Recitation 2: Introduction to Research Methods for Politics

Dept. of Politics, NYU

POL-850

Spring 2020

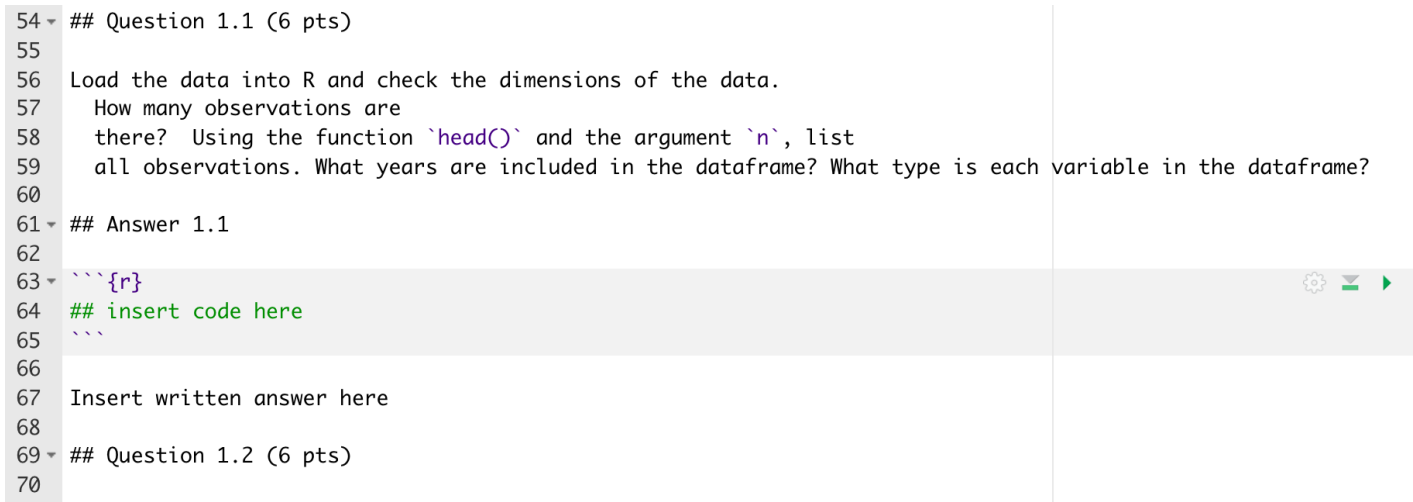
Reminder

Homework 1 due this Wednesday (2.12), 5 pm, in my mailbox:

- Only typed work (pdf file generated by R Markdown), no handwriting.
- When answering a question, please include related R code.

How to use the template

1. Put your code in the R code chunk (gray chunk)



```
54 ## Question 1.1 (6 pts)
55
56 Load the data into R and check the dimensions of the data.
57   How many observations are
58   there?  Using the function `head()` and the argument `n`, list
59   all observations. What years are included in the dataframe? What type is each variable in the dataframe?
60
61 ## Answer 1.1
62
63 ```{r}
64 ## insert code here
65 ```
66
67 Insert written answer here
68
69 ## Question 1.2 (6 pts)
70
```

2. Comment your code using #

Your Friends to Work in R (1)

Coding can be exhausting, but not if you get help from some friends:

- The code I use during recitation. You can find it on NYU Classes, and use it as a baseline for your own code.
- Google: a query like "How to compute mean in R" will return dozens of results, with practical examples and code snippets. **Ideal when you do not know how to do something.**

Your Friends to Work in R (2)

Coding can be exhausting, but not if you get help from some friends:

- R, by using the question mark: `?mean`. **Ideal when you know a function exists, but you are not sure about the details.**
- The TAs!

Agenda

1. Review: Causal Effects
2. Relational Operators
3. Basic functions (ifelse, subset)

Causal Effects

Causal Effects

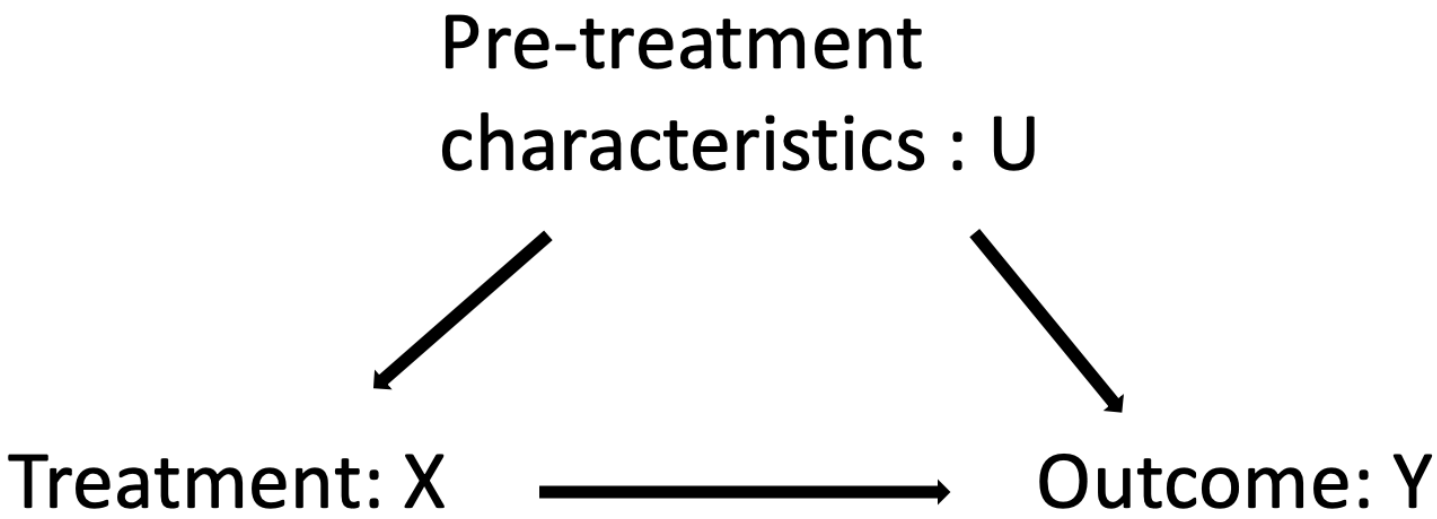
1. We are interested in : Causal Effect of X on Y
   (e.g. Class size $\rightarrow$ Student performance?
   Democracy $\rightarrow$ Economic development ?)
2. Potential outcome if individual i in the treatment group: $Y_i(1)$
   Potential outcome if individual i in the control group: $Y_i(0)$
3. Individual causal effect: $\Delta Y_i = Y_i(1) - Y_i(0)$
4. Fundamental Problem of Causal Inference:
   "To measure causal effects, we need to compare the factual outcome with the counterfactual outcome, but we can never observe the counterfactual outcome."

Causal Effects

1. Average causal effect: $\overline{\Delta Y} = \overline{Y(1)} - \overline{Y(0)}$
   where $\overline{Y(1)}$ and $\overline{Y(0)}$ are the average outcomes for *all* observations.
2. We still face the missing data problem; Can you see why?
3. We can estimate it by: $\widehat{\overline{\Delta Y}} = \widehat{\overline{Y(1)}} - \widehat{\overline{Y(0)}}$
   where $\widehat{\overline{Y(1)}}$ and $\widehat{\overline{Y(0)}}$ are the average outcomes for *observed* units.
4. Are $\overline{Y(1)}$ and $\widehat{\overline{Y(1)}}$ the same? If not, when can we say $\widehat{\overline{\Delta Y}}$ is a good estimate for $\overline{\Delta Y}$?

Causal Effects

1. How to define "a good estimate"? In the textbook, authors say $\widehat{\overline{\Delta Y}}$ is unbiased.
2. How to guarantee "unbiased"?



Relational Operators

Why Bother?

Our ultimate goal as empiricists is to (credibly) compare units under treatment status ($T = 1$) and control status ($T = 0$).

- This requires splitting our data based on T!

This way, we will be able to:

- Compare outcomes: is $Y(1) = Y(0)$?
- Check that units are comparable based on observables: is $X(1) = X(0)$?
- And much more...

Behind Subsetting: Logical Operators

To subset, we need R to understand whether a given condition is satisfied (TRUE) or not (FALSE). TRUE and FALSE belong to a new class of objects in R, the one of *logicals*.

```
 5  class(TRUE)
 6
 7  as.integer(TRUE)
 8  as.integer(FALSE)
 9
10  x <- c(TRUE, FALSE, TRUE) # a vector with logical values
11
12  mean(x) # proportion of TRUEs
13  sum(x) # number of TRUEs
14
15  FALSE & TRUE
16  TRUE & TRUE
17  TRUE | FALSE
18  FALSE | FALSE
19  TRUE & FALSE & TRUE
20
```

NOTE: Use "AND" ("&") and "OR" ("|") to evaluate multiple conditions.

Logical Operators to Evaluate Relationships

```
32  4 == 4
33  4 > 3
34
35  x <- c(3, 2, 1, -2, -1)
36  x == -1
37  x >= 2
38  x != 1
39
40  ## logical conjunction of two vectors with logical values
41  (x > 0) & (x <= 2)
42
43  ## logical disjunction of two vectors with logical values
44  (x > 2) | (x <= -1)
```

NOTE: The relational operator "! =" means "not equal to".

Practice with STAR datset

```
49  star <- read.csv("STAR.csv")
50  head(star, n = 10)  #### show first 10 rows
51
52  star$classtype == "small"
53
54  sum(star$classtype == "small")/nrow(star)
55  mean(star$classtype == "small")
56
57  tmp1 <- (star$classtype == "small") & (star$reading >= 600)
58  tmp2 <- (star$classtype == "regular") & (star$reading >= 600)
59
60  mean(tmp1)
61  mean(tmp2)
```

Basic Functions: ifelse and subset

Introduction: ifelse

Syntax: ifelse(condition, return values if yes, return values if no)

```
70  ifelse(star$classtype == "regular", 1, 0)
71
72  star$regular <- ifelse(star$classtype == "regular", 1, 0) ## create a new var
73  head(star, n=4)
74
75  star$treatment <- ifelse(star$classtype == "small", 1, 0) ## create a new var
76
77  mean(star$math)
78  mean(star$math[star$treatment == 1])  ## mean for treatment group
79
80  #### difference-in-means estimators
81  mean(star$math[star$treatment==1]) - mean(star$math[star$treatment==0])
82  mean(star$reading[star$treatment==1]) - mean(star$reading[star$treatment==0])
83  mean(star$graduated[star$treatment==1]) - mean(star$graduated[star$treatment==0])
```

Introduction: subset

Syntax: subset(data.frame, condition)

```
88  subset(star, classtype == "small")
89
90  ## sub dataset only contains small class
91  star_small <- subset(star, classtype == "small")
92  star_small
93
94  star_regular <- subset(star, classtype == "regular")
95
96  ### now, we can use this new data.frame
97  mean(star_small$math) - mean(star_regular$math)
98  mean(star_small$reading) - mean(star_regular$reading)
99  mean(star_small$graduated) - mean(star_regular$graduated)
```

Introduction: subset

We can specify multiple conditions to the function subset.

```
103 star_small_grad <- subset(star,
104                               subset = (classtype == "small" & graduated == 1))
105 star_small_nongrad <- subset(star,
106                               subset = (classtype == "small" & graduated == 0))
107
108 mean(star_small_grad$reading) - mean(star_small_nongrad$reading)
109 mean(star_small_grad$math) - mean(star_small_nongrad$math)
110
```