Difference-in-differences

If you want to follow along with this example, you can download the data below:

- injury.csv

Program background

In 1980, Kentucky raised its cap on weekly earnings that were covered by worker's compensation. We want to know if this new policy caused workers to spend more time unemployed. If benefits are not generous enough, then workers could sue companies for on-the-job injuries, while overly generous benefits could cause moral hazard issues and induce workers to be more reckless on the job, or to claim that off-the-job injuries were incurred while at work.

The main outcome variable we care about is `log_duration` (in the original data as `ldurat`, but we rename it to be more human readable), or the logged duration (in weeks) of worker's compensation benefits. We log it because the variable is fairly skewed—most people are unemployed for a few weeks, with some unemployed for a long time. The policy was designed so that the cap increase did not affect low-earnings workers, but did affect high-earnings workers, so we use low-earnings workers as our control group and high-earnings workers as our treatment group.

The data is included in the **wooldridge** R package as the `injury` dataset, and if you install the package, load it with `library(wooldridge)`, and run `?injury` in the console, you can see complete details about what's in it. To give you more practice with loading data from external files, I exported the injury data as a CSV file (using `write_csv(injury, "injury.csv")`) and included it here.

These are the main columns we'll worry about for now:

- `durat` (which we'll rename to `duration`): Duration of unemployment benefits, measured in weeks
- `ldurat` (which we'll rename to `log_duration`): Logged version of `durat` (`log(durat)`)

- `after_1980` (which we'll rename to `after_1980`): Indicator variable marking if the observation happened before (0) or after (1) the policy change in 1980. This is our **time** (or *before/after* variable)
- `highearn`: Indicator variable marking if the observation is a low (0) or high (1) earner. This is our **group** (or *treatment/control*) variable

Load and clean data

First, let's download the dataset (if you haven't already), put in a folder named `data`, and load it:

- injury.csv

```
library(tidyverse)  # ggplot(), %>%, mutate(), and friends
library(broom)  # Convert models to data frames
library(scales)  # Format numbers with functions like comma(), percent(), and dollar()
library(modelsummary)  # Create side-by-side regression tables
library(readr)
```

```
# Load the data.
# It'd be a good idea to click on the "injury_raw" object in the Environment
# panel in RStudio to see what the data looks like after you load it
injury_raw <- read_csv("injury.csv")
```

Next we'll clean up the `injury_raw` data a little to limit the data to just observations from Kentucky. we'll also rename some awkwardly-named columns:

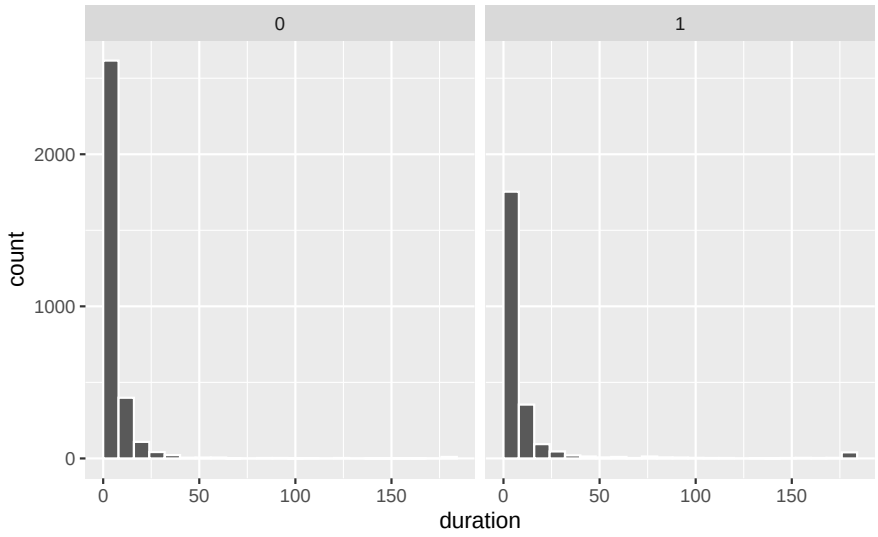
```
injury <- injury_raw %>%
  filter(ky == 1) %>%
  # The syntax for rename is `new_name = original_name`
  rename(duration = durat, log_duration = ldurat,
         after_1980 = afchnge)
```

Exploratory data analysis

First we can look at the distribution of unemployment benefits across high and low earners (our control and treatment groups):

```
ggplot(data = injury, aes(x = duration)) +
  # binwidth = 8 makes each column represent 2 months (8 weeks)
```

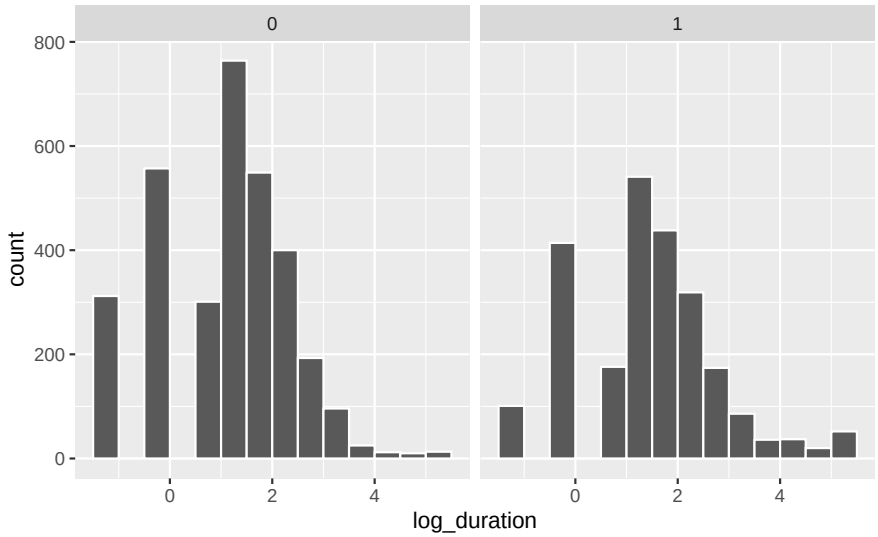
```
  # boundary = 0 make it so the 0-8 bar starts at 0 and isn't -4 to 4
  geom_histogram(binwidth = 8, color = "white", boundary = 0) +
  facet_wrap(vars(highearn))
```



The distribution is really skewed, with most people in both groups getting between 0-8 weeks of benefits (and a handful with more than 180 weeks! that's 3.5 years!)

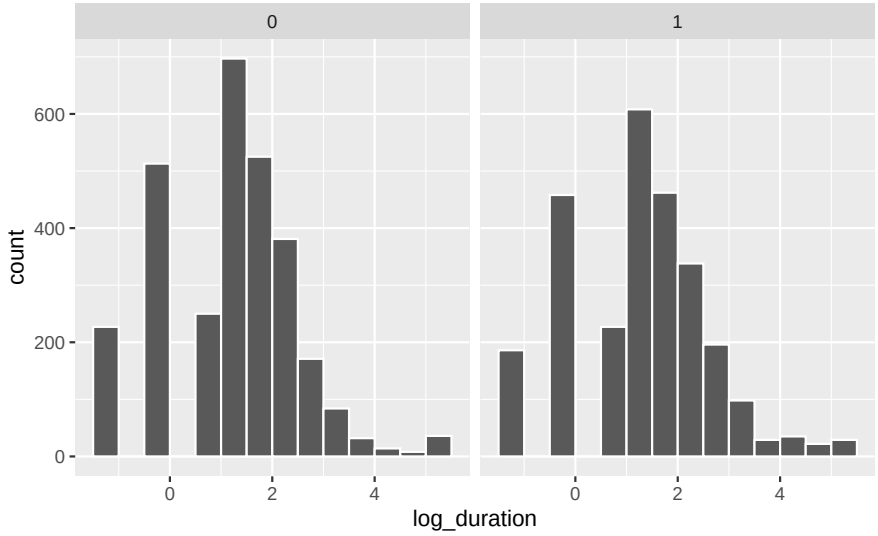
If we use the log of duration, we can get a less skewed distribution that works better with regression models:

```
ggplot(data = injury, mapping = aes(x = log_duration)) +
  geom_histogram(binwidth = 0.5, color = "white", boundary = 0) +
  # Uncomment this line if you want to exponentiate the logged values on the
  # x-axis. Instead of showing 1, 2, 3, etc., it'll show e^1, e^2, e^3, etc. and
  # make the labels more human readable
  # scale_x_continuous(labels = trans_format("exp", format = round)) +
  facet_wrap(vars(highearn))
```



We should also check the distribution of unemployment before and after the policy change. Copy/paste one of the histogram chunks and change the faceting:

```
ggplot(data = injury, mapping = aes(x = log_duration)) +
  geom_histogram(binwidth = 0.5, color = "white", boundary = 0) +
  facet_wrap(vars(after_1980))
```

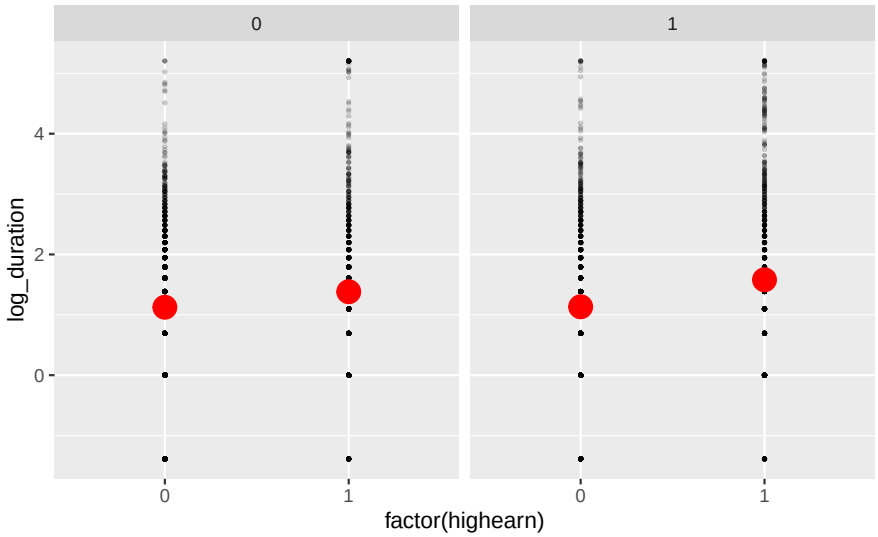


The distributions look normal-ish, but we can't really easily see anything different between the before/after and treatment/control groups. We can plot the averages, though. There are

a few different ways we can do this.

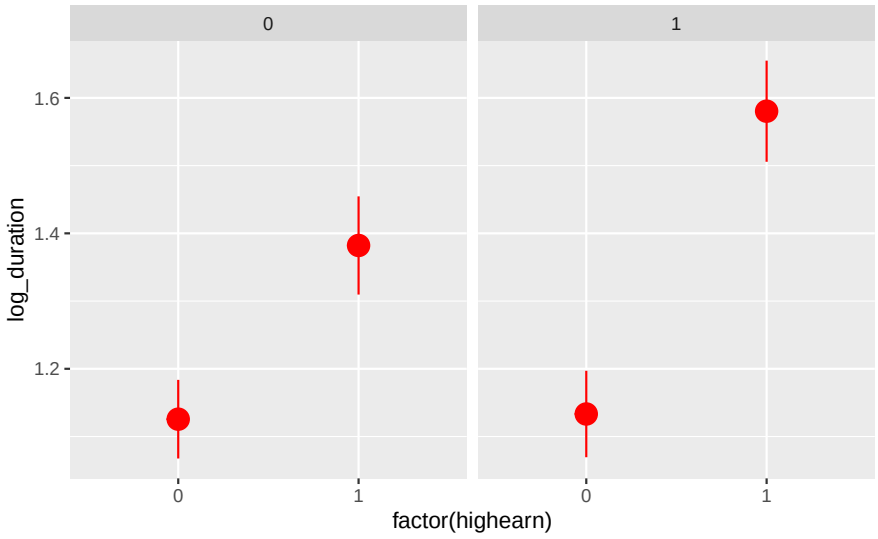
You can use a `stat_summary()` layer to have ggplot calculate summary statistics like averages. Here we just calculate the mean:

```
ggplot(injury, aes(x = factor(highearn), y = log_duration)) +
  geom_point(size = 0.5, alpha = 0.2) +
  stat_summary(geom = "point", fun = "mean", size = 5, color = "red") +
  facet_wrap(vars(after_1980))
```



But we can also calculate the mean and 95% confidence interval:

```
ggplot(injury, aes(x = factor(highearn), y = log_duration)) +
  stat_summary(geom = "pointrange", size = 1, color = "red",
               fun.data = "mean_se", fun.args = list(mult = 1.96)) +
  facet_wrap(vars(after_1980))
```

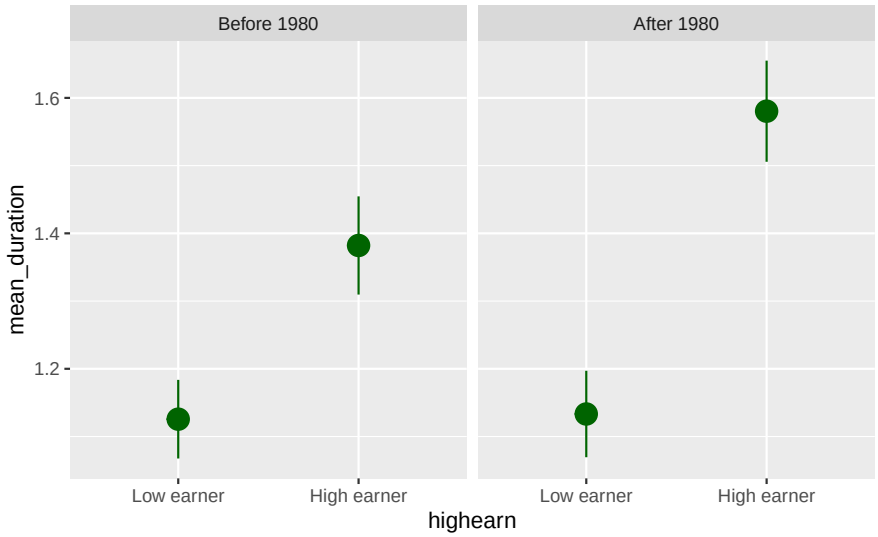


We can already start to see the classical diff-in-diff plot! It looks like high earners after 1980 had longer unemployment on average.

We can also use `group_by()` and `summarize()` to figure out group means before sending the data to ggplot. I prefer doing this because it gives me more control over the data that I'm plotting:

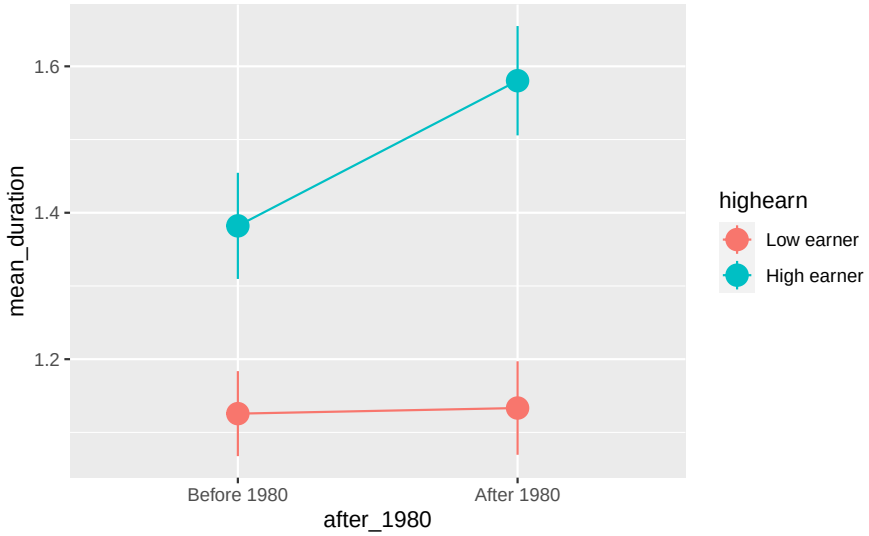
```
plot_data <- injury %>%
  # Make these categories instead of 0/1 numbers so they look nicer in the plot
  mutate(highearn = factor(highearn, labels = c("Low earner", "High earner")),
         after_1980 = factor(after_1980, labels = c("Before 1980", "After 1980"))) %>%
  group_by(highearn, after_1980) %>%
  summarize(mean_duration = mean(log_duration),
            se_duration = sd(log_duration) / sqrt(n()),
            upper = mean_duration + (1.96 * se_duration),
            lower = mean_duration + (-1.96 * se_duration))

ggplot(plot_data, aes(x = highearn, y = mean_duration)) +
  geom_pointrange(aes(ymin = lower, ymax = upper),
                  color = "darkgreen", size = 1) +
  facet_wrap(vars(after_1980))
```



Or, plotted in the more standard diff-in-diff format:

```
ggplot(plot_data, aes(x = after_1980, y = mean_duration, color = highearn)) +
  geom_pointrange(aes(ymin = lower, ymax = upper), size = 1) +
  # The group = highearn here makes it so the lines go across categories
  geom_line(aes(group = highearn))
```



Diff-in-diff by hand

We can find that exact difference by filling out the 2x2 before/after treatment/control table:

	Before 1980	After 1980	
Low earners	A	B	B − A
High earners	C	D	D − C
	C − A	D − B	(D − C) − (B − A)

A combination of `group_by()` and `summarize()` makes this really easy:

```
diffs <- injury %>%
  group_by(after_1980, highearn) %>%
  summarize(mean_duration = mean(log_duration),
            # Calculate average with regular duration too, just for fun
            mean_duration_for_humans = mean(duration))
diffs
## # A tibble: 4 x 4
## # Groups:   after_1980 [2]
##   after_1980 highearn mean_duration mean_duration_for_humans
##        <dbl>    <dbl>         <dbl>                    <dbl>
## 1          0        0          1.13                     6.27
## 2          0        1          1.38                    11.2
## 3          1        0          1.13                     7.04
## 4          1        1          1.58                    12.9
```

We can pull each of these numbers out of the table with some `filter()`s and `pull()`:

```
before_treatment <- diffs %>%
  filter(after_1980 == 0, highearn == 1) %>%
  pull(mean_duration)

before_control <- diffs %>%
  filter(after_1980 == 0, highearn == 0) %>%
  pull(mean_duration)

after_treatment <- diffs %>%
  filter(after_1980 == 1, highearn == 1) %>%
  pull(mean_duration)

after_control <- diffs %>%
```

```
  filter(after_1980 == 1, highearn == 0) %>%
  pull(mean_duration)

diff_treatment_before_after <- after_treatment - before_treatment
diff_treatment_before_after
## [1] 0.2

diff_control_before_after <- after_control - before_control
diff_control_before_after
## [1] 0.0077

diff_diff <- diff_treatment_before_after - diff_control_before_after
diff_diff
## [1] 0.19
```

The diff-in-diff estimate is 0.19, which means that the program causes an increase in unemployment duration of 0.19 logged weeks. Logged weeks is nonsensical, though, so we have to interpret it with percentages (here's a handy guide!; this is Example B, where the dependent/outcome variable is logged). Receiving the treatment (i.e. being a high earner after the change in policy) causes a 19% increase in the length of unemployment.

```
ggplot(diffs, aes(x = as.factor(after_1980),
                  y = mean_duration,
                  color = as.factor(highearn))) +
  geom_point() +
  geom_line(aes(group = as.factor(highearn))) +
  # If you use these lines you'll get some extra annotation lines and
  # labels. The annotate() function lets you put stuff on a ggplot that's not
  # part of a dataset. Normally with geom_line, geom_point, etc., you have to
  # plot data that is in columns. With annotate() you can specify your own x and
  # y values.
  annotate(geom = "segment", x = "0", xend = "1",
           y = before_treatment, yend = after_treatment - diff_diff,
           linetype = "dashed", color = "grey50") +
  annotate(geom = "segment", x = "1", xend = "1",
           y = after_treatment, yend = after_treatment - diff_diff,
           linetype = "dotted", color = "blue") +
  annotate(geom = "label", x = "1", y = after_treatment - (diff_diff / 2),
           label = "Program effect", size = 3)
```