Evaluating the Impact of Missing Data Mechanisms on Multiple Imputation: A Simulation Study with E-Commerce Data

Brian Cervantes Alvarez

2024-12-05

Abstract

Missing data can bias statistical analyses and compromise the validity of conclusions. This report explores the performance of Multiple Imputation by Chained Equations (MICE) under three different missing data mechanisms: Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR). We introduce controlled missingness at varying levels (10%, 30%, 50%, and 70%) into a key outcome variable, “Refunds,” in an e-commerce dataset. The dataset includes predictor variables such as “Purchased Item Count,” “Refunded Item Count,” “Total Revenue,” and “Sales Tax.” After applying MICE to impute missing values, we assess how these mechanisms and levels of missingness affect regression coefficients, bias, and standard errors. The results show that while MICE works well under MAR and remains tolerable under MCAR, it struggles to correct for MNAR missingness. A final bar plot with error bars offers an aggregate view of coefficient estimates under different conditions. The findings highlight the critical role of understanding the missingness mechanism and the potential need for more advanced methods if MNAR conditions are suspected.

Introduction

In many real-world analyses, missing data are inevitable. The mechanism behind why data are missing significantly influences the validity of any statistical inference. The three main missingness mechanisms are:

- **MCAR (Missing Completely at Random):** Probability of missingness is unrelated to observed or unobserved data. Results under MCAR remain unbiased but can lose efficiency.
- **MAR (Missing at Random):** Probability of missingness depends only on observed data. By conditioning on the right observed variables, unbiased estimation is possible.
- **MNAR (Missing Not at Random):** Probability of missingness depends on unobserved values. Standard imputation methods may fail to correct the resulting biases without additional modeling assumptions.

Multiple Imputation by Chained Equations (MICE) is a popular method for handling missing data, particularly effective under MAR. MICE iteratively builds regression models for each incomplete variable using other variables, generating multiple complete datasets. Results are then combined using Rubin's rules, producing valid statistical inferences that incorporate imputation uncertainty.

In this study, we introduce MCAR, MAR, and MNAR missingness at various levels (10%, 30%, 50%, and 70%) into the "Refunds" variable of an e-commerce dataset. The dataset includes "Purchased Item Count," "Refunded Item Count," "Total Revenue," and "Sales Tax" as predictors. After applying MICE, we evaluate coefficient estimates, bias relative to the original full-data model, and the associated standard errors. Finally, we present a comparative bar plot to summarize coefficient shifts under these mechanisms.

Methodology

Data and Variables

The dataset includes the following variables:

- **Refunds:** Outcome variable.
- **Purchased Item Count:** Number of purchased items per order.
- **Refunded Item Count:** Number of refunded items per order.
- **Total Revenue:** Total revenue per order.
- **Sales Tax:** Sales tax applied to the order.

We focus on missingness introduced into the *Refunds* variable.

Missingness Mechanisms

1. **MCAR:** Missingness introduced randomly, independent of any other variables.
2. **MAR:** Missingness in "Refunds" is related to "Total Revenue," with lower revenue orders more likely to have missing "Refunds."
3. **MNAR:** Missingness in "Refunds" is related to the actual "Refunds" value, such that larger refunds are more likely to be missing.

Computation Theory of M.I.C.E.

MICE generates imputations by iteratively fitting models for each incomplete variable. For variables $X_1, \dots, X_p$:

1. Start with initial guesses for missing values.
2. For each variable X_j with missing data, regress X_j on the other variables (using current imputations for them):

$$X_j^{(t)} \sim f_j(X_1^{(t)}, \dots, X_{j-1}^{(t)}, X_{j+1}^{(t-1)}, \dots, X_p^{(t-1)})$$

3. Update the imputations based on these fitted models.
4. Cycle through all variables until convergence.

After obtaining multiple imputed datasets, we analyze each and combine results using Rubin's rules. If Q_k is the estimate from the k-th imputed dataset, and U_k its variance:

$$\bar{Q} = \frac{1}{m}\sum_{k=1}^{m} Q_k, \quad U = \frac{1}{m}\sum_{k=1}^{m} U_k, \quad B = \frac{1}{m-1}\sum_{k=1}^{m} (Q_k - \bar{Q})^2.$$

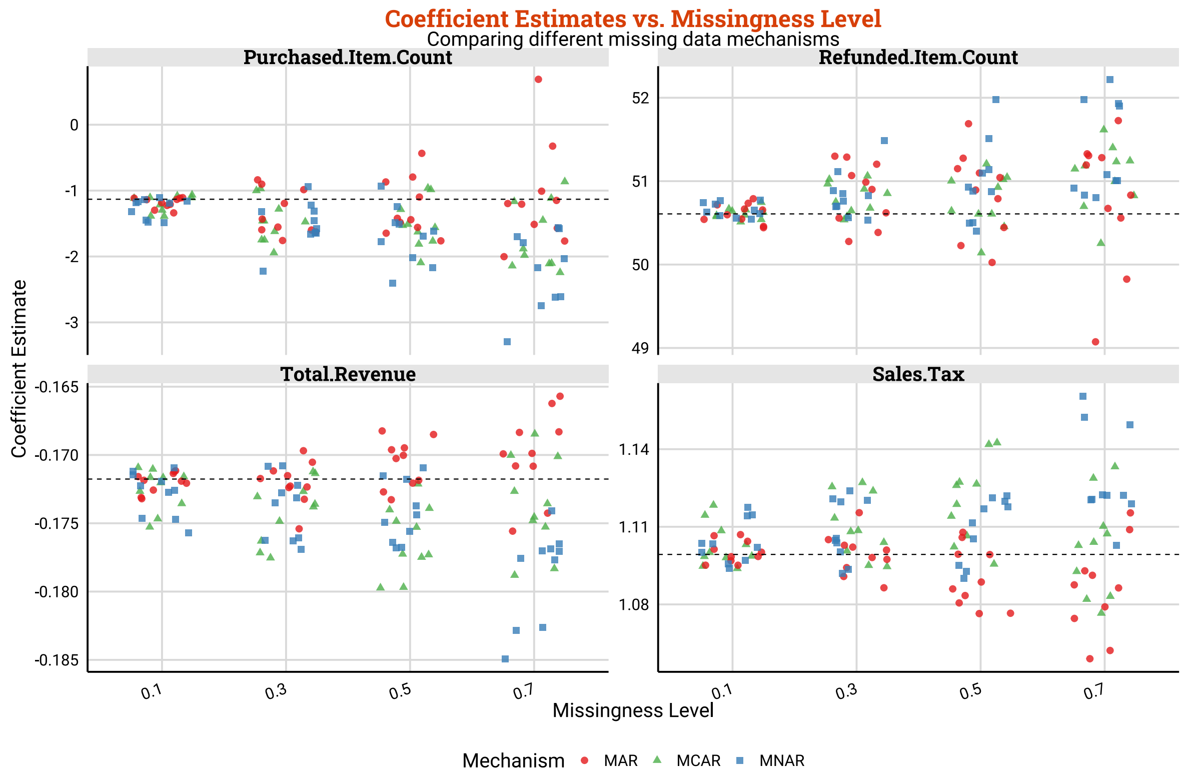
Total variance:

$$T = U + \left(1 + \frac{1}{m}\right) B.$$

This ensures both within-imputation and between-imputation variability are accounted for.

Results

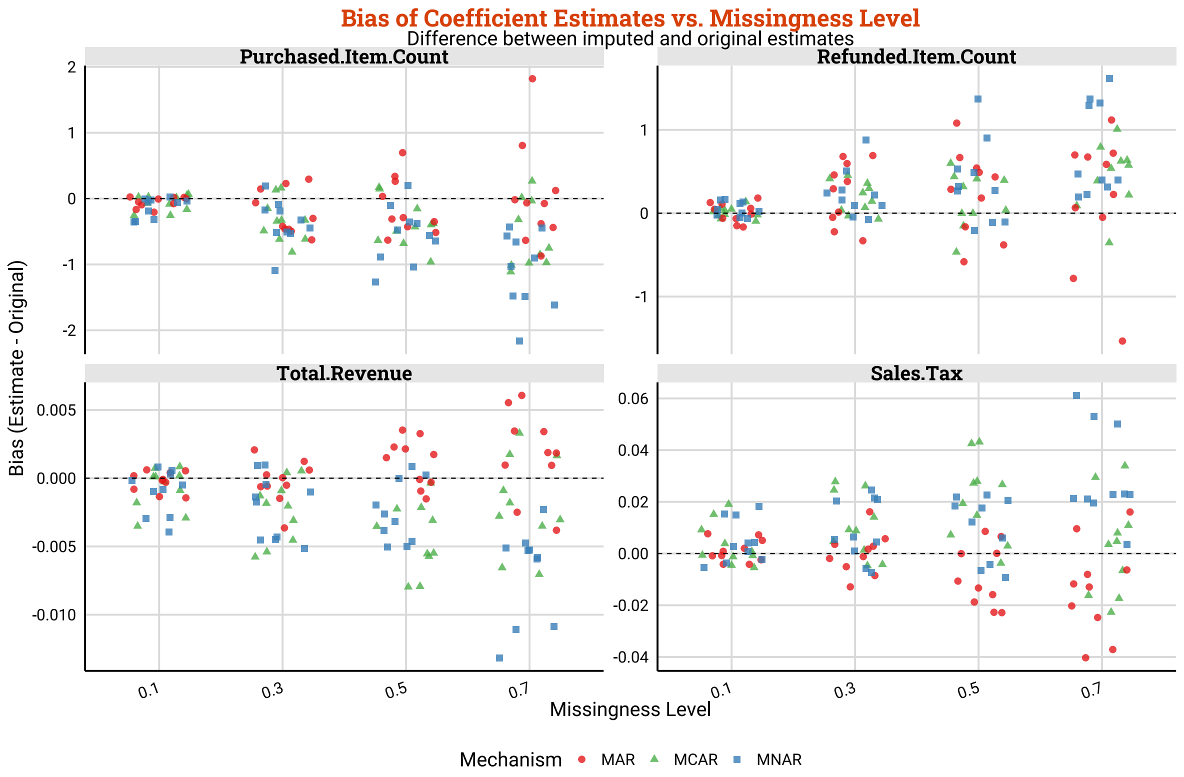
Coefficient Estimates vs. Missingness Level



Interpretation:

- Under MAR, estimates remain close to the original at lower missingness but may diverge as missingness reaches 50%-70%.
- MCAR conditions produce estimates that generally remain centered around the original, though with more variability as missingness grows.
- MNAR conditions show early and substantial deviations from the original, indicating that MICE struggles to correct bias when missingness depends on the unobserved "Refunds" values.

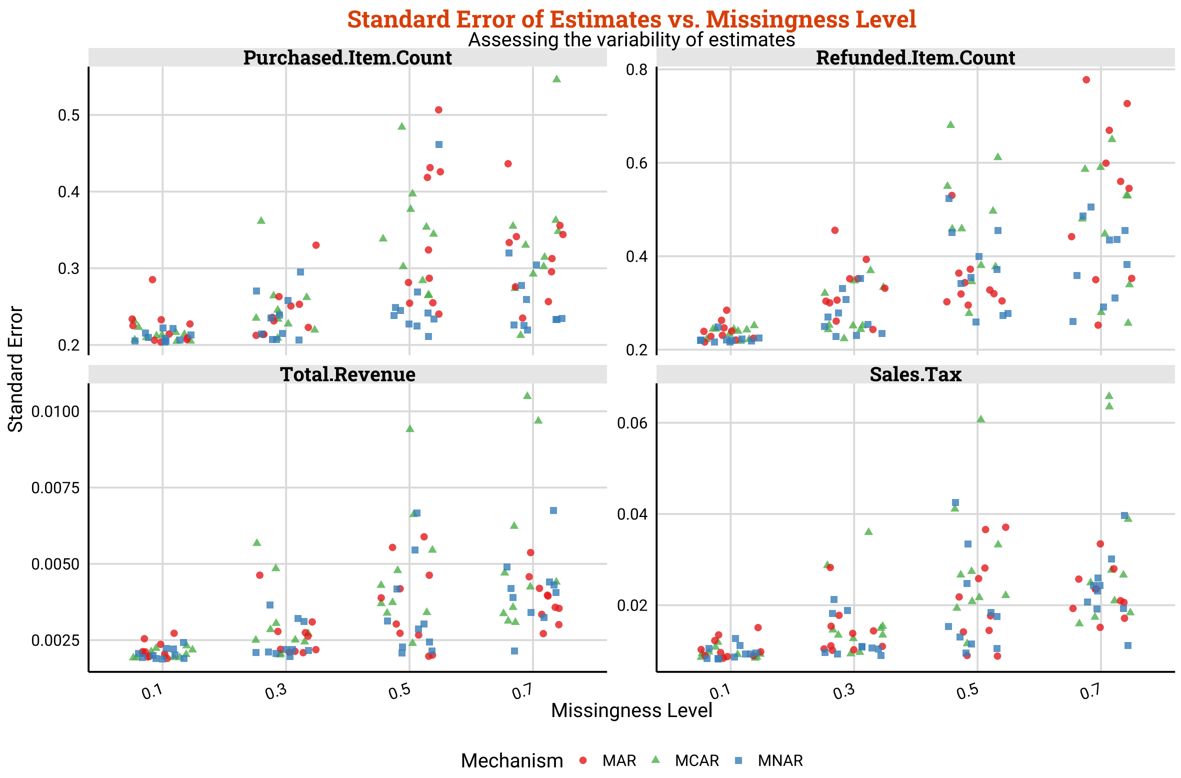
Bias of Coefficient Estimates vs. Missingness Level



Interpretation:

- MAR: Bias is minimal at low missingness but may increase at higher levels.
- MCAR: Bias remains relatively small, showing that randomness in missingness does not systematically skew estimates, though it does inflate variance.
- MNAR: Noticeable bias emerges even at low missingness and tends to grow, reflecting that standard MICE cannot fully address MNAR conditions.

Standard Error of Estimates vs. Missingness Level

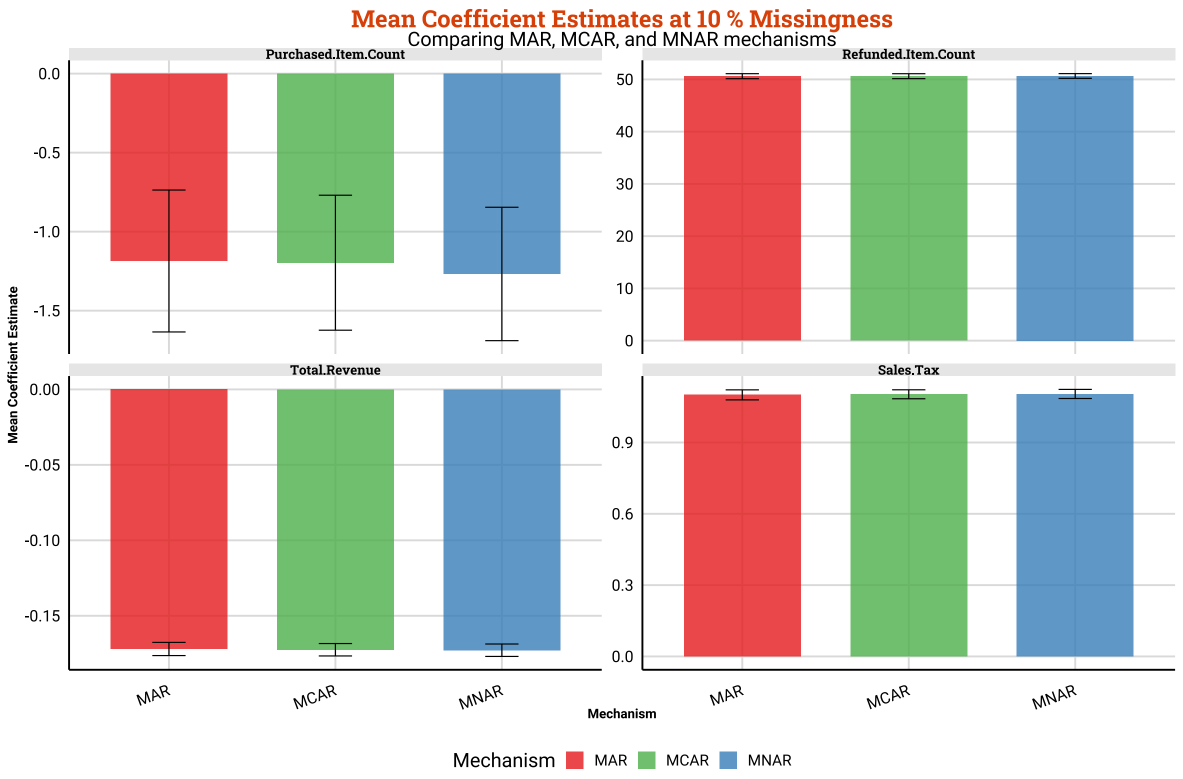


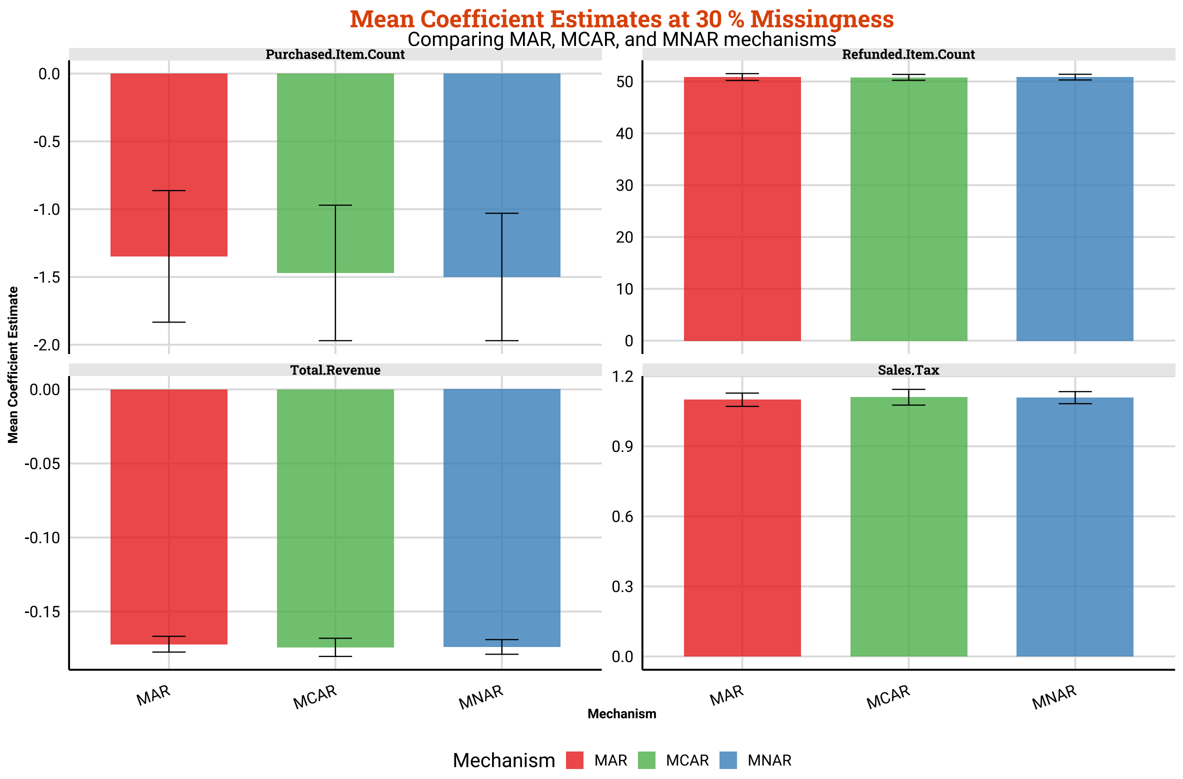
Interpretation:

- MAR: Standard errors grow as missingness increases, signifying rising uncertainty.
- MCAR: Standard errors increase predictably with missingness, due to reduced sample sizes.
- MNAR: Standard errors can be erratic and large, showing increased uncertainty and volatility in the estimates.

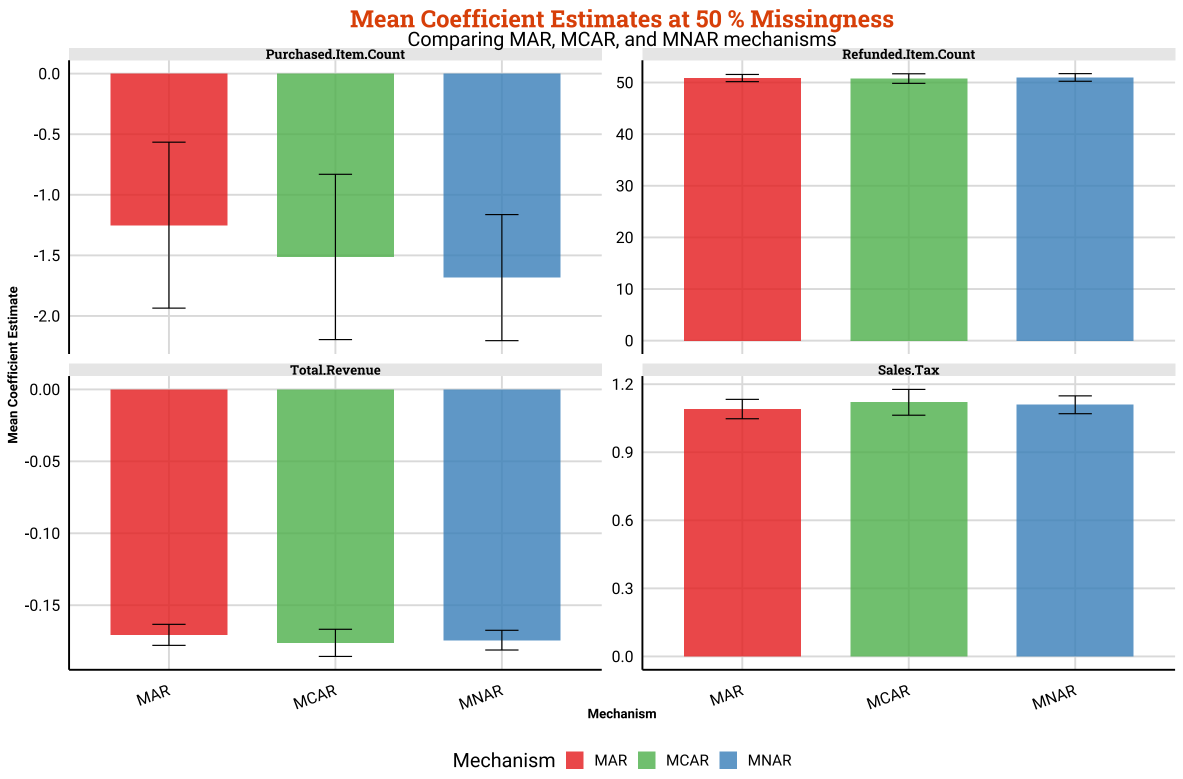
Final Bar Plot of Coefficients with Error Bars

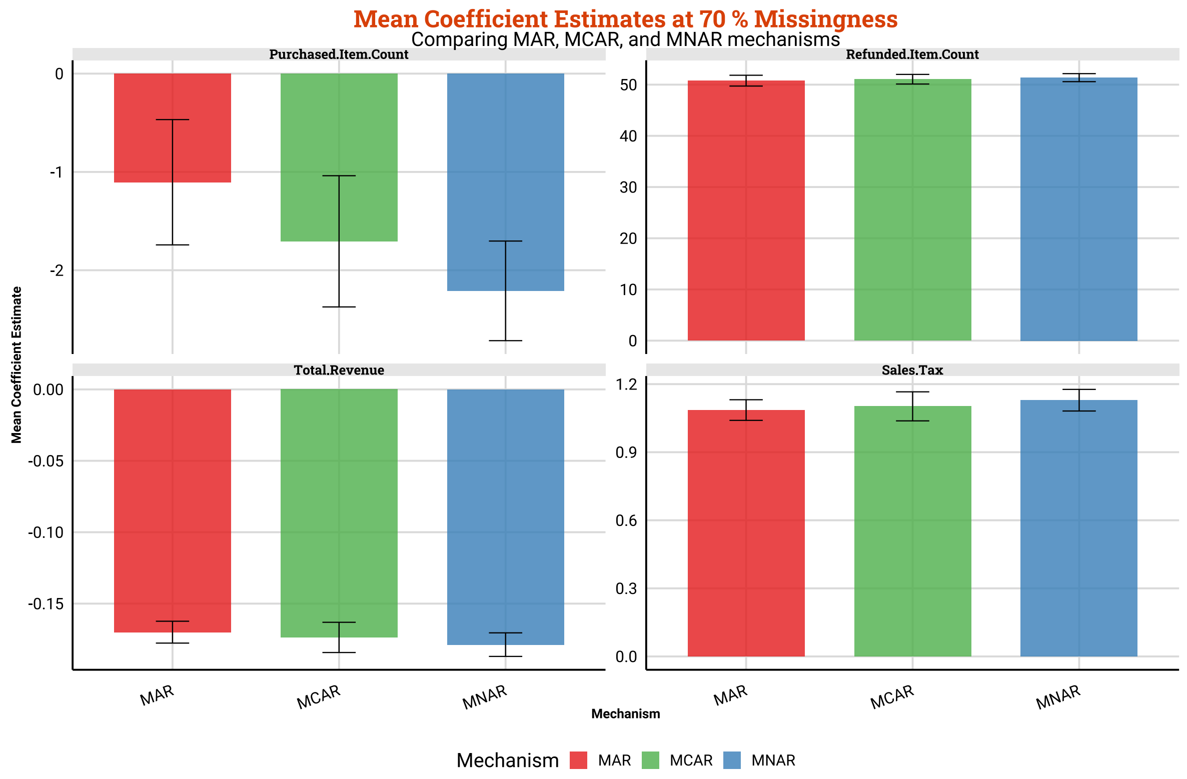
To provide an aggregated view, we compare MAR, MCAR, and MNAR mean estimates for each variable, including ± 2 standard error bars.


Mean Coefficient Estimates at 10 % Missingness
Comparing MAR, MCAR, and MNAR mechanisms
Purchased.Item.Count
Refunded.Item.Count
Total.Revenue
Sales.Tax
Mean Coefficient Estimate
Mechanism
MAR
MCAR
MNAR
0.0
-0.5
-1.0
-1.5
50
40
30
20
10
0
0.00
-0.05
-0.10
-0.15
0.9
0.6
0.3
0.0
Mechanism
MAR
MCAR
MNAR



Mean Coefficient Estimates at 30 % Missingness
Comparing MAR, MCAR, and MNAR mechanisms
Purchased.Item.Count
Refunded.Item.Count
Total.Revenue
Sales.Tax
Mean Coefficient Estimate
0.0
-0.5
-1.0
-1.5
-2.0
50
40
30
20
10
0
0.00
-0.05
-0.10
-0.15
1.2
0.9
0.6
0.3
0.0
MAR
MCAR
MNAR
Mechanism
Mechanism
MAR
MCAR
MNAR


Mean Coefficient Estimates at 50 % Missingness
Comparing MAR, MCAR, and MNAR mechanisms
Purchased.Item.Count
Refunded.Item.Count
Total.Revenue
Sales.Tax
Mean Coefficient Estimate
Mechanism
MAR
MCAR
MNAR
0.0
-0.5
-1.0
-1.5
-2.0
50
40
30
20
10
0
0.00
-0.05
-0.10
-0.15
1.2
0.9
0.6
0.3
0.0
Mechanism 🟥 MAR 🟩 MCAR 🟦 MNAR



Interpretation of the Bar Plots:

10% Missingness: At this low level of missingness, MAR and MCAR produce mean estimates that are close to one another and relatively stable across all variables. MNAR shows slightly wider error bars for variables like `Purchased.Item.Count` and `Total.Revenue`, indicating increased uncertainty, but the estimates are still reasonably close to those from MAR and MCAR.

30% Missingness: The variability in estimates becomes more apparent as missingness increases. MAR conditions remain relatively consistent across variables, with smaller error bars, particularly for `Refunded.Item.Count` and `Sales.Tax`. MCAR introduces slightly larger error bars for some variables, suggesting more variability in the estimates. MNAR exhibits the most notable deviations, with `Purchased.Item.Count` and `Total.Revenue` showing both higher uncertainty and potential bias.

50% Missingness: At this moderate level of missingness, the divergence between mechanisms becomes more pronounced. MAR still provides reasonably stable estimates, but MCAR begins to exhibit noticeable increases in error bars for `Purchased.Item.Count` and `Total.Revenue`. MNAR estimates, while consistent for `Refunded.Item.Count` and `Sales.Tax`, display substantial deviations and larger error bars for `Purchased.Item.Count` and `Total.Revenue`, emphasizing the challenges in imputing data missing under MNAR assumptions.

70% Missingness: At this extreme level of missingness, all mechanisms show increased uncertainty, with MNAR showing the greatest bias and widest error bars. MAR still produces the most stable estimates, but even here, error bars for variables like `Purchased.Item.Count` and `Total.Revenue` widen significantly. MCAR also reflects more variability, though it generally performs better than MNAR. The Sales.Tax variable appears less impacted by missingness overall compared to the other variables.

General Observations:

- MAR consistently outperforms the other mechanisms in terms of producing stable and accurate estimates across all levels of missingness.
- MCAR introduces increasing variability as missingness grows, but the estimates remain reasonable for moderate levels (10–30%).
- MNAR is the most problematic mechanism, with higher bias and uncertainty, especially for variables like `Purchased.Item.Count` and `Total.Revenue.` These deviations underscore the limitations of standard imputation methods like MICE under MNAR assumptions.

Conclusion

By introducing controlled missingness into “Refunds” under MAR, MCAR, and MNAR conditions, we find that MICE handles MAR data effectively at low to moderate missingness levels and remains unbiased under MCAR but less efficient. Under MNAR, however, the method struggles, yielding biased and uncertain estimates even at low levels of missingness.

Key Takeaways:

- **MAR:** MICE provides near-unbiased estimates, especially when missingness is not extreme.

- **MCAR:** Although unbiased, precision declines with increased missingness, as expected from random data loss.

- **MNAR:** MICE does not correct the inherent bias arising when the probability of missingness depends on unobserved values.

In practice, identifying the likely missingness mechanism is crucial. Analysts should consider advanced imputation techniques or sensitivity analyses when MNAR conditions are suspected.

Further Study

Future research could explore:

- Comparing MICE with alternative methods (Bayesian imputation, machine learning approaches) under each mechanism.
- Examining other outcome types (binary, count) or more complex models.
- Performing sensitivity analyses, pattern-mixture, or selection models tailored to MNAR data.

References

- van Buuren, S., & Groothuis-Oudshoorn, K. (2011). *mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software*, 45(3), 1-67.
  URL: https://CRAN.R-project.org/package=mice
- Rubin, D.B. (1987). *Multiple Imputation for Nonresponse in Surveys.* John Wiley & Sons.
- Little, R.J.A., & Rubin, D.B. (2019). *Statistical Analysis with Missing Data* (3rd ed.). John Wiley & Sons.
- Wickham, H., Hester, J., & Bryan, J. (2018). *readr: Read Rectangular Text Data.* R package version 2.1.4.
  URL: https://CRAN.R-project.org/package=readr
- Wickham, H., François, R., Henry, L., & Müller, K. (2023). *dplyr: A Grammar of Data Manipulation.* R package version 1.1.4.
  URL: https://CRAN.R-project.org/package=dplyr
- Robinson, D., Hayes, A., & Couch, S. (2023). *broom: Convert Statistical Objects into Tidy Tibbles.* R package version 1.0.5.
  URL: https://CRAN.R-project.org/package=broom
- Wickham, H., & Girlich, M. (2023). *tidyr: Tidy Messy Data.* R package version 1.3.0.
  URL: https://CRAN.R-project.org/package=tidyr
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York.
  URL: https://ggplot2.tidyverse.org/
- Qiu, M. (2022). *showtext: Using Fonts More Easily in R Graphs.* R package version 0.9-6.
  URL: https://CRAN.R-project.org/package=showtext

Appendix

Simulation Study Source Code

```r
# Load necessary libraries for data manipulation, imputation, and
 ↪  visualization
library(readr)    # For reading CSV files
library(dplyr)    # For data wrangling
library(mice)     # For multiple imputation
library(broom)    # For summarizing model results
library(tidyr)    # For reshaping data
library(ggplot2)  # For creating plots
library(showtext) # For custom fonts in plots

# Load fonts and set up the theme
font_add_google("Roboto", "Roboto")         # Add Roboto font for general
 ↪  text
font_add_google("Roboto Slab", "Roboto Slab") # Add Roboto Slab font for
 ↪  titles
showtext_auto()                            # Enable font rendering for plots

# Define colors for the theme
linkColor <- "#D73F09"      # Custom color for titles and links
bodyBgColor <- "#ffffff"    # Background color for plots
bodyTextColor <- "#000000"  # Text color for axis labels and legend

# Define a custom theme for consistent styling across plots
themeOSUStyle <- function(base_size = 18, base_family = "Roboto") {
  theme_minimal(base_size = base_size, base_family = base_family) %+replace%
    theme(
      plot.title = element_text(family = "Roboto Slab", face = "bold", size =
       ↪  rel(1.2), hjust = 0.5, color = linkColor),
      plot.subtitle = element_text(family = "Roboto", size = rel(1), hjust =
       ↪  0.5, color = bodyTextColor),
      axis.title = element_text(family = "Roboto", size = rel(1), color =
       ↪  bodyTextColor),
      axis.text = element_text(family = "Roboto", size = rel(0.8), color =
       ↪  bodyTextColor),
      legend.title = element_text(family = "Roboto", size = rel(1), color =
       ↪  bodyTextColor),
      legend.text = element_text(family = "Roboto", size = rel(0.8), color =
       ↪  bodyTextColor),
      plot.background = element_rect(fill = bodyBgColor, color = NA),
      panel.background = element_rect(fill = bodyBgColor, color = NA),
      panel.grid.major = element_line(color = "grey85"),
      panel.grid.minor = element_blank(),
      axis.line = element_line(color = bodyTextColor),
      strip.background = element_rect(fill = "grey90", color = NA),
      strip.text = element_text(family = "Roboto Slab", face = "bold", size =
       ↪  rel(1), color = bodyTextColor),
      axis.text.x = element_text(angle = 20, hjust = 1),
      legend.position = "bottom",
      legend.background = element_rect(fill = "transparent", color = NA),
      legend.box.background = element_rect(fill = "transparent", color = NA)
    )
}

# Data Preparation: Load the dataset and select relevant columns
returnsData <- read_csv("order_dataset.csv") # Import the dataset
selCols <- c("Refunds", "Purchased Item Count", "Refunded Item Count", "Total
 ↪  Revenue", "Sales Tax")
analysisData <- returnsData[, selCols] # Subset the dataset to the selected
 ↪  columns
colnames(analysisData) <- make.names(colnames(analysisData)) # Ensure column
 ↪  names are valid R variable names

# Define functions to introduce missingness into the data

# Missing Completely at Random (MCAR): Randomly remove values
introduceMcar <- function(data, column, perc) {
  n <- nrow(data)                               # Number of rows in the
 ↪  dataset
  numMissing <- floor(perc * n)                # Calculate the number of
 ↪  missing values
  mcarIndices <- sample(1:n, size = numMissing, replace = FALSE) # Randomly
 ↪  select indices
  data[mcarIndices, column] <- NA              # Set the selected values to NA
  return(data)                                 # Return the modified dataset
}

# Missing at Random (MAR): Remove values based on another variable
introduceMar <- function(data, targetCol, driverCol, perc) {
  n <- nrow(data)                               # Number of rows in the
 ↪  dataset
```

```
  numMissing <- floor(perc * n)                # Calculate the number of
 ↪  missing values

  # Scale the driver variable to a 0-1 range
  revScaled <- (data[[driverCol]] - min(data[[driverCol]])) /
               (max(data[[driverCol]]) - min(data[[driverCol]]) + 1e-6)
  probs <- (1 - revScaled + 1e-6)              # Higher values have lower
 ↪  probability of missingness
  probs <- probs / sum(probs)                  # Normalize the probabilities

  marIndices <- sample(1:n, size = numMissing, replace = FALSE, prob = probs)
 ↪  # Sample indices based on probabilities
  data[marIndices, targetCol] <- NA            # Set the target column values
 ↪  to NA
  return(data)                                 # Return the modified dataset
}

# Missing Not at Random (MNAR): Remove values based on the variable itself
introduceMnar <- function(data, column, perc) {
  n <- nrow(data)                               # Number of rows in the
 ↪  dataset
  numMissing <- floor(perc * n)                # Calculate the number of
 ↪  missing values

  # Handle complete cases and probabilities
  completeIndices <- which(!is.na(data[[column]])) # Indices of non-missing
 ↪  values
  nComplete <- length(completeIndices)             # Number of complete values
  numMissing <- min(numMissing, nComplete)         # Adjust number of missing
 ↪  values if necessary

  # Scale the target variable to determine probabilities
  refundsValues <- data[[column]][completeIndices] + abs(min(data[[column]],
 ↪  na.rm = TRUE)) + 1e-6
  probs <- refundsValues / sum(refundsValues)      # Higher values have higher
 ↪  probability of missingness

  mnarIndices <- sample(completeIndices, size = numMissing, replace = FALSE,
 ↪  prob = probs) # Sample indices
  data[mnarIndices, column] <- NA                  # Set the target column
 ↪  values to NA
  return(data)                                     # Return the modified
   ↪  dataset
```

```r
}

# Function to generate datasets with different missingness mechanisms and
 ↪  levels
generateMissingData <- function(data, mechanism, column, levels, driverCol =
 ↪  NULL) {
  results <- list()                              # Initialize an empty list
 ↪  to store results
  for (level in levels) {                        # Loop over missingness
   ↪  levels
    for (i in 1:10) {                            # Repeat the process 10
     ↪  times for each level
      dataCopy <- data                           # Create a copy of the
 ↪  dataset
      set.seed(i + as.numeric(level * 10))       # Set a seed for
       ↪  reproducibility

      # Apply the appropriate missingness mechanism
      if (mechanism == "MCAR") {
        dataMissing <- introduceMcar(dataCopy, column, level)
      } else if (mechanism == "MAR" && !is.null(driverCol)) {
        dataMissing <- introduceMar(dataCopy, column, driverCol, level)
      } else if (mechanism == "MNAR") {
        dataMissing <- introduceMnar(dataCopy, column, level)
      }

      # Store the resulting dataset in the list
      results[[paste0(mechanism, "_", level, "_run", i)]] <- dataMissing
    }
  }
  return(results)                                # Return the list of
   ↪  datasets
}

# Generate datasets for each missingness mechanism and level
missingnessLevels <- c(0.1, 0.3, 0.5, 0.7)       # Define missingness levels
mcarData <- generateMissingData(analysisData, mechanism = "MCAR", column =
 ↪  "Refunds", levels = missingnessLevels)
marData <- generateMissingData(analysisData, mechanism = "MAR", column =
 ↪  "Refunds", levels = missingnessLevels, driverCol = "Total.Revenue")
mnarData <- generateMissingData(analysisData, mechanism = "MNAR", column =
 ↪  "Refunds", levels = missingnessLevels)
```

```r
# Function for imputing missing data using the mice package
imputeData <- function(data) {
  mice(data, method = "pmm", maxit = 5, seed = 42, printFlag = FALSE) #
   ↪ Predictive mean matching
}

# Function for analyzing imputed data
analyzeData <- function(miceModel) {
  fit <- with(miceModel, lm(Refunds ~ Purchased.Item.Count +
 ↪ Refunded.Item.Count + Total.Revenue + Sales.Tax)) # Fit linear model
  pooled <- pool(fit)                     # Pool the results from multiple
 ↪ imputations
  summary(pooled)                         # Return the pooled summary
}

# Function to evaluate all datasets
evaluateDatasets <- function(datasets) {
  results <- list()                       # Initialize an empty list for
 ↪ results
  for (name in names(datasets)) {         # Loop through each dataset
    data <- datasets[[name]]              # Extract the dataset
    if (sum(!is.na(data$Refunds)) > 10) { # Ensure there are sufficient
     ↪ non-missing values
      miceModel <- imputeData(data)       # Perform imputation
      regressionResults <- analyzeData(miceModel) # Analyze the imputed data
      regressionResults$MissingPattern <- name    # Add the pattern name
      results[[name]] <- regressionResults        # Store the results
    }
  }
  bind_rows(results)                      # Combine all results into a single
   ↪ data frame
}

# Analyze and summarize the datasets
mcarResults <- evaluateDatasets(mcarData) # Analyze MCAR datasets
marResults <- evaluateDatasets(marData)   # Analyze MAR datasets
mnarResults <- evaluateDatasets(mnarData) # Analyze MNAR datasets

# Impute and analyze the original dataset (no missing data)
originalMiceModel <- imputeData(analysisData) # Impute original data
originalRegression <- analyzeData(originalMiceModel) # Analyze original data
```

```r
# Plot 1: Coefficient Estimates vs. Missingness Level
ggplot(filteredPlotData, aes(x = Level, y = estimate, color = Pattern, shape
 ↪  = Pattern)) +
  # Add points for individual estimates with slight jitter for visibility
  geom_jitter(width = 0.25, size = 2.5, alpha = 0.8) +
  # Add horizontal dashed lines for the original (non-missing) coefficient
   ↪  estimates
  geom_hline(data = originalCoefficients %>% filter(term %in%
   ↪  c("Purchased.Item.Count", "Refunded.Item.Count", "Sales.Tax",
   ↪  "Total.Revenue")),
             aes(yintercept = OriginalEstimate), linetype = "dashed", color =
              ↪  "black") +
  # Create separate panels for each term, scaling axes independently
  facet_wrap(~term, scales = "free_y") +
  # Define custom colors and shapes for the missingness mechanisms
  scale_color_manual(values = c("MAR"="#E41A1C", "MCAR"="#4DAF4A",
   ↪  "MNAR"="#377EB8")) +
  scale_shape_manual(values = c("MAR"=19, "MCAR"=17, "MNAR"=15)) +
  # Add labels for title, subtitle, axes, and legend
  labs(
    title = "Coefficient Estimates vs. Missingness Level",
    subtitle = "Comparing different missing data mechanisms",
    x = "Missingness Level",
    y = "Coefficient Estimate",
    color = "Mechanism",
    shape = "Mechanism"
  ) +
  # Apply the custom theme for consistent styling
  themeOSUStyle()

# Plot 2: Bias of Coefficient Estimates vs. Missingness Level
ggplot(filteredPlotData, aes(x = Level, y = Bias, color = Pattern, shape =
 ↪  Pattern)) +
  # Add points for bias with jitter
  geom_jitter(width = 0.25, size = 2.5, alpha = 0.8) +
  # Add a horizontal dashed line at 0 (no bias)
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  # Create separate panels for each term with independent axes
  facet_wrap(~term, scales = "free_y") +
  # Define custom colors and shapes for mechanisms
  scale_color_manual(values = c("MAR"="#E41A1C", "MCAR"="#4DAF4A",
   ↪  "MNAR"="#377EB8")) +
  scale_shape_manual(values = c("MAR"=19, "MCAR"=17, "MNAR"=15)) +
  # Add plot labels
  labs(
    title = "Bias of Coefficient Estimates vs. Missingness Level",
    subtitle = "Difference between imputed and original estimates",
    x = "Missingness Level",
    y = "Bias (Estimate - Original)",
    color = "Mechanism",
    shape = "Mechanism"
  ) +
  # Apply custom theme
  themeOSUStyle()

# Plot 3: Standard Error of Estimates vs. Missingness Level
ggplot(filteredPlotData, aes(x = Level, y = std.error, color = Pattern, shape
 ↪  = Pattern)) +
  # Add points for standard errors with jitter
  geom_jitter(width = 0.25, size = 2.5, alpha = 0.8) +
  # Create separate panels for each term with independent axes
  facet_wrap(~term, scales = "free_y") +
  # Define custom colors and shapes for mechanisms
  scale_color_manual(values = c("MAR"="#E41A1C", "MCAR"="#4DAF4A",
   ↪  "MNAR"="#377EB8")) +
  scale_shape_manual(values = c("MAR"=19, "MCAR"=17, "MNAR"=15)) +
  # Add plot labels
  labs(
    title = "Standard Error of Estimates vs. Missingness Level",
    subtitle = "Assessing the variability of estimates",
    x = "Missingness Level",
    y = "Standard Error",
    color = "Mechanism",
    shape = "Mechanism"
  ) +
  # Apply custom theme
  themeOSUStyle()

# Loop through missingness levels to generate bar plots
for (level in missingnessLevels) {
  # Filter data for the current missingness level and compute error bars
  barData <- meanEstimates %>%
    filter(Level == level, Pattern %in% c("MAR", "MCAR", "MNAR"),
           term %in% c("Purchased.Item.Count", "Refunded.Item.Count",
 ↪  "Total.Revenue", "Sales.Tax")) %>%
```

```r
    left_join(originalCoefficients, by = "term") %>%
    mutate(
      lower = meanEstimate - 2 * meanStdError, # Compute lower bound of error
       ↪  bars
      upper = meanEstimate + 2 * meanStdError  # Compute upper bound of error
       ↪  bars
    )

  # Generate bar plot for the current level of missingness
  plot <- ggplot(barData, aes(x = Pattern, y = meanEstimate, fill = Pattern))
↪   +
    # Add bar heights for mean estimates
    geom_col(position = position_dodge(width = 0.9), width = 0.7, alpha =
     ↪  0.8) +
    # Add error bars to show variability
    geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2, position =
     ↪  position_dodge(width = 0.9)) +
    # Create separate panels for each term
    facet_wrap(~term, scales = "free_y") +
    # Define custom colors for mechanisms
    scale_fill_manual(values = c("MAR" = "#E41A1C", "MCAR" = "#4DAF4A",
     ↪  "MNAR" = "#377EB8")) +
    # Add labels for title, subtitle, axes, and legend
    labs(
      title = paste("Mean Coefficient Estimates at", level * 100, "%
       ↪  Missingness"),
      subtitle = "Comparing MAR, MCAR, and MNAR mechanisms",
      x = "Mechanism",
      y = "Mean Coefficient Estimate",
      fill = "Mechanism"
    ) +
    # Apply custom theme and adjust panel and legend styling
    themeOSUStyle() +
    theme(
      strip.text = element_text(face = "bold", color = "black", size = 12),
      axis.title = element_text(face = "bold", size = 12),
      legend.position = "bottom",
      legend.text = element_text(size = 14)
    )

  # Print the plot for the current missingness level
  print(plot)
```

```
}
```