17.835: Problem Set 2

Professor: In Song Kim
TA: Jesse Clark, Sean Shiyao Liu, and Nicole Wilson

Due: Wednesday September 23, 11:00 AM
(Late submissions will not be accepted)

Please post all questions to the Piazza discussion forum, which you can access through the link on the class website or at:

https://piazza.com/mit/fall2020/17835

For this and future problem sets, you will need to submit two files onto the Canvas (under `Assignment`): 1) your write-up with your answers to all questions (including codes and results for computational questions and any related graphics) as a PDF file, and 2) your code as an `R` file (so that we can verify it runs without errors). Both files should be identified with your last name (e.g., `clark.R` and `clark.pdf`). Please ensure that all of these are completed *before* class on the day of the due date – late submissions will be automatically flagged to us by Canvas. For problems that require calculations, please show all the steps of your work. For problems involving `R`, please ensure that your code is thoroughly commented.

This week's problems focus on different forms of inference. As we progress through the problems, you will learn about observational data and experimental data and how matching algorithms can help us make inferences that approximate experimental designs. For all the plots in this problem set, you may use the base R graphics or the `ggplot2()` library in R (more information on `ggplot` can be found here `http://ggplot2.tidyverse.org/`). Another package you may want to install and use for this problem set is `matrixStats`. You can do the whole problem set, however, using base `R`. That is, without `ggplot2` or `matrixStats`.

Problem 1: Media Control and Electoral Advantage

In this problem, we will learn about matching, a tool that helps us make treatment and control units more comparable along a range of observable covariate dimensions. For example, treated units and control units might differ substantively in terms of their characteristics, but matching allows us to select comparable control units for each treated unit. If the assumptions that matching requires (which you can see summarized on the Lecture slides) hold, then matching allows us to get an estimate of causal effects. That is, matching is especially useful in settings where randomization is hard, like when we are interested in learning about the effects of phenomena that happen in real life (such as violence, poverty, access to political resources, etc.) but that we would not be willing and/or able to randomize.

Here we will work with the data in `boas_hidalgo_2011.RData`, from the Boas and Hidalgo (2011) article "Controlling the Airwaves: Incumbency Advantage and Community Radio in Brazil."[1] The authors are interested in the impact of politicians' ownership of radio stations on their likelihood of winning an election. Boas and Hidalgo find that politicians who acquired radio licenses for community radio stations before an election earned higher vote shares and were more likely to be elected than those

[1] Boas, Taylor C., and F. Daniel Hidalgo. "Controlling the airwaves: Incumbency advantage and community radio in Brazil." American Journal of Political Science 55, no. 4 (2011): 869-885.

who did not, arguing that this is one mechanism through which the "incumbency advantage" operates. They focus their analysis on city council elections in Brazil, and their data primarily come from publicly available administrative records. We will focus on the causal effect of community radio control, which appears in the data as `treat`, on vote share, `pctVV`. Other variables in the data are:

- `treat`: treatment indicator of getting a radio license in the time period before the election (1= yes, 0=no)
- `pctVV`: percent of valid votes won, the main outcome variable we care
- `log.valid.votes`: log of size of the electorate (number of valid votes)
- `pt_pres_1998`: percent of the presidential vote share won by the Workers' Party (PT) in 1998 in the municipality
- `occMedia`: dummy variable for whether the politician has a career in media
- `male`: whether the politician is male or female
- `log.total.assets`: log of the politician's total assets
- `log.num.apps`: log of municipal-level competition for radio licenses (the number of entities that responded to the call for applications)

1. Before we estimate the causal impact of getting a radio license, we want to know more about our outcome variable and obtain a baseline result. To do so, first summarize the outcome variable `pctVV` to produce a mean, median, maximum and minimum.

   Second, we want to get a naive difference-in-means estimator for comparison. To get the naive estimator, take the difference in means between the treatment group and the control group on `pctVV`, percent of valid votes won.

2. Briefly explain – in your own words – why matching might be a useful tool for causal inference in this context. Why can't we just compare the outcomes of politicians who received radio licenses before an election with those who did not?

3. Before we write our own matching function, compare the balance between treated (`treat=1`) and control units (`treat=0`).

   To do this, first calculate the difference in means for all the covariates in the dataset (i.e. all columns except the treatment indicator and the outcome of interest). You may find the `colMeans()` function useful. In order for comparisons across variables to not be driven by differences in the scale of the variables, divide difference in means by the standard deviation of the corresponding variable (which you can obtain with the functions `sd()`, or `colSds()` in the package `matrixStats`). Store the standardized differences in means for each of the covariate in a data frame.

   Then plot these differences for each covariate. Label the axes and give the plot an informative title. Briefly describe your findings and what they imply for causal inference, especially on why the naive difference-in-means estimator estimated in the first problem may be problematic.

   We will later compare balance (i.e. the standardized difference in means) *after* matching.

4. Now we are writing our own matching function. There are many different ways to implement matching. While the authors use a genetic matching algorithm, we will use Mahalanobis distance matching as introduced in lecture (see lecture slide **16-17**). The equation is as follows:

$$D_{ij} = \sqrt{(X_i - X_j)^\top \Sigma^{-1} (X_i - X_j)}$$

where Σ is the sample covariance matrix of X (see also lecture slides). First, write your own function using the `function()` command to implement this equation.

The input of your function should be the dataset as a whole, and two indices indicating two rows in your dataset. The output of your function should be the pair-wise Mahalanobis distance between the two rows that you specify previously in your dataset.

You can check if your function works by comparing your outputs against the outputs from the `mahalanobis()` function in R. Note that you need to take the square root of the `mahalanobis()` output to be able to compare your results. In `R`, use `t()` and `solve()` for transpose and taking the inverse, respectively.

5. Use this function to match control units to *each* treated unit based on the Mahalanobis distance between the two observations. Based on this new matched dataset, calculate the average treatment effect among treated units (i.e. those who successfully got radio licenses) on vote share. We do this in the following steps through two nested loops.

   First, select a set of variables to match on. Statistical theory indicates that researchers should match on pre-treatment variables that determine the probability that a unit receives the treatment to better defend the strong ignorability assumption required for causal estimation. In this context, getting a radio license is the treatment. For this problem, we ask you to match on the following six variables:

```
vars <- c("log.valid.votes", "pt_pres_1998", "male", "log.total.assets",
          "log.num.apps", "occMedia")
```

   For each treated unit `j` iterate through all control units `i` and calculate the Mahalanobis distance for each pair using a loop. This will produce a distance measure for each pair. Store the distance measure for each control unit and select the **two** closest control matches for that treated unit. Estimate the treatment effect of this treated unit by taking the difference in vote share between this treated unit and the average of the two matched control units. Append the treatment effect of this unit in the vector `att.comb` with the funciton `append()`.

   Repeat this step for all treated units in the dataset with another loop wrapped around this first loop. The mean of the vector `att.comb` is an estimator for the average treatment effect of the treated (ATT).

   The generic structure should be as follows. For k treatment units and n control units, find the two closest control matches based on the Mahalanobis distance. We will use all control units over and over again, allowing treated units to be matched to control units that have previously been matched to another treated unit. In other words, one control unit might be matched to several treatment units. This is called "matching with replacement."

   Your loop should have the following basic structure:

```
for(j in 1:k){      # loop through all k treated units
...
  for(i in 1:n){    # loop through all n control units
    ...
  }
}
```

Your final output will be a vector with a length equal to the number of treated units whereby each element stores the estimated individual treatment effect: the difference between the observed outcome of the treated unit and the mean of the observed outcome of the matched two control units. In the final step, take the mean of this object to get the Average Treatment effect on the Treated (ATT) of radio licenses on vote shore. Make sure to store which unique control units have been selected as control units, since you will need them in the next sub-question.

What does your finding imply? Compare your results with the mean of your outcome variable.

6. (Extra Credit) When we are matching, we should end up with treatment and control groups that are more similar to one another. Compare the standardized difference in means after matching to the standardized difference in means before matching. Merge the previously-stored difference in means pre-matching with the post-matching difference in means that you just calculated and plot both differences. Did matching improve balance overall? What could be done to further improve balance, including other matching methods? Based on these results, should you trust more on the native estimator or the matching estimator? (*Hint:* In production of this graph, use only unique controlling units. That is, if a controlling unit has been matched to multiple treatment units, only keep one of them in the calculation for the differences in means.).

Problem 2: Red State, Blue State, Rich State, and Poor State

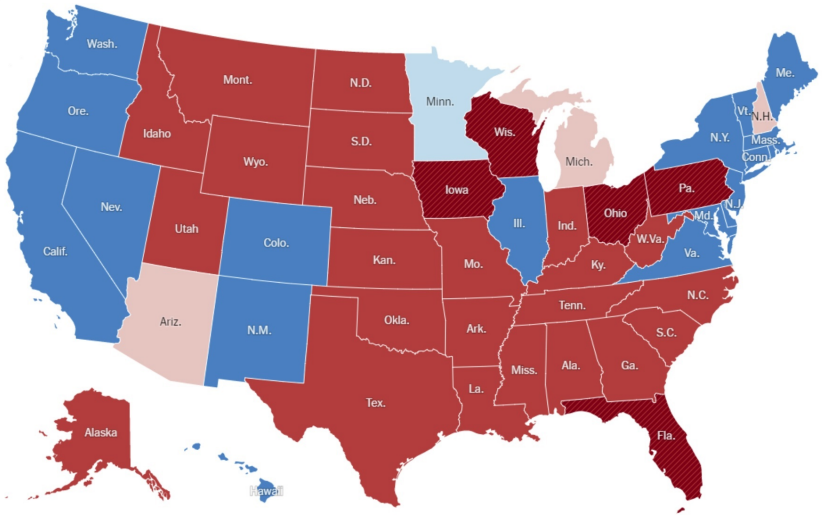

Figure 1: 2016 Electoral College Map. *Source: The New York Times*

Ecological fallacy is one of the key concepts related to the study of causal inference and the "Red-

State-Blue-State Paradox" represents a classical example of ecological fallacy in political science.[2] Despite journalistic accounts of "angry white working-class" being staunch Trump supporters, the Republican Party is regarded as the party of the rich. GOP endorsed various economic policies that favor the interests of the high income class: tax cut and deregulation of business, for examples. However, at aggregate-level, red states, i.e., states in which voters predominatedly support GOP candidates, are less economically developed than blue states that vote Democratic candidates. Why are poorer states like Idaho, North Dakota, and Mississippi more Republican than richer states such as Connecticut, New York and Massachusetts while rich voters tend to support Republican candidates? In this Problem, we are going to explore this puzzle by analyzing the Cooperative Congressional Election Study (CCES) survey data.[3] First, load the `cces_2012.csv` from the class website. The dataset contains the following variables:

- `state_abb`: state abbreviations
- `vote_gop`: Whether the respondent voted Republican candidate Mitt Romney in 2012 presidential election: 1 = Voted Romney, 0 = Otherwise.
- `income`: A five-point scale measuring respondent's family income: 1 = "Below $\$20,000$", 2 = "$\$20,000 - \$39,999$", 3 = "$\$40,000 - \$69,999$", 4 = "$\$70,000 - \$99,999$", 5 = "Above $\$100,000$." In the following questions, we call it "income level" or "income group."

1. Create a scatter plot to show the relationship between the state-level income and GOP vote share in each state.

   To do so, first use `tapply` function to calculate the proportion of respondents who voted Romney in each state. Name the new variable `gop_state_share`.[4]

   Second, use `tapply` function to compute the mean of `income` variable in each state. Name the new variable `income_state`.

   Third, plot the average income level of each state on the X-axis and GOP vote share in each state on Y-axis. Set `ylim` to `c(0, 0.8)`. Set Use `text` argument to plot the state abbreviations instead of points. Set the color of text to red.

   Finally, use `abline` argument to add a line that describes the bivariate relationship between state-level income and GOP vote share. (*Hint*: See Question 1.5 in Problem Set 1 for an example of how to implement this).

2. Then we create a bar plot to show the individual-level relationship between income and support for the Republican party.

[2] Since early 2000, the Republican party has been associated with red and the Democratic party has been associated with blue (See Figure 1).

[3] This problem is based on prior research done by a prominent political scientist and statistician, also an MIT alumnus majoring in Courses 8 and 18, Andrew Gelman. Gelman's original study involved state and county-level election returns and multilevel modeling techniques. In this problem, we will show that only using survey data and some basic statistical manipulation are sufficient to understand and paradox and reconcile the conflicting patterns. If you are interested in more advanced techniques or concerned about the problems of using surveys (which we mentioned in Pset one), you may consult Andrew Gelman, Boris Shor, Joseph Bafumi and David Park (2008), "Rich State, Poor State, Red State, Blue State: What's the Matter with Connecticut?", *Quarterly Journal of Political Science*: Vol. 2: No. 4, pp 345-367. `http://dx.doi.org/10.1561/100.00006026`

[4] *Hint*: You may calculate the total number of respondents who voted GOP in each state (name it `gop_state_vote`) and total number of respondents in each state (name it **`total_state_n`**): `gop_state_share` = `gop_state_vote`÷`total_state_n`.

To do so, first, use `tapply` function to compute the proportion of respondents who voted GOP candidate *in each income level*. Name the new variable `gop_income_share`.[5]

Second, create a bar chart to reflect the GOP vote share in each income level. Use `barplot` function to create the plot and make sure that X-axis represents each income group and Y-axis represents GOP vote share. Set `ylim` to `c(0, 0.6)`. On the same bar chart, use `points` function to plot five points that represent the Republican vote share in each of five income groups. Set `pch` to 16 and the size of the points, `cex` to 2. Set `ylim` to `c(0, 0.6)`. Use `lines` function to connect those points by levels of income.

Based on two graphs created above, what do you find about the relationship between income and support for GOP candidate? Do you notice the puzzle?

3. Why would this happen? To reconciliate those paradoxical patterns, we look at what happens within each state. For this question, we choose Massachusetts, a high-income-blue state, Wisconsin, a mid-income-purple state, and Mississipi, a low-income-red state as cases to further explore the puzzle. We will make a plot that compares the relationship between income and GOP voteshare across these three states.

   First, create an empty plot. Then use `abline(lm())` to plot three lines that represent the bivariate relationships between *individual* income and *individual* vote for GOP candidate in only Massachusetts, Wisconsin, and Mississipi, respectively.[6] To do this, you may subset the data or use conditional command.

   Second, set the colors of those lines to blue (MA), purple (WI), and red (MS), set the line types that can distinguish those three lines, and set the thickness (`lty`) of the lines to 4. Also add a legend that indicates three lines. Finally, on the same plot, add three points that represent the average income levels and GOP vote share in *each of three states*. Then add a line that connects those three points *by the average income level*. To connect those points in the right order, you may want to sort the data of three states by descending order.

4. Based on the figure you just created, what do you find? You may notice that the slopes of the threes lines are different. Which one is steeper and which one is shallower? What does the slope tell us about the relationship between individual income and support for GOP candidate?

   Also, use `cor` function to estimate the correlation coefficients between individual income and support for GOP candidate in Massachusetts and Mississipi. Which coefficient is larger? Do the cofficients tell us something about the paradox?

[5] You may follow similar procedure to create GOP vote share in each state to create the GOP vote share by income group.

[6] Note that there are more sophisticated ways to model and visualize the bivariate relationship when the dependent variable is binary (i.e., 0 and 1): using logit/probit model and plotting the predicted probabilities. For this Problem set, a linear model is sufficient to capture the basic ideas of the puzzle. Also, to obtain the desired outcomes, make sure that you created an empty scatter plot first and then use `abline(lm())` argument to add those lines.