WP5

Before continuing on this assignment

Install the package `formatR`. Install the `scales` package.

The Actual Assignment

Use the following data for Questions 1 to 3.

```
data <- tibble(unit = 1:6, Y = c(1, NA_real_, 1, 0, 1, NA_real_),
    R = c(1, 0, 1, 1, 1, 0), x1 = c(0, 0, 0, 0, 1, 1), x2 = c(3,
        7, 9, 5, 4, 3))
```

For the data in this assignment, consider R to be an indicator of whether or not an outcome is missing or not.

Question 1

Suppose the missingness of outcomes is completely at random. This is referred to as `MCAR`. In this situation, to get an estimate, we impute the values of missing data as equivalent to the group's expectation as a whole.

Estimate $E[Y]$ under MCAR.

How often do you think MCAR holds? Why?

Question 2

Suppose missingness of outcome is missing at random, once we condition on covariates. In this situation, we have one (or multiple) covariates and take the conditional expectation of our outcome within each value (or strata) of the covariate.

Have you seen a procedure that does the same algorithm as Q2? Assuming the answer is no explain why.

Question 3

If the answer to Q2 is yes, run that algorithm find an estimate of $E[Y]$. What has to hold about this algorithm?

Question 4

For this problem, use the Q4 dataset contained in Q4.csv. The dataset is a pure simulation, but you can think of this as the effect of taking a class on causal inference on knowledge of political science.

Read in the data and save it as d4.

a) Run a usual estimate of Y on D. Do not worry about covariates. What answer do you get for the effect of D on Y?

b) Often, we can better estimate our outcome by using information about treatment using a procedure called inverse probability weighting. This is a multi-step procedure. Thanks to a famous theorem (Rosenbaum and Rubin 1983), the propensity score is a way to cut down on all the possible covariates

> predicting treatment to a single uni-dimensional variable. A propensity score is the probability of being assigned to a particular treatment, conditional on pre-treatment characteristics.

To estimate this, for old reasons we often use a logistic regression to predict the value of the treatment variable given relevant pre-treatment covariates. In R, a logistic regression is done like the following:

```r
example <- glm(D ~ x1 + x2, family = binomial(link = "logit"),
    data = data)
```

Run a propensity score model with the dataset. Save your results to a variable called pscore.

Next, once we have the propensity score, we generate inverse probability weights following the formula.

$$\frac{Treatment}{Propensity} - \frac{1 - Treatment}{1 - Propensity}$$

In R, an example to do this is as follows:

```r
data_w_weights <- broom::augment_columns(example_model, data,
    type.predict = "response") %>%
    # Not necessary but makes it more readable
rename(prop = .fitted) %>%
    mutate(inverse_probability_weight = (D/prop) + ((1 - D)/(1 -
        prop)))
```

Adapt the example to get the inverse probability weights for the dataset in the problem.

Finally, now that we have the weights, we can run a weighted least squares regression model. Happily, all this requires to compute is supplying R with the appropriate weights as a variable.

In `estimatr` we can do this with:

```r
lm_robust(Y ~ D, data = data, weights = column_of_weights) %>%
    tidy()
```

Adapt this example for this problem. Make and interpret a coefficient plot of your answer.

Coda

We are doing this by hand to describe the process of inverse probability weighting. In an actual project, we would be highly unlikely to do this by hand. Instead, we would take advantage of whatever package was commonly used in our language of choice. Doing the example by hand allows us to see the mechanics to understand what the "best" library is doing.

Inverse probability weighting is applied to account for different proportions of observations within strata in the population of interest. It can also be awfully unstable and prone to bias if the estimated propensity scores are small. This is because the propensity scores are showing up in the denominators and because the standard way to estimate them (logistic regression) can become unstable at the distribution's tails.