Lab 5: Survey Experimental Designs

Due: Monday, February 12, 11:59 PM

Name: Your name here

Mac ID: The first half of your Mac email address

1 Factorial Design

The following code simulates a 2x2 factorial experiment. You can see section 18.5 of the RD textbook for details. This is similar to the Tomz and Weeks (2013) reading but with one fewer condition or factor.

Our model must consider that we have two conditions, `Z1` and `Z2`. Each can be 1 or 0, and we must also consider what happens to the potential outcomes when they are both 1.

```
N = 1000 # sample size

# three different effects
Z1_effect = 0.2
Z2_effect = 0.1
interaction = 0.1

factorial_model = declare_model(
  N = N,
```

```
  U = rnorm(N),
  potential_outcomes(Y ~ Z1_effect * Z1 +
                       Z2_effect * Z2 +
                       interaction * Z1 * Z2 +
                       U,
                     conditions = list(Z1 = c(0,1),
                                       Z2 = c(0,1)))
)
```

We also have many inquiries now. We won't look at all the possibilities today. For each factor, we can consider three inquiries. For example, for `Z1` we can do:

1. The effect of switching `Z1` from 0 to 1 when `Z2` is 0
2. The effect of switching `Z1` from 0 to 1 when `Z2` is 1
3. The general effect over `Z1` across different values of `Z2`

The first two are called **Conditional Average Treatment Effects** (CATE) because their effect is conditional on holding other conditions fixed. The third effect is the ATE as we learned during class, and it's just the average of all the CATEs for each factor.

The reciprocal is true for `Z2`, so we have six inquiries or estimands in total.

```
factorial_inquiry = declare_inquiry(
  # Z1
  CATE_Z1_0 = mean(Y_Z1_1_Z2_0 - Y_Z1_0_Z2_0),
  CATE_Z1_1 = mean(Y_Z1_1_Z2_1 - Y_Z1_0_Z2_1),
  ATE_Z1 = mean(CATE_Z1_0, CATE_Z1_1),
  # Z2
  CATE_Z2_0 = mean(Y_Z1_0_Z2_1 - Y_Z1_0_Z2_0),
  CATE_Z2_1 = mean(Y_Z1_1_Z2_1 - Y_Z1_1_Z2_0),
```

```
  ATE_Z2 = mean(CATE_Z2_0, CATE_Z2_1)
)
```

Our data strategy needs to specify the assignment and measurement of outcomes.

```
factorial_assignment = declare_assignment(Z1 = complete_ra(N),
                                          Z2 = complete_ra(N))

factorial_measurement = declare_measurement(Y = reveal_outcomes(Y ~ Z1 + Z2))
```

And then we need as many estimators as estimands. Luckily, we can write the estimator for the two ATEs in one line of code. Notice how the names of the estimators are the same as the estimands but with CATE and ATE in lowercase. R is case sensitive so it will understand upper and lowercase versions as different objects.

```
#Z1
cate_Z1_0 = declare_estimator(Y ~ Z1, subset = (Z2 == 0),
                              inquiry = "CATE_Z1_0",
                              label = "CATE_Z1_0")

cate_Z1_1 = declare_estimator(Y ~ Z1, subset = (Z2 == 1),
                              inquiry = "CATE_Z1_1",
                              label = "CATE_Z1_1")

#Z2
cate_Z2_0 =  declare_estimator(Y ~ Z2, subset = (Z1 == 0),
                               inquiry = "CATE_Z2_0",
                               label = "CATE_Z2_0")
```

```
cate_Z2_1 = declare_estimator(Y ~ Z2, subset = (Z1 == 1),
                              inquiry = "CATE_Z2_1",
                              label = "CATE_Z2_1")
# ATEs
ate = declare_estimator(Y ~ Z1 + Z2,
                        term = c("Z1", "Z2"),
                        inquiry = c("ATE_Z1", "ATE_Z2"),
                        label = "ATE")
```

We can now put all the pieces together.

```
factorial_design = factorial_model + factorial_inquiry +
  factorial_assignment + factorial_measurement +
  cate_Z1_0 + cate_Z1_1 + cate_Z2_0 + cate_Z2_1 + ate
```

Task 1

Diagnose your `factorial_design` and show the bias and RMSE. On statistical grounds, which estimators have lower bias? Which estimators are more precise or have lower variance? Why do you think so?

Hint: *Remember to use* `set.seed()`

Task 2

What happens to bias and RMSE when you transform the design into a 2x2x2 factorial design? Explain why you think that happens.

Hints:

1. *This requires you to use the current design as a template to write a new one*
2. *The model now needs to specify multiple interaction effects. This may help to visualize what the model will need:*

```r
# code chunk set to hide results
# This is just for your reference
expand.grid(
  Z1 = c(0, 1),
  Z2 = c(0, 1),
  Z3 = c(0, 1)
)
```

The first row would be the baseline outcome encoded by **`U`**. *We then have three effects in which one of the* `Z_*` *is 1 and the rest is zero. The remaining four rows are interactions that you will need to encode. You will have to make up new effect sizes. They can be positive or negative, but I would advice not making them too different from the current ones.*

3. *Each factor now has 5 inquiries. For example, for* **`Z1`** *we need to specify* **`CATE_Z1_00`**, **`CATE_Z1_10`**, **`CATE_Z1_01`**, **`CATE_Z1_11`**, *and* **`ATE_Z1`**. *So fifteen inquiries in total.*
4. *This is how I would write one of the CATE estimators. Notice the use of the* **&** *operator in the* **`subset`** *argument.*

```r
# code chunk set to not evaluate
cate_Z1_00 = declare_estimator(Y ~ Z1,
                                subset = (Z2 == 0 & Z3 == 0),
                               inquiry = "CATE_Z1_00",
                               label = "CATE_Z1_00")
```

2 Answers

2.1 Task 1

Work on your answer here.

2.2 Task 2

Work on your answer here.