ISSN 2816-8089

REVIEW ARTICLE

Handling Missing Data in Real-World Evidence Studies Managing Missing Data in RWE Studies

Purvi Kalra[*]

Manager, Biostatistics & Programming, Ephicacy Lifescience Analytics, Bangalore, 560076, Karnataka, India

Abstract

Missing data is a pervasive challenge in real-world evidence (RWE) studies, arising from incomplete or inconsistent data collection. Proper handling of missing data is critical to ensure the validity and reliability of study outcomes. This paper explores strategies to address missingness, focusing on mechanisms, methods, and tools available to researchers.

Understanding the underlying mechanism of missingness—Missing Completely at Random (MCAR), Missing at Random (MAR), or Missing Not at Random (MNAR)—is the foundation of effective data management. Methods such as complete case analysis and single imputation are suitable for MCAR scenarios but may introduce bias or underestimate variability. Advanced approaches, like multiple imputation and maximum likelihood estimation, better address MAR data, preserving uncertainty and improving robustness. For MNAR cases, sensitivity analyses are essential to evaluate the impact of missingness on study conclusions.

Innovative tools in R, including mice, missForest, VIM, and naniar, enable effective imputation, visualization, and modeling of missing data. Machine learning techniques and Bayesian approaches offer promising alternatives for complex datasets. Combining methods, such as multiple imputation followed by sensitivity analysis, ensures more reliable inferences.

Best practices emphasize assessing missingness patterns, transparent documentation of assumptions, and thorough reporting of strategies used to address missing data. Adopting these approaches minimizes bias and enhances the credibility of RWE studies, ultimately supporting better-informed healthcare decisions. This paper underscores the importance of a systematic, informed approach to handling missing data in RWE.

Key Words: *Missing data; Real-World Evidence (RWE); Missing data mechanisms; Multiple imputation; Sensitivity analysis; Machine learning; Bayesian approaches; R tools; Data quality; Healthcare decision-making*

[*]Corresponding Author: *Purvi Kalra, Manager, Biostatistics & Programming, Ephicacy Lifescience Analytics, Bangalore, Karnataka, India; Email:* purvi.kalra@ephicacy.com

***Received Date**: December 31, 2024, **Accepted Date**: January 23, 2025, **Published Date**: February 12, 2025*

***Citation:** Kalra P. Handling Missing Data in Real-World Evidence Studies Managing Missing Data in RWE Studies. Int J Bioinfor Intell Comput. 2025;4(1):58-81.*

1. Introduction

In an era where real-world evidence (RWE) increasingly shapes healthcare decisions and policies, the data quality underlying these studies is paramount. Yet, missing data remains an unavoidable and persistent challenge, threatening the validity of findings and the reliability of conclusions. The absence of systematic approaches to handle this issue can lead to biased outcomes and undermine the credibility of RWE studies, which are often pivotal in translating research into actionable healthcare insights.

This paper addresses the pressing need for robust strategies to manage missing data, exploring the mechanisms of missingness, advanced imputation techniques, and innovative tools that empower researchers to mitigate its impact. By adopting a structured and informed approach, researchers can ensure the robustness of their analyses, ultimately supporting more accurate and impactful healthcare decisions.

1.1. Understanding the missing data mechanism

Handling missing data effectively begins with identifying the mechanism that governs its occurrence. This understanding forms the cornerstone for selecting the most appropriate methods to address missingness, as different mechanisms require tailored approaches.

1.2. Missing Completely at Random (MCAR)

In the MCAR mechanism, the probability of data being missing is entirely unrelated to any observed or unobserved variables in the dataset. For instance, if a random technical issue causes the loss of a subset of survey responses, the missing data can be considered MCAR. The randomness of this missingness implies no bias in the dataset, making MCAR the most straightforward scenario to address. Researchers often employ methods like complete case analysis, which involves analyzing only the cases with complete data, or single imputation methods, assuming no systematic difference between missing and observed data. However, these approaches may reduce sample size and statistical power.

1.3. Missing at Random (MAR)

The MAR mechanism occurs when the missingness is related to the observed data but not to the missing values themselves. For example, in a clinical study, older participants might be less likely to complete follow-up surveys, and their missingness is associated with the observed age variable. Addressing MAR requires more sophisticated methods like multiple imputation or maximum likelihood estimation, which account for the relationship between missingness and observed data. These methods preserve uncertainty in imputed values and improve robustness, making them suitable for scenarios where the MAR assumption holds.

1.4. Missing Not at Random (MNAR)

In the MNAR mechanism, the missingness depends on the unobserved (missing) data itself, which poses the most challenging scenario. For example, in a study measuring income, individuals with higher incomes might be more likely to omit their responses due to privacy concerns, resulting in systematic missingness tied to the unreported values. Since MNAR cannot be fully addressed using observed data alone, sensitivity analyses become crucial.

These analyses explore how various assumptions about the missingness mechanism impact study conclusions, ensuring that the results remain valid under different plausible scenarios.

1.5. Importance of identifying the mechanism

Accurately identifying the missing data mechanism is critical for guiding the choice of an appropriate handling method. Misclassifying the mechanism—for instance, treating MNAR data as MAR—can lead to biased estimates, underestimated variability, and erroneous conclusions. Researchers should begin with an assessment of missingness patterns, using statistical tests or visualization tools to hypothesize the most likely mechanism. Combining this understanding with domain knowledge further enhances the accuracy of the identification process.

The choice of missingness mechanism—Missing Completely at Random (MCAR), Missing at Random (MAR), or Missing Not at Random (MNAR)—significantly impacts the interpretation of results in RWE studies. If data are MCAR, the missingness does not depend on observed or unobserved values, allowing unbiased analyses with complete-case data. Under MAR, missingness depends on observed data, and proper imputation methods can minimize bias. However, for MNAR, where missingness depends on unobserved values, failing to account for the underlying mechanism can lead to biased estimates and misleading conclusions. By systematically addressing the underlying mechanism, researchers can implement effective strategies that minimize bias, preserve data integrity, and bolster the reliability of RWE study outcomes.

2. Methods to Handle Missing Data

Effectively handling missing data is essential for ensuring the validity and reliability of real-world evidence (RWE) studies. The choice of method depends on the missing data mechanism (MCAR, MAR, or MNAR) and the specific research context. Below, we explore key methods, their applications, advantages, and limitations.

2.1. Complete case analysis (Listwise Deletion)

Complete Case Analysis (CCA), often referred to as Listwise Deletion, is a common method to handle missing data in real-world evidence (RWE) settings. In this approach, only the cases (observations) with no missing data for the variables of interest are retained for analysis.

This technique is straightforward but assumes that the missing data mechanism is either Missing Completely at Random (MCAR) or minimally impactful if Missing at Random (MAR). Using CCA in clinical trials has its pros and cons, especially in RWE studies where missingness is prevalent due to the less controlled environment.

Application: Best suited for data missing completely at random (MCAR), where missingness does not introduce bias.
Advantages: Easy to implement and interpret. Avoids assumptions needed for data imputation methods. Supported in most statistical software, including R.
Limitations: Reduction in sample size can decrease statistical power. If data are not MCAR, excluding cases may lead to biased estimates. Findings may not represent the population due to exclusion of incomplete cases [1-3].
Example: Imagine a study using electronic health records (EHRs) to assess the effectiveness of a new diabetes medication. Variables include:

- Patient age
- HbA1c levels
- Body mass index (BMI)
- Medication adherence
- Outcome: Reduction in HbA1c after 6 months

Some data may be missing due to incomplete records, patient dropout, or unrecorded variables.

- Simulate a Dataset
  ```
  set.seed(123)
  library(dplyr)
  /Simulating a dataset with missing values /
  data <- tibble( patient_id = 1:20,
  age = c(sample(40:70, 15, replace = TRUE), rep(NA, 5)),
  hbA1c = c(runif(18, 6, 10), rep(NA, 2)),
  bmi = c(runif(17, 20, 35), rep(NA, 3)),
  adherence = sample(c(0, 1), 20, replace = TRUE),
  outcome = c(runif(18, -1, -0.1), rep(NA, 2)) )
  ```

- Identify Missing Data
  ```
  summary(data)
  sapply(data, function(x) sum(is.na(x))) # Count missing values
  ```

- Perform Complete Case Analysis
  ```
  complete_cases <- na.omit(data) /Remove rows with any missing values/
  print(complete_cases)
  ```

- Fit a Model
  ```
  /Example: Linear regression using complete cases/
  model <- lm(outcome ~ age + hbA1c + bmi + adherence, data = complete_cases)
  summary(model)
  ```

2.2. Single imputation

Single imputation is a technique used to handle missing data by replacing missing values with a plausible estimate, such as the mean, median, mode, or predictions from a regression model. In real-world evidence (RWE) studies, this method is often used to address missing data resulting from the less controlled nature of such environments (e.g., electronic health records or patient-reported outcomes) [1,2].

Advantages: Straightforward and easy to implement. Retains all cases for analysis. Can be tailored (mean, median, regression) based on the nature of the data.
Limitations: Ignores uncertainty in the imputed values. Assumes the imputed value is correct, which may not reflect the true data distribution. Not Suitable for MAR or MNAR, Assumes MCAR or minimal impact under MAR [4].
Example: A study evaluates the effectiveness of a new hypertension drug using RWE data from patient records. The dataset includes:

- Patient age
- Blood pressure (BP) before treatment
- Body mass index (BMI)

- Adherence (binary)
- Outcome: Reduction in BP after 6 months

Some variables have missing data.

- Simulate a Dataset

  ```
  set.seed(123)
  library(dplyr) data <- tibble( patient_id = 1:20,
  age = c(sample(40:70, 18, replace = TRUE), rep(NA, 2)),
  bp_before = c(runif(19, 120, 180), NA),
  bmi = c(runif(17, 20, 35), rep(NA, 3)),
  adherence = sample(c(0, 1), 20, replace = TRUE),
  outcome = c(runif(18, -10, -5), rep(NA, 2)) )
  print(data)
  ```

- Visualize Missing Data

  ```
  library(ggplot2)
  library(naniar)
  gg_miss_var(data) + labs(title = "Missing Data Pattern")
  ```

- Mean Imputation

  ```
  data_mean_imputed <- data %>%
  mutate ( age = ifelse(is.na(age), mean(age, na.rm = TRUE), age),
  bp_before = ifelse(is.na(bp_before), mean(bp_before, na.rm = TRUE),
  bp_before),
  bmi = ifelse(is.na(bmi), mean(bmi, na.rm = TRUE), bmi),
  outcome = ifelse(is.na(outcome), mean(outcome, na.rm = TRUE), outcome))
  ```

- Median Imputation

  ```
  data_median_imputed <- data %>%
  mutate( age = ifelse(is.na(age), median(age, na.rm = TRUE), age),
  bp_before =ifelse(is.na(bp_before), median(bp_before,na.rm =TRUE),
  bp_before),
  bmi = ifelse(is.na(bmi), median(bmi, na.rm = TRUE), bmi),
  outcome = ifelse(is.na(outcome), median(outcome, na.rm = TRUE), outcome))
  ```

- Regression Imputation

  ```
  /Fit a regression model for BMI as an example/
  fit <- lm(bmi ~ age + bp_before + adherence, data = data, na.action =
  na.exclude) data$predicted_bmi <- predict(fit, newdata = data)
  data_regression_imputed <- data %>%
  mutate(bmi = ifelse(is.na(bmi), predicted_bmi, bmi)) %>%
  select(-predicted_bmi)
  ```

2.3. Multiple imputation

Multiple Imputation (MI) is a robust method for handling missing data, commonly used in real-world evidence (RWE) studies. Unlike single imputation, MI accounts for uncertainty by generating multiple plausible datasets with different imputed values and pooling the results. This ensures valid statistical inference while minimizing bias [1,2].

Application: Particularly useful for data missing at random (MAR).
Advantages: By creating multiple datasets, MI incorporates variability in imputed values. Ensures appropriate estimates of standard errors and p-values.
Limitations: Requires multiple analyses and pooling. Requires careful specification of the imputation model. Relies on data being Missing at Random (MAR) [4,5].
Example: An RWE study evaluates the effect of a new diabetes drug using data from electronic health records. Variables include:

- Patient age
- Baseline HbA1c
- BMI
- Adherence (binary)
- Outcome: HbA1c reduction after 6 months

```
Summary(data)
```

- Visualize Missing Data
  ```
  library(naniar)
  gg_miss_var(data) + labs (title = "Missing Data Pattern")
  ```

- Perform Multiple Imputation
  ```
  library(mice)
  /Impute missing data/
  imputed_data <- mice (data, m = 5, method = "pmm", maxit = 10, seed = 123)
  ```

- Analyze Imputed Datasets
  ```
  /Example: Fit a linear model on each dataset/
  fit <- with (imputed_data, lm(outcome ~ age + hbA1c + bmi + adherence))
  /Pool results across imputations/ pooled_results <- pool(fit)
  summary(pooled_results)
  ```

- Visualize Results
  ```
  /Density plot of imputed values for HbA1c/
  Stripplot (imputed_data, hbA1c ~ .imp, pch = 20, cex = 1.5, col = "blue", main =
  "Imputed Values for HbA1c")
  ```

2.4. Regression models

Regression models are powerful statistical tools used in real-world evidence (RWE) studies to analyze relationships between variables and adjust for confounders. These models are essential for deriving valid and generalizable insights from observational data in clinical trials, where randomization may not be possible.

Application: Suitable when missingness aligns with MAR assumptions.
Advantages: Can accommodate continuous, binary, time-to-event, and count outcomes. Essential in non-randomized RWE studies. Clear insights into the relationship between variables [6].
Limitations: Ensure assumptions (e.g., linearity, independence) are met. Overfitting can occur with too many predictors. RWE data may have biases affecting external validity [7-9].
Example: A real-world study evaluates the effect of a new drug for hypertension. Variables include:

- Patient age

- Baseline systolic blood pressure (SBP)
- Adherence (binary: 1 = adherent, 0 = non-adherent)
- Drug type (binary: 1 = new drug, 0 = standard drug)
- Outcome: SBP reduction after 6 months

Simulate Dataset

-
  ```
  set.seed(123)
  library(dplyr)
  data <- tibble( patient_id = 1:100,
  age = sample(40:80, 100, replace = TRUE),
  baseline_sbp = rnorm(100, mean = 150, sd = 10),
  adherence = sample(c(0, 1), 100, replace = TRUE),
  drug_type = sample(c(0, 1), 100, replace = TRUE),
  sbp_reduction = rnorm(100, mean = -15, sd = 5) + 0.5 * adherence + 2 * drug_type)
  summary(data)
  ```

- Fit a Linear Regression Model
  ```
  /Linear regression to assess SBP reduction/
  fit_lm <- lm(sbp_reduction ~ age + baseline_sbp + adherence + drug_type, data =
  data)
  summary(fit_lm)
  ```
  Key Outputs:
  Coefficients: Show the effect of each predictor on SBP reduction.
  Adjusted R-squared: Indicates the proportion of variance explained by the model.

- Fit a Logistic Regression Model
  For example, converting the outcome to a binary variable (e.g., significant reduction = SBP reduction > 10 mmHg).
  ```
  data <- data %>%
  mutate(significant_reduction = ifelse(sbp_reduction > -10, 1, 0))
  fit_logit <- glm(significant_reduction ~ age + baseline_sbp + adherence + drug_type,
  data = data, family = binomial)
  summary(fit_logit)
  /Odds ratios/ exp(coef(fit_logit))
  ```

- Cox Proportional Hazards Model
  For time-to-event outcomes like time to achieve a 10 mmHg reduction in SBP.
  ```
  library(survival)
  /Simulating time-to-event data/ data <- data %>%
  mutate(time_to_reduction = rexp(100, rate = 0.1))
  fit_cox <- coxph(Surv(time_to_reduction, significant_reduction) ~ age + adherence
  + drug_type, data = data) summary(fit_cox)
  ```

- Visualize Results
  ```
  /Visualization of linear regression predictions/
  library(ggplot2) data$predicted <- predict(fit_lm)
  ggplot(data, aes(x = predicted, y = sbp_reduction)) +
  geom_point() +
  geom_smooth(method = "lm") +
  ```

labs(title = "Predicted vs Actual SBP Reduction", x = "Predicted Reduction", y = "Actual Reduction")

2.5. Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation (MLE) is a widely used statistical method to estimate the parameters of a model by maximizing the likelihood function, which measures the probability of observing the given data under different parameter values. In the context of real-world evidence (RWE) in clinical trials, MLE can be applied to estimate parameters for models that describe treatment effects, patient outcomes, or other clinical phenomena using observational or real-world data [10].

Application: Commonly used in mixed-effects models, structural equation modeling, and other advanced statistical frameworks [11].
Advantages: MLE can incorporate covariates to model individual differences. Extensions like expectation maximization (EM) can handle missing data. Using covariates in the model helps adjust for confounding [12].
Limitations: Requires strong assumptions about the data and may be complex to implement.
Example: Suppose we want to evaluate the effectiveness of a new diabetes medication on HbA1c levels using observational data. The data includes patient demographics, baseline HbA1c levels, and HbA1c measurements after treatment initiation. Because this is real- world data, it may have challenges like missing data, heterogeneity, and potential confounding factors [13].

We aim to estimate the treatment effect using a linear regression model, where the change in HbA1c levels (ΔHbA1c\Delta HbA1cΔHbA1c) is modeled as a function of treatment (XXX) and covariates (ZZZ).

$$\Delta HbA1c_i = \beta_0 + \beta_1 X_i + \beta_2 Z_i + \in_i \tag{1}$$

where:
β0\beta_0β0: Intercept
β1\beta_1β1: Treatment effect
β2\beta_2β2: Effect of covariates
ϵi: Error term, assumed to follow a normal distribution N(0,σ2)

Applying MLE

MLE estimates the parameters β0, β1, β2,
σ2 by maximizing the likelihood function:

$$L\left(\beta_0, \beta_1, \beta_2, \sigma^2 \mid data\right) = \pi_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(y_i - \left(\beta_0 + \beta_1 X_i + \beta_2 Z_i\right)\right)^2}{2\sigma^2} \right) \tag{2}$$

- Simulated example data set.seed(123)
  n <- 100
  X <- rbinom(n, 1, 0.5) /Treatment indicator (0 or 1)/ Z <- rnorm(n, 50, 10) /Covariate (e.g., age)/
  epsilon <- rnorm(n, 0, 1) /Error term/ beta_0 <- 1
  beta_1 <- -0.5
  beta_2 <- 0.1 y <- beta_0 + beta_1 * X + beta_2 * Z + epsilon /Change in HbA1c/

```
/Log-likelihood function/ log_likelihood <- function(params) { beta_0 <- params[1]
beta_1 <- params[2] beta_2 <- params[3] sigma <- params[4]
/Predicted values/
y_pred <- beta_0 + beta_1 * X + beta_2 * Z
/Log-likelihood
ll <- -0.5 * n * log(2 * pi) - n * log(sigma) - sum((y - y_pred)^2) / (2 * sigma^2) return(-
ll) /Negative because optim minimizes/}
/Initial parameter guesses/
init_params <- c(beta_0 = 0, beta_1 = 0, beta_2 = 0, sigma = 1)
/Optimization using MLE/fit <- optim(init_params, log_likelihood, method =
"BFGS", hessian = TRUE) fit$par /Estimated parameters/
```

- Interpretation
  The MLE estimates of β0, β1, β2, σ are the parameter values that maximize the likelihood of observing the given data [14,15].
  β1 represents the estimated treatment effect on ΔHbA1c.

2.6. Weighting methods

Weighting methods are critical in RWE to address confounding and ensure the observed treatment effects are valid and reliable. They allow researchers to balance covariates across treatment groups, particularly in non-randomized studies. Common weighting methods include Inverse Probability of Treatment Weighting (IPTW), propensity score weighting, and calibrated weighting.

Application: Appropriate for MAR mechanisms.
Advantages: Ensures representativeness of the dataset by compensating for missingness.
Limitations: Weighting cannot adjust for unmeasured covariates. Sensitivity analyses are recommended. May arise from very small or large propensity scores. Address by trimming or using stabilized weights. Incorrect specification of the propensity score model can bias results.
Example: Inverse Probability of Treatment Weighting (IPTW)
You are analyzing real-world data from a healthcare database to evaluate the effect of a new antihypertensive drug on systolic blood pressure (SBP) reduction. Since this is an observational study, the treatment assignment is not random and may depend on patient characteristics such as age, baseline SBP, and comorbidities [16]. To estimate the treatment effect unbiasedly, we use IPTW to create a pseudo-population where the distribution of covariates is balanced across treatment groups [9,14,17,18].

Steps

1. Model the propensity Score
   The propensity score $\left(e\left(X\right)\right)$ is the probability of receiving the treatment given covariates $\left(X\right)$:
   $$e\left(X\right)=P\left(T=1|X\right) \tag{3}$$
   This can be modeled using logistic regression.
2. Calculate Weights
   The IPTW weight for each individual is computed as:
   $$w_i=\frac{1}{e\left(X_i\right)}ifT_i=1, w_i=\frac{1}{1-e\left(X_i\right)}ifT_i=0 \tag{4}$$

3. Estimate the Treatment Effect
   Use weighted regression or weighted mean differences to estimate the treatment effect.

- Simulated data

```
set.seed(123)
 n <- 200
age <- rnorm(n, mean = 60, sd = 10) /Age/
baseline_sbp <- rnorm(n, mean = 140, sd = 15) /Baseline SBP/
comorbidity <- rbinom(n, 1, 0.3) /Comorbidity (0/1)/
treatment <- rbinom(n, 1, plogis(0.5 * (age - 60) + 0.3 * comorbidity - 0.2 *
baseline_sbp)) /Treatment assignment/
sbp_reduction <- -5 * treatment + 0.2 * baseline_sbp + rnorm(n, 0, 5) /SBP reduction/

/Combine into a data frame/
data <- data.frame(age, baseline_sbp, comorbidity, treatment, sbp_reduction)

/Propensity score model/
propensity_model <- glm(treatment ~ age + baseline_sbp + comorbidity, family =
binomial, data = data)
data$propensity_score <- predict(propensity_model, type = "response")

/Calculate IPTW weights/
data$weight <- ifelse(data$treatment == 1,
1 / data$propensity_score,
1 / (1 - data$propensity_score))

/Check balance before and after weighting/ library(tableone)
unweighted_table<-CreateTableOne(vars=c("age","baseline_sbp", "comorbidity"),
strata="treatment",data = data, test = FALSE) weighted_table<-
svyCreateTableOne(vars=c("age","baseline_sbp", comorbidity"),
strata = "treatment",
data=svydesign(ids=~1, weights = ~weight, data = data), test = FALSE)
print(unweighted_table) print(weighted_table)

/Weighted regression/
weighted_model<-lm(sbp_reduction ~ treatment,data=data,weights = data$weight)
summary(weighted_model)
```

Results

- Balance Check: Covariates should be balanced between treatment groups after weighting (e.g., standardized mean difference < 0.1).
- Weighted Regression: The coefficient of the treatment variable represents the estimated average treatment effect (ATE) adjusted for confounding .

2.7. Other weighting methods

- **Stabilized weights:** Prevent extreme weights by normalizing with the marginal probability of treatment.

$$w_i = \frac{P(T)}{e(X_i)} if T_i = 1, w_i = \frac{1-P(T)}{1-e(X_i)} if T_i = 0 \tag{5}$$

- **Overlap weights:** Focus on the population with overlapping propensity scores, emphasizing regions of equipoise.

$$w_i = T_i.(1-e(X_i)) + (1-T_i).e(X_i) \tag{6}$$

- **Entropy balancing:** Directly balance covariates by reweighting the data without explicitly modeling the propensity score.
- **Generalizes propensity scores:** Extend the propensity score framework for continuous or multinational treatments.

Sensitivity analysis

Sensitivity analysis is crucial in real-world evidence (RWE) to assess the robustness of study results against potential biases, such as unmeasured confounding, missing data, model assumptions, and analytical choices. In clinical trials, sensitivity analysis helps determine whether conclusions remain consistent under varying scenarios or assumptions [19,20].

Application: Crucial for MNAR scenarios, where the missingness depends on unobserved data.
Advantages: Provides insights into the potential impact of missingness on results.
Limitations: Does not resolve missing data but assesses its influence on findings.
Example: Using observational data, you are evaluating the effect of a new cholesterol-lowering drug on LDL cholesterol levels. While propensity score weighting was used to balance observed covariates, you suspect there might be unmeasured confounders (e.g., dietary habits) that could bias the results. Sensitivity analysis helps quantify the potential impact of such unmeasured confounders.

Steps

- Base Analysis: Perform the primary analysis using propensity score-weighted regression to estimate the treatment effect.
- Quantify Unmeasured Confounding: Use sensitivity analysis techniques like E-values, bounding approaches, or bias formulas to evaluate how strong an unmeasured confounder would need to be to explain away the observed effect [14].
- Vary Key Parameters: Modify assumptions (e.g., missing data mechanisms, model specifications) and assess their impact on results.

Step 1: Primary analysis

- Simulated Data set.seed(123)

  ```
  n <- 300
  age <- rnorm(n, mean = 60, sd = 10) /Age/
  baseline_ldl <- rnorm(n, mean = 150, sd = 20) /Baseline LDL/ smoking <- rbinom(n, 1, 0.3) /Smoking status/
  treatment<-rbinom(n, 1, plogis(0.5 *(age - 60) + 0.4 * smoking - 0.3 * baseline_ldl)) /Treatment/
  ldl_reduction <- -10 *treatment + 0.2 * baseline_ldl + smoking * 5 + rnorm(n, 0, 5) /LDL reduction/
  ```

- /Combine data/
  ```
  data <- data.frame(age, baseline_ldl, smoking, treatment, ldl_reduction)
  ```

- /Propensity score model/
  ```
  propensity_model <- glm(treatment ~ age + baseline_ldl + smoking, family =
  binomial, data = data)
  data$propensity_score <- predict(propensity_model, type = "response")
  ```

- /IPTW weights/
  ```
  data$weight <- ifelse(data$treatment == 1, 1 / data$propensity_score, 1 / (1 -
  data$propensity_score))
  ```

- /Weighted regression/
  ```
  primary_model <- lm(ldl_reduction ~ treatment, data = data, weights = data$weight)
  summary(primary_model)
  ```

Step 2: Sensitivity analysis for unmeasured confounding

- E-value Calculation

The E-value quantifies the minimum strength of association an unmeasured confounder must have with both the treatment and the outcome to explain away the observed effect fully [18].

```
library(EValue)
/Estimate from primary analysis/
treatment_effect <- coef(primary_model)["treatment"] conf_int <-
confint(primary_model)["treatment", ]
```

```
/Calculate E-value/
evalue<-evalues.OLS(treatment_effect,se=
summary(primary_model)$coefficients["treatment", "Std. Error"]) print(evalue)
```

Step 3: Varying parameters
Scenario: Adding Simulated Confounder
Simulate an unmeasured confounder (UUU) and re-estimate the treatment effect.

- Simulate an unmeasured confounder (e.g., dietary habits)
  ```
  data$unmeasured <- rbinom(n, 1, plogis(0.3 * data$age + 0.5 * data$smoking))

  /Include the unmeasured confounder in the model/
  adjusted_model <- lm(ldl_reduction ~ treatment + unmeasured, data = data, weights
  = data$weight) summary(adjusted_model)
  ```

Results

1. E-value: Provides the robustness of the effect size against unmeasured confounding.
2. Simulated Confounder: Adjusted models can demonstrate the impact of potential unmeasured variables.
3. Missing Data: Analyzing multiple missing data mechanisms helps evaluate the sensitivity of results to different assumptions.

3. Machine Learning Algorithms

Machine learning (ML) algorithms are increasingly used in real-world evidence (RWE) to derive insights from large, complex, and heterogeneous datasets [21]. They are applied for predicting patient outcomes, identifying treatment responders, detecting adverse events, and uncovering patterns in healthcare data. These techniques complement traditional methods by handling non-linear relationships, high-dimensional data, and interactions among covariates [14, 22, 23].

3.1. Key use cases of ML in RWE

1. **Propensity Score Modeling**: Using ML for robust estimation of propensity scores in non-randomized studies.
2. **Predictive Modeling**: Identifying patients likely to respond to treatments.
3. **Risk Stratification:** Categorizing patients based on disease progression or treatment risk.
4. **Clustering:** Unsupervised learning to discover subpopulations or treatment patterns.
5. **Causal Inference:** Estimating treatment effects using algorithms like targeted maximum likelihood estimation (TMLE) or double machine learning (DML).

Advantages: Adaptable to various data types and capable of handling non-linear relationships.
Limitations: May overfit data and require careful validation.
Example: You want to predict whether patients with Type 2 Diabetes (T2D) will achieve HbA1c < 7% after initiating a new antidiabetic drug, using real-world data (EHRs and claims). The dataset includes demographic, clinical, and treatment information.

Steps

1. **Data preparation**
   Preprocess data (handle missing values, encode categorical variables, standardize numeric variables).
   Split the data into training and testing sets.
2. **Algorithm selection**
   Use classification algorithms like logistic regression, random forests, gradient boosting machines (GBM), or neural networks for binary outcome prediction.
3. **Model training**
   Train the model on the training set and tune hyperparameters.
4. **Model evaluation**
   Evaluate model performance using metrics like accuracy, AUC-ROC, precision, and recall.
5. **Interpretation**
   Use tools like SHAP (SHapley Additive exPlanations) to interpret feature importance

- Load necessary libraries

  ```
  library(caret) /For model training/
  library(randomForest) /Random Forest/
  library(pROC) /ROC curves/
  ```

```
library(dplyr) /Data manipulation/
library(ggplot2) /Visualization/
```

- Simulated Data
  ```
  set.seed(123)
  n <- 500
  age <- rnorm(n, 60, 10) /Age/
  bmi <- rnorm(n, 28, 5) /BMI/
  baseline_hba1c <- rnorm(n, 8, 1.5) /Baseline HbA1c/
  drug <- rbinom(n, 1, 0.5) /Drug (new=1, standard=0)
  duration <- rnorm(n, 6, 2) /Treatment duration in months/
  achieved_target <- rbinom(n, 1, plogis(-0.1*age + 0.05*bmi - 0.5*baseline_hba1c
  + 0.8*drug)) /Outcome/
  data <- data.frame(age, bmi, baseline_hba1c, drug, duration, achieved_target)
  ```

- /Split data into training and test sets/
  ```
  set.seed(123)
  train_index <- createDataPartition(data$achieved_target, p = 0.7, list = FALSE)
  train_data <- data[train_index, ]
  test_data <- data[-train_index, ]
  ```

- /Train Random Forest model/
  ```
  set.seed(123) rf_model <- randomForest(achieved_target ~ ., data = train_data, ntree
  = 500, importance = TRUE)
  ```

- /Evaluate model/
  ```
  rf_predictions <- predict(rf_model, test_data, type = "response")
  confusion_matrix<-confusionMatrix(as.factor(rf_predictions),
  as.factor(test_data$achieved_target))
  print(confusion_matrix)
  ```

- /ROC curve/
  ```
  roc_curve<-roc(test_data$achieved_target,as.numeric(rf_predictions))
  plot(roc_curve, main = "ROC Curve for Random Forest Model")
  ```

- /Feature Importance/
  ```
  importance <- varImpPlot(rf_model, main = "Variable Importance")
  ```

Results

1. **Model Performance:**

- Confusion matrix provides accuracy, sensitivity, and specificity.
- ROC-AUC quantifies the model’s ability to discriminate between classes.

2. **Feature Importance:**

- Identifies key predictors of achieving HbA1c < 7%.

3.2. Bayesian methods

Bayesian methods are increasingly used in clinical trials and real-world evidence (RWE) to incorporate prior knowledge, handle uncertainty, and update evidence dynamically as data accumulate. These methods are especially valuable in RWE due to their ability to model complex data structures, address missing data, and account for heterogeneity [24].

- **Key applications of bayesian methods in RWE**

  i) **Hierarchical modeling**: Handling variability across patient populations, sites, or time periods [25].
  ii) **Bayesian network meta-analysis**: Combining evidence from multiple sources to assess treatment efficacy.
  iii) **Bayesian adaptive trials**: Dynamically modifying trial designs based on interim data.
  iv) **Causal inference:** Estimating treatment effects while incorporating prior information about confounders.
  v) **Missing data:** Imputing missing values using Bayesian approaches.

Advantages: Ability to incorporate prior information and handle complex data structures. Provides full posterior distributions, enabling direct probabilistic statements. Naturally incorporates uncertainty in parameter estimates. Combines data from various sources, including historical and real-world data [26,27].
Limitations: Bayesian methods, especially MCMC, can be computationally expensive. Results are sensitive to the choice of priors, which may introduce bias if poorly chosen. Requires expertise to build, validate, and interpret Bayesian models [28-31].
Example: You are analyzing the effect of a new antihypertensive drug on systolic blood pressure (SBP) reduction across multiple hospitals. Real-world data often include site- level variability, which can be modeled using a Bayesian hierarchical approach.

Steps

1. **Data setup**
   Prepare data for modeling SBP reduction as a function of treatment, accounting for hospital-level variability.
2. **Define the model**
   Use a hierarchical structure:
   Level 1: Patient-level SBP reduction.
   Level 2: Hospital-specific effects.
3. **Set priors**
   Incorporate prior knowledge about the treatment effect and variability.
4. **Posterior inference**
   Use Markov Chain Monte Carlo (MCMC) methods to estimate posterior distributions.
5. **Interpret results**
   Summarize posterior distributions to draw conclusions about treatment effects.

- Simulated Data

```
set.seed(123)
n_hospitals <- 10
patients_per_hospital <- 50 n <- n_hospitals * patients_per_hospital
hospital <- rep(1:n_hospitals, each = patients_per_hospital)
treatment <- rbinom(n, 1, 0.5) # 1 = treated, 0 = control
hospital_effect <- rnorm(n_hospitals, mean = 0, sd = 2)
baseline_sbp <- rnorm(n, mean = 140, sd = 10)
sbp_reduction <- -5 * treatment + hospital_effect[hospital] + rnorm(n, mean = 0, sd
= 5)
data <- data.frame(hospital, treatment, baseline_sbp, sbp_reduction)
```

- Bayesian Hierarchical Model Using rstanarm

  ```
  /Load library/
  library(rstanarm)
  /Define Bayesian hierarchical model/
  model <- stan_lmer(
  sbp_reduction ~ treatment + (1 | hospital),
  data = data, prior = normal(0, 5), /Weakly informative prior for fixed effects/
  prior_intercept = normal(0, 10), /Prior for intercept/
  prior_covariance = decov(1, 1) # Prior for random effects covariance)
  /Summary of results/ print(summary(model))
  /Posterior predictive checks/ pp_check(model)
  ```

Results

1. **Treatment effect**: The posterior distribution provides credible intervals for the effect of treatment on SBP reduction.
2. **Hospital variability**: Estimates of between-hospital variability highlight heterogeneity.
3. **Posterior predictive checks**: Ensure model fit and plausibility of results.

3.3. Choosing the right method

The choice of a method to handle missing data should be guided by:

- **Mechanism of missingness:** Understanding whether the data is MCAR, MAR, or MNAR is foundational.
- **Study context:** Consideration of study design, sample size, and computational resources.
- **Study goals:** Balancing simplicity, computational demands, and the need for robust and unbiased estimates.

By systematically applying these methods, researchers can minimize the biases introduced by missing data, ensuring the integrity and reliability of RWE studies.

Future advancements in machine learning and Bayesian approaches offer transformative solutions for handling missing data in complex real-world evidence (RWE) datasets. Machine learning methods, such as Random Forests, Deep Learning, and time-series models like LSTMs, excel in capturing nonlinear relationships, handling high-dimensional data, and imputing temporal patterns. Bayesian approaches enhance uncertainty quantification, incorporate prior domain knowledge, and can explicitly model Missing Not at Random (MNAR) mechanisms. Hybrid methods, like Bayesian Neural Networks, combine the strengths of both paradigms, enabling robust and transparent imputation. Together, these advancements promise greater scalability, accuracy, and trustworthiness in addressing the challenges of missing data in RWE.

3.4. Tools and libraries in R for handling missing data

R offers a rich ecosystem of packages designed to handle missing data effectively. These tools support imputation, visualization, and exploratory analysis, enabling researchers to address missingness systematically and efficiently. Below is an overview of key R libraries and their features.

3.5. MICE (Multivariate imputation by chained equations)

Multivariate Imputation by Chained Equations (MICE), also known as fully conditional specification or sequential regression multiple imputation, is a statistical method used to handle missing data. It is widely employed in clinical research, epidemiology, and other fields where incomplete datasets are common. MICE estimate missing values by iteratively imputing them based on other variables in the dataset, considering the multivariate relationships.

- **Purpose:** Implements multiple imputation by generating plausible values for missing data based on chained equations.
- **Key Features:**
  Supports various data types, including categorical and continuous variables.
  Produces multiple completed datasets for robust analysis.
  Results can be combined using Rubin's rules for inference.
- **Example:**
  R
  ```
  library(mice)
  imputed_data<-mice(data,method="pmm",m=5)
  complete_data <- complete (imputed_data, 1)
  ```

3.6. Miss forest

MissForest is a non-parametric imputation method based on the Random Forest algorithm. It is widely used to handle missing data in datasets with complex, nonlinear relationships and interactions between variables. Unlike parametric methods, missForest does not assume a specific distribution of the data, making it a versatile and robust choice for imputing missing values.

- **Purpose:** Utilizes Random Forests to impute missing values for both categorical and continuous variables.
- **Key Features:**
  Handles non-linear relationships and complex interactions in the data.
  Retains a low level of imputation error, especially for large datasets.
- **Example:**
  R
  ```
  library(missForest)
  imputed_data <- missForest(data)
  ```

3.7. Hmisc

The Hmisc package in R, developed by Frank E. Harrell Jr., is a comprehensive library designed for data analysis, visualization, and manipulation, with a strong focus on missing data handling. It is particularly useful in statistical modeling, medical research, and clinical data analyses. The package provides tools for data summary, imputation, descriptive statistics, and advanced visualization, making it a versatile choice for data scientists and biostatisticians.

- **Purpose**: Provides tools for single imputation and descriptive statistics.
- **Key Features**:

Enables simple imputation using mean, median, or regression models.
Offers comprehensive utilities for data exploration and summaries.

- **Example:**
  R
  library(Hmisc)
  imputed_data <- impute(data$variable, fun = mean)

3.8. VIM (Visualization and imputation of missing data)

The VIM package in R provides tools for the visualization and imputation of missing data, with a strong focus on exploring the structure and patterns of missingness. It is widely used in research fields like epidemiology, clinical trials, and social sciences, where understanding the nature of missing data is crucial before applying imputation methods. The package integrates methods for identifying missing data patterns, diagnosing the impact of missingness, and implementing various imputation techniques. Additionally, its visualization features help in evaluating the quality of imputations.

- **Purpose:** Specializes in visualizing missing data patterns and performing imputation.
- **Key Features:**
  Generates graphical representations like aggr plots and scatter plots with missing values highlighted.
  Facilitates understanding of missingness mechanisms.
- **Example:**
  R
  library(VIM)
  aggr_plot <- aggr(data, col = c("navyblue", "red"), numbers = TRUE, sortVars = TRUE)

3.9. Naniar

The naniar package in R is a powerful tool for handling, visualizing, and imputing missing data. Its capabilities make it particularly relevant for real-world evidence (RWE) studies, where missing data is common due to the nature of observational datasets.

- **Purpose:** Offers comprehensive tools for exploring, visualizing, and imputing missing data.
- **Key Features:**
  Simplifies missing data exploration with functions like gg_miss_var and gg_miss_case.
  Provides flexible options for imputation and analysis.
- **Example:**
  R
  library(naniar)
  gg_miss_var(data)

3.10. Amelia

The Amelia package in R is a robust tool for multiple imputation of missing data. It is particularly valuable for real-world evidence (RWE), where missing data is a persistent challenge due to the observational nature of datasets.

- **Purpose:** Focuses on multiple imputation of missing data using an expectation-maximization algorithm with bootstrapping.
- **Key Features:**

Efficient for time-series and cross-sectional data.
Generates multiple datasets for robust inference.

- **Example:**
  ```
  R
  library(Amelia)
  imputed_data <- amelia(data, m = 5)
  ```

3.11. Choosing the right tool

The selection of an R package depends on the nature of the missing data and the study's objectives:

- For multiple imputation, consider mice or Amelia.
- For datasets with non-linear relationships, missForest is a strong choice.
- Use VIM or naniar for visualization and pattern analysis.
- For simple imputation and descriptive summaries, Hmisc provides a quick solution.

By leveraging these tools, researchers can systematically explore and address missingness, ensuring more reliable and transparent outcomes in RWE studies.

4. Best Practices for Handling Missing Data

Adopting best practices for addressing missing data is crucial to ensuring the validity and transparency of real-world evidence (RWE) studies. These practices guide researchers in systematically managing missingness, minimizing biases, and enhancing the credibility of their findings. Below are key recommendations:

4.1. Assess patterns of missingness

Understanding the extent and nature of missing data is the first step in addressing it effectively.

- **Why It’s Important:** Patterns of missingness provide insights into the potential mechanisms (MCAR, MAR, or MNAR) and guide the selection of appropriate handling methods.
- **How to Implement:**
  Use visualization tools like VIM and naniar to explore missing data patterns visually.
  Generate heatmaps, aggregation plots, and scatterplots to identify relationships between missingness and observed variables.
- Example:
  ```
  R
  library (VIM) aggr(data,col=c("navyblue","red"),numbers=TRUE)
  library(naniar) gg_miss_var(data)
  ```

4.2. Document assumptions

Clearly state and justify the assumptions about the missing data mechanism.

- **Why It’s Important:** Transparency about assumptions (e.g., MAR or MNAR) ensures that the methods used align with the characteristics of the data and helps reviewers and readers evaluate the robustness of the results.
- **How to Implement:**

Include a section in study documentation or publications explicitly describing the hypothesized missing data mechanism.
Support assumptions with statistical tests and domain knowledge.

4.3. Combine methods

Using a combination of methods often yields more robust results.

- **Why It's Important:** No single method addresses all aspects of missing data. Combining techniques, such as multiple imputation and sensitivity analysis, ensures a comprehensive approach.
- **How to Implement:**
  Perform multiple imputation to handle MAR data and follow up with sensitivity analysis to test robustness under MNAR scenarios.
  Use advanced machine learning tools like missForest alongside traditional imputation techniques to improve accuracy.
- Example Workflow:
  Use mice for multiple imputations:
  R
  ```
  library(mice)
  imputed_data <- mice (data, method = "pmm", m = 5)
  ```
- Perform sensitivity analysis to test assumptions:
  Adjust parameters and compare results under different scenarios.

4.4. Report handling in publications

Transparency in reporting missing data handling is essential for reproducibility and credibility.

- **Why It's Important:** Readers and stakeholders must understand how missingness was addressed to assess the validity of the findings.
- **How to Implement:**
  The extent of missing data.
  Assumptions about the mechanism.
  Methods used for handling and the rationale behind their selection.
  Sensitivity analyses and their implications.
- **Example in a manuscript:**
  "Missing data accounted for 12% of the dataset. Data were assumed to be MAR based on an analysis of missingness patterns and domain knowledge. Multiple imputation using chained equations (mice package in R) was applied, followed by sensitivity analysis to assess the robustness of findings under MNAR scenarios."

4.5. Managing stakeholders

Stakeholder involvement is critical in shaping best practices for addressing missing data in real-world evidence (RWE) studies, as it ensures that solutions are aligned with practical and regulatory needs. Clinicians provide insights into the clinical relevance of variables and plausible ranges for imputations, helping to ground the data-handling process in real-world medical contexts. Policymakers and regulators contribute by defining transparency, reproducibility, and compliance standards, ensuring that missing data methodologies meet evidentiary requirements for decision-making. Collaborating with stakeholders fosters consensus on acceptable methods, supports domain-specific adaptations, and enhances the credibility and applicability of RWE findings in healthcare policy and practice [32, 33].

4.6. Enhancing reproducibility in missing data handling

To improve the reproducibility of missing data handling in RWE studies, it is essential to document and standardize every step of the process. Using transparent pipelines with well-documented code in languages like R or Python ensures that analyses can be easily replicated. Leveraging version-controlled environments (e.g., Git) helps track changes to imputation methods or datasets over time. Sharing detailed imputation protocols, including assumptions, diagnostics, and rationale for chosen methods, supports reproducibility across studies. Finally, adopting reproducible research tools like R Markdown or Jupyter Notebooks facilitates seamless sharing of workflows, from data preparation to imputation and final analysis [34, 35].

5. Conclusion

Handling missing data is a pivotal aspect of real-world evidence (RWE) studies, directly influencing the validity and reliability of findings. Missingness, if not addressed systematically, can introduce bias, undermine statistical power, and compromise study conclusions. This paper provided a comprehensive exploration of the challenges and solutions related to missing data, focusing on understanding mechanisms, applying appropriate methods, leveraging tools, and adhering to best practices.

5.1. Key takeaways

- **Understanding missing data mechanisms**
  Recognizing the nature of missingness—MCAR, MAR, or MNAR—is foundational. It informs the selection of methods for handling missing data, ensuring that the chosen approach aligns with the data's characteristics.
- **Methods for handling missing data**
  A range of methods exists, from simple techniques like complete case analysis and single imputation to advanced approaches like multiple imputation, maximum likelihood estimation, and machine learning models. Each method has its strengths and limitations, necessitating careful consideration based on the study’s context and goals.
- **Tools in R**
  R offers a robust suite of packages, such as mice, missForest, VIM, naniar, and Amelia, which provide researchers with powerful capabilities for exploring, visualizing, and imputing missing data. The flexibility and depth of these tools empower researchers to handle missingness effectively and efficiently.

5.2. Best practices

Adopting best practices, including assessing patterns of missingness, documenting assumptions, combining complementary methods, and transparent reporting strategies, enhances the credibility of RWE studies. These practices minimize bias and facilitate reproducibility, ensuring that findings are robust and reliable.

5.3. Final thoughts

Addressing missing data is a technical challenge and a methodological imperative in RWE studies. By combining a thorough understanding of missingness mechanisms, selecting and applying appropriate methods, utilizing advanced tools, and adhering to best practices, researchers can produce high-quality, transparent, and credible results. These efforts ultimately contribute to better-informed healthcare decisions, fostering confidence in the outcomes derived from RWE studies.

Future advancements, such as the integration of machine learning and Bayesian approaches, promise further improvements in handling missing data. However, the principles of systematic assessment, transparency, and robustness will remain the cornerstone of effective data management in RWE research.

References

1. Little RJ, Rubin DB. Statistical Analysis with Missing Data (3rdedn),Wiley, USA. 2019.

2. Sterne JA, White IR, Carlin JB, et al. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. BMJ. 2009;338.

3. Buuren SV. Flexible imputation of missing data (2ndedn), CRC press, Taylor & Francis Group, London, United Kingdon. 2018.

4. Enders CK. Applied missing data analysis (2ndedn), Guilford Publications, New York, USA. 2022.

5. Buuren SV, Groothuis-Oudshoorn K. mice: Multivariate imputation by chained equations in R. J Stat Softw. 2011;45:1-67.

6. Harrell FE. Regression modeling strategies. R package version. 2012.

7. Hosmer DW, Lemeshow S, Sturdivant RX. Applied logistic regression (3rdedn), John Wiley & Sons, New Jersy, Canada. 2013.

8. Kleinbaum DG, Klein M. Survival analysis a self-learning text (3rdedn), Springer, Berlin, Germany. 1996.

9. Hernan MA, Robins JM. Causal Inference: What If Chapman Hall/CRC, Boca Raton. 2020.

10. Casella G, Berger RL. Statistical Inference Duxbury Press. Pacific Grove, CA. 2002.

11. Pawitan Y. In all likelihood: statistical modelling and inference using likelihood. Oxford University Press, Oxford, New York, USA. 2001.

12. Wickham, H. Advanced R (2ndedn) CRC Press, New York, USA. 2019.

13. Venables WN, Ripley BD. Modern Applied Statistics with S, Springer, Berlin, Germany. 2002.

14. Sherman RE, Anderson SA, Dal Pan GJ, et al. Real-world evidence—what is it and what can it tell us. N Engl J Med. 2016;375:2293-7.

15. Makady A. Real-world evidence for coverage decisions: Opportunities and challenges. Front Pharmacol. 2017;8:297.

16. Rosenbaum PR, Rubin DB. The central role of the propensity score in observational studies for causal effects. Biometrika. 1983;70:41-55.

17. Austin PC, Stuart EA. Moving towards best practice when using inverse probability of treatment weighting (IPTW) using the propensity score to estimate causal treatment effects in observational studies. Stat Med. 2015;34:3661-79.

18. FDA framework for real-world evidence: provides insights into using RWE in regulatory decision-making. 2018.

19. VanderWeele TJ. Explanation in causal inference: methods for mediation and interaction, Oxford University Press, Oxford, New York, USA. 2015.

20. Rubin DB. Causal inference using potential outcomes: Design, modeling, decisions. J Am Stat Assoc. 2005;100:322-31.

21. Kuhn M, Johnson K. Applied Predictive Modeling, Springer, Berlin, Germany. 2013.

22. Geron A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow (3rdedn), Aurelien Geron. 2022.

23. Chen R, Snyder M. Real-world evidence: How machine learning can transform clinical trials. Nat Rev Drug Discov. 2019;18:749-50.

24. Gelman A. Bayesian Data Analysis, 3rd: Boca Raton. Stat Sci. 2013.

25. Spiegelhalter DJ, Abrams KR, Myles JP. Bayesian approaches to clinical trials and health-care evaluation, John Wiley & Sons, England, United Kingdom. 2004.

26. Berry DA. Bayesian clinical trials. Nat Rev Drug Discov. 2006;5:27-36.

27. Parmar MKB. Bayesian adaptive randomized trials: From concept to reality. Nat Rev Clin Oncol. 2016;13:377–85.

28. Little RJ. A test of missing completely at random for multivariate data with missing values. J Am Stat Assoc. 1988;83:1198-202.

29. Pisică D, Dammers R, Boersma E, et al. Tenets of good practice in regression analysis. a brief tutorial. World Neurosurg. 2022;161:230-9.

30. Van Buuren S, Groothuis-Oudshoorn K. mice: Multivariate imputation by chained equations in R. J Stat Softw. 2011;45:1-67.

31. Stekhoven DJ. missForest: Nonparametric missing value imputation using random forest. Astrophys Source Code Lib. 2015:1505.

32. https://cran.r-project.org/web/packages/Hmisc/index.html

33. https://statistikat.github.io/VIM/#:~:text=Imputation%20functions%20such%20as%20kNN,highlighting%20missing%20and%20imputed%20values

34. https://cran.r-project.org/web/packages/naniar/vignettes/getting-started-w-naniar.html

35. https://cran.r-project.org/web/packages/Amelia/index.html