Dhirubhai Ambani Institute of Information and Communication Technology

Gandhinagar, Gujarat

Group Number : Group_4

Group Members :

Ghori Zeel Jivrajbhai (202201287)

Aryankumar Panchasara (202201056)

Nischay Agrawal (202411031)

Course : IT 462 Exploratory Data Analysis

Professor : Gopinath Panda

Semester : Autumn 2024

Assignment 1 : missingno package

Date : 19/9/24

Package documentation : Introduction

`missingno` is a Python library that provides simple and effective visualizations for missing data. The goal of the library is to make it easy to understand the extent and patterns of missing values in a dataset, which can help in deciding how to handle them. By visualizing missing data, you can quickly identify issues such as missing data patterns, correlations between missing values in different columns, and how missing values are distributed across rows and columns.

Code for Installation : pip install missingno

Barplot:

- **Purpose**: Displays a bar chart showing the count of missing values per column.

- **Usage**: This tool helps to quickly identify which columns have missing values and the amount of missing data in each column. On the left side of the plot, the y-axis scale ranges from 0.0 to 1.0 where 1.0 represents 100% data completeness and if the bar height is less than, it indicates that there are more missing values within that column. On the right hand side, the scale is measured in indexed values and represents the maximum number of rows within the data frame. On the top the numbers represent the total count of the non null values within that column.

Matrix Plot:

- **Purpose**: Provides a matrix-like visualization where missing values are shown as vertical lines, and the columns of the dataset are represented on the y-axis.

- **Usage**: It is particularly useful for visualizing missing data in depth-related or time-series datasets

  Visualizing Missing Data Locations:

1. **White Areas**: The plot turns white wherever there are missing values, allowing you to quickly see which parts of your data are incomplete.
2. **Sparkline**: Located on the right side, this small plot summarizes the completeness of each row. It shows overall data completeness, general shape of data availability across rows, row with minimum nullities, column count,etc

3. **Sparkline Interpretation**:
   Complete Rows: If a row has no missing values, its line in the sparkline is positioned at the far right.
   Increasing Missing Values: As missing values increase in a row, the line moves towards the left, indicating decreasing completeness.

Heatmap:

- **Purpose**: Identifies correlations or relationships between missing values in different columns. It helps determine if missing values in one column are related to missing values in another.

- **Usage**: Best suited for smaller datasets due to its detail and clarity in showing correlations.

  Interpreting the Heatmap

- High Positive Correlation: Values Close to 1: Indicates that when a column has missing values, another column also tends to have missing values.
- High Negative Correlation: Values Close to -1: Indicates that when a column has missing values, another column is likely to have data values present (vice versa).
- Low or No Correlation: Values Close to 0: Suggests there is little to no relationship between the missing values in the columns. Low correlations further indicate that the data are MAR.
- Very Strong Negative Correlation: Values <-1: Indicates a very strong inverse relationship between missing values in two columns.

Dendrogram:

- **Purpose**: Visualizes hierarchical clustering of missing values.

- **Usage**:

1. **Complete Columns**: Columns grouped together at level zero are complete, meaning their missing data patterns are not correlated with each other.
2. **Partial Correlation**: Columns are in the same branch as the complete columns, indicating that their missing values might be somewhat related to those of the grouped columns, but less strongly.

- **Tree-like Structure**: Shows columns as nodes in a tree, with branches representing similarities in missing data.
- **Grouping**: Columns grouped together at the same level indicate they share similar patterns of missing values.
- **Separation**: Columns that are more separated in the tree have less correlation in their missing data patterns.

How to get idea of type of missing value using heatmap :

1. MCAR (Missing Completely At Random)

- **Correlation Range**: Close to 0 (from -0.2 to 0.2).
- **Interpretation:** If the missingness in one column is unrelated to the missingness in other columns, the correlation between them should be near zero. This indicates that the missing data is likely completely random and not dependent on any observed variables.
- **What to look for:** The heatmap will show light colors, suggesting little to no correlation across columns.

2. MAR (Missing At Random)

- **Correlation Range:** Moderate to high positive correlation (from 0.3 to 1).
- **Interpretation:** If two columns have a moderate to high positive correlation, it means that the missingness in one column is likely related to missingness in the other. This implies that the data is MAR because the missing values in one column can be explained by the values in another column.
- **What to look for:** The heatmap will show darker colors (high positive correlation) for pairs of columns with missing data that are related.

3. NMAR (Not Missing At Random)

- NMAR occurs when the missingness is related to the unobserved values themselves (e.g., people with lower income tend to not report their income).
- **Key point:** NMAR cannot be detected just from visualizations like heatmaps or any external variables. It is the hardest to diagnose without domain knowledge.

How to get idea of type of missing value using dendrogram:

- Low-height connections suggest that columns are strongly related in terms of missing values, pointing to MAR.
- High-height connections or sparse clustering suggests missing values are independent, which supports MCAR.
- Unique patterns or isolated columns might hint at NMAR, but require domain expertise to confirm.

In-built functions in missingno python library :

FUNCTIONS	DESCRIPTION
missingno.bar(df)	Creates a bar chart to display number of missing values.
missingno.matrix(df)	Creates a matrix plot that visualizes location of missing data
missingno.heatmap(df)	Produces a heatmap that shows correlations.
missingno.dendrogram(df)	Generates a dendrogram that clusters
missingno.nullity_filter (df, filter='top', p=0.5)	Filters columns or rows based on the percentage of missing values.
missingno.nullity_sort(df)	Sorts dataset columns based on number of missing values.
missingno.geoplot (df, x='longitude', y='latitude')	Plots missing data geographically using longitude and latitude information.
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 the size of the dataset to make visualizations easier to interpret.
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.

User-defined functions in missingno python library

Function visualize_incomplete_columns(df, threshold=0.5):

1. Compute the percentage of missing values for each column:
missing_percentages = df.isnull().mean()
2. Find columns where the missing percentage is greater than the threshold:
incomplete_columns = missing_percentages[missing_percentages > threshold].index.tolist()
3. If : incomplete_columns is not empty:
Print the column names with more than threshold*100% missing values. Visualize the missing data in these columns using msno.bar(df[incomplete_columns]).
Else: Print a message indicating no columns have more than threshold*100% missing values.

Function visualize_incomplete_rows(df, threshold=0.5):

1. Compute the percentage of missing values for each row:
row_nullity = df.isnull().mean(axis=1)
2. Find rows where the missing percentage is greater than the threshold:
incomplete_rows = df.loc[row_nullity > threshold]
3. If : incomplete_rows is not empty:
Print the rows with more than threshold*100% missing values.
Visualize the missing data in these rows using msno.matrix(incomplete_rows).
Else: Print a message indicating no rows have more than threshold*100% missing values.

GITHUB LINK :

[Exploratory_Data_Analysis/Group4_missingno at main · zeelghori27012004/Exploratory_Data_Analysis (github.com)]

Google drive link :

https://drive.google.com/drive/folders/1IT4SnwyDAP3EqtRZ6Zp_PNVKwprNWBmT?usp=sharing