Project 3

Nikant Yadav

2024-06-03

Measuring the effect of a Sugar Tax

This project deals with using the differences-in-differences method on the 2014 sugar tax in US to learn how before-and-after comparisons are done in practice.

In 2014, city of Berkeley in California implemented a tax on SSB distributors, with the aim of discouraging SSB consumption. We analyse the effects of this tax on different stakeholders, in the following project.

We will begin with defining two groups -

- *Treatment group:* those who were affected by the policy
- *Control group:* those who were not affected by the policy

First, we will look at the price data from the treatment group (stores in Berkeley) to see what happened to the price of sugary and non-sugary beverages after the tax.

We begin this project by setting the working directory and importing the data file.

```
suppressWarnings({
library(readxl)
library(tidyverse)

var_info <- read_excel("Dataset Project 3.xlsx", sheet="Data Dictionary")
dat <- read_excel("Dataset Project 3.xlsx", sheet = "Data")
})
```

Data Dictionary, contains some information about the variables, and Data, contains the actual data.

```
str(dat)
```

```
## tibble [2,175 × 12] (S3: tbl_df/tbl/data.frame)
##  $ store_id      : num [1:2175] 16 16 16 16 16 16 16 16 16 16 ...
##  $ type          : chr [1:2175] "WATER" "TEA" "TEA" "WATER" ...
##  $ store_type    : num [1:2175] 2 2 2 2 2 2 2 2 2 2 ...
##  $ type2         : chr [1:2175] NA NA NA NA ...
##  $ size          : num [1:2175] 33.8 23 23 33.8 128 64 128 64 63.9 144 ...
##  $ price         : num [1:2175] 1.69 0.99 0.99 1.69 3.79 2.79 3.79 2.79 4.59 5.99 ...
##  $ price_per_oz  : num [1:2175] 0.05 0.043 0.043 0.05 0.0296 ...
##  $ price_per_oz_c: num [1:2175] 5 4.3 4.3 5 2.96 ...
##  $ taxed         : num [1:2175] 0 1 1 0 0 0 0 0 0 1 ...
##  $ supp          : num [1:2175] 0 0 0 0 0 0 0 0 0 1 ...
##  $ time          : chr [1:2175] "DEC2014" "DEC2014" "DEC2014" "DEC2014" ...
##  $ product_id    : num [1:2175] 29 32 33 38 40 41 42 43 44 50 ...
```

,

How the product information was recorded? Three Store Price Surveys (SPS) were conducted both before and after the implementation of the tax to assess the price of beverages in a sample of stores frequently visited by participants. The sample was diversified to ensure that the representation of non-participation was included in it. Prices of top-selling beverages were collected by trained collectors, supplemented by similar products where needed. Prices were collected in a systematic manner on all features of the data. They were entered into a database with great care, through a tablet and back-up paper forms, employing double entry and comparison to minimize errors. In addition, stores with listed beverages were added to the supplementary options. Generally, the surveys were well done and incorporated the price changes amidst the tax adjustments, clearly enabling the data to be retained with integrity.

Verifying that the number of stores in the dataset is the same as that stated in the 'S1 Text' (26)

In the dataset, each store has been given a unique id (store_id).

```
temp <- list()
count <- 1

for (i in 1:(length(dat$store_id) - 1)) {
  if (dat$store_id[i + 1] != dat$store_id[i]) {
    unique_store <- TRUE
    if (length(temp) > 0) {
      for (j in 1:length(temp)) {
        if (dat$store_id[i + 1] == temp[[j]]) {
          unique_store <- FALSE
          break
        }
      }
    }
    if (unique_store) {
      temp[[count]] <- dat$store_id[i]
      count <- count + 1
    }
  }
}


print(count-2)
```

```
## [1] 26
```

We use a simple algorithm to calculate the number of unique store ids. We define a variable *count* with an initial value of 1. For every iteration where *unique_store* variable attains the value *TRUE*, count increases by one. The algorithm basically compares elements within *dat* $store_id * and also compares the elements of * (dat$store_id)$ and *temp* to increase the count every time a unique element is found.

A simpler way to achieve this task is using the inbuilt functions of R:

```
no_stores <- length(unique(dat$store_id))
no_products <- length(unique(dat$product_id))

print(no_stores )
```

```
## [1] 26
```

```
print(no_products)
```

```
## [1] 247
```

Before proceeding with the analysis, we will use summary measures to see how many observations are in the treatment and control group, and how the two groups differ across some variables of interest.

Frequency table showing the number (count) of store observations in December 2014 and June 2015

We start with the frequency table that shows the number of stores of different types in each time period.

```
library(mosaic)
tally(~store_type + time, data = dat, margins = TRUE, format = "count")
```

```
##           time
## store_type DEC2014 JUN2015 MAR2015 Total
##      1         177     209     158   544
##      2         407     391     327  1125
##      3          87     102      73   262
##      4          73      96      75   244
##      Total     744     798     633  2175
```

We use the tally function of the mosaic library, which creates the frequency table.

Another way to create such table:

```
library(vcd)
structable(store_type ~ time, data=dat )
```

```
##         store_type   1   2   3   4
## time
## DEC2014            177 407  87  73
## JUN2015            209 391 102  96
## MAR2015            158 327  73  75
```

This method, uses the *structable* function of the *vcd* library of R. While both functions can be used for summarizing categorical data, structable is more focused on creating structured contingency tables, while tally is more flexible and can be integrated into data manipulation pipelines

Frequency table showing the number of taxed and non-taxed beverages in December 2014 and June 2015

We will proceed with using the *tally* function.

```
tally(~store_type + taxed + time, data = dat, margins = TRUE, format = "count")
```

```
## , , time = DEC2014
##
##            taxed
## store_type    0    1 Total
##      1       92   85   177
##      2      196  211   407
##      3       44   43    87
##      4       34   39    73
##      Total  366  378   744
##
## , , time = JUN2015
##
##            taxed
## store_type    0    1 Total
##      1      111   98   209
##      2      192  199   391
##      3       52   50   102
##      4       44   52    96
##      Total  399  399   798
##
## , , time = MAR2015
##
##            taxed
## store_type    0    1 Total
##      1       88   70   158
##      2      154  173   327
##      3       36   37    73
##      4       31   44    75
##      Total  309  324   633
##
## , , time = Total
##
##            taxed
## store_type    0    1 Total
##      1      291  253   544
##      2      542  583  1125
##      3      132  130   262
##      4      109  135   244
##      Total 1074 1101  2175
```

taxed 0 means - not taxed taxed 1 means - taxed In both periods, the number of taxed and non-taxed beverages is almost similar for each store type.

Frequency table showing the number of each product type

```
tally(~type + time, data = dat, margins = TRUE, format = "count")
```

```
##              time
## type          DEC2014 JUN2015 MAR2015 Total
##   ENERGY           56      58      49   163
##   ENERGY-DIET      49      54      35   138
##   JUICE            70      64      52   186
##   JUICE DRINK      19      17       6    42
##   MILK             63      61      53   177
##   SODA            239     262     215   716
##   SODA-DIET       128     174     127   429
##   SPORT            11      16      12    39
##   SPORT-DIET        2       2       0     4
##   TEA              52      45      41   138
##   TEA-DIET          6       6       8    20
##   WATER            48      38      34   120
##   WATER-SWEET       1       1       1     3
##   Total           744     798     633  2175
```

SODA type products have the highest number of occurrences, whereas WATER-SWEET has the lowest. More popular type beverages tend to have more occurrences, because they are available at more stores.

Conditional Mean

An average of a variable, taken over a subgroup of observations that satisfy certain conditions, rather than all observations.

The average price of taxed and non-taxed beverages, according to time period and store type.

```
dat$period_test <- NA

sid_list = unique(dat$store_id)

pid_list = unique(dat$product_id)

for (s in sid_list) {
  for (p in pid_list) {
    temp <- subset(dat, product_id == p & store_id == s)
    temp_time <- temp$time
    test <- (
      any(temp_time == "DEC2014") ||
      any(temp_time == "JUN2015") ||
      any(temp_time == "MAR2016"))
    dat$period_test[dat$product_id == p &
      dat$store_id == s] <- test
  }
}
```

So what we're going to do is create a new variable called *period_test*. We'll want to check if each product in each store has observations for all three periods (DEC2014, JUN2015 and MAR2016). So we'll iterate over each store and each product ID. If observations exist for all three periods for a specific product in a specific store, we'll assign *period_test=1* (or TRUE). Otherwise, we'll assign it 0 (or FALSE). This new variable will act as a boolean indicator showing whether each product in each store has data for all three periods.

```
dat_c <- subset(dat, (period_test == TRUE & supp == 0))
```

We create a new data frame to remove all products that have not been observed in all three periods.

Now we can calculate the means of *price_per_oz_c* by grouping the data according to store_type, taxed, and time using a method called **piping**:

```
table_res <- dat_c %>%
  group_by(taxed, store_type, time) %>%
  summarize(n = length(price_per_oz_c),
    avg.price = mean(price_per_oz_c)) %>%
  spread(time, avg.price) %>%
  print()
```

```
## # A tibble: 24 × 6
## # Groups:   taxed, store_type [8]
##    taxed store_type     n DEC2014 JUN2015 MAR2015
##    <dbl>      <dbl> <int>   <dbl>   <dbl>   <dbl>
##  1     0          1    50    NA      NA      11.1
##  2     0          1    51    10.8    NA      NA
##  3     0          1    62    NA      11.5    NA
##  4     0          2   109    NA      NA      11.6
##  5     0          2   167    12.2    NA      NA
##  6     0          2   169    NA      12.7    NA
##  7     0          3    28    NA      NA      14.7
##  8     0          3    29    12.3    NA      NA
##  9     0          3    36    NA      12.4    NA
## 10     0          4    26    NA      NA      13.7
## # ℹ 14 more rows
```

PART 2

It is also possible that the changes in Berkeley were not solely due to tax, but instead were also influenced by other events that happened in Berkeley and in the neighboring areas. We do the differences-in-differences analysis, using:

The treatment group: Beverages in Berkeley The control group: Beverages in surrounding areas

Based on the S5 Table (https://tinyco.re/7724734) it can be easily judged that the researchers chose suitable comparison stores, because almost all parameters for both experiment population are same.

We use a separate data set containing price data including information on date, location (Berkeley or Non-Berkeley), beverage group and the average price for that month.

```
library(readstata13)
PoSd <- read.dta13("public_use_weighted_prices2.dta")
str(PoSd)
```

```
## 'data.frame':    2728 obs. of  8 variables:
##  $ year          : num  2013 2013 2013 2013 2013 ...
##  $ quarter       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ month         : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ location      : chr  "Berkeley" "Non-Berkeley" "Non-Berkeley" "Berkeley" ...
##  $ beverage_group: chr  "soda" "soda" "soda" "soda" ...
##  $ tax           : chr  "Non-taxed" "Non-taxed" "Non-taxed" "Taxed" ...
##  $ price         : num  4.85 3.51 3.89 3.68 3.52 ...
##  $ under_report  : num  NA NA NA NA NA NA NA NA NA NA ...
##  - attr(*, "datalabel")= chr ""
##  - attr(*, "time.stamp")= chr "17 Feb 2017 13:48"
##  - attr(*, "formats")= chr [1:8] "%12.0g" "%12.0g" "%12.0g" "%12s" ...
##  - attr(*, "types")= int [1:8] 65526 65526 65526 12 28 9 65526 65526
##  - attr(*, "val.labels")= Named chr [1:8] "" "" "" "" ...
##   ..- attr(*, "names")= chr [1:8] "" "" "" "" ...
##  - attr(*, "var.labels")= chr [1:8] "" "" "" "" ...
##  - attr(*, "version")= int 118
##  - attr(*, "label.table")= list()
##  - attr(*, "expansion.fields")= list()
##  - attr(*, "byteorder")= chr "LSF"
##  - attr(*, "orig.dim")= int [1:2] 2728 8
##  - attr(*, "data.label")= chr(0)
```

For each month and location (Berkeley or Non-Berkeley), there are prices for a variety of beverage categories, and we know whether the category is taxed or not.

Average price in each month for taxed and non-taxed beverages, according to location.

We use the piping method to create a summary table:

```
table_test <- PoSd %>%
  group_by(year, month, location, tax) %>%
  summarize(avg.price = mean(price)) %>%
  spread(location, avg.price) %>%
  print()
```

```
## # A tibble: 78 × 5
## # Groups:   year, month [39]
##     year month tax       Berkeley `Non-Berkeley`
##    <dbl> <dbl> <chr>        <dbl>          <dbl>
##  1  2013     1 Non-taxed     5.72           5.35
##  2  2013     1 Taxed         8.69           7.99
##  3  2013     2 Non-taxed     5.81           5.36
##  4  2013     2 Taxed         8.65           8.18
##  5  2013     3 Non-taxed     5.86           5.42
##  6  2013     3 Taxed         8.82           8.19
##  7  2013     4 Non-taxed     5.86           5.64
##  8  2013     4 Taxed         9.02           8.25
##  9  2013     5 Non-taxed     5.79           5.18
## 10  2013     5 Taxed         8.68           7.76
## # ℹ 68 more rows
```

This table shows average price in each month for taxed and non-taxed beverages, according to their location. Let’s refine this a little

```
tax_table <- subset(table_test, tax == "Taxed")
ntax_table <- subset(table_test, tax == "Non-taxed")
```

```
print(tax_table)
```

```
## # A tibble: 39 × 5
## # Groups:   year, month [39]
##     year month tax   Berkeley `Non-Berkeley`
##    <dbl> <dbl> <chr>    <dbl>          <dbl>
##  1  2013     1 Taxed     8.69           7.99
##  2  2013     2 Taxed     8.65           8.18
##  3  2013     3 Taxed     8.82           8.19
##  4  2013     4 Taxed     9.02           8.25
##  5  2013     5 Taxed     8.68           7.76
##  6  2013     6 Taxed     8.57           7.43
##  7  2013     7 Taxed     8.23           7.19
##  8  2013     8 Taxed     8.82           7.49
##  9  2013     9 Taxed     9.02           7.80
## 10  2013    10 Taxed     8.98           7.75
## # ℹ 29 more rows
```

```
print(ntax_table)
```

```
## # A tibble: 39 × 5
## # Groups:   year, month [39]
##     year month tax       Berkeley `Non-Berkeley`
##    <dbl> <dbl> <chr>        <dbl>          <dbl>
##  1  2013     1 Non-taxed     5.72           5.35
##  2  2013     2 Non-taxed     5.81           5.36
##  3  2013     3 Non-taxed     5.86           5.42
##  4  2013     4 Non-taxed     5.86           5.64
##  5  2013     5 Non-taxed     5.79           5.18
##  6  2013     6 Non-taxed     5.76           5.03
##  7  2013     7 Non-taxed     5.90           5.10
##  8  2013     8 Non-taxed     5.83           5.08
##  9  2013     9 Non-taxed     5.83           5.08
## 10  2013    10 Non-taxed     5.80           5.18
## # ℹ 29 more rows
```

We create two separate tables for taxed and non-taxed beverages.

Plotting Line Chart

First, we convert the tables to timeseries data, that helps in easy plotting of the graphs.

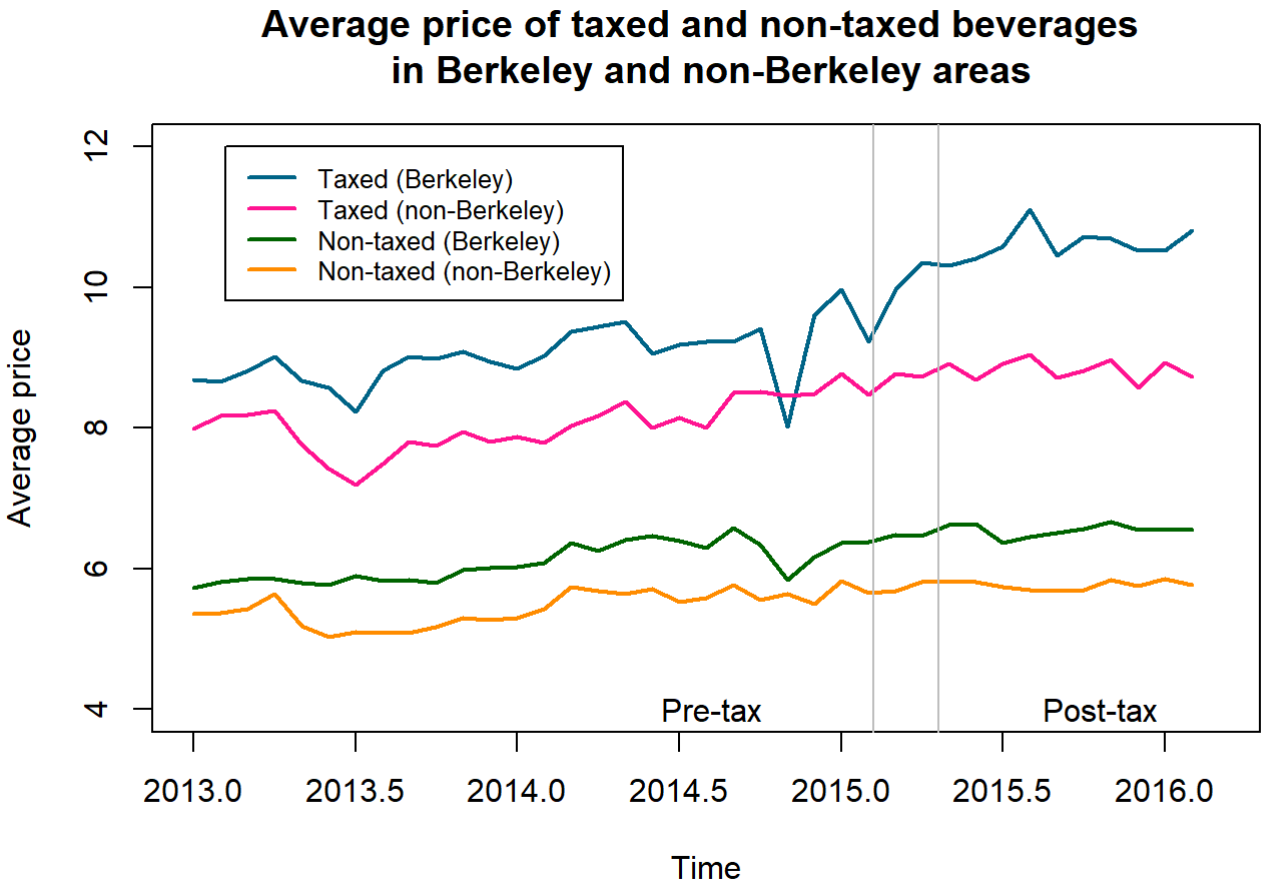
```
tax_table$Berkeley <- ts(tax_table$Berkeley,
  start = c(2013, 1), end = c(2016, 3), frequency = 12)
tax_table$'Non-Berkeley' <- ts(tax_table$'Non-Berkeley',
  start = c(2013, 1), end = c(2016, 3), frequency = 12)
ntax_table$Berkeley <- ts(ntax_table$Berkeley,
  start = c(2013, 1), end = c(2016, 3), frequency = 12)
ntax_table$'Non-Berkeley' <- ts(ntax_table$'Non-Berkeley',
  start = c(2013, 1), end = c(2016, 3), frequency = 12)
```

Next, we use the *plot* and the *line* function to plot the graphs.

```
plot(tax_table$Berkeley, col = "deepskyblue4", lwd = 2,
  ylab = "Average price", xlab = "Time", ylim = c(4, 12))
lines(tax_table$Berkeley, col = "deepskyblue4", lwd = 2)

title("Average price of taxed and non-taxed beverages \n in Berkeley and non-Berkeley areas")
lines(tax_table$'Non-Berkeley',
  col = "deeppink", lwd = 2)
lines(ntax_table$Berkeley,
  col = "darkgreen", lwd = 2)
lines(ntax_table$'Non-Berkeley',
  col = "darkorange", lwd = 2)
abline(v = 2015.1, col = "grey")
abline(v = 2015.3, col = "grey")

text(2014.6, 4, "Pre-tax")
text(2015.8, 4, "Post-tax")
legend(2013.1, 12, lwd = 2, lty = 1, cex = 0.8,
  legend = c("Taxed (Berkeley)", "Taxed (non-Berkeley)",
    "Non-taxed (Berkeley)", "Non-taxed (non-Berkeley)"),
  col = c("deepskyblue4", "deeppink",
    "darkgreen", "darkorange"))
```



Observation The following things can be observed from this graph:

- Non-taxed goods in Berkeley are more expensive than those outside Berkeley.
- The average price trend of non-taxed beverages in Berkeley and Non-Berkeley shops, remain almost same in the pre-tax and the post-tax period

- For the taxed beverages, the difference in average price increases in Berkeley region in the post-tax period.

The difference in prices after the implementation of the tax, subtracting the difference in prices before the tax, gives us the effect of the tax.

p-value analysis According to the journal paper, when comparing the mean Berkeley and non-Berkeley price of sugary beverages after the tax, the p-value is smaller than 0.00001, and it is 0.63 for non-sugary beverages after the tax.

Case Study

Suppose that you have the authority to conduct your own sugar tax natural experiment in two neighboring towns, Town A and Town B. Outline how you would conduct the experiment to ensure that any changes in outcomes (prices, consumption of sugary beverages) are due to the tax and not due to other factors.

Solution: The two towns would be chosen, such that the features of their populations are almost same. Also, neither of the two towns should have been exposed to media/advertising campaigns to reduce the usage of SSBs.

A larger, controlled sample, optimally with higher SSB consumption will be considered, to reduce the standard error.It will be ensured that there are no other policy changes that could affect the outcomes in the observation period