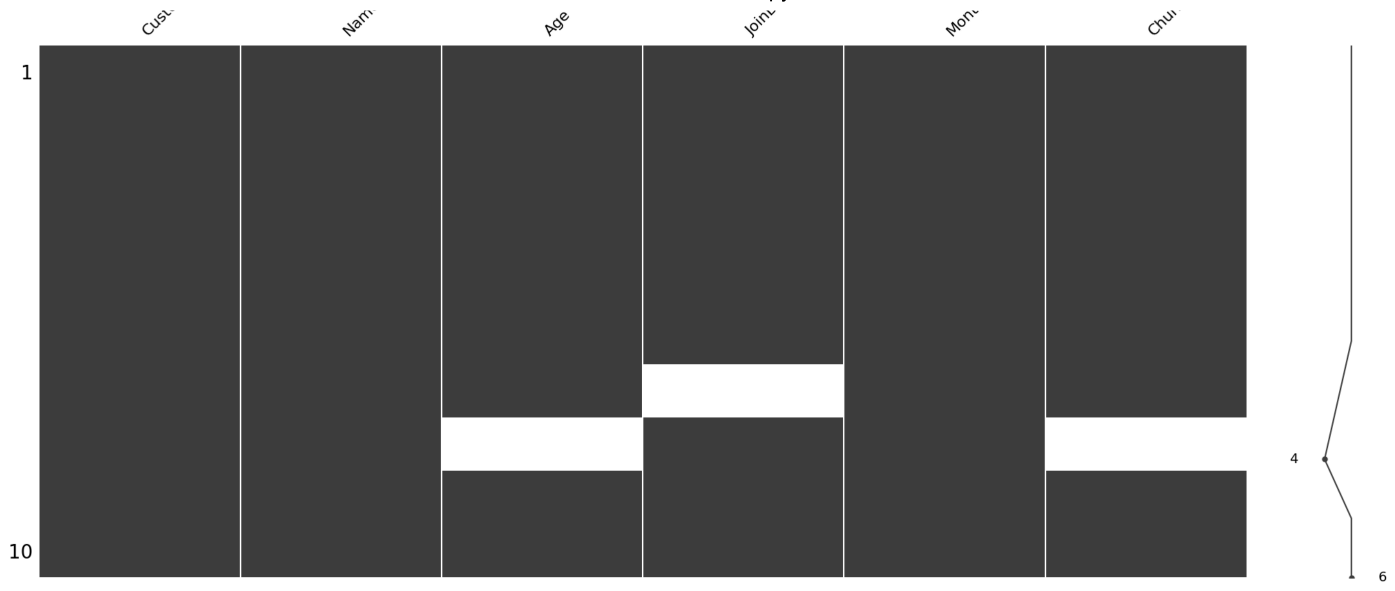
```
import pandas as pd


# Creating the dataset
data = {
    "CustomerID": ["C001", "C002", "C003", "C004", "C004", "C005", "C006", "C007", "C008", "C008"],
    "Name": ["John", "Alice", "BOB", "BOBY", "Eve", "eve", "Steve", "Ramu", "mary", "Bob"],
    "Age": [25, 34, 17, 29, 29, 120, -5, None, 220, 30],
    "JoinDate": ["12/1/2024", "11/15/2023", "6/1/2022", "6/1/2022", "12/5/2024", "invalid_date", None, "1/1/2024", "3/5/2023", "3/5/202
    "MonthlyCharges": [29.85, 56.95, 4000, 75.5, 75.5, 45.99, 60, 49.99, -30, 55],
    "Churn": ["No", "Yes", "No", "No", "No", "Yes", "No", None, "Yes", "No"]
}


# Converting the dictionary into a pandas DataFrame
df = pd.DataFrame(data)


# Display the dataset
print(df)
# Checking for missing values using isnull()
print("Missing values using isnull():")
print(df.isnull())


# Checking for missing values using isna()
print("\nMissing values using isna():")
print(df.isna())


# Checking for non-missing values using notna()
print("\nNon-missing values using notna():")
print(df.notna())


# Checking for non-missing values using notnull()
print("\nNon-missing values using notnull():")
print(df.notnull())


# Checking if any column has missing values using any()
print("\nColumns with missing values using any():")
print(df.isnull().any())


# Checking if any column has missing values using sum()
print("\nCount of missing values in each column using sum():")
print(df.isnull().sum())

import missingno as msno
import pandas as pd


# Assuming 'df' is your DataFrame


# Visualizing missing values using msno.matrix()
msno.matrix(df)
```

```
  CustomerID   Name    Age      JoinDate  MonthlyCharges Churn
0       C001   John   25.0     12/1/2024           29.85    No
1       C002  Alice   34.0    11/15/2023           56.95   Yes
2       C003    BOB   17.0      6/1/2022         4000.00    No
3       C004   BOBY   29.0      6/1/2022           75.50    No
4       C004    Eve   29.0     12/5/2024           75.50    No
5       C005    eve  120.0  invalid_date           45.99   Yes
6       C006  Steve   -5.0          None           60.00    No
7       C007   Ramu    NaN      1/1/2024           49.99  None
8       C008   mary  220.0      3/5/2023          -30.00   Yes
9       C008    Bob   30.0      3/5/2023           55.00    No
Missing values using isnull():
   CustomerID   Name    Age  JoinDate  MonthlyCharges  Churn
0       False  False  False     False           False  False
1       False  False  False     False           False  False
2       False  False  False     False           False  False
3       False  False  False     False           False  False
4       False  False  False     False           False  False
5       False  False  False     False           False  False
6       False  False  False      True           False  False
7       False  False   True     False           False   True
8       False  False  False     False           False  False
9       False  False  False     False           False  False

Missing values using isna():
   CustomerID   Name    Age  JoinDate  MonthlyCharges  Churn
0       False  False  False     False           False  False
1       False  False  False     False           False  False
2       False  False  False     False           False  False
3       False  False  False     False           False  False
4       False  False  False     False           False  False
5       False  False  False     False           False  False
6       False  False  False      True           False  False
7       False  False   True     False           False   True
8       False  False  False     False           False  False
9       False  False  False     False           False  False

Non-missing values using notna():
   CustomerID  Name    Age  JoinDate  MonthlyCharges  Churn
0        True  True   True      True            True   True
1        True  True   True      True            True   True
2        True  True   True      True            True   True
3        True  True   True      True            True   True
4        True  True   True      True            True   True
5        True  True   True      True            True   True
6        True  True   True     False            True   True
7        True  True  False      True            True  False
8        True  True   True      True            True   True
9        True  True   True      True            True   True

Non-missing values using notnull():
   CustomerID  Name    Age  JoinDate  MonthlyCharges  Churn
0        True  True   True      True            True   True
1        True  True   True      True            True   True
2        True  True   True      True            True   True
3        True  True   True      True            True   True
4        True  True   True      True            True   True
5        True  True   True      True            True   True
6        True  True   True     False            True   True
7        True  True  False      True            True  False
8        True  True   True      True            True   True
9        True  True   True      True            True   True

Columns with missing values using any():
CustomerID        False
Name              False
Age                True
JoinDate           True
MonthlyCharges    False
Churn              True
dtype: bool

Count of missing values in each column using sum():
CustomerID        0
Name              0
Age               1
JoinDate          1
MonthlyCharges    0
Churn             1
dtype: int64
<Axes: >
```



```
import pandas as pd


# Assuming 'df' is your original DataFrame


# Fill missing values for 'Age' and 'MonthlyCharges' using mean
df['Age'] = df['Age'].fillna(df['Age'].mean())
df['MonthlyCharges'] = df['MonthlyCharges'].fillna(df['MonthlyCharges'].mean())


# Fill missing values for 'Churn' using mode (since it's categorical)
df['Churn'] = df['Churn'].fillna(df['Churn'].mode()[0])


# Fill missing values for 'JoinDate' using mode (since it's categorical)
df['JoinDate'] = df['JoinDate'].fillna(df['JoinDate'].mode()[0])


# Fill missing values for 'Name' using mode (since it's categorical)
df['Name'] = df['Name'].fillna(df['Name'].mode()[0])


# Display the new dataset with missing values filled
print(df)


df['Age'] = df['Age'].replace(-5, df['Age'].mean())  # Replace -5 with the mean of Age
df['MonthlyCharges'] = df['MonthlyCharges'].replace(-30, df['MonthlyCharges'].mean())
df['JoinDate'] = df['JoinDate'].replace('invalid_date', df['JoinDate'].mode()[0])  # Replace 'invalid_date' with the mode of JoinDate
# Display the new dataset with missing values filled
print(df)
```

```
  CustomerID   Name         Age      JoinDate  MonthlyCharges Churn
0       C001   John   25.000000     12/1/2024           29.85    No
1       C002  Alice   34.000000    11/15/2023           56.95   Yes
2       C003    BOB   17.000000      6/1/2022         4000.00    No
3       C004   BOBY   29.000000      6/1/2022           75.50    No
4       C004    Eve   29.000000     12/5/2024           75.50    No
5       C005    eve  120.000000  invalid_date           45.99   Yes
```

```
6       C006  Steve   -5.000000      3/5/2023           60.00    No
7       C007   Ramu   55.444444      1/1/2024           49.99    No
8       C008   mary  220.000000      3/5/2023          -30.00   Yes
9       C008    Bob   30.000000      3/5/2023           55.00    No
  CustomerID   Name         Age    JoinDate  MonthlyCharges Churn
0       C001   John   25.000000   12/1/2024          29.850    No
1       C002  Alice   34.000000  11/15/2023          56.950   Yes
2       C003    BOB   17.000000    6/1/2022        4000.000    No
3       C004   BOBY   29.000000    6/1/2022          75.500    No
4       C004    Eve   29.000000   12/5/2024          75.500    No
5       C005    eve  120.000000    3/5/2023          45.990   Yes
6       C006  Steve   55.444444    3/5/2023          60.000    No
7       C007   Ramu   55.444444    1/1/2024          49.990    No
8       C008   mary  220.000000    3/5/2023         441.878   Yes
9       C008    Bob   30.000000    3/5/2023          55.000    No
```

```
import pandas as pd


# Assuming 'df' is your DataFrame


# Displaying the unique names in the 'Name' column
unique_names = df['Name'].unique()


# Printing the unique names
print(unique_names)
```

```
['John' 'Alice' 'BOB' 'BOBY' 'Eve' 'eve' 'Steve' 'Ramu' 'mary' 'Bob']
```

```
import pandas as pd


# Assuming 'df' is your DataFrame


# Identify the duplicated rows based on 'CustomerID'
duplicated_rows = df[df.duplicated(subset='CustomerID', keep=False)]


# Display duplicated rows
print("Duplicated rows based on 'CustomerID':")
print(duplicated_rows)


# Remove duplicated rows based on 'CustomerID'
df_cleaned = df.drop_duplicates(subset='CustomerID', keep='first')


# Display the cleaned dataset
print("\nDataset after removing duplicated rows:")
print(df_cleaned)
```

```
Duplicated rows based on 'CustomerID':
  CustomerID  Name    Age   JoinDate  MonthlyCharges Churn
3       C004  BOBY   29.0   6/1/2022          75.500    No
4       C004   Eve   29.0  12/5/2024          75.500    No
8       C008  mary  220.0   3/5/2023         441.878   Yes
9       C008   Bob   30.0   3/5/2023          55.000    No

Dataset after removing duplicated rows:
  CustomerID   Name         Age    JoinDate  MonthlyCharges Churn
0       C001   John   25.000000   12/1/2024          29.850    No
1       C002  Alice   34.000000  11/15/2023          56.950   Yes
2       C003    BOB   17.000000    6/1/2022        4000.000    No
3       C004   BOBY   29.000000    6/1/2022          75.500    No
5       C005    eve  120.000000    3/5/2023          45.990   Yes
6       C006  Steve   55.444444    3/5/2023          60.000    No
7       C007   Ramu   55.444444    1/1/2024          49.990    No
8       C008   mary  220.000000    3/5/2023         441.878   Yes
```