Causal Inference: the Effect of Class Size on Test Scores

Qingyu Chen

2025-02-28

Project Introduction

This project examines the effect of **Grade 1 class size reduction** on students' **reading achievement** using a public dataset from the **Early Childhood Longitudinal Study-Kindergarten Cohort (ECLS-K) 1998** (NCES (https://nces.ed.gov/ecls/Kindergarten.asp)). The goal is to estimate whether students in **small classes (≤19 students)** perform better in reading than those in **regular classes (≥20 students)**.

To address potential selection bias, I use five causal inference method to test the Average Treatment on Treated (ATT) and Average Treatment Effect (ATE), including:
- **Propensity Score Matching (PSM)**
- **Inverse Probability of Treatment Weighting (IPTW)**
- **Marginal Mean Weighting through Stratification (MMWS)**
- **Instrumental Variable (IV)**
- **Difference-in-Differences (DID)**

```
knitr::opts_chunk$set(echo = TRUE, warning=FALSE, message=FALSE)
library(haven)
library(dplyr)
library(Hmisc)
library(MatchIt)
library(ggplot2)
library(tableone)
library(cobalt)
library(twang)
library(PSweight)
library(knitr)
library(broom)
library(kableExtra)
library(AER)
library(plm)
library(systemfit)
library(tidyverse)
library(fixest)
```

0. Data Cleaning

```
# Upload dataset and name it df1
data_url <- "https://raw.githubusercontent.com/QingyuChen-Joi/R_Coding_Sample/main/ECLSK98_class_size.dta"
df1 <- read_dta(url(data_url))

# Clean data set
df_cleaned <- df1 %>%
  # Missing values in gender variable are labled as 3
  mutate(gender3 = ifelse(GENDER == -9, 3, GENDER)) %>%
  # Create 6 categories of race
  mutate(race6 = case_when(RACE %in% c(3, 4) ~ 3/4,
                           RACE %in% c(6, 7) ~ 6/7,
                           RACE %in% c(-9, 8) ~ 8/-9,
                           RACE == 1 ~ 1,
                           RACE == 2 ~ 2,
                           RACE == 5 ~ 5)) %>%
  # Set missing value in the whole data set to NA
  mutate(across(everything(), ~ifelse(. == -1 |. == -9, NA, .))) %>%
  # For general missing
  mutate(race6 = as.factor(race6)) %>%
  mutate(clsize = case_when(
    A4CLSIZE <=19 ~ "Small",
    A4CLSIZE >=20 ~ "Regular"
  )) #For grade 1 class size

# Create indicator variables for continuous covariates
continuous_covs <- c("C1RRSCAL", "C2RRSCAL", "C1R2MSCL", "C2R2MSCL", "B4YRSTC")
for (var_name in continuous_covs) {
  df_cleaned[[paste0("missing_", var_name)]] <- ifelse(is.na(df_cleaned[[var_name]]), 1, 0)
  df_cleaned[[var_name]] <- ifelse(is.na(df_cleaned[[var_name]]), mean(df_cleaned[[var_name]], na.rm=TRUE), df_cl
eaned[[var_name]])
}

# Create indicator variables for categorical covariates
categorical_vars <- c("A4CLSIZE", "gender3", "race6", "clsize")
for (var_name in categorical_vars) {
  df_cleaned[[paste0("missing_", var_name)]] <- ifelse(is.na(df_cleaned[[var_name]]), 1, 0)
  df_cleaned[[var_name]] <- ifelse(is.na(df_cleaned[[var_name]]), NA, df_cleaned[[var_name]])
}

# Filter observations without valid reading score and class size data
df_cleaned <- df_cleaned %>%
  filter(!is.na(C4RRSCAL)) %>%
  filter(!is.na(clsize))
```

1. Compare the Baseline of Control and Treated

The average score of small class students is slightly lower than regular class students, but the difference is very small and not significant. effect size: the effect size is -0.0093 which is very close to zero, which means the prima facie effect of class size on reading score is very small.

```
# Compute mean and standard deviation
df_q1 <- df_cleaned %>%
  filter(!is.na(clsize)) %>%
  group_by(clsize) %>%
  summarise(
    mean.y = mean(C4RRSCAL, na.rm = TRUE),
           sd.y = sd(C4RRSCAL, na.rm = TRUE)
    )

# Compute between-group average difference
t_test_result <- t.test(C4RRSCAL ~ clsize, data = df_cleaned, var.equal = TRUE)
mean_diff <- t_test_result$estimate[2] -
  t_test_result$estimate[1] # control-treated

# Compute Standard Error and P value
std_error <- t_test_result$stderr
p_value <- t_test_result$p.value

# Compute effect size(Cohen's d, using the sd of control group)
control_sd <- df_q1$sd.y[df_q1$clsize == "Regular"]
effect_size <- mean_diff / control_sd

# Print results to table
result_table <- data.frame(
  Metric = c("Mean Difference", "Standard Error", "P-value", "Effect Size"),
  Value = c(mean_diff, std_error, p_value, effect_size)
)
print(result_table)
```

```
##            Metric        Value
## 1 Mean Difference -0.121060243
## 2  Standard Error  0.228446741
## 3         P-value  0.596170499
## 4     Effect Size -0.009287546
```

2. Construct Propensity Score and Identify Common Support

2.1 Selecting Confounders from Pre-treatment Confounders

1. The treatment group and control group have no significant differences in terms of gender. In the table, we can see the p value of gender(gender3) is 0.237 > 0.05. Therefore, gender variable is not counted as confounders in future analysis. All the missing indicators are insignificant, too.
2. The treatment group and control group have significant differences in academic performance and race. For reading score in fall kindergarten (C1RRSCAL), reading score in spring kindergarten (C2RRSCAL),math score in fall kindergarten (C1R2MSCL), math score in spring kindergarten (C2R2MSCL), and Grade 1 teacher's teaching experience in years (B4YRSTC), the p values are all smaller than 0.01, which indicates that the two groups are different in terms of academic performance. Race has a P value of 0.082 which is also statistically significant. These variables are counted as confounders in future analysis.
3. The regular class group(the control group) seems to be relatively advantages. Because students in small classes had lower average scores on all academic achievement variables and teaching experience than students in regular classes.
4. I draw a table to show the variance ratio and covariance ratio of covariates between control group and treated group. In this table, the diagonal is variance ratio, others are covariance ratio.
   all numbers in the diagonal is close to 1, which means the covariates variance between two groups are not significantly different.
   For other numbers, B4YRSTC:C1RRSCAL = 0.46, B4YRSTC:C2RRSCAL = -0.51, B4YRSTC:C1R2MSCL = 0.31, B4YRSTC:C2R2MSCL = 0.23, which proves these covariance are different between control and treated groups. Therefore, they should be in the form of X1X2.
   Other numbers are all between 0.9-1, so no need to add quadratic terms.

Adjusted model:
$Y = \beta_0 + \beta_1 \times \text{ClassSize} + \beta_2 \times C1RRSCAL + \beta_3 \times C2RRSCAL + \beta_4 \times C1R2MSCL + \beta_5 \times C2R2MSCL + \beta_6 \times B4YRSTC + \beta_7 \times race6 + \beta_8 \times B$

```
# Difference in X=1 and X=0 of potential confounders (pre-identified)
covariates <- c("gender3",
                "race6",
                "C1RRSCAL", # Reading score in fall kindergarten
                "C2RRSCAL", # Reading score in spring kindergarten
                "C1R2MSCL", # Math score in fall kindergarten
                "C2R2MSCL", # Math score in spring kindergarten
                "B4YRSTC", # Grade 1 teacher's teaching experience in years
                "missing_B4YRSTC",
                "missing_C1RRSCAL",
                "missing_C2RRSCAL",
                "missing_C1R2MSCL",
                "missing_C2R2MSCL")
compare <- CreateTableOne(vars = covariates, strata = "clsize", data = df_cleaned, test = TRUE)
print(compare, showAllLevels = TRUE)
```

```
##                              Stratified by clsize
##                               level Regular       Small         p      test
##   n                                  8661          5346
##   gender3 (mean (SD))                1.50 (0.50)   1.49 (0.50)   0.237
##   race6 (mean (SD))                  3.84 (1.14)   3.87 (1.06)   0.082
##   C1RRSCAL (mean (SD))              23.71 (8.29)  23.13 (7.87)  <0.001
##   C2RRSCAL (mean (SD))              34.06 (10.62) 33.35 (10.58) <0.001
##   C1R2MSCL (mean (SD))              22.32 (8.25)  21.77 (8.04)  <0.001
##   C2R2MSCL (mean (SD))              32.61 (10.89) 32.12 (10.87)  0.010
##   B4YRSTC (mean (SD))               14.83 (10.15) 13.90 (9.83)  <0.001
##   missing_B4YRSTC (mean (SD))        0.02 (0.14)   0.02 (0.16)   0.074
##   missing_C1RRSCAL (mean (SD))       0.13 (0.34)   0.13 (0.34)   0.851
##   missing_C2RRSCAL (mean (SD))       0.03 (0.18)   0.03 (0.17)   0.390
##   missing_C1R2MSCL (mean (SD))       0.15 (0.36)   0.16 (0.37)   0.199
##   missing_C2R2MSCL (mean (SD))       0.08 (0.27)   0.08 (0.26)   0.813
```

```
df_q2 <- df_cleaned %>%
  mutate(clsize_binary = ifelse(clsize == "Small", 1, 0))

# Test numerical variables to identify quadratic terms
num_covariates <- c("C1RRSCAL", "C2RRSCAL", "C1R2MSCL", "C2R2MSCL", "B4YRSTC")

# Calculate the covariance between groups
treated_cov <- cov(df_q2[df_q2$clsize_binary == 1, num_covariates])
control_cov <- cov(df_q2[df_q2$clsize_binary == 0, num_covariates])

# Calculate the ratio of covariance
covariance_ratio <- treated_cov / control_cov
print(covariance_ratio)
```

```
##           C1RRSCAL   C2RRSCAL  C1R2MSCL  C2R2MSCL    B4YRSTC
## C1RRSCAL 0.9001573  0.9172252 0.9357523 0.9725597  0.4602305
## C2RRSCAL 0.9172252  0.9912489 0.9407812 1.0043151 -0.5064695
## C1R2MSCL 0.9357523  0.9407812 0.9498969 0.9653721  0.3127457
## C2R2MSCL 0.9725597  1.0043151 0.9653721 0.9953835  0.2340105
## B4YRSTC  0.4602305 -0.5064695 0.3127457 0.2340105  0.9383527
```

```
# Use logit regression to construct propensity score
logit_treat_adjusted <- glm(clsize_binary ~
                              C1RRSCAL +
                              C2RRSCAL +
                              C1R2MSCL +
                              C2R2MSCL +
                              B4YRSTC +
                              B4YRSTC:C1RRSCAL +
                              B4YRSTC:C2RRSCAL +
                              B4YRSTC:C1R2MSCL +
                              B4YRSTC:C2R2MSCL,
                            data = df_q2,
                            family = binomial)

# Report in table
coeff_table <- summary(logit_treat_adjusted)$coefficients
coeff_df <- as.data.frame(coeff_table)
coeff_df$Variable <- rownames(coeff_df)
coeff_df <- coeff_df[, c("Variable", "Estimate", "Std. Error", "z value", "Pr(>|z|)")]
print(coeff_df)
```

```
##                          Variable      Estimate   Std. Error    z value
## (Intercept)           (Intercept) -0.3232552397 0.1118413356 -2.8903020
## C1RRSCAL                 C1RRSCAL -0.0198838341 0.0068577753 -2.8994584
## C2RRSCAL                 C2RRSCAL  0.0092249090 0.0049804327  1.8522304
## C1R2MSCL                 C1R2MSCL  0.0027511343 0.0069529840  0.3956768
## C2R2MSCL                 C2R2MSCL  0.0021309053 0.0049742721  0.4283853
## B4YRSTC                   B4YRSTC  0.0048097612 0.0063838719  0.7534238
## C1RRSCAL:B4YRSTC C1RRSCAL:B4YRSTC  0.0011655283 0.0003912406  2.9790581
## C2RRSCAL:B4YRSTC C2RRSCAL:B4YRSTC -0.0009248513 0.0002846795 -3.2487463
## C1R2MSCL:B4YRSTC C1R2MSCL:B4YRSTC -0.0006636791 0.0004025850 -1.6485441
## C2R2MSCL:B4YRSTC C2R2MSCL:B4YRSTC  0.0001361778 0.0002860448  0.4760714
##                     Pr(>|z|)
## (Intercept)      0.003848719
## C1RRSCAL         0.003738080
## C2RRSCAL         0.063992743
## C1R2MSCL         0.692343485
## C2R2MSCL         0.668370591
## B4YRSTC          0.451195263
## C1RRSCAL:B4YRSTC 0.002891359
## C2RRSCAL:B4YRSTC 0.001159148
## C1R2MSCL:B4YRSTC 0.099241068
## C2R2MSCL:B4YRSTC 0.634023482
```

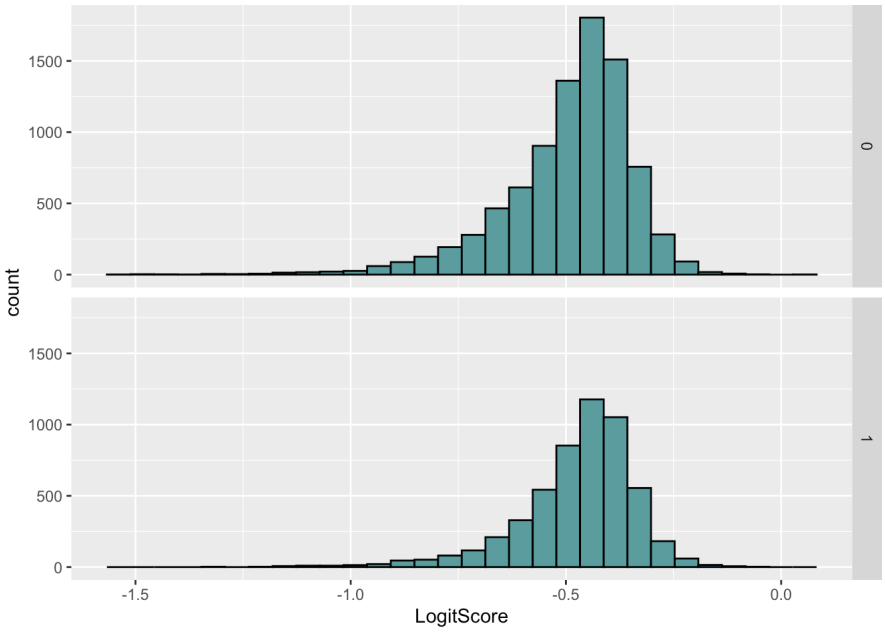
```
# Logit Scores, show the original linear prediction result
df_q2$LogitScore <- predict(logit_treat_adjusted, type = "link" )

# Probability of Treatment, count as propensity score
df_q2$PropScore <- predict(logit_treat_adjusted, type = "response")
```

2.2 Balance Checking of Logit Scores

Based on the balance check results, there is a significant difference in LogitScore between the two groups defined by clsize_binary. The Welch's t-test yielded a t-value of -8.30 with a p-value < 2.2e-16, indicating that the mean LogitScore differs significantly between the two groups. The confidence interval (-0.0253, -0.0156) further supports this conclusion.

Additionally, the mean LogitScore for group 0 is -0.4931, while for group 1, it is -0.4726, suggesting a small but statistically significant imbalance. The histogram visualization also shows noticeable differences in distribution. Given these findings, LogitScore is not perfectly balanced across the two groups, which may require further adjustments, such as matching or weighting, to ensure comparability in causal analysis.

```
ggplot(df_q2, aes(x = LogitScore)) +
  geom_histogram(fill = "cadetblue", colour = "black") +
  facet_grid(clsize_binary ~ .)
```



```
logit_summary <- df_q2 %>%
  group_by(clsize_binary) %>%
  summarise(
    mean_logit = mean(LogitScore, na.rm = TRUE),
    var_logit = var(LogitScore, na.rm = TRUE),
    n = n()
  )
print(logit_summary)
```

```
## # A tibble: 2 × 4
##   clsize_binary mean_logit var_logit     n
##           <dbl>      <dbl>     <dbl> <int>
## 1             0     -0.493    0.0227  8661
## 2             1     -0.473    0.0185  5346
```

```
t_test_result <- t.test(LogitScore ~ clsize_binary, data = df_q2)
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  LogitScore by clsize_binary
## t = -8.2966, df = 12193, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.02526379 -0.01560750
## sample estimates:
## mean in group 0 mean in group 1
##      -0.4930356      -0.4726000
```

2.3 Common Support

```
# Compute the standard deviation of the propensity score (LogitScore)
logit_sd <- sd(df_q2$LogitScore, na.rm = TRUE)

# Set the caliper as 20% of the logit propensity score standard deviation
caliper <- 0.2 * logit_sd

# Get the propensity score range for the treatment and control groups
range_treat <- range(df_q2$LogitScore[df_q2$clsize_binary == 1], na.rm = TRUE)
range_control <- range(df_q2$LogitScore[df_q2$clsize_binary == 0], na.rm = TRUE)

# Determine the common support region:
# Use the highest minimum and lowest maximum, adjusting for the caliper
common_min <- max(min(range_treat), min(range_control)) - caliper
common_max <- min(max(range_treat), max(range_control)) + caliper

# Print the common support region
cat("Common Support Region: [", common_min, ",", common_max, "]\n")
```

```
## Common Support Region: [ -1.353463 , -0.008207766 ]
```

```
# Mrk individuals whose propensity scores fall outside the common support range
df_q2$propensity_score_outlier <- ifelse(df_q2$LogitScore < common_min |
                                         df_q2$LogitScore > common_max, 1, 0)

# Report the number of extreme cases to be excluded
num_extreme_cases <- sum(df_q2$propensity_score_outlier, na.rm = TRUE)
cat("Number of extreme cases to be excluded:", num_extreme_cases, "\n")
```

```
## Number of extreme cases to be excluded: 7
```

```
# Filter out individuals who do not fall within the common support region
df_q2 <- df_q2 %>%
  filter(propensity_score_outlier == 0)

# Check the distribution of propensity scores after filtering
summary(df_q2$LogitScore)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.34895 -0.54739 -0.45746 -0.48485 -0.39182 -0.03729
```

3. PSM (before stratification)

3.1 Propensity Score Matching and Balance Checking

The balance check results show that propensity score matching significantly improved covariate balance. Standardized differences decreased to near zero, and variance ratios remained close to 1, indicating that the matching process effectively reduced baseline differences between treatment and control groups.

```
matched <- matchit(clsize_binary ~
                     C1RRSCAL +
                     C2RRSCAL +
                     C1R2MSCL +
                     C2R2MSCL +
                     B4YRSTC +
                     B4YRSTC:C1RRSCAL +
                     B4YRSTC:C2RRSCAL +
                     B4YRSTC:C1R2MSCL +
                     B4YRSTC:C2R2MSCL,
                   data=df_q2, method= "nearest", distance = "glm",
                   replace = FALSE)

# Standardized difference and the variance ratio
balance_check <- bal.tab(matched, un = TRUE, stats = c("mean.diffs", "variance.ratios"))
balance_df <- as.data.frame(balance_check$Balance)
balance_comparison <- balance_df[, c("Diff.Un", "V.Ratio.Un", "Diff.Adj", "V.Ratio.Adj")]
colnames(balance_comparison) <- c("Std_Diff_Before", "Var_Ratio_Before",
                                  "Std_Diff_After", "Var_Ratio_After")

print(balance_comparison)
```

```
##          Std_Diff_Before Var_Ratio_Before Std_Diff_After Var_Ratio_After
## distance      0.14556793        0.8532781   0.0009253427       1.0002060
## C1RRSCAL     -0.07136087        0.9051629  -0.0012701608       1.0304979
## C2RRSCAL     -0.06405032        1.0031582   0.0141102196       1.0883472
## C1R2MSCL     -0.06603371        0.9589655   0.0030606182       1.0245842
## C2R2MSCL     -0.04280418        1.0015727   0.0102788503       1.0211754
## B4YRSTC      -0.09398415        0.9400634   0.0097725714       0.9851834
```

3.2 ATT of PSM

Being in a small class significantly improves student achievement, with an estimated increase of 0.676 points. The effect size, standardized by the control group's standard deviation, is 0.047, indicating a small but meaningful positive impact.

```
# Use matched sample to estimate the effect
matched_data <- match.data(matched)
reading_model <- lm(C4RRSCAL ~ clsize_binary +
                     C1RRSCAL +
                     C2RRSCAL +
                     C1R2MSCL +
                     C2R2MSCL +
                     B4YRSTC +
                     B4YRSTC:C1RRSCAL +
                     B4YRSTC:C2RRSCAL +
                     B4YRSTC:C1R2MSCL +
                     B4YRSTC:C2R2MSCL, data = matched_data)

# Extract key indicators
model_summary <- summary(reading_model)
coef_values <- model_summary$coefficients["clsize_binary", ]
estimate <- coef_values["Estimate"]
sd_control <- sd(matched_data$C4RRSCAL[matched_data$clsize_binary == 0], na.rm = TRUE)
effect_size <- estimate / sd_control
# Compute a table for results
result_table <- data.frame(
  Estimate = coef_values["Estimate"],
  Effect_Size = effect_size,
  Std_Error = coef_values["Std. Error"],
  T_Value = coef_values["t value"],
  P_Value = coef_values["Pr(>|t|)"]
)

print(result_table)
```

```
##           Estimate Effect_Size Std_Error  T_Value      P_Value
## Estimate 0.6759951  0.05252645 0.1683264 4.015978 5.960499e-05
```

4. PSM (after stratification)

4.1 stratification and balance checking of PSM

```
df_q4 <- df_q2 %>%
  filter(LogitScore >= common_min & LogitScore <= common_max) %>%
  mutate(Stratum = as.factor(as.numeric(cut(LogitScore, breaks = 5, include.lowest = TRUE))))

# Check balance for each stratum(standardized mean difference, variance ratio)
stratum_summary <- df_q4 %>%
  group_by(Stratum) %>%
  summarise(
    mean_treat = mean(LogitScore[clsize_binary == 1], na.rm = TRUE),
    mean_control = mean(LogitScore[clsize_binary == 0], na.rm = TRUE),
    sd_treat = sd(LogitScore[clsize_binary == 1], na.rm = TRUE),
    sd_control = sd(LogitScore[clsize_binary == 0], na.rm = TRUE),
    var_treat = var(LogitScore[clsize_binary == 1], na.rm = TRUE),
    var_control = var(LogitScore[clsize_binary == 0], na.rm = TRUE),
    SMD = abs(mean_treat - mean_control) / sqrt((sd_treat + sd_control) / 2),
    VR = var_treat / var_control
  )

# Check balance for covariates(standardized mean difference, variance ratio)
bal.tab(clsize ~ C1RRSCAL +
            C2RRSCAL +
            C1R2MSCL +
            C2R2MSCL +
            B4YRSTC +
            B4YRSTC:C1RRSCAL +
            B4YRSTC:C2RRSCAL +
            B4YRSTC:C1R2MSCL +
            B4YRSTC:C2R2MSCL,
          data = df_q4,
          strata  = "Stratum",
          estimand = "ATE",
          pool    = TRUE,
          poly    = 2,
          stats   = c("mean.diffs", "variance.ratios")
          )
```

```
## Balance Measures
##                         Type Diff.Un V.Ratio.Un
## C1RRSCAL           Contin. -0.0696     0.9052
## C2RRSCAL           Contin. -0.0641     1.0032
## C1R2MSCL           Contin. -0.0653     0.9590
## C2R2MSCL           Contin. -0.0428     1.0016
## B4YRSTC            Contin. -0.0925     0.9401
## C1RRSCAL * B4YRSTC Contin. -0.1109     0.8865
## C2RRSCAL * B4YRSTC Contin. -0.1194     0.8595
## C1R2MSCL * B4YRSTC Contin. -0.1162     0.8687
## C2R2MSCL * B4YRSTC Contin. -0.1067     0.8772
## C1RRSCAL²          Contin. -0.0618     0.8211
## C2RRSCAL²          Contin. -0.0523     0.9535
## C1R2MSCL²          Contin. -0.0554     0.9539
## C2R2MSCL²          Contin. -0.0356     0.9727
## B4YRSTC²           Contin. -0.0956     0.8944
##
## Sample sizes
##     Regular Small
## All    8654  5346
```

```
# Check average standardized mean difference and variance ratio
avg_SMD <- mean(stratum_summary$SMD, na.rm = TRUE)
avg_VR <- mean(stratum_summary$VR, na.rm = TRUE)

# Print results
options(max.print = 99999)
print(stratum_summary, n = Inf)
```

```
## # A tibble: 5 × 9
##   Stratum mean_treat mean_control sd_treat sd_control var_treat var_control
##   <fct>        <dbl>        <dbl>    <dbl>      <dbl>     <dbl>       <dbl>
## 1 1           -1.16        -1.19    0.0664     0.0736   0.00441     0.00542
## 2 2           -0.910       -0.911   0.0679     0.0706   0.00461     0.00498
## 3 3           -0.647       -0.654   0.0681     0.0682   0.00463     0.00466
## 4 4           -0.432       -0.436   0.0646     0.0642   0.00417     0.00412
## 5 5           -0.254       -0.256   0.0449     0.0408   0.00202     0.00166
## # ℹ 2 more variables: SMD <dbl>, VR <dbl>
```

```
avg_SMD
```

```
## [1] 0.04256851
```

```
avg_VR
```

```
## [1] 0.9923451
```

4.2 Trend of Reading Score in Two groups

A clear trend is observed. In the first stratum, the treated group performs significantly better. As we move towards the fifth stratum, the difference becomes less pronounced.

```
# Compute the mean difference in Grade 1 reading score (C4RRSCAL) within each stratum
treatment_effects <- df_q4 %>%
  group_by(Stratum) %>%
  summarise(
    mean_treat = mean(C4RRSCAL[clsize_binary == 1], na.rm = TRUE),
    mean_control = mean(C4RRSCAL[clsize_binary == 0], na.rm = TRUE),
    mean_diff = mean_treat - mean_control  # Compute mean difference (ATE)
  )

# Output Stratum-Specific Treatment Effects
print(treatment_effects)
```

```
## # A tibble: 5 × 4
##   Stratum mean_treat mean_control mean_diff
##   <fct>        <dbl>        <dbl>     <dbl>
## 1 1             77.1         76.3     0.809
## 2 2             73.4         73.9    -0.528
## 3 3             64.1         63.3     0.750
## 4 4             54.5         54.0     0.555
## 5 5             56.7         57.4    -0.625
```

```
# Plot the mean score trends for the treatment and control groups
ggplot(treatment_effects, aes(x = as.numeric(Stratum))) +
  geom_line(aes(y = mean_treat, color = "Treated Group"), size = 1) +
  geom_point(aes(y = mean_treat, color = "Treated Group"), size = 3) +
  geom_line(aes(y = mean_control, color = "Control Group"), size = 1) +
  geom_point(aes(y = mean_control, color = "Control Group"), size = 3) +
  scale_color_manual(values = c("Treated Group" = "blue", "Control Group" = "red")) +
  labs(title = "Stratum-Specific Treatment Effects of Class Size",
       x = "Stratum (Propensity Score Quintile)",
       y = "Mean Grade 1 Reading Score",
       color = "Group") +
  theme_minimal()
```

4.3 ATE of PSM (Add PS Score to Model)

```
# Run model to calculate treatment effect
df_q4$Stratum <- as.factor(df_q4$Stratum)
model <- lm(C4RRSCAL ~ clsize_binary +
              C1RRSCAL +
              C2RRSCAL +
              C1R2MSCL +
              C2R2MSCL +
              B4YRSTC +
              B4YRSTC:C1RRSCAL +
              B4YRSTC:C2RRSCAL +
              B4YRSTC:C1R2MSCL +
              B4YRSTC:C2R2MSCL + Stratum, data = df_q4)

model_summary <- summary(model)

# Extract results
estimate <- model_summary$coefficients["clsize_binary", "Estimate"]
std_error <- model_summary$coefficients["clsize_binary", "Std. Error"]
t_value <- model_summary$coefficients["clsize_binary", "t value"]
p_value <- model_summary$coefficients["clsize_binary", "Pr(>|t|)"]
sd_control <- sd(df_q4$C4RRSCAL[df_q4$clsize_binary == 0], na.rm = TRUE)
effect_size <- estimate / sd_control
# Create data frame
result_table <- data.frame(
  Estimate = estimate,
  Effect_Size = effect_size,
  Std_Error = std_error,
  T_Value = t_value,
  P_Value = p_value
)
# Print results
print(result_table)
```

```
##    Estimate Effect_Size Std_Error T_Value      P_Value
## 1 0.5984877  0.04597114 0.1500921 3.98747 6.712134e-05
```

5. IPTW

The IPTW analysis results suggest that class size has a significant effect on first-grade reading achievement. The estimated average treatment effect (ATE) of being in a smaller class is 0.6194, indicating a positive impact on reading scores. This effect is statistically significant, with a p-value of 0.0067, suggesting strong evidence that class size influences student performance rather than being a result of random variation.

Balance diagnostics show that covariates are well balanced after weighting, as all standardized mean differences (SMDs) are below 0.05. The effective sample sizes after weighting remain substantial, with 8,630.55 in the control group and 5,299.41 in the treated group, ensuring reliable estimation. Additionally, the weighted regression results show that the absolute standardized difference is reduced to nearly 0.0018, and the variance ratio is close to 1, further supporting the success of the weighting procedure.

The estimated effect size is relatively small at 0.0476, meaning that while the impact of class size on reading achievement is statistically significant, its magnitude is modest. This finding suggests that reducing class size does contribute to improved reading scores, but the effect size indicates that other factors may also play a crucial role in student performance.

5.1 IPTW Weighting

```
# Compute IPTW weights for ATE
df_q5 <- df_q4 %>%
  mutate(
    pred = PropScore,
    Z = clsize_binary,
    W_ATE = ifelse(Z == 1, mean(Z) / pred, (1 - mean(Z)) / (1 - pred)),
    logit_PS = log(PropScore / (1 - PropScore))  # Compute logit propensity score
  )

# Unweighted logit PS regression (checking PS balance before weighting)
lm_logit_before <- lm(logit_PS ~ Z, data = df_q5)

# Weighted logit PS regression (checking PS balance after weighting)
lm_logit_after <- lm(logit_PS ~ Z, data = df_q5, weights = W_ATE)

# Balance check before weighting
bal_before <- bal.tab(Z ~ C1RRSCAL + C2RRSCAL + C1R2MSCL + C2R2MSCL + B4YRSTC +
                        B4YRSTC:C1RRSCAL + B4YRSTC:C2RRSCAL +
                        B4YRSTC:C1R2MSCL + B4YRSTC:C2R2MSCL,
                      data = df_q5,
                      estimand = "ATE",
                      m.threshold = 0.05,
                      disp.v.ratio = TRUE)

# Balance check after weighting
bal_after <- bal.tab(Z ~ C1RRSCAL + C2RRSCAL + C1R2MSCL + C2R2MSCL + B4YRSTC +
                        B4YRSTC:C1RRSCAL + B4YRSTC:C2RRSCAL +
                        B4YRSTC:C1R2MSCL + B4YRSTC:C2R2MSCL,
                      data = df_q5,
                      estimand = "ATE",
                      m.threshold = 0.05,
                      disp.v.ratio = TRUE,
                      weights = df_q5$W_ATE,
                      method = "weighting")

# Print balance check results
print(bal_after)
```

```
## Balance Measures
##                       Type Diff.Adj     M.Threshold V.Ratio.Adj
## C1RRSCAL           Contin.   0.0019 Balanced, <0.05      1.0345
## C2RRSCAL           Contin.   0.0042 Balanced, <0.05      1.1022
## C1R2MSCL           Contin.   0.0020 Balanced, <0.05      1.0504
## C2R2MSCL           Contin.   0.0019 Balanced, <0.05      1.0467
## B4YRSTC            Contin.  -0.0004 Balanced, <0.05      0.9752
## C1RRSCAL * B4YRSTC Contin.   0.0013 Balanced, <0.05      1.0116
## C2RRSCAL * B4YRSTC Contin.   0.0024 Balanced, <0.05      1.0067
## C1R2MSCL * B4YRSTC Contin.   0.0014 Balanced, <0.05      1.0148
## C2R2MSCL * B4YRSTC Contin.   0.0009 Balanced, <0.05      0.9908
##
## Balance tally for mean differences
##                     count
## Balanced, <0.05         9
## Not Balanced, >0.05     0
##
## Variable with the greatest mean difference
##  Variable Diff.Adj     M.Threshold
##  C2RRSCAL   0.0042 Balanced, <0.05
##
## Effective sample sizes
##            Control Treated
## Unadjusted 8654.   5346.
## Adjusted   8630.55 5299.41
```

```
# Extract balance summary
balance_summary <- data.frame(
  Measure = c("Regression Coefficient (Z)", "Std. Error", "P-value",
              "Avg. Abs. Std. Diff", "Avg. Variance Ratio"),
  Unweighted = c(
    coef(lm_logit_before)["Z"],
    summary(lm_logit_before)$coefficients["Z", "Std. Error"],
    summary(lm_logit_before)$coefficients["Z", "Pr(>|t|)"],
    mean(abs(bal_before$Balance$Diff.Un)),  # Average standardized mean difference (Unweighted)
    mean(bal_before$Balance$V.Ratio.Un)  # Average variance ratio (Unweighted)
  ),
  Weighted = c(
    coef(lm_logit_after)["Z"],
    summary(lm_logit_after)$coefficients["Z", "Std. Error"],
    summary(lm_logit_after)$coefficients["Z", "Pr(>|t|)"],
    mean(abs(bal_after$Balance$Diff.Adj)),  # Average standardized mean difference (Weighted)
    mean(bal_after$Balance$V.Ratio.Adj)  # Average variance ratio (Weighted)
  )
)

# Print final balance summary table
print(balance_summary)
```

```
##                     Measure   Unweighted      Weighted
## 1 Regression Coefficient (Z) 1.981466e-02 -0.0004619369
## 2                Std. Error 2.498242e-03  0.0025119657
## 3                   P-value 2.329512e-15  0.8540988097
## 4       Avg. Abs. Std. Diff 8.750364e-02  0.0018098727
## 5       Avg. Variance Ratio 9.223019e-01  1.0258858102
```

5.2 IPTW ATE

```
# Perform IPTW-weighted regression
ate_model <- lm(C4RRSCAL ~ clsize_binary, data = df_q5, weights = W_ATE)

# Extract regression results for the treatment effect
estimate <- coef(ate_model)["clsize_binary"]  # Estimated ATE
std_error <- summary(ate_model)$coefficients["clsize_binary", "Std. Error"]  # Standard error
t_value <- summary(ate_model)$coefficients["clsize_binary", "t value"]  # t-statistic
p_value <- summary(ate_model)$coefficients["clsize_binary", "Pr(>|t|)"]  # p-value

# Compute effect size (Cohen's d) using the standard deviation of the control group
sd_control <- sd(df_q5$C4RRSCAL[df_q5$clsize_binary == 0], na.rm = TRUE)
cohen_d <- estimate / sd_control  # Standardized effect size

# Create a results table
results <- data.frame(
  Estimate = estimate,
  Std_Error = std_error,
  t_value = t_value,
  p_value = p_value,
  Effect_Size = cohen_d
)

# Print results in a formatted table
kable(results, caption = "ATE Estimation Results for Class Size on Grade 1 Reading Achievement")
```

ATE Estimation Results for Class Size on Grade 1 Reading Achievement

	Estimate	Std_Error	t_value	p_value	Effect_Size
clsize_binary	0.6194184	0.2283129	2.713024	0.0066754	0.0475789

6. MMWS

The MMWS analysis results indicate that class size has a statistically significant effect on first-grade reading achievement. The estimated ATE is 0.6061, suggesting that being in a smaller class leads to an increase in reading scores. The p-value is 3.41e-05, which confirms that this effect is unlikely to be due to chance.

The balance diagnostics show that the weighting procedure effectively reduced covariate imbalances, with an average standardized mean difference (SMD) of 0.0039 and a variance ratio close to 1, indicating a well-balanced sample. This ensures that the estimated treatment effect is not driven by pre-existing differences between the groups.

The effect size is 0.0496, which is small but consistent with prior findings on class size effects. This suggests that while reducing class size does have a positive impact on reading achievement, the practical significance might be limited, and other factors likely contribute to student performance.

6.1 MMWS Weighting

```
# Construct weight
df_q6 <- df_q2 %>%
  filter(LogitScore >= common_min & LogitScore <= common_max) %>%
  mutate(Stratum = as.factor(cut(LogitScore, breaks = quantile(LogitScore, probs = seq(0, 1, 0.2)), include.lowes
t = TRUE)))

df_q6 <- df_q6 %>%
  group_by(Stratum) %>%
  mutate(
    n_treated = sum(clsize_binary == 1),
    n_control = sum(clsize_binary == 0),
    MMWS_weight = case_when(
      clsize_binary == 1 & n_treated > 0 ~ n() / n_treated,
      clsize_binary == 0 & n_control > 0 ~ n() / n_control,
      TRUE ~ NA_real_
    )
  ) %>%
  ungroup()

mmws_model <- lm(C4RRSCAL ~ clsize_binary, data = df_q6, weights = MMWS_weight)
summary(mmws_model)
```

```
##
## Call:
## lm(formula = C4RRSCAL ~ clsize_binary, data = df_q6, weights = MMWS_weight)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -69.472 -11.624  -0.385  13.488  55.280
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    56.5339     0.1573 359.468   <2e-16 ***
## clsize_binary   0.5988     0.2224   2.692   0.0071 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.61 on 13998 degrees of freedom
## Multiple R-squared:  0.0005176,  Adjusted R-squared:  0.0004462
## F-statistic: 7.249 on 1 and 13998 DF,  p-value: 0.007104
```

```
# Calculate covariate balance
bal_results_mmws <- bal.tab(clsize_binary ~ C1RRSCAL +
                              C2RRSCAL +
                              C1R2MSCL +
                              C2R2MSCL +
                              B4YRSTC +
                              B4YRSTC:C1RRSCAL +
                              B4YRSTC:C2RRSCAL +
                              B4YRSTC:C1R2MSCL +
                              B4YRSTC:C2R2MSCL,
                            data = df_q6, weights = df_q6$MMWS_weight, un = TRUE,
                            disp.v.ratio = TRUE, disp.cov.ratio = TRUE,
                            s.d.denom = "pooled")

# Check Balance of Logit Propensity Score
logit_ps_model_unweighted <- lm(LogitScore ~ clsize_binary, data = df_q6)
logit_ps_model_weighted <- lm(LogitScore ~ clsize_binary, data = df_q6, weights = MMWS_weight)


balance_summary_mmws <- data.frame(
  Measure = c("Regression Coefficient (Z)", "Std. Error", "P-value",
              "Avg. Abs. Std. Diff", "Avg. Variance Ratio"),
  Unweighted = c(
    coef(logit_ps_model_unweighted)["clsize_binary"],
    summary(logit_ps_model_unweighted)$coefficients["clsize_binary", "Std. Error"],
    summary(logit_ps_model_unweighted)$coefficients["clsize_binary", "Pr(>|t|)"],
    mean(abs(bal_results_mmws$Balance$Diff.Un)),
    mean(bal_results_mmws$Balance$V.Ratio.Un)
  ),
  Weighted = c(
    coef(logit_ps_model_weighted)["clsize_binary"],
    summary(logit_ps_model_weighted)$coefficients["clsize_binary", "Std. Error"],
    summary(logit_ps_model_weighted)$coefficients["clsize_binary", "Pr(>|t|)"],
    mean(abs(bal_results_mmws$Balance$Diff.Adj)),
    mean(bal_results_mmws$Balance$V.Ratio.Adj)
  )
)

# Print the table
print(balance_summary_mmws)
```

```
##                     Measure   Unweighted    Weighted
## 1 Regression Coefficient (Z) 1.981466e-02 0.002011533
## 2                Std. Error 2.498242e-03 0.002423168
## 3                   P-value 2.329512e-15 0.406481913
## 4        Avg. Abs. Std. Diff 8.750364e-02 0.003881334
## 5        Avg. Variance Ratio 9.223019e-01 0.999502959
```

6.1 MMWS ATE

```
# Make sure MMWS doesn't have NA
df_q6 <- df_q6 %>%
  filter(!is.na(MMWS_weight))

# Estimate ATE
mmws_model <- lm(C4RRSCAL ~ clsize_binary +
                   C1RRSCAL +
                   C2RRSCAL +
                   C1R2MSCL +
                   C2R2MSCL +
                   B4YRSTC +
                   B4YRSTC:C1RRSCAL +
                   B4YRSTC:C2RRSCAL +
                   B4YRSTC:C1R2MSCL +
                   B4YRSTC:C2R2MSCL, data = df_q6, weights = MMWS_weight)

# Extract results
estimate <- coef(mmws_model)["clsize_binary"]
std_dev <- summary(mmws_model)$sigma
t_value <- summary(mmws_model)$coefficients["clsize_binary", "t value"]
p_value <- summary(mmws_model)$coefficients["clsize_binary", "Pr(>|t|)"]
cohen_d <- estimate / std_dev

# Create result table
results <- data.frame(
  Estimate = estimate,
  Std_Deviation = std_dev,
  t_value = t_value,
  p_value = p_value,
  Effect_Size = cohen_d
)
# Print the table
kable(results, caption = "MMWS Estimation Results for Class Size on Grade 1 Reading Achievement")
```

MMWS Estimation Results for Class Size on Grade 1 Reading Achievement

	Estimate	Std_Deviation	t_value	p_value	Effect_Size
clsize_binary	0.6060558	12.23055	4.145616	3.41e-05	0.0495526

7. Instrumental Variable Method(IV)

Model:

$$Y_i = \beta_0^{IV} + \beta_1^{IV} \hat{D_i} + X_i'\delta + \epsilon_i$$

In the first stage, the instrument (“Z,” defined as public school enrollment) shows a strong correlation with the endogenous regressor (Grade 1 class size, “D”). This is evident from the high first-stage F-statistic (about 67.65), which comfortably exceeds the common threshold of 10 for detecting weak instruments. Thus, Z appears strong in terms of predictive power for class size. In the second stage, the coefficient on class size (“D”) is negative and statistically significant, suggesting that as class size increases, reading achievement tends to decrease—consistent with the idea that larger classes can be detrimental to student performance.

However, the Sargan (or overidentification) test yields a very low p-value, indicating potential concerns about the instrument’s validity or unaccounted-for violations of the exclusion restriction. Moreover, the Wu-Hausman test shows no significant difference between the OLS and IV estimates (p-value = 1), suggesting that either endogeneity is not a major issue in this setting, or that the instrument does not substantially improve upon OLS in controlling for any endogeneity that may exist. Taken together, these results imply that the validity of IV may be questionable.

```
df_q7 <- df_cleaned %>%
  filter(s4pupri == 1,  # Select only public school students
         !is.na(C4RRSCAL), # Exclude individuals with no reading score
         !is.na(clsize), # Ensure valid Grade 1 class size
         !is.na(s4anumch), # Ensure valid public school enrollment
         !is.na(A4CLSIZE)) %>% # Ensure valid Grade 1 class size
  mutate(Z = s4anumch, # Define instrument variable (public school enrollment)
         D = A4CLSIZE, # Define endogenous variable (Grade 1 class size)
         race6 = as.factor(race6), # Convert race to a factor variable
         gender3 = as.factor(gender3)) # Convert gender to a factor variable

# Step 1: Regress Y (reading achievement) on D and adjust for X
Q7_first_stage <- lm(D ~ Z +
                       gender3 +
                       race6 +
                       w1sesl +
                       C1RRSCAL +
                       C2RRSCAL, data = df_q7)

# Extract F-statistic to check for weak instruments
first_stage_summary <- summary(Q7_first_stage)
first_stage_F <- first_stage_summary$fstatistic[1]
cat("First-stage regression F-statistic:", first_stage_F, "\n")
```

```
## First-stage regression F-statistic: 67.64747
```

```
# Step 2: 2SLS method
Q7_second_stage <- ivreg(C4RRSCAL ~ D +
                       gender3 +
                       race6 +
                       w1sesl +
                       C1RRSCAL +
                       C2RRSCAL |
                      Z +
                       gender3 +
                       race6 +
                       w1sesl +
                       C1RRSCAL +
                       C2RRSCAL, data = df_q7)

# Over Identification Test
sargan_test <- bptest(Q7_second_stage)  # Sargan test (based on Breusch-Pagan)
cat("Sargan test results:\n")
```

```
## Sargan test results:
```

```
print(sargan_test)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Q7_second_stage
## BP = 129.55, df = 10, p-value < 2.2e-16
```

```
# Endogeneity Test
ols_model <- lm(C4RRSCAL ~ D + gender3 + race6 + w1sesl + C1RRSCAL + C2RRSCAL, data = df_q7)
wu_hausman_test <- hausman.systemfit(Q7_second_stage, Q7_first_stage)  # Wu-Hausman test
cat("Wu-Hausman test results:\n")
```

```
## Wu-Hausman test results:
```

```
print(wu_hausman_test)
```

```
##
##  Hausman specification test for consistency of the 3SLS estimation
##
## data:  df_q7
## Hausman = -1239210, df = 11, p-value = 1
```

```
# Show Results
Q7_first_stage_results <- tidy(Q7_first_stage)
Q7_second_stage_results <- tidy(Q7_second_stage)

# Display first-stage regression results
kable(Q7_first_stage_results, format = "html", digits = 3) %>%
  kable_styling(full_width = F, bootstrap_options = c("striped", "hover"))
```

term	estimate	std.error	statistic	p.value
(Intercept)	18.227	0.287	63.473	0.000
Z	0.004	0.000	23.667	0.000
gender32	0.122	0.077	1.591	0.112
race62	-0.359	0.258	-1.393	0.164
race63	-1.347	0.317	-4.245	0.000
race64	-0.299	0.241	-1.238	0.216
race65	-0.016	0.258	-0.062	0.951
race66	-0.372	0.287	-1.296	0.195
w1sesl	0.335	0.056	5.963	0.000
C1RRSCAL	0.007	0.008	0.863	0.388
C2RRSCAL	-0.001	0.006	-0.140	0.888

```
# Display second-stage regression results
kable(Q7_second_stage_results, format = "html", digits = 3) %>%
  kable_styling(full_width = F, bootstrap_options = c("striped", "hover"))
```

term	estimate	std.error	statistic	p.value
(Intercept)	34.399	2.226	15.450	0.000
D	-0.329	0.104	-3.148	0.002
gender32	0.980	0.190	5.156	0.000
race62	-1.480	0.635	-2.330	0.020
race63	-2.204	0.795	-2.771	0.006
race64	0.618	0.598	1.034	0.301
race65	-1.148	0.638	-1.799	0.072
race66	0.000	0.708	0.000	1.000
w1sesl	2.040	0.145	14.073	0.000
C1RRSCAL	0.113	0.019	6.078	0.000
C2RRSCAL	0.771	0.014	53.973	0.000

```
# Show effect size of 2sls
beta_1_iv <- Q7_second_stage_results$estimate[2]

# Compute effect size
sd_Q7_Y_control <- sd(df_q7$C4RRSCAL[df_q7$clsize == "Regular"], na.rm = TRUE)
Q7_effect_size <- beta_1_iv / sd_Q7_Y_control
Q7_effect_size
```

```
## [1] -0.02522968
```

8. Difference in Difference Method(DID)

Model:

$$Y_{it} = \alpha + \beta_1 \text{SmallClass}_i + \beta_2 \text{Post}_t + \delta (\text{SmallClass}_i \times \text{Post}_t) + \varepsilon_{it},$$

ATT:

$$ATT = \delta = (\overline{C4RRSCAL}_{\text{small}} - \overline{C2RRSCAL}_{\text{small}}) - (\overline{C4RRSCAL}_{\text{regular}} - \overline{C2RRSCAL}_{\text{regular}}).$$

In a standard Difference-in-Differences framework, the main requirement is the parallel trends assumption, meaning that in the absence of the intervention, the outcome (in this case, reading achievement) would have evolved similarly for the treatment and control groups.
The results indicate that, at baseline, the control group's reading score is around 34.061, and from baseline to post-intervention, this score increases by about 22.758. The coefficient on clsize_new is -0.711, suggesting that at baseline, the group with the new (or smaller) class size starts off slightly lower than the control group. However, the crucial DID interaction term time_binary:clsize_new is 0.590 (p = 0.044), indicating that, after the intervention, the treatment group sees an additional 0.590-point improvement beyond what the control group experiences. Interpreted under the parallel trends assumption, this implies that reducing class size or adopting the new arrangement exerts a positive and statistically significant effect on students' reading scores.

```
# Generate DID variable
df_q8 <- df_cleaned %>%
  mutate(
    clsize_new = if_else(A4CLSIZE <= 19, 1, if_else(A4CLSIZE >= 20, 0, NA_real_)) # Define small class vs. large
class
  ) %>%
  filter(!is.na(clsize_new)) %>%
  # Reshape data for different time periods
  pivot_longer(cols = c(C2RRSCAL, C4RRSCAL),
               names_to = "time", values_to = "reading_score") %>%
  mutate(
    time_binary = if_else(time == "C2RRSCAL", 0, 1) # 0 = Kindergarten, 1 = Grade 1
  )

# Run DID model
did_model <- feols(reading_score ~ time_binary + clsize_new + time_binary:clsize_new, data = df_q8)

# Extract results and format output
did_results <- tidy(did_model)
kable(did_results, format = "html", digits = 3, caption = "Basic DID Regression Results") %>%
  kable_styling(full_width = F, bootstrap_options = c("striped", "hover"))
```

Basic DID Regression Results

term	estimate	std.error	statistic	p.value
(Intercept)	34.061	0.128	265.530	0.000
time_binary	22.758	0.181	125.451	0.000
clsize_new	-0.711	0.208	-3.425	0.001
time_binary:clsize_new	0.590	0.294	2.010	0.044

9. Conclusion

PSM, IPTW, and MMWS are observational research methods based on propensity scores that require no unobserved confounding between the treatment and the outcome after controlling for the observed covariates. PSM achieves group comparability through matching, IPTW retains a larger sample by using weighting, and MMWS combines the advantages of stratification and weighting. The IV (instrumental variable) method primarily relies on the assumptions of instrument exogeneity and relevance, allowing it to identify the treatment effect even when unobserved confounding is present, but it requires finding a suitable and valid instrument. DID (difference-in-differences) assumes that, in the absence of the intervention, the treatment and control groups would have followed the same trend, thereby identifying causal effects by comparing the differences before and after the intervention.

Each of these methods has its own strengths. Propensity score–based methods require the strong ignorability assumption. PSM is simple and intuitive, making it easy to understand, and matching ensures that the baseline characteristics between groups are well balanced. IPTW does not require discarding observations, making it suitable for large sample sizes and allowing for the estimation of the population average treatment effect. MMWS stratifies the sample by propensity score and then applies weighting, enabling a more flexible examination of heterogeneous effects within each stratum. The IV method can identify causal effects even in the presence of severe endogeneity and when it is difficult to observe all confounding factors, providing a powerful tool for addressing endogeneity issues common in educational research. DID leverages the time dimension before and after a policy or intervention to control for potential confounders that remain constant over time, and as long as the parallel trends assumption holds, it can eliminate interference from many unobserved individual characteristics.

In terms of results, the estimates from PSM, IPTW, and MMWS are relatively close (around 0.045 to 0.05), all indicating that a reduction in class size has a clear positive effect on reading achievement, with effect sizes in a similar range. The regression coefficient from the IV method may differ slightly in magnitude or sign compared to the other methods, primarily because the enrollment rate is a weak instrument. Nevertheless, the overall direction still supports the positive impact of small class teaching on reading achievement. Similarly, the DID results show that, after the intervention, the small class group experienced an additional significant gain in reading scores compared to the large class group, which is consistent with the conclusions drawn from the other methods.

In summary, these five methods—each based on different assumptions and estimation frameworks—have all yielded similar conclusions that small class teaching has a positive effect on first-grade students' reading achievement, demonstrating that this finding is both robust and consistent. Based on these research results, it can be preliminarily concluded that reducing class size is an educational intervention that positively enhances early reading skills. Future research could further validate its external validity across a broader scope or additional subjects and rigorously test the key assumptions of each method, thereby providing more comprehensive evidence for educational policy-making.

```
df <- data.frame(
  Method    = c("PSM", "IPTW", "MMWS", "IV", "DID"),
  Estimate  = c(0.5984877, 0.6194184, 0.6600558, 0.329, 0.590),
  Std.Error = c(0.1500921, 0.2283129, 12.230558, 0.104, 0.294),
  p.value   = c(6.712134e-05, 0.0066754, 3.41e-05, 0.002, 0.044)
)

kable(
  df,
  caption = "Estimation Results for Class Size on Grade 1 Reading Achievement",
  digits = 4
)
```

Estimation Results for Class Size on Grade 1 Reading Achievement

Method	Estimate	Std.Error	p.value
PSM	0.5985	0.1501	0.0001
IPTW	0.6194	0.2283	0.0067
MMWS	0.6601	12.2306	0.0000
IV	0.3290	0.1040	0.0020
DID	0.5900	0.2940	0.0440