### \*\*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)

```

data = {

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

       "Age": [28, None, 35, 29],

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

   }

df = pd.DataFrame(data)

print(df)

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

df['Age']=df['Age'].fillna(value=df['Age'].mean())

print(df)

3. Drop rows where any column has missing data.

df.dropna(inplace=True,axis=0)

print(df)

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

1. Add a new column `"Salary"` with the following values: `[50000, 60000, 70000, 65000]`.
2. Remove the `"City"` column from the DataFrame.

df["Salary"]=[50000, 60000, 70000]

df=df.drop(columns="City",axis=1)

print(df)

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

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

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

df["city"]=["Delhi", "Mumbai", "Chennai"]

df.sort\_values(by="Age",inplace=True)

df.sort\_values(by=["Age","city"],ascending=[False,False],inplace=True)

print(df)

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

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

grouped\_df = df.groupby('city')['Age'].mean()

print(grouped\_df)

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

grouped\_df = df.groupby(['city','Age']).size().reset\_index(name='Count')

print(grouped\_df)

### \*\*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]

})

```

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

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

print(merged\_df\_inner)

Merge the same DataFrames using a left join.

merged\_df\_inner = pd.merge(df1, df1, on="Name", how="left")

print(merged\_df\_inner)