Amit Divekar | Assignment 1: Set A

DataFrame Operations - Set A

In [1]:

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

Q1. Create a dictionary that stores the mobile names list as value for 'Mobiles' key and their price list as value for 'Price' key. Create DataFrame from this dictionary.

In [2]:

```
data = {
    'Mobiles': ['Samsung', 'iPhone', 'OnePlus', 'Redmi'],
    'Price': [20000, 70000, 40000, 15000]
}

df_mobiles = pd.DataFrame(data)

print("Mobile Names and Prices DataFrame:")
print(df_mobiles)
```

```
Mobile Names and Prices DataFrame:
   Mobiles  Price
0  Samsung  20000
1   iPhone  70000
2  OnePlus  40000
3    Redmi  15000
```

Q2. Create a data.csv file containing employee name, designation, salary of an employee. Display the file as DataFrame.

In [3]:

```
df_employees = pd.read_csv('data.csv')

print("Employee DataFrame from CSV:")
print(df_employees)
```

```
Employee DataFrame from CSV:
   Name Designation  Salary
0  Amit     Manager   45000
1  Ravi       Clerk   20000
2  Neha          HR   38000
```

Q3. Write a python program to calculate sum, mean, median, mode, Standard deviation of following values: CA : [87, 89, 98, 94, 78, 77]

In [4]:

```
CA = np.array([87, 89, 98, 94, 78, 77])

total = np.sum(CA)
mean = np.mean(CA)
median = np.median(CA)

values, counts = np.unique(CA, return_counts=True)
mode = values[counts == counts.max()]

std_dev = np.std(CA, ddof=1)

print("Sum =", total)
print("Mean =", mean)
print("Median =", median)
print("Mode =", mode)
print("Standard Deviation =", std_dev)
```

```
Sum = 523
Mean = 87.16666666666667
Median = 88.0
Mode = [77 78 87 89 94 98]
Standard Deviation = 8.424171571535487
```

Q4. Write a python program to fill missing values (NaN) in a DataFrame using fillna() with mean value and detect duplicates using duplicated() method.

In [5]:

```
data = {
    'Marks': [85, 90, np.nan, 75, 90, np.nan],
    'Age': [20, 21, 22, np.nan, 21, 22]
}

df_missing = pd.DataFrame(data)

print("Original DataFrame")
print(df_missing)
```

```
Original DataFrame
   Marks   Age
0   85.0  20.0
1   90.0  21.0
2    NaN  22.0
3   75.0   NaN
4   90.0  21.0
5    NaN  22.0
```

In [6]:

```
df_filled = df_missing.fillna(df_missing.mean())
```

```
print("\nDataFrame after filling NaN with mean")
print(df_filled)
```

```
DataFrame after filling NaN with mean
   Marks   Age
0   85.0  20.0
1   90.0  21.0
2   85.0  22.0
3   75.0  21.2
4   90.0  21.0
5   85.0  22.0
```

In [7]:

```
duplicates = df_filled.duplicated()

print("\nDuplicate Rows")
print(duplicates)
```

```
Duplicate Rows
0    False
1    False
2    False
3    False
4     True
5     True
dtype: bool
```

Q5. Create a DataFrame from a dictionary and display the first 5 rows.

In [8]:

```
data = {
    'Name': ['Amit', 'Neha', 'Rahul', 'Priya', 'Kiran', 'Sneha', 'Om'],
    'Age': [20, 21, 22, 23, 24, 25, 26],
    'Marks': [85, 89, 78, 88, 92, 80, 95]
}

df_students = pd.DataFrame(data)

print(df_students.head())
```

```
    Name  Age  Marks
0   Amit   20     85
1   Neha   21     89
2  Rahul   22     78
3  Priya   23     88
4  Kiran   24     92
```

Q6. Load sales.csv file into a DataFrame

In [9]:

```
df_sales = pd.read_csv('sales.csv')

print("Original Sales DataFrame:")
print(df_sales)
```

```
Original Sales DataFrame:
  customer_name   product  quantity  price region
0          Amit    Laptop         1  50000  North
1         Sneha     Mouse         5    500  South
2         Rahul  Keyboard         2   1500  North
3         Pooja   Monitor         1  15000   East
4         Kiran       USB        10    200  North
```

a) Check its shape

In [10]:

```
print("\na) Shape of DataFrame:")
print(df_sales.shape)
```

```
a) Shape of DataFrame:
(5, 5)
```

b) Add a new column total_amount = price * quantity

In [11]:

```
df_sales['total_amount'] = df_sales['price'] * df_sales['quantity']

print("\nb) DataFrame with Total Amount")
print(df_sales)
```

```
b) DataFrame with Total Amount
  customer_name   product  quantity  price region  total_amount
0          Amit    Laptop         1  50000  North         50000
1         Sneha     Mouse         5    500  South          2500
2         Rahul  Keyboard         2   1500  North          3000
3         Pooja   Monitor         1  15000   East         15000
4         Kiran       USB        10    200  North          2000
```

c) Select only the columns: customer_name, product, quantity, price

In [12]:

```
selected_df = df_sales[['customer_name', 'product', 'quantity', 'price']]

print("\nc) Selected Columns")
print(selected_df)
```

```
c) Selected Columns
  customer_name   product  quantity  price
0          Amit    Laptop         1  50000
1         Sneha     Mouse         5    500
2         Rahul  Keyboard         2   1500
3         Pooja   Monitor         1  15000
4         Kiran       USB        10    200
```

d) Filter rows where the region is "North"

In [13]:

```
north_region = df_sales[df_sales['region'] == "North"]

print("\nd) Rows where Region is North")
print(north_region)
```

```
d) Rows where Region is North
  customer_name   product  quantity  price region  total_amount
0          Amit    Laptop         1  50000  North         50000
2         Rahul  Keyboard         2   1500  North          3000
4         Kiran       USB        10    200  North          2000
```