chap10-ipw

yousri elfattah

2024-10-02

# Chapter 10

# Causal Inference with Observational Data

## Inverse Probability Weighting (IPW) and Propensity Score Matching (PSM)

1. **Fit the Propensity Score Model**: Use logistic regression to calculate propensity scores (the probability of treatment given covariates).
2. **Calculate Inverse Probability Weights**: Inverse the propensity scores to obtain weights.
3. **Fit the Weighted Outcome Model**: Use these weights in the outcome regression model to estimate the ATE.

### Loading the dataset:

# Load necessary library  
library(haven) # for read\_dta function  
  
# Load the dataset  
data <- haven::read\_dta("cattaneo2.dta")  
  
# Convert to standard data frame for full column display  
data\_df <- as.data.frame(data)  
  
# Inspect the first few rows and column names  
head(data\_df)

## bweight mmarried mhisp fhisp foreign alcohol deadkids mage medu fage fedu  
## 1 3459 1 0 0 0 0 0 24 14 28 16  
## 2 3260 0 0 0 1 0 0 20 10 0 0  
## 3 3572 1 0 0 1 0 0 22 9 30 9  
## 4 2948 1 0 0 0 0 0 26 12 30 12  
## 5 2410 1 0 0 0 0 0 20 12 21 14  
## 6 3147 0 0 0 0 0 0 27 12 40 12  
## nprenatal monthslb order msmoke mbsmoke mrace frace prenatal birthmonth  
## 1 10 30 2 0 0 1 1 1 12  
## 2 6 42 3 0 0 0 0 1 7  
## 3 10 17 3 0 0 1 1 1 3  
## 4 10 34 2 0 0 1 1 1 1  
## 5 12 0 1 0 0 1 1 1 3  
## 6 9 0 1 0 0 1 1 1 4  
## lbweight fbaby prenatal1  
## 1 0 0 1  
## 2 0 0 1  
## 3 0 0 1  
## 4 0 0 1  
## 5 1 1 1  
## 6 0 1 1

# Check the structure of the dataset to see data types  
# Use capture.output to capture the output of str(data)  
str\_output <- capture.output(str(data))  
writeLines(str\_output, "str\_data\_output.txt")  
  
# Print to console to verify  
cat("Output of str(data) saved to str\_data\_output.txt\n")

## Output of str(data) saved to str\_data\_output.txt

### Initialize the CausalModel

library(CausalModels)  
library(survey)

## Loading required package: grid

## Loading required package: Matrix

## Loading required package: survival

##   
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':  
##   
## dotchart

# Convert relevant columns to factors  
data$mbsmoke <- as.factor(data$mbsmoke)  
data$mage <- as.factor(data$mage)  
confounders <- c("mmarried", "prenatal1", "mage", "fbaby")  
  
# Initialize the parameters for CausalModels package  
CausalModels::init\_params(  
 outcome = "bweight",   
 treatment = "mbsmoke",   
 covariates = confounders,   
 data = data  
)

## Successfully initialized!  
##   
## Summary:  
##   
## Outcome - bweight   
## Treatment - mbsmoke   
## Covariates - [ mmarried, prenatal1, mage, fbaby ]   
##   
## Size - 4642 x 23   
##   
## Default formula for outcome models:   
## bweight ~ mbsmoke + mage + mmarried + (mbsmoke \* mmarried) + I(mmarried \* mmarried) + prenatal1 + (mbsmoke \* prenatal1) + I(prenatal1 \* prenatal1) + fbaby + (mbsmoke \* fbaby) + I(fbaby \* fbaby)   
##   
## Default formula for propensity models:   
## mbsmoke ~ mage + mmarried + I(mmarried \* mmarried) + prenatal1 + I(prenatal1 \* prenatal1) + fbaby + I(fbaby \* fbaby)

### Estimating Average Treatment Effect:

# Fit the propensity score model  
propensity\_model <- glm(  
 formula = mbsmoke ~ mage + mmarried + I(mmarried \* mmarried) + prenatal1 + I(prenatal1 \* prenatal1) + fbaby + I(fbaby \* fbaby),  
 family = binomial(link = "logit"),   
 data = data  
)  
  
# Calculate propensity scores  
propensity\_scores <- predict(propensity\_model, type = "response")  
  
# Compute inverse probability weights  
data$weights <- ifelse(data$mbsmoke == 1, 1 / propensity\_scores, 1 / (1 - propensity\_scores))  
  
# Fit the weighted outcome model using survey package  
outcome\_model <- svyglm(  
 formula = bweight ~ mbsmoke + mage + mmarried + (mbsmoke \* mmarried) + I(mmarried \* mmarried) + prenatal1 +   
 (mbsmoke \* prenatal1) + I(prenatal1 \* prenatal1) + fbaby + (mbsmoke \* fbaby) + I(fbaby \* fbaby),  
 design = svydesign(ids = ~1, weights = ~weights, data = data)  
)  
  
# Summary of the weighted outcome model  
summary(outcome\_model)

##   
## Call:  
## svyglm(formula = bweight ~ mbsmoke + mage + mmarried + (mbsmoke \*   
## mmarried) + I(mmarried \* mmarried) + prenatal1 + (mbsmoke \*   
## prenatal1) + I(prenatal1 \* prenatal1) + fbaby + (mbsmoke \*   
## fbaby) + I(fbaby \* fbaby), design = svydesign(ids = ~1, weights = ~weights,   
## data = data))  
##   
## Survey design:  
## svydesign(ids = ~1, weights = ~weights, data = data)  
##   
## Coefficients: (3 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2343.18 20.82 112.527 < 2e-16 \*\*\*  
## mbsmoke1 -217.74 49.55 -4.394 1.14e-05 \*\*\*  
## mage14 904.29 152.08 5.946 2.95e-09 \*\*\*  
## mage15 686.56 165.01 4.161 3.23e-05 \*\*\*  
## mage16 833.07 69.07 12.062 < 2e-16 \*\*\*  
## mage17 948.25 79.92 11.866 < 2e-16 \*\*\*  
## mage18 996.84 53.67 18.573 < 2e-16 \*\*\*  
## mage19 925.36 52.74 17.544 < 2e-16 \*\*\*  
## mage20 919.00 54.71 16.798 < 2e-16 \*\*\*  
## mage21 930.39 46.25 20.117 < 2e-16 \*\*\*  
## mage22 938.58 54.98 17.072 < 2e-16 \*\*\*  
## mage23 973.81 52.14 18.679 < 2e-16 \*\*\*  
## mage24 908.89 48.78 18.632 < 2e-16 \*\*\*  
## mage25 895.77 53.54 16.731 < 2e-16 \*\*\*  
## mage26 914.96 56.88 16.085 < 2e-16 \*\*\*  
## mage27 915.17 61.94 14.776 < 2e-16 \*\*\*  
## mage28 899.12 58.26 15.434 < 2e-16 \*\*\*  
## mage29 949.66 50.47 18.815 < 2e-16 \*\*\*  
## mage30 833.20 84.52 9.858 < 2e-16 \*\*\*  
## mage31 871.82 80.15 10.877 < 2e-16 \*\*\*  
## mage32 916.23 54.21 16.901 < 2e-16 \*\*\*  
## mage33 881.23 72.79 12.106 < 2e-16 \*\*\*  
## mage34 964.51 69.56 13.865 < 2e-16 \*\*\*  
## mage35 922.58 79.75 11.569 < 2e-16 \*\*\*  
## mage36 717.04 159.46 4.497 7.07e-06 \*\*\*  
## mage37 1144.69 153.04 7.479 8.89e-14 \*\*\*  
## mage38 672.19 204.33 3.290 0.001011 \*\*   
## mage39 992.37 150.82 6.580 5.23e-11 \*\*\*  
## mage40 1118.68 159.76 7.002 2.88e-12 \*\*\*  
## mage41 1018.41 86.48 11.776 < 2e-16 \*\*\*  
## mage42 626.44 167.67 3.736 0.000189 \*\*\*  
## mage43 1092.13 149.02 7.329 2.73e-13 \*\*\*  
## mage44 889.83 121.88 7.301 3.35e-13 \*\*\*  
## mage45 1043.81 40.79 25.590 < 2e-16 \*\*\*  
## mmarried 178.69 28.80 6.204 6.00e-10 \*\*\*  
## prenatal1 69.64 28.52 2.442 0.014650 \*   
## fbaby -75.18 20.82 -3.610 0.000309 \*\*\*  
## mbsmoke1:mmarried -66.70 48.21 -1.383 0.166630   
## mbsmoke1:prenatal1 -34.38 48.36 -0.711 0.477194   
## mbsmoke1:fbaby 129.47 50.33 2.572 0.010135 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 313542.9)  
##   
## Number of Fisher Scoring iterations: 2

# Extract ATE  
ate\_ipw <- coef(outcome\_model)["mbsmoke1"]  
se\_ipw <- sqrt(diag(vcov(outcome\_model)))["mbsmoke1"]  
  
# Calculate Confidence Intervals  
ci\_low <- ate\_ipw - 1.96 \* se\_ipw  
ci\_high <- ate\_ipw + 1.96 \* se\_ipw  
  
# Print Average Treatment Effect (ATE)  
cat("Estimate - ", ate\_ipw, "\n")

## Estimate - -217.7389

cat("SE - ", se\_ipw, "\n")

## SE - 49.5498

cat("95% CI - (", ci\_low, ", ", ci\_high, ")\n")

## 95% CI - ( -314.8565 , -120.6213 )

## Balance Plots

Creating balance plots to compare the distribution of covariates between treated and control groups, both before and after matching or weighting, is essential to check the quality of propensity score matching or weighting. This can be done using the MatchIt package in R for propensity score matching and the cobalt package for balance plots.

### Steps to Create Balance Plots:

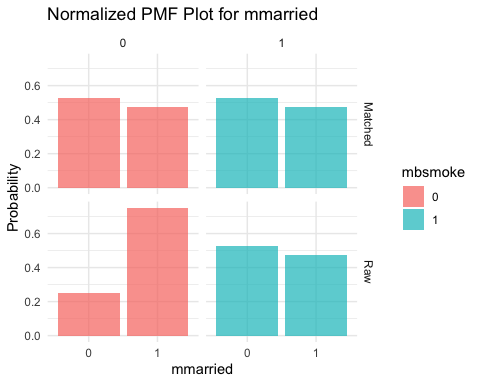
1. **Fit the Propensity Score Model**: Compute the propensity scores.
2. **Apply Matching/Weighting**: Use MatchIt to perform matching.
3. **Balance Plots**: Use cobalt to create balance box plots for covariates.

# Load necessary libraries  
library(MatchIt)  
library(ggplot2)  
library(tidyverse)

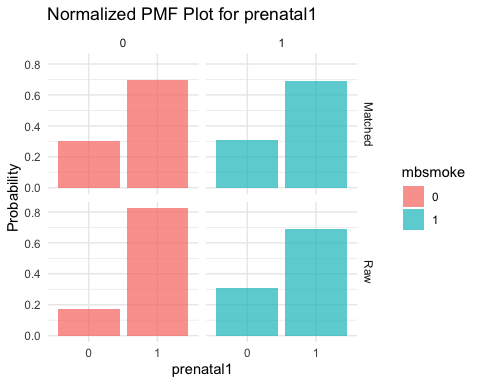
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.1  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ tidyr::expand() masks Matrix::expand()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ tidyr::pack() masks Matrix::pack()  
## ✖ tidyr::unpack() masks Matrix::unpack()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Assuming 'data' is your dataset  
data$mbsmoke <- as.factor(data$mbsmoke)  
  
# Perform propensity score matching  
psm <- matchit(mbsmoke ~ mmarried + prenatal1 + mage + fbaby,  
 data = data,  
 method = "nearest",  
 distance = "logit")  
  
# Extract the matched data, specifying a different name for weights  
matched\_data <- match.data(psm, weights = "matching\_weights")  
  
# Extract common columns for raw and matched data  
common\_columns <- intersect(names(data), names(matched\_data))  
  
# Subset data to have the same columns  
data\_common <- data[common\_columns]  
matched\_data\_common <- matched\_data[common\_columns]  
  
# Add Group column for comparison  
data\_common$Group <- "Raw"  
matched\_data\_common$Group <- "Matched"  
  
# Combine both datasets  
combined\_data <- rbind(data\_common, matched\_data\_common)  
  
# Function to create normalized PMF plots for covariates  
create\_pmf\_plots <- function(combined\_data, covariate) {  
 # Create a subset of data for plotting  
 plot\_data <- combined\_data %>%  
 select(Group, mbsmoke, all\_of(covariate)) %>%  
 gather(key = "Variable", value = "Value", -Group, -mbsmoke)  
   
 # Convert Factors to Numeric for PMF  
 if (is.factor(plot\_data$Value)) {  
 plot\_data$Value <- as.numeric(as.factor(plot\_data$Value))  
 }  
   
 # Calculate normalized counts within each group and treatment status  
 plot\_data <- plot\_data %>%  
 group\_by(Group, mbsmoke, Value) %>%  
 summarize(Count = n()) %>%  
 group\_by(Group, mbsmoke) %>%  
 mutate(Probability = Count / sum(Count)) %>%  
 ungroup()  
   
 # Create PMF plots  
 plot <- ggplot(plot\_data, aes(x = factor(Value), y = Probability, fill = mbsmoke)) +  
 geom\_bar(stat = "identity", position = "dodge", alpha = 0.7) +  
 facet\_grid(Group ~ mbsmoke, scales = "free\_x") +  
 theme\_minimal() +  
 labs(title = paste("Normalized PMF Plot for", covariate), x = covariate, y = "Probability")  
   
 print(plot)  
}  
  
# List of covariates  
covariates <- c("mmarried", "prenatal1", "mage", "fbaby")  
  
# Loop to create and print PMF plots for all covariates  
for (covariate in covariates) {  
 create\_pmf\_plots(combined\_data, covariate)  
}

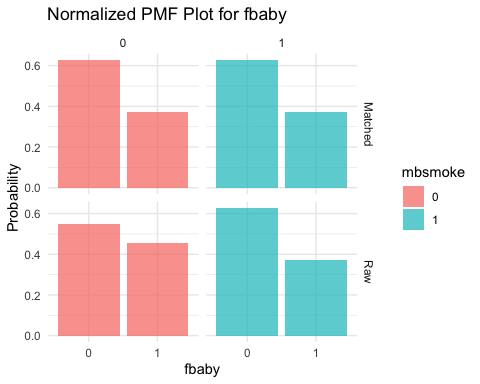
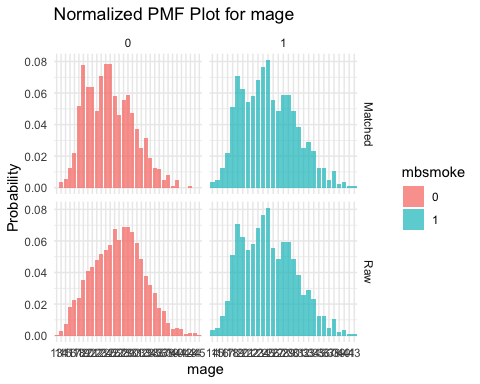
## `summarise()` has grouped output by 'Group', 'mbsmoke'. You can override using  
## the `.groups` argument.  
## `summarise()` has grouped output by 'Group', 'mbsmoke'. You can override using  
## the `.groups` argument.



## `summarise()` has grouped output by 'Group', 'mbsmoke'. You can override using  
## the `.groups` argument.



## `summarise()` has grouped output by 'Group', 'mbsmoke'. You can override using  
## the `.groups` argument.



### Session Info

sessionInfo()

## R version 4.4.1 (2024-06-14)  
## Platform: aarch64-apple-darwin20  
## Running under: macOS 15.0.1  
##   
## Matrix products: default  
## BLAS: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib   
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib; LAPACK version 3.12.0  
##   
## locale:  
## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
##   
## time zone: America/Los\_Angeles  
## tzcode source: internal  
##   
## attached base packages:  
## [1] grid stats graphics grDevices utils datasets methods   
## [8] base   
##   
## other attached packages:  
## [1] lubridate\_1.9.3 forcats\_1.0.0 stringr\_1.5.1 dplyr\_1.1.4   
## [5] purrr\_1.0.2 readr\_2.1.5 tidyr\_1.3.1 tibble\_3.2.1   
## [9] tidyverse\_2.0.0 ggplot2\_3.5.1 MatchIt\_4.5.5 survey\_4.4-2   
## [13] survival\_3.7-0 Matrix\_1.7-0 CausalModels\_0.2.0 haven\_2.5.4   
##   
## loaded via a namespace (and not attached):  
## [1] gtable\_0.3.5 xfun\_0.47 lattice\_0.22-6 tzdb\_0.4.0   
## [5] vctrs\_0.6.5 tools\_4.4.1 generics\_0.1.3 sandwich\_3.1-1   
## [9] fansi\_1.0.6 highr\_0.11 pkgconfig\_2.0.3 lifecycle\_1.0.4   
## [13] causaldata\_0.1.3 compiler\_4.4.1 farver\_2.1.2 munsell\_0.5.1   
## [17] mitools\_2.4 codetools\_0.2-20 htmltools\_0.5.8.1 yaml\_2.3.10   
## [21] pillar\_1.9.0 MASS\_7.3-61 boot\_1.3-31 multcomp\_1.4-26   
## [25] tidyselect\_1.2.1 digest\_0.6.37 mvtnorm\_1.3-1 stringi\_1.8.4   
## [29] labeling\_0.4.3 geepack\_1.3.11.1 splines\_4.4.1 fastmap\_1.2.0   
## [33] colorspace\_2.1-1 cli\_3.6.3 magrittr\_2.0.3 utf8\_1.2.4   
## [37] broom\_1.0.6 TH.data\_1.1-2 withr\_3.0.1 scales\_1.3.0   
## [41] backports\_1.5.0 timechange\_0.3.0 rmarkdown\_2.28 zoo\_1.8-12   
## [45] chk\_0.9.2 hms\_1.1.3 evaluate\_0.24.0 knitr\_1.48   
## [49] rlang\_1.1.4 Rcpp\_1.0.13 glue\_1.7.0 DBI\_1.2.3   
## [53] rstudioapi\_0.16.0 R6\_2.5.1