ECON 773: Assignment 1

Tadhg Taylor-McGreal (400330297), Stella Till (Student number), Jeneta Lutijic(?) (student Number)

2026-01-01

Table of contents

Preface

Goal

The goals of this assignment are to:

- make sure your `R`, `Positron` are set up properly
- learn to install packages in `R`
- learn to edit text in `quarto`
- learn to load data in `R`
- learn to use 5 `dplyr` verbs for basic computations with data in `R`
- learn to create some visualizations using `ggplot`
- learn to render an `.qmd` to `.pdf` (via Typst)

Instructions

Before you start the assignment, make sure to `installRand Positron. Positron comes with command-line tools forquarto, which in turn includestypst`.

You will use `quarto` for generating your assignment output file. You begin with this script downloaded from A2L. Make sure that it is placed in the same folder as any data that came with it. Instructions for editing `quarto` are discussed in class, or see the Quarto website.

To submit this assignment:

- edit the `author` and `date` fields in the `YAML` (lines 1-9). Do not touch any other line in the YAML.
- complete the questions
- render to pdf
- email to TA

Some additional instructions:

1. leave all the text between `## Question` and `### Answer` unchanged and write your answers between `### Answer` and the next `## Question`
2. for each question that involves `R` code, **do not only write R code**, but add at least one sentence before the code explaining what you are going to do, and at least one line after the R code interpreting the result
3. check spelling before submissions
4. once your assignment is complete:
   - *Render* it to pdf
   - inspect the resulting `.pdf`: would you want to grade it?
   - submit

To render this document, click the *Render* button in the menu just above the top of this file. Alternatively, use the command palette. This step may fail until you install additional packages.

Job corps

Installing packages and loading data

This question will be demonstrated in class.

For this and the next few questions, we will use the data used in H3.1. To load the data, you first have to install the `R` package that accompanies the book. To install a package in `R`, which you need to do once for every `R` installation, run `install.packages(<PACKAGE_NAME>)` in your console. We will use data from the `causalweight` package. To install it, run:

```
install.packages("causalweight")
install.packages("causalweight", repos = "https://cloud.r-project.org")
```

The `#| eval: false` switch in the options of the above code chunk ensure that the code is not run whenever you render this `.qmd` file. Once per `R` session, and once in each `.qmd` file, you need to load the functionality of installed packages that you wish to use. You can do this by `library(<PACKAGE_NAME)`. In this case:

```
library(causalweight)
```

```
Loading required package: ranger
```

After a `library` command, the functions and data sets in a given package are available to you. To load the `JC` data from the `causalweight` package:

```
data(JC)
```

We will explore this data set using tools from the `tidyverse` library. If that collection of packages is not yet installed, run:

```
install.packages("tidyverse")
install.packages("gt")
```

Load all the functionality in the `tidyverse`:

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0
──
✔ dplyr     1.1.4     ✔ readr     2.1.6
✔ forcats   1.0.1     ✔ stringr   1.6.0
✔ ggplot2   4.0.1     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.2
```

```
✔ purrr     1.2.0
── Conflicts ─────────────────────────────────────── tidyverse_conflicts()
──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(gt)
```

We will discover how to use the `tidyverse` as we go. Have a look at this vignette for the ideas behind it, and at the book R4DS for a fantastic introduction to how to use it. In this assignment, we will focus on using

- `dplyr` for data manipulation, start your practice in this R4DS chapter
- `ggplot` for data visualization, see Grammar for Graphics.

We start by putting `JC` in `tibble` format, and by having a look at the top lines using `tinytable`.

```
JC <- as_tibble(JC)
JC |>
    select(1:6) |>
    head() |>
    gt()
```

assignment	female	age	white	black	hispanic
0	0	24	0	1	0
1	1	18	1	0	0
0	1	18	0	1	0
1	1	17	1	0	0
1	0	21	0	0	1
1	0	17	0	0	1

To bring up a description of the variables, ask:

```
?JC
```

Answer the following questions, using one `dplyr` verb each:

1. using `arrange`, sort the observations by age (first) and years of education at assignment (second)

```
JC <- JC |>
  dplyr::arrange(age, educ)
```

2. make a new, binarized, variable `educ_high` that equals `TRUE` if the years of education at assignment is 12 or greater, and `FALSE` otherwise. Use:
   - `mutate`
   - `ifelse`
   - the pipe operator `|>` to pass `JC` to `mutate`
   - `-> JC` to overwrite the original `tibble`

```
JC |>
  dplyr::mutate(educ_high = ifelse(educ >= 12, TRUE, FALSE)) -> JC
```

3. use `select` to keep only the 5 variables: `assignment`; the weekly earnings in fourth year after assignment; the variable indicating `female`; the education variable that you just created; the variable that indicates whether education is missing at assignment. Save the result in a new tibble called `JC_short`

```
JC_short <- JC |>
  dplyr::select(age, assignment, earny4, female, educ, educmis)
```

4. starting from `JC_short`, use `filter` to keep only the observations for which `assignment` equals 1 and save the results as `JC_short_TG`. Create an analogous `JC_short_CG`.

```
JC_short_TG <- JC_short |>
  dplyr::filter(assignment == 1)

JC_short_CG <- JC_short |>
  dplyr::filter(assignment == 0)
```

5. compute the mean weekly earnings in fourth year in the `JC_short_TG` tibble, and compare it to the analogous mean in `JC_short_CG`.

```
TG_mean <- JC_short_TG |>
  dplyr::summarise(mean_earny4 = mean(earny4, na.rm = TRUE))

CG_mean <- JC_short_CG |>
  dplyr::summarise(mean_earny4 = mean(earny4, na.rm = TRUE))

TG_mean
```

```
# A tibble: 1 × 1
  mean_earny4
```

```
        <dbl>
1        214.
```

```
CG_mean
```

```
# A tibble: 1 × 1
  mean_earny4
        <dbl>
1        198.
```

Interpret the final result, and compare it to the result on H, p. 21.

Answer

```
as.numeric(TG_mean$mean_earny4) - as.numeric(CG_mean$mean_earny4)
```

```
[1] 16.05513
```

…

Piping and `group_by`

This question will be demonstrated in class.

You will practice how to use a sequence of `pipes`, use `group_by`. You continue with the `JC_short` data that we created in the previous question.

First, use `group_by` to group the observations in `JC_short` by `female`, and comment on the result.

Second, pipe the result from `group_by` into a `summarize` command to see the means for each group in one table. Interpret the result.

Third, start with a `filter` command that keeps only those with education information available, then group by `assignment`, then compute the mean of `earny4` for each group. Interpret the result.

Fourth, repeat this, grouping by `female` **and** `assignment`. Interpret your result, discussing the conditional average treatment effect (CATE) for men and for women. Compare it to the unconditional ATE. Is the unconditional ATE an average of the two CATEs? Comment on this finding.

Answer

…

Weeks worked

You may be interested in the effect of the program on variables other than earnings. This question focuses on the proportion of weeks employed in fourth year after assignment.

First, modify the code in the previous question to answer this question using the `JC` data. This question is about the ATE, so do not split out by another variable. Interpret your findings.

Second, using `group_by` and `summarise` to split out results separately by `educ_high`. Here, `educ_high` plays the role of `female` in the first two parts of the previous question.

Finally, explore whether the effect differs depending on whether individuals have at least one child at assignment. Group only by "one child" variable, do not continue to condition on education.

Answer

...

Boxplots

This exercise will be demonstrated in class.

You will practice using `ggplot` to create data visualizations. Learning to work with `ggplot` could be its own course. In this course, it will be sufficient to modify the code discussed in class. For a deeper dive, start with these resources.

You will continue with the `JC` data. A useful data summary for our purpose is the boxplot.

First, make a boxplot of `earny4`. Second, make a boxplot of log earnings for those with positive earnings. Third, split out each of the two boxplots by `assignment`.

Answer

...

Scatterplots

You are going to use scatterplots to visualize the relationship between pre-program earnings, post-program earnings, and treatment assignment. This question will require you to figure out how `geom_point` works. Use the sources provided in the instructions.

First, create a scatterplot with average weekly gross earnings at assignment on the horizontal axis, and weekly earnings in fourth year after assignment on the vertical axis. Use only individuals that have positive earnings at both of those moments. Use log earnings in your plot. Hint: scatter plots use `geom_point` instead of `geom_boxplot`.

Second, add a least squares fit by using `geom_smooth`.

Third, split the results out by `assignment`, by setting `colour = assignment`. Interpret your findings.

Answer

...

HIV information experiment

Loading data

From the `causaldata` package, load the `thornton_hiv` data, and then turn it into a tibble after removing all missing data using `drop_na`. You can use `?thornton_hiv` to find the variable descriptions. This exercise is based on the replication in The Mixtape, Chapter 4.

You can read Chapter 4 as a secondary source about the material we discussed this week. Please read Section 4.1.5 before attempting this exercise, to read the necessary background about this experiment. Reading this section is also part of your self-study about SUTVA.

> Respondents in rural Malawi were offered a free door-to-door HIV test and randomly assigned no voucher or vouchers ranging from $1–$3. These vouchers were redeemable once they visited a nearby voluntary counseling and testing center (VCT).

In this data set, which variable corresponds to D? Which variable corresponds to Y?

First, use `group_by` and `summarize` to compute the group-specific means. Friendly reminder to use `drop_na()`! Second, repeat this exercise to compute a group-specific mean for each value of `tinc`. Third, take the resulting table and plot it using `geom_point` and/or `geom_line`. Interpret the results.

Answer

…

Treatment effects by age

You will now analyze the effect of `age` by adapting the approach we used for `educ`.

First, create a binarized variable, cutting off `age` at a value that you can determine. Second, compute the means for control and treatment group for each value of the binarized variable. Interpret your results.

Answer

...

For troubleshooting: do not edit or remove

```
sysname

"Darwin"

release

"25.2.0"

version
"Darwin Kernel Version 25.2.0: Tue Nov 18 21:09:49 PST 2025;
root:xnu-12377.61.12~1/RELEASE_ARM64_T8142"

nodename

"tadhgs-M5-MacBook-Pro.local"

machine

"arm64"

login

"root"

user

"tadhg"

effective_user

"tadhg"
```

```
[1] "2026-01-12 13:38:26 EST"
```