Name: Gaurang Vaghela
Rollno: BEAD-22561
DMV Practical 11
Problem Statement: Analyzing Sales Performance by Region in a Retail Company

1. Import the "Retail_Sales_Data.csv" dataset.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
```

```
df= pd.read_csv("D:/College/BE/CL-1/DMV Codes and Datasets/DMV New/customer_shopping_data.csv")
df.head()
```

	invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	5/8/2022	Kanyon
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	12/12/2021	Forum Istanbul
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	9/11/2021	Metrocity
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	16/05/2021	Metropol AVM
4	I337046	C189076	Female	53	Books	4	60.60	Cash	24/10/2021	Kanyon

2. Explore the dataset to understand its structure and content.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99457 entries, 0 to 99456
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   invoice_no      99457 non-null  object 
 1   customer_id     99457 non-null  object 
 2   gender          99457 non-null  object 
 3   age             99457 non-null  int64  
 4   category        99457 non-null  object 
 5   quantity        99457 non-null  int64  
 6   price           99457 non-null  float64
 7   payment_method  99457 non-null  object 
 8   invoice_date    99457 non-null  object 
 9   shopping_mall   99457 non-null  object 
dtypes: float64(1), int64(2), object(7)
memory usage: 7.6+ MB
```

```
df.isnull().sum()
```

```
invoice_no        0
customer_id       0
gender            0
age               0
category          0
quantity          0
price             0
payment_method    0
invoice_date      0
shopping_mall     0
dtype: int64
```

3. Group the sales data by region and calculate the total sales amount for each region.

```
# Group sales data by region (shopping mall)
region_sales = df.groupby("shopping_mall")["price"].sum().sort_values(ascending=False)

print("Sales by Shopping Mall:")
print(region_sales)
```

```
Sales by Shopping Mall:
shopping_mall
Mall of Istanbul     13851737.62
Kanyon               13710755.24
Metrocity            10249980.07
Metropol AVM          6937992.99
Istinye Park          6717077.54
Zorlu Center          3509649.02
Cevahir AVM           3433671.84
Viaport Outlet        3414019.46
Emaar Square Mall     3390408.31
Forum Istanbul        3336073.82
Name: price, dtype: float64
```

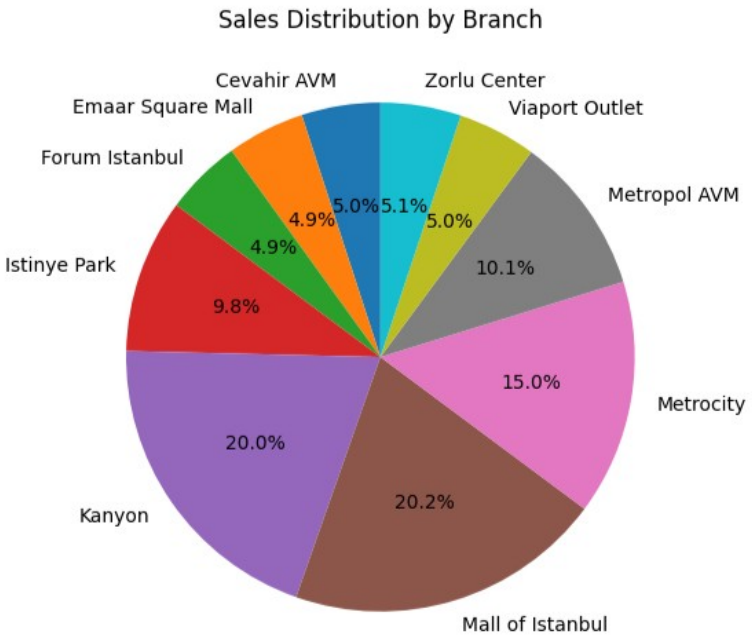
```
# Group sales data by region (shopping mall)
region_sales = df.groupby("category")["price"].sum().sort_values(ascending=False)

print("Sales by Shopping Mall:")
print(region_sales)
```

```
Sales by Shopping Mall:
category
Clothing           31075684.64
Shoes              18135336.89
Technology         15772050.00
Cosmetics           1848606.90
Toys                1086704.64
Food & Beverage      231568.71
Books                226977.30
Souvenir             174436.83
Name: price, dtype: float64
```

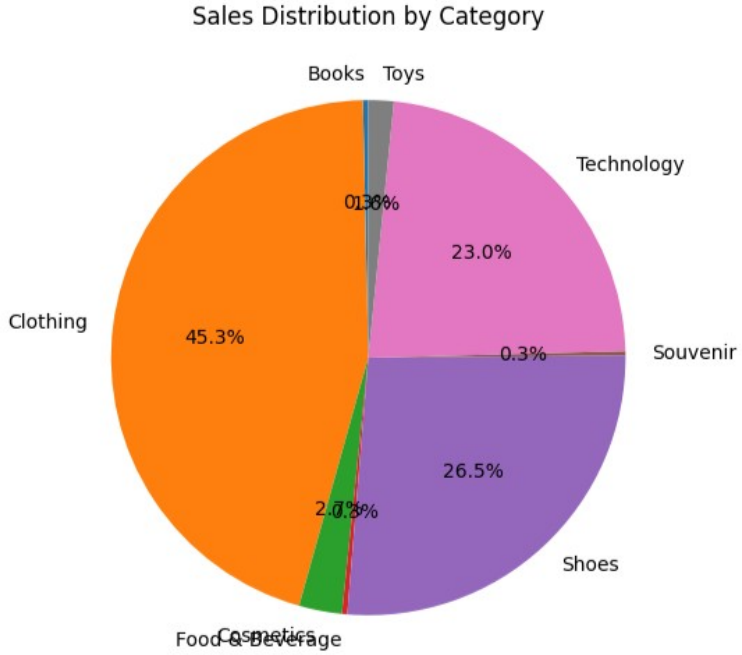
4. Create bar plots or pie charts to visualize the sales distribution by region.

```
# Pie chart for sales by branch
plt.figure(figsize=(6, 6))  # use plt.figure instead of plt.Figure
plt.pie(
    branch_sales["price"],
    labels=branch_sales.index,
    autopct="%1.1f%%",   # shows percentage with 1 decimal place
    startangle=90
)
plt.title("Sales Distribution by Branch")
plt.show()
```



5. Create bar plots or pie charts to visualize the sales distribution by region.

```
# Pie chart for sales by branch
plt.figure(figsize=(6, 6))  # use plt.figure instead of plt.Figure
plt.pie(
    category_sales["price"],
    labels=category_sales.index,
    autopct="%1.1f%%",   # shows percentage with 1 decimal place
    startangle=90
)
plt.title("Sales Distribution by Category")
plt.show()
```



6. Create stacked bar plots or grouped bar plots to compare the sales amounts across different regions and product categories.

```
combined_pivot = df.pivot_table(index="shopping_mall", columns="category", values="price", aggfunc="sum")
# grouped bar chart for sales of different categories at different branches
combined_pivot.plot(kind="bar", figsize=(10, 6))
plt.show()
```