Overview:

The goal of this project is to outperform existing techniques in the literature related to nonmonotone missing data.

Completed:

- **Implemented simulation of monotone MAR data**: This is correspondingly easier than the subsequent nonmonotone MAR simulation. For this simulation we use the following approach:

  1. Generate X, Y_1, and Y_2 for elements $i = 1, \dots, n$.
  2. Using the covariate X, determine the probability p_1 of Y_1 being observed for each element i.
  3. Based on p_1, determine if $R_1 = 1$.
  4. If $R_1 = 0$, then $R_2 = 0$. Otherwise, using variables X and Y_1, determine the probability p_{12}.
  5. Based on p_{12} determine if $R_2 = 1$.

  At the end of the algorithm, we have determined the values of binary variables R_1 and R_2 for each i and if either of them are equal to 1, the corresponding level of Y_k. As is common in this literature, the values of R_1 and R_2 determine if the corresponding variable Y_1 or Y_2 is missing or observed with $R = 1$ indicating Y being observed.

- **Implemented simulation of nonmonotone MAR data**: Following the approach of [**robins1997non**], I construct a nonmonotone MAR simulation with two response variables Y_1 and Y_2 and one covariate X. The algorithm to generate the data is the following:

  1. Generate X, Y_1, and Y_2 for elements $i = 1, \dots, n$.
  2. Using the covariate X_i, generate probabilities for each element i p_0, p_1, and p_2 such that $p_0 + p_1 + p_2 = 1$.
  3. Select one option based on the three probabilities for each element i. If 0 is selected: $R_1 = 0$ and $R_2 = 0$. If 1 is selected $R_1 = 1$. If 2 is selected, $R_2 = 1$.
  4. We take the next step in multiple cases. If 0 was selected, we are done. If 1 was selected, we generate probabilities p_{12} based on X and Y_1. Then based on this probability, we determine if $R_2 = 1$. In the same manner, if 2 was selected in the previous step, we generate probabilities p_{21} based on X and Y_2. Then based on this probability, we determine if $R_2 = 1$.

  Like the monotone MAR simulation this algorithm produces similar final results with the determination of binary variables R_1 and R_2 and variables X, Y_1, and Y_2. Unlike the monotone MAR case, the nonmonotone MAR includes observations with Y_2 observed and Y_1 missing.

- **Simulation 1 with Monotone MAR**: Following the algorithm described in the monotone MAR simulation bullet, we first generate data from the following distributions:

  $$X_i \overset{iid}{\sim} N(0,1)$$
  $$Y_{1i} \overset{iid}{\sim} N(0,1)$$
  $$Y_{2i} \overset{iid}{\sim} N(\theta,1)$$

  Then, we create the probabilities $p_1 = \text{logistic}(x_i)$ and $p_{12} = \text{logistic}(y_{1i})$. Since, both x_i and y_1 are standard normal distributions, each of these probabilities is approximately 0.5 in expectation.

  The goal of this simulation is to estimate θ. Alternatively, we can express this as solving the estimating equation:

  $$g(\theta) \equiv Y_2 - \theta = 0.$$

  We estimate θ using the following procedures:

  - Oracle: This computes $\bar{Y}$ using *both* the observed and missing data.
  - IPW-Oracle: This is an IPW estimator using only the observed values of Y_2. The weights (inverse probabilities) use the actual probabilities.
  - IPW-Est: This is an IPW estimator using the probabilities that have been estimated by a logistic model.
  - Semi: This is the monotone semiparametric efficient estimator from Slide 11 (Equation 2) of Dr. Kim's Nonmonotone Missingness presentation.

  We run this simulation with different values of θ, sample size of 1000, and 1000 Monte Carlo replications. Each algorithm for each replication generates $\hat{\theta}$. In the subsequent tables, we compute the bias, standard deviation (sd), t-statistic (where we test for a significant difference between the Monte Carlo mean $\hat{\theta}$ and the true θ) and the p-value of the t-statistic.

Table 1: True Value is -5

algorithm	bias	sd	tstat	pval
oracle	0.001	0.033	0.680	0.248
ipworacle	-0.012	0.392	-0.973	0.165
ipwest	0.007	0.186	1.178	0.120
semi	0.001	0.074	0.538	0.295

Table 2: True Value is 0

algorithm	bias	sd	tstat	pval
oracle	-0.001	0.031	-1.091	0.138
ipworacle	-0.001	0.085	-0.201	0.420
ipwest	0.000	0.085	-0.029	0.488
semi	0.000	0.079	0.112	0.455

Table 3: True Value is 5

algorithm	bias	sd	tstat	pval
oracle	0.000	0.033	-0.468	0.320
ipworacle	0.010	0.383	0.857	0.196
ipwest	-0.006	0.176	-1.020	0.154
semi	0.000	0.077	-0.049	0.481

Overall, these results are mostly what I would have expected. All of the algorithms estimate the true value of θ correctly in each case, with the oracle estimate having the smallest variance followed by the semiparametric algorithm. If there is anything surprising it is that the IPW estimator has better performance with the estimated weights compared to the true weights. However, I think that this is a known phenomenon.

- **Simulation 1 with Nonmonotone MAR**: Like the monotone simulation, we generate data using:

$$\begin{aligned} X_i &\overset{iid}{\sim} N(0,1) \\ Y_{1i} &\overset{iid}{\sim} N(0,1) \\ Y_{2i} &\overset{iid}{\sim} N(\theta,1) \end{aligned}$$

However, because we have nonmonotone data, our "Stage 1" probabilities are different. We compute the true Stage 1 probabilities being proportional to the following values:

$$\begin{aligned} p_0 &\propto |x-2| \\ p_1 &\propto |x| \\ p_2 &\propto |x+2|. \end{aligned}$$

However, we keep the same structure for the Stage 2 probabilities with: $p_{12} = \text{logistic}(y_1)$ and $p_{21} = \text{logistic}(y_2)$. The goal remains to estimate θ. We continue to use the Oracle algorithm and the IPW-Oracle algorithm. Since we have nonmonotone MAR data, we use the "Proposed" algorithm that is described on Slide 25 (Equation 12) of Dr. Kim's presentation. The response models and outcome models were estimated using logistic regression and OLS and correctly specified. This yields the following results:

Table 4: True Value is -5

algorithm	bias	sd	tstat	pval
oracle	0.000	0.033	0.144	0.443
ipworacle	0.026	0.882	0.925	0.178
proposed	0.008	0.065	4.061	0.000

Table 5: True Value is 0

algorithm	bias	sd	tstat	pval
oracle	-0.001	0.032	-1.112	0.133
ipworacle	0.001	0.074	0.462	0.322
proposed	0.001	0.051	0.444	0.329

Table 6: True Value is 5

algorithm	bias	sd	tstat	pval
oracle	0.001	0.031	0.609	0.271
ipworacle	-0.001	0.199	-0.181	0.428
proposed	-0.001	0.050	-0.343	0.366

While most of these numbers are to be expected, the proposed algorithm exhibits considerable bias with the true value of $\theta = -5$. Why does this happen? When $\theta = -5$, $p_{21} = \text{logistic}(y_2)$ is very small ($\text{logistic}(-5) = 0.007$). This means that in many replications every single observation that is initially selected to have $R_2 = 1$ chooses to have $R_1 = 0$. I think that this is causing the problem, but I am not totally sure how to check.

- **Simulation 2 with Nonmonotone MAR**: We also want to simulate data that is correlated. For this simulation, we focus on $\text{Cov}(Y_1, Y_2)$. While the probabilities are generated the same as the previous simulation the data generating process is

$$\begin{bmatrix} X_i \\ Y_{1i} \\ Y_{2i} \end{bmatrix} \overset{iid}{\sim} N\left(\begin{bmatrix} 0 \\ 0 \\ \theta \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \sigma_{yy} \\ 0 & \sigma_{yy} & 1 \end{bmatrix} \right).$$

We are still interested in $\bar{Y}_2$ and we still run 1000 simulation with 1000 observations. In all of the next simulations the true value of $\theta = 0$. The results are the following:

Table 7: True Value is 0. Cor(Y1, Y2) = 0.1

algorithm	bias	sd	tstat	pval
oracle	-0.002	0.032	-2.056	0.020
ipworacle	0.002	0.077	0.913	0.181
proposed	0.002	0.048	0.992	0.161

Table 8: True Value is 0. Cor(Y1, Y2) = 0.5

algorithm	bias	sd	tstat	pval
oracle	-0.001	0.032	-0.766	0.222
ipworacle	0.000	0.079	-0.018	0.493
proposed	-0.006	0.050	-3.534	0.000

Table 9: True Value is 0. Cor(Y1, Y2) = 0.9

algorithm	bias	sd	tstat	pval
oracle	0.000	0.033	0.221	0.413
ipworacle	-0.002	0.085	-0.593	0.277
proposed	-0.035	0.053	-20.992	0.000

While a small correlation (0.1) in Table 7 (also see Table 5 for no correlation), shows that the proposed algorithm with an insignificant level of bias, with a stronger correlation the bias increases.

- **Simulation 3 with Nonmonotone MAR**: This simulation aims to see if the proposed algorithm is doubly robust. First, we check with a misspecified outcome model. In this case the data generating procedure is the following:

$$\begin{bmatrix} X_i \\ \varepsilon_{1i} \\ \varepsilon_{2i} \end{bmatrix} \overset{iid}{\sim} N \left(\begin{bmatrix} 0 \\ 0 \\ \theta \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \sigma_{yy} \\ 0 & \sigma_{yy} & 1 \end{bmatrix} \right).$$

Then,

$$y_{1i} = x_i + x_i^2 \varepsilon_{1i} \text{ and } y_{2i} = -x_i + x_i^3 + \varepsilon_{2i}.$$

This procedure causes X to influence both Y_1 and Y_2 and we still have correlation in the error terms of Y_1 and Y_2. However, since neither Y_1 nor Y_2 are linear in X, the model will be misspecified. The response mechanisms are first generated MCAR with a probability of either Y_1 or Y_2 being the first variable observed to be 0.4. (There is a 0.2 probability neither is observed.) Then the probability of the other variable being observed is proportional to logistic(y_k) where y_k is the y that has been observed. To ensure that the proposed method has the correct propensity score we use the oracle probabilites instead of estimating them. This yields the following:

Table 10: True Value is 0. Cor(Y1, Y2) = 0

algorithm	bias	sd	tstat	pval
oracle	-0.001	0.107	-0.257	0.399
ipworacle	-0.002	0.150	-0.554	0.290
proposed	-0.001	0.121	-0.339	0.367

Commments from JK

Table 11: True Value is 0. Cor(Y1, Y2) = 0.1

algorithm	bias	sd	tstat	pval
oracle	0.001	0.106	0.450	0.326
ipworacle	0.000	0.149	0.127	0.449
proposed	0.002	0.120	0.780	0.218

Table 12: True Value is 0. Cor(Y1, Y2) = 0.5

algorithm	bias	sd	tstat	pval
oracle	0.001	0.105	0.390	0.348
ipworacle	0.002	0.150	0.666	0.253
proposed	0.000	0.119	0.021	0.491

- Three variables (X, Y_1, Y_2) should be correlated.
- I think the proposed method is doubly robust. So, you may make a simulation setup testing DR property. I will explain it further.