lab4_2024_v1

February 20, 2024

1 Lab 4: The Tennessee STAR Experiment

1.1 Methods/concepts: treatment effect estimation in stratified experiments, bar graphs, multivariable regression, statistical inference, statistical vs. practical significance

Name: Shreya Chaturvedi

Email: shreyachaturvedi@hks.harvard.edu

HUID: 31575036

Lab: Thursday 3pm at HKS

Date: February 22nd, 2024

LAB DESCRIPTION

The Tennessee Student/Teacher Achievement Ratio (STAR) Experiment was implemented in 1985-1986 in 79 schools, involving more than 11,600 students. Both students and teachers were randomly assigned to small and regular size classes starting in kindergarten.

In this lab, you will measure the causal effect of class size on student achievement in kindergarten, as measured by year-end test scores for $N = 5,710$ kindergarten children. For more details on the variables included in these data, see Table 1. A list and description of each of the R commands needed for this lab are contained in Table 2. For more background on the experiment, see Krueger (1999) or Chetty et al. (2011).

1.2 QUESTIONS

1. In the Tennessee STAR Experiment, *both* students and teachers were randomly assigned to small and large classes. Explain briefly why it is important to randomly assign not just students but also teachers in order to determine the causal effect of class size.

```
[1]: # QUESTION 1 Code
```

Question 1 Answer

The random assignment of both students and teachers in the STAR experiment was key to establishing the causal effect of class size on student achievement. It ensured that any differences in outcomes can be attributed to class size rather than other factors, such as teacher quality or teaching style. By randomizing both student and teacher assignments, the experiment controls for potential biases and confounding variables, enhancing the internal validity of the results and allowing for a more accurate and generalizable understanding of how class size affects educational outcomes.

2. Using the **star.dta** file, how does average class size (*class_size*) compare in small kindergarten classes vs. regular kindergarten classes (small == 1 vs. small == 0)?

[2]:

```
#clear the workspace
rm(list=ls()) # removes all objects from the environment
#Install and load haven package
if (!require(haven)) install.packages("haven");
library(haven)
library(dplyr)
library(tidyr)
library(knitr)
library(ggplot2)
options(warn = -1)

#Load stata data set
download.file("https://raw.githubusercontent.com/ekassos/ec50_s24/main/star.
 ↪dta", "star.dta", mode = "wb")
star <- read_dta("star.dta")
#head(star)
# QUESTION 2 Code
with(subset(star, small == 1), mean(class_size, na.rm=TRUE)) #Mean when small␣
 ↪is 1
with(subset(star, small == 0), mean(class_size, na.rm=TRUE)) #Mean when small␣
 ↪is 0

#table(star$small)
```

```
Loading required package: haven


Attaching package: ‘dplyr’


The following objects are masked from ‘package:stats’:

    filter, lag


The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

15.0957632037145

22.5232004013042

Question 2 Answer

As shown above, the average class sizes in small kindergarten classes (small == 1) is approximately 15 whereas the average class sizes in large kindergarten classes (small ==0) is approximately 22.5

3. At the end of kindergarten school year, students took four Stanford Achievement Tests: Math-SAT *math*, Reading-SAT *read*, Word-SAT *wordskill*, and Listening-SAT *listen*. It is common in education research to convert test scores into more meaningful units. One way is to generate a new variable *sat_index* that combines the exam scores into one overall metric measured in “standard deviation units” (or σ’s in the lingo of education researchers) as follows:[1]

    1. For each of the four exam scores, subtract the *control group mean* and divide by the *control group standard deviation* to define four “standardized” exam scores. Some pseudo code is: *standardized math score = (math score − control_mean(math score))* ÷ *control_sd(math score)*, where *control_mean(math score)* and *control_sd(math score)* are calculated for observations with small == 0.

        Report summary statistics (mean, standard deviation, minimum, and maximum) for the four new variables that you generated.

    2. Then generate *sat_index* as the average of these four standardized exam scores. Some pseudo code is: *sat_index* = mean(*standardized math score, standardized reading score, standardized word score, standardized listening score).* Report summary statistics (mean, standard deviation, minimum, and maximum).

    3. Plot a histogram of *sat_index* for small kindergarten classes (small == 1) and for regular kindergarten classes (small == 0). Include an image of your histogram in your solutions. What do you notice in the histograms?

1. [^](#cite_ref-1) For example, this method was used to study multiple outcomes in the Moving to Opportunity Experiment by Larry Katz and co-authors.

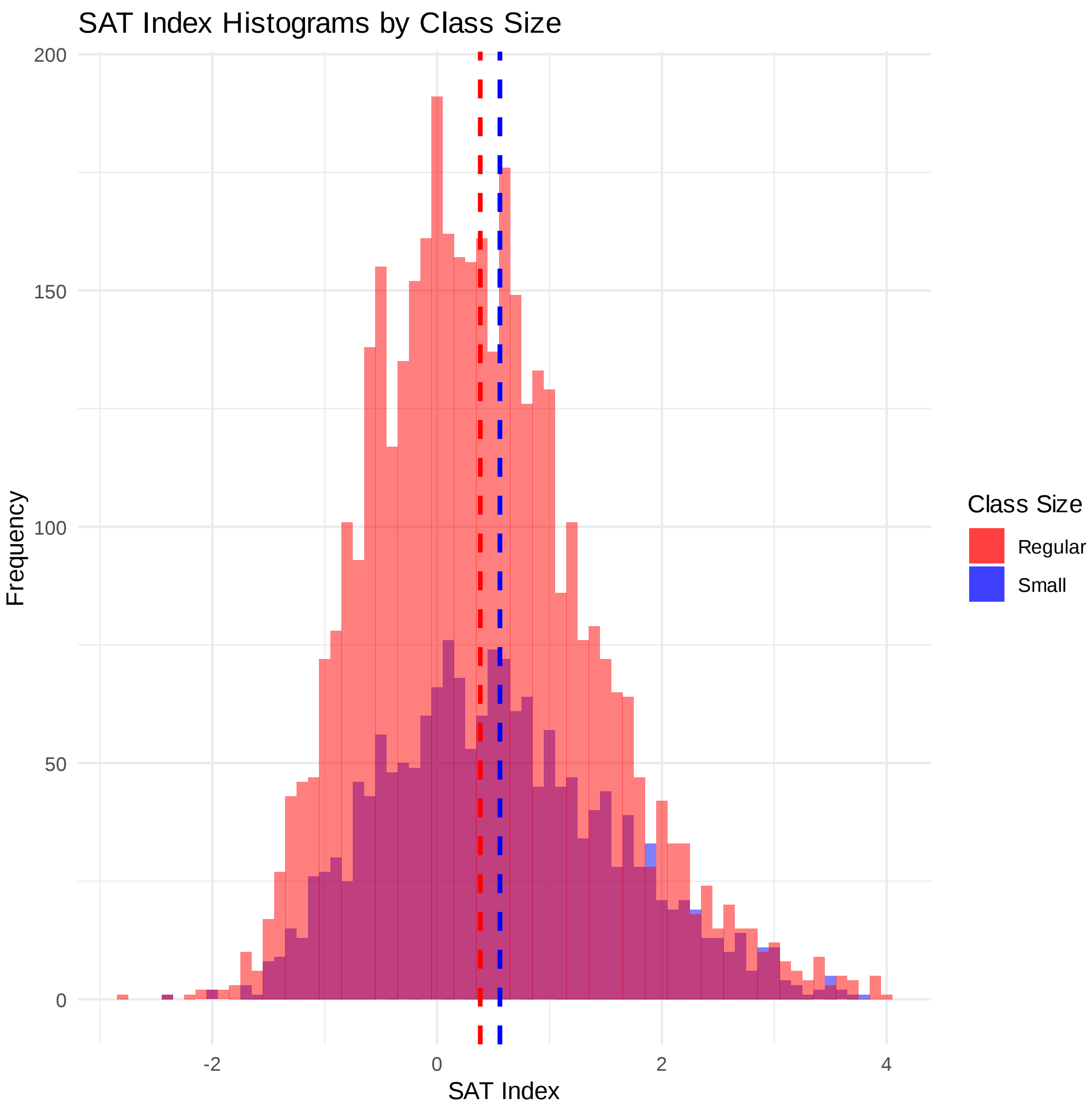
[3]:

```
# QUESTION 3 Code
##### Part A
#Subset data frame to control group
control_group <- subset(star, small == 0)

#Store mean and standard deviation of each of the four test scores
#Standardizing scores in the next step
math_control_mean <- mean(control_group$math, na.rm = T)
math_control_sd <- sd(control_group$math, na.rm = T)
star$math_std <- (star$math - math_control_mean) / math_control_sd

read_control_mean <- mean(control_group$read, na.rm = T)
read_control_sd <- sd(control_group$read, na.rm = T)
star$read_std <- (star$math - read_control_mean) / read_control_sd

wordskill_control_mean <- mean(control_group$wordskill, na.rm = T)
wordskill_control_sd <- sd(control_group$wordskill, na.rm = T)
```

```
star$wordskill_std <- (star$wordskill - wordskill_control_mean) /␣
 ↪wordskill_control_sd

listen_control_mean <- mean(control_group$listen, na.rm = T)
listen_control_sd <- sd(control_group$listen, na.rm = T)
star$listen_std <- (star$listen - listen_control_mean) / listen_control_sd

#Code to create a nice table with all the stats in one place
summary_stats <- star %>%
  summarise(
    Math_Mean = mean(math_std, na.rm = TRUE),
    Math_SD = sd(math_std, na.rm = TRUE),
    Math_Min = min(math_std, na.rm = TRUE),
    Math_Max = max(math_std, na.rm = TRUE),
    Read_Mean = mean(read_std, na.rm = TRUE),
    Read_SD = sd(read_std, na.rm = TRUE),
    Read_Min = min(read_std, na.rm = TRUE),
    Read_Max = max(read_std, na.rm = TRUE),
    Wordskill_Mean = mean(wordskill_std, na.rm = TRUE),
    Wordskill_SD = sd(wordskill_std, na.rm = TRUE),
    Wordskill_Min = min(wordskill_std, na.rm = TRUE),
    Wordskill_Max = max(wordskill_std, na.rm = TRUE),
    Listen_Mean = mean(listen_std, na.rm = TRUE),
    Listen_SD = sd(listen_std, na.rm = TRUE),
    Listen_Min = min(listen_std, na.rm = TRUE),
    Listen_Max = max(listen_std, na.rm = TRUE)
  ) %>%
  pivot_longer(everything(), names_to = "Statistic", values_to = "Value") %>%
  separate(Statistic, into = c("Variable", "Metric"), sep = "_") %>%
  pivot_wider(names_from = "Metric", values_from = "Value")

print(summary_stats)

##### Part B
#Calculating a composite sat index and then creating summary stats
star$sat_index <- rowMeans(cbind(star$math_std, star$read_std,␣
 ↪star$wordskill_std, star$listen_std), na.rm = TRUE)
sat_index_mean <- mean(star$sat_index, na.rm = TRUE)
sat_index_sd <- sd(star$sat_index, na.rm = TRUE)
sat_index_min <- min(star$sat_index, na.rm = TRUE)
sat_index_max <- max(star$sat_index, na.rm = TRUE)
print("Sat Index Summary Statistics")
cat("Mean: ", sat_index_mean, "\n")
cat("Standard Deviation: ", sat_index_sd, "\n")
cat("Minimum: ", sat_index_min, "\n")
cat("Maximum: ", sat_index_max, "\n")
```

```
##### Part C
# Calculate mean sat_index for small and regular classes
mean_small <- mean(subset(star, small == 1)$sat_index, na.rm = TRUE)
mean_regular <- mean(subset(star, small == 0)$sat_index, na.rm = TRUE)

# Plot histogram of sat_index for small and regular classes with mean lines
ggplot(star, aes(x = sat_index, fill = factor(small))) +
  geom_histogram(data = subset(star, small == 1), binwidth = 0.1, alpha = 0.5,␣
 ↪position = "identity") +
  geom_histogram(data = subset(star, small == 0), binwidth = 0.1, alpha = 0.5,␣
 ↪position = "identity") +
  geom_vline(xintercept = mean_small, color = "blue", linetype = "dashed", size␣
 ↪= 1) +
  geom_vline(xintercept = mean_regular, color = "red", linetype = "dashed",␣
 ↪size = 1) +
  labs(title = "SAT Index Histograms by Class Size",
       x = "SAT Index",
       y = "Frequency",
       fill = "Class Size") +
  scale_fill_manual(values = c("0" = "red", "1" = "blue"), labels =␣
 ↪c("Regular", "Small")) +
  theme_minimal()
```

```
# A tibble: 4 × 5
  Variable    Mean    SD   Min   Max
  <chr>      <dbl>
<dbl> <dbl> <dbl>
1 Math      0.0519  1.02 -3.50
3.05
2 Read      1.61    1.53 -3.68  6.10
3 Wordskill 0.0477  1.01 -2.80
4.40
4 Listen    0.0337  1.00 -4.22
4.07
[1] "Sat Index Summary Statistics"
Mean:  0.4368537
Standard Deviation:  1.007125
Minimum:  -2.755427
Maximum:  4.034547
```



Question 3 Answer

Some key observations from the two histograms show that: Clearly, the regular class size had many more observations than the small class size (as indicated by the taller histogram). The mean for regular class size is slightly lower than the small class size (as can be seen by the histogram appearing to be slightly shifted).

4. Returning to question 1, we will assess whether the data are consistent with *teachers* having been randomly assigned to classrooms by testing for balance of teacher characteristics. The STAR experiment consisted of 324 teachers, but there are 5,710 students in these data. We will conduct this and all of our subsequent analyses in this lab at the teacher-level, rather than at the student-level.

    1. Aggregate the data by *teacher_id*, so that you end up with a 324 observation data set with information on *small*, *school_id*, *teacher_id*, *teacher_masters*, *teacher_white*,

teacher_black, *teacher_experience* as well as the mean of *sat_index* and *class_size* across all the students in the teacher's class (which we'll use in question 5). Report means for all the variables in the resulting data set separately for small and large classes.

2. Estimate a linear regression (lm in R or regress in stata) of *teacher_experience* on an intercept and *small*. Use the estimated coefficient on small to report the difference in average teacher experience in small vs. large classes. Calculate a 95% confidence interval for this difference: Regression coefficient on *small* ± 1.96 × standard error.
3. Repeat question b for *teacher_masters*, *teacher_white*, and *teacher_black*.
4. Are the differences in teacher characteristics in small vs. large classes *statistically significantly different from zero*? Are they practically significant? What do you conclude about whether the random assignment was successful in balancing teacher characteristics?

[4]:

```
# QUESTION 4 Code
### PART A
# Aggregating data by teacher_id
teacher_aggregated_data <- star %>%
  group_by(teacher_id, small, school_id, teacher_masters, teacher_white, teacher_black, teacher_experience) %>%
  summarise(
    mean_sat_index = mean(sat_index, na.rm = TRUE),
    mean_class_size = mean(class_size, na.rm = TRUE),
    .groups = "drop"
    )
means_for_small_classes <- teacher_aggregated_data %>%
  filter(small == 1) %>%
  summarise(across(everything(), mean, na.rm = TRUE))
print(means_for_small_classes)

means_for_large_classes <- teacher_aggregated_data %>%
  filter(small == 0) %>%
  summarise(across(everything(), mean, na.rm = TRUE))
print(means_for_large_classes)

##### PART B
#Running a regression for teacher experience
reg_experience <- lm(teacher_experience ~ small, data = teacher_aggregated_data)
summary(reg_experience)
confint(reg_experience, level = 0.95)

##### PART C
#Running a regression for teacher masters
reg_masters <- lm(teacher_masters ~ small, data = teacher_aggregated_data)
summary(reg_masters)
confint(reg_masters, level = 0.95)

#Running a regression for teacher white
reg_white <- lm(teacher_white ~ small, data = teacher_aggregated_data)
```

```r
summary(reg_white)
confint(reg_white, level = 0.95)

#Running a regression for teacher black
reg_black <- lm(teacher_black ~ small, data = teacher_aggregated_data)
summary(reg_black)
confint(reg_black, level = 0.95)
```

```
# A tibble: 1 × 9
  teacher_id small school_id teacher_masters teacher_white teacher_black
       <dbl> <dbl>
<dbl>           <dbl>
<dbl>         <dbl>
1  21003518.     1
210035.            0.323         0.866         0.134
#   3 more variables: teacher_experience <dbl>, mean_sat_index <dbl>,
#   mean_class_size <dbl>
# A tibble: 1 × 9
  teacher_id small school_id teacher_masters teacher_white teacher_black
       <dbl> <dbl>
<dbl>           <dbl>
<dbl>         <dbl>
1  21145023.     0
211450.            0.365         0.822         0.173
#   3 more variables: teacher_experience <dbl>, mean_sat_index <dbl>,
#   mean_class_size <dbl>


Call:
lm(formula = teacher_experience ~ small, data = teacher_aggregated_data)

Residuals:
    Min      1Q  Median      3Q     Max
-9.3807 -4.3807 -0.3807  3.6193 17.9764

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.3807     0.4152  22.591   <2e-16 ***
small        -0.3571     0.6632  -0.538    0.591
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.828 on 322 degrees of freedom
Multiple R-squared:  0.0008994,	Adjusted R-squared:  -0.002203
F-statistic: 0.2899 on 1 and 322 DF,  p-value: 0.5907
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	8.563792	10.1976297
small	-1.661906	0.9477286

```
Call:
lm(formula = teacher_masters ~ small, data = teacher_aggregated_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.3655 -0.3655 -0.3228  0.6345  0.6772 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.36548    0.03403  10.741   <2e-16 ***
small       -0.04265    0.05435  -0.785    0.433    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4776 on 322 degrees of freedom
Multiple R-squared:  0.001909,	Adjusted R-squared:  -0.001191 
F-statistic: 0.6157 on 1 and 322 DF,  p-value: 0.4332
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	0.2985376	0.43242684
small	-0.1495743	0.06427913

```
Call:
lm(formula = teacher_white ~ small, data = teacher_aggregated_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.8661  0.1339  0.1558  0.1777  0.1777 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.82234    0.02619  31.400   <2e-16 ***
small        0.04381    0.04183   1.047    0.296    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3676 on 322 degrees of freedom
Multiple R-squared:  0.003395,	Adjusted R-squared:  0.0002995 
F-statistic: 1.097 on 1 and 322 DF,  p-value: 0.2958
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	0.77081243	0.8738576
small	-0.03848733	0.1261007

```
Call:
lm(formula = teacher_black ~ small, data = teacher_aggregated_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.1726 -0.1726 -0.1726 -0.1339  0.8661 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.17259    0.02599   6.640 1.33e-10 ***
small       -0.03873    0.04152  -0.933    0.352    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3648 on 322 degrees of freedom
Multiple R-squared:  0.002696,	Adjusted R-squared:  -0.0004017 
F-statistic: 0.8703 on 1 and 322 DF,  p-value: 0.3516
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	0.1214524	0.22372525
small	-0.1204078	0.04294665

Question 4 Answer

Summarizing the regression outputs here, we see the following: 1. Teacher Experience: Coefficient: -0.357 Standard Error: 0.6632 P-Value: 0.591 2. Teacher Masters: Coefficient: -0.042 Standard Error: 0.05435 P-Value: 0.433 3. Teacher White: Coefficient: 0.043 Standard Error: 0.0418 P-Value: 0.296 5. Teacher Black: Coefficient: -0.0387 Standard Error:0.4152 P-Value: 0.352

A negative coefficient means that a lower value of the variable is associated with a higher value of small. However, since none of the p-values for these coefficients are below the conventional significance threshold of 0.05, we can conclude that there are no statistically significant differences in these teacher characteristics between small and large classes. Additionally, the coefficients themselves are small in magnitude, suggesting that even if there were statistical significance, the practical or real-world significance of these differences might be minimal.

5. The STAR experiment was a *stratified randomized experiment*, also known as a *randomized block experiment*, because students were randomly assigned to classes at their own school. The *strata* were therefore the school. Intuitively, students could only be randomly assigned to a class at their school and not for example a school across town. The practical implication is that it was as-if each of the 79 schools conducted their own separate experiment.

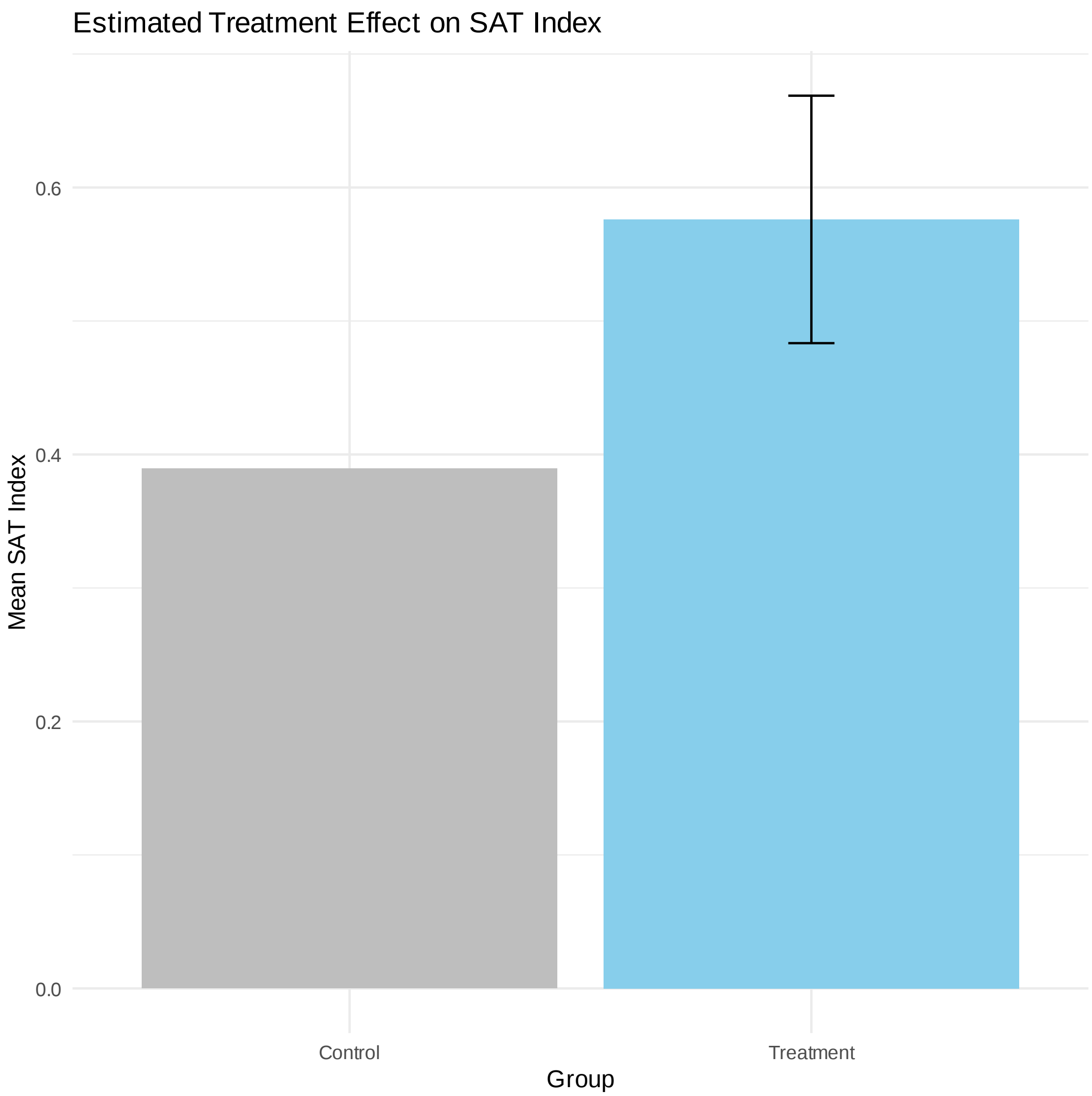
   The most standard approach to obtain one overall estimate is to modify the regressions we ran in Lab 3 by adding indicator variables for each school as additional control variables. This is now a *multivariable regression.* Recall that we only care about the regression coefficient on the variable *small*, and can safely ignore the 79 other estimated coefficients.

1. Using the teacher-level data with 324 observations, run a multivariable regression of *sat_index* on the small class indicator *small*, controlling for school fixed effects (e.g., regress with i.school_id in Stata; or lm with factor(school_id) in R).
2. Use the estimated coefficient on the small class indicator *small* to report an estimate of the causal effect of the experiment.[2] Calculate a 95% confidence interval for this causal effect: Regression coefficient on *small* ± 1.96 × standard error.
3. Visualize the estimated treatment effect using a bar graph, with one bar representing the control group and a second bar representing the treatment group. The height of the bar for the control group should equal the control group mean of *sat_index*. The height of the bar for the treatment group should equal the sum of the control group mean and regression coefficient on *small* from the regression in part a. Add square brackets to the treatment group bar to visualize the 95% confidence interval from part b.

2. [^](#cite_ref-2) For example, this method was used to study multiple outcomes in the Moving to Opportunity Experiment by Larry Katz and co-authors.

[5]:
```
# QUESTION 5 Code
##### PART A and B
#Running a regression with school_id fixed effects
reg_school_fe <- lm(mean_sat_index ~ small + factor(school_id), data =␣
 ↪teacher_aggregated_data)
summary_reg_school_fe <- summary(reg_school_fe)
#summary(reg_school_fe)
cat("Coefficients (Intercept and 'small'):\n")
print(summary_reg_school_fe$coefficients[1:2,], digits = 3)
print(confint(reg_school_fe)["small", ])

##### PART C
# Extracting relevant values for the graph
control_mean_sat_index <-␣
 ↪mean(teacher_aggregated_data$mean_sat_index[teacher_aggregated_data$small ==␣
 ↪0], na.rm = TRUE)
#print(control_mean_sat_index)
coef_treatment <- summary(reg_school_fe)$coefficients["small", "Estimate"]
#print(coef_treatment)
se_treatment <- summary(reg_school_fe)$coefficients["small", "Std. Error"]
ci_95 <- 1.96 * se_treatment  # for 95% CI
print(ci_95)
print(control_mean_sat_index + coef_treatment + ci_95)

# Create a data frame for plotting by using the values extracted above
plot_data <- data.frame(
  group = c("Control", "Treatment"),
  mean_sat_index = c(control_mean_sat_index, control_mean_sat_index +␣
 ↪coef_treatment),
  ci_low = c(NA, control_mean_sat_index + coef_treatment - ci_95),
  ci_high = c(NA, control_mean_sat_index + coef_treatment + ci_95)
)

# Create the bar graph
ggplot(plot_data, aes(x = group, y = mean_sat_index)) +
  geom_bar(stat = "identity", position = "dodge", fill = c("grey", "skyblue")) +
  geom_errorbar(aes(ymin = ci_low, ymax = ci_high), width = .1) +
  labs(title = "Estimated Treatment Effect on SAT Index",
       x = "Group",
       y = "Mean SAT Index") +
  theme_minimal()
```

```
Coefficients (Intercept and 'small'):
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.519     0.2371   -2.19 0.029419
small          0.187     0.0473    3.95 0.000104
     2.5 %     97.5 %
0.09346679 0.27977202
[1] 0.09269228
[1] 0.6687673
```



Question 5 Answer

(Answer here; include your images if needed.)

Graph shown above.

6. Create an annotated/commented do-file, .ipynb Jupyter Notebook, or .R file that can replicate all your analyses above. This will be the final code that you submit on Gradescope. The motivation for using do-files and .R files is described on page 4, which has been adapted from training materials used by Innovations for Poverty Action (IPA) and the Abdul Latif Jameel Poverty Action Lab (J-PAL).

Final Submission Checklist for Lab 4

If you're working with R

If you're working with Stata

Lab 4 Write-Up:

PDF of your answers. For graphs, you must save them as images (e.g., .png files) and insert them into the document.

Lab 4 Code:

.R script file, well-annotated replicating all your analyses;OR

.ipynb file

Lab 4 Write-Up:

PDF of your answers. For graphs, you must save them as images (e.g., .png files) and insert them into the document.

Lab 4 Code:

do-file, well-annotated replicating all your analyses;AND

log-file, not a .smcl file, with the log showing the output generated by your final do-file.

If you're working with an .ipynb notebook

It is likely that your .ipynb file will be greater than 1 MB in size. Therefore, for this assignment please submit both your *well-annotated* **.ipynb file** and **a .PDF version of this file**. The notebook should replicate all your analyses for Lab 4 (with enough comments that a principal investigator on a research project would be able to follow and understand what each step of the code is doing).

1.3 How to submit your assignment

Step 1 Access the lab assignment under the "Assignments" tab on Canvas
Step 2 Access Gradescope from Canvas
Step 3 Access the lab assignment on Gradescope
Step 4 Upload your files *Check What files to submit to confirm what files you need to submit.*
Step 5 What you'll see after submitting your lab assignment
Step 6 Check your submitted files
Step 7 You'll receive an email confirmation as well

1.4 What files to submit

If you're using Python Notebook to write your R code, and a document editor to write your answers

If you're using a Python Notebook to write your R code AND to write your answers

1.5 WHAT ARE DO-FILES AND .R FILES AND WHY DO WE NEED ONE?

Let's imagine the following situation - you just found out you have to present your results to a partner– all the averages you produced and comparisons you made. Suppose you also found out that the data you had used to produce all these results was not completely clean, and have only just fixed it. You now have incorrect numbers and need to re-do everything.

How would you go about it? Would you reproduce everything you did for Lab 1 from scratch? Can you do it? How long would it take you to do? Just re-typing all those commands into Stata or R in order and checking them would take an hour.

An important feature of any good research project is that the results should be reproducible. For Stata and R the easiest way to do this is to create a text file that lists all your commands in order, so anyone can re-run all your Stata or R work on a project anytime. Such text files that are produced within Stata or linked to Stata are called do-files, because they have an extension .do (like intro_exercise.do). Similarly, in R, these files are called .R files because they have an extension of .R. These files feed commands directly into Stata or R without you having to type or copy them into the command window.

An added bonus is that having do-files and .R files makes it very easy to fix your typos, re-order commands, and create more complicated chains of commands that wouldn't work otherwise. You can now quickly reproduce your work, correct it, adjust it, and build on it.

Finally, do-files and .R files make it possible for multiple people to work on a project, which is necessary for collaborating with others or when you hand off a project to someone else.

1.5.1 Figure 1Tennessee Student-Teacher Achievement Ratio (STAR) Experiment

Note: Image is from Mosteller (1995) in his review of the history of and results from the Tennessee STAR experiment.

1.6 DATA DESCRIPTION, FILE: star.dta

The data consist of $N = 5,710$ kindergarten children in the Tennessee Student/Teacher Achievement Ratio (STAR) Experiment. For more information about the STAR Experiment and these data, see Alan B. Krueger (1999) "Experimental Estimates of Education Production Functions," *Quarterly Journal of Economics* 114(2): 497-532; and Raj Chetty, John Friedman, Nathaniel Hilger, Emmanuel Saez, Diane Schanzenbach, and Danny Yagan (2011) "How Does Your Kindergarten Classroom Affect Your Earnings? Evidence from Project STAR," *Quarterly Journal of Economics* 126(4): 1593-1660. Various excellent textbooks also present analyses of the data from the STAR Experiment, including Stock and Watson (2019, Chapter 13), Angrist and Pischke (2009, Chapter 2), and Imbens and Rubin (2015, Chapter 9).

TABLE 1

Variable Definitions

	Variable	Label	Obs.	Mean	St. Dev.	Min	Max
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
1	*student_id*	Student id	5,710	n/a	n/a	n/a	n/a
2	*school_id*	Kindergarten school id	5,710	n/a	n/a	n/a	n/a
3	*teacher_id*	Kindergarten teacher id	5,710	n/a	n/a	n/a	n/a
4	*class_size*	Class size in kindergarten	5,710	20.28	3.966	12	28
5	*read*	Kindergarten reading SAT test score	5,710	436.9	31.76	358	627
6	*math*	Kindergarten math SAT test score	5,710	485.8	47.75	320	626
7	*listen*	Kindergarten listening SAT test score	5,710	537.6	33.14	397	671
8	*wordskill*	Kindergarten word study skills SAT score	5,710	434.5	36.84	331	593
9	*small*	Small classroom in kindergarten	5,710	0.302	0.459	0	1
10	*female*	Student is female	5,710	0.487	0.500	0	1
11	*freelunch*	Student receives Free or Reduced Price Lunch	5,710	0.480	0.500	0	1
12	*teacher_masters*	Kindergarten Teacher has a Master's Degree	5,710	0.354	0.478	0	1
13	*teacher_white*	Kindergarten Teacher is White	5,710	0.839	0.368	0	1
14	*teacher_black*	Kindergarten Teacher is Black	5,710	0.158	0.364	0	1
15	*teacher_experience*	Kindergarten Teacher's Years of Experience	5,710	9.326	5.762	0	27

Note: Table describes variables in star.dta.

1.7 TABLE 2: R Commands

R command

Description

```
#clear the workspace
rm(list=ls()) # removes all objects from the environment

#Install and load haven package
if (!require(haven)) install.packages("haven"); library(haven)

#Load stata data set
download.file("https://raw.githubusercontent.com/ekassos/ec50_s24/main/star.dta", "star.dta", m
star <- read_dta("star.dta")
```

This sequence of commands shows how to open Stata datasets in R. The first block of code clears the work space. The second block of code installs and loads the "haven" package. The third block of code downloads and loads in star.dta.

The summary command will report information on what is included in the data set loaded into memory, including information on the number of missing observations NAs for each variable.

```
#Code to report basic summary statistics for a variable
summary(star$yvar)

#Get standard deviations too
sd(star$test_score_index)
```

This code shows how to basic summary statistics such as mean, minimum, maximum, and number of NAs for variable yvar. The second line shows how to get the standard deviation, too.

```
#Summary stats for one variable
mean(star$yvar, na.rm=TRUE)

#Summary stats for observations with treatment_group == 1
#Subset data
new_df <- subset(star, treatment_group == 1)

#Report mean
mean(new_df$yvar, na.rm=TRUE)

#Alternatively, do it all at once using the with() function
with(subset(star, treatment_group == 1), mean(yvar, na.rm=TRUE))

#Summary stats for observations with treatment_group == 0
with(subset(star, treatment_group == 0), mean(yvar, na.rm=TRUE))

#Alternatively, get both means using tapply()
tapply(star$yvar, star$treatment_group, mean)

#Alternatively, get both means using by()
by(star$yvar, list(star$treatment_group), mean)
```

We used these commands in previous labs. These commands report means for yvar. The first line calculates these statistics across the full sample. The other lines illustrate how to calculate these statistics for observations meeting certain criteria: when another variable in the data is equal to 1, or equal to 0. The first few examples use the subset() function to pick out only the observations in a data frame that meet certain criteria. We can combine this with the with() function. We also have seen how to use the tapply() function to report the mean of yvar grouped by another variable treatment_group. We can also use the by() function to do the same thing.

```
#Code to generate standardized version of variable

#Subset data frame to control group
cntrl <- subset(star, small == 0)

#Store mean and standard deviation of yvar
yvar_cntrl_mean <- mean(cntrl$yvar, na.rm = T)
yvar_cntrl_sd <- sd(cntrl$yvar, na.rm = T)

#Generate standardized version of yvar and add to original df
star$yvar_std <- (star$yvar - yvar_cntrl_mean) / yvar_cntrl_sd
```

These commands show how to generate a new variable that equals yvar minus the control group mean and divided by the control group standard deviation. I start by subsetting the data frame to just the control units. Then I store the mean and standard deviation of yvar computed in this data frame. Finally, I generate a new variable yvar_std in the original data frame that equals yvar minus the control group mean, and divided by the control group standard deviation.

```
#Load tidyverse
if (!require(tidyverse)) install.packages("tidyverse"); library(tidyverse)

#Draw histograms for two groups
ggplot(star, aes(x=yvar,
                 fill=factor(small, labels=c("Large", "Small")),
                 y=..density..)) +
  geom_histogram(alpha=0.2, position="identity") +
  labs(x = "End-of-Year KG Test Index", fill = "Class Size")

#Save the graph
ggsave("histogram.png")
```

These commands show how to draw histograms for different groups on the same axes. I start by loading the tidyverse library. Then I use ggplot with geom_histogram() as in Lab 1. To get two histograms on the same axes, I specify certain options in the the aes() part of the main ggplot() part of the code. I tell it to plot a histogram of the variable yvar (x=yvar) and to do it on the density scale (y=..density..). To plot two overlapping histograms, I specify fill = factor(small). The factor() part of this code tells ggplot that the groups are defined by whether the variable small equals 1 or 0; otherwise it will treat small as a continuous variable.
I also include, labels=c(“Large”, “Small”) so that the graph will be labelled with Large and Small rather than just 0 and 1.
In the geom_hist() part of the command, I specify the option alpha=0.2 to refer to the opacity of the bars, allowing them to be partially see through. Values of alpha range from 0 to 1, with lower values corresponding to more transparent colors. I also specify the position=“identity” option to get both histograms on the same axes. Finally, the labs() in the last line specifies the x-axis label and a label for the legend (the fill part).

```
#Load tidyverse
if (!require(tidyverse)) install.packages("tidyverse"); library(tidyverse)

#Create grouped table by_class
by_class <- group_by(star,
                     teacher_id,
                     school_id,
                     small,
                     teacher_masters,
                     teacher_white,
                     teacher_black,
                     teacher_experience)

#Create new data frame called classes
classes <- summarise(by_class,
                    yvar = mean(yvar, na.rm = TRUE),
                    xvar = mean(xvar, na.rm = TRUE))

#Describe new data frame that we have created
summary(classes)
```

These commands show how to convert the data from student-level data to teacher-level data. We start by loading the tidyverse library. Then we use group_by() to create a new grouped table called by_class. This function takes an existing tbl and converts it into a grouped tbl where operations are performed "by group." The first argument of group_by() is the data frame to be grouped. The other part of the code specifies the grouping is by teacher_id. I also list various other variables that are always constant for all students taught by the same teacher (experience, race, education, small vs. large class, and school). These variables will be included as variables in the collapsed data frame. Then we use summarise() function to define a new data frame with the mean of variables called yvar and xvar grouped as specified by the by_class grouped table we created earlier.

```
#Report summary statistics split by different groups

#Various ways to do this.  First tapply()
tapply(classes$yvar, classes$treatment_group, mean)
tapply(classes$yvar, classes$treatment_group, sd)

#Alternatively, by()
by(classes$yvar, list(classes$treatment_group), mean)
by(classes$yvar, list(classes$treatment_group), sd)

#Third - Tidyverse summarise_all()
classes %>% group_by(treatment_group) %>% summarise_all("mean")
classes %>% group_by(treatment_group) %>% summarise_all("sd")

#To report all variables, add this line before running:
options(dplyr.width = Inf)
classes %>% group_by(treatment_group) %>% summarise_all("mean")
classes %>% group_by(treatment_group) %>% summarise_all("sd")
```

We used these commands in Lab 1. These commands shows how to report summary statistics separately by groups defined by another variable, enabling for example summary statistics to be computed separately for a treatment group and a control group. The first example uses the tapply() function to report the mean and standard deviation of yvar grouped by another variable treatment_group. The second example uses the by() function to do the same thing.
The third example uses a combination of commands from the tidyverse library to report the means and standard deviations for all the variables in the data frame all at once with summarise_all() . By default, only the first several variables will be displayed. The options(dplyr.width = Inf) line will change the default to show summary statistics for all the variables

```
#Load packages
if (!require(sandwich)) install.packages("sandwich"); library(sandwich)
if (!require(lmtest)) install.packages("lmtest"); library(lmtest)

#Estimate linear regression
```

```
mod1 <- lm(yvar ~ treatment_group, data=classes)

#Report coefficients and standard errors
coeftest(mod1, vcov = vcovHC(mod1, type="HC1"))

#Add school fixed effects
mod2 <- lm(yvar ~ treatment_group + factor(school_id), data= classes)

#Report coefficients and standard errors
coeftest(mod2, vcov = vcovHC(mod2, type="HC1"))
```

These commands report estimated regression coefficients from a regression of yvar on an intercept and a variable treatment_group. The sandwich and lmtest packages are used to report standard errors that allow unequal variances in the two groups via the option type="HC1".

The second block reports estimated regression coefficients from a regression of yvar on an intercept, a variable treatment_group, and school fixed effects. The factor(school_id) creates separate indicator variables for each school identifier.

```
#Bar graph
#Load tidyverse library
if (!require(tidyverse)) install.packages("tidyverse"); library(tidyverse)

#Create a data frame with three columns
#Column 1 is the height of the two bars (in blue)
#Column 2 is the standard error (in purple)
#Column 3 is the group names (in red)
df <- data.frame(c(0.001, 0.4),
c(NA, 0.5),
c("Control group", "Treatment group"))

# Change name of 1st column of df to "Moved"
names(df)[1] <- "Moved"

# Change name of 2nd column of df to "se"
names(df)[2] <- "se"

# Change name of 3rd column of df to "Group"
names(df)[3] <- "Group"

#Add upper bound on 95% CI
df$ub <- df$Moved + 1.96*df$se

#Add lower bound on 95% CI
df$lb <- df$Moved - 1.96*df$se

# Bar graph displaying results
ggplot(data=df, aes(x=Group, y=Moved)) +
  geom_bar(stat="identity", fill="navy") +
  geom_errorbar(aes(ymin=lb, ymax=ub), width=.1, color="red") +
  labs(y = "Moved Using Experimental Voucher")

ggsave("fig1_test.png")
```

These commands show how to draw an Opportunity Insights style bar graph as in Lab 3, but with the addition of 95% confidence bars for the bar corresponding to the treatment group. The new part is in purple. We use geom_errorbar() in the ggplot line to create the bracket showing the 95% confidence interval.