Missing Data

Why Are Missing Values a Problem?

For Predictors:

- Many models cannot handle missing values.
- Many feature engineering techniques cannot handle missing values.
- Model performance can degrade.

For Target Variables:

- Almost all models cannot handle missing values.

How Do Missing Values Occur?

Structural Absence

- Value is not applicable

Measurement and Collection Failures

- Random: battery died, connection lost
- Systematic: respondent skipped a survey question

Data Integration Issues

- Merging datasets: keys don't match → missing values
- Column exists only in one source

Structural Absence

Example: *Ames Housing Data*, column `Alley`

Question:
Why is the column missing?
What can it be replaced with?

Strategy for structural absence

Keep the information on missing values! You can

- Create a new binary feature:

```
df["Alley_missing"] = df["Alley"].isna().astype(int)
```

and optionally

- Impute the original Alley values.

Here, to **impute** means to fill in missing data with estimated values based on the available information.

Randomly Missing Values

Three types:

- **MCAR**: Missing Completely At Random
- **MAR**: Missing At Random
- **NMAR** or **MNAR**: Not Missing At Random

Example: Income and Age

Two variables: **Income** and **Age**
Missing values: in **Income**

- **MCAR**: Missingness is independent of both income and age.
- **MAR**: Missingness depends on age. (Older individuals report income less often.)
- **NMAR**: Missingness depends on income. (Higher earners less likely to report.)

Example: Weighing Objects on Surfaces

Using a scale on various surfaces:

- **MCAR**: Values are missing randomly regardless of object or surface.
- **MAR**: More missing values on soft surfaces.
- **NMAR**: Missingness increases with weight. (E.g., heavier objects appear later, battery drains.)

Formula Representation of Missingness

Let R be the mask for missing values

For example:

```
X =       Age    Income    Gender
        [ 45      60k        M    ]
        [ 32      NaN        F    ]
        [ 38      55k        M    ]
        [ 50      NaN        F    ]

R =      [ 1        1        1    ]
         [ 1        0        1    ]
         [ 1        1        1    ]
         [ 1        0        1    ]
```

Formula Representation of Missingness

Let $X = (X_{\text{obs}}, X_{\text{mis}})$ be observed and missing parts
Let ϕ be model parameters

- **MCAR**: $P(R = 0|X_{\text{obs}}, X_{\text{mis}}, \phi) = P(R = 0|\phi)$
- **MAR**: $P(R = 0|X_{\text{obs}}, X_{\text{mis}}, \phi) = P(R = 0|\phi, X_{\text{obs}})$
- **NMAR**: $P(R = 0|X_{\text{obs}}, X_{\text{mis}}, \phi)$ depends on both X_{obs} and X_{mis}

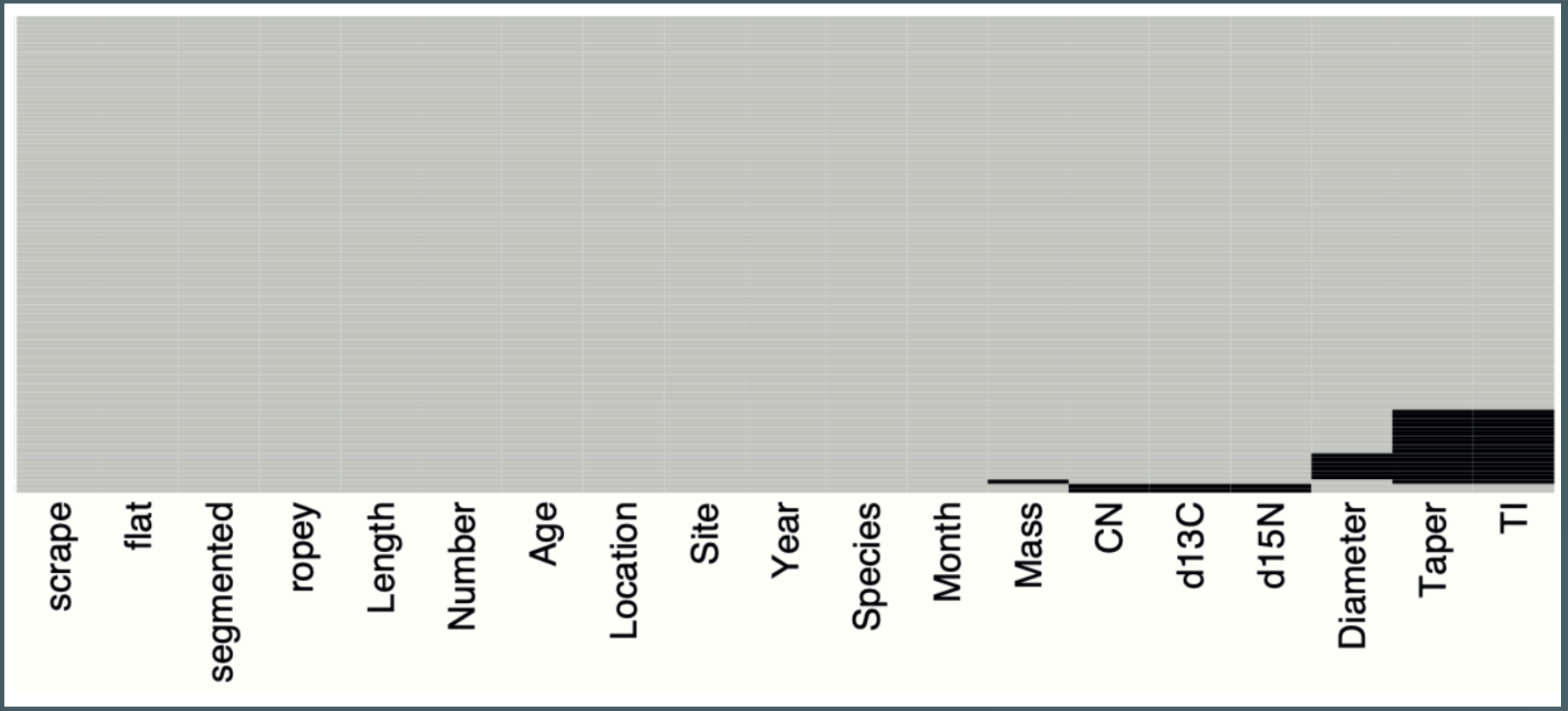
Practical Implications

- **MCAR**: You can delete samples without bias; many imputation methods apply.
- **MAR**: Imputation methods work.
- **NMAR**: Imputation can introduce bias or distort results.

Visualizing Missing Values

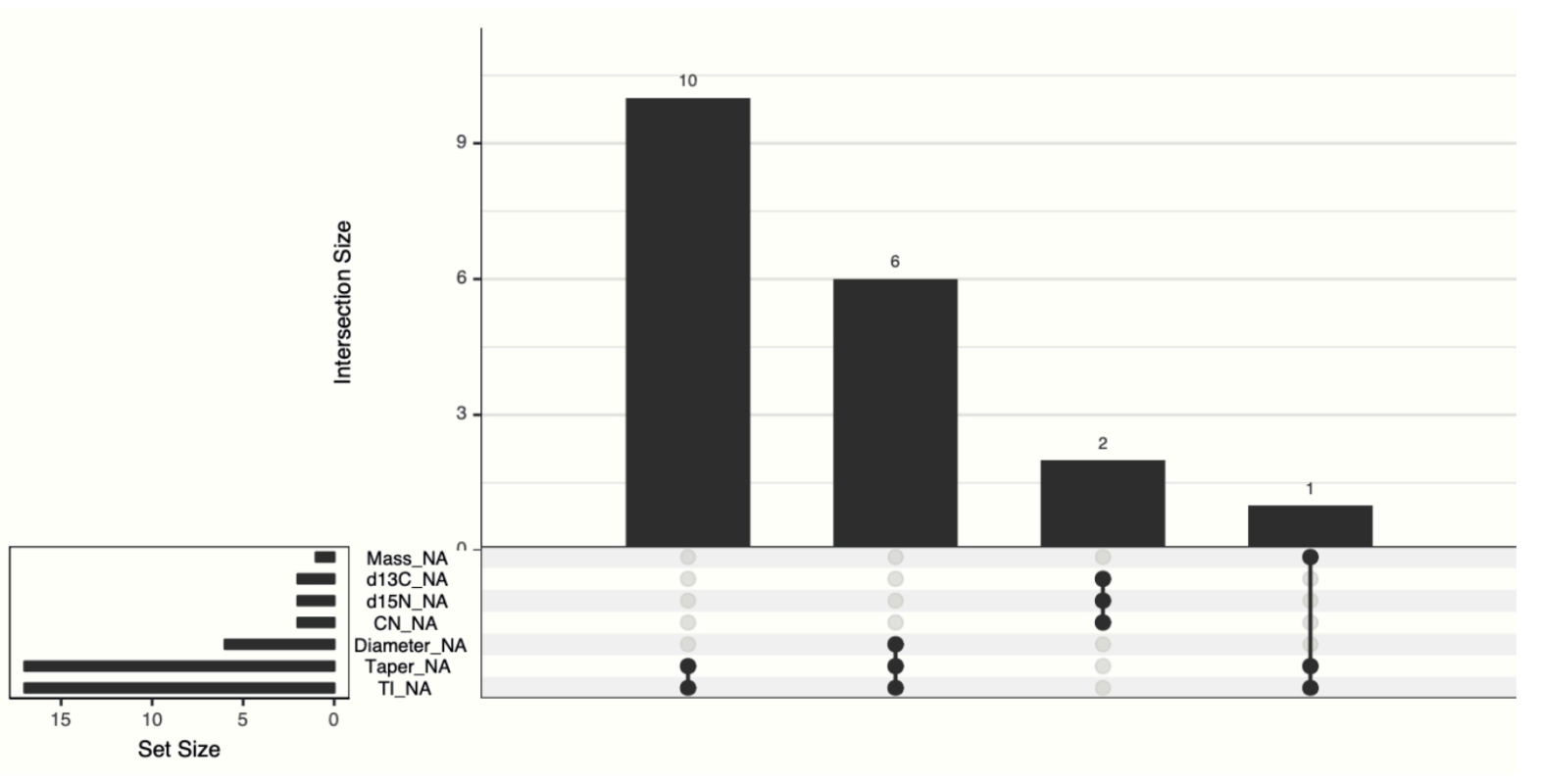
A first step in deciding on a strategy is visualizing the missing values.

Heatmap



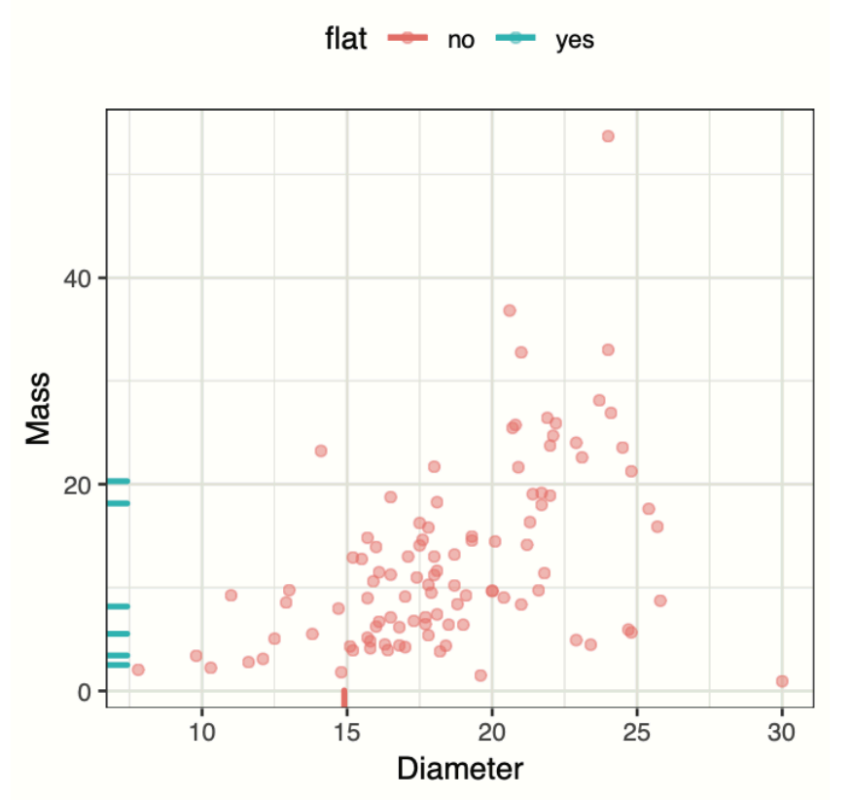
Visualizing Missing Values

Co-occurrence Plot



Visualizing Missing Values

Scatterplot



Summary Tables for Large Datasets

Create summaries showing:

- Proportion of missing values per predictor
- Proportion of missing values per sample

Task: Compute and display these for the `scat` dataset.

Strategy: Do Not Impute

- Some models have built-in handling of missing values.
- Acts as a baseline strategy.

Strategy: Delete Missing

Two approaches:

- Delete samples (rows)
- Delete predictors (columns)

Rules of Thumb:

- Samples are usually more valuable than predictors.
- Deleting under MCAR is unbiased.
- Deleting under NMAR can introduce bias.

Delete Example: Chicago Train Ridership

Each sample (day) has missing values.
Most missing values are concentrated in certain stations.

Question: What should we delete?

Imputation for Statistical Inference

Historically, missing data methods focused on **preserving statistical validity**, not prediction accuracy.

Goals of Inferential Imputation

- Ensure test statistics (e.g. p-values, confidence intervals) are valid
- Maintain tractable and interpretable statistical distributions
- Enable **hypothesis testing** despite incomplete data

Imputation for Statistical Inference

Multiple Imputation

1. Create multiple versions of the dataset, each with **different estimates** for the missing values
2. Compute test statistics for **each** imputed dataset
3. Combine the results (e.g., average coefficients, pool standard errors)

Helps reflect **uncertainty** due to missing data in downstream statistical inference.

Imputation for Predictive Modeling

Imputation in predictive contexts serves a different goal:
Accurate estimation of missing values to maximize future model performance

Key Differences from Inference

- Most models don't assume a specific distribution. Thus multiple imputation is not relevant.
- To still capture the variation of imputation, imputation should be performed *within resampling*
- Repeated imputation would greatly increase computation time
- Objective: **predict missing values as closely as possible**, not estimate distribution

Imputation for Predictive Modeling

Requirements for Predictive Imputation

- **Tolerant of other missing variables** in the row
- Produces **compact**, efficient prediction equations, since we are stacking models
- Handles **mixed types** (numeric + categorical)
- **Stable** under outliers or noise

Bottom Line

> Predictive imputation is about **accuracy**, not statistical validity.
> It must generalize well — including to future unseen test data.

Imputation Strategy: Impute Most Common

Replace missing values with:

- **Categorical**: Most frequent value
- **Numerical**: Global mean

Task: Impute missing values in the Chicago Train dataset with global mean. Visualize via heatmap.

Strategy: Impute by Group (Concept Most Common)

Impute using group-level statistics:

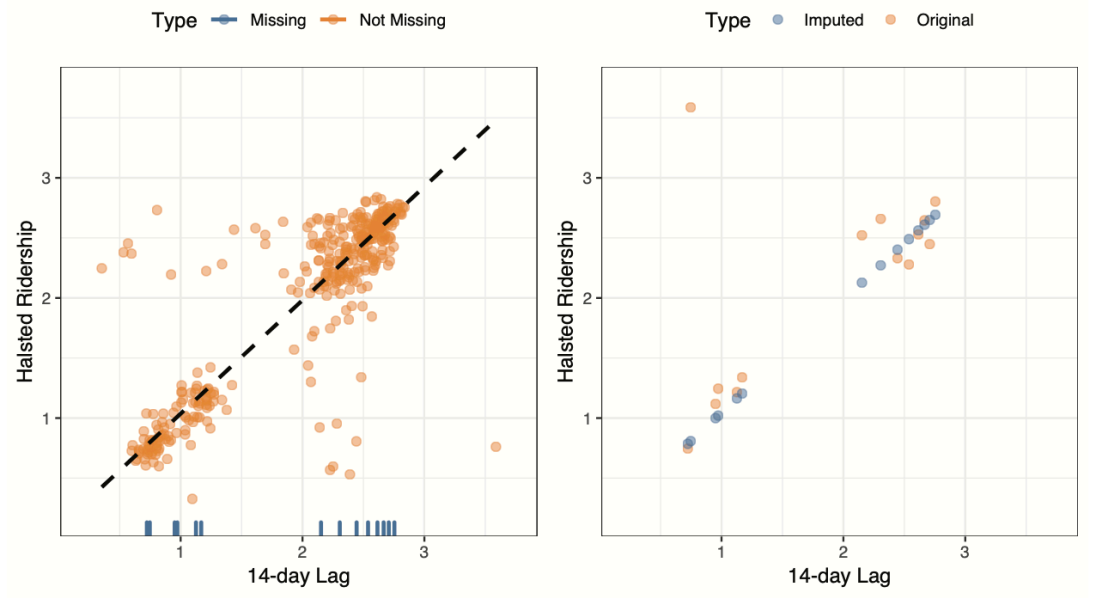
- Use other predictors to form groups
- Replace with group-wise mean or mode

Task: Impute using station-wise mean for `'87th (Red Line)'`. Visualize via heatmap.

Strategy: Impute using linear model

Use a linear model to impute missing values.

Example: 14 or 7 day lag in the Chicago train data



Strategy: k-Nearest Neighbors Imputation

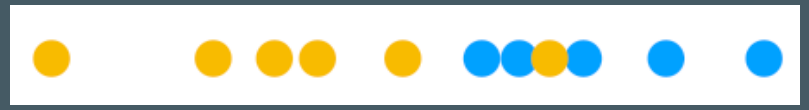
Find `k` closest complete samples.
Impute using average of neighbors.

Task: Impute using `k=2` for `'87th (Red Line)'`. Visualize via heatmap.

Expectation Maximization (EM) for Imputation

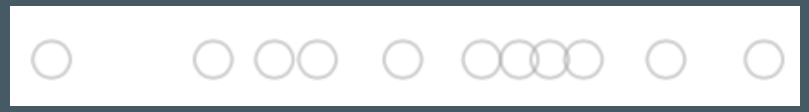
Soft clustering approach with unknown parameters and clusters.

If clusters are known, estimating (μ, σ) is no problem:



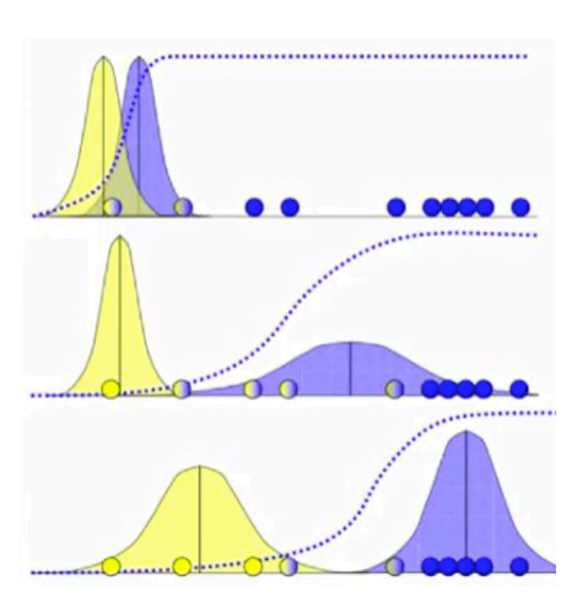
Expectation Maximization (EM) for Imputation

If parameters (μ, σ) are known, filling in the clusters is easy.



EM: Algorithm Steps

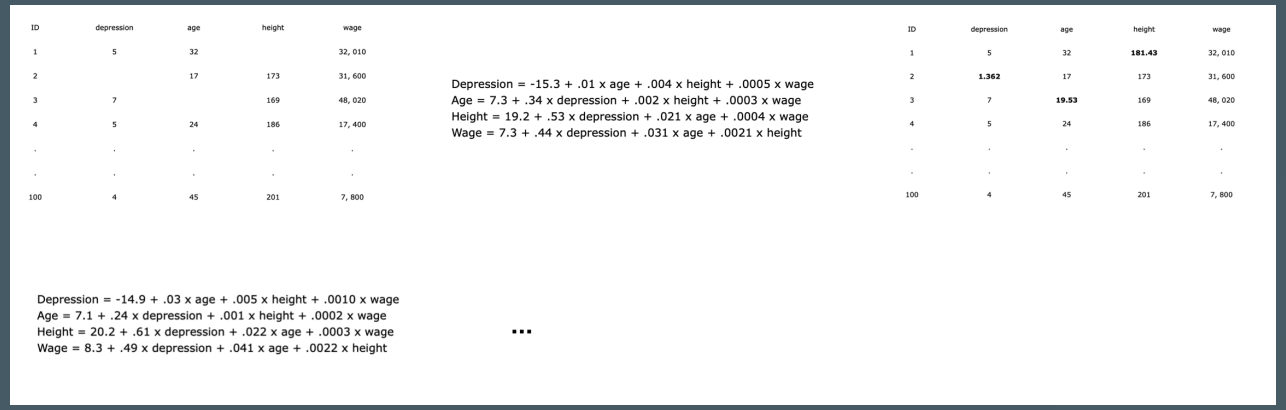
1. Start with initial guesses for μ and σ.
2. Compute probabilities (responsibilities) of cluster membership.
3. Update parameters using weighted means and variances.
4. Repeat until convergence.



EM for Missing Values

Assume a distribution `f_θ(X_obs, X_mis)`

- Treat `X_mis` as latent variables.
- Seek a Maximum Likelihood Estimate for θ.
- Alternate between estimating `X_mis` and updating θ.



Example: Iterative Imputation Process

Features: A, B, C (all with missing values)

1st Iteration:

- Estimate A from B and C
- Estimate B from A (imputed) and C
- Estimate C from A and B

2nd Iteration:

- Repeat with updated estimates

Strategy: Iterative Impute using Bayesian ridge regression

Bayesian Ridge Regression

A probabilistic version of linear regression:

- Estimates a **distribution over coefficients**, not just point values
- Provides **uncertainty** (variance) for each prediction

Strategy: Iterative Impute using Bayesian ridge regression

Multivariate imputation using other predictors, like Bayesian ridge regression
Iterate over columns until convergence like in EM algorithm

Task: Use iterative imputer (max_iter=2). Visualize for `'87th (Red Line)'`.