

ETDS Practical:1

```
In [1]: import pandas as pd
```

```
In [2]: #Step 1 :Create Sample dataset
data_aamna={

    'Region':["North","South","East","West","North","South","East","West"],
    'Product': ['A', 'B', 'A', 'B', 'C', 'C', 'B', 'A'],
    'Sales': [150, 200, 300, 400, 250, 180, 220, 310],
    'Quantity': [10, 15, 20, 25, 12, 14, 16, 18],
}
```

```
In [3]: df=pd.DataFrame(data_aamna)
print("Sample Dataset:\n",df)
```

Sample Dataset:

	Region	Product	Sales	Quantity
0	North	A	150	10
1	South	B	200	15
2	East	A	300	20
3	West	B	400	25
4	North	C	250	12
5	South	C	180	14
6	East	B	220	16
7	West	A	310	18

```
In [6]: #Step 2 Grouping and aggregation
#Aggregating Sales by Region(Sum Aggregation)
sales_by_region_aamna=df.groupby("Region")["Sales"].sum()
print("\nTotal Sales by Region:\n",sales_by_region_aamna)
```

Total Sales by Region:

Region	Sales
East	520
North	400
South	380
West	710

Name: Sales, dtype: int64

```
In [7]: #Aggregating Sales and Qunatity by Product(Meanb Aggregation)
mean_by_product_aamna=df.groupby("Product")[["Sales","Quantity"]].mean()
print("\nMean Sales and Quantity by Product:\n", mean_by_product_aamna)
```

Mean Sales and Quantity by Product:

Product	Sales	Quantity
A	253.333333	16.000000
B	273.333333	18.666667
C	215.000000	13.000000

```
In [8]: #Aggregating Count of Sales by Region(Count Aggregation)
count_by_region_aamna=df.groupby("Region")["Sales"].count()
print("\nCount of Sales Records by Region:\n",count_by_region_aamna)
```

```
Count of Sales Records by Region:
Region
East      2
North     2
South     2
West      2
Name: Sales, dtype: int64
```

```
In [9]: #Custom Aggregation :Calculate Min and Max sales by Region
custom_aggregation_aamna=df.groupby("Region")["Sales"].agg(["min","max"])
print("\nCustom Aggregation(Min and Max Sales by Region):\n",custom_aggregation_aamna)
```

```
\nCustom Aggregation(Min and Max Sales by Region):
      min   max
Region
East    220   300
North   150   250
South   180   200
West    310   400
```

```
In [16]: #Step 3:Multilevel Aggregation
#Aggregating Sales by Region and Product
multi_level_agg_aamna=df.groupby(["Region","Product"])["Sales"].sum()
print("\nSales by Region and Product\n",multi_level_agg_aamna)
```

```
Sales by Region and Product
Region  Product
      A        300
          B        220
      North   A        150
          C        250
      South   B        200
          C        180
      West    A        310
          B        400
Name: Sales, dtype: int64
```

```
In [36]: # Step 4: Reset Index for Multi-Level Aggregation
multi_level_agg_reset_aamna = multi_level_agg_aamna.reset_index()
print("\nSales by Region and Product (Reset Index):\n", multi_level_agg_reset_aamna)
```

Sales by Region and Product (Reset Index):

	Region	Product	Sales
0	East	A	300
1	East	B	220
2	North	A	150
3	North	C	250
4	South	B	200
5	South	C	180
6	West	A	310
7	West	B	400

Objective:

To understand and implement:

Time Aggregation: Aggregating data over different time periods (e.g., monthly, yearly).

Spatial Aggregation: Aggregating data by spatial attributes (e.g., by region, city).

```
In [18]: import pandas as pd

# Step 1: Extend Dataset with Time and Spatial Data
data_aamna_g = {
    'Region': ['North', 'South', 'East', 'West', 'North', 'South', 'East', 'West'],
    'City': ['City1', 'City2', 'City3', 'City4', 'City1', 'City2', 'City3', 'City4'],
    'Product': ['A', 'B', 'A', 'B', 'C', 'C', 'B', 'A'],
    'Sales': [150, 200, 300, 400, 250, 180, 220, 310],
    'Quantity': [10, 15, 20, 25, 12, 14, 16, 18],
    'Date': pd.to_datetime(['2024-01-01', '2024-01-02', '2024-02-01', '2024-02-03',
                           '2024-03-01', '2024-03-02', '2024-04-01', '2024-04-03'])
}
```

```
In [22]: df=pd.DataFrame(data_aamna_g)
print("Extended Dataset:\n",df)
```

Extended Dataset:

	Region	City	Product	Sales	Quantity	Date
0	North	City1	A	150	10	2024-01-01
1	South	City2	B	200	15	2024-01-02
2	East	City3	A	300	20	2024-02-01
3	West	City4	B	400	25	2024-02-03
4	North	City1	C	250	12	2024-03-01
5	South	City2	C	180	14	2024-03-02
6	East	City3	B	220	16	2024-04-01
7	West	City4	A	310	18	2024-04-03

```
In [23]: # -----
# Time Aggregation
# -----

# Step 2: Set Date Column as Index (optional)
df.set_index('Date', inplace=True)
```

```
In [26]: #Aggregating Sales By Month
monthly_sales_aamna=df.resample("M")["Sales"].sum(0)
print("\nTotal Sales by Month:\n", monthly_sales_aamna)
```

Total Sales by Month:

Date	Sales
2024-01-31	350
2024-02-29	700
2024-03-31	430
2024-04-30	530

Freq: M, Name: Sales, dtype: int64

```
In [28]: # Aggregating Sales by Quarter
quarterly_sales_aamna = df.resample('Q')['Sales'].sum()
print("\nTotal Sales by Quarter:\n", quarterly_sales_aamna)
```

Total Sales by Quarter:

Date	Sales
2024-03-31	1480
2024-06-30	530

Freq: Q-DEC, Name: Sales, dtype: int64

```
In [30]: # Aggregating Sales by Year
yearly_sales_aamna = df.resample('Y')['Sales'].sum()
print("\nTotal Sales by Year:\n", yearly_sales_aamna)
```

Total Sales by Year:

Date	Sales
2024-12-31	2010

Freq: A-DEC, Name: Sales, dtype: int64

```
In [31]: #Reset index to Restore Original Structure
df.reset_index(inplace=True)
```

In [32]: # -----

```
# Spatial Aggregation
```

```
# -----
```

Step 3: Aggregating Sales by Region

```
sales_by_region_aamna=df.groupby("Region")["Sales"].sum()
print("\nTotal Sales by Region:\n",sales_by_region_aamna)
```

Total Sales by Region:

Region

Region	Sales
East	520
North	400
South	380
West	710

Name: Sales, dtype: int64

In [33]: *#Aggregating Sales by City*

```
sales_by_city_aamna=df.groupby("City")["Sales"].sum()
print("\nTotal Sales by City :\n",sales_by_city_aamna)
```

Total Sales by City :

City

City	Sales
City1	400
City2	380
City3	520
City4	710

Name: Sales, dtype: int64

In [34]:

Aggregating Sales by Region and City

```
sales_by_region_city_aamna = df.groupby(['Region', 'City'])['Sales'].sum()
print("\nTotal Sales by Region and City:\n", sales_by_region_city_aamna)
```

Total Sales by Region and City:

Region	City	Sales
East	City3	520
North	City1	400
South	City2	380
West	City4	710

Name: Sales, dtype: int64

In [35]: *# Step 4: Export Spatial Aggregation Results*

```
sales_by_region_city_reset_aamna = sales_by_region_city_aamna.reset_index()
```

```
sales_by_region_city_reset_aamna.to_csv("spatial_aggregation.csv", index=False)
print("\nSpatial aggregation data saved to 'spatial_aggregation.csv'")
```

Spatial aggregation data saved to 'spatial_aggregation.csv'

In []: