Handling Missing Data: Methods, Impact, and Practical Applications

1. Title & Research Question

Research Question: How do different missing data handling methods — including deletion, simple imputation, and model-based techniques — affect the accuracy and reliability of downstream analyses?

Relevance:
Missing data is an inevitable challenge in real-world datasets, arising from various factors such as non-responses in surveys, sensor failures, or data corruption. Improper handling of missing values can lead to biased estimates, reduced statistical power, and inaccurate predictions. This chapter explores the theoretical foundations of missing data mechanisms and compares practical imputation techniques to understand their impact on data analysis outcomes.

---

2. Theory and Background

When working with real-world data, missing values are almost guaranteed. Maybe a respondent skipped a survey question, a sensor stopped recording halfway, or some data got lost during collection. Regardless of the cause, how we deal with these gaps can dramatically change the results of any analysis. To handle them well, it's important to first understand *why* data goes missing.

Types of Missing Data

1. Missing Completely at Random (MCAR)
This is the ideal scenario. The fact that data is missing has nothing to do with the data itself — observed or unobserved. For example, if some survey forms got accidentally shredded, the missingness is unrelated to any characteristic of the respondents. Analyses performed on MCAR data remain unbiased if you simply remove the missing rows, although you lose some statistical power.

2. Missing at Random (MAR)
Here, the missingness depends on *observed* data but not on the *missing* values themselves. For example, younger respondents might be more likely to skip questions about income. If age is recorded, we can use it to model and handle the missing income data. This is the most common and practically manageable case.

3. Missing Not at Random (MNAR)
This is the trickiest. The missingness depends on the *unobserved* data itself. For example, people with very high or very low incomes might be less willing to report their income. In these situations, ignoring the missingness or using simple methods can produce biased results. MNAR often requires more advanced modeling or strong assumptions.

Why Naive Deletion Doesn't Always Work

A common quick fix is to drop rows or columns with missing data. While this works fine under MCAR, it can introduce serious bias under MAR or MNAR. Imagine deleting all rows where income is missing, and those missing values mostly come from younger participants. You'd end up analyzing a dataset that doesn't represent the real population anymore.

A Quick Look at Approaches

Researchers have developed a wide range of methods to tackle missing data: from simple tricks like filling with the mean, to sophisticated model-based imputations that use relationships between variables. The right method depends on the missingness mechanism, the proportion of missing values, and the goals of the analysis. In the following sections, we'll frame this as a clear problem, analyze constraints, and then walk through different imputation strategies with practical examples.

---

3. Problem Statement

To explore and compare different missing data handling methods, we will use the **Titanic dataset**, a well-known dataset from Kaggle that records information about passengers aboard the Titanic. This dataset contains both numerical and categorical features and has natural missingness in several columns, making it an excellent candidate for demonstrating various imputation strategies.

Input

- A dataset containing passenger information, including features such as `Age`, `Cabin`, `Embarked`, and `Fare`.
- The dataset includes missing values in multiple columns:
- **Age**: ~20% missing
- **Cabin**: ~77% missing
- **Embarked**: a few entries missing
- The target variable is `Survived` (0 = did not survive, 1 = survived).

Output

- A cleaned and imputed version of the dataset where missing values have been handled using different methods:
- Deletion (e.g., dropping rows/columns)
- Simple imputation (mean/median/mode)
- KNN or regression-based imputation
- Comparative evaluation of how each method affects downstream analysis — in this case, the accuracy of a classification model predicting survival.

Sample Input Table

PassengerId	Pclass	Sex	Age	Cabin	Embarked	Survived
1	3	male	22.0	NaN	S	0
2	1	female	38.0	C85	C	1
3	3	female	26.0	NaN	S	1
4	1	female	35.0	C123	S	1
5	3	male	NaN	NaN	S	0

Sample Output Table (Imputed with Median for Age)

PassengerId	Pclass	Sex	Age	Cabin	Embarked	Survived
1	3	male	22.0	NaN	S	0
2	1	female	38.0	C85	C	1
3	3	female	26.0	NaN	S	1
4	1	female	35.0	C123	S	1
5	3	male	29.7	NaN	S	0

This section sets the stage for analyzing how different imputation methods influence the resulting dataset and the predictive model's performance.

---

4. Problem Analysis

Before jumping into different methods to fix missing data, it’s important to analyze *what kind of missingness we’re dealing with* and the practical constraints that come with the Titanic dataset.

Patterns of Missingness

The Titanic dataset doesn’t have missing values randomly sprinkled everywhere — they appear in distinct patterns: - **Age** has about 20% missing entries. This isn’t entirely random. Age is more likely to be missing for third-class passengers and males, hinting that the missingness could depend on observed features like `Pclass` and `Sex`. This suggests a **Missing at Random (MAR)** mechanism. - **Cabin** is missing for nearly 77% of passengers. The sheer volume of missing entries makes direct imputation difficult and potentially misleading. Cabin information might also be systematically missing for lower-class passengers who didn’t have assigned cabins. This leans toward MNAR (Missing Not at Random). - **Embarked** has just two missing entries — these can be handled quite easily with simple imputation (e.g., using the mode) without affecting overall analysis.

Constraints and Challenges

1. **High Missingness in Cabin:** Since Cabin is missing for the majority of entries, imputing it meaningfully is tough. Dropping it might make more sense, or we could engineer a simpler binary feature like "Cabin known: Yes/No."
2. **Mixed Data Types:** Numerical (`Age`, `Fare`) vs. Categorical (`Sex`, `Embarked`) require different imputation strategies. For example, mean imputation doesn't apply to categorical variables.
3. **Downstream Effects:** Different imputation methods can shift distributions and impact model performance. For example, mean-imputing Age might distort the relationship between age and survival.
4. **Computational Practicality:** The dataset is relatively small, so even model-based imputations like KNN or regression are feasible without performance issues.

Logical Approach

- **Step 1:** Explore the missingness visually (e.g., null heatmaps, group comparisons) to confirm MAR/MNAR patterns.
- **Step 2:** Handle easy cases first — Embarked via mode imputation, Cabin via dropping or simplifying.
- **Step 3:** Focus on Age imputation using different methods:
- Mean/Median for simplicity.
- KNN imputation to use neighboring passengers' attributes.
- Regression-based imputation using Pclass, Sex, SibSp, Parch, and Fare.
- **Step 4:** Compare the effects of each method on downstream survival prediction accuracy using a classification model.

By understanding these patterns and constraints upfront, we can make smarter choices about which imputation method to apply, rather than blindly filling in blanks.

---

5. Solution Explanation

There's no single "best" way to handle missing data — the right choice depends on the data, the missingness mechanism, and the goal of the analysis. Here, we'll walk through different methods applied to the Titanic dataset, explaining the logic behind each one.

1. Deletion Methods

a. Row Deletion
The simplest method is to remove any rows with missing values. For example: - Remove passengers with missing `Age`, `Embarked`, or `Cabin`.

```
For each row in dataset:
    If any required feature is missing:
        Drop the row
```

✅Pros: Easy to implement; preserves integrity of observed values.
❌Cons: Can reduce dataset size and introduce bias if missingness isn't MCAR. In Titanic, deleting all Cabin-missing rows would eliminate most third-class passengers.

b. Column Deletion
If a feature has too many missing values (like `Cabin` with 77% missing), dropping the entire column can sometimes be justified.

✅Pros: Useful when a feature isn't critical.
❌Cons: Risk of losing potentially valuable information.

---

2. Simple Imputation

Simple imputation involves replacing missing values with a single statistic like the mean, median, or mode.

a. Mean/Median Imputation (Numerical)
For `Age`, we can replace missing values with the median age of observed passengers.

```
median_age = median(Age where Age is not missing)
For each row:
    If Age is missing:
        Age = median_age
```

Median is preferred over mean because Age is right-skewed — median is more robust to outliers.

b. Mode Imputation (Categorical)
For `Embarked`, which has only two missing values, we can replace missing entries with the most common embarkation port.

```
mode_embarked = mode(Embarked)
For each row:
    If Embarked is missing:
        Embarked = mode_embarked
```

✅Pros: Very easy and fast.
❌Cons: Doesn't capture variability; can distort distributions if too many values are missing.

---

3. K-Nearest Neighbors (KNN) Imputation

KNN imputation uses the values of the *k* most similar rows to fill in missing data. For missing `Age`, for example: - Select k nearest neighbors based on other features like Pclass, Sex, Fare, SibSp, Parch. - Impute Age as the average Age of those neighbors.

```
For each row with missing Age:
    Find k nearest rows (by Euclidean distance on selected features)
    Age = average(Age of neighbors)
```

✅Pros: More accurate than simple imputation when related features exist. Captures patterns in the data.
❌Cons: Computationally more expensive. Sensitive to how distance is defined and scaled.

---

4. Regression-Based Imputation

Regression imputation builds a predictive model to estimate missing values. For `Age`, we can train a regression model using passengers with known ages:

Features: Pclass, Sex (encoded), SibSp, Parch, Fare, Embarked
Target: Age

Steps: 1. Train a regression model on rows where Age is not missing. 2. Predict Age for rows where it is missing.

```
model = train_regression(features, Age where Age is not missing)
For each row with missing Age:
    Age = model.predict(features)
```

✅Pros: Leverages relationships between variables; generally more accurate.
❌Cons: Assumes model is correct; can underestimate variance (since it predicts single values).

---

5. Multiple Imputation (Optional Advanced)

Multiple imputation involves creating multiple plausible imputations and combining results to account for uncertainty. While more advanced, this can be explored using libraries like `miceforest` or `statsmodels`.

Choosing the Right Approach

For the Titanic dataset, a **hybrid strategy** makes sense: - Drop `Cabin` or convert it to a binary indicator (Cabin known: Yes/No). - Mode-impute `Embarked` (only two missing). - Compare `Age` imputations using: - Median (baseline) - KNN (structure-aware) - Regression (model-based)

We will then compare how these methods affect model performance when predicting survival.

---

6. Results and Data Analysis

(Present code snippets, tables, or visualizations from the Jupyter Notebook. Compare methods in terms of performance metrics, distribution preservation, and impact on downstream analysis. Provide insightful commentary.)

---

7. References

(Include relevant academic papers, books, and software documentation. Ensure correct citation format.)
