````
---
title: "CMA2"
output: html_document
date: "2025-05-01"
---

```{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}
#4.1) Treatment assignment balances the distribution of baseline covariates.
library(dplyr)
library(tidyr)
library(broom)


jobs2 <- read_dta("jobs2.dta")


baseline_vars <- c("econ_hard", "sex", "age", "nonwhite", "educ", "income")
treatment_var <- "treat"

#  t-test
run_t_test <- function(var) {
  t.test(jobs2[[var]] ~ jobs2[[treatment_var]]) %>%
    tidy() %>%
    mutate(variable = var) %>%
    select(variable, estimate1, estimate2, p.value)
}

#  chi-squared test (for categorical)
run_chisq_test <- function(var) {
  tbl <- table(jobs2[[var]], jobs2[[treatment_var]])
  chisq.test(tbl) %>%
    tidy() %>%
    mutate(variable = var) %>%
    select(variable, p.value)
}

# Run t-tests on numeric vars
numeric_vars <- c("econ_hard", "age", "educ", "income")
t_results <- bind_rows(lapply(numeric_vars, run_t_test))

# Run chi-sq tests on categorical vars
categorical_vars <- c("sex", "nonwhite")
c_results <- bind_rows(lapply(categorical_vars, run_chisq_test))

# Merge
balance_results <- full_join(t_results, c_results, by = "variable")
````
````

```
print(balance_results)

```
```{r}
# 4.2) Experimental estimate of the total effect of treatment on the probability of
reemployment


# Mean reemployment by treatment group
reemployment_means <- jobs2 %>%
  group_by(treat) %>%
  summarize(
    mean_work1 = mean(work1, na.rm = TRUE),
    n = n()
  )

print(reemployment_means)

# Difference in means via t-test
treatment_effect <- t.test(work1 ~ treat, data = jobs2)

# Tidy output with estimate and CI
treatment_effect_result <- tidy(treatment_effect)
print(treatment_effect_result)

# Alternative estimate via linear regression
model <- lm(work1 ~ treat, data = jobs2)
summary(model)
```
```{r}
#4.3) Estimate of the total effect of treatment on job search self-efficacy

reemployment_means <- jobs2 %>%
  group_by(treat) %>%
  summarize(
    mean_work1 = mean(work1, na.rm = TRUE),
    n = n()
  )

print(reemployment_means)

# Difference in means via t-test
treatment_effect <- t.test(work1 ~ treat, data = jobs2)

# Tidy output with estimate and CI
treatment_effect_result <- tidy(treatment_effect)
print(treatment_effect_result)

# Alternatively, estimate via linear regression
model <- lm(work1 ~ treat, data = jobs2)
summary(model)
```

```{r}
# 5.1: ATE, NDE, NIE, treat -> employ, mediated by Jobseek
library(mediation)

# Mediator model: job search self-efficacy ~ treatment
model_m <- lm(job_seek ~ treat, data = jobs2)

# Outcome model: employment ~ treatment + mediator
model_y <- lm(work1 ~ treat + job_seek, data = jobs2)
```

```
# Run mediation analysis
mediation_result <- mediate(
  model.m = model_m,
  model.y = model_y,
  treat = "treat",
  mediator = "job_seek",
  boot = TRUE,
  sims = 1000
)



print_mediation_summary <- function(med_obj) {
  cat("Causal Mediation Analysis Results\n")
  cat("---------------------------------\n")

  # Extract estimates and CIs
  ATE <- med_obj$total.effect
  ATE_ci <- med_obj$total.ci
  ATE_p <- med_obj$total.p

  NDE <- med_obj$d0
  NDE_ci <- med_obj$d0.ci
  NDE_p <- med_obj$d0.p

  NIE <- med_obj$z0
  NIE_ci <- med_obj$z0.ci
  NIE_p <- med_obj$z0.p

  prop_med <- med_obj$n0
  prop_med_ci <- med_obj$n0.ci
  prop_med_p <- med_obj$n0.p

  # Print formatted summary
  cat(sprintf("Average Treatment Effect (ATE):    %.6f (95%% CI: %.6f, %.6f), p = %.3f\n",
              ATE, ATE_ci[1], ATE_ci[2], ATE_p))

  cat(sprintf("Natural Direct Effect (NDE):       %.6f (95%% CI: %.6f, %.6f), p = %.3f\n",
              NDE, NDE_ci[1], NDE_ci[2], NDE_p))

  cat(sprintf("Natural Indirect Effect (NIE):     %.6f (95%% CI: %.6f, %.6f), p = %.3f\n",
              NIE, NIE_ci[1], NIE_ci[2], NIE_p))

  cat(sprintf("Proportion Mediated:               %.6f (95%% CI: %.6f, %.6f), p = %.3f\n",
              prop_med, prop_med_ci[1], prop_med_ci[2], prop_med_p))

  cat("\nMethod: Nonparametric Bootstrap (", med_obj$sims, " simulations)\n", sep = "")
  cat("Sample Size: ", med_obj$n, "\n", sep = "")
}


print_mediation_summary(mediation_result)


```




```{r}


# Ensure treat is coded as 0 (control) and 1 (treatment)
```

```
jobs2$treat <- as.numeric(jobs2$treat)
if (!all(jobs2$treat %in% c(0, 1))) {
  jobs2$treat <- ifelse(jobs2$treat == min(jobs2$treat), 0, 1)
}

# Filter complete cases
model_data <- jobs2 %>%
  dplyr::select(treat, job_seek, work1, econ_hard, sex, age, nonwhite, educ, income) %>%
  filter(complete.cases(.))

# Fit mediator and outcome models
model_m <- lm(job_seek ~ treat + econ_hard + sex + age + nonwhite + educ + income, data =
model_data)
model_y <- lm(work1 ~ treat + job_seek + econ_hard + sex + age + nonwhite + educ + income,
data = model_data)

# Run causal mediation analysis
med <- mediate(model.m = model_m, model.y = model_y, treat = "treat", mediator =
"job_seek", boot = TRUE, sims = 1000)


summary(med)


```
```{r}


# Ensure treat is 0/1
jobs2$treat <- as.numeric(jobs2$treat)
if (!all(jobs2$treat %in% c(0, 1))) {
  jobs2$treat <- ifelse(jobs2$treat == min(jobs2$treat), 0, 1)
}

# Filter complete cases
model_data <- jobs2 %>%
  dplyr::select(treat, job_seek, work1, econ_hard, sex, age, nonwhite, educ, income) %>%
  filter(complete.cases(.))

# Mediator model
model_m <- lm(job_seek ~ treat + econ_hard + sex + age + nonwhite + educ + income, data =
model_data)

# Outcome model with interaction term
model_y <- lm(work1 ~ treat * job_seek + econ_hard + sex + age + nonwhite + educ + income,
data = model_data)

# Mediation analysis (no INT argument needed)
med <- mediate(
  model.m = model_m,
  model.y = model_y,
  treat = "treat",
  mediator = "job_seek",
  boot = TRUE,
  sims = 1000
)

# Print result
summary(med)

```

```{r}
# Ensure treat is 0/1
```

```
jobs2$treat <- as.numeric(jobs2$treat)
if (!all(jobs2$treat %in% c(0, 1))) {
  jobs2$treat <- ifelse(jobs2$treat == min(jobs2$treat), 0, 1)
}

# Filter data
model_data <- jobs2 %>%
  dplyr::select(treat, job_seek, work1, econ_hard, sex, age, nonwhite, educ, income) %>%
  filter(complete.cases(.))

# Linear regression for job search self-efficacy
lm_job_seek <- lm(job_seek ~ treat + econ_hard + sex + age + nonwhite + educ + income,
data = model_data)
summary(lm_job_seek)

# Logistic regression for employment
logit_work1 <- glm(work1 ~ treat + job_seek + econ_hard + sex + age + nonwhite + educ +
income,
                   data = model_data, family = binomial(link = "logit"))
summary(logit_work1)

```
```{r}

# Ensure treat is binary
jobs2$treat <- as.numeric(jobs2$treat)
if (!all(jobs2$treat %in% c(0, 1))) {
  jobs2$treat <- ifelse(jobs2$treat == min(jobs2$treat), 0, 1)
}

# Prepare complete-case data
model_data <- jobs2 %>%
  dplyr::select(treat, job_seek, work1, econ_hard, sex, age, nonwhite, educ, income) %>%
  filter(complete.cases(.))

# Linear model for mediator
model_m <- lm(job_seek ~ treat + econ_hard + sex + age + nonwhite + educ + income, data =
model_data)

# Logistic model for outcome
model_y <- glm(work1 ~ treat + job_seek + econ_hard + sex + age + nonwhite + educ +
income,
               data = model_data, family = binomial(link = "logit"))

#Mediation analysis with 2000 simulations
med <- mediate(
  model.m = model_m,
  model.y = model_y,
  treat = "treat",
  mediator = "job_seek",
  sims = 2000,
  boot = TRUE
)

# === Step 4: Print results
summary(med)

```
```{r}


# Ensure treat is binary
jobs2$treat <- as.numeric(jobs2$treat)
if (!all(jobs2$treat %in% c(0, 1))) {
```

```
  jobs2$treat <- ifelse(jobs2$treat == min(jobs2$treat), 0, 1)
}

# Filter complete cases for variables
model_data <- jobs2 %>%
  dplyr::select(treat, job_seek, work1, econ_hard, sex, age, nonwhite, educ, income) %>%
  filter(complete.cases(.))

# Linear mediator model
model_m <- lm(job_seek ~ treat + econ_hard + sex + age + nonwhite + educ + income, data =
model_data)

# Logistic outcome model WITH treatment × mediator interaction
model_y <- glm(
  work1 ~ treat * job_seek + econ_hard + sex + age + nonwhite + educ + income,
  family = binomial(link = "logit"),
  data = model_data
)

#  Mediation analysis with interaction, 2000 bootstraps
med_result <- mediate(
  model.m = model_m,
  model.y = model_y,
  treat = "treat",
  mediator = "job_seek",
  boot = TRUE,
  sims = 2000
)


summary(med_result)
```

```{r}



jobs2$treat <- as.numeric(jobs2$treat)
if (!all(jobs2$treat %in% c(0, 1))) {
  jobs2$treat <- ifelse(jobs2$treat == min(jobs2$treat), 0, 1)
}

# Complete case data
data_ipw <- jobs2 %>%
  dplyr::select(treat, job_seek, work1, econ_hard, sex, age, nonwhite, educ, income) %>%
  filter(complete.cases(.))

# Propensity score model 1 (main)
ps_model1 <- glm(treat ~ econ_hard + sex + age + nonwhite + educ + income,
                 data = data_ipw, family = binomial)

ps1 <- predict(ps_model1, type = "response")

# Propensity score model 2 (robust/alt spec)
ps_model2 <- glm(treat ~ econ_hard * age + sex + nonwhite + educ + income,
                 data = data_ipw, family = binomial)

ps2 <- predict(ps_model2, type = "response")

# Stabilized weights
p_treat <- mean(data_ipw$treat == 1)

# Model 1 weights
```

```
sw1 <- ifelse(data_ipw$treat == 1, p_treat / ps1, (1 - p_treat) / (1 - ps1))

# Model 2 weights
sw2 <- ifelse(data_ipw$treat == 1, p_treat / ps2, (1 - p_treat) / (1 - ps2))

# === Step 4: Censor weights (1st and 99th percentiles)
cap_weights <- function(w) {
  q <- quantile(w, probs = c(0.01, 0.99))
  w[w < q[1]] <- q[1]
  w[w > q[2]] <- q[2]
  return(w)
}

sw1_capped <- cap_weights(sw1)
sw2_capped <- cap_weights(sw2)

#  Weighted regression for total effect (model 1)
ate_model1 <- glm(work1 ~ treat, family = binomial, data = data_ipw, weights = sw1_capped)
summary(ate_model1)

#: Mediation using model 1 weights
# Fit mediator and outcome models (unweighted)
m_model <- lm(job_seek ~ treat + econ_hard + sex + age + nonwhite + educ + income, data =
data_ipw)
y_model <- glm(work1 ~ treat + job_seek + econ_hard + sex + age + nonwhite + educ +
income,
               data = data_ipw, family = binomial)

# Mediation analysis (not IPW-corrected, but using the same covariates)
med_ipw <- mediate(
  model.m = m_model,
  model.y = y_model,
  treat = "treat",
  mediator = "job_seek",
  boot = TRUE,
  sims = 1000
)


summary(med_ipw)

```
```{r}


# binary
jobs2$treat <- as.numeric(jobs2$treat)
if (!all(jobs2$treat %in% c(0, 1))) {
  jobs2$treat <- ifelse(jobs2$treat == min(jobs2$treat), 0, 1)
}

# Prepare complete-case data
model_data <- jobs2 %>%
  dplyr::select(treat, job_seek, work1, econ_hard, sex, age, nonwhite, educ, income) %>%
  filter(complete.cases(.))

# Linear mediator model
model_m <- lm(job_seek ~ treat + econ_hard + sex + age + nonwhite + educ + income, data =
model_data)

# Logistic outcome model with interaction
model_y <- glm(work1 ~ treat * job_seek + econ_hard + sex + age + nonwhite + educ +
income,
               data = model_data, family = binomial)
```

```
# Mediation analysis with 90% CI, 2000 bootstraps
med_final <- mediate(
  model.m = model_m,
  model.y = model_y,
  treat = "treat",
  mediator = "job_seek",
  boot = TRUE,
  sims = 2000,
  conf.level = 0.90
)

# Print 90% CI summary
summary(med_final)

```
```{r}
# Extract 2000 bootstrap simulations of ACME (NIE)
acme_vals <- med_final$d0.sims  # or med_final$z0.sims depending on method

# Two-sided p-value by inverting percentile method
p_val_nie <- 2 * min(
  mean(acme_vals <= 0),
  mean(acme_vals >= 0)
)

cat("Two-sided percentile bootstrap p-value for NIE:", round(p_val_nie, 4), "\n")

# Decision at α = 0.1
if (p_val_nie < 0.1) {
  cat("Reject H0: NIE ≠ 0 at α = 0.1 level.\n")
} else {
  cat("Fail to reject H0: insufficient evidence that NIE ≠ 0 at α = 0.1 level.\n")
}

```
```
```