Hands on exercise on Pandas:

### \*\*Exercise 5: Handling Missing Values\*\*

*'''1. Create a DataFrame with missing values:*

*```python*

*data = {*

*"Name": ["Amit", "Neha", "Raj", "Priya"],*

*"Age": [28, None, 35, 29],*

*"City": ["Delhi", "Mumbai", None, "Chennai"]*

*}*

*df = pd.DataFrame(data)*

*'''*

import pandas as pd

data = {

"Name": ["Amit", "Neha", "Raj", "Priya"],

"Age": [28, None, 35, 29],

"City": ["Delhi", "Mumbai", None, "Chennai"]

}

df = pd.DataFrame(data)

#2. Fill missing values in the `"Age"` column with the average age.

df['Age'].fillna(df['Age'].mean(), inplace=True)

print(df)

#3. Drop rows where any column has missing data.

df.dropna(inplace=True)

print(df)

### \*\*Exercise 6: Adding and Removing Columns\*\*

#1. Add a new column `"Salary"` with the following values: `[50000, 60000, 70000, 65000]`.

df['Salary'] = [50000, 60000, 70000, 65000]

#2. Remove the `"City"` column from the DataFrame.

df\_dropped = df.drop(columns=['City'], inplace=True)

print(df\_dropped)

### \*\*Exercise 7: Sorting Data\*\*

#1. Sort the DataFrame by `"Age"` in ascending order.

sorted\_df = df.sort\_values(by="Age", ascending=True)

print(sorted\_df)

#2. Sort the DataFrame first by `"City"` and then by `"Age"` in descending order.

df\_sorted\_by\_city\_age = df.sort\_values(by=['City', 'Age'], ascending=[True, False])

print(df\_sorted\_by\_city\_age)

### \*\*Exercise 8: Grouping and Aggregation\*\*

#1. Group the DataFrame by `"City"` and calculate the average `"Age"` for each city.

df\_grouped\_by\_city = df.groupby('City')['Age'].mean()

print(df\_grouped\_by\_city)

#2. Group the DataFrame by `"City"` and `"Age"`, and count the number of occurrences for each group.

df\_grouped\_by\_city\_age = df.groupby(['City', 'Age']).size()

print(df\_grouped\_by\_city\_age)

### \*\*Exercise 9: Merging DataFrames\*\*

'''1. Create two DataFrames:A

```python

df1 = pd.DataFrame({

"Name": ["Amit", "Neha", "Raj"],

"Department": ["HR", "IT", "Finance"]

})

df2 = pd.DataFrame({

"Name": ["Neha", "Raj", "Priya"],

"Salary": [60000, 70000, 65000]

})

```

'''

df1 = pd.DataFrame({

"Name": ["Amit", "Neha", "Raj"],

"Department": ["HR", "IT", "Finance"]

})

df2 = pd.DataFrame({

"Name": ["Neha", "Raj", "Priya"],

"Salary": [60000, 70000, 65000]

})

#2. Merge `df1` and `df2` on the `"Name"` column using an inner join.

df\_inner\_join = pd.merge(df1, df2, on="Name", how="inner")

print(df\_inner\_join)

#3. Merge the same DataFrames using a left join.

df\_left\_join = pd.merge(df1, df2, on="Name", how="left")

print(df\_left\_join)