EXPERIMENT 2

Aim:

To perform data manipulation and analysis using Python pandas, including selection, filtering, updating, and aggregation of data.

To demonstrate handling missing values, sorting, and basic statistical operations on datasets.

Procedure:

- Create a DataFrame from a dictionary and display it using print() or head().
- Access specific columns, rows, or subsets using indexing, loc, and conditional filtering.
- Add, update, or delete columns and rows in the DataFrame.
- Group data using groupby() and perform aggregation functions like mean, max, or sum.
- Handle missing values using fillna() or dropna(), and sort data with sort_values().

Code:

In [2]:
```
import pandas as pd
data={
   'Name':['Alice','Bob','Charlie','David'],
   'Age':[24,27,22,32],
   'City':['New York','Los Angeles','Chicago','Houston']
}
df=pd.DataFrame(data)
print(df)
```

```
Name  Age       City 0    Alice   24     New York

1         Bob   27  Los Angeles
2         Charlie   22     Chicago
3         David   32     Houston
```

In [3]:
```
print(df.head(2))
```

```
Name  Age       City 0  Alice   24     New York

1   Bob   27  Los Angeles
```

In [4]:
```
print(df.columns)
```

```
Index(['Name', 'Age', 'City'], dtype='object')
```

In [5]:
```
print(df.shape)
```

```
(4, 3)
```

In [6]:
```
print(df['Name'])
```

```
0      Alice
1        Bob
2    Charlie
3      David
Name: Name, dtype: object
```

In [7]:
```
print(df[['Name','City']])
```

```
      Name      City
0    Alice  New York
1      Bob  Los Angeles
2  Charlie   Chicago
3    David   Houston
```

In [8]:
```
print(df.loc[df['Name']=='Charlie'])
```

```
      Name  Age     City
2  Charlie   22  Chicago
```

In [9]:
```
print(df[df['Age']>25])
```

```
      Name  Age      City
1      Bob   27  Los Angeles
3    David   32   Houston
```

In [10]:
```
print(df[(df['City']=='Chicago') | (df['City']=='Houston')] )
```

```
      Name  Age     City
2  Charlie   22  Chicago
3    David   32  Houston
```

In [11]:
```
print(df[df['Age'].between(23,30)])
```

```
      Name  Age      City
0    Alice   24  New York
1      Bob   27  Los Angeles
```

In [12]:
```
df['Score']=[85,90,88,95] print(df)
```

```
      Name  Age      City  Score
0    Alice   24  New York     85
1      Bob   27  Los Angeles     90
2  Charlie   22     Chicago     88
```

3 David 32 Houston 95 In [13]:

```
df.loc[df['Name']=='Bob','Age']=28 print(df)
```

Name Age City Score 0 Alice 24 New York 85

1 Bob 28 Los Angeles 90
2 Charlie 22 Chicago 88
3 David 32 Houston 95 In [14]:

```
df=df.drop('City',axis=1) print(df)
```

Name Age Score 0 Alice 24 85

1 Bob 28 90
2 Charlie 22 88
3 David 32 95 In [15]:

```
df=df[df['Name'] != 'David']
print(df)
```

In [16]:

Name Age Score 0 Alice 24 85

1 Bob 28 90
2 Charlie 22 88 In [17]:

```
data={
  'Department':['HR','IT','HR','IT'],    'Salary':[30000,50000,35000,55000],
  'Experience':[2,5,3,6]
}
df=pd.DataFrame(data) print(df.groupby('Department')['Salary'].mean())
```

Department

HR 32500.0

IT 52500.0

Name: Salary, dtype: float64

In [18]:

```
print(df.groupby('Department')['Experience'].max())
```

Department

HR 3

IT 6

Name: Experience, dtype: int64

In [19]:
```
print(df['Salary'].sum())
```

170000 In [20]:

```
data = {
'Student': ['John', 'Emma', 'Sam', 'Olivia'],
'Marks': [80, None, 75, 90]
}
df=pd.DataFrame(data) print(df.fillna(0))
```

Student Marks

0 John 80.0

1 Emma 0.0

2 Sam 75.0

3 Olivia 90.0

In [21]:
```
print(df.dropna())
```

Student Marks

0 John 80.0

2 Sam 75.0

3 Olivia 90.0

In [22]:
```
print(df.sort_values(by='Marks',ascending=False))
```

Student Marks

3 Olivia 90.0

0 John 80.0

2 Sam 75.0

1 Emma NaNIn [23]:

```
data = {
'Product': ['Laptop', 'Tablet', 'Smartphone', 'Monitor', 'Keyboard'],
'Price': [70000, 30000, 25000, 15000, 2000],
'Stock': [10, 25, 50, 15, 100]
}
df = pd.DataFrame(data) print(df.loc[[0,2]])
```

Product Price Stock 0 Laptop 70000 10

2 Smartphone 25000 50 In [24]:

```
data = {
'Subject': ['Math', 'Science', 'English'],
'Marks': [88, 92, 85]
}
df=pd.DataFrame(data,index=['Student1','Student2','Student3'])
print(df.loc['Student2'])
```

```
Subject    Science
Marks         92
Name: Student2, dtype: object
```

Result:

Thus the python program to perform different pandas operation are written and executed successfully and the output got verified.