**Hierarchial Indexing combining and merging datasets reshape**

**24.Data aggregation and group operation in pandas**

import pandas as pd

import numpy as np

# Sample dataset

data = pd.DataFrame({

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

    'Employee': ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Frank', 'Grace'],

    'Salary': [50000, 55000, 60000, 65000, 70000, 75000, 80000],

    'Experience': [4, 5, 6, 7, 7, 8, 9]})

print("Original Data:\n", data)

# Grouping by 'Department' and aggregating salary statistics

grouped\_data = data.groupby('Department').agg({

    'Salary': ['sum', 'mean', 'max', 'min'],

    'Experience': ['mean', 'count']})

print("\nAggregated Data:\n", grouped\_data)

# Using groupby() with custom functions

def salary\_range(s):

   return s.max() - s.min()

custom\_grouped = data.groupby('Department').agg({'Salary': salary\_range})

print("\nCustom Aggregation - Salary Range:\n", custom\_grouped)

# Filtering groups (Selecting departments with mean salary > 60000)

filtered\_groups = data.groupby('Department').filter(lambda x: x['Salary'].mean() > 60000)

print("\nFiltered Groups (Mean Salary > 60000):\n", filtered\_groups)

# Using transform() to broadcast aggregated values

data['Avg\_Salary\_Department'] = data.groupby('Department')['Salary'].transform('mean')

print("\nData with Department Avg Salary:\n", data)

**Output**: Original Data:

   Department Employee  Salary  Experience

0         HR    Alice   50000           4

1    Finance      Bob   55000           5

2         HR  Charlie   60000           6

3         IT    David   65000           7

4    Finance      Eve   70000           7

5         IT    Frank   75000           8

6         HR    Grace   80000           9

Aggregated Data:

             Salary                             Experience

               sum          mean    max    min       mean count

Department

Finance     125000  62500.000000  70000  55000   6.000000     2

HR          190000  63333.333333  80000  50000   6.333333     3

IT          140000  70000.000000  75000  65000   7.500000     2

Custom Aggregation - Salary Range:

             Salary

Department

Finance      15000

HR           30000

IT           10000

Filtered Groups (Mean Salary > 60000):

   Department Employee  Salary  Experience

0         HR    Alice   50000           4

1    Finance      Bob   55000           5

2         HR  Charlie   60000           6

3         IT    David   65000           7

4    Finance      Eve   70000           7

5         IT    Frank   75000           8

6         HR    Grace   80000           9

Data with Department Avg Salary:

   Department Employee  Salary  Experience  Avg\_Salary\_Department

0         HR    Alice   50000           4           63333.333333

1    Finance      Bob   55000           5           62500.000000

2         HR  Charlie   60000           6           63333.333333

3         IT    David   65000           7           70000.000000

4    Finance      Eve   70000           7           62500.000000

5         IT    Frank   75000           8           70000.000000

6         HR    Grace   80000           9           63333.333333

**25.Perform statistical analysis and visualizalization**

import pandas as pd

import numpy as np

# Sample dataset

data = pd.Series([11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24])

# 1. Statistical Analysis

mean\_value = data.mean()

median\_value = data.median()

std\_dev\_value = data.std()

print(f"Mean: {mean\_value}")

print(f"Median: {median\_value}")

print(f"Standard Deviation: {std\_dev\_value}")

# 2. Histogram (Using Pandas' built-in function)

histogram\_data = data.value\_counts().sort\_index()

print("\nHistogram (Frequency of Values):\n", histogram\_data)

# 3. Box Plot Statistics (Using NumPy)

Q1 = data.quantile(0.25)  # First Quartile

Q3 = data.quantile(0.75)  # Third Quartile

IQR = Q3 - Q1  # Interquartile Range

min\_val = data.min()

max\_val = data.max()

print("\nBox Plot Statistics:")

print(f"Q1 (25th Percentile): {Q1}")

print(f"Median (50th Percentile): {median\_value}")

print(f"Q3 (75th Percentile): {Q3}")

print(f"IQR (Interquartile Range): {IQR}")

print(f"Min Value: {min\_val}")

print(f"Max Value: {max\_val}

**Output:**

Mean: 17.5

Median: 17.5

Standard Deviation: 4.183300132670378

Histogram (Frequency of Values):

 11    1

12    1

13    1

14    1

15    1

16    1

17    1

18    1

19    1

20    1

21    1

22    1

23    1

24    1

Name: count, dtype: int64

Box Plot Statistics:

Q1 (25th Percentile): 14.25

Median (50th Percentile): 17.5

Q3 (75th Percentile): 20.75

IQR (Interquartile Range): 6.5

Min Value: 11

Max Value: 24