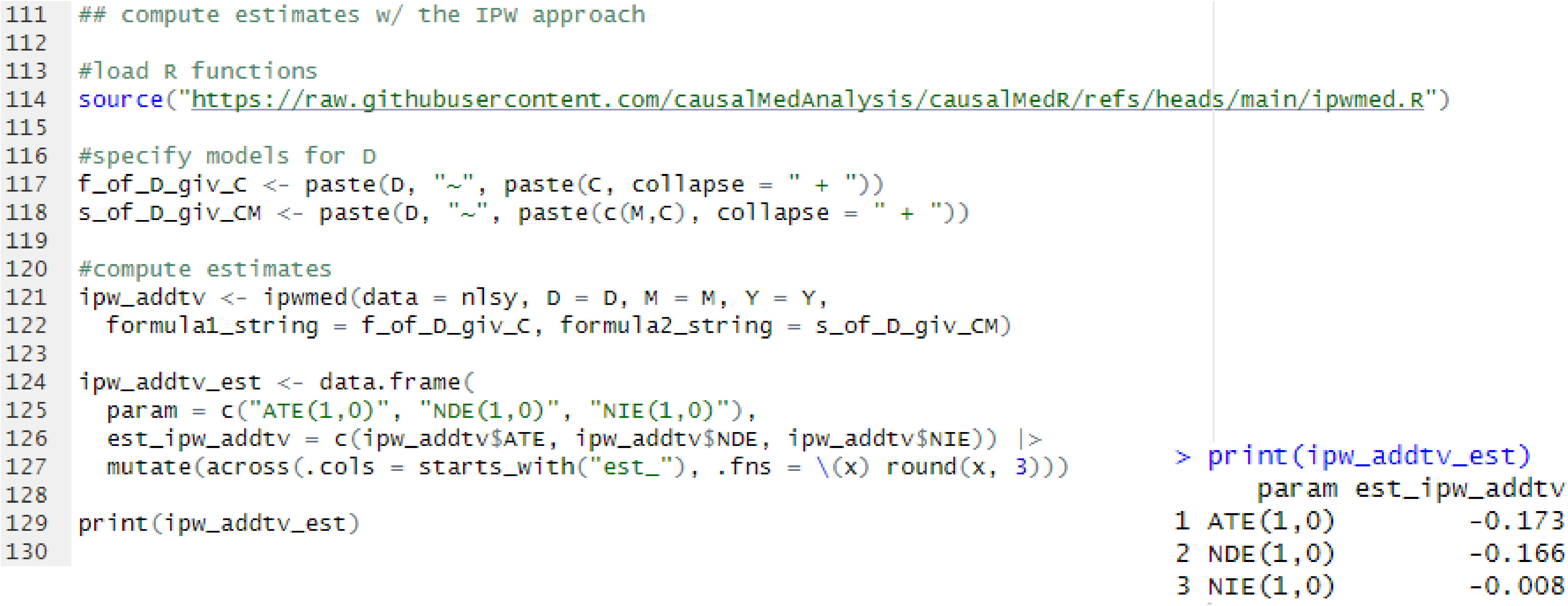
- Natural direct and indirect effects can be estimated via weighting with two different GLMs for the probability of exposure

- These estimators are consistent provided that the assumptions required for identification are satisfied and provided that the models used for the exposure are correctly specified

- Limitations
  - Difficult to use and often unstable with continuous or many valued exposures
  - Highly sensitive to model misspecification

Example: NLSY79

- Compute estimates using IPW with logit models for the exposure



```r
## compute estimates w/ the IPW approach

#load R functions
source("https://raw.githubusercontent.com/causalMedAnalysis/causalMedR/refs/heads/main/ipwmed.R")

#specify models for D
f_of_D_giv_C <- paste(D, "~", paste(C, collapse = " + "))
s_of_D_giv_CM <- paste(D, "~", paste(c(M,C), collapse = " + "))

#compute estimates
ipw_addtv <- ipwmed(data = nlsy, D = D, M = M, Y = Y,
  formula1_string = f_of_D_giv_C, formula2_string = s_of_D_giv_CM)

ipw_addtv_est <- data.frame(
  param = c("ATE(1,0)", "NDE(1,0)", "NIE(1,0)"),
  est_ipw_addtv = c(ipw_addtv$ATE, ipw_addtv$NDE, ipw_addtv$NIE)) |>
  mutate(across(.cols = starts_with("est_"), .fns = \(x) round(x, 3)))

print(ipw_addtv_est)
```

```
> print(ipw_addtv_est)
     param est_ipw_addtv
1 ATE(1,0)        -0.173
2 NDE(1,0)        -0.166
3 NIE(1,0)        -0.008
```

Inference

- Thus far we have focused exclusively on point estimation, but point estimates may differ from their target estimands due to the random variability that arises from sampling

- Recall that every estimator is a function of a random sample and, therefore, has a probability distribution itself, known as the sampling distribution

- The shape and spread of this distribution reflect the degree of uncertainty, or sampling variability, afflicting an estimator, and determines the results of hypothesis tests and confidence intervals

The nonparametric bootstrap

- Mediation estimators are complex and not all of them have known sampling distributions that can be analytically derived

- Thus, we will use the nonparametric bootstrap to construct sampling distributions for our estimators, which can then be used for inference

- The nonparametric bootstrap is a highly versatile approach that constructs an approximate sampling distribution using only the data from our original sample

The nonparametric bootstrap

- To construct a bootstrap sample, individuals in the original data are re-sampled with replacement
- Then, using this bootstrap sample, an estimate or set of estimates are computed following the same procedures with which the original data were analyzed
- This process is repeated many times, with new bootstrap samples constructed at each step via sampling with replacement from the original data
  - After many repetitions, a representation of the sampling distribution can be constructed empirically using the full set of estimates, which is known as the *bootstrap distribution*

Bootstrap confidence intervals

- To construct a bootstrap confidence interval:

1. Randomly sample the observed data with replacement

2. Compute an estimate for the target parameter of interest, denoted as $\hat{\theta}_b$

3. Repeat the prior steps many times

   - Specifically, for $b = 1,2, \dots, B$, where $10^3 \leq B \leq 10^4$, repeat steps 1 and 2, saving $\hat{\theta}_b$ from each iteration, to construct a bootstrap distribution

4. Compute percentiles of the bootstrap distribution

   - The upper and lower bounds of a τ-percent confidence interval are given by the $0.5(100 - \tau)$th and $\big(100 - 0.5(100 - \tau)\big)$th percentiles, respectively, of the bootstrap estimates $\{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_B\}$

Bootstrap p-values

- Another approach to drawing inferences in the presence of sampling uncertainty is to perform a null hypothesis test

- Hypothesis tests begin by stating a null hypothesis about an estimand

- Then, to evaluate whether the data are inconsistent with this hypothesis, we summarize the degree of evidence they provide against it using a p-value

  - A *p-value* is the probability of observing a point estimate as or even more extreme than the one obtained from the sample data, assuming the null hypothesis is true

  - If this probability is very small, it indicates that obtaining the point estimate we computed from the observed data is unlikely given the null hypothesis, and thus we would conclude that the sample data are inconsistent with this hypothesis and reject it

Bootstrap p-values

- To construct a bootstrap p-value:

    1. Randomly sample the observed data with replacement

    2. Compute an estimate for the target parameter of interest, denoted as $\hat{\theta}_b$

    3. Repeat the prior steps many times

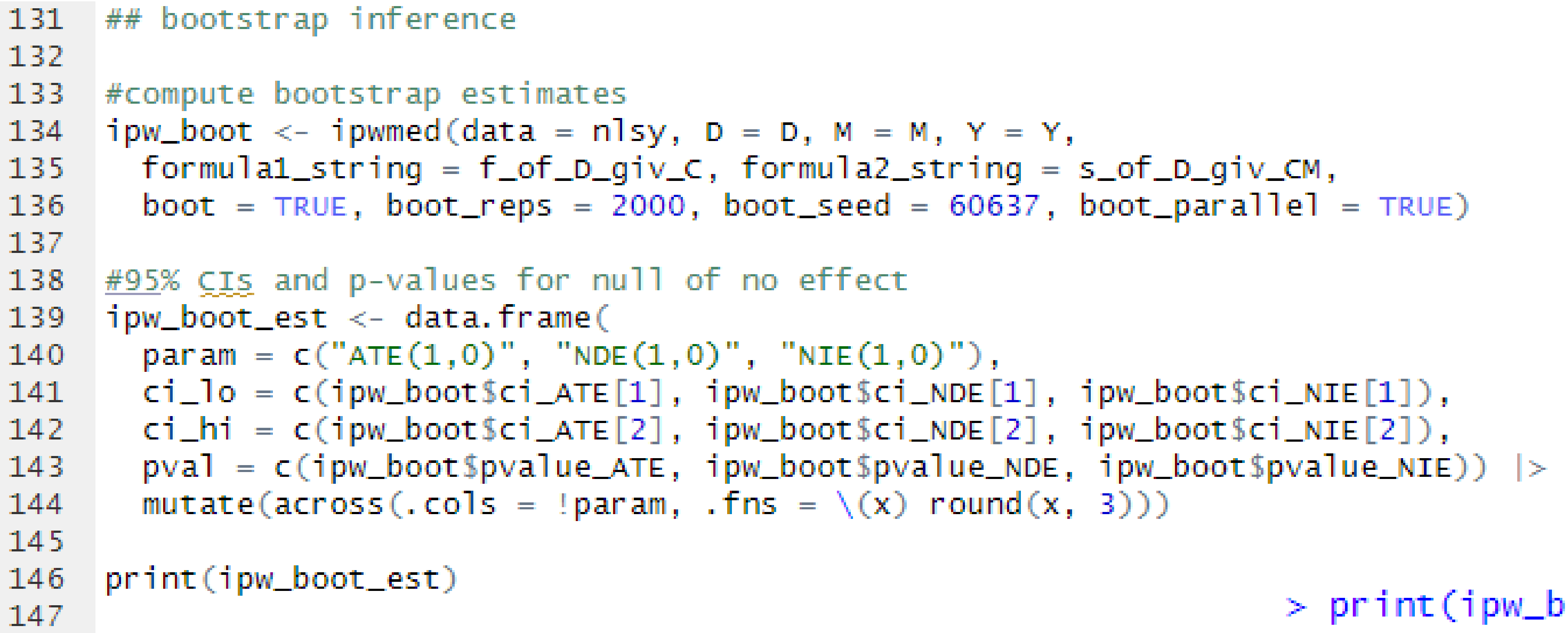
        - Specifically, for $b = 1,2,\ldots,B$, where $10^3 \leq B \leq 10^4$, repeat steps 1 and 2, saving $\hat{\theta}_b$ from each iteration, to construct a bootstrap distribution

    4. Compute a p-value by inverting a bootstrap confidence interval

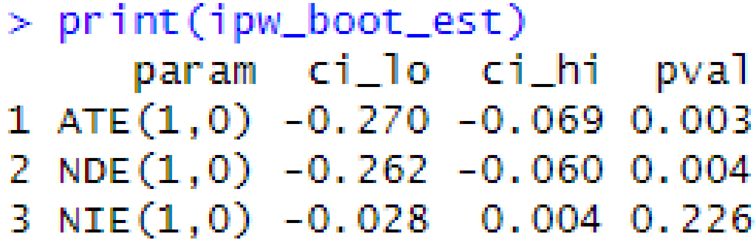
        - Compute $p = 2 \times \min\left(\hat{\pi}(\hat{\theta}_b < \theta_{\text{null}}), \hat{\pi}(\hat{\theta}_b > \theta_{\text{null}})\right)$, where $\hat{\pi}(\hat{\theta}_b < \theta_{\text{null}})$ and $\hat{\pi}(\hat{\theta}_b > \theta_{\text{null}})$ respectively denote the proportion of bootstrap estimates that are less than, and greater than, the hypothesized value of the target parameter

Example: NLSY79

- Compute 95% confidence intervals and p-values for tests of the null hypothesis that the natural effects are zero using IPW and the bootstrap



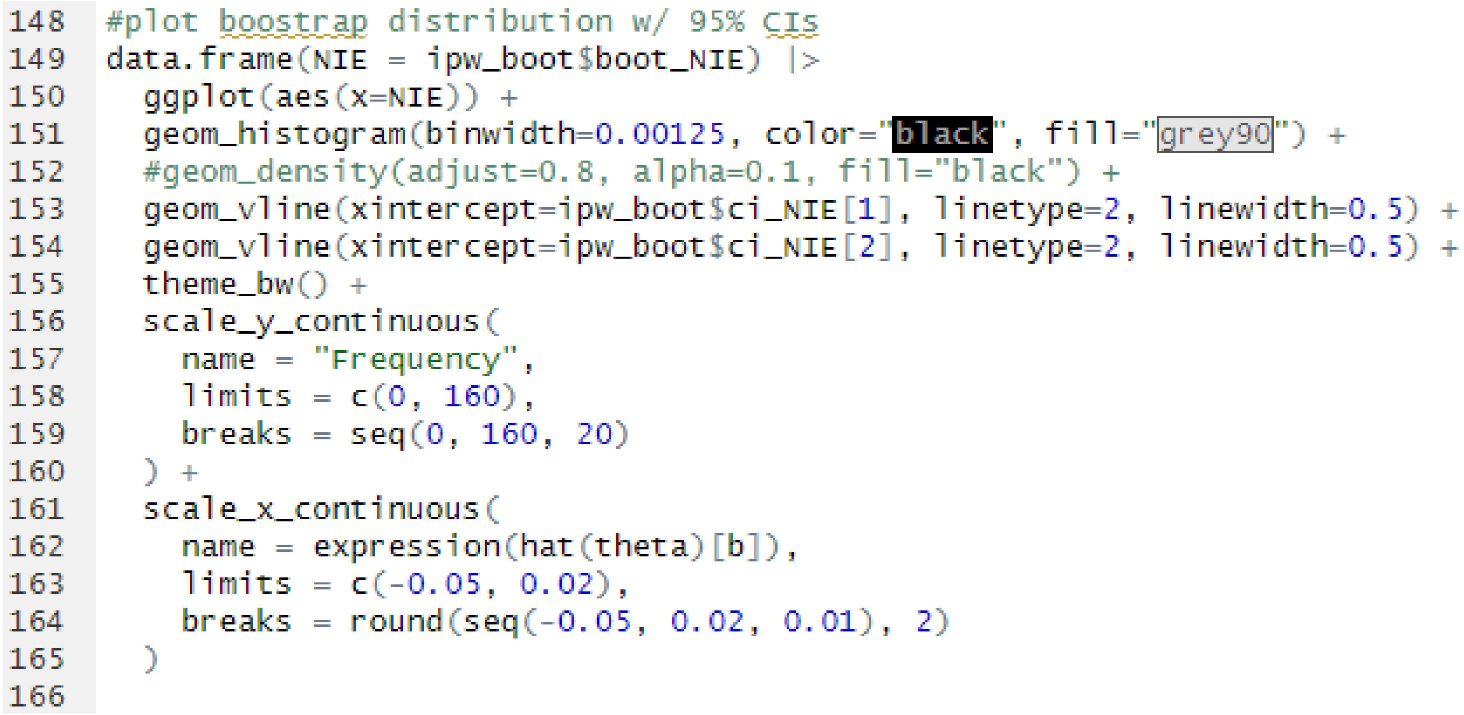
```r
## bootstrap inference

#compute bootstrap estimates
ipw_boot <- ipwmed(data = nlsy, D = D, M = M, Y = Y,
  formula1_string = f_of_D_giv_C, formula2_string = s_of_D_giv_CM,
  boot = TRUE, boot_reps = 2000, boot_seed = 60637, boot_parallel = TRUE)

#95% CIs and p-values for null of no effect
ipw_boot_est <- data.frame(
  param = c("ATE(1,0)", "NDE(1,0)", "NIE(1,0)"),
  ci_lo = c(ipw_boot$ci_ATE[1], ipw_boot$ci_NDE[1], ipw_boot$ci_NIE[1]),
  ci_hi = c(ipw_boot$ci_ATE[2], ipw_boot$ci_NDE[2], ipw_boot$ci_NIE[2]),
  pval = c(ipw_boot$pvalue_ATE, ipw_boot$pvalue_NDE, ipw_boot$pvalue_NIE)) |>
  mutate(across(.cols = !param, .fns = \(x) round(x, 3)))

print(ipw_boot_est)

```



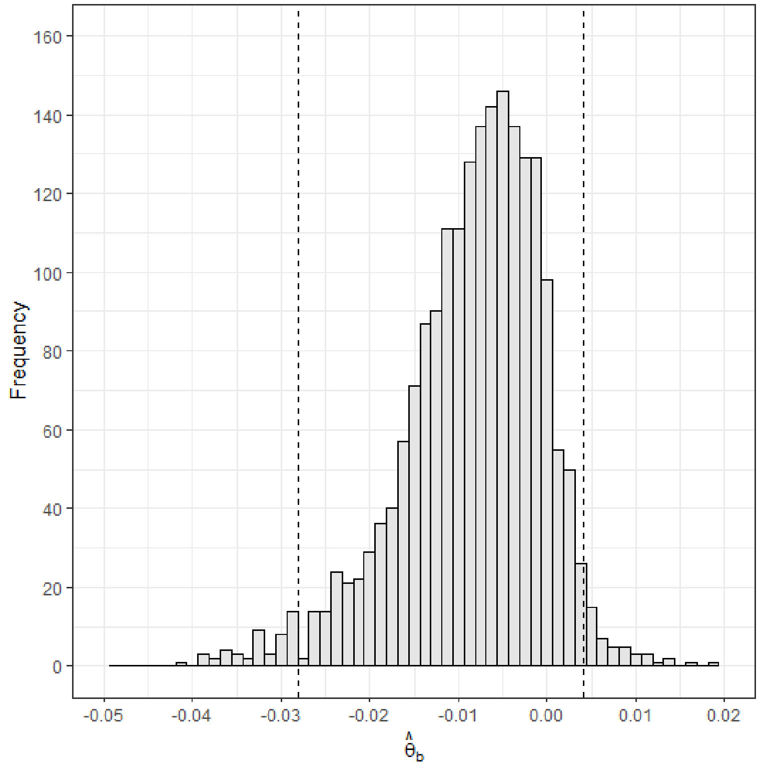
```
> print(ipw_boot_est)
     param  ci_lo  ci_hi  pval
1 ATE(1,0) -0.270 -0.069 0.003
2 NDE(1,0) -0.262 -0.060 0.004
3 NIE(1,0) -0.028  0.004 0.226
```

Example: NLSY79

- Plot the bootstrap distribution and 95% CI for the natural indirect effect

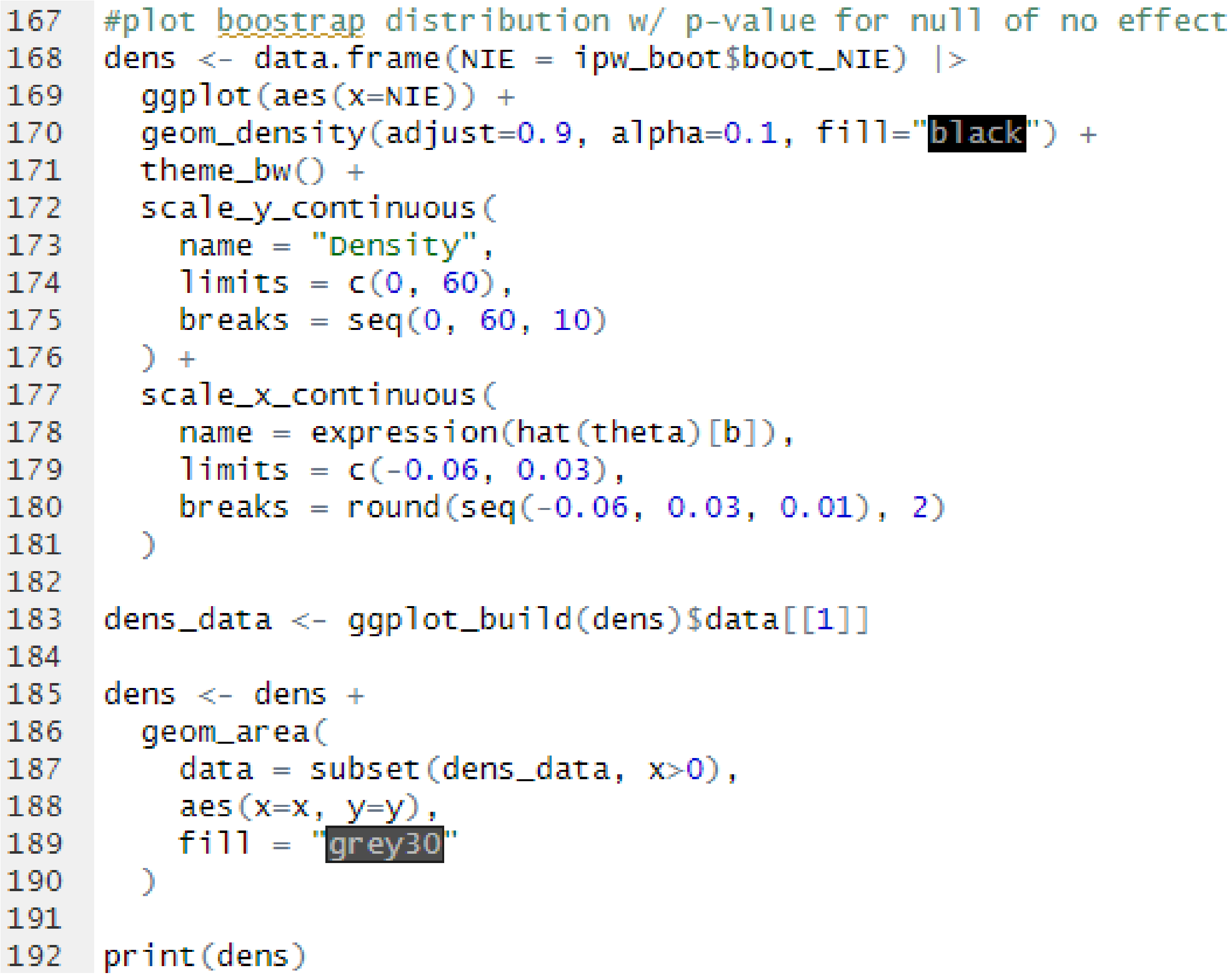


```
#plot boostrap distribution w/ 95% CIs
data.frame(NIE = ipw_boot$boot_NIE) |>
  ggplot(aes(x=NIE)) +
  geom_histogram(binwidth=0.00125, color="black", fill="grey90") +
  #geom_density(adjust=0.8, alpha=0.1, fill="black") +
  geom_vline(xintercept=ipw_boot$ci_NIE[1], linetype=2, linewidth=0.5) +
  geom_vline(xintercept=ipw_boot$ci_NIE[2], linetype=2, linewidth=0.5) +
  theme_bw() +
  scale_y_continuous(
    name = "Frequency",
    limits = c(0, 160),
    breaks = seq(0, 160, 20)
  ) +
  scale_x_continuous(
    name = expression(hat(theta)[b]),
    limits = c(-0.05, 0.02),
    breaks = round(seq(-0.05, 0.02, 0.01), 2)
  )
```

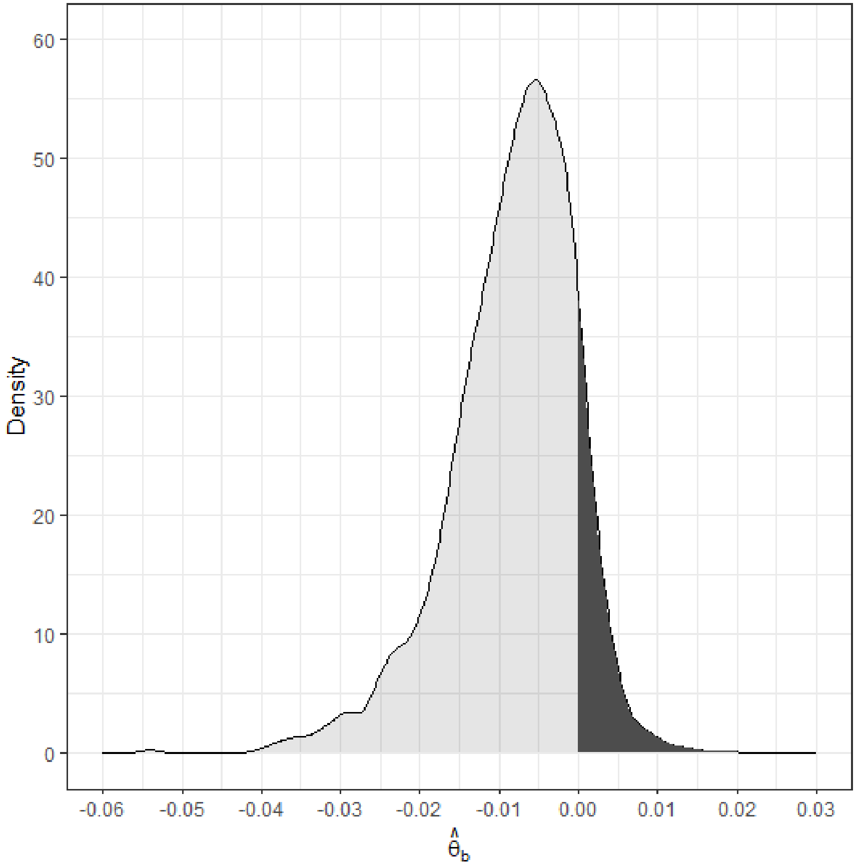


Example: NLSY79

- Plot the bootstrap distribution and p-value for the natural indirect effect



```
#plot boostrap distribution w/ p-value for null of no effect
dens <- data.frame(NIE = ipw_boot$boot_NIE) |>
  ggplot(aes(x=NIE)) +
  geom_density(adjust=0.9, alpha=0.1, fill="black") +
  theme_bw() +
  scale_y_continuous(
    name = "Density",
    limits = c(0, 60),
    breaks = seq(0, 60, 10)
  ) +
  scale_x_continuous(
    name = expression(hat(theta)[b]),
    limits = c(-0.06, 0.03),
    breaks = round(seq(-0.06, 0.03, 0.01), 2)
  )

dens_data <- ggplot_build(dens)$data[[1]]

dens <- dens +
  geom_area(
    data = subset(dens_data, x>0),
    aes(x=x, y=y),
    fill = "grey30"
  )

print(dens)
```



Code

- Learn more about linmed, medsim, and ipwmed on github
  - R:
    - https://github.com/causalMedAnalysis/causalMedR
  - Stata:
    - https://github.com/causalMedAnalysis/linmed
    - https://github.com/causalMedAnalysis/medsim
    - https://github.com/causalMedAnalysis/ipwmed