

Handling Missing Values

```
In [ ]: import pandas as pd

# Load the dataset
df = pd.read_csv('your_dataset.csv')

# Check for missing values
print(df.isnull().sum())

# Remove rows with missing values
df = df.dropna()

# Fill missing values with mean
df = df.fillna(df.mean())

# Fill missing values with median
df = df.fillna(df.median())

# Forward fill missing values
df = df.ffill()

# Backward fill missing values
df = df.bfill()
```

In []: In the above code snippet, we first load the dataset using `pd.read_csv()` function. Then, we use the `isnull().sum()` method to check the number of missing values in each column.

Next, we demonstrate a few common techniques for handling missing data:

Removing rows with missing values: `df.dropna()` will drop all rows that contain any missing values. Filling missing values with mean or median: `df.fillna(df.mean())` fills missing values with the mean value of each column, while `df.fillna(df.median())` fills missing values with the median value of each column.

Forward filling and backward filling: `df.ffill()` fills missing values with the last valid value in the column (forward fill), while `df.bfill()` fills missing values with the next valid value in the column (backward fill).

You can choose the appropriate method(s) based on your dataset and the nature of the missing data. Remember to adapt the code to match the specific dataset and column names in your data.

