~~~
---
title: "CMA3"
output:
  pdf_document: default
  html_document: default
date: "2025-05-14"
---

```{r setup, include=FALSE}
library(haven)

setwd("/Users/josephepstein/Desktop/Causal Mediation Analysis")


list.files(pattern = "\\.dta$")


jobs2 <- read_dta("jobs2.dta")


View(jobs2)
```




```{r cars}
library(dplyr)
library(ggplot2)
library(broom)



# Compute the 80th percentile of the self-efficacy measure 'job_seek'
percentile_80 <- quantile(jobs2$job_seek, 0.80, na.rm = TRUE)

# Create a new variable for the mediator (job search self-efficacy) set at its 80th
percentile
jobs2$job_seek_80 <- ifelse(jobs2$job_seek >= percentile_80, jobs2$job_seek,
percentile_80)

# Linear regression with outcome 'work1', treatment 'treat', mediator 'job_seek_80', and
covariates
linear_model <- lm(work1 ~ treat * job_seek_80 + econ_hard + sex + age + nonwhite + educ +
income, data = jobs2)


summary(linear_model)

# Logistic regression with outcome 'work1' (logit model)
logit_model <- glm(work1 ~ treat * job_seek_80 + econ_hard + sex + age + nonwhite + educ +
income,
                   data = jobs2, family = binomial(link = "logit"))


summary(logit_model)

# Extract point estimates for the controlled direct effect (CDE) from both models

# For the linear model, the controlled direct effect is the coefficient of the interaction
term 'treat:job_seek_80'
linear_cde <- tidy(linear_model) %>%
  filter(term == "treat:job_seek_80") %>%
  select(estimate)
~~~

~~~
# For the logit model, the controlled direct effect is also the coefficient of the
interaction term 'treat:job_seek_80'
logit_cde <- tidy(logit_model) %>%
  filter(term == "treat:job_seek_80") %>%
  select(estimate)


cat("Controlled Direct Effect (Linear Model): ", linear_cde$estimate, "\n")
cat("Controlled Direct Effect (Logit Model): ", logit_cde$estimate, "\n")

```



```{r}
library(haven)
library(dplyr)
library(boot)
library(broom)

# Set working directory
setwd("/Users/josephepstein/Desktop/Causal Mediation Analysis")


jobs2 <- read_dta("jobs2.dta")

# Compute the 80th percentile of the self-efficacy measure 'job_seek'
percentile_80 <- quantile(jobs2$job_seek, 0.80, na.rm = TRUE)

# Create a new variable for the mediator (job search self-efficacy) set at its 80th
percentile
jobs2$job_seek_80 <- ifelse(jobs2$job_seek >= percentile_80, jobs2$job_seek,
percentile_80)

# Define the function to fit the logit model and extract CDE
bootstrap_function <- function(data, indices) {
  # Resample the data
  resampled_data <- data[indices, ]

  # Fit logit model
  logit_model <- glm(work1 ~ treat * job_seek_80 + econ_hard + sex + age + nonwhite + educ
+ income,
                     data = resampled_data, family = binomial(link = "logit"))

  # Extract the coefficient for the interaction term 'treat:job_seek_80' / CDE
  coef_interaction <- coef(logit_model)["treat:job_seek_80"]

  return(coef_interaction)
}

# Perform percentile bootstrap with 1000 replications
set.seed(123)  # For reproducibility
bootstrap_results <- boot(data = jobs2, statistic = bootstrap_function, R = 1000)

# Compute the 90% CI for the CDE
bootstrap_ci <- quantile(bootstrap_results$t, c(0.05, 0.95))

# CI
cat("90% Confidence Interval for the Controlled Direct Effect (CDE): ", bootstrap_ci,
"\n")

# Conduct hypothesis test (alpha = 0.1) for H0: CDE = 0 by computing p-value
bootstrap_test_stat <- bootstrap_results$t
p_value <- mean(bootstrap_test_stat >= 0)  # One-sided test for positive CDE
~~~

~~~
p_value_two_sided <- 2 * min(p_value, 1 - p_value)  # Two-sided test


cat("p-value for the hypothesis test H0: CDE = 0 (alpha = 0.1): ", p_value_two_sided,
"\n")


bootstrap_mean <- mean(bootstrap_results$t)
cat("Mean CDE from bootstrap samples: ", bootstrap_mean, "\n")

```
~~~

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

~~~
```{r}

setwd("/Users/josephepstein/Downloads")


library(haven)


plowUse_1_ <- read_dta("plowUse (1).dta")


View(plowUse_1_)
```

```{r}

library(lmtest)  # For robust standard errors
str(plowUse_1_)

# Run the first regression to estimate residuals for the mediator (ln_income)
mediator_model <- lm(ln_income ~ plow + agricultural_suitability + tropical_climate +
                     large_animals + rugged + polity2_2000, data = plowUse_1_,
                     na.action = na.exclude)

# residuals of mediator
plowUse_1_$resid_mediator <- residuals(mediator_model)

# Run the second regression to estimate the residuals for women in politics-outcome
outcome_model <- lm(women_politics ~ plow + agricultural_suitability + tropical_climate +
                    large_animals + rugged + polity2_2000, data = plowUse_1_,
                    na.action = na.exclude)

# residuals of outcome
plowUse_1_$resid_outcome <- residuals(outcome_model)

# indirect effect estimate: Regression of residualized outcome on residualized mediator
indirect_model <- lm(resid_outcome ~ resid_mediator + plow + resid_mediator:plow, data =
plowUse_1_)

# direct effect estimate: Regression of residualized outcome on exposure (plow)
direct_model <- lm(resid_outcome ~ plow + resid_mediator + resid_mediator:plow, data =
plowUse_1_)


summary(indirect_model)
~~~

~~~
summary(direct_model)
```
```{r}
library(haven)
library(lmtest)
library(boot)


setwd("/Users/josephepstein/Downloads")  # Adjust the path to your dataset
plowUse_1_ <- read_dta("plowUse (1).dta")


str(plowUse_1_)

#  first regression to estimate residuals for the mediator (ln_income)
mediator_model <- lm(ln_income ~ plow + agricultural_suitability + tropical_climate +
                     large_animals + rugged + polity2_2000, data = plowUse_1_,
                     na.action = na.exclude)

# Get residuals of mediator
plowUse_1_$resid_mediator <- residuals(mediator_model)

# second regression to estimate residuals for the outcome (women in politics)
outcome_model <- lm(women_politics ~ plow + agricultural_suitability + tropical_climate +
                    large_animals + rugged + polity2_2000, data = plowUse_1_,
                    na.action = na.exclude)

# residuals of the outcome
plowUse_1_$resid_outcome <- residuals(outcome_model)

# Define a function to compute the indirect and direct effects for a bootstrap sample
compute_effects <- function(data, indices) {
  # Resample the data
  resampled_data <- data[indices, ]

  # Re-run the regression for the mediator and outcome
  mediator_model_boot <- lm(ln_income ~ plow + agricultural_suitability + tropical_climate
+
                            large_animals + rugged + polity2_2000, data = resampled_data,
                            na.action = na.exclude)
  resampled_data$resid_mediator <- residuals(mediator_model_boot)

  outcome_model_boot <- lm(women_politics ~ plow + agricultural_suitability +
tropical_climate +
                           large_animals + rugged + polity2_2000, data = resampled_data,
                           na.action = na.exclude)
  resampled_data$resid_outcome <- residuals(outcome_model_boot)

  # Estimate the indirect effect: Regression of residualized outcome on residualized
mediator
  indirect_model_boot <- lm(resid_outcome ~ resid_mediator + plow + resid_mediator:plow,
data = resampled_data)
  indirect_effect <- coef(indirect_model_boot)["resid_mediator"]

  # Estimate the direct effect: Regression of residualized outcome on exposure (plow)
  direct_model_boot <- lm(resid_outcome ~ plow + resid_mediator + resid_mediator:plow,
data = resampled_data)
  direct_effect <- coef(direct_model_boot)["plow"]

  return(c(direct_effect, indirect_effect))
}

# Perform the bootstrap with 1000 replications
set.seed(123)  # For reproducibility
~~~

~~~
bootstrap_results <- boot(data = plowUse_1_, statistic = compute_effects, R = 1000)

# View the bootstrap results
bootstrap_results

# Calculate 95% confidence intervals for the direct and indirect effects
direct_ci <- quantile(bootstrap_results$t[, 1], probs = c(0.025, 0.975))
indirect_ci <- quantile(bootstrap_results$t[, 2], probs = c(0.025, 0.975))

# CI
cat("95% Confidence Interval for Direct Effect:", direct_ci, "\n")
cat("95% Confidence Interval for Indirect Effect:", indirect_ci, "\n")

# Calculate the p-value for the test of the null hypothesis that the effects are zero
# null hypothesis: Direct effect = 0
p_value_direct <- mean(bootstrap_results$t[, 1] >= 0 | bootstrap_results$t[, 1] <= 0)

# null hypothesis: Indirect effect = 0
p_value_indirect <- mean(bootstrap_results$t[, 2] >= 0 | bootstrap_results$t[, 2] <= 0)


cat("p-value for Direct Effect = 0:", p_value_direct, "\n")
cat("p-value for Indirect Effect = 0:", p_value_indirect, "\n")

```
~~~