CAPTER – 5

Working with Missing Data

*Learning Topics*



* Checking for missing values using isnull() and notnull()
* Checking for missing values in graphical format
* Filling null values

**Code link:** [*Working\_with\_Missing\_Data*](https://github.com/ramasureshvijjana/Data_Science/tree/master/03_Dealing_with_missing_data)



Missing Data can occur when no information is provided for one or more items or for a whole unit. It is a very big problem in real life scenario. Missing Data can also refer as NA (Not Available) values in pandas. Sometimes many datasets simply arrive with missing data, either because it exists and was not collected or it never existed. For Example, suppose different user being surveyed may choose not to share their income, some user may choose not to share the address in this way many datasets went missing.

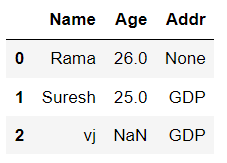
In Pandas missing data is represented by two values:

* ***None*:** None is a python single object that is often used for represents the missing data in python code.
* ***NaN*:** NaN (Not a Number) is a special floating-point value recognized by all systems that use the standard IEEE floating-point representation

Pandas treat None and *NaN* as essentially interchangeable for indicating missing or null values. To facilitate this convention, there are several useful functions for detecting, removing, and replacing null values in Pandas DataFrame:

* isnull()
* notnull()
* dropna()
* fillna()
* replace()
* interpolate()

### **1. Checking for Missing Values Using isnull() and notnull()**

 The *isnull()* or *notnull()* function are helpful to check the missing values in Pandas DataFrame. Both function help in checking whether a value is *null* or not. These functions can also be used in Pandas Series in order to find null values in a series. The *isnull()* function returns a DataFrame with Boolean values which are True for *NaN* values.

**1.1. Code:**

|  |
| --- |
| 1. import pandas as pd 2. # Checking whether the total data set has null values or not 3. print("isnull result of DataFrame:") 4. display(data.isna()) 5. # Checking whether a single column has null values or not 6. print(f"isnull result of Series: \n\n{data['Addr'].isnull()}") 7. ###############SECOND WAY################ 8. # Checking whether the total data set has null values or not 9. print("notnull result of DataFrame:") 10. display(data.notna()) 11. # Checking whether a single column has null values or not 12. print(f"notnull result of Series: \n\n{data['Addr'].notnull()}") |

***Output:***

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
| OwlIn above program, the second way output is reversing of first way output. |

### **2. Checking for Missing Values in Graphical Format**

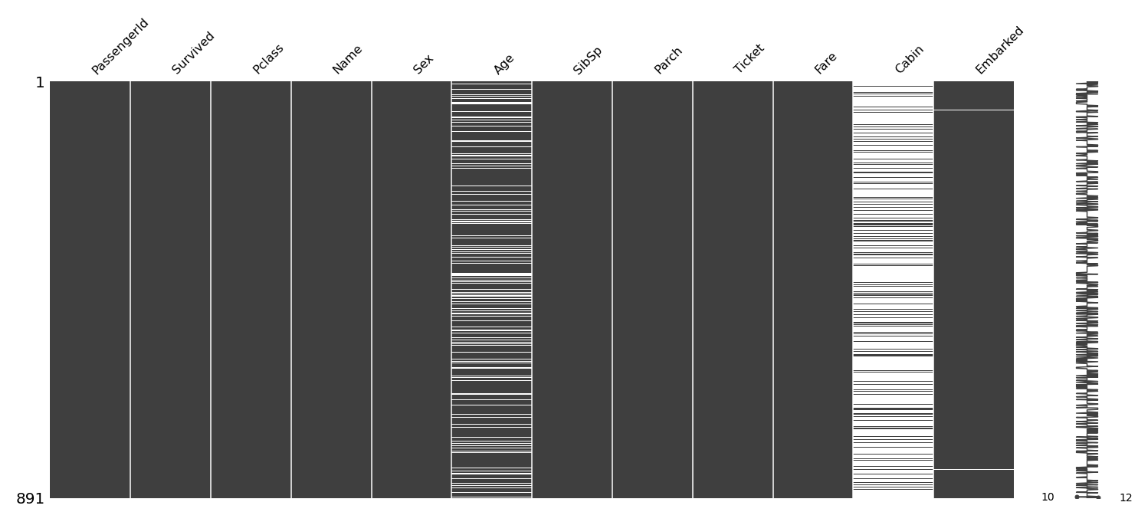
The missingno library helps to Identify the nature of null values in the data by using different types of graphs like matrix, bar graph, heatmap:

**2.1. Code:**

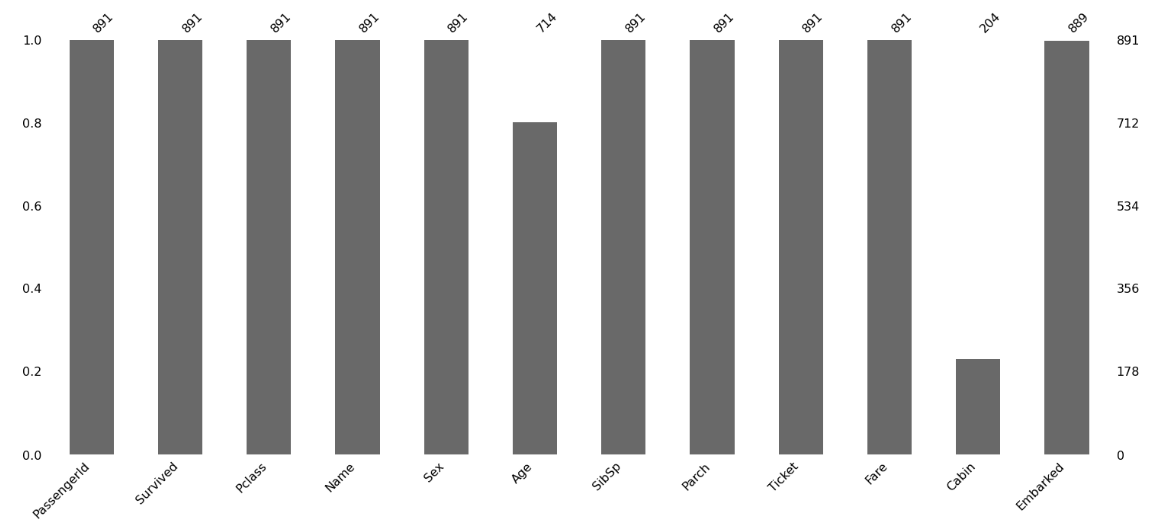
|  |
| --- |
| 1. # Installing missingno package 2. pip install missingno 3. # Importing the package 4. import missingno as msno 5. # printing matrix graph. 6. msno.matrix(data) 7. # printing bar graph. 8. msno.bar(data) 9. # printing heatmap graph. 10. msno.heatmap(data) |

***Output:***

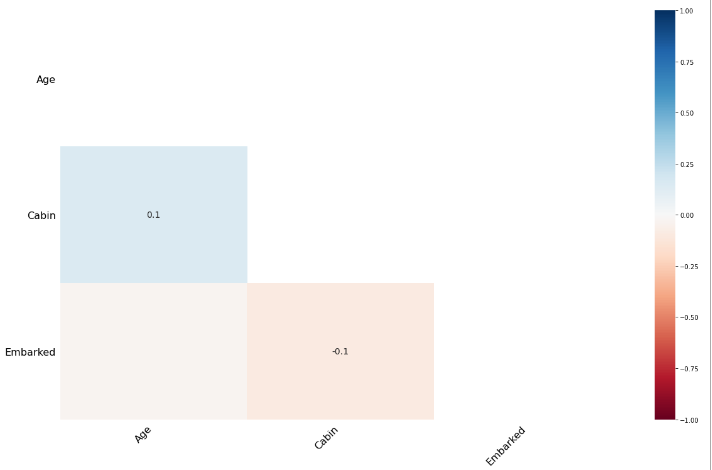
Printing matrix graph:



Printing matrix graph:



Printing matrix graph:



**3. Filling Null Values**

In order to fill null values we will use fillna*()*, *replace()* and *interpolate()* function. These functions replace *NaN* values with some other value of their own. The *interpolate()* function is basically uses in various interpolation technique to fill the missing values rather than hard-coding the value.

|  |  |
| --- | --- |
| Keyword / Function | *Use* |
| df.fillna(0) | *Fill null values with a specific value (0)* |
| df.fillna(method='pad') | *Fill null values with previous value.*  *Note: It can’t fill null value if first value is null in a column. Because first values don’t have previous value.* |
| df.fillna(method='bfill') | *Fill null values with next value.*  *Note: It can’t fill null value if last value is null in a column. Because last values don’t have next value.* |
| df.fillna('str\_val') | *Filling null values in string type of data.* |
| df.dropna() | *Dropping rows if any row has latest one null value.* |
| df.dropna(how = 'all') | *Dropping rows if all column values are null.* |
| df.dropna(axis = 1) | *Dropping columns if any column has latest one null value.* |
| df.dropna(  subset = ['clm\_1', 'clm\_2']) | *Dropping rows if all subset column values are null.* |