Exploratory Data Analysis - IT 462

Assignment - 1

Missingno Package

Group-ID: 10

Team Members:

1. Hammad Farid - 202312108
2. Anmol Rangwani - 202312027
3. Yuvraj Bhadoriya - 202411002

Introduction:

In data's ever-evolving saga, Missingno shines—a guiding star lighting the path to understanding datasets' missing puzzle pieces.

Missingno is a Python library that serves as a data detective, specializing in uncovering the enigma of missing data in datasets. It offers a toolkit of visualizations and tools designed to reveal the patterns and extent of missing values. With its clever name derived from "missing data" and "no values," Missingno provides data scientists and analysts with a compass to navigate the complexities of incomplete information. By offering visual insights like matrix plots, heatmaps, and dendrograms, this library transforms the challenge of missing data into an opportunity for deeper understanding and informed decision-making in data analysis and preprocessing.

1. Bar Plot (`msno.bar`)

The bar plot shows the count of non-missing values for each column in the dataset.

- **X-axis**: The names of the columns (variables) in the dataset.
- **Y-axis**: The count of non-missing values for each column.
- **purpose:**This plot is particularly useful for getting an immediate sense of which columns have missing data and to what extent. Columns with little or no missing data will have higher bars, while columns with significant amounts of missing data will have shorter bars.

2. Matrix Plot (`msno.matrix`)

The matrix plot shows a visual overview of the missing data in the dataset, where missing values are represented as white lines and present data as colored lines.

- **X-axis**: The index (rows) of the dataset.
- **Y-axis**: The names of the columns (variables).
- **purpose:**This plot is useful for identifying if there are specific patterns in missingness across rows. For example, it can help detect if certain rows are completely missing data, or if certain columns have a correlated missing pattern across many rows.

3. Dendrogram (`msno.dendrogram`)

The dendrogram clusters variables based on the similarity of their missingness patterns. It helps identify groups of columns that share similar missingness structures.

- **X-axis:** The column names.
- **Clusters: Groups of columns with similar missing data patterns.**
- **Purpose:**The dendrogram is useful for understanding if there are hierarchical relationships between variables in terms of missing data. It can reveal clusters of variables that are more likely to have missing values together.

4. Heatmap (`msno.heatmap`)

The heatmap shows correlations between the missingness of different columns. A strong correlation indicates that when one column has missing values, the other column is also likely to have missing values.

- **X-axis and Y-axis**: The column names (variables) in the dataset.
- **Color gradient**: Indicates the degree of correlation between missingness across columns.
- **Purpose:**This visualization is particularly useful for determining whether missing data in one variable is related to missing data in another variable, which can provide insights into whether the missing data mechanism is **Missing Completely at Random (MCAR)**, **Missing at Random (MAR)**, or **Missing Not at Random (MNAR)**.

Reading the Heatmap :

The color scale of the heatmap shows the strength of the correlation, with darker colors indicating higher correlations between missing values and lighter colors indicating lower or no correlation.

A correlation close to 1 (or -1) indicates a strong relationship between missingness in two columns, suggesting the missingness is not random.

A correlation near 0 suggests no relationship between the missingness of columns, pointing to MCAR.

In-built functions in missingno python library :

FUNCTIONS	DESCRIPTION
missingno.bar(df)	Creates a bar chart representing the number of missing values.
missingno.matrix(df)	Creates matrix plot for visualization of missing values
missingno.heatmap(df)	Creates a heatmap which shows correlation
missingno.dendrogram(df)	Creates a clustering dendrogram
missingno.nullity_filter (df, filter='top', p=0.5)	Used for filtering number of rows and columns according to the percentage of missing data
missingno.nullity_sort(df)	Sorting based on number of missing values
missingno.geoplot (df, x='longitude', y='latitude')	Geographically plot the missing values using longitude and latitude
missingno.upsample (df, n=2)	Increases the frequency of missing data patterns in the matrix plot to simulate larger datasets.
missingno.downsample (df, n=2)	Reduces size of dataset to make interpretation and visualization easier
missingno.warnings. filterwarnings('ignore')	Suppresses warnings related to missing data visualizations.
missingno.matrix (df, color=(0.25, 0.25, 0.25))	Provides the ability to customize the color palette of visualizations.

How To identify the type of missing data using a heatmap in `missingno`:

1. **MCAR (Missing Completely at Random -ranges from - 0.2 to 0.2):**
   - **No correlation** between missing values across columns.
   - Heatmap shows **low or no correlation** between missing data.
2. **MAR (Missing at Random-ranges from 0.3 to 1)**:
   - **Moderate correlation** between missingness in columns.

    - Heatmap shows **some correlation**, suggesting missing data is related to other observed variables.

3. **MNAR (Missing Not at Random)**:

    - **High correlation** between missing values.

    - Heatmap shows **strong correlation**, indicating missing data is related to the values themselves.

User-defined functions using missingno python library:

1.User Defined function for Finding the columns which exceeds the threshold using missingno:

Step 1: Calculate the percentage of missing values for each column using `df.isnull().mean()`. This provides the proportion of missing values in each column.

Step 2: Identify columns where the missing percentage exceeds the threshold. These are stored in the `incomplete_columns` list.

Step 3: If `incomplete_columns` has entries, print the column names and visualize the missing data using `msno.bar()`. If no columns exceed the threshold, print a message.

2.User Defined function for Finding the Rows which exceeds the threshold using missingno:

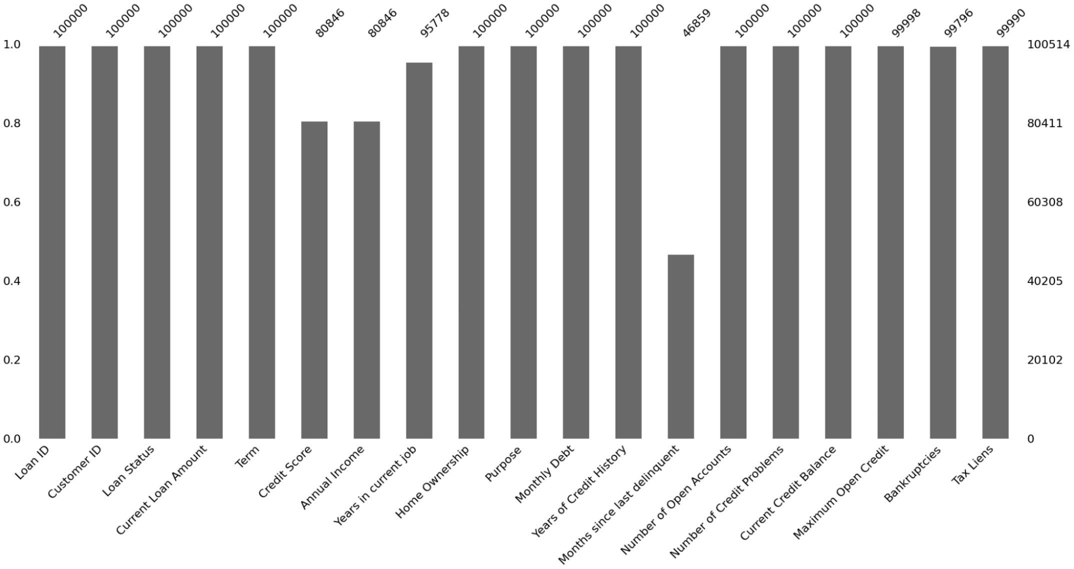
Step 1:Compute missing values per row: `row_nullity = df.isnull().mean(axis=1)` calculates the percentage of missing values for each row.

Step 2:Filter rows exceeding threshold: `incomplete_rows = df.loc[row_nullity > threshold]` selects rows with missing percentages greater than the threshold.

Step 3:Check and visualize:

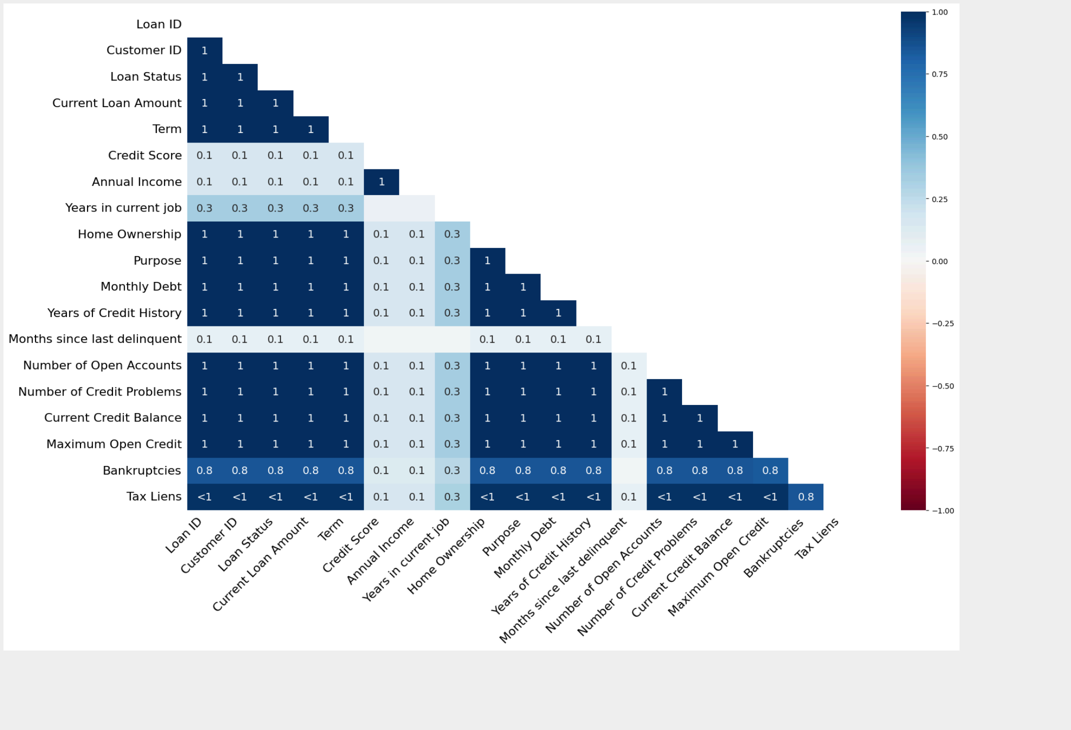
- **If `incomplete_rows` is not empty, print row indices and visualize with `msno.matrix()`.**

- **If no rows meet the threshold, print a message.**

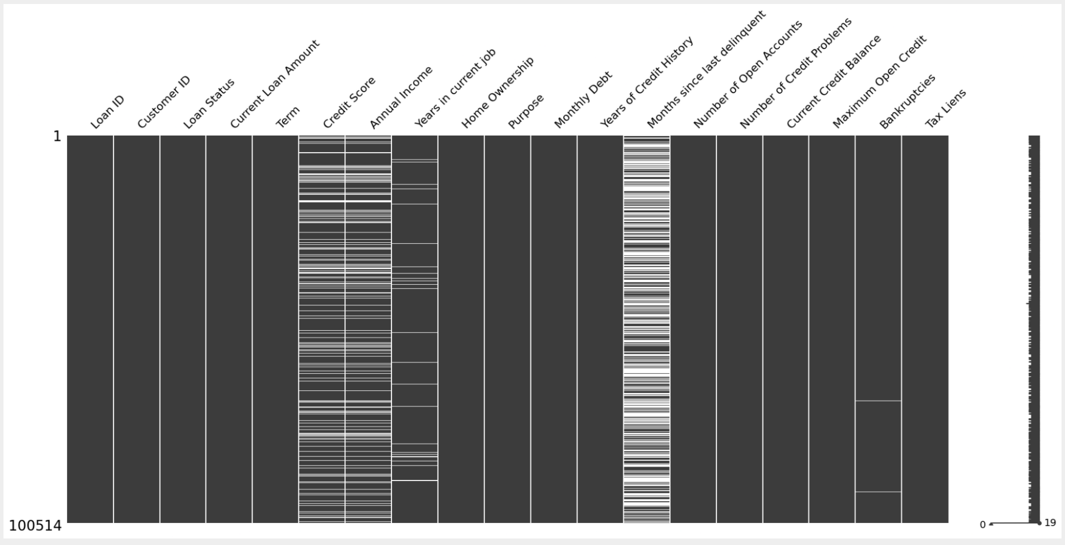


Bar plot

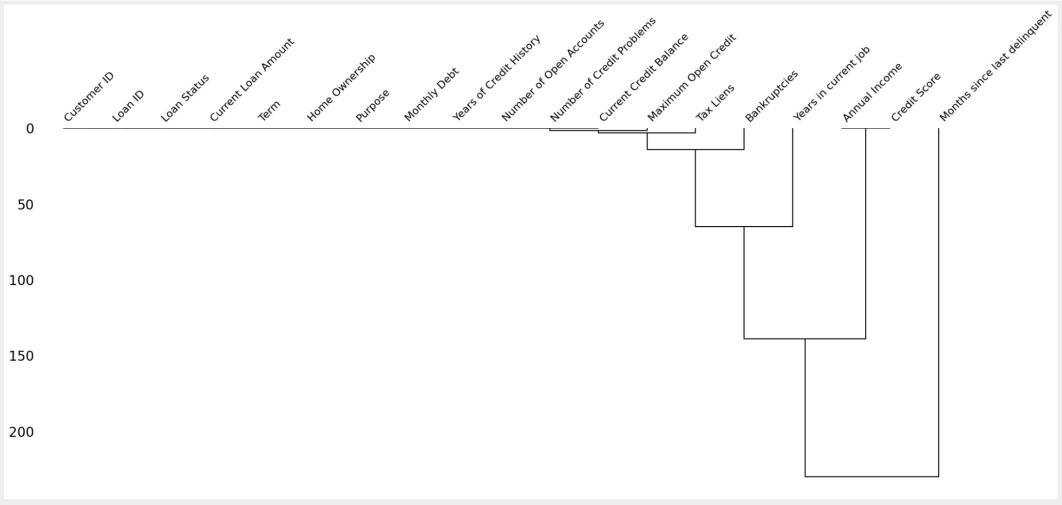
Here, from the above bar plot we can conclude that the column '**Months since last delinquent'** have the most no. missing values.



Heatmap



Matrix plot



Dendrogram

Github: https://github.com/yuvraj-daiict/exploratory-data-analysis

Colab: eda_group_10_assignment_1.ipynb