Denis Ostroushko - HW5

Introduction

Imputation and Variable Processing

For the purpose of this assignment we retain the same imputation schemes we used in the previous two assignments. We will use imputation with the median of observed values and replace missing values with modes for categorical predictors.

Since we will need to create two models that have all possible confounders in the data set, we need to be careful with variable inclusion. We drop variable `Hisp` because it is highly correlated with other variables that contain race and ethnic information. We also `Drug.Add` due to the issues with its imputation. When imputed with the most common level "No", which indicated no drug use, this variable has one unique level. Such zero variance predictors can cause problems with fitting models, so we will avoid using it in our analyses.

Variables `BMI`, `BL.Cig.Day`, `BL.Drks.Day`, `N.living.kids` are imputed with medians like `N.prev.preg`, `Birthweight` in the previous assignments.

`Use.Alc` is imputed with a mode like `Race_ethnicity`, `Use.Tob`

New variable for assignment 5: `V5..BOP`: which has 91 missing values. We will impute the variable with the median of available data points.

Note: I left small steps and explanations for myself for future use

Problem 1

1 - A

I fit a simple additive model with no interactions for expected values of mediation. I provide code for the regression model in the chunk below:

```r
mediator_model <-
  lm(
    mediator ~
      Group + Race_ethnicity + Public.Asstce + Use.Tob + N.prev.preg +
      Live.PTB + BL.GE + BL..BOP + BL..PD.4 + BL..CAL.3,

    data = data
  )
```

1 - B

As in the previous assignments, I consider the same roster of predictors for the outcome models. Note that I add mediator as a predictor of outcome now, as well as an interaction of a mediator and a treatment assignment variable. I also fit *extended* version of outcome models that include interaction of a group variable with mode than just mediator variable. I will provide reasoning for this modeling choice later on.

Code below fits regression outcome model for the pregnancy outcome:

```r
pregnancy_model <-
  glm(
    `Preg.ended...37.wk` ~ Group * mediator +
      Race_ethnicity + Public.Asstce + Use.Tob + N.prev.preg +
      Live.PTB + BL.GE + BL..BOP + BL..PD.4 + BL..CAL.3,

    data = data,
    family = "binomial"
  )

pregnancy_model_extended <-
  glm(
    `Preg.ended...37.wk` ~ Group * (mediator +
      Race_ethnicity + Public.Asstce + Use.Tob + N.prev.preg +
      Live.PTB + BL.GE + BL..BOP + BL..PD.4 + BL..CAL.3),

    data = data,
    family = "binomial"
  )
```

We also obtain outcome regression models for the birth weight outcome:

```r
borthweight_model <-
  glm(
    Birthweight ~
    Group * mediator + Race_ethnicity + Public.Asstce + Use.Tob +
              N.prev.preg + Live.PTB + BL.GE + BL..BOP +
              BL..PD.4 + BL..CAL.3,

    data = data,
    family = "gaussian"
  )

borthweight_model_extended <-
  glm(
    Birthweight ~
    Group * (mediator + Race_ethnicity + Public.Asstce + Use.Tob +
              N.prev.preg + Live.PTB + BL.GE + BL..BOP +
              BL..PD.4 + BL..CAL.3),

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

1 - C

In this section I first estimate *CDE* as a fucntion of some fixed mediator value using simpler, *not extended*, regressio models. I do this in order to compare two approaches:

- Approach 1: obtain regression coefficients from the outcome model for the treatment variable and interaction between mediator and outcome. Use a simple formula provided in the lecture slides: $CDE(m) = \theta_{treatment} + \theta_{interaction} \times m$
- Approach 2: Holding mediator variable constand, i.e. assign the same value for everyone in the available sample of data, estiamte $E(Y^{1,m})$ and $E(Y^{0,m})$, and estimate $CDE(m) = E(Y^{1,m}) - E(Y^{0,m})$

Using *extended* versions of the model, approach one is no longer possible.

Since CDE is a function of mediator, I estimate CDE for a range of possible mediator values. I consider using sample mean $\pm$ 1, 2, 3 standard errors of the mean. These quantities are estimatef from the sample. Mean of mediator variable is 9.04 and stardard error is 0.79. In order to flexibly estiamte 7 CDE values, I use a function. You can see code for `cde_estimator` fucntion in the appendix.

Possible values of mediator considered are printed below:

```
[1]  6.67  7.46  8.25  9.04  9.84 10.63 11.42
```

Birthweight CDE - Approach 1

Seven CDE estiamtes for each level of medaitor value usign regression coefficients are printed below:

```
[1] -0.06486996  5.53496767 11.13480530 16.73464293 22.33448056 27.93431819
[7] 33.53415582
```

As the average value of mediator increases, CDE for birthweight increases as well.

Birthweight CDE - Approach 2

Seven CDE estiamtes for each level of medaitor value using expected values of counterfactual outcomes are printed below:

```
[1] -0.06486996  5.53496767 11.13480530 16.73464293 22.33448056 27.93431819
[7] 33.53415582
```

Using a differenct method, we observed the same results.

Approach 1 and 2 summary

I approached the problem of estimating CDE for the birth weight outcome using simple method first because this is the only setting that would enable me to verify that approach 1 and 2 procude the same result. Additionally, comparison of approach 1 and 2 is only prossible here beucase we use an identify link for the mean in the outcome model. Pregnancy model will use a logit link function, which will not allow us to extract coefficients and plug them into the formula easily.

Birthweight CDE with Extended Outcome Model

We now employ extended version of outcome models for pregnancy and birth weight to get seven estimates of CDE. We then compare birth weight CDE estimates with estiamtes from previous section.

Birth weight CDE estiamtes for the two models are printed below:

```
[1] "Extended model based estiamtes"
```

```
[1] -25.073824 -18.191972 -11.310121  -4.428270   2.453582   9.335433  16.217285
```

```
[1] "Reduced model based estiamtes"
```

```
[1] -0.06486996  5.53496767 11.13480530 16.73464293 22.33448056 27.93431819
[7] 33.53415582
```

The two methods produce different values of CDE estiamtes. However, as we can see, they tell the same story: as the value of mediator that we control increaes, CDE increases as well.

Pregnancy CDE with Extended Outcome Model

Since pregagncy outcome model uses logit link for the mean fucntion, I only print out CDE estimates. Due to the use of *expit* fucntion, exact formula of CDE take on a complex form. Estiamtes are printed below:

```
[1] "Extended model based estiamtes"
```

```
[1] 0.025313377 0.021603950 0.017959638 0.014379947 0.010864359 0.007412326
[7] 0.004023276
```

```
[1] "Reduced model based estiamtes"
```

```
[1] 0.028730967 0.024958549 0.021251651 0.017609730 0.014032219 0.010518527
[7] 0.007068043
```

Estimates using two version of the outcome pregnancy model produce more similar results here. They also show the same trend: as the value of the mediator increases, CDE decreases

Birthweight NIE

I again employ function to reuse the models and get NIE estiamte. This is an unconditional quantity, so we will get just the one value. Birthweight NIE is given below:

```
[1] 93.8044
```

Pregnancy - NIE

Pregnancy NIE

```
[1] -0.08412877
```

All results

In this section I oresent final answers. Using bootstrap, I obtain standard errors for the estiamtes of CDE and NIE for each outocme. Since we obtained seven CDE estiamtes, I obtain standard error for each of the seven values.

Results are shown on Figure 2 and Figure 1

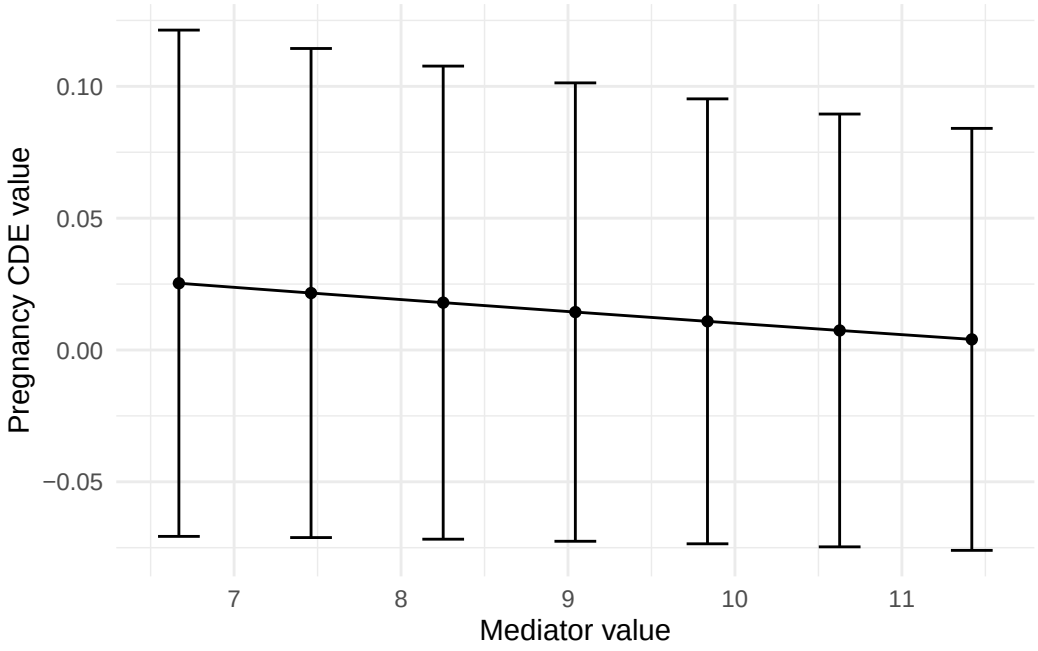


Figure 1: We observe similar and high variance for each level of control mediator value

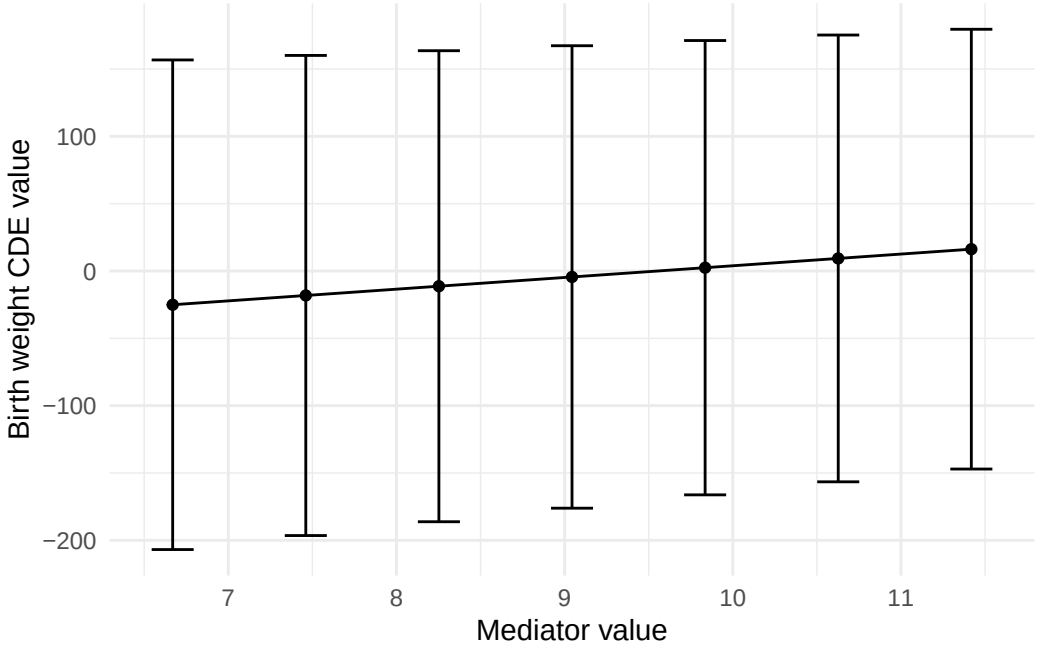


Figure 2: We observe similar and high variance for each level of control mediator value

And now we are ready to summarize the resutls. For each CDE result, I hold the value of mediator at the sample mean, which I stated earlier.

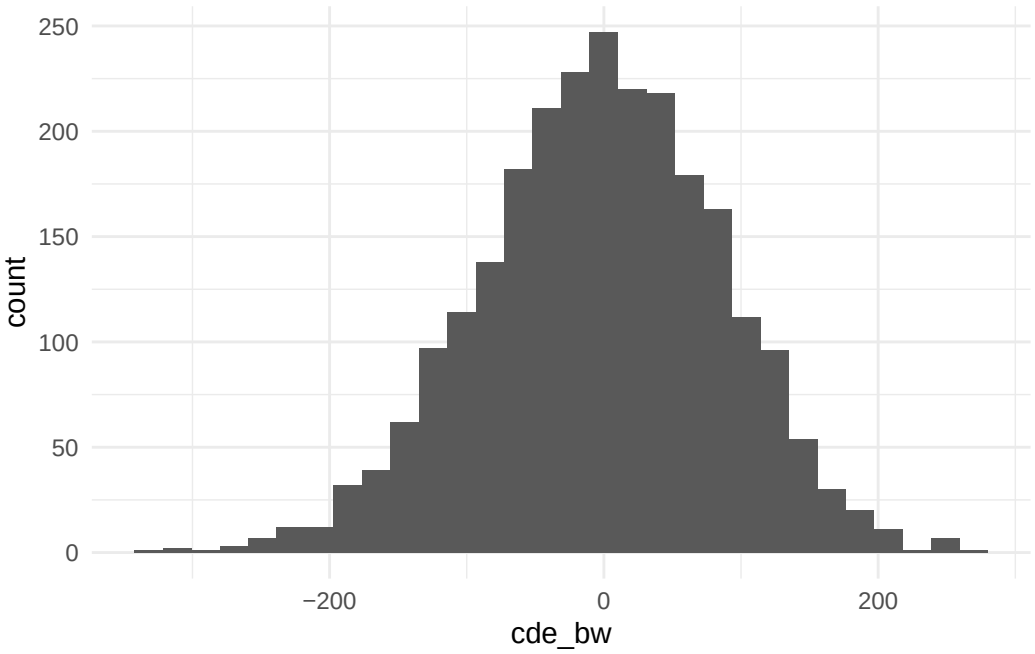
- Birth weight CDE at sample mean:

```
[1] -4.42827
```

- Birth weight CDE standard error:

```
[1] 87.61698
```

- Birth weight CDE bootstrap sampling distribution. Slightly skewed, but okay to use:



- Birth weight CDE 95% confidence interval:

```
[1] "(167.3, -176.16)"
```

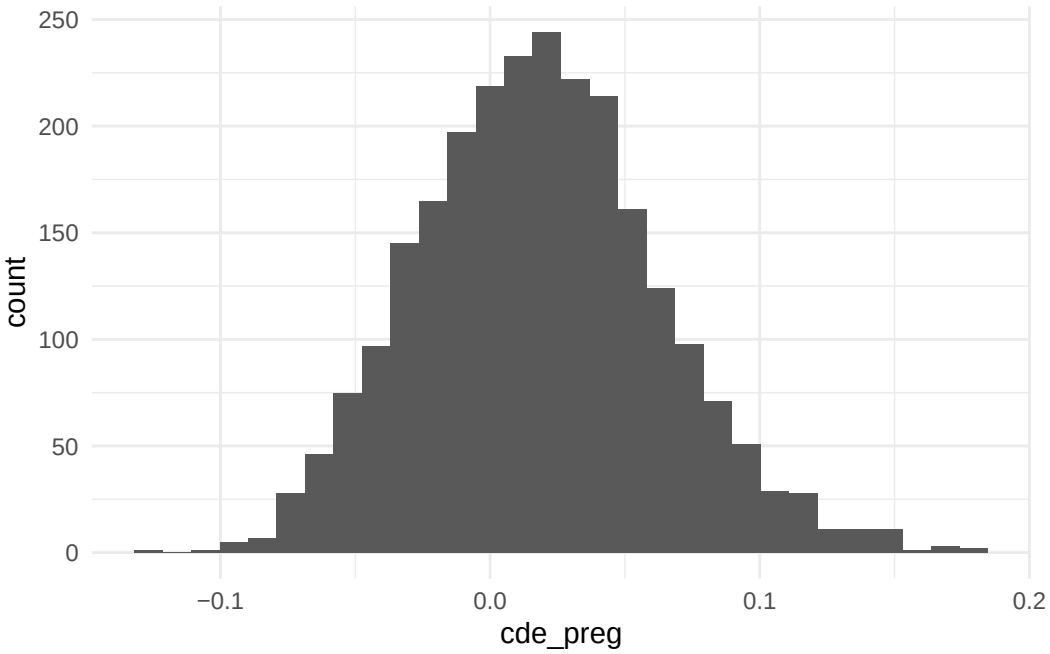
- Pregnancy CDE at sample mean:

```
[1] 0.01437995
```

- Birth weight CDE standard error:

```
[1] 0.04435521
```

- Birth weight CDE bootstrap sampling distribution. Slightly skewed, but okay to use:



- Birth weight CDE 95% confidence interval:

```
[1] "(0.1, -0.07)"
```

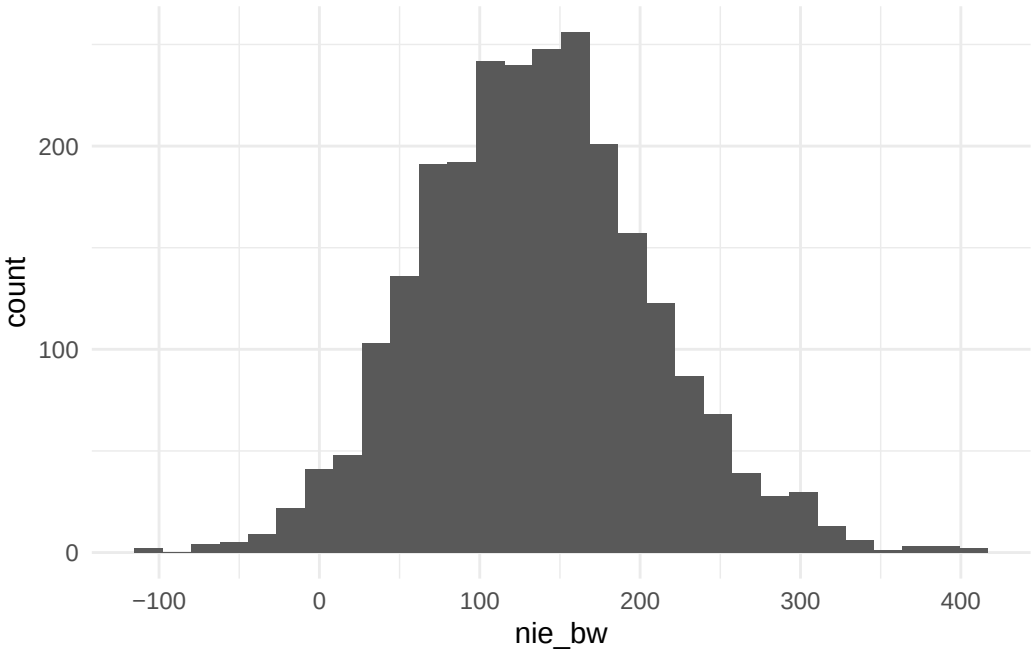
- Birth weight NIE:

```
[1] 93.8044
```

- Birth weight NIE standard error:

```
[1] 71.71617
```

- Bootstrap sampling distribution for the standard error estiamtion, which appears balanced. 95% normal confidence interval is appropriate to use.



- Birth weight NIE 95% normal confidence interval:

```
[1] "(234.37, -46.76)"
```

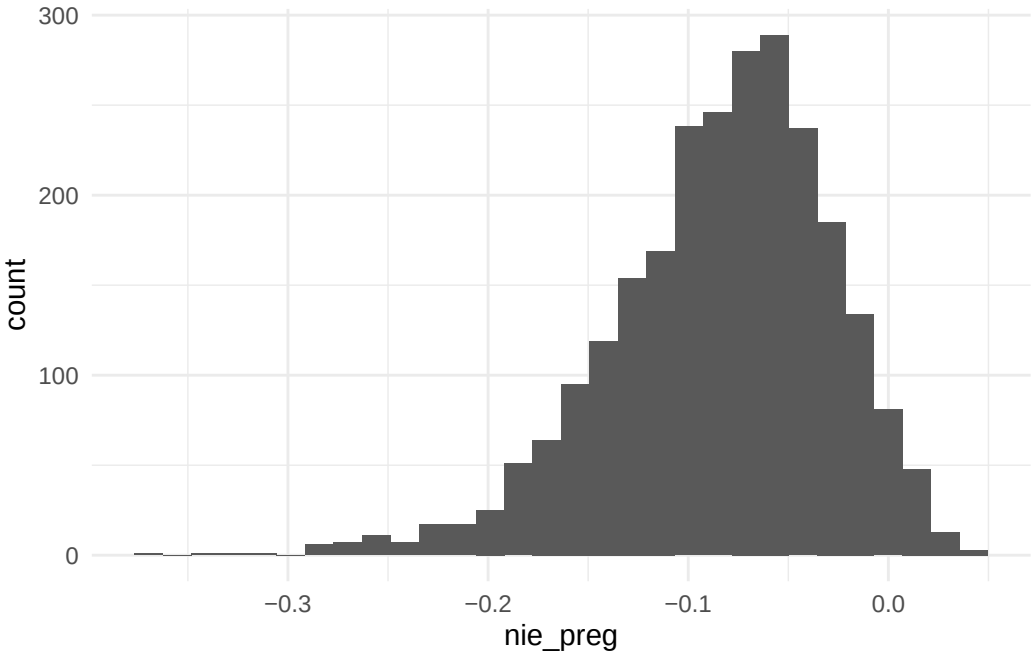
- Pregnancy NIE:

```
[1] -0.08412877
```

- Birth weight NIE standard error:

```
[1] 0.05574509
```

- Bootstrap sampling distribution for the standard error estiamtion, which appears skewed. It might make more sence to use 2.5^{th} and 97.5^{th} quantiles of this distribution to get confidence interval bounds.



- Pregnancy NIE 95% normal confidence interval:

```
[1] "(0.03, -0.19)"
```

Appendix

Functions

```
cde_estimator = function(DATA, MEDIATOR_LEVEL, MODEL){

    mean(predict(object = MODEL,
                 newdata = DATA %>% select(-Group, -mediator) %>%
                                    mutate(Group = "T",
                                           mediator = MEDIATOR_LEVEL),
                 type = "response")
         ) -

    mean(predict(object = MODEL,
                  newdata = DATA %>% select(-Group, -mediator) %>%
                                     mutate(Group = "C",
                                            mediator = MEDIATOR_LEVEL),
                  type = "response")
         )
}

cde_estimator_2 = function(MEDIATOR_LEVEL, MODEL){
  coef(MODEL)[which(names(coef(MODEL)) == "GroupT")]  +
    coef(MODEL)[which(names(coef(MODEL)) == "GroupT:mediator")] * MEDIATOR_LEVEL
}
```

```
nie_method_1 = function(DATA, OUTCOME_MODEL, MEDIATOR_MODEL){

  mediator_under_trt <-
    predict(object = MEDIATOR_MODEL,
            newdata = DATA %>% select(-Group, -mediator) %>%
                               mutate(Group = "T"),
            type = "response"
            )

  outcome_under_trt <-
    predict(object = OUTCOME_MODEL,
            newdata = DATA %>% select(-Group, -mediator) %>%
                               mutate(Group = "T",
                                      mediator = mediator_under_trt),
            type = "response"
```

```
            )


  mediator_under_no_trt <-
    predict(object = MEDIATOR_MODEL,
            newdata = DATA %>% select(-Group, -mediator) %>%
                               mutate(Group = "C"),
            type = "response"
            )

  outcome_under_no_trt <-
    predict(object = OUTCOME_MODEL,
            newdata = DATA %>% select(-Group, -mediator) %>%
                               mutate(Group = "T",
                                      mediator = mediator_under_no_trt),
            type = "response"
            )

  effect = mean(outcome_under_trt) - mean(outcome_under_no_trt)
  return(effect)
}
```

Bootstrap

```
K = 2500

resutls_cde <-
  data.frame(
    iter = integer(),
    order = integer(),

    cde_bw = numeric(),
    cde_preg = numeric()
  )

resutls_nie <-
  data.frame(
    iter = 1:K,

    nie_bw = rep(NA,K),
```

```
    nie_preg = rep(NA,K)
  )

for(i in 1:K){
  print(i)

  ### resample data, calculate values for other conditional effects
  boot_data <- data[sample(1:nrow(data), replace = T), ]

  boot_mediator_X_bar = mean(boot_data$mediator) ## actually for this data averages match
  boot_mediator_X_bar_se = sd(boot_data$mediator)/sqrt(nrow(boot_data))

  boot_potential_mediator_values = seq(from = boot_mediator_X_bar - 3 * boot_mediator_X_ba
                                       to = boot_mediator_X_bar + 3 * boot_mediator_X_bar_se,
                                       by = boot_mediator_X_bar_se)

  ### train models
  boot_borthweight_model_extended <-
    glm(
      Birthweight ~
      Group * (mediator + Race_ethnicity + Public.Asstce + Use.Tob +
                 N.prev.preg + Live.PTB + BL.GE + BL..BOP +
                 BL..PD.4 + BL..CAL.3),

      data = boot_data,
      family = "gaussian"
    )

  boot_pregnancy_model <-
    glm(
      `Preg.ended...37.wk` ~ Group * (mediator +
        Race_ethnicity + Public.Asstce + Use.Tob + N.prev.preg +
        Live.PTB + BL.GE + BL..BOP + BL..PD.4 + BL..CAL.3),

      data = boot_data,
      family = "binomial"
    )

  boot_mediator_model <-
    lm(
      mediator ~
```

```
        Group + Race_ethnicity + Public.Asstce + Use.Tob + N.prev.preg +
        Live.PTB + BL.GE + BL..BOP + BL..PD.4 + BL..CAL.3,

      data = boot_data
    )


  ### calculate effects

  ### bw cde
  sapply(boot_potential_mediator_values, function(X) cde_estimator(boot_data, X,
                                                                   boot_borthweight_model_

  ### pregnancy cde
  sapply(potential_mediator_values, function(X) cde_estimator(data, X, boot_pregnancy_mode

  ### save down cde results
  resutls_cde <-
    rbind(
      resutls_cde,

      data.frame(
        iter = i,
        order = seq(from = 1, to = length(boot_potential_mediator_values), by = 1),
        cde_bw = bw_cde_boot,
        cde_preg = preg_cde_boot
      )
    )
  ### bw nie
  nie_method_1(boot_data, boot_borthweight_model_extended, boot_mediator_model) -> nie_bw_

  ### preg nie
  nie_method_1(boot_data, boot_pregnancy_model, boot_mediator_model) -> nie_preg_boot

  ### save down cde results
  resutls_nie$nie_bw[i] <- nie_bw_boot
  resutls_nie$nie_preg[i] <- nie_preg_boot
}

beepr::beep(2)
```

```r
write.csv(resutls_cde, "cde_bootstrap_results.csv")
write.csv(resutls_nie, "nie_bootstrap_results.csv")

summary(resutls_cde$cde_bw)
hist(resutls_cde$cde_bw)

summary(resutls_cde$cde_preg)
hist(resutls_cde$cde_bw)

summary(resutls_nie$nie_bw)
hist(resutls_nie$nie_bw)

summary(resutls_nie$nie_preg)
hist(resutls_nie$nie_preg)
```

```r
mediation_model <- mediate(
  model.y = borthweight_model,
  model.m = mediator_model,
  treat = "Group",
  mediator = "mediator",
  robustSE = TRUE  # Optional, for robust standard errors
)

summary(mediation_model)
```