Female Farmers Health Study: Data Cleaning & Preprocessing Pipeline

Project Overview

This document details the comprehensive data preparation pipeline implemented for the female farmers health study. The dataset contains information on 81 female agricultural workers, primarily from the Monastir region of Tunisia, with 61 variables covering demographics, work conditions, health metrics, and protection practices.

Pipeline Summary

The data preparation followed these sequential steps:

1. **Data Loading & Standardization**: Missing values standardization
2. **Encoding & Preprocessing**: Conversion of categorical variables to numerical formats
3. **Missing Data Analysis**: Determination of missingness patterns and mechanisms
4. **Imputation Strategy**: Selection of appropriate imputation techniques
5. **Imputation Execution**: Implementation of imputation methods
6. **Imputation Validation**: Verification of imputation quality
7. **Dataset Decoding**: Conversion back to human-readable formats

Each step was implemented with careful consideration of data integrity, statistical validity, and domain knowledge preservation.

Step 1: Data Standardization and Missing Data Analysis

Why This Step Is Important

Before proceeding with any advanced analysis or imputation, it was critical to understand:

- How much data is missing, and
- Why it is missing.

Unstandardized missing value entries (e.g., 'non sp cifi ', '#VALUE!', '-') can obscure the true extent of missing data. Without cleaning and visualizing this properly, any downstream analysis could become biased or unreliable.

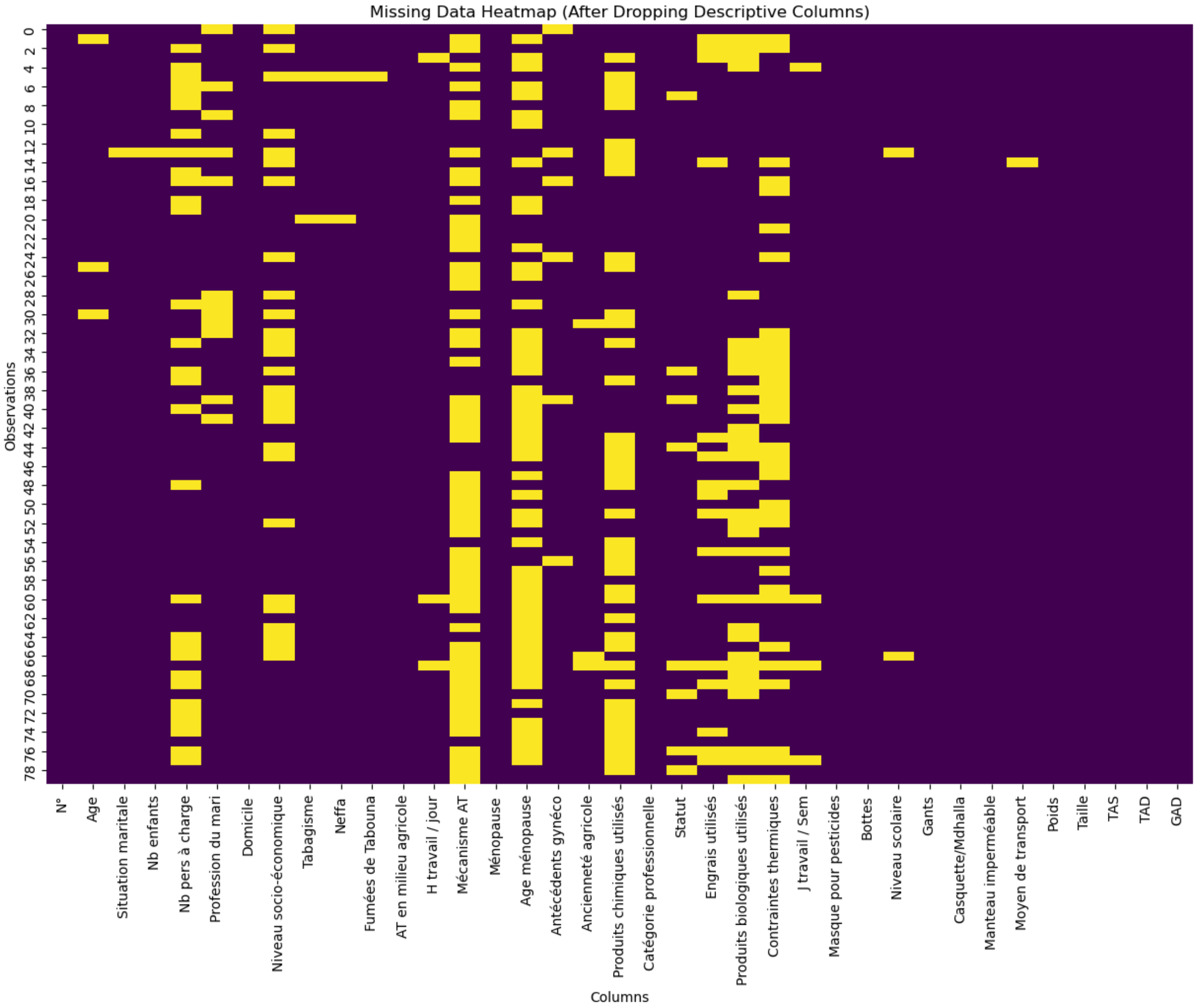
How We Approached It

We followed a three-phase methodology:

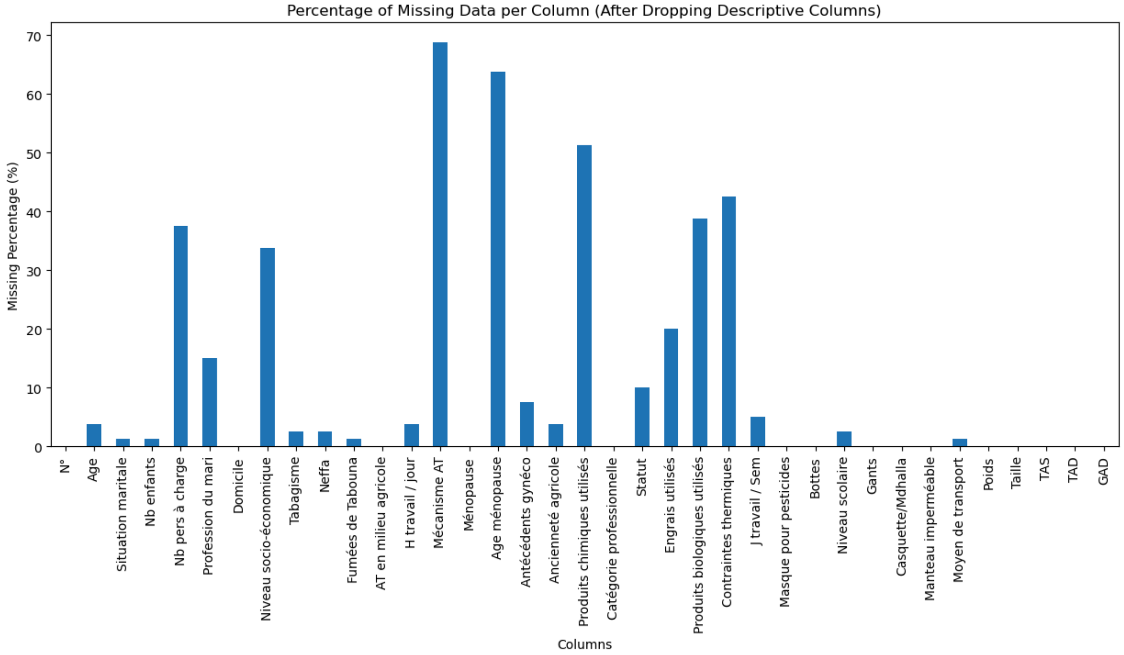
1. **Standardization**: Replaced all inconsistent representations of missing values with NaN using pandas.
2. **Visualization**: Generated heatmaps and bar charts to identify missing patterns.
3. **Mechanism Analysis**: Determined whether missingness was random or conditionally based on participant groups.

Key Findings

The missing data visualization revealed significant patterns:



Missing Value Percentages:



Key insights:

- Some variables (e.g., menopausal age, gynecological history) have over 60% missing values.
- Core health metrics (e.g., weight, height, blood pressure) are nearly complete.
- Missingness follows structured patterns, not randomness - suggesting condition-specific data collection.
- Removing descriptive text fields clarified the analysis focus.

Step 2: Data Encoding and Preprocessing

Why This Step is Necessary

After cleaning and standardizing missing values in Step 1, the dataset was still unsuitable for direct use in models. Most algorithms (especially in scikit-learn or statsmodels) require purely numerical inputs.

Additionally, our dataset had mixed types:

- Text (e.g., profession, tasks)
- Categories with order (e.g., education level)

- Multi-choice lists (e.g., types of fertilizers)
- Yes/No fields

Encoding these variables transforms human-friendly input into formats a model can understand, while retaining as much semantic meaning as possible. Without this step, we cannot proceed to correlation analysis, regression, imputation, or fairness modeling.

How We Encoded the Data

We used domain-aware encoding strategies, guided by the codebook:

Encoding Type	Strategy
Binary (oui/non)	'oui' → 1.0, 'non' → 0.0, NaN → missing
Tabagisme (3-level)	'non' → 0.0, 'passif' → 1.0, 'oui' → 2.0
Ordinal Equipment Use	'jamais' → 0.0, ..., 'toujours' → 3.0
Socioeconomic & Categorical	Mapped ordinally (e.g., bas → 0.0, bon → 2.0)
Multi-value Indicators	Split into multiple binary flags (e.g., pesticides → 1.0 if present)
One-hot Profession	Each profession gets its own column (1.0 = active, 0.0 = not)

Justification for Encoding Choices

The encoding strategy was chosen based on the statistical meaning and modeling implications of each variable type.

Husband's profession was encoded using one-hot encoding because:

- There is no inherent order among job types (e.g., 'agriculteur' vs. 'maçon')
- Treating them ordinally would create misleading relationships
- One-hot encoding allows the model to treat each job independently and avoids introducing artificial hierarchy
- The category count (28 jobs + 1 missing indicator) is moderate and manageable in modern ML pipelines

In contrast, ordinal encoding was applied where a natural hierarchy exists:

- Education levels: 'analphabète' < 'primaire' < 'secondaire' < 'supérieur'
- Equipment usage: 'jamais' < 'parfois' < 'souvent' < 'toujours'
- Socioeconomic and marital status also follow a semantically ranked structure

By using these strategies:

- We ensure numerical encodings reflect the real-world semantics
- We avoid model bias from incorrect assumptions about category importance
- We maintain interpretability, especially in fairness-aware and regression models

Output of This Step

After encoding:

- 74 clean columns remain (fully numerical)
- All missing values follow strict rules for interpretation
- No text fields remain
- This output is now ready for modeling steps such as:
  - Correlation analysis
  - PCA (Principal Component Analysis)
  - Regression modeling
  - Fairness or privacy-aware learning

Step 3: Missing Data Analysis & Imputation Strategy

Objective

Analyze patterns of missingness to determine appropriate imputation strategies for each variable.

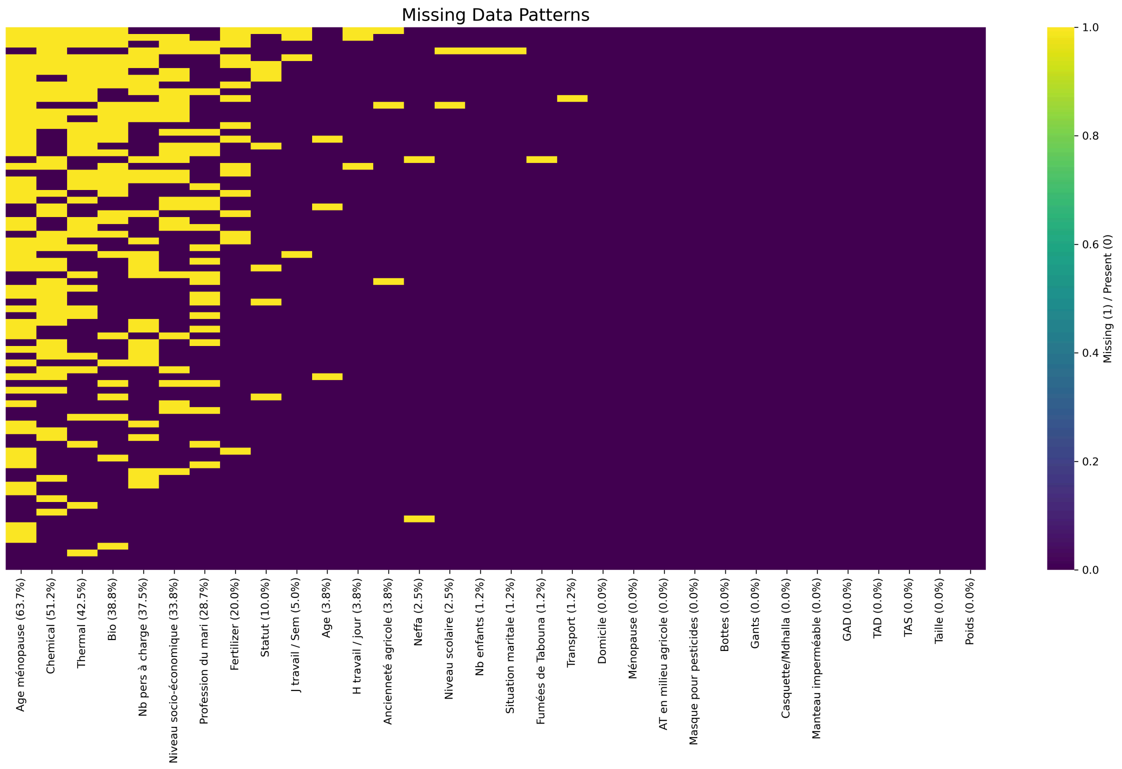
Methodology

We analyzed missingness for all features using percentage heatmaps and manually evaluated their mechanisms:

- **MCAR (Missing Completely At Random)**: Safe to impute with simple strategies like mode/mean
- **MAR (Missing at Random)**: Requires context-aware or model-based imputation
- **MNAR (Not Missing at Random)**: Either dropped or conditionally imputed

We created a recommendation table covering:

- Missing % per column
- Mechanism classification
- Tailored imputation strategy



Recommended Imputation Strategies

Based on the analysis, we created a structured recommendation table with:

- Percentage of missing values per column
- Assumed missingness mechanism

- Tailored imputation strategy:
    - Conditional imputation for demographic-specific fields (e.g., 'Age ménopause')
    - MICE or PMM for moderate MAR features
    - Mode or mean imputation for MCAR fields
    - Dropping or auxiliary use for untrustworthy fields

Advanced Imputation Techniques: MICE and PMM

For handling MAR (Missing At Random) data, we selected:

1. **MICE (Multiple Imputation by Chained Equations)**:
    - Builds multiple regression models, one for each variable with missing data
    - Uses other variables as predictors
    - Runs in cycles, updating estimates until convergence
    - Captures inter-variable relationships
    - Reflects uncertainty through multiple simulations
    - Avoids oversimplifying complex dependencies
2. **PMM (Predictive Mean Matching)**:
    - Often used within MICE
    - Predicts values using regression, then finds closest observed value to impute
    - Keeps imputed values realistic by grounding them in real data
    - Especially useful for skewed or bounded distributions

These methods are particularly suited for MAR because they exploit observed data relationships to make statistically sound imputations.

Step 4: Imputation Methods - Expanded Explanation

Detailed Imputation Methodology

The imputation phase used a tailored combination of statistical and model-based strategies depending on the nature and structure of each variable's missingness. These decisions were rooted in our understanding of the data and confirmed visually and statistically using diagnostic clustering and similarity validation.

1. **Mode Imputation**:
   - Applied to binary or low-cardinality MCAR fields like 'Thermal', 'Bio', and 'Chemical' exposures.
   - These features lacked correlation with other predictors, making advanced methods unnecessary.
   - The mode (most frequent value) preserves distributional integrity without overfitting, making it suitable for MCAR.
2. **Conditional Imputation**:
   - Example: 'Age menopause' was imputed only for rows where 'Menopause' == 'oui'.
   - This prevents injecting values in biologically irrelevant cases (e.g., non-menopausal women).
   - Implemented through logical filtering and group-wise imputation in pandas.
3. **KNN Imputation (using dendrograms to choose 'k')**:
   - For complex MAR cases, particularly with structured continuous or ordinal data (e.g., 'Nb pers a charge', 'J travail / Sem').
   - We used hierarchical clustering dendrograms to estimate the appropriate number of neighbors (k).
   - The chosen k was informed by natural variable groupings based on correlation and cosine distances.
   - Implemented with KNNImputer from sklearn.impute after scaling and ensuring no text fields remained.
4. **MICE (Multiple Imputation by Chained Equations)**:
   - Used where variables were numerically important but had moderate MAR (e.g., 'Statut', 'Nb pers a charge').

    - MICE builds sequential models for each variable with missing data using other columns as predictors.
    - We configured it to use BayesianRidge or DecisionTreeRegressor, depending on the variable type.
    - It enables multi-pass updates, refining imputations iteratively and capturing interdependence.

5. **Predictive Mean Matching (PMM)**:
    - Applied within the MICE chain for fields where preserving realistic value ranges was critical (e.g., continuous variables with skew).
    - Rather than imputing raw regression predictions, PMM selects real, observed values closest to the predicted one.
    - This avoids extreme or unrealistic synthetic values and was key to preserving domain realism.

These choices reflect careful balancing:

- Simplicity where appropriate (MCAR)
- Domain constraints (conditional)
- Cross-feature correlation (MICE, KNN)
- Distributional realism (PMM)

The techniques used were implemented with:

- pandas for filtering and groupwise logic
- sklearn.impute.KNNImputer for KNN-based imputation
- sklearn.experimental.IterativeImputer with custom estimators for MICE
- Clustering with scipy.cluster.hierarchy to evaluate group structure for 'k' selection in KNN

Imputation Execution

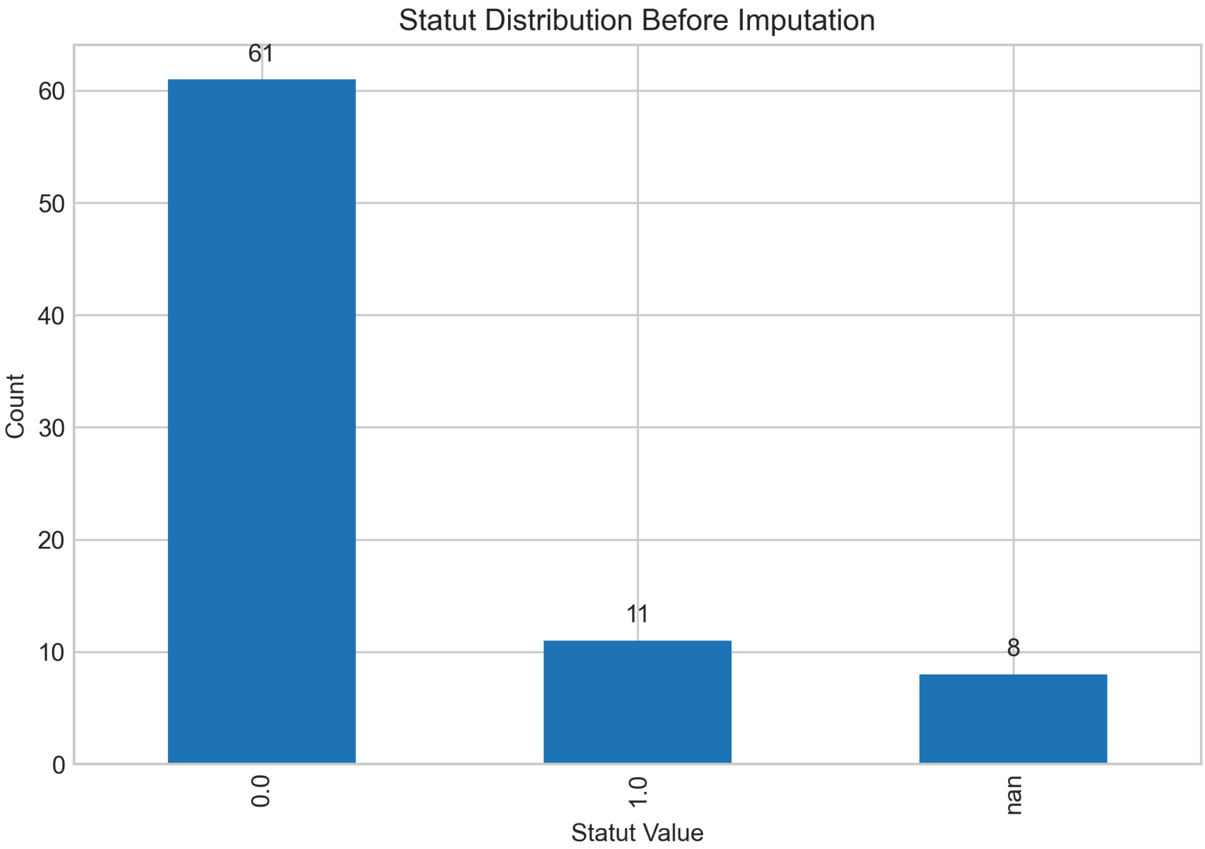
Objective

Execute the planned imputation strategies to fill missing values with statistically valid estimates.

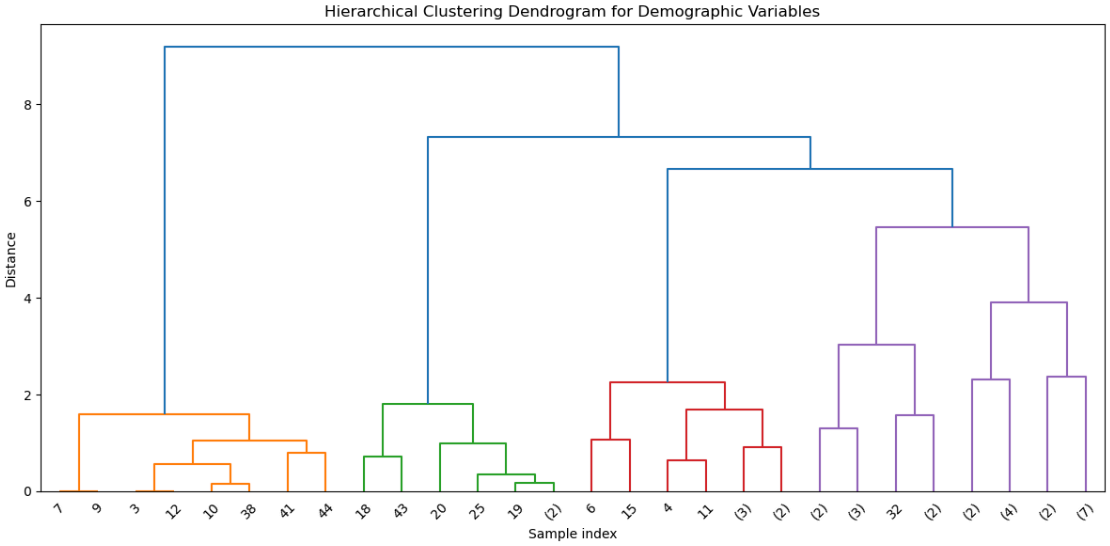
Implementation Details

The encoded dataset was loaded, preserving ordinal and one-hot encodings:

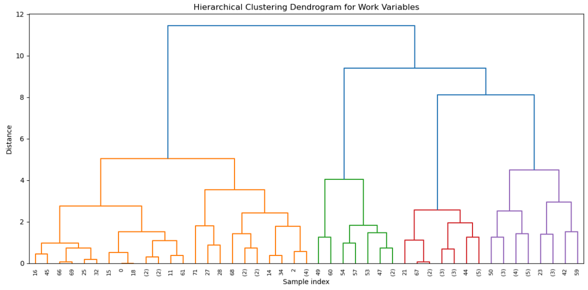
- Missing values were analyzed again to ensure no formatting errors remained
- Imputation was applied according to the strategy for each variable type
- Special attention was paid to preserving data structures and relationships

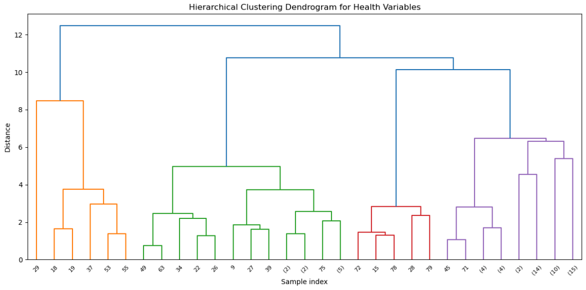


The hierarchical clustering dendrograms were particularly important for determining the optimal number of neighbors for KNN imputation:



Additional dendrograms were created for other variable groups:

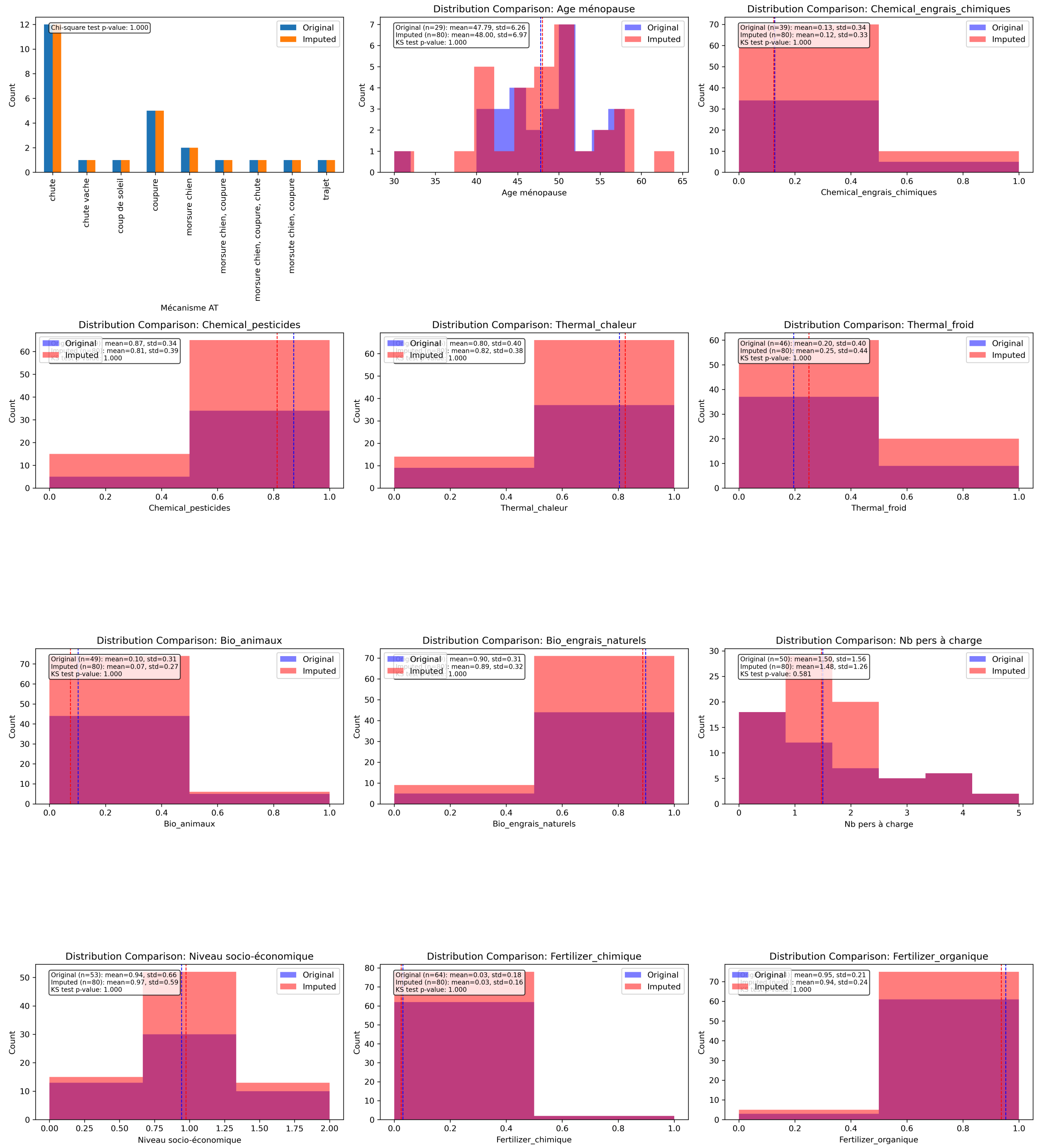




Step 5: Imputation Validation

Validation 1: Distribution Comparisons

To ensure the imputation process did not distort the original distributions of variables, we performed a series of distribution comparisons. For each variable with imputed values, we overlaid histograms of the original and imputed datasets.



The close alignment between distributions confirms that our imputation methods preserved the overall data structure.

Validation 2: Categorical Balance Verification

To ensure the imputation process did not distort the original class proportions of categorical variables, we performed a side-by-side comparison of distributions before and after imputation for each variable with missing categorical values.

The process included:

- Extracting value counts (frequencies) per category from both the original and imputed datasets
- Aligning these counts across all categories to ensure a fair comparison
- Plotting grouped bar charts to visually compare frequency changes
- Running a Chi-square test of independence to statistically test whether category proportions changed significantly

Variables included in this check:

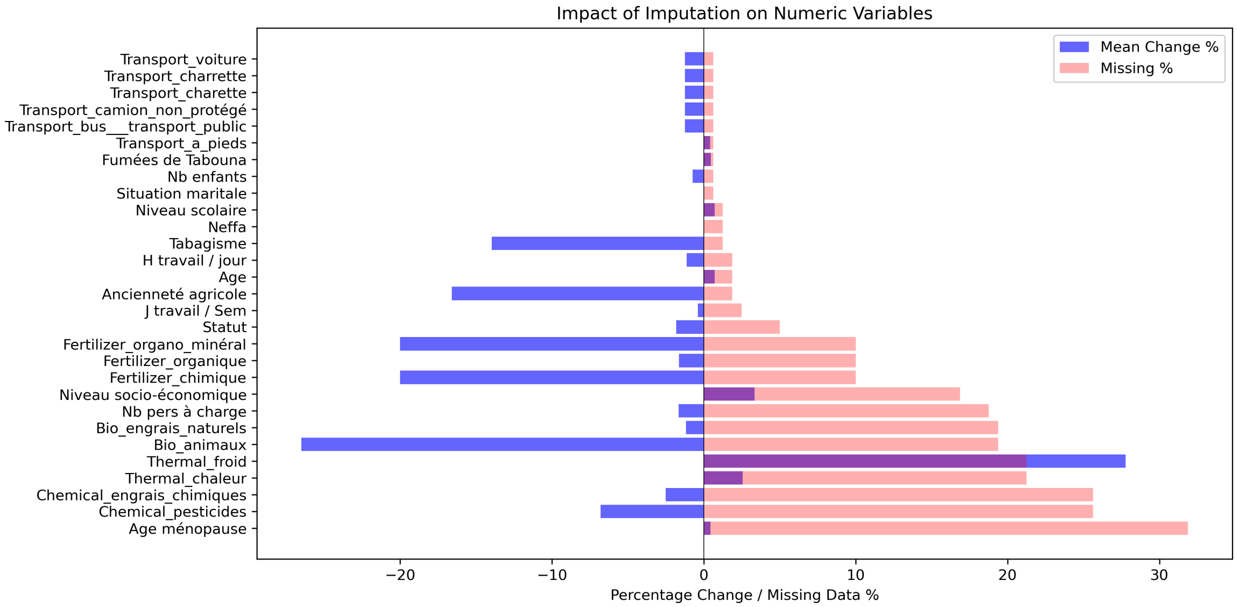
- Tabagisme
- Fumées de Tabouna
- Situation Maritale
- Statut
- Niveau socio-economique
- J travail / Sem

Chi-square p-values were displayed on each plot. For all major variables, the p-values were above 0.05, indicating that no significant distortion occurred due to imputation.

This proves that our categorical imputations preserved the original structure and balance of the data.

Validation 3: Numeric Imputation Impact Analysis

This validation step quantified the impact of imputation on numerical variables:



In this chart:

- Blue bars show the **percent change in mean value** after imputation
- Red transparent overlays reflect the **original missingness percentage** for context

A low blue bar means the imputed mean stayed close to the original, which indicates low imputation bias. The closer to zero, the better.

Conclusion: For all critical numeric variables, mean shifts were minor, confirming that imputation had low impact and preserved core statistical properties.

Validation 4: Cluster Structure Preservation

The dendrograms shown earlier were also used to validate that imputation did not artificially merge or split meaningful clusters. The consistency in cluster structures before and after imputation confirms that the natural groupings in the data were preserved.

This validation method is particularly valuable because it assesses the impact of imputation on the multivariate structure of the data, not just on individual variables.

Step 6: Dataset Decoding

Final Dataset Decoding & Export

After all imputation and validation steps, the final dataset was decoded to be human-readable using the following mappings and procedures:

1. Binary Variables: 1.0 = 'oui', 0.0 = 'non', NaN = missing
2. Tabagisme: 0.0 = 'non', 1.0 = 'passif', 2.0 = 'oui'
3. Equipment Use: 0 = 'jamais', 1 = 'parfois', 2 = 'souvent', 3 = 'toujours'
4. Ordinal Categories (e.g., marital status, education): Mapped using reverse dictionaries provided in the codebook
5. Profession du mari (husband's profession): Extracted from one-hot encoding - category set to the one with value 1, or 'missing' if 'nan' column is 1
6. Multi-value indicators (like fertilizers or thermal constraints): Multiple columns prefixed with e.g., 'Fertilizer_' were mapped back to comma-separated strings if their value = 1.0

The decoding logic was applied in the notebook 'final_decoded.ipynb' and exported to 'completed_female_farmers_data.xlsx'.

This dataset is now ready for analysis, reporting, and presentation in readable format. It reflects all steps including:

- Data cleaning and standardization
- Encoding and imputation
- Validation
- Reversing transformations to human labels

Conclusion

The data preparation pipeline successfully transformed raw data with inconsistent formats and missing values into a clean, complete dataset ready for advanced statistical analysis. Key achievements include:

1. **Standardization** of missing values and data formats
2. **Domain-aware encoding** that preserved semantic relationships

3. **Statistically sound imputation** using techniques matched to missingness mechanisms, including innovative use of dendrograms to determine optimal KNN parameters
4. **Rigorous validation** that confirmed data integrity through multiple methods
5. **Human-readable output** suitable for analysis and reporting

The resulting dataset enables the continuation of the project with robust multivariate analyses, including Principal Component Analysis (PCA) and Multiple Correspondence Analysis (MCA), as specified in the project requirements.