Lab 4: Reducing Sensitivity Bias

Due: Monday, February 5, 11:59 PM

Name: Your name here

Mac ID: The first half of your Mac email address

1 Warning!

There is a considerable difficulty spike in this lab! You should try your best to complete it. Give it your best, but don't worry if you can't complete everything! Remember that you are being evaluated by how much work you put into learning, not by completeness or correctness. The main tip I have for you is to work together! Labs will go more smoothly after this one.

2 Part 1: Bias-variance tradeoff in list experiments

This week, we learned about techniques to prevent respondents from concealing their answer to sensitive questions. We saw how these research designs reduce sensitivity bias, but this bias-reduction is not free.

A golden rule in statistics is **bias-variance** tradeoff. **Bias** is similar to validity, an *estimator* is unbiased if it produces estimates that average to the true, unobserved value of the quantity of interest (the *estimand*) over repeated hypothetical realizations of the same event generation process.

An estimator is **precise** (or has **low variance** or is *reliable*) when it produces *estimates* that are close to each other over *repeated hypothetical realizations* of the same *event generation process*.

This figure illustrates how estimators can be biased, precise, both, or neither. It is the same figure we saw for validity and reliability: https://popw23.gustavodiaz.org/slides/figs/darts_bias.png

The issue is that you can rarely improve upon one of the two dimensions without sacrificing the other, hence the tradeoff. In the context of research designs to reduce sensitivity bias, direct questions are biased, but because they are straightforward they are precise.

On the flip side, research designs to reduce sensitivity bias do so by introducing noise in the measurement, which makes the resulting estimates more imprecise.

We can illustrate this with the list experiment. The following code simulates a list experiment and compares it to a direct question. We start with our model.

```
N = 500

#  Model
list_model = declare_model(
  N = N,
  control_count = rbinom(N, size = 3, prob = 0.5),
  sensitive = rbinom(N, size = 1, prob = 0.3),
  lie = case_when(sensitive == 0 ~ 0L,
                  sensitive == 1 ~ rbinom(N, size = 1, prob = 0.2)),
  potential_outcomes(Y_list ~ sensitive * Z + control_count)
)
```

Our model considers a survey of `N = 500` respondents. In this case, we are agnostic about where the sample came from. `control_count` generates 3 baseline items, each of them applying to a respondent with probability 0.5. In turn, any given respondent holds the `sensitive` trait of interest with probability 0.3. Which will roughly approximate to a prevalence rate of about 30%. You may want to spend some time trying to understand what the `rbinom` function does.

The `lie` part looks weird, but it's just saying that if a person holds the sensitive trait (sensitive = 1), then they conceal the truth if asked directly with probability 0.2 If sensitive = 0, they simply tell the truth since there is nothing to conceal. `case_when` is a very interesting function that you may want to read the help documentation for.

The `potential_outcomes` step is also new. We are using it to declare how people would respond to being assigned to the treatment or control group. As you will see later, `Z` will take a value of 1 under treatment and 0 under control. If $Z = 1$, the true response to the "how many" question will be `sensitive + control_count`. If $Z = 0$, the response will be just `control_count`. We will talk more about the potential outcomes framework next week.

The next step is the **inquiry**, which in this case is the proportion of people who hold the sensitive trait.

```
# Inquiry
list_inquiry = declare_inquiry(prevalence_rate = mean(sensitive))
```

Then we declare our data strategy.

```
# Data
list_assignment = declare_assignment(Z = complete_ra(N))

list_measurement = declare_measurement(Y_list = reveal_outcomes(Y_list ~ Z),
                                       Y_direct = sensitive - lie)
```

Which in this case has two parts, first we indicate that condition `Z` in the list experiment is assigned by complete random assignment. Then we realize outcomes for responses to both the list experiment and direct questions.

Finally, we need answer strategies for each of our outcomes. We have two estimators for the same inquiry.

```
# Answer
list_estimator = declare_estimator(Y_list ~ Z,
                                   inquiry = "prevalence_rate",
                                   label = "list")

direct_estimator = declare_estimator(Y_direct ~ 1,
                                     inquiry = "prevalence_rate",
                                     label = "direct")
```

We can now put everything together.

```
list_design = list_model + list_inquiry + list_assignment +
  list_measurement + list_estimator + direct_estimator
```

If you are curious, you can use the `draw_data` function to see how one realization of our design looks like. But make sure you do not print this output to the console! I included a code chunk option to prevent this.

```
draw_data(list_design)
```

One more thing we have to do is define standards to evaluate the bias and precision of our estimators.

```
list_diagnosands = declare_diagnosands(
  bias = mean(estimand - estimate),
  RMSE = sqrt(mean((estimand - estimate)^2))
)
```

We are using one of many ways to measure bias. Our measure of precision is the Root Mean Squared Error, which is similar to how we measure bias but the squared part penalizes estimates that deviate too much from the estimand.

Let's use the `diagnose_design` function to compare the list experiment with direct responses.

```
set.seed(1234) # replace with your student ID

list_diag = diagnose_design(list_design, diagnosands = list_diagnosands)

# Inspect the object but hide output in the pdf
list_diag
```

```
  estimator   bias  RMSE
1    direct  0.060 0.061
2      list -0.004 0.084
```

So you can see that the direct question has more bias than the list experiment. This is because in our model some participants conceal the truth under the direct question. The list experiment has essentially zero bias because our model assumes that the list experiment works.

However, you can also see that the direct question has lower RMSE than the list experiment, which means that it is more precise. This suggests that using a list experiment will produce estimates that vary more than the direct question over multiple hypothetical realizations of the event generation process.

This implies that we may not be able to convince our audience that the list experiment performs any better than asking questions directly. We may get unlucky and get an estimate that is even further from the truth.

Task 1

Can the list experiment perform better in terms of RMSE if we increase the sample size? Repeat the research design above using different values of N until you find the smallest possible sample size under which the RMSE of the list experiment is smaller than that of the direct question.

Some tips:

- *There are multiple ways to accomplish this. You can try multiple values of N at once or do one at a time. In either case, using the* ***`redesign`*** *function may save you some time.*
- *If you end up doing one value of N at a time, you only need to report the value that you end up sticking with, but you need to explain with words the procedure you used to come up with that number.*

3 Part 2: Randomized and non-randomized response designs

3.1 Randomized response

The following code simulates a randomized-response research design and compares it to a direct question. Respondents will answer *yes* if the dice comes 1, 2, 3, or 4, and will answer truthfully if the dice shows 5 or 6.

```
## Model
rr_model = declare_model(
  N = 1000,
  sensitive = rbinom(N, size = 1, prob = 0.1),
  lie = case_when(sensitive == 0 ~ 0L,
                  sensitive == 1 ~ rbinom(N, size = 1, prob = 0.5)),
  direct = sensitive - lie
)

# In this case we need to declare potential outcomes as
# a separate step.
# Say yes if forced to, otherwise tell truth
rr_pot = declare_potential_outcomes(Y_Z_yes = 1, Y_Z_truth = sensitive)

## Inquiry
rr_inquiry = declare_inquiry(prevalence_rate = mean(sensitive))

## Data
# Forced to say yes with probability 4/6
rr_assignment = declare_assignment(Z = complete_ra(N, prob = 4/6,
    conditions = c("truth", "yes")))

rr_measurement = declare_reveal(Y, Z)
```

For the answer strategy, we will need to create custom estimator functions.

```
# RR estimator
rr = function(data){
  with(data,
    data.frame(estimate = (mean(Y) - 4/6)/(1 - 4/6)))
}
```

```
# Direct question estimator
direct = function(data){
  with(data,
    data.frame(estimate = mean(direct)))
}
```

And then we can encode these as custom estimators in `DeclareDesign`.

```
## Answer
rr_estimator = declare_estimator(handler = label_estimator(rr),
                                 inquiry = "prevalence_rate",
                                 label = "Randomized response")

rr_direct = declare_estimator(handler = label_estimator(direct),
                              inquiry = "prevalence_rate",
                              label = "Direct question")
```

Now we can put everything together

```
rr_design = rr_model + rr_pot +rr_inquiry + rr_assignment +
  rr_measurement + rr_estimator + rr_direct
```

And then diagnose:

```
rr_diag = diagnose_design(rr_design)

# Inspect but hide output in pdf
rr_diag
```

```
            estimator  bias  rmse
1     Direct question -0.05 0.050
```

```
2 Randomized response  0.00 0.013
```

3.2 Crosswise model

In one of the assigned readings, Oliveros and Gingerich (2020) use a crosswise model instead of a randomized response model. This variant is a **non-randomized response** design. In their case, their vignette looks like this:

Cross-wise model example

How many of the following statements are true

- My mother was born in OCTOBER, NOVEMBER, OR DECEMBER
- In order to avoid paying a traffic ticket, I would be willing to pay a bribe to a police officer

Please indicate your answer below

A. **both** statements are true OR **neither** statement is true

B. **one** of the two statements is true

This design has two advantages. First, respondents never answer to the sensitive question directly, which protects anonymity even more than the randomized response design. Second, the question does not depend on a randomization device.

This design gives us everything we need to estimate the proportion of people who hold the sensitive trait, which in this case is the willingness to bribe a police officer. Let's call that quantity of interest π.

We need an innocuous statement that is completely unrelated to the sensitive item but that the researcher can know its probability p ahead of time. The only requirement is that $p \neq 0.5$. In this case $p = 0.265$ was found in a phone survey conducted before the present study.

Respondents can only choose A or B, so we can express this as $\lambda = 1$ for A and $\lambda = 0$ for B.

The formula for the proportion of respondents who will select option A is

$$\lambda = \text{Prob(both true)} + \text{Prob(neither true)}$$

we can replace with the quantities described above to get

$$\lambda = p\pi + (1 - p)(1 - \pi)$$

and then solve for π to get the estimator $\hat{\pi}$

$$\hat{\pi} = \frac{\lambda + p - 1}{2p - 1}$$

Which we read as "pi hat." The hat reminds us this is an estimator, meaning an approximation of the estimand or the true prevalence rate.

Task 2

Write a research design that simulates the crosswise model as described here. How does it perform in comparison to the direct question and randomized response in terms of bias and RMSE? On statistical grounds, which of the three research designs (direct, RR, crosswise) is better for the current even generation process? Which one seems better in terms of ensuring anonymity? For the last two questions, explain why you think so.

Here are a few hints:

- *You will need to write a new estimator for the crosswise model*
- *The crosswise model has no random component, so there won't be a potential outcomes part*
- *You will need to encode individual responses for p and λ in the model part*
- *You do not need to include a comparison to the randomized response nor the direct question in your research design. Among other things, this means you do not need*

> *to include* `lie` *in the event generation process*
>
> - *The* `case_when()` *function is your friend*
>
> *This is the first time I ask you to write a research design from scratch. If you get stuck and are running out of time, I would still like to see your code even if it doesn't work along with an explanation of what you were hoping to accomplish. You can use the* `error = TRUE` *code chunk option to make sure your PDF knits even if there are errors in your code.*

4 Answers

Include your answers in the sub-sections below. You should show your code and its output when relevant. Avoid printing overly long objects, such as simulated data.

Feel free to intersperse writing and code. You should include enough prose to make sure I understand why you did what you did and how it led you to the final answer. When in doubt, it is better to err on the side of explaining more than necessary.

4.1 Task 1

4.2 Task 2