2. Introduction to causality

Frank Edwards

9/11/2019

Causality

The canonical question in causal inference

- Does treatment x affect outcome y
- In medicine, does a treatment affect a patient
- Typically designed by randomly assigning patients to treatment and control groups, where treatment groups are exposed to x, and control groups are not

The fundamental problem of causal inference

How much did the treatment matter?

We answer this question with counterfactuals:

What would have happened if treated units were untreated? What would have happened if untreated units were treated?

For an observation i, where $Y_i(1)$ indicates treatment and $Y_i(0)$ indicates no treatment, the causal effect of the treatment is defined as

$Y_i(1) - Y_i(0)$

The fundamental problem of causal inference

How much did the treatment matter?

We answer this question with counterfactuals:

What would have happened if treated units were untreated? What would have happened if untreated units were treated?

For an observation *i*, where $Y_i(1)$ indicates treatment and $Y_i(0)$ indicates no treatment, the causal effect of the treatment is defined as

$Y_i(1) - Y_i(0)$

Why is this a problematic definition?

Causal questions in social science

- Does race impact hiring decisions?
    - A Black candidate applied for a job, but did not get it.
    - Would a Black candidate have been offered a job if they were white?
- Does the minimum wage increase unemployment?
    - Unemployment went up in a city after the minimum wage increased
    - Would unemployment have gone up were there not an increase in the minimum wage?
- Does community policing decrease crime?
    - A police department implemented community policing in certain neighborhoods, and reported crime went down
    - Would reported crime have gone down without community policing?

Experimental research

Evaluates how treatments causally effect outcomes by assigning different levels of treatment to different observations, then measuring the corresponding values of the outcome

Using an experiment to estimate the effects of a criminal record on employment

Pager, Devah. “The mark of a criminal record.” American journal of sociology 108.5 (2003): 937-975.

> *With over 2 million individuals currently incarcerated, and over half a million prisoners released each year, the large and growing number of men being processed through the criminal justice system raises important questions about the consequences of this massive institutional intervention. This article focuses on the consequences of incarceration for the employment outcomes of black and white job seekers. The present study adopts an experimental audit approach—in which matched pairs of individuals applied for real entry‐level jobs—to formally test the degree to which a criminal record affects subsequent employment opportunities. The findings of this study reveal an important, and much underrecognized, mechanism of stratification. A criminal record presents a major barrier to employment, with important implications for racial disparities.*

Research questions

1. Do employers use criminal histories to make hiring decisions?
2. Is racial discrimination a major barrier to employment?
3. Does the effect of a criminal record differ for white and Black applicants?

What counterfactuals are needed for each question?

1. Do employers use criminal histories to make hiring decisions?

What counterfactuals are needed for each question?

1. Do employers use criminal histories to make hiring decisions?
2. Is racial discrimination a major barrier to employment?

What counterfactuals are needed for each question?

1. Do employers use criminal histories to make hiring decisions?
2. Is racial discrimination a major barrier to employment?
3. Does the effect of a criminal record differ for white and Black applicants?

Before we begin

- git pull to grab the data file and slides

```
dat<-read_csv("./data/criminalrecord.csv")
```

Variables in the data

`jobid` Job ID number

`callback` 1 if tester received a callback, 0 if the tester did not receive a callback.

`black` 1 if the tester is black, 0 if the tester is white.

`crimrec` 1 if the tester has a criminal record, 0 if the tester does not.

`interact` 1 if tester interacted with employer during the job application, 0 if tester does not interact with employer.

`city` 1 is job is located in the city center, 0 if job is located in the suburbs.

`distance` Job's average distance to downtown.

`custserv` 1 if job is in the costumer service sector, 0 if it is not.

`manualskill` 1 if job requires manual skills, 0 if it does not.

Take a look at the data

```
glimpse(dat)
```

```
## Observations: 696
## Variables: 9
## $ jobid       <dbl> 108, 113, 101, 64, 33, 73, 4, 125, 8
## $ callback    <dbl> 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
## $ black       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ crimrec     <dbl> 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1,
## $ interact    <dbl> 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
## $ city        <dbl> 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1,
## $ distance    <dbl> 15, 20, 15, 7, 5, 10, 17, 15, 3, 16,
## $ custserv    <dbl> 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
## $ manualskill <dbl> 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1,
```

Exploring the data: univariate crosstabs

```
table(dat$black)
```

```
##
##   0   1
## 300 396
```

```
table(dat$crimrec)
```

```
##
##   0   1
## 349 347
```

Exploring the data: bivariate crosstabs

```r
table(dat$black, dat$crimrec)
```

```
##
##       0   1
##   0 150 150
##   1 199 197
```

```r
table(dat$black == 1, dat$callback)
```

```
##
##           0   1
##   FALSE 224  76
##   TRUE  358  38
```

Summary tables

```r
dat %>%
  group_by(black) %>%
  summarise(cases = n())
```

```
## # A tibble: 2 x 2
##   black cases
##   <dbl> <int>
## 1     0   300
## 2     1   396
```

Summary tables are very flexible

```r
dat %>%
  group_by(black) %>%
  summarise(cases = n(),
            callback=sum(callback),
            callback_pct = sum(callback)/n())
```

```
## # A tibble: 2 x 4
##   black cases callback callback_pct
##   <dbl> <int>    <dbl>        <dbl>
## 1     0   300       76       0.253
## 2     1   396       38       0.0960
```

We can also use multiple variables for grouping

```r
dat %>%
  group_by(black, crimrec) %>%
  summarise(cases = n(),
            callback=sum(callback),
            callback_pct = sum(callback)/n())
```

```
## # A tibble: 4 x 5
## # Groups:   black [2]
##   black crimrec cases callback callback_pct
##   <dbl>   <dbl> <int>    <dbl>        <dbl>
## 1     0       0   150       51       0.34
## 2     0       1   150       25       0.167
## 3     1       0   199       28       0.141
## 4     1       1   197       10       0.0508
```

Subsetting and an aside on logicals

Logicals in R

```r
temp<-c(TRUE, FALSE, TRUE)
str(temp)
```

```
##  logi [1:3] TRUE FALSE TRUE
```

```r
sum(temp)
```

```
## [1] 2
```

```r
mean(temp)
```

```
## [1] 0.6666667
```

Logical operators

```r
## AND: &
TRUE & FALSE
```

```
## [1] FALSE
```

```r
## OR: |
TRUE | FALSE
```

```
## [1] TRUE
```

```r
## NOT: !
!TRUE
```

```
## [1] FALSE
```

That’s neat (but kinda useless?)

It is very useful!

```
2<3
```

```
## [1] TRUE
```

```
2<3 & 2>3
```

```
## [1] FALSE
```

```
2<3 | 2>3
```

```
## [1] TRUE
```

```
!(2<3)
```

```
## [1] FALSE
```

We often use logicals in conjunction with comparisons

- < and > less than and greater than
- <= and >= less/greater than or equal to
- == equal to
- != not equal to
- %in% element in vector

Examples

```
temp<-c(2,3,4,5)
3<temp
```

```
## [1] FALSE FALSE  TRUE  TRUE
```

```
3==temp
```

```
## [1] FALSE  TRUE FALSE FALSE
```

```
3%in%temp
```

```
## [1] TRUE
```

Let’s use these to subset

```
## Note that recoding here is not needed
## For convenience, make dat$black a logical
dat<-dat %>%
  mutate(black = black==1)
head(dat)
```

```
## # A tibble: 6 x 9
##   jobid callback black crimrec interact  city distance
##   <dbl>    <dbl> <lgl>   <dbl>    <dbl> <dbl>    <dbl>
## 1   108        1 FALSE       1        1     0       15
## 2   113        0 FALSE       0        1     0       20
## 3   101        1 FALSE       0        0     0       15
## 4    64        1 FALSE       0        0     1        7
## 5    33        0 FALSE       1        0     1        5
## 6    73        0 FALSE       1        0     1       10
```

Use this variable to subset the data into Black/white applicants

```
dat_blk<-dat %>%
  filter(black==TRUE)
dat_wht<-dat %>%
  filter(black!=TRUE)

nrow(dat_blk)
```

```
## [1] 396
```

```
nrow(dat_wht)
```

```
## [1] 300
```

```
nrow(dat)
```

```
## [1] 696
```

Let’s subset into race/crimrec datasets

```
dat_blk_crim<-dat_blk %>%
  filter(black==TRUE & crimrec==1)
dat_wht_crim<-dat_wht %>%
  filter(black==FALSE&crimrec>0)
nrow(dat_blk_crim)
```

```
## [1] 197
```

```
nrow(dat_wht_crim)
```

```
## [1] 150
```

Questions on logicals and filters?

Recoding and conditionals

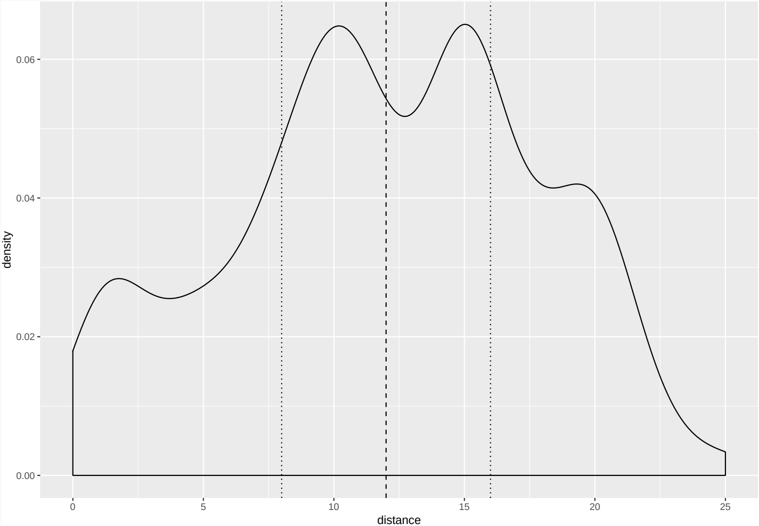
Let’s make distance categorical, with cuts at the 25th, 50th, and 75th quantile

```
summary(dat$distance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    8.00   12.00   11.96   16.00   25.00       2
```

```
## remove pesky NA values
dat<-dat %>% filter(!(is.na(distance)))
```

Visualizing quantiles: remember area under the curve?



Making a recode with one condition: ifelse()

Make a new variable for distance, with value “near” if below the median, and “far” if above

```r
dat <- dat %>%
  mutate(distance_binary = ifelse(
    distance<median(distance), # CONDITION
    "near", # IF TRUE
    "far")) # IF FALSE
table(dat$distance_binary)
```

```
##
##  far near
##  370  324
```

Making a recode with multiple conditions: case_when()

```
dat <- dat %>%
  mutate(distance_cat =
           case_when(
             distance < quantile(distance, 0.25) ~ "very cl
             distance < quantile(distance, 0.5)  ~ "close",
             distance < quantile(distance, 0.75) ~ "kinda f
             distance >= quantile(distance, 0.75) ~ "super
           ))
table(dat$distance_cat)
```

```
##
##      close  kinda far  super far very close
##        176        184        186        148
```

Using factors to re-order things

```r
dat<-dat %>%
  mutate(distance_cat = factor(distance_cat,
                         levels = c(
                           "very close",
                           "close",
                           "kinda far",
                           "super far"
                         )))
table(dat$distance_cat)
```

```
##
## very close      close  kinda far  super far
##        148        176        184        186
```

Returning to Pager’s experiment

The counterfactual and potential outcomes

```
## # A tibble: 694 x 3
##    crimrec callback_crimTRUE callback_crimFALSE
##      <dbl>             <dbl>              <dbl>
##  1       1                 1                 NA
##  2       0                NA                  0
##  3       0                NA                  1
##  4       0                NA                  1
##  5       1                 0                 NA
##  6       1                 0                 NA
##  7       1                 0                 NA
##  8       1                 1                 NA
##  9       0                NA                  0
## 10       1                 0                 NA
## # ... with 684 more rows
```

The causal effect

For observation *i* is equal to callback_crimTRUE_i - callback_crimFALSE_i

The fundamental problem of causal inference is that we only observe one of these outcomes

Randomized experiments (or RCTs)

- By randomizing assignment to treatment, we can treat units as equivalent
- If units are equivalent, we can estimate the average treatment effect as a difference in means on the outcome between the treatment and control group
- If we don’t randomize, we have no assurance that the treated and control groups are equivalent, meaning we don’t have a strong case that we’ve observed the counterfactual

The SATE for Pager's experiment

We assume that we can estimate the counterfactual for people with criminal records (i.e. no criminal record), by using the mean value of the callback outcome for people assigned to have no criminal record.

```r
dat_crimrecT <- dat %>%
  filter(crimrec==1) %>%
  summarise(callback = mean(callback))
dat_crimrecF <- dat %>%
  filter(crimrec==0) %>%
  summarise(callback = mean(callback))

dat_crimrecT$callback - dat_crimrecF$callback
```

```
## [1] -0.1287456
```

Final results of the analysis

```r
dat %>%
  group_by(black, crimrec) %>%
  summarise(callback = mean(callback))
```

```
## # A tibble: 4 x 3
## # Groups:   black [2]
##   black crimrec callback
##   <lgl>   <dbl>    <dbl>
## 1 FALSE       0   0.34
## 2 FALSE       1   0.167
## 3 TRUE        0   0.141
## 4 TRUE        1   0.0459
```

Next week

- Homework: More data frame practice. Use tidyverse commands like mutate(), filter() and summarise() to complete exercise 1.5.2. Ignore instructions to store objects as vectors if you like. Due 9/17 at noon.
- Causality, part 2. Observational studies
- Measuring characteristics of the distribution of a variable