Experiment No. 12

Aim : Data Aggregation.

Problem Statement: Analyzing Sales Performance by Region in a Retail Company and perform the following tasks – 1. Import the dataset. 2. Explore the dataset. 3. Identify the relevant variables for aggregating sales data. 4. Group the sales data by region. 5. Create bar plots or pie charts. 6. Identify the top-performing regions. 7. Group the sales data, and 8. Create stacked bar plots or grouped bar plots.

Dataset: "Retail_Sales_Data.csv"

Software Requirements : Python and Jupyter Notebook.

Hardware Requirements : 8GB RAM, Storage and Processor.

Objective : i) Analyze sales performance by region. ii) Visualize the sales distribution. iii) Identify key product categories. iv) Compare regional sales trends.

Theory : Data Aggregation

Data aggregation refers to the process of collecting and summarizing data from various sources to provide a comprehensive view or to perform analysis. This can be done in several ways, depending on the type of data and the goals of the analysis.

Aggregation Techniques

Aggregation techniques are used to combine data to make it more manageable and insightful. These techniques can be categorized into single-level and multilevel aggregation. Here's an overview of both:

Single-Level Aggregation

Single-level aggregation involves summarizing data at one specific level or granularity. This technique is straightforward and typically used when you want to analyze data from a single perspective or dimension.

Multilevel Aggregation

Multilevel aggregation involves summarizing data across multiple levels of granularity or dimensions. This technique is useful for analyzing data in a more nuanced way by examining different hierarchies or layers.

Key concepts of Data Aggregation

1.Grouping

Grouping is the process of organizing data into categories or groups based on shared attributes or dimensions. To simplify data analysis by aggregating records that share common characteristics.

2.Summarization

Summarization involves calculating aggregate metrics that provide a concise view of the data. To reduce the complexity of data by providing summary statistics that highlight key information.

3.Hierarchical Aggregation

Hierarchical aggregation refers to summarizing data at various levels of a hierarchy or dimensions, often used in multidimensional data models. To provide insights at different levels of granularity, allowing for detailed analysis from a high-level overview to granular details.

4.Visualization

Visualization involves representing aggregated data in graphical formats to make it easier to interpret and analyze. To provide a visual representation of data that highlights patterns, trends, and insights that might be less obvious in raw data or summary tables.

Implantation

Step No.1 - Import the Dataset.

```
import pandas as pd

import numpy as np

df = pd.read_csv(r"C:\Users\saira\Downloads\Retail_Sales_Data (1).csv")

df.head()
```

	Transaction Date	Region	Product Category	Quantity Sold	Sales Amount	Customer Name	Transaction ID	Payment Method
0	2019-01-16	West	Home Decor	9	909.84	Melinda Pham	7b094307-bcd3-4f16-84a7-2bca783fff4f	Credit Card

1 2021-09-17 North Clothing 8 900.29 Shelly Perez fb437a2e-4ebf-4807-b84e-f2dfae83541a Credit Card

2 2020-03-27 East Electronics 3 506.07 Scott White b6ead965-ed1c-4bdc-95ac-864685467abd Online Banking

3 2019-02-11 South Clothing 9 744.70 Gloria Williams 400773f4-a820-47b6-b3c4-2cc2a5467e73 Cash

4 2020-01-15 East Books 4 245.55 Michael Sims 10b62e7a-38f8-4f27-a989-b99b55d76223 Cash

df.tail()

Transaction Date Region Product Category Quantity Sold Sales Amount Customer Name Transaction ID Payment Method

95 2020-06-26 East Electronics 3 914.06 Erica Franklin DVM 56855833-4f68-4312-b5e0-88c7dc7ce72b Online Banking

96 2021-07-04 South Electronics 3 652.93 Ricky Walsh 7e03f607-9b90-4075-b05f-e704c81fb165 PayPal

97 2020-04-08 North Books 9 640.88 Luis Wong 4f1f1533-6fb0-468d-80ac-a8c1db865b1a Online Banking

98 2021-12-24 East Electronics 1 727.21 Christopher Reese 0c3d09c3-469c-4b2f-860e-89365bab7f88 Online Banking

99 2020-10-04 South Clothing 4 554.22 Barry Johnson 95921b52-e166-4d8f-86c6-2149cf1398d8 Credit Card

Step No2. - Explore the Dataset.

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 100 entries, 0 to 99

Data columns (total 8 columns):

Column Non-Null Count Dtype

--- ------ -------------- -----

0 Transaction Date 100 non-null object

1 Region 100 non-null object

2 Product Category 100 non-null object

3 Quantity Sold 100 non-null int64

4 Sales Amount 100 non-null float64

5 Customer Name 100 non-null object

6 Transaction ID 100 non-null object

7 Payment Method 100 non-null object

dtypes: float64(1), int64(1), object(6)

memory usage: 6.4+ KB

df.describe()

Quantity Sold Sales Amount

count 100.000000 100.000000

mean 5.700000 544.873300

std 2.904194 276.530738

min 1.000000 23.140000

25% 3.000000 336.295000

50% 5.500000 554.715000

75% 8.000000 781.757500

max 10.000000 984.850000

df.shape

(100, 8)

df.columns

```
Index(['Transaction Date', 'Region', 'Product Category', 'Quantity Sold',
       'Sales Amount', 'Customer Name', 'Transaction ID', 'Payment Method'],
      dtype='object')
df.dtypes
Transaction Date     object
Region               object
Product Category     object
Quantity Sold         int64
Sales Amount        float64
Customer Name        object
Transaction ID       object
Payment Method       object
dtype: object
```

Step No.3 - Identify relevant variables.

```
df['Region'].unique()
array(['West', 'North', 'East', 'South'], dtype=object)
df['Sales Amount'].describe()
count    100.000000
mean     544.873300
std      276.530738
min       23.140000
25%      336.295000
50%      554.715000
75%      781.757500
```

max 984.850000

Name: Sales Amount, dtype: float64

df['Product Category'].unique()

array(['Home Decor', 'Clothing', 'Electronics', 'Books'], dtype=object)

Step No.4 - Group sales data by region and calculate total sales amount.

sales_by_region = df.groupby('Region')['Sales Amount'].sum().reset_index()

sales_by_region

	Region	Sales Amount
0	East	14382.28
1	North	13031.74
2	South	11300.33
3	West	15772.98

Step No.5 - Create bar plots or pie charts to visualize the sales distribution by region.

import matplotlib.pyplot as plt

Bar plot

plt.figure(figsize=(10, 6))

plt.bar(sales_by_region['Region'], sales_by_region['Sales Amount'])

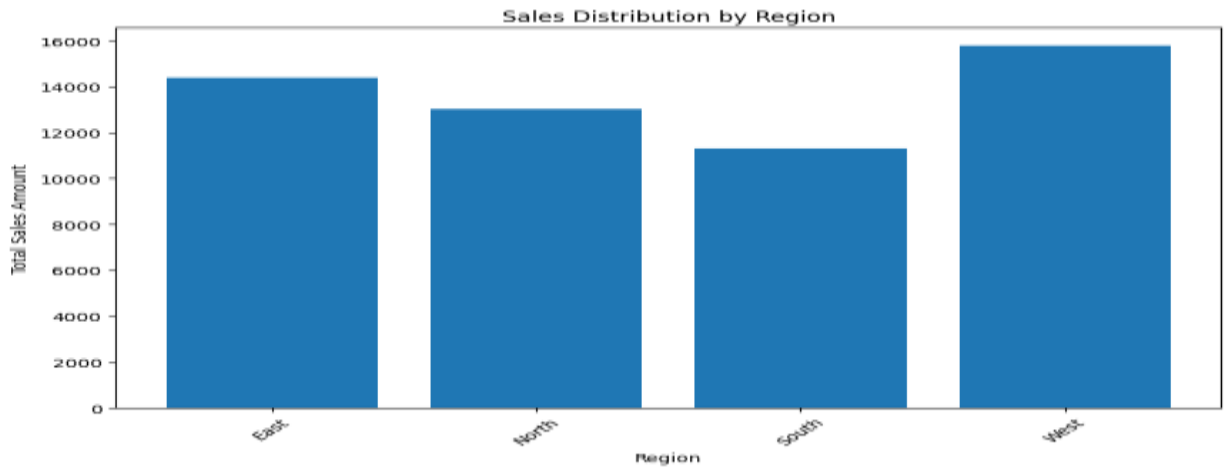
plt.xlabel('Region')

plt.ylabel('Total Sales Amount')

plt.title('Sales Distribution by Region')

plt.xticks(rotation=45)

plt.show()



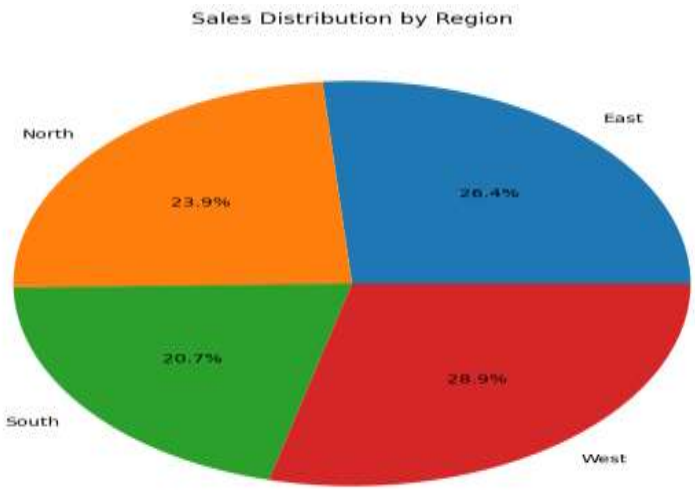
Pie chart

plt.figure(figsize=(8, 8))

plt.pie(sales_by_region['SalesAmount'],labels=sales_by_region['Region'], autopct='%1.1f%%')

plt.title('Sales Distribution by Region')

plt.show()



Step No.6 - Identify top-performing regions.

Sort the regions by sales amount in descending order

top_regions = sales_by_region.sort_values(by='Sales Amount', ascending=False)

top_regions

Region Sales Amount

3 West 15772.98

0 East 14382.28

1 North 13031.74

2 South 11300.33

Step No.7 - Group sales data by region and product category to calculate total sales amount for each combination.

```
sales_by_region_category = df.groupby(['Region', 'Product Category'])['Sales Amount'].sum().unstack().fillna(0)

sales_by_region_category
```

Product Category Books Clothing Electronics Home Decor

Region

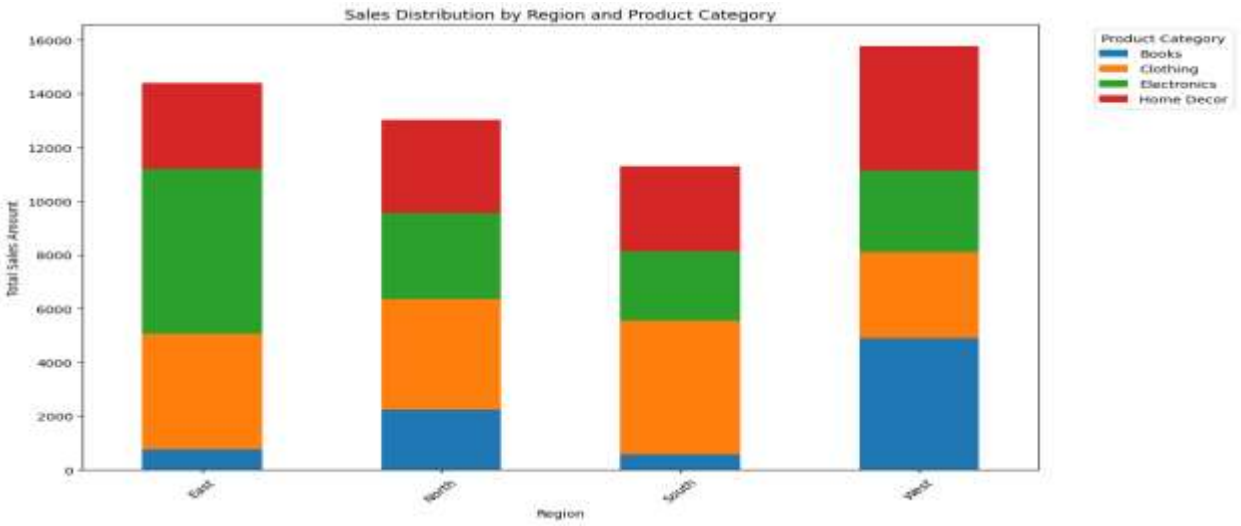
East 759.89 4293.54 6153.44 3175.41

North 2235.48 4121.55 3208.94 3465.77

South 573.38 4977.85 2581.59 3167.51

West 4907.12 3185.51 3043.47 4636.88

```
# Step No.8 - Create stacked bar plots or grouped bar plots.

# Stacked bar plot

sales_by_region_category.plot(kind='bar', stacked=True, figsize=(12, 8))

plt.xlabel('Region')

plt.ylabel('Total Sales Amount')

plt.title('Sales Distribution by Region and Product Category')

plt.legend(title='Product Category', bbox_to_anchor=(1.05, 1), loc='upper left')

plt.xticks(rotation=45)

plt.show()
```



Grouped bar plot

sales_by_region_category.plot(kind='bar', figsize=(12, 8))

plt.xlabel('Region')

plt.ylabel('Total Sales Amount')

plt.title('Sales Distribution by Region and Product Category')

plt.legend(title='Product Category', bbox_to_anchor=(1.05, 1), loc='upper left')

Conlusion : We can successfully analysing sales performance by region in retail company on a “Reatail_sales_data.csv”. This analysis helps in identifying in top performing regions and understanding sales distribution and also strategic decision making.