Lab 11: Quasi-experiments

Due: Monday, April 1, 11:59 PM

Name: Your name here

Mac ID: The first half of your Mac email address

1 Regression discontinuity

The following code declares a regression discontinuity design (RDD) with an arbitrary bandwidth. The model needs to account for potential outcomes varying along with the score `X`.

```r
# cutoff determines treatment assignment
cutoff = 0.5
deg = 1
bw = 0.05

# functions to simulate potential outcomes based on
# score values
# See RD textbook chapter 16.5 for an explanation
control = function(X) {
  as.vector(poly(X, 4, raw = TRUE) %*% c(.7, -.8, .5, 1))}
```

```
treatment = function(X) {
  as.vector(poly(X, 4, raw = TRUE) %*% c(0, -1.5, .5, .8)) + .15}

# M
model = declare_model(
  N = 500,
  # unobserved variation
  U = rnorm(N, 0, 0.1),
  # score centered on cutoff point
  X = runif(N, 0, 1) + U - cutoff,
  # treatment indicator
  D = 1 * (X > 0),
  # potential outcomes depend on score
  Y_D_0 = control(X) + U,
  Y_D_1 = treatment(X) + U
)
```

The inquiry is the **Local Average Treatment Effect** (LATE) defined exactly at the cutoff. This quantity is not directly observable, but we will try to approximate it with our answer strategy.

```
# I
inq = declare_inquiry(
  LATE = treatment(0.5) - control(0.5)
)
```

Our data strategy simply reveals outcomes.

```
# D
reveal = declare_measurement(
  Y = reveal_outcomes(Y ~ D)
)
```

Our answer strategy reveals outcomes and estimates the LATE using a linear regression with an arbitrary polynomial function for X with degree `deg` and arbitrary bandwidth `bw`. We would normally use software to calibrate these, but here we will play with them manually to understand the bias-variance tradeoff.

```
# A
est = declare_estimator(
  Y ~ D * poly(X, degree = deg),
  subset = X > -1*bw & X < bw,
  .method = lm_robust,
  inquiry = "LATE",
  label = "parametric"
)
```

Task 1

- Diagnose the current design. How does it perform in terms of bias and RMSE?
- What happens if you use a higher order polynomial for `degree`? How does it affect bias and RMSE? Does that make the design better or worse? Explain (*Hint: Polynomials higher than 2 or 3 may make things too complicated.*)
- Stick to a value of `degree` that you like and see what happens when you increase the bandwidth `bw` to a number between the current value and 0.2. How does it affect performance in terms of bias and RMSE?
- Try now with the maximum possible bandwidth of `bw = 0.5` and keeping every-

thing else the same as in the previous task. How does it affect performance in terms of bias and RMSE?

- Based on the previous tasks. What degree of polynomial and bandwidth would you recommend for this design? Why do you think so? (You do not need to write code for this one)

2 Difference-in-differences

Difference-in-differences (DD) estimation gets much more complicated once we consider applications in which units can receive treatment at different time periods. The problem is that these are also the most common, interesting, or meaningful applications to public policy!

To dimension the extent of the problem, the following code simulates a DD design. Our model assumes two groups (treatment, control), an arbitrary number of time periods defined by `N_time_periods`, and units in the treatment group that take up treatment at randomly selected time periods. We start with two periods, which is the canonical design, except that some units can start in the treatment group.

```
N_time_periods = 2

# M
model = declare_model(
  units = add_level(
    N = 50,
    # unit level heterogeneity
    U_unit = rnorm(N),
    # unit in treatment group if above median of U
    D_unit = if_else(U_unit > median(U_unit), 1, 0),
    # select at random a time period for treatment
    D_time = sample(1:N_time_periods, N, replace = TRUE)
  ),
  periods = add_level(
    N = N_time_periods,
    U_time = rnorm(N),
    nest = FALSE
  ),
  unit_period = cross_levels(
    by = join_using(units, periods),
    U = rnorm(N),
    potential_outcomes(
      Y ~ U + U_unit + U_time +
                         D * (0.2 - 1 * (D_time - as.numeric(periods))),
      conditions = list(D = c(0,1))),
    D = if_else(D_unit == 1 & as.numeric(periods) >= D_time, 1, 0),
    D_lag = lag_by_group(D, groups = units, n = 1, order_by = periods)
  )
)
```

Notice that treatment assignment depends on the values of `U_unit`, which means our estimates will only be valid if we assume parallel trends. Furthermore, we also assume that treatment effect varies by when units receive treatment.

In this case, the effect is lower the later the unit receives treatment. This may be the case when policies have larger payoffs for early adopters. This comes from the `D * (0.2 - 1 * (D_time - as.numeric(periods)))` part of the `potential_outcomes` function.

We have two inquiries. The first inquiry is the **Average Treatment effect on the Treated (ATT)**. This is the difference between the observed post-treatment outcome in the treatment group and the unobserved counterfactual of the same group without receiving treatment.

For this particular version of the design, another inquiry is the ATT among those who *just switched* from control to treatment, so we are focusing on the potential outcomes of units who are treated now but were not treated one period ago. This means we are ignoring those who start the study in the treatment group, as well as the potential outcomes from periods outside the switching window.

```r
inq1 = declare_inquiry(
  ATT = mean(Y_D_1 - Y_D_0),
  subset = D == 1
)

inq2 = declare_inquiry(
    ATT_switchers = mean(Y_D_1 - Y_D_0),
    subset = D == 1 & D_lag == 0 & !is.na(D_lag)
  )
```

There is no randomization procedure, so the data strategy only realizes the observed outcomes based on the potential outcomes.

```r
reveal = declare_measurement(Y = reveal_outcomes(Y ~ D))
```

The estimator is a linear regression that controls for variability that is unrelated to the treatment at the unit and period level using "fixed effects." This is equivalent to calculating all the group means individually, but much less cumbersome. Because we are including fixed effects for two sources of variability, we call this a "two-way fixed effect" model. Many other estimators exist that aim to overcome the difficulties that emerge from multiple treatment periods, but for our purposes it is sufficient to focus on the canonical estimator.

```r
est = declare_estimator(
  Y ~ D,
  fixed_effects = ~ units + periods,
  .method = lm_robust,
  inquiry = c("ATT", "ATT_switchers"),
  label = "twoway-fe"
)
```

Task 2

- Diagnose the current design. How does it perform in terms of bias, RMSE, and power?
- Why is the performance for the `ATT` inquiry different from the performance for the `ATT_switchers` inquiry? (*Hint: This does not need code but a careful read of the material for this week, especially chapter 16.3 of the RD book.*)
- What happens to performance in terms of bias, RMSE, and power when you increase the number of time periods? Show at least two values larger than 2 but smaller or equal than 10. You do not need to give a specific answer for each value. Instead, explain the general trend and elaborate on why it happens.
- Is there an optimal number of time periods? What makes you say so? (You only need to show code and output for this part if you haven't already done so).

3 Answers

Remember to use `set.seed()` *and to show the code and output for every step!*

3.1 Task 1

Work on your answers here.

3.2 Task 2

Work on your answers here.