Project 1: Counterfactual Disparity Measures in a Simple Setting

Ashley I Naimi, PhD
Associate Professor
Director of Graduate Studies
Dept of Epidemiology
Emory University

ashley.naimi@emory.edu

EMORY | ROLLINS SCHOOL OF PUBLIC HEALTH

Overview

- Context: We'll start with a simple dataset
- Exposure x, mediator m, outcome y, exposure-outcome confounder c, and mediator-outcome confounder l
- These Slides: Three Phases - data management and exploration; analysis; interpretation
- A total of seven steps covering:
    - Setup Project Folder
    - Data Management and Exploration
    - Import, Transform, Explore Raw Data
    - Regression Analysis
    - CDM Analysis
    - Interpretation

Overview: The Data

```
a <- read_csv("./project1_data_raw.csv")

head(a)
```

```
## # A tibble: 6 × 5
##       c     x     l     m     y
##   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 1.19      0     0     1  122.
## 2 0.733     0     0     0  120.
## 3 1.59      0     0     1  122.
## 4 0.406     1     0     0  120.
## 5 0.273     0     0     0  118.
## 6 0.572     0     0     0  119.
```

Phase 1: Data Management and Exploration

Step 1: Construct a Project Folder

- Create a folder called `Disparities_Project1`
- Create Subfolders: `data`, `code`, `figures`, `misc`, `sandbox`, and `report`
- Move the `project1_data_raw.csv` into the `data` subfolder
- Create an RStudio Project in the `Disparities_Project1` folder

Step 2: A Data Management and Exploration File

- In the `code` subfolder, create two code files:
  - `data_man.R`
  - `main_analysis.R`
- Open the `data_man.R` file and include the **preamble** at the top:

```
packages <- c("data.table","tidyverse","skimr","here", "lmtest", "sandwich")

for (package in packages) {
  if (!require(package, character.only=T, quietly=T)) {
    install.packages(package, repos='http://lib.stat.cmu.edu/R/CRAN')
  }
}

for (package in packages) {
  library(package, character.only=T)
}
```

Step 3: Import, Explore, and Transform Data

- Import the `project1_data_raw.csv` into R using the `read_csv()` function
- You will need to use the `here()` function
    - `a <- read_csv(here("data", "project1_data_raw.csv"))`
- Explore the data:
    - use the `head()` and/or `tail()` functions
    - use the `skim()` function from the `skimr` package
    - use `ggplot()` to generate some figures
- Transform the data:
    - use the `mutate()` function to log-transform the confounding variable `c`.
    - in this case, re-write the original `c` as a log transformed version `log(c)`.
    - verify that the transformation worked

Step 4: Export the Data

- Create a new data file called `project1_data_analysis.csv` with the transformed `c`
- You should use the `write_csv()` function and the `here()` function to export to the `data` subfolder

Phase 2: Analysis

Step 1: Set Up and Import

- Open the `main_analysis.R` file and include the **preamble** at the top. Add the `lmtest` and `sandwich` packages to the list.
- Import the `project1_data_analysis.csv` data, and explore. For example:

```
head(a)
tail(a)

GGally::ggpairs(a[,c("c","y")])
```

Step 2: Basic Regression

- Fit an outcome model regressing y against all other variables in the data
- Fit a propensity score model for the exposure x adjusting for c
  - Create a PS overlap plot for the exposure and save the figure using `ggsave()`
  - Construct IP weights for the exposure and look at the distribution using `summary()`
- Fit a propensity score model for the mediator m adjusting for x, l, and c
  - Create a PS overlap plot for the mediator and save the figure using `ggsave()`
  - Construct IP weights for the exposure and look at the distribution using `summary()`
- Interpret these models, figures, and summaries

Step 3: Exposure - Outcome Association

a. Fit a linear regression model for the unadjusted association between y and x. Use the `lm` function.

b. Fit a conditionally adjusted regression model for the association between y and x adjusting for c using the `lm` function.

c. Fit a marginally adjusted regression model (g computation) for the association between y and x adjusted for c. Use the bootstrap to obtain standard errors.

d. Fit an inverse probability weighted regression model to estimate the association between y and c adjusted for c. Use the robust (sandwich) variance estimator to obtain standard errors.

Step 3a: Exposure - Outcome Association

a. Fit a linear regression model for the unadjusted association between y and x. Use the `lm` function. For example:

```
# crude
res_tabl1a <- summary(lm(y ~ x, data = a))$coefficients[2, 1:2]
```

Step 3b: Exposure - Outcome Association

b. Fit a conditionally adjusted regression model for the association between y and x adjusting for c using the `lm` function. For example:

```
# conditionally adjusted
res_tabl1b <- summary(lm(y ~ x + c, data = a))$coefficients[2, 1:2]
```

Step 3c: Exposure - Outcome Association

c. Fit a marginally adjusted regression model (g computation) for the association between y and x adjusted for c. Use the bootstrap to obtain standard errors. For example:

```
# marginally standardized
mod <- lm(y ~ x + c, data = a)

mu1 <- predict(mod, newdata=transform(a,x=1), type = "response")
mu0 <- predict(mod, newdata=transform(a,x=0), type = "response")
theta <- mean(mu1) - mean(mu0)

boot_res <- NULL
for(i in 1:500){
  set.seed(i)
  index <- sample(1:nrow(a), nrow(a), replace = T)
  boot_dat <- a[index,]

  mod_ <- lm(y ~ x + c, data = boot_dat)
  mu1_ <- predict(mod_, newdata=transform(boot_dat,x=1), type = "response")
  mu0_ <- predict(mod_, newdata=transform(boot_dat,x=0), type = "response")
  boot_res <- rbind(boot_res,
                    mean(mu1_) - mean(mu0_))

}

res_tabl2 <- c(theta, sd(boot_res))
```

Step 3d: Exposure - Outcome Association

d. Fit an inverse probability weighted regression model to estimate the association between y and c adjusted for c. Use the robust (sandwich) variance estimator to obtain standard errors. For example:

```
# ps model for the exposure
ps_model <- glm(x ~ c, data = a, family = binomial("logit"))
summary(ps_model)

# construct exposure weights

a <- a %>%  mutate(sw_x = (mean(x)/ps_model$fitted.values)*x +
                     ((1 - mean(x))/(1 - ps_model$fitted.values))*(1 - x))

summary(a$sw_x)

# ip weighting

mod_ipw <- lm(y ~ x, data = a, weights=sw_x)

res_tabl3 <- coeftest(mod_ipw,
                      vcov = vcovHC(mod_ipw, type = "HC3"))[2,1:2]
```

Step 4: Counterfactual Disparity Measures

a. Estimate the association that would remain if we set the mediator to `m = 0` using the structural transformation method.

b. Estimate the same CDM in Step 4a using inverse probability weighting.

For both steps, you'll need to determine and evaluate the referent level for the mediator:

```
# determine what the referent level for the mediator is:
a %>%
  group_by(m) %>%
  count()
```

Step 4a: Counterfactual Disparity Measures

a. Using the structural transformation method.

```
## using structural transformation
# start with a regression for estimating effect of mediator on outcome:
struct_trans1 <- lm(y ~ x + l + m + c + x*m, data = a)
st_coefs <- summary(struct_trans1)$coefficients[c("m","x:m"), 1]
# create transformed outcome
a <- a %>% mutate(y_tilde = y - m*st_coefs[1] - x*m*st_coefs[2])
# estimate CDM
cdm_est <- summary(lm(y_tilde ~ x + c, data=a))$coefficients[2,1]

# bootstrap
boot_res_cdm <- NULL
for(i in 1:500){
  set.seed(i)
  index <- sample(1:nrow(a), nrow(a), replace = T)
  boot_dat <- a[index,]

  mod_ <- lm(y_tilde ~ x + c, data = boot_dat)
  cdm_est_ <- summary(mod_)$coefficients[2,1]
  boot_res_cdm <- rbind(boot_res_cdm, cdm_est_)
}

res_tabl4 <- c(cdm_est,sd(boot_res_cdm))
```

Step 4b: Counterfactual Disparity Measures

a. Using IP weighting.

```
## propensity score model exposure
ps_model <- glm(x ~ c, data = a, family = binomial("logit"))
# construct exposure weights
a <- a %>%  mutate(sw_x = (mean(x)/ps_model$fitted.values)*x +
                     ((1 - mean(x))/(1 - ps_model$fitted.values))*(1 - x))
summary(a$sw_x)

## propensity score model mediator
ps_model_m <- glm(m ~ l + x + c, data = a, family = binomial("logit"))
# construct mediator weights
a <- a %>%  mutate(sw_m = (mean(m)/ps_model_m$fitted.values)*m +
                     ((1 - mean(m))/(1 - ps_model_m$fitted.values))*(1 - m))
summary(a$sw_m)

ipw_model <- lm(y ~ x + m + x*m, data = a, weights = sw_x*sw_m)
res_tabl5 <- coeftest(ipw_model,
                      vcov = vcovHC(ipw_model, type = "HC3"))[2,1:2]
```

Step 5: Combine All Results

```
## pulling the results together
res_tabl <- data.frame(
  rbind(res_tabl1a,
        res_tabl1b,
        res_tabl2,
        res_tabl3,
        res_tabl4,
        res_tabl5)
)

row.names(res_tabl) <- c("ATE: Crude",
                         "ATE: Conditionally Adjusted",
                         "ATE: Marginally Standardized",
                         "ATE: IP Weighted",
                         "CDM: Structural Transformation",
                         "CDM: IP Weighted")
res_tabl <- res_tabl %>%
  rownames_to_column(var = "Method")

res_tabl <- res_tabl %>%
  mutate(LCL = Estimate - 1.96*Std..Error,
         UCL = Estimate + 1.96*Std..Error)

## export results to spreadsheet
write_csv(res_tabl, here("misc","project1_results.csv"))
```

Step 5: Combine the Results

	Estimate	Std Error	LCL	UCL
ATE: Crude	3.03	0.14	2.76	3.31
ATE: Conditionally Adjusted	2.50	0.10	2.31	2.69
ATE: Marginally Standardized	2.50	0.11	2.27	2.73
ATE: IP Weighted	2.53	0.16	2.22	2.84
CDM: Structural Transformation	2.13	0.12	1.93	2.34
CDM: IP Weighted	2.16	0.17	1.81	2.50

Phase 3: Interpretation

Step 6: Why are the CDMs different from the ATEs?

- ATEs in the Table above represent a **difference in two means**:
  - Mean of y if $x = 1$
  - Mean of y if $x = 0$
- CDMs in the Table above represent a **difference in two means**:
  - Mean of y if $x = 1$ if m was set to zero
  - Mean of y if $x = 0$ if m was set to zero
- The CDM can be *smaller* than the ATE for two reasons:
  - The mean of y if $x = 1$ for the CDM is *lower* than for the ATE
  - The mean of y if $x = 0$ for the CDM is *higher* than for the ATE
- Let's explore this in our data

Step 6: Why are the CDMs different from the ATEs?

```
ate_mu1_ipw <- mean(predict(mod_ipw, newdata=transform(a,x=1), type="response"))
ate_mu0_ipw <- mean(predict(mod_ipw, newdata=transform(a,x=0), type="response"))

cdm_mu1_st <- mean(predict(cdm_model, newdata=transform(a,x=1), type="response"))
cdm_mu0_st <- mean(predict(cdm_model, newdata=transform(a,x=0), type="response"))

cdm_mu1_ipw <- mean(predict(ipw_model, newdata=transform(a,x=1), type="response"))
cdm_mu0_ipw <- mean(predict(ipw_model, newdata=transform(a,x=0), type="response"))

write_csv(
  tibble(
    Model = c("ATE: Marginally Standardized",
              "ATE: IP Weighted",
              "CDM: Structural Transformation",
              "CDM: IP Weighted"),
    mu1 = c(mean(mu1), ate_mu1_ipw, cdm_mu1_st, cdm_mu1_ipw),
    mu0 = c(mean(mu0), ate_mu0_ipw, cdm_mu0_st, cdm_mu0_ipw)
  ),
  file = here("misc","predicted_outcomes_cdm.csv")
)
```

Step 6: Why are the CDMs different from the ATEs?

	mu1	mu0
ATE: Marginally Standardized	122.95	120.45
ATE: IP Weighted	122.97	120.45
CDM: Structural Transformation	122.37	120.24
CDM: IP Weighted	122.64	120.50

- In both cases, `mu1` for the ATE goes down when translating to the CDM.
- For the structural transformation method, `mu0` goes down (but less than `mu1` goes down, leading to $CDM < ATE$)
- For the IP weighted method, `mu0` increases, leading to $CDM < ATE$

Step 7: Interpretation

After adjusting for confounding variables, the mean of y among individuals with $x = 1$ is 2.5 units (95% CI: 2.31, 2.69) higher than the mean of y among individuals with $x = 0$.

This mean difference would be reduced to 2.13 units (95% CI: 1.93, 2.34) if the mediator value m was set to zero for all individuals in the population.

These results suggest that M explains roughly $\frac{2.5-2.13}{2.5} \approx 15\%$ of the overall association between x and y.

Q&A

Project 1: Counterfactual Disparity Measures in a Simple Setting

Ashley I Naimi, PhD
Associate Professor
Emory University

ashley.naimi@emory.edu
ainaimi

EMORY | ROLLINS SCHOOL OF PUBLIC HEALTH