ECON 773: Assignment 1

The BLUE Team, Jeneta Ljutic (400138420), Tadhg Taylor-McGreal (400330297), Stella Till (400364649)

2026-01-12

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 install `R` and Positron. Positron comes with command-line tools for `quarto`, which in turn includes `typst`.

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.

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)
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`

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`
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`.
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`.

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

Answer

To answer the first question, we can use the 'arrange' function from 'dplyr'

```
JC |> arrange(age, educ)
```

```
# A tibble: 9,240 × 46
   assignment female   age white black hispanic  educ educmis geddegree
hsdegree
        <dbl>  <dbl> <dbl> <dbl> <dbl>    <dbl> <dbl>   <dbl>     <dbl>
<dbl>
 1          1      0    16     0     1        0     0       1         0
0
 2          0      0    16     1     0        0     0       1         0
0
 3          1      0    16     0     1        0     0       1         0
0
 4          0      0    16     0     0        1     0       1         0
0
 5          0      1    16     0     0        1     0       1         0
0
 6          1      1    16     0     0        1     0       1         0
0
 7          0      0    16     0     0        1     0       1         0
0
 8          1      1    16     0     1        0     0       1         0
0
 9          1      0    16     0     0        1     0       1         0
0
10          1      0    16     0     1        0     0       1         0
0
# i 9,230 more rows
```

```
# ℹ 36 more variables: english <dbl>, cohabmarried <dbl>, haschild <dbl>,
#   everwkd <dbl>, mwearn <dbl>, hhsize <dbl>, hhsizemis <dbl>, educmum <dbl>,
#   educmummis <dbl>, educdad <dbl>, educdadmis <dbl>, welfarechild <dbl>,
#   welfarechildmis <dbl>, health <dbl>, healthmis <dbl>, smoke <dbl>,
#   smokemis <dbl>, alcohol <dbl>, alcoholmis <dbl>, everwkdy1 <dbl>,
#   earnq4 <dbl>, earnq4mis <dbl>, pworky1 <dbl>, pworky1mis <dbl>, …
```

…

From visual inspection, it looks like the 'arrange' command worked.

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

JC |> select(educ, educmis, educ_high)
```

```
# A tibble: 9,240 × 3
    educ educmis educ_high
   <dbl>   <dbl> <lgl>
 1    12       0 TRUE
 2     8       0 FALSE
 3    10       0 FALSE
 4    10       0 FALSE
 5    12       0 TRUE
 6     0       1 FALSE
 7    13       0 TRUE
 8    10       0 FALSE
 9    11       0 FALSE
10    11       0 FALSE
# ℹ 9,230 more rows
```

This worked, and we checked it by visually inspecting the resulting data set in the viewer.

To answer the third question, we will use 'select' to keep only the relevant variables.

```
(JC |> select(assignment, female, earny4, educ_high, educmis) -> JC_short)
```

```
# A tibble: 9,240 × 5
   assignment female earny4 educ_high educmis
        <dbl>  <dbl>  <dbl> <lgl>       <dbl>
 1          0      0  265.  TRUE            0
 2          1      1  217.  FALSE           0
 3          0      1   11.9 FALSE           0
 4          1      1   18.3 FALSE           0
 5          1      0  221.  TRUE            0
 6          1      0  672.  FALSE           1
 7          1      0  334.  TRUE            0
 8          0      1  226.  FALSE           0
 9          0      0  324.  FALSE           0
10          1      0  334.  FALSE           0
# i 9,230 more rows
```

To keep only the observations corresponding to the treatment group, we will use the 'filter' function, applied to the 'assignment' variable.

::: {.cell}

```
JC_short |> filter(assignment ==1) -> JC_short_TG
JC_short |> filter(assignment ==0) -> JC_short_CG
```

:::

It worked! I am going to use the 'mean' function to compute the average annual earnings ('earny4') for the treatment group and the control group separetly. We start with the treatment group, which corresponds to 'JC_short_TG'.

::: {.cell}

```
mean(JC_short_TG$earny4)
```

::: {.cell-output .cell-output-stdout}

```
[1] 213.981
```

::: :::

Before deriving any conclusions, let's look at the control group.

::: {.cell}

```
mean(JC_short_CG$earny4)
```

::: {.cell-output .cell-output-stdout}

```
[1] 197.9258
```

::: :::

The average annual earnings, 4 years after the program, are higher by about 16 dollars in the treatment group. This agrees with the finding in H, p. 21. ## 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

```
JC_short |>
    group_by(female) |>

    summarize(EY = mean(earny4))
```

```
# A tibble: 2 × 2
  female    EY
   <dbl> <dbl>
1      0  236.
2      1  171.
```

```
JC_short |>
    filter(educmis == 0) |>
    group_by(assignment) |>
    summarize(EY = mean(earny4))
```

```
# A tibble: 2 × 2
  assignment    EY
       <dbl> <dbl>
1          0  198.
2          1  214.
```

```
JC_short |>
    filter(educmis == 0) |>
    group_by(female, assignment) |>
    summarize(EY = mean(earny4))
```

```
`summarise()` has grouped output by 'female'. You can override using the
`.groups` argument.
```

```
# A tibble: 4 × 3
# Groups:   female [2]
  female assignment    EY
   <dbl>      <dbl> <dbl>
1      0          0  223.
2      0          1  246.
3      1          0  160.
4      1          1  177.
```

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

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

```
mean_TG <- JC_short_TG |>
  summarize(mean_earny4 = mean(earny4, na.rm = TRUE)) |>
  pull(mean_earny4)

mean_CG <- JC_short_CG |>
  summarize(mean_earny4 = mean(earny4, na.rm = TRUE)) |>
  pull(mean_earny4)

ATE_diff <- mean_TG - mean_CG

mean_TG
```

```
[1] 213.981
```

```
mean_CG
```

```
[1] 197.9258
```

```
ATE_diff
```

```
[1] 16.05513
```

...

```
JC_short |>
  filter(educmis == 0) |>
  group_by(female, assignment) |>
  summarize(EY = mean(earny4), .groups = "drop")
```

```
# A tibble: 4 × 3
  female assignment    EY
   <dbl>      <dbl> <dbl>
1      0          0  223.
2      0          1  246.
3      1          0  160.
4      1          1  177.
```

```
JC_short |>
  filter(educmis == 0) |>
  group_by(female, assignment) |>
  summarize(EY = mean(earny4), .groups = "drop") |>
  pivot_wider(names_from = assignment,
              values_from = EY,
              names_prefix = "assign_") |>
  mutate(CATE = assign_1 - assign_0)
```

```
# A tibble: 2 × 4
  female assign_0 assign_1  CATE
   <dbl>    <dbl>    <dbl> <dbl>
1      0     223.     246.  23.1
2      1     160.     177.  17.6
```

In the first step, nothing happened. We just indicated to R, that we will group by female. In the second step, we saw that the expected earnings for males is 236 and 171 for females. This indicates that females earn less 65 less than males, annually.

In the second step, we see that those in the control group earn 198 and in the treatment group earning are 214. Here, we filter for only those with education.

In the third step, we group by treatment assignment and sex. We see that for males with the treatment, they earn 246 and for males without the treatment, they earn 223. For females with the treatment they earn 177 adn for females without the treatment they earn 160. Males earn more both with and without the treatment. If this is a RCT, we say that the gap is caused by the treatment. Males and females do not have the same treatment effect. The effect is 23 for men and 17 for women. Previously, we said that the overall effect is 16. The difference ATE_diff = mean_TG – mean_CG is the (unadjusted) estimated average treatment effect of assignment to Job Corps on fourth-year weekly earnings.

Fourth, This table reports mean fourth-year earnings separately by gender (female) and treatment status (assignment). For each gender, the conditional average treatment effect (CATE) is the difference in mean earnings between the treatment group (assignment = 1) and the control group (assignment = 0).

The CATE for men is the treatment–control difference among individuals with female = 0.

The CATE for women is the analogous difference among individuals with female = 1.

Comparing the two CATEs shows whether the Job Corps program had heterogeneous effects by gender. If the CATEs differ, this indicates that the impact of the program is not the same for men and women.

The unconditional ATE (computed earlier without conditioning on gender) is a weighted average of these two CATEs, where the weights are the proportions of men and women in the sample. Therefore, the unconditional ATE will generally not equal the simple average of the male and female CATEs unless the two groups are the same size.

This finding highlights that unconditional treatment effects can mask important heterogeneity across subgroups, and that policy conclusions may differ depending on which population is emphasized.

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

```
JC_short |>
    group_by(female) |>

    summarize(EY = mean(earny4))
```

```
# A tibble: 2 × 2
  female    EY
   <dbl> <dbl>
1      0  236.
2      1  171.
```

```
JC_short |>
    filter(educmis == 0) |>
    group_by(assignment) |>
    summarize(EY = mean(earny4))
```

```
# A tibble: 2 × 2
  assignment    EY
       <dbl> <dbl>
1          0  198.
2          1  214.
```

```
JC_short |>
    filter(educmis == 0) |>
    group_by(female, assignment) |>
    summarize(EY = mean(earny4))
```

```
`summarise()` has grouped output by 'female'. You can override using the
`.groups` argument.
```

```
# A tibble: 4 × 3
# Groups:   female [2]
  female assignment    EY
   <dbl>      <dbl> <dbl>
1      0          0  223.
2      0          1  246.
3      1          0  160.
4      1          1  177.
```

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

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

```
mean_TG <- JC_short_TG |>
  summarize(mean_earny4 = mean(earny4, na.rm = TRUE)) |>
  pull(mean_earny4)

mean_CG <- JC_short_CG |>
  summarize(mean_earny4 = mean(earny4, na.rm = TRUE)) |>
  pull(mean_earny4)

ATE_diff <- mean_TG - mean_CG

mean_TG
```

```
[1] 213.981
```

```
mean_CG
```

```
[1] 197.9258
```

```
ATE_diff
```

```
[1] 16.05513
```

…

```
JC_short |>
  filter(educmis == 0) |>
  group_by(female, assignment) |>
  summarize(EY = mean(earny4), .groups = "drop")
```

```
# A tibble: 4 × 3
  female assignment    EY
   <dbl>      <dbl> <dbl>
1      0          0  223.
2      0          1  246.
3      1          0  160.
4      1          1  177.
```

```
JC_short |>
  filter(educmis == 0) |>
  group_by(female, assignment) |>
  summarize(EY = mean(earny4), .groups = "drop") |>
  pivot_wider(names_from = assignment,
              values_from = EY,
              names_prefix = "assign_") |>
  mutate(CATE = assign_1 - assign_0)
```

```
# A tibble: 2 × 4
  female assign_0 assign_1  CATE
   <dbl>    <dbl>    <dbl> <dbl>
1      0     223.     246.  23.1
2      1     160.     177.  17.6
```

In the first step, nothing happened. We just indicated to R, that we will group by female. In the second step, we saw that the expected earnings for males is 236 and 171 for females. This indicates that females earn less 65 less than males, annually.

In the second step, we see that those in the control group earn 198 and in the treatment group earning are 214. Here, we filter for only those with education.

In the third step, we group by treatment assignment and sex. We see that for males with the treatment, they earn 246 and for males without the treatment, they earn 223. For females with the treatment they earn 177 adn for females without the treatment they earn 160. Males earn more both with and without the treatment. If this is a RCT, we say that the gap is caused by the treatment. Males and females do not have the same treatment effect. The effect is 23 for men and 17 for women. Previously, we said that the overall effect is 16. The difference ATE_diff = mean_TG – mean_CG is the (unadjusted) estimated average treatment effect of assignment to Job Corps on fourth-year weekly earnings.

Fourth, This table reports mean fourth-year earnings separately by gender (female) and treatment status (assignment). For each gender, the conditional average treatment effect (CATE) is the difference in mean earnings between the treatment group (assignment = 1) and the control group (assignment = 0).

The CATE for men is the treatment–control difference among individuals with female = 0.

The CATE for women is the analogous difference among individuals with female = 1.

Comparing the two CATEs shows whether the Job Corps program had heterogeneous effects by gender. If the CATEs differ, this indicates that the impact of the program is not the same for men and women.

The unconditional ATE (computed earlier without conditioning on gender) is a weighted average of these two CATEs, where the weights are the proportions of men and women in the sample. Therefore, the unconditional ATE will generally not equal the simple average of the male and female CATEs unless the two groups are the same size.

This finding highlights that unconditional treatment effects can mask important heterogeneity across subgroups, and that policy conclusions may differ depending on which population is emphasized.

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

1. We will use the JC data to calculate the ATE for the porportion of weeks employed in the fourth year after assignment, 'pworky4'. The code below returns the average values for treatement and control as well as the ATE in a table. First, 'group_by' splits the data into groups based on the value of the assignment variable. Next, 'summarise' collapses each group into a single row by computing the average of pworky4 within each assignment group. The 'pivot_wider' step reshapes the data from a long format into a wide format, creating separate columns for each value of assignment (prefixed with "assignment_") and filling them with the corresponding group means. Finally, 'mutate' adds the ATE variable.

```
JC |>
  group_by(assignment) |>
  summarise(
    mean_pworky4 = mean(pworky4, na.rm = TRUE),
    .groups = "drop"
  ) |>
  (\(df) pivot_wider(
    df,
    names_from = assignment,
    values_from = mean_pworky4,
    names_prefix = "assignment_"
  ))() |>
  mutate(
    ATE = assignment_1 - assignment_0
  )
```

```
# A tibble: 1 × 3
  assignment_0 assignment_1   ATE
         <dbl>        <dbl> <dbl>
1         58.3         61.6  3.27
```

The estimated ATE is 3.27, indicating that assignment to the treatment increased proportion of weeks employed in the fourth year after assignment by approximately three weeks on average, relative to the control group.

2. Next, we extend the analysis by estimating the ATE conditional on education group (using the 'educ_high' variable created above) to examine whether the impact of the treatment differs based on education level. The logic is very similar to the part 1, but this time we are grouping by 'educ_high' and 'assignment'.

```
JC |>
  group_by(educ_high, assignment) |>
  summarise(
    mean_pworky4 = mean(pworky4, na.rm = TRUE),
    .groups = "drop"
  ) |>
  (\(df) pivot_wider(
    df,
    names_from = assignment,
    values_from = mean_pworky4,
    names_prefix = "assignment_"
  ))() |>
  mutate(
    CATE = assignment_1 - assignment_0
  )
```

```
# A tibble: 2 × 4
  educ_high assignment_0 assignment_1  CATE
  <lgl>            <dbl>        <dbl> <dbl>
1 FALSE             55.8         58.7  2.91
2 TRUE              67.7         71.1  3.35
```

For individuals with 12 years of education or more ('educ_high' = 1), the treatment increases proportion of weeks employed in the fourth year after assignment by 3.35 weeks. For those with 12 years of education or less (educ_high = 0), the estimated effect is slightly smaller, at 2.91 weeks. These results suggest that the program has a slightly larger impact on employment for higher-educated individuals, although the difference is relatively small (0.44 weeks).

3. Using 'haschild' ('haschild' = 1 indicates at least one child, 'haschild' = 0 indicates no children at assignment), we'll explore the effect of the program conditional on whether individuals have at least one child at assignment.

```
JC |>
  group_by(haschild, assignment) |>
  summarise(
    mean_pworky4 = mean(pworky4, na.rm = TRUE),
    .groups = "drop"
```

```
  ) |>
  (\(df) pivot_wider(
    df,
    names_from = assignment,
    values_from = mean_pworky4,
    names_prefix = "assignment_"
  ))() |>
  mutate(
    CATE = assignment_1 - assignment_0
  )
```

```
# A tibble: 2 × 4
  haschild assignment_0 assignment_1  CATE
     <dbl>        <dbl>        <dbl> <dbl>
1        0         59.0         61.7  2.76
2        1         55.6         61.2  5.61
```

For individuals without children at assignment ('haschild' = 0), the estimated CATE is 2.76 weeks of employment in the fourth year, while for those with at least one child at assignment ('haschild' = 1), the effect is substantially larger at 5.76 weeks (for a difference of 3 weeks). These results indicate that the treatment has a stronger impact on employment for parents, suggesting that the program may be particularly beneficial for individuals with children.

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.

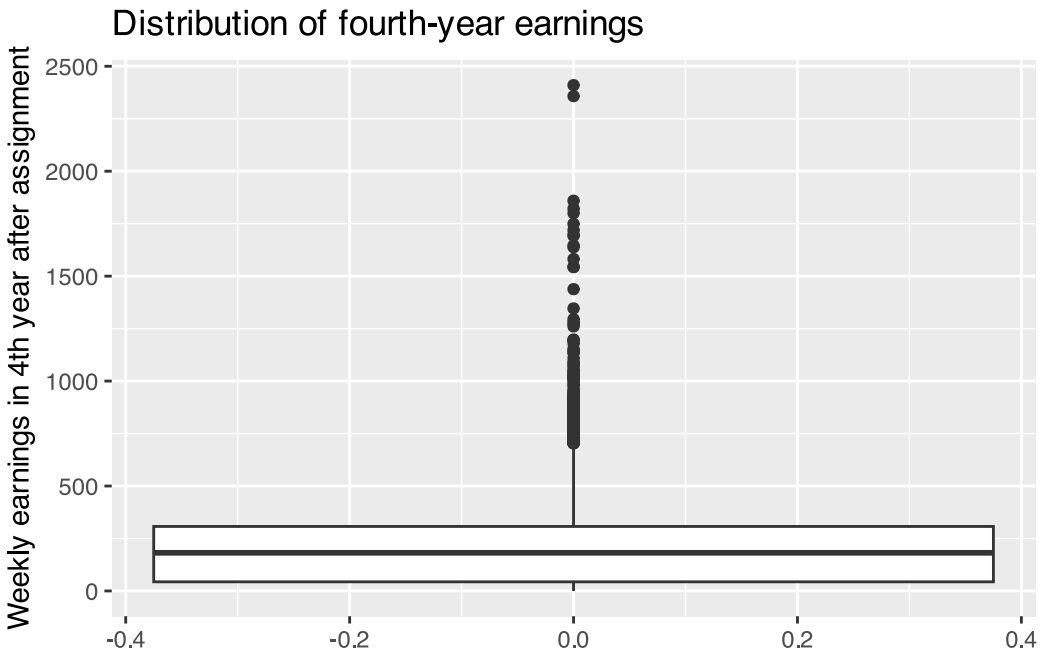
You will continue with the `JC` data. A useful data summary for our purpose is the boxplot.

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

To code our boxplot we use 'ggplot'. Using 'ggplot' allows us to build the plot step by step using JC data. Withing our 'ggplot' bracket, we indicate that the y axis should show the category for earnings after the fourth year of the program. Following 'ggplot', we use 'geom_boxplot'. This tells R to put the JC data into a boxplot. In the last step of the code, we set the titles. We repeated this code skeleton for all of the following boxplots, changing the variables to either be in logs, sorted by assignment or both. In the second step, these changes are made by indicating under 'ggplot' to filter earny4>0, this ensures that only positive values are included, essential when taking a logarithim. To split out each of the boxplots by 'assignment', we type factor(assignment, ...) in the first line of 'ggplot' code.

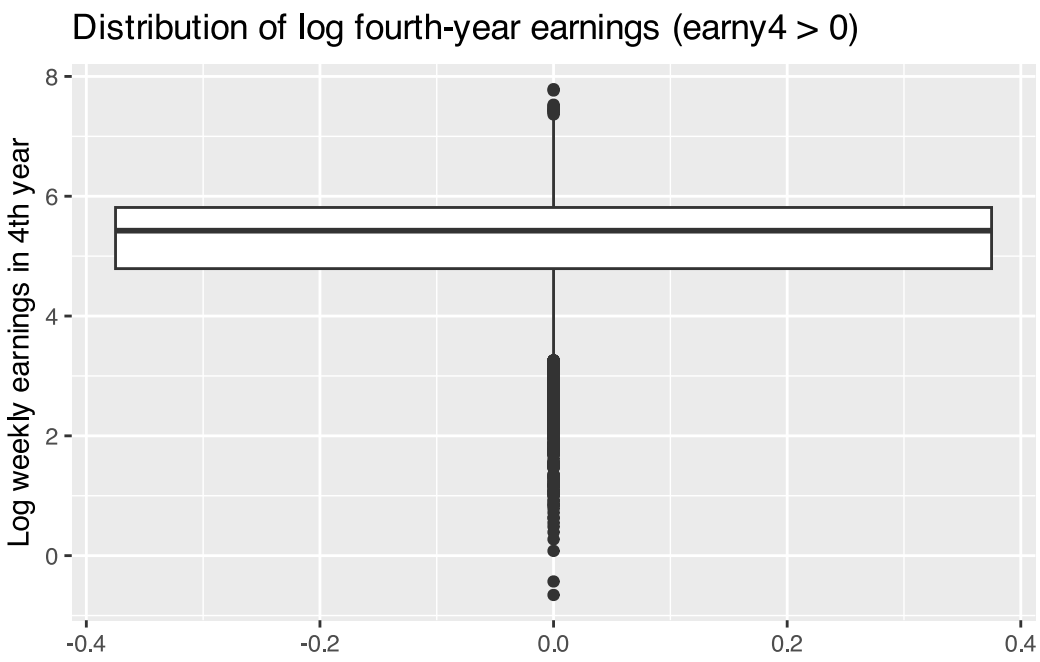
```
ggplot(JC, aes(y = earny4)) +
  geom_boxplot() +
  labs(
    y = "Weekly earnings in 4th year after assignment",
    title = "Distribution of fourth-year earnings"
  )
```



This boxplot shows the weekly earnings in the fourth year after assignment on the y-axis and In this boxplot we see a skew to the right. This indicates that most individuals earn relatively low wages. The high earning individuals appear to be outliers. This suggests that only some individuals benefit largely in the fourth year post Job Corps program which indicates that the treatment did not have the same effect across all participants.

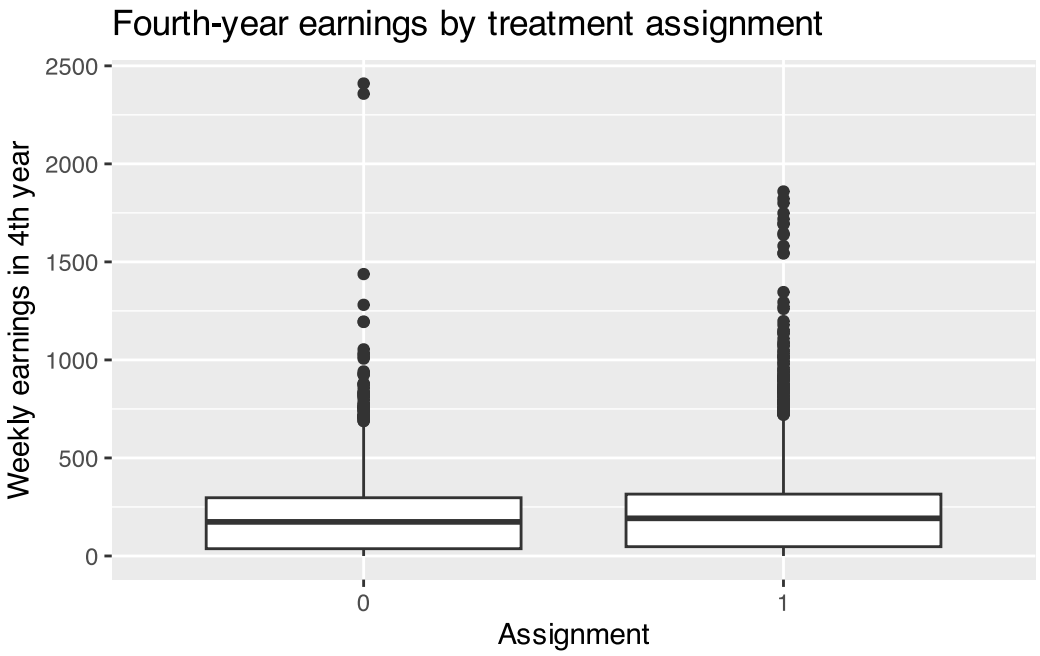
In this plot, the effect of earnings are skewed by the extreme high values. This is because in a chart using levels, the differences are absolute and are thus highly influenced by outliers. This motivates the use of log earnings in the following charts since taking the logs compresses the upper tail of the distribution allow for comparisons which reflect proportional differences in earnings rather than absolute differences.

```
ggplot(
  JC |> filter(earny4 > 0),
  aes(y = log(earny4))
) +
  geom_boxplot() +
  labs(
    y = "Log weekly earnings in 4th year",
    title = "Distribution of log fourth-year earnings (earny4 > 0)"
  )
```



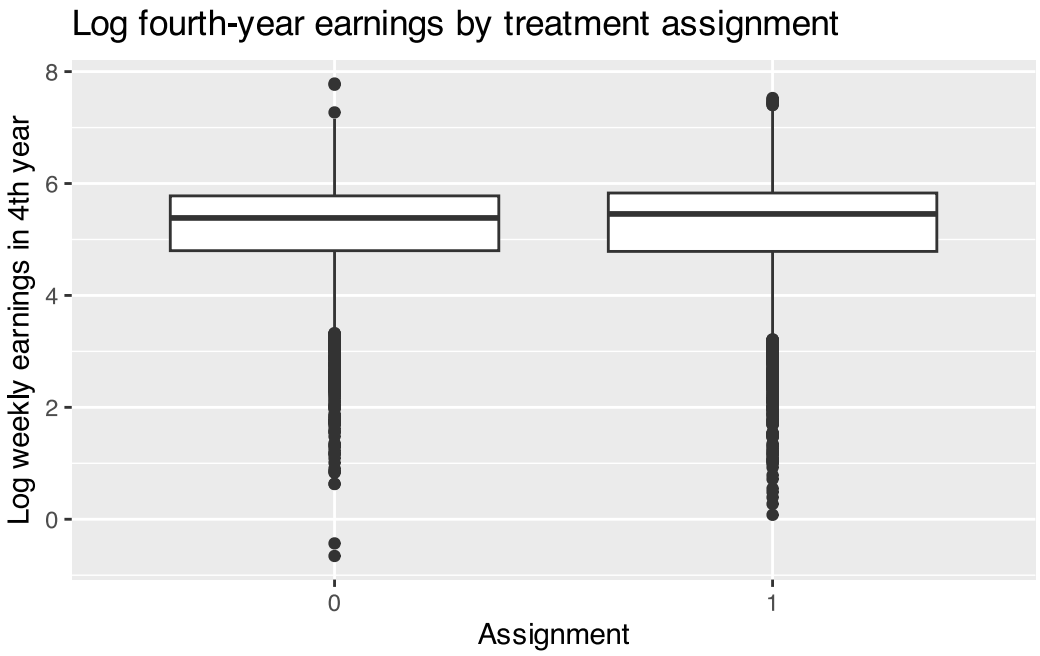
The second boxplot shows the distribution of log earnings for those with positive earnings. In this plot, we see that the earnings are more centred. This indicates that once we restrict for positive earnings, measured on a log scale the upper tail one the distribution becomes compressed. There remain some lower outliers, indicating that some individuals continue to have low but positive earnings. Thus, even when separating for positive earnings, heterogeneity of the results remain suggesting that there are other characteristics that influence earnings in the period four years after the program.

```
ggplot(JC, aes(x = factor(assignment), y = earny4)) +
  geom_boxplot() +
  labs(
    x = "Assignment",
    y = "Weekly earnings in 4th year",
    title = "Fourth-year earnings by treatment assignment"
  )
```



The boxplot, shows the weekly earnings in the fourth year after the treatment and is grouped by assignment on the x-axis. The y-axis shows the level of weekly earnings. In this plot, all earnings are positive. Dividing the earnings distribution by assignment highlights the differences in earnings between the control group (assignment = 0) and the treatment group (assignment = 1). We see a strong right-skew of earnings in both groups. This indicates that most individuals earn relatively low wages. We can note that there are few outliers in the control group with very high earnings. The outliers in both the treatment and the control group suggest that there may be heterogeneous program effects. However, it is important to distinguish that the treatment group has shows a higher median level of earnings. This suggests that there was a positive effect from the Jobs Corps program.

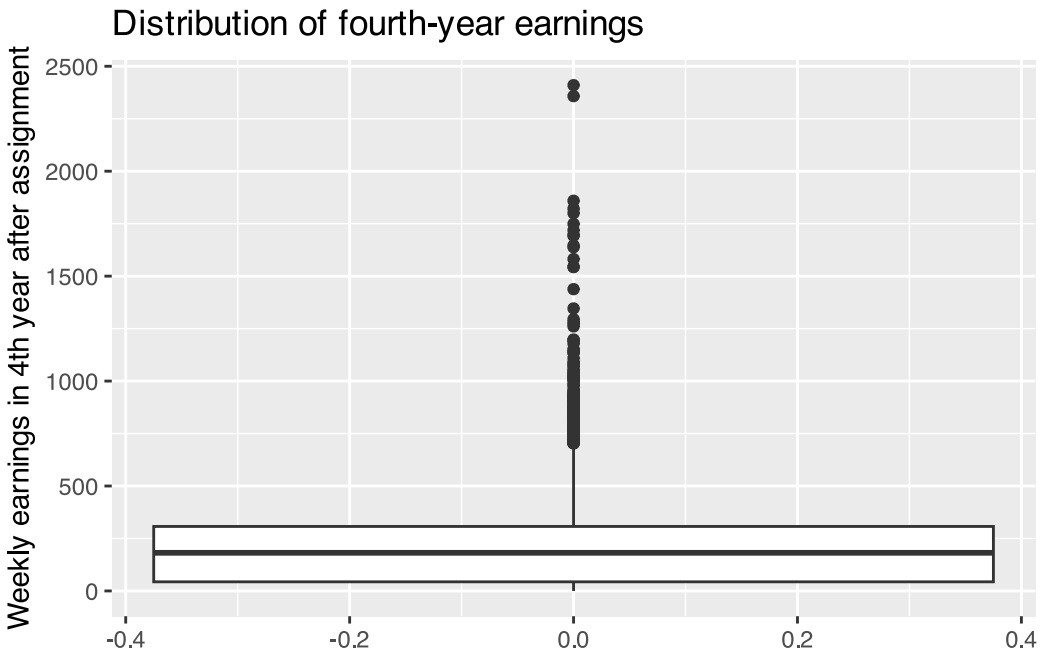
```
ggplot(
  JC |> filter(earny4 > 0),
  aes(x = factor(assignment), y = log(earny4))
) +
  geom_boxplot() +
  labs(
    x = "Assignment",
    y = "Log weekly earnings in 4th year",
    title = "Log fourth-year earnings by treatment assignment"
  )
```



This final boxplot shows the positive earnings by assignment. It compares the distribution of log weekly earnings in the fourth year after the assignment between the control group and the (assignment = 0) and the treatment group (assignment = 1). All individuals in this plot are restricted to have positive earnings. On the boxplot, we see that there is a higher median log earnings in the treatment group. This indicates that those assigned to the Jobs Corps training program have higher typical earnings four years after the treatment assignment. This suggests a positive average treatment effect of Jobs Corps on earnings.

There is substantial overlap between the distributions of the treatment and control group. This suggests that the programs effects were moderate and there may be other characteristics driving certain individuals to earn higher wages.

```
ggplot(JC, aes(y = earny4)) +
  geom_boxplot() +
  labs(
    y = "Weekly earnings in 4th year after assignment",
    title = "Distribution of fourth-year earnings"
  )
```



)

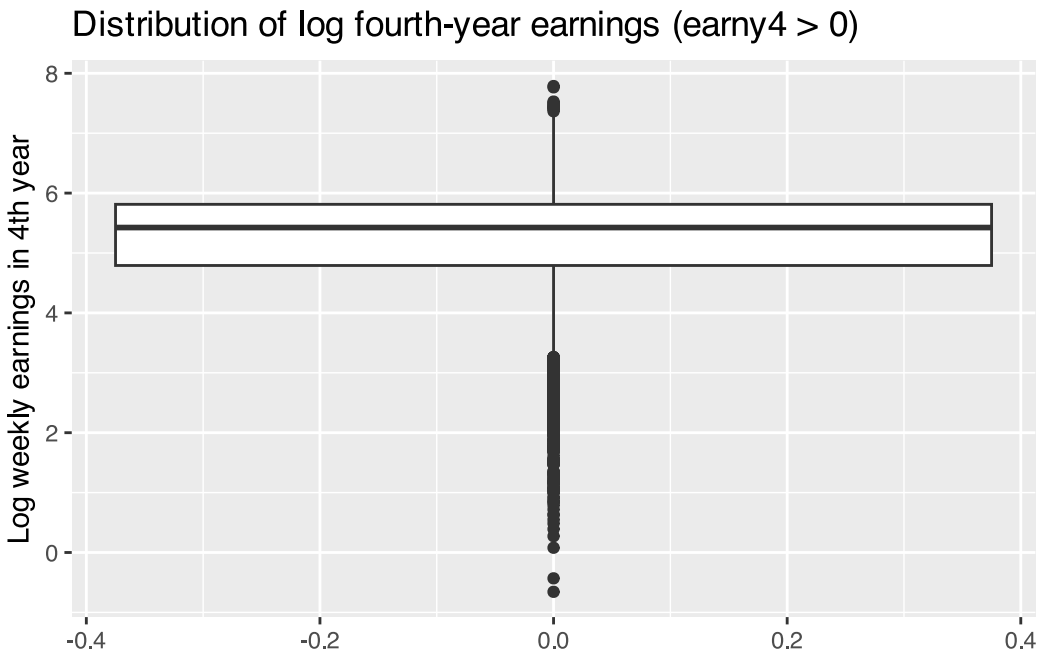
```
This boxplot shows the weekly earnings in the fourth year after assignment on
the y-axis and In this boxplot we see a skew to the right. This indicates that
most individuals earn relatively low wages. The high earning individuals
appear to be outliers. This suggests that only some individuals benefit
largely in the fourth year post Job Corps program which indicates that the
treatment did not have the same effect across all participants.

In this plot, the effect of earnings are skewed by the extreme high values.
This is because in a chart using levels, the differences are absolute and are
thus highly influenced by outliers. This motivates the use of log earnings in
the following charts since taking the logs compresses the upper tail of the
distribution allow for comparisons which reflect proportional differences in
earnings rather than absolute differences.


::: {.cell}

```{.r .cell-code}
ggplot(
  JC |> filter(earny4 > 0),
  aes(y = log(earny4))
) +
  geom_boxplot() +
  labs(
```
```

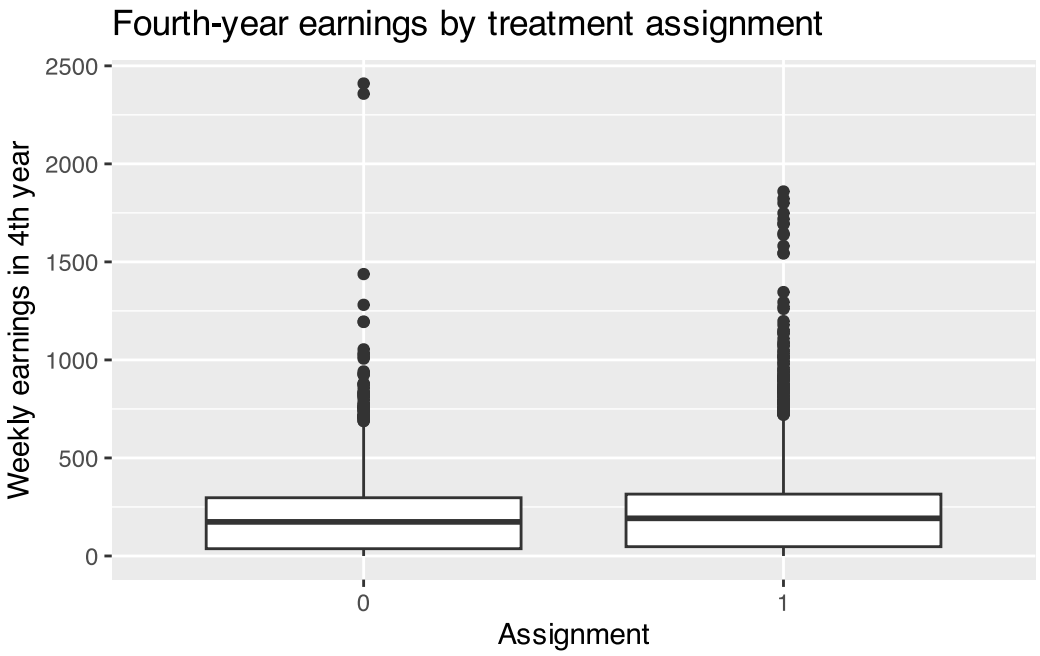
```
    y = "Log weekly earnings in 4th year",
    title = "Distribution of log fourth-year earnings (earny4 > 0)"
  )
```



:::

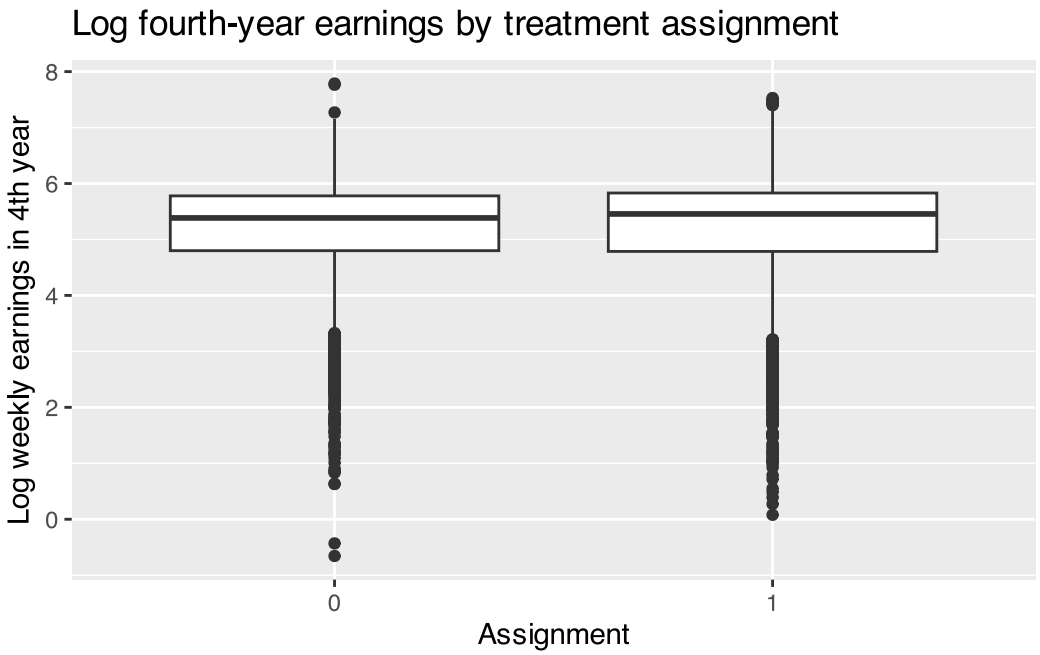
The second boxplot shows the distribution of log earnings for those with positive earnings. In this plot, we see that the earnings are more centred. This indicates that once we restrict for positive earnings, measured on a log scale the upper tail one the distribution becomes compressed. There remain some lower outliers, indicating that some individuals continue to have low but positive earnings. Thus, even when separating for positive earnings, heterogeneity of the results remain suggesting that there are other characteristics that influence earnings in the period four years after the program.

```
ggplot(JC, aes(x = factor(assignment), y = earny4)) +
  geom_boxplot() +
  labs(
    x = "Assignment",
    y = "Weekly earnings in 4th year",
    title = "Fourth-year earnings by treatment assignment"
  )
```



The boxplot, shows the weekly earnings in the fourth year after the treatment and is grouped by assignment on the x-axis. The y-axis shows the level of weekly earnings. In this plot, all earnings are positive. Dividing the earnings distribution by assignment highlights the differences in earnings between the control group (assignment = 0) and the treatment group (assignment = 1). We see a strong right-skew of earnings in both groups. This indicates that most individuals earn relatively low wages. We can note that there are few outliers in the control group with very high earnings. The outliers in both the treatment and the control group suggest that there may be heterogeneous program effects. However, it is important to distinguish that the treatment group has shows a higher median level of earnings. This suggests that there was a positive effect from the Jobs Corps program.

```
ggplot(
  JC |> filter(earny4 > 0),
  aes(x = factor(assignment), y = log(earny4))
) +
  geom_boxplot() +
  labs(
    x = "Assignment",
    y = "Log weekly earnings in 4th year",
    title = "Log fourth-year earnings by treatment assignment"
  )
```



This final boxplot shows the positive earnings by assignment. It compares the distribution of log weekly earnings in the fourth year after the assignment between the control group and the (assignment = 0) and the treatment group (assignment = 1). All individuals in this plot are restricted to have positive earnings. On the boxplot, we see that there is a higher median log earnings in the treatment group. This indicates that those assigned to the Jobs Corps training program have higher typical earnings four years after the treatment assignment. This suggests a positive average treatment effect of Jobs Corps on earnings.

There is substantial overlap between the distributions of the treatment and control group. This suggests that the programs effects were moderate and there may be other characteristics driving certain individuals to earn higher wages.

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`.

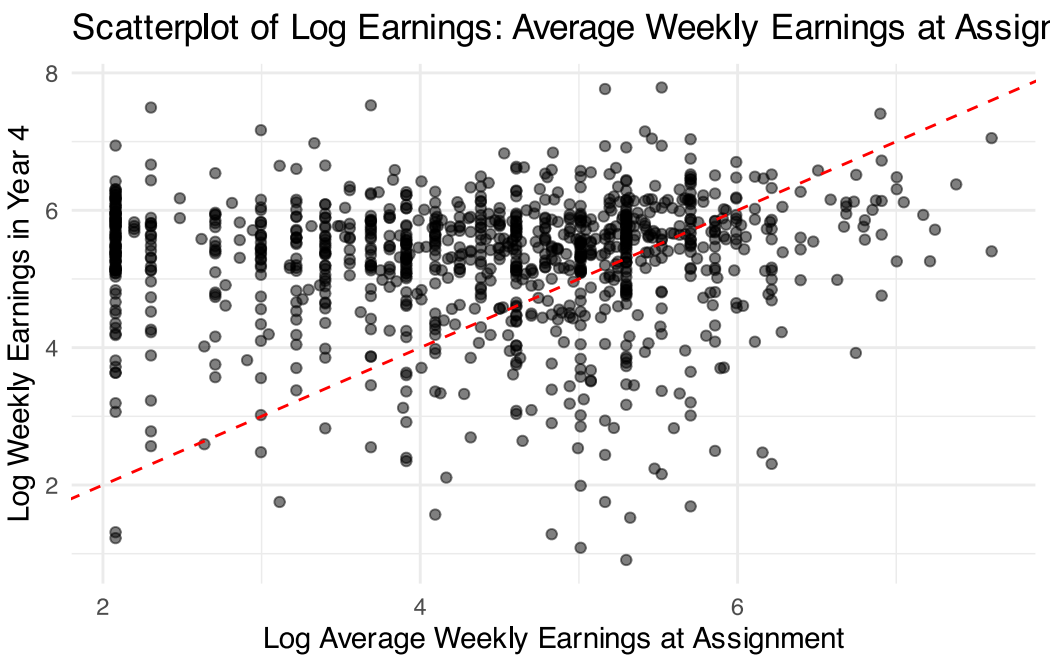
Third, split the results out by `assignment`, by setting `colour = assignment`. Interpret your findings.

Answer

1. We will use 'ggplot' and 'geom_point' to create a scatterplot of logged earnings at assignment, 'mwearn' and at year 4, 'earny4'. The code filters the data to include only individuals with positive earnings at assignment and in year 4 so that taking logarithms is valid. It creates a scatterplot of log earnings at assignment versus log earnings in year 4, adds semi-transparent points, and overlays a dashed 45-degree reference line showing where earnings would be equal in both periods.

```
library(ggplot2)

JC |>
  filter(mwearn > 0, earny4 > 0) |>  # keep only positive earnings
  ggplot(aes(x = log(mwearn), y = log(earny4))) +
  geom_point(alpha = 0.5) +          # semi-transparent points for clarity
  geom_abline(
    slope = 1,
    intercept = 0,
    linetype = "dashed",
    color = "red"
  ) +                                # 45-degree reference line
  labs(
    x = "Log Average Weekly Earnings at Assignment",
    y = "Log Weekly Earnings in Year 4",
    title = "Scatterplot of Log Earnings: Average Weekly Earnings at
Assignment vs Year 4"
  ) +
  theme_minimal()
```



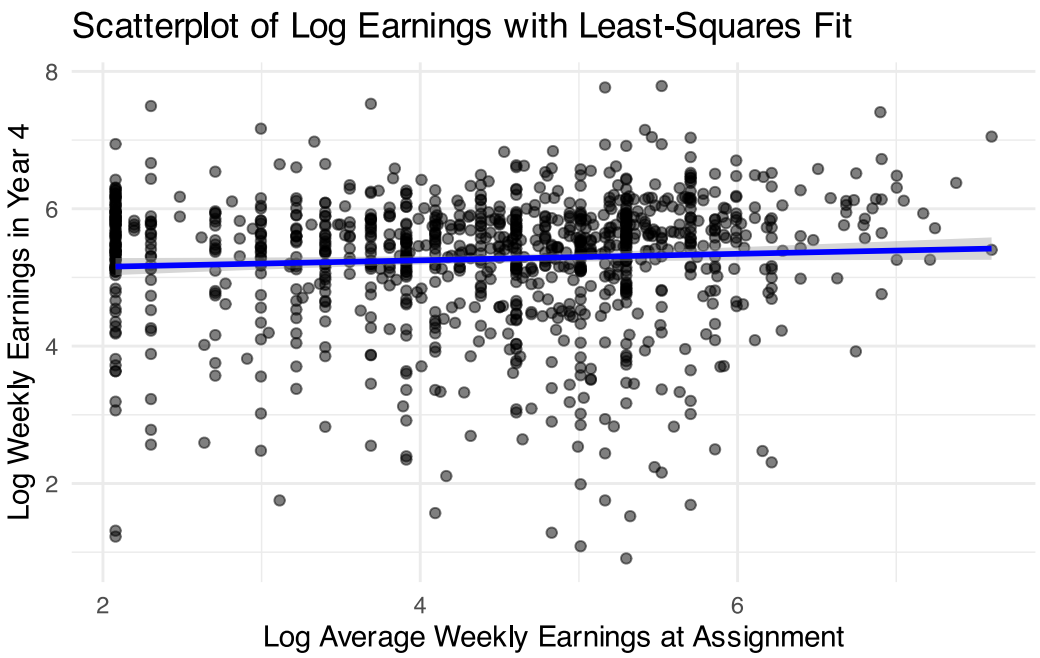
The scatterplot of log earnings at assignment versus log earnings in the fourth year shows that most individuals are concentrated in the middle of the earnings distribution (between log earnings of 4 and 6 in year 4 and at assignment). There is only a slight upward-right trend, indicating a weak positive relationship between initial and later earnings. This suggests that while higher earnings at assignment are somewhat associated with higher earnings in year 4, there is substantial variation in earnings growth across individuals.

2. Next, we will use geom_smooth() to add a least-squares regression line to our scatterplot to show the overall trend in log earnings. The code is similar to above, but overlays a least-squares regression line with a confidence band using geom_smooth(method = "lm").

```
JC |>
  filter(mwearn > 0, earny4 > 0) |>
  ggplot(aes(x = log(mwearn), y = log(earny4))) +
  geom_point(alpha = 0.5) +                    # points
  geom_smooth(
    method = "lm",
    se = TRUE,
    color = "blue"
  ) +                                          # least-squares fit with
confidence band
  labs(
    x = "Log Average Weekly Earnings at Assignment",
    y = "Log Weekly Earnings in Year 4",
    title = "Scatterplot of Log Earnings with Least-Squares Fit"
```

```
  ) +
  theme_minimal()
```

```
`geom_smooth()` using formula = 'y ~ x'
```



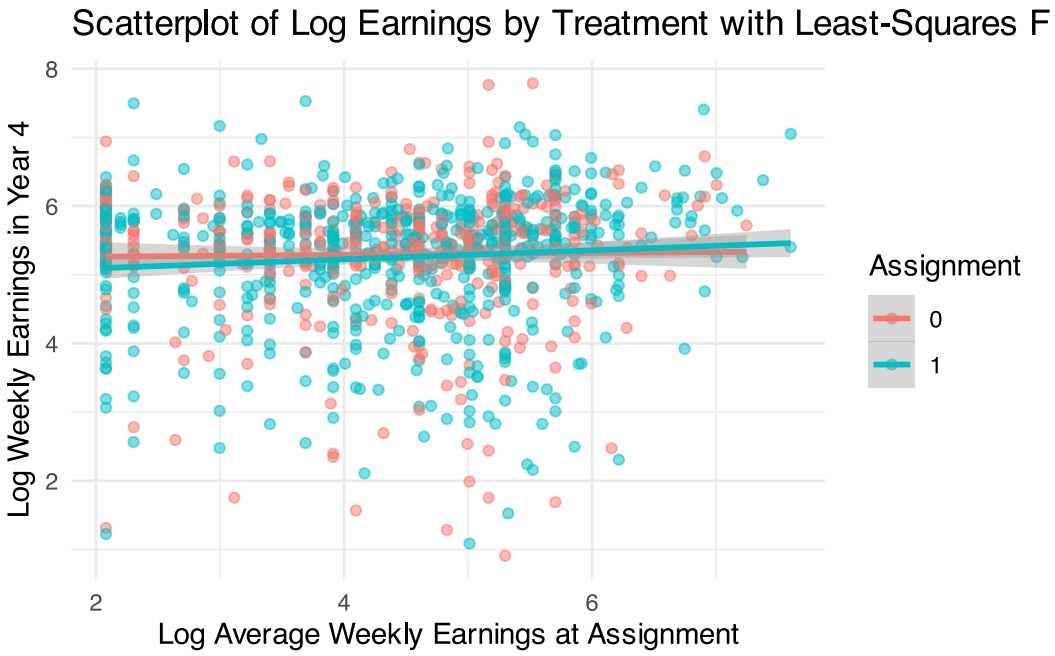
The scatterplot with a least-squares fit shows that earnings at assignment are only weakly predictive of earnings in the fourth year. The slope of the regression line is slightly upward, indicating that higher initial earnings are associated with slightly higher later earnings, but the relationship is modest. Most individuals cluster near the middle of the distribution, and there is considerable variation in earnings growth across individuals.

3. Lastly, we will add color by treatment group to see separate trends for treated and control groups and their fitted lines. This is again similar to the code above, but adds colours to points by treatment group ('assignment') using 'factor' to treat assignment as a categorical variable rather than a numeric one. This ensures that each group gets a separate colour and that geom_smooth() draws separate regression lines with confidence bands for each group.

```
JC |>
  filter(mwearn > 0, earny4 > 0) |>
  ggplot(aes(x = log(mwearn), y = log(earny4), color = factor(assignment))) +
  geom_point(alpha = 0.5) +                    # semi-transparent points
  geom_smooth(method = "lm", se = TRUE) +      # separate regression lines by
color with confidence bands
  labs(
```

```
    x = "Log Average Weekly Earnings at Assignment",
    y = "Log Weekly Earnings in Year 4",
    color = "Assignment",
    title = "Scatterplot of Log Earnings by Treatment with Least-Squares Fit"
  ) +
  theme_minimal()
```

```
`geom_smooth()` using formula = 'y ~ x'
```



The control group's line (denoted in orange) is relatively flat, suggesting that initial earnings have little effect on year-4 earnings for this group. In contrast, the treated group's line (denoted in blue) is slightly steeper, indicating that higher initial earnings are associated with higher year-4 earnings, and that the treatment amplifies earnings growth. Overall, this suggests that the treatment has a slightly larger positive impact on earnings, particularly for individuals who started with higher earnings, while individual variation remains substantial.

...

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

```
# Packages
library(causaldata)
library(dplyr)
library(tidyr)
library(ggplot2)
library(tibble)

data("thornton_hiv")

thornton <- thornton_hiv |>
  drop_na() |>
  as_tibble()

thornton |>
  group_by(any) |>
  summarize(
    mean_got = mean(got),
    n = n(),
    .groups = "drop"
  )
```

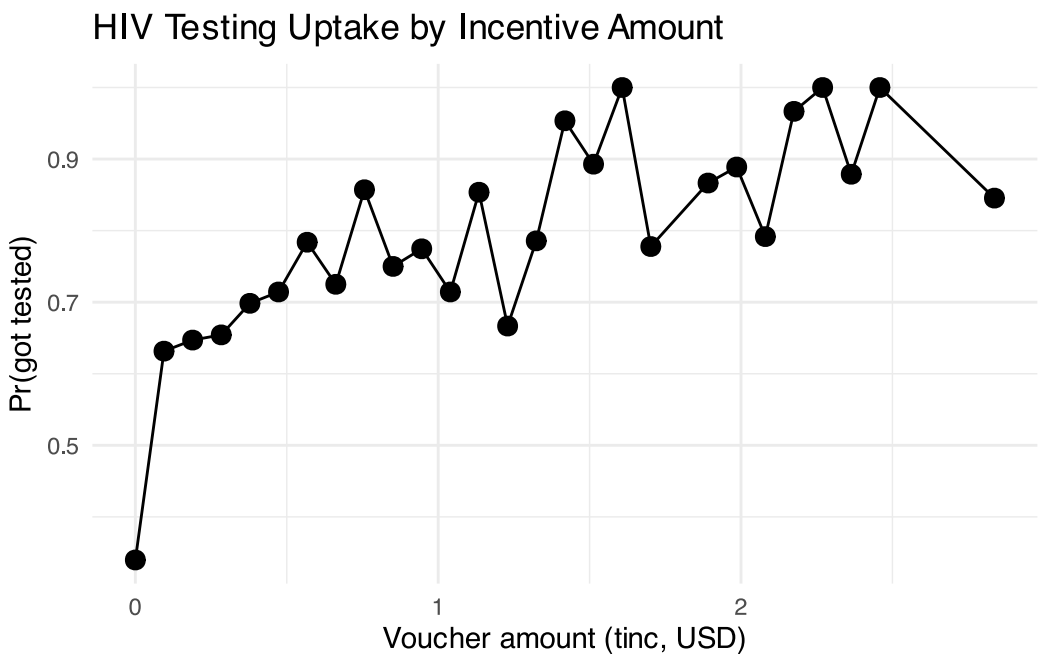
```
# A tibble: 2 × 3
    any mean_got     n
  <dbl>    <dbl> <int>
1     0    0.340   621
2     1    0.791  2204
```

```
by_amount <- thornton |>
  group_by(tinc) |>
  summarize(
    mean_got = mean(got),
    n = n(),
    .groups = "drop"
  )

by_amount
```

```
# A tibble: 27 × 3
     tinc mean_got     n
    <dbl>    <dbl> <int>
 1 0         0.340   621
 2 0.0946    0.632    57
 3 0.189     0.647   153
 4 0.284     0.654    81
 5 0.378     0.698    63
 6 0.473     0.714   203
 7 0.567     0.784    37
 8 0.662     0.725    40
 9 0.756     0.857     7
10 0.851     0.75      8
# ℹ 17 more rows
```

```
ggplot(by_amount, aes(x = tinc, y = mean_got)) +
  geom_point(size = 3) +
  geom_line() +
  labs(
    x = "Voucher amount (tinc, USD)",
    y = "Pr(got tested)",
    title = "HIV Testing Uptake by Incentive Amount"
  ) +
  theme_minimal()
```


HIV Testing Uptake by Incentive Amount
Pr(got tested)
0.9
0.7
0.5
0
1
2
Voucher amount (tinc, USD)

...

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

```
median(JC$age, na.rm = TRUE)
```

```
[1] 18
```

To choose an age of the cutoff range, we use the sample median age. To determine this we ran the code above. The output is that the sample median age is 18. Thus, those in the older group are those aged 18 or older and those in the younger group are 17 and below.

```
age_cut <- median(JC$age, na.rm = TRUE)

JC <- JC |>
  mutate(age_high = age >= age_cut)
JC |>
  count(age_high)
```

```
# A tibble: 2 × 2
  age_high     n
  <lgl>    <int>
1 FALSE     3740
2 TRUE      5500
```

This splits the sample into a "younger" and "older" group. In the younger group, we have 3740 observations and 5500 observations in the older group. Age_higher = FALSE indicates the younger age group.

```
JC |>
  filter(educmis == 0) |>
  group_by(age_high, assignment) |>
  summarize(EY = mean(earny4), .groups = "drop") |>
  pivot_wider(names_from = assignment,
              values_from = EY,
              names_prefix = "assign_") |>
  mutate(CATE = assign_1 - assign_0)
```

```
# A tibble: 2 × 4
  age_high assign_0 assign_1  CATE
  <lgl>       <dbl>    <dbl> <dbl>
1 FALSE        177.     188.  10.7
2 TRUE         214.     231.  17.0
```

To compute the means for the control and treatment group for each value of the binarized variable we ran the code above. The CATE is the conditional average treatment effect and shows the different in mean earnings between the treatment and control.

Our results show that the CATE for younger individuals is 10.7. This means that assignment to Jobs Corps increases fourth-year earnings by about 10.7 units for younger individuals. The control mean for younger individuals is 177 and the treatment mean is 188.

For older individuals (age_high = TRUE), the control mean is 214 and the treatment mean is 231. The CATE is 17. Thus, our findings suggest that the increased average earnings is higher for older individuals. This suggests heterogeneous treatment effects by age.

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] "2026-01-14 18:25:39 EST"
```