

Class #15: Working with Pandas (Infinity) -

Part 2

1. Hierarchical indexing in Pandas.
2. Handling missing data in Pandas.
3. Data wrangling with Pandas.
4. Useful methods and operations in Pandas. Assignment #15:

2. Handling missing data in Pandas.

Missing data is very common in many data analysis applications. pandas has a great ability to deal with the missing data.

Let's learn some convenient methods to deal with **missing data in pandas**:

- isnull(), isna(), notnull(), dropna(), fillna(),

একটি pandas ডেটাফ্রেম তৈরি nan value সহ এবং সেটি প্রিন্ট কর:

```
import numpy as np
import pandas as pd

# Step 1: Define the data dictionary
data_dict = {
    'A': [1, 2, np.nan, 4, np.nan],
    'B': [np.nan, np.nan, np.nan, np.nan, np.nan],
    'C': [11, 12, 13, 14, 15],
    'D': [16, np.nan, 18, 19, 20]
}

# Step 2: Create the DataFrame from the dictionary
df = pd.DataFrame(data_dict)

# Step 3: Print the DataFrame using an f-string
print(f"DataFrame:\n{df}")
```

```
DataFrame:
   A   B   C   D
0  1.0 NaN 11 16.0
1  2.0 NaN 12  NaN
2  NaN NaN 13 18.0
3  4.0 NaN 14 19.0
4  NaN NaN 15 20.0
```

1. ডেটা ডিকশনারি তৈরি:

- data_dict নামের একটি ডিকশনারি তৈরি করা হয়েছে, যেখানে বিভিন্ন কলামের জন্য ডেটা দেওয়া আছে (যেমন 'A', 'B', 'C', 'D')। কিছু ভালু np.nan রয়েছে, যা মানে অনুপস্থিত ডেটা (missing data)।
- 2. ডেটাফ্রেম তৈরি:
 - pd.DataFrame(data_dict) ব্যবহার করে ডিকশনারি থেকে একটি পাণ্ডাস ডেটাফ্রেম তৈরি করা হয়েছে।
- 3. ডেটাফ্রেম প্রিন্ট:
 - print(f"DataFrame:\n{df}") ব্যবহার করে ডেটাফ্রেমটি প্রিন্ট করা হয়েছে। f-string ব্যবহার করা হয়েছে যাতে ডেটাফ্রেমটি সুন্দরভাবে আউটপুটে প্রদর্শিত হয়।

pandas ডেটাফ্রেম তৈরি করে এবং তারপর ডেটাফ্রেমে অনুপস্থিত (missing) মান, missing value এর সংখ্যা চেক করে।

```
import numpy as np
import pandas as pd

# Step 1: Define the data dictionary
data_dict = {
    'A': [1, 2, np.nan, 4, np.nan],
    'B': [np.nan, np.nan, np.nan, np.nan, np.nan],
    'C': [11, 12, 13, 14, 15],
    'D': [16, np.nan, 18, 19, 20]
}

df = pd.DataFrame(data_dict)
print(f"DataFrame:\n{df}\n")

# Check for missing values whole DataFrame
print(f"Check for missing values in the entire DataFrame:\n{df.isnull()}\n")

# Sum of missing values in each column whole DataFrame
print(f"Summation of missing values in each column:\n{df.isnull().sum()}\n")

# Check and sum missing values only for Only column 'A'
print(f"Missing values in column 'A':\n{df['A'].isnull()}\n")
print(f"Summation of missing values in column 'A':\n{df['A'].isnull().sum()}\n")
```

```
DataFrame:
   A  B  C  D
0  1.0 NaN 11 16.0
1  2.0 NaN 12  NaN
2  NaN NaN 13 18.0
3  4.0 NaN 14 19.0
4  NaN NaN 15 20.0
```

1. ডেটাফ্রেম তৈরি:

- 🔗 একটি ডিকশনারির মাধ্যমে ডেটাফ্রেম তৈরি করা হয়েছে, যেখানে কিছু কলামে np.nan ব্যবহার করা হয়েছে যা অনুপস্থিত মানের প্রতীক।

Check for missing values in the entire DataFrame:

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

Summation of missing values in each column:


```
A      2
B      5
C      0
D      1
dtype: int64
```


Missing values in column 'A':

```
0      False
1      False
2       True
3      False
4       True
Name: A, dtype: bool
```


Summation of missing values in column 'A': 2

2. অনুপস্থিত মান চেক করা:

 `df.isnull()` ব্যবহার করে পুরো ডেটাবেইজে কোন মানগুলি অনুপস্থিত (missing) তা চেক করা হয়েছে।

 `df.isnull().sum()` ব্যবহার করে প্রতিটি কলামে মোট কতগুলো অনুপস্থিত মান আছে তা বের করা হয়েছে।

3. শুধু 'A' কলামে অনুপস্থিত মান চেক:

 'A' কলামে কোন মানগুলি অনুপস্থিত তা `df['A'].isnull()` দিয়ে চেক করা হয়েছে এবং তার পরিমাণ `df['A'].isnull().sum()` দিয়ে বের করা হয়েছে।

`df....isnull().sum()` সর্বমোট কতগুলো NaN আছে, দেখিয়ে দিবে।

এই কোডটি মূলত ডেটাবেইজে কোন কলামে কোন মান অনুপস্থিত রয়েছে, এবং তার পরিমাণ কত, তা বের করতে ব্যবহৃত হয়।

`isnull` এর পরিবর্তে `isna` ব্যবহার। DataFrame এ row তে `loc/iloc` ব্যবহার

```
import numpy as np
import pandas as pd

# Step 1: Define the data dictionary
data_dict = {
    'A': [1, 2, np.nan, 4, np.nan],
    'B': [np.nan, np.nan, np.nan, np.nan, np.nan],
    'C': [11, 12, 13, 14, 15],
    'D': [16, np.nan, 18, 19, 20]
}
df = pd.DataFrame(data_dict)
print(f"DataFrame:\n{df}\n")

# Check for missing values in the entire DataFrame
print(f"Missing values in the entire DataFrame:\n{df.isna()}\n")
# Sum of missing values in each column
print(f"Sum of missing values in each column:\n{df.isna().sum()}\n")

# Check and sum missing values for row 3
```

```
print(f"Sum of missing values in row 3:
{df.loc[3].isnull().sum()}")
```

DataFrame:

	A	B	C	D
0	1.0	NaN	11	16.0
1	2.0	NaN	12	NaN
2	NaN	NaN	13	18.0
3	4.0	NaN	14	19.0
4	NaN	NaN	15	20.0

Missing values in the entire DataFrame:

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

Sum of missing values in each column:

```
A      2
B      5
C      0
D      1
dtype: int64
```

Sum of missing values in row 3:
1

1. ডেটাফ্রেম তৈরি:

data_dict ডিকশনারি ব্যবহার করে একটি পাণ্ডাস ডেটাফ্রেম তৈরি করা হয়েছে, যেখানে কিছু মান np.nan (অনুপস্থিত) দেওয়া হয়েছে।

2. ডেটাফ্রেমে অনুপস্থিত মান চেক:

df.isna() ব্যবহার করে ডেটাফ্রেমের প্রতিটি সেলের মধ্যে কোন মানটি অনুপস্থিত তা চেক করা হয়।

3. কলামভিত্তিক অনুপস্থিত মানের পরিমাণ:

df.isna().sum() ব্যবহার করে প্রতিটি কলামে কতটি অনুপস্থিত মান আছে তা গণনা করা হয় এবং প্রিন্ট করা হয়।

4. রো ৩-এ অনুপস্থিত মানের পরিমাণ:

df.loc[3].isnull().sum() দিয়ে রো ৩-এ কতটি মান অনুপস্থিত তা বের করা হয় এবং আউটপুটে দেখানো হয়।

এভাবে কোডটি ডেটাফ্রেমের অনুপস্থিত মান চেক এবং তাদের পরিমাণ নির্ণয় করে।

একটি pandas ডেটাফ্রেম তৈরি এবং তার মধ্যে মান অনুপস্থিত নয় (non-missing) চেক করে, সাথে তার পরিমাণও গণনা করাঃ

```
import numpy as np
import pandas as pd

# Define the data dictionary
data_dict = {
    'A': [1, 2, np.nan, 4, np.nan],
    'B': [np.nan, np.nan, np.nan, np.nan, np.nan],
    'C': [11, 12, 13, 14, 15],
    'D': [16, np.nan, 18, 19, 20]
}

df = pd.DataFrame(data_dict)
print(f"DataFrame:\n{df}\n")
# Display shape of the DataFrame
print(f"Shape of the DataFrame: {df.shape}\n")

# Display non-missing values
print(f"Non-missing values in the DataFrame:\n{df.notnull()}\n")
```

```
# Sum of non-missing values in each column
print(f"Sum of non-missing values in each
column:\n{df.notnull().sum()}\n")

# Total number of non-missing values in the DataFrame
print(f"Total non-missing values in the DataFrame:
{df.notnull().sum().sum()}")
```

DataFrame:

	A	B	C	D
0	1.0	NaN	11	16.0
1	2.0	NaN	12	NaN
2	NaN	NaN	13	18.0
3	4.0	NaN	14	19.0
4	NaN	NaN	15	20.0

Shape of the DataFrame: (5, 4)

Non-missing values in the DataFrame:


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

Sum of non-missing values in each column:


```
A      3
B      0
C      5
D      4
dtype: int64
```

Total non-missing values in the DataFrame: 12


1. ডেটাফ্রেম তৈরি:

 data_dict ডিকশনারির মাধ্যমে একটি ডেটাফ্রেম (df) তৈরি করা হয়েছে, যেখানে কিছু মান np.nan দিয়ে অনুপস্থিত করা হয়েছে।


2. ডেটাফ্রেমের আকার দেখানো:

 df.shape দিয়ে ডেটাফ্রেমের সারি ও কলামের সংখ্যা (রো × কলাম) দেখানো হয়।


3. অনুপস্থিত মান চেক:

 df.notnull() ব্যবহার করে ডেটাফ্রেমের প্রতিটি সেলে কোন মানটি অনুপস্থিত নয় (non-missing) তা চেক করা হয়। এটি True বা False ফেরত দেয়।

4. কলামভিত্তিক অনুপস্থিত মানের পরিমাণ:

 df.notnull().sum() ব্যবহার করে প্রতিটি কলামে কতটি মান অনুপস্থিত নয়, তার পরিমাণ গণনা করা হয়।

5. ডেটাফ্রেমে মোট অনুপস্থিত মানের পরিমাণ:

 df.notnull().sum().sum() দিয়ে ডেটাফ্রেমে মোট কতটি non-missing মান আছে, তা গণনা করা হয়।

একটি ডেটাফ্রেম তৈরি করে এবং তার মধ্যে কিছু পরিসংখ্যানিক হিসাব:

```
import numpy as np
import pandas as pd

# Define the data dictionary
data_dict = {
    'A': [1, 2, np.nan, 4, np.nan],
    'B': [np.nan, np.nan, np.nan, np.nan, np.nan],
    'C': [11, 12, 13, 14, 15],
    'D': [16, np.nan, 18, 19, 20]
}
```

```
df = pd.DataFrame(data_dict)
print(f"DataFrame:\n{df}\n")

# Sum of values in column "A" (NaN treated as 0)
print(f"Sum of values in column 'A': {df['A'].sum()}")

# Mean of values in column "A" (NaN ignored)
print(f"Mean of values in column 'A': {df['A'].mean()}")

# Sum of values in row 3
print(f"Sum of values in row 3: {df.loc[3].sum()}")
```

DataFrame:

	A	B	C	D
0	1.0	NaN	11	16.0
1	2.0	NaN	12	NaN
2	NaN	NaN	13	18.0
3	4.0	NaN	14	19.0
4	NaN	NaN	15	20.0

Sum of values in column 'A': 7.0

Mean of values in column 'A': 2.3333333333333335

Sum of values in row 3: 37.0

1. **ডেটাস্ট্রাকচার তৈরি:** data_dict ডিকশনারির মাধ্যমে ডেটাস্ট্রাকচার তৈরি করা হয়েছে, যেখানে কিছু মান np.nan দিয়ে অনুপস্থিত (missing) রাখা হয়েছে।
2. **কলাম "A"-এর মানের যোগফল:** df['A'].sum() ব্যবহার করে কলাম "A"-এর সমস্ত মানের যোগফল বের করা হয়েছে। এখানে np.nan মানগুলোকে 0 হিসেবে গণনা হয়।
3. **কলাম "A"-এর গড়:** df['A'].mean() ব্যবহার করে কলাম "A"-এর গড় বের করা হয়েছে। এখানে np.nan মানগুলো উপেক্ষা করা হয় (অর্থাৎ, গড় বের করার সময় np.nan গণনায় আসেনা)। $(1+2+4)/3$
4. **রো ৩-এর মানের যোগফল:** df.loc[3].sum() ব্যবহার করে রো ৩-এর সমস্ত মানের যোগফল বের করা হয়েছে। $4+14+19 = 37$

একটি pandas ডেটাস্ট্রাকচার তৈরি করে এবং তারপর কিছু NAN ডেটা পরিষ্কার (cleaning) করার কাজ:

```
import numpy as np
import pandas as pd

# Define the data dictionary
data_dict = {
    'A': [1, 2, np.nan, 4, np.nan],
    'B': [np.nan, np.nan, np.nan, np.nan, np.nan],
    'C': [11, 12, 13, 14, 15],
    'D': [16, np.nan, 18, 19, 20]
}
df = pd.DataFrame(data_dict)
print(f"DataFrame:\n{df}\n")

# Drop rows with any missing values and display
print(f"DataFrame after dropping rows with missing values:\n{df.dropna(axis=0)}\n")
```

```
# Drop columns with any missing values and display
print(f"DataFrame after dropping columns with missing
values:\n{df.dropna(axis=1)}\n")
```

```
# Display the original DataFrame again
print(f"Original DataFrame:\n{df}")
```

```
DataFrame:
   A    B    C    D
0  1.0 NaN  11  16.0
1  2.0 NaN  12   NaN
2  NaN NaN  13  18.0
3  4.0 NaN  14  19.0
4  NaN NaN  15  20.0
```

```
DataFrame after dropping
rows with missing values:
Empty DataFrame
Columns: [A, B, C, D]
Index: []
```

```
DataFrame after dropping
columns with missing
values:
```

```
   C
0  11
1  12
2  13
3  14
4  15
```

```
Original DataFrame:
   A    B    C    D
0  1.0 NaN  11  16.0
1  2.0 NaN  12   NaN
2  NaN NaN  13  18.0
3  4.0 NaN  14  19.0
4  NaN NaN  15  20.0
```

1. **ডেটাফ্রেম তৈরি:** data_dict নামের ডিকশনারি থেকে একটি pandas ডেটাফ্রেম (df) তৈরি করা হয়েছে, যেখানে কিছু মান np.nan দিয়ে অনুপস্থিত (missing) রাখা হয়েছে।
2. **dropna(axis=0):** এই ফাংশনটি ডেটাফ্রেমের যে **রো-গুলোতে (rows)** NaN মান রয়েছে, সেগুলো মুছে ফেলে নতুন ডেটাফ্রেম প্রদর্শন করবে।
3. **dropna(axis=1):** এই ফাংশনটি ডেটাফ্রেমের যে **কলামগুলোতে (columns)** NaN মান রয়েছে, সেগুলো মুছে ফেলে নতুন ডেটাফ্রেম প্রদর্শন করবে।
4. **মূল ডেটাফ্রেম পুনরায় প্রদর্শন:** শেষের print(f"Original DataFrame:\n{df}") স্টেটমেন্টটি আবার মূল ডেটাফ্রেম দেখাবে, যেটি অবিকল থাকবে (যেহেতু dropna() ফাংশনগুলি নতুন কপি তৈরি করে, আসল ডেটাফ্রেম পরিবর্তন হয় না)।

এইভাবে, কোডটি ডেটাফ্রেমের missing (অনুপস্থিত) ডেটা মুছে ফেলতে সাহায্য করে এবং বিভিন্ন অপশনে কীভাবে কাজ করে তা দেখায়।

একটি pandas ডেটাফ্রেম তৈরি করা হয়েছে যেখানে কিছু missing (NaN) ভ্যালু রয়েছে। এরপর বিভিন্ন পদ্ধতি ব্যবহার করে সেই missing ভ্যালুগুলি পূর্ণ করা হয়েছে।

```
import numpy as np
import pandas as pd

# Define the data dictionary
data_dict = {
    'A': [1, 2, np.nan, 4, np.nan],
    'B': [np.nan, np.nan, np.nan, np.nan, np.nan],
    'C': [11, 12, 13, 14, 15],
```

```

'D': [16, np.nan, 18, 19, 20]
}
df = pd.DataFrame(data_dict)
print(f"DataFrame:\n{df}\n")

# Drop columns with less than 3 non-NaN values
print(f"After dropping columns with less than 3 non-NaN
values:\n{df.dropna(thresh=3, axis=1)}\n")

# Fill missing values with 'MINHAZ'
print(f"After filling missing values with
'MINHAZ':\n{df.fillna('MINHAZ')}\n")
# Fill missing values with the mean of each column
print(f"After filling missing values with column
means:\n{df.fillna(df.mean())}\n")

# Forward fill missing values
print(f"After forward filling missing values
(ffill):\n{df.fillna(method='ffill')}\n")
# Backward fill missing values
print(f"After backward filling missing values
(bfill):\n{df.fillna(method='bfill')}")

```

DataFrame:

	A	B	C	D
0	1.0	NaN	11	16.0
1	2.0	NaN	12	NaN
2	NaN	NaN	13	18.0
3	4.0	NaN	14	19.0
4	NaN	NaN	15	20.0

After dropping columns with less than 3 non-NaN values:

	A	C	D
0	1.0	11	16.0
1	2.0	12	NaN
2	NaN	13	18.0
3	4.0	14	19.0
4	NaN	15	20.0

After filling missing values with 'MINHAZ':

	A	B	C	D
0	1.0	MINHAZ	11	16.0
1	2.0	MINHAZ	12	MINHAZ
2	MINHAZ	MINHAZ	13	18.0
3	4.0	MINHAZ	14	19.0
4	MINHAZ	MINHAZ	15	20.0

After filling missing values with column means:

	A	B	C	D
0	1.000000	NaN	11	16.00

1. **ডেটাফ্রেম তৈরি:** data_dict ব্যবহার করে একটি ডেটাফ্রেম তৈরি করা হয়েছে, যেখানে কিছু NaN মান রয়েছে।

যে মান নাই, সেটা NaN আসবে।

2. **Drop columns with less than 3 non-NaN values:** dropna(thresh=3, axis=1) ব্যবহার করে এমন কলামগুলি বাদ দেওয়া হয়েছে যেগুলিতে ৩টির কম non-NaN মান ছিল।

3. **Fill missing values with 'MINHAZ':** fillna('MINHAZ') ব্যবহার করে সমস্ত NaN মানকে 'MINHAZ' দিয়ে পূর্ণ করা হয়েছে।

4. **Fill missing values with the mean of each column:** fillna(df.mean()) ব্যবহার করে প্রতিটি

1	2.000000	NaN	12	18.25
2	2.333333	NaN	13	18.00
3	4.000000	NaN	14	19.00
4	2.333333	NaN	15	20.00

After forward filling missing values (ffill):

	A	B	C	D
0	1.0	NaN	11	16.0
1	2.0	NaN	12	16.0
2	2.0	NaN	13	18.0
3	4.0	NaN	14	19.0
4	4.0	NaN	15	20.0

After backward filling missing values (bfill):

	A	B	C	D
0	1.0	NaN	11	16.0
1	2.0	NaN	12	18.0
2	4.0	NaN	13	18.0
3	4.0	NaN	14	19.0
4	NaN	NaN	15	20.0

কলামের NaN মানকে ওই কলামের গড় (mean) দিয়ে পূর্ণ করা হয়েছে।

5. Forward fill missing values:

fillna(method='ffill') ব্যবহার করে আগের valid মানটি NaN এর পরে পূর্ণ করা হয়েছে।

Forward fill এ উপরের row থেকে মান নিবে

6. Backward fill missing values:

fillna(method='bfill') ব্যবহার করে পরবর্তী valid মানটি NaN এর আগে পূর্ণ করা হয়েছে।

backward fill এ নিচের row থেকে মান নিবে।

Implace = true লিখলেই data তে স্থায়ীভাবে মান ঢুকবে। permanent change

```
import numpy as np
import pandas as pd

# Create a dictionary with data
data = {
    'A': [1, 2, np.nan, 4, np.nan],
    'B': [np.nan, np.nan, np.nan, np.nan, np.nan],
    'C': [11, 12, 13, 14, 15],
    'D': [16, np.nan, 18, 19, 20]
}

# Create a DataFrame from the dictionary
df = pd.DataFrame(data)

# Fill missing values with 'MINHAZ' and show the result
print(f"Fill missing values with 'MINHAZ':\n{df.fillna('MINHAZ')}\n")

# Fill missing values with 0 and show the result
print(f"Fill missing values with 0:\n{df.fillna(0)}\n")

# Show the original DataFrame
print(f"Original DataFrame:\n{df}\n")
```

```
# Fill missing values with 0 in the original DataFrame and show the result
df.fillna(0, inplace=True)
print(f"DataFrame after filling missing values with 0 in-place:\n{df}\n")
```

Fill missing values with 'MINHAZ':

	A	B	C
D			
0	1.0	MINHAZ	11
16.0			
1	2.0	MINHAZ	12
MINHAZ			
2	MINHAZ	MINHAZ	13
18.0			
3	4.0	MINHAZ	14
19.0			
4	MINHAZ	MINHAZ	15
20.0			

Fill missing values with 0:

	A	B	C	D
0	1.0	0.0	11	16.0
1	2.0	0.0	12	0.0
2	0.0	0.0	13	18.0
3	4.0	0.0	14	19.0
4	0.0	0.0	15	20.0

Original DataFrame:

	A	B	C	D
0	1.0	NaN	11	16.0
1	2.0	NaN	12	NaN
2	NaN	NaN	13	18.0
3	4.0	NaN	14	19.0
4	NaN	NaN	15	20.0

DataFrame after filling missing values with 0 in-place:

	A	B	C	D
0	1.0	0.0	11	16.0
1	2.0	0.0	12	0.0
2	0.0	0.0	13	18.0
3	4.0	0.0	14	19.0
4	0.0	0.0	15	20.0

1. ডেটা ডিকশনারি তৈরি:

একটি ডিকশনারি (data) তৈরি করা হয়েছে, যেখানে কিছু কলামের মধ্যে মিসিং মান (np.nan) রয়েছে।

2. DataFrame তৈরি:

pd.DataFrame(data) ব্যবহার করে ডিকশনারি থেকে একটি DataFrame তৈরি করা হয়েছে।

3. Missing values পূর্ণ করা:

MINHAZ দিয়ে পূর্ণ করা: .fillna('MINHAZ')

ব্যবহার করে, যেখানে মিসিং ভ্যালু আছে, সেগুলো 'MINHAZ' দিয়ে পূর্ণ করা হয়।

0 দিয়ে পূর্ণ করা: .fillna(0) ব্যবহার করে, মিসিং ভ্যালুগুলি 0 দিয়ে পূর্ণ করা হয়।

4. অরিজিনাল DataFrame দেখানো:

মূল DataFrameটি দেখানো হয়েছে, যেখানে মিসিং ভ্যালু এখনও আছে।

5. In-place পূর্ণকরণ:

.fillna(0, inplace=True) ব্যবহার করে, মূল DataFrame-এ মিসিং ভ্যালু গুলি 0 দিয়ে সরাসরি পূর্ণ করা হয়েছে (এটা পরিবর্তন করে DataFrame-এ ফিরে আসে)।

এই কোডটি মূলত দেখাচ্ছে কিভাবে বিভিন্ন মান দিয়ে মিসিং ভ্যালু পূর্ণ করা যায় এবং কিভাবে in-place পরিবর্তন করা যায়।