Notes by:- JPwebdevelopers.

Missing Data in Pandas

→ Missing data is a common issue in datasets. They can occur due to incomplete data entry, errors in data collection, or missing records.

→ Python provides multiple way to detect, handle and fill missing values efficiently.

Checking for Missing Values

isna / isnull() :- Returns True for missing values and False otherwise.

Program

```
import Pandas as pd
import numpy as np

data = {'A': [1, 2, np.nan, 4],
        'B': [5, np.nan, np.nan, 8],
        'C': ['a', 'b', 'c', None]}
df = pd.DataFrame(data)
print (df.isna())
```

Output :-

	A	B	C
0	False	False	False
1	False	True	False
2	True	True	False
3	False	False	True

Notes by:- jpwebdevelopers

Summing Missing Values

```
print(df.isna().sum())
```

Output

```
A    1
B    2
C    1
```


(2) Dropping Missing Values

If missing data is not acceptable, we can remove rows or columns with missing values.

dropna()

- Remove rows with missing values:-
  df_cleaned = df.dropna()

- Remove columns with any missing values:-
  df_cleaned = df.dropna(axis=1)
  (This removes all columns containing NaN)

- Remove rows where all values are missing:-
  df_cleaned = df.dropna(how='all')

- Remove rows with missing value in specific column:-
  df_cleaned = df.dropna(subset=['A', 'B'])

(3) Filling Missing Values

Instead of dropping missing data, we can fill it with a suitable value.

(i) fillna() :- The fillna() function is used to replace missing value. We can replace NaN with a specific value like 0.


```
import pandas as pd
import numpy as np
```

```
data = {'Name': ['Palvi', 'jp', 'web', 'developers'],
        'Age': [25, np.nan, 30, np.nan],
        'Score': [85, 90, np.nan, 88]}

df = pd.DataFrame(data)
df_filled = df.fillna(0)
print(df_filled)
```

Output:

	Name	Age	Score
0	Palvi	25.0	85.0
1	jp	0.0	90.0
2	web	30.0	0.0
3	developers	0.0	88.0

#JPNotes

(2) Fill with Previous or Next Value (fillna())

We can fill NaN with the previous or next values in column.

Fill with Previous Value (Forward Fill)

```
df_forward = df.fillna(method='ffill')
print(df_forward)
```

Output:

	Name	Age	Score
0	Palvi	25.0	85.0
1	JP	25.0	90.0
2	web	30.0	90.0
3	developers	30.0	88.0

* Each NaN is replaced with the Previous value.

- fill with Next Value (Backward Fill)

```
df_forward = df.fillna(method='bfill')
print(df_forward)
```

@jpwebdevelopers

Output

	Name	Age	Score
0	Palvi	25.0	85.0
1	JP	30.0	90.0
2	web	30.0	88.0
3	developers	NaN	88.0

* Each NaN is replaced with the Next value.

(3) Replace Missing Values (replace())

We can replace any specific value like -1 using replace().

Replace NaN with -1

```
df_replace = df.replace(np.nan, -1)
Print(df_replace)
```

Output

	Name	Age	Score
0	Palvi	25.0	85.0
1	JP	-1.0	90.0
2	web	30.0	-1.0
3	developers	-1.0	88.0

#JPNotes

(4) Filling Missing Values using interpolation

The interpolate() function estimates missing values based on other values.

Fill missing values with linear interpolation

```
df_interpolated = df.interpolate()
print(df_interpolated)
```

Output

	Name	Age	Score
0	Palvi	25.0	85.0
1	JP	27.5	90.0
2	web	30.0	89.0
3	developers	30.0	88.0

→ Missing values are filled by estimating between known values.

Notes by :- jpwebdevelopers