dmv6

November 3, 2024

1 Data Aggregation Problem Statement: Analyzing Sales Performance by Region in a Retail Company. The goal is to perform data aggregation to analyze the sales performance by region and identify the topperforming regions.

[3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[4]:
```
df = pd.read_csv(r"C:\Users\dell\Desktop\DMV and ML\DMV␣
↪Datasets\retail_sales_data.csv")
df.head()
```

[4]:

	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

[5]:
```
# df["invoice_date"]=pd.to_datetime(df["invoice_date"])
```

[6]:
```
df.describe()
```

[6]:

	age	quantity	price
count	99457.000000	99457.000000	99457.000000

```
mean      43.427089      3.003429    689.256321
std       14.990054      1.413025    941.184567
min       18.000000      1.000000      5.230000
25%       30.000000      2.000000     45.450000
50%       43.000000      3.000000    203.300000
75%       56.000000      4.000000   1200.320000
max       69.000000      5.000000   5250.000000
```

[7]:
```
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
```

[8]:
```
df.isna().sum()
```

[8]:
```
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
```

[9]:
```
df.duplicated().sum()
```

[9]:
```
0
```

[10]:
```
df.columns
```

[10]:
```
Index(['invoice_no', 'customer_id', 'gender', 'age', 'category', 'quantity',
       'price', 'payment_method', 'invoice_date', 'shopping_mall'],
      dtype='object')
```

[11]:
```
# df.drop(['invoice_no', 'customer_id', 'gender', 'age',␣
 ↪'payment_method'],axis=1,inplace=True)
# df.head()
```

[12]:
```
df["Sales"]=df["quantity"]* df["price"]
df.head()
```

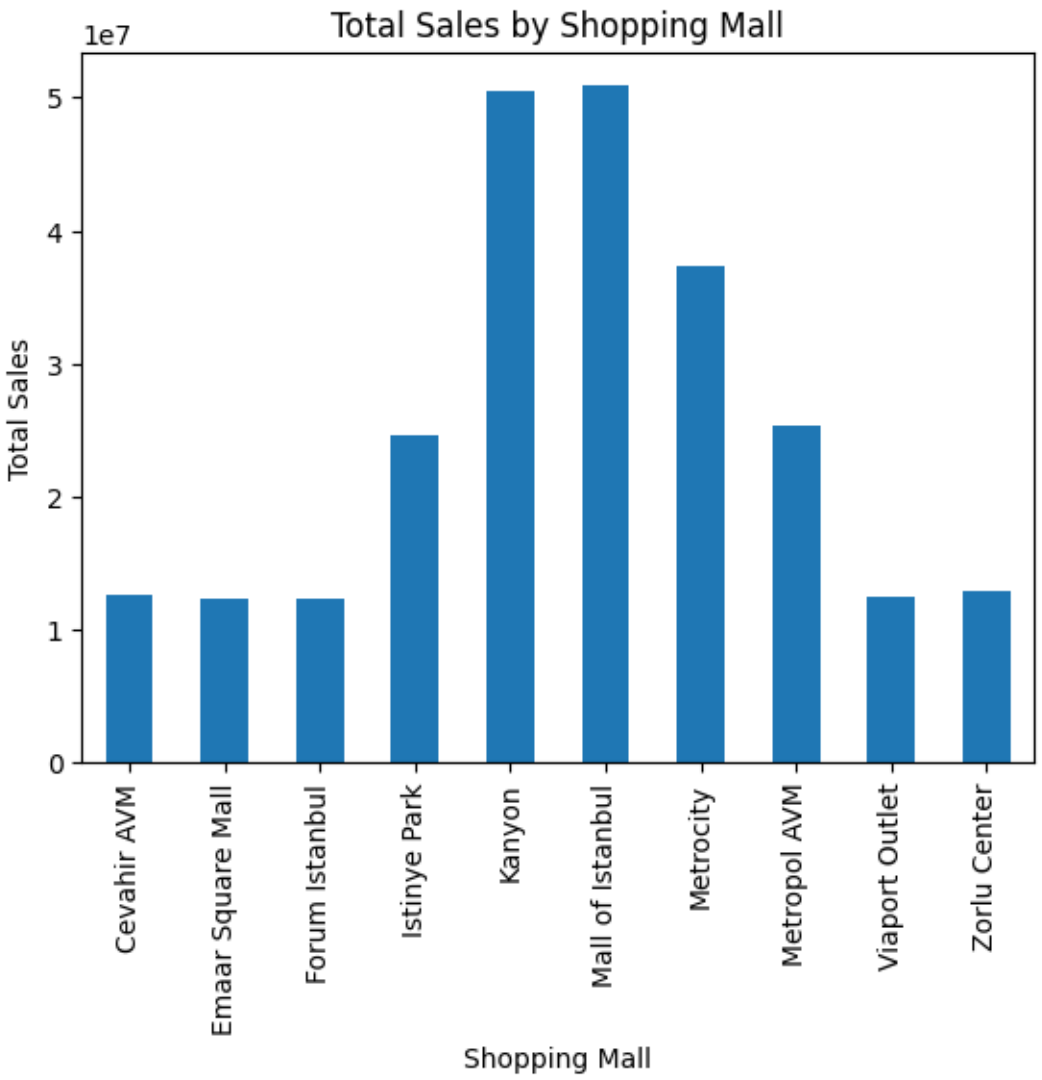
[12]:

	invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall	Sales
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	5/8/2022	Kanyon	7502.00
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	12/12/2021	Forum Istanbul	5401.53
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	9/11/2021	Metrocity	300.08
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	16/05/2021	Metropol AVM	15004.25
4	I337046	C189076	Female	53	Books	4	60.60	Cash	24/10/2021	Kanyon	242.40

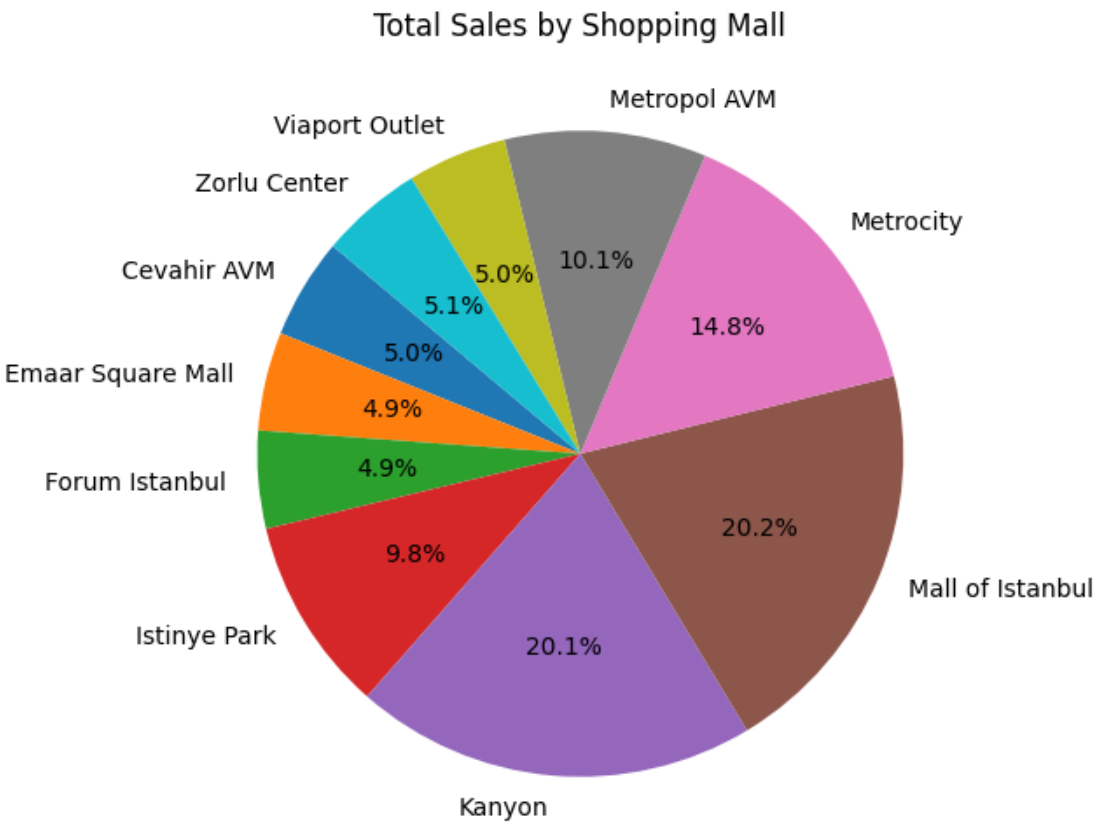
[13]:
```
region_sales= df.groupby("shopping_mall")["Sales"].sum()
region_sales
```

[13]:
```
shopping_mall
Cevahir AVM          12645138.20
Emaar Square Mall    12406100.29
Forum Istanbul       12303921.24
Istinye Park         24618827.68
Kanyon               50554231.10
Mall of Istanbul     50872481.68
Metrocity            37302787.33
Metropol AVM         25379913.19
Viaport Outlet       12521339.72
Zorlu Center         12901053.82
Name: Sales, dtype: float64
```

[14]:
```
region_sales.plot(kind="bar")
plt.title("Total Sales by Shopping Mall")
plt.xlabel("Shopping Mall")
plt.ylabel("Total Sales")
plt.show()
```



[15]:
```
plt.figure(figsize=(6,6))
region_sales.plot(kind='pie', autopct='%1.1f%%', startangle=140)
plt.title("Total Sales by Shopping Mall")
plt.ylabel('')  # Remove the y-label for better aesthetics
plt.show()
```



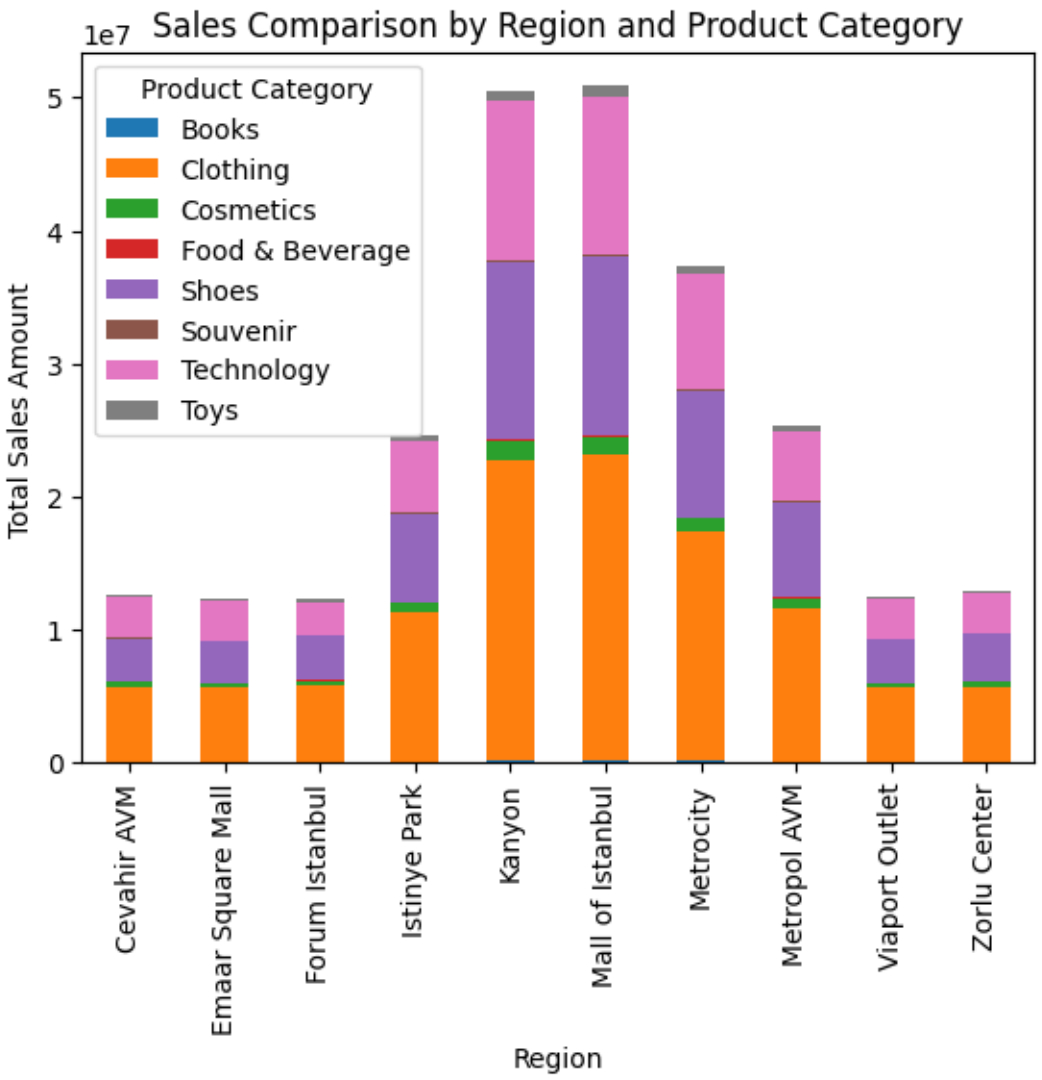
[16]:
```
region_category_sales= df.groupby(["shopping_mall","category"])["Sales"].sum().
↪unstack()
region_category_sales
```

[16]:

shopping_mall	Books	Clothing	Cosmetics	Food & Beverage	Shoes	Souvenir	Technology	Toys
Cevahir AVM	44541.00	5706321.28	321214.00	44010.45				
Emaar Square Mall	41995.80	5590490.40	338941.76	40610.95				
Forum Istanbul	42056.40	5792444.24	353172.76	39162.24				
Istinye Park	76083.30	11253900.24	655357.88	85918.44				
Kanyon	163029.15	22609527.60	1369550.78	166497.05				
Mall of Istanbul	172240.35	22947417.68	1367517.78	171177.90				
Metrocity	125911.65	17226692.56	991860.04	129902.74				
Metropol AVM	83718.90	11568084.00	680770.38	88638.04				
Viaport Outlet	39632.40	5604594.16	347439.70	41662.18				
Zorlu Center	45343.95	5697318.88	367037.82	41955.06				

```
Cevahir AVM          3243918.85   29723.82   3051300.0  204108.80
Emaar Square Mall    3089675.16   30943.74   3094350.0  179092.48
Forum Istanbul       3327942.65   32879.19   2516850.0  199413.76
Istinye Park         6641481.22   68925.48   5436900.0  400261.12
Kanyon              13383190.83  127399.53  11944800.0  790236.16
Mall of Istanbul    13467814.80  127540.29  11828250.0  790522.88
Metrocity            9519296.37   94227.09   8608950.0  605946.88
Metropol AVM         7149825.21   67869.78   5327700.0  413306.88
Viaport Outlet       3194704.91   27319.17   3066000.0  199987.20
Zorlu Center         3535601.47   28996.56   2987250.0  197550.08
```

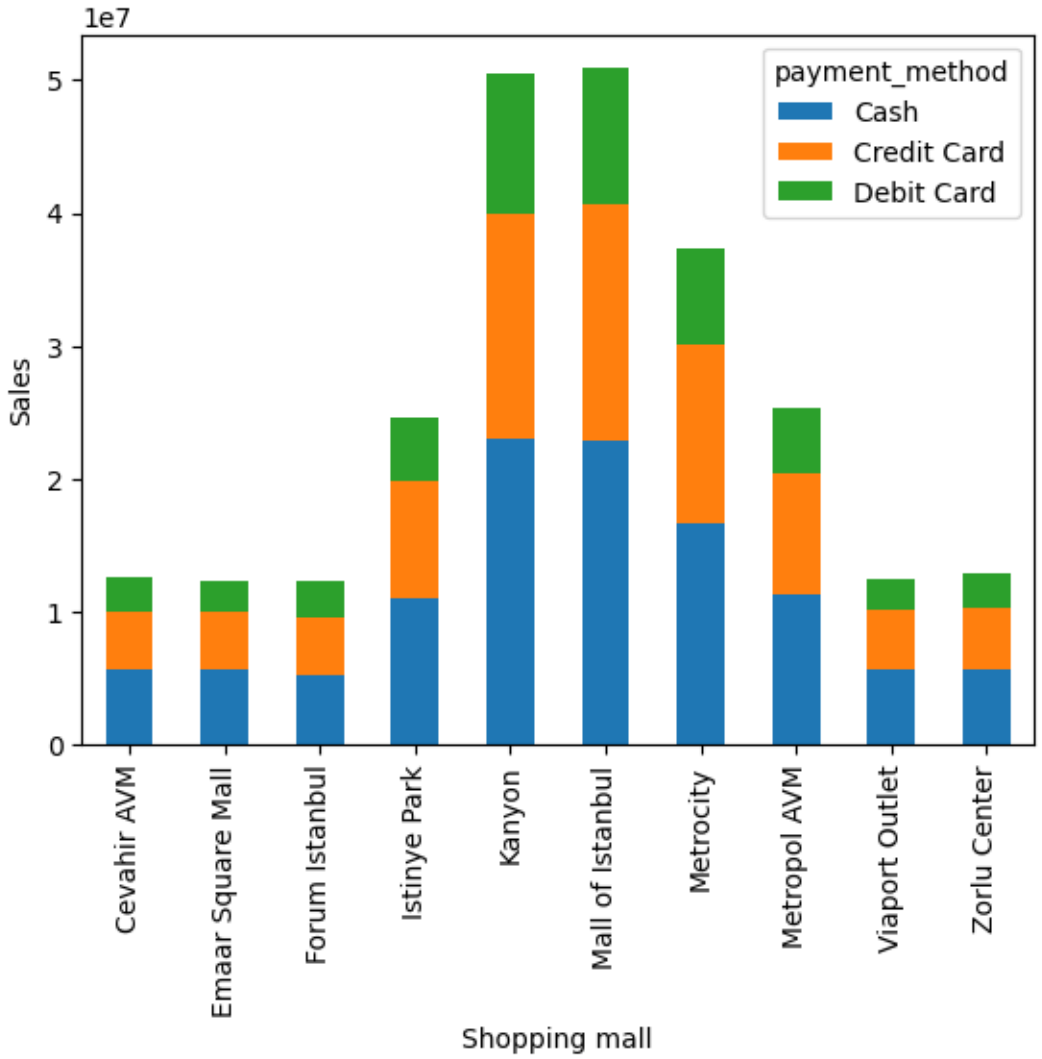
[17]:
```
region_category_sales.plot(kind="bar",stacked=True)
plt.title("Sales Comparison by Region and Product Category")
plt.xlabel("Region")
plt.ylabel("Total Sales Amount")
plt.legend(title="Product Category")
plt.show()
```



[18]:
```
temp = df.groupby(["shopping_mall", "payment_method"])["Sales"].sum().unstack()
plt.figure(figsize=(10,10))
temp.plot(kind='bar', stacked=True)
plt.xlabel("Shopping mall")
plt.ylabel("Sales")
```

[18]:
```
Text(0, 0.5, 'Sales')
```

```
<Figure size 1000x1000 with 0 Axes>
```



[21]:
```
mf = df.groupby("category")["price"].mean()
mf
```

[21]:
```
category
Books                 45.568621
Clothing             901.084021
Cosmetics            122.448626
Food & Beverage       15.671948
Shoes               1807.388568
Souvenir              34.894345
Technology          3156.935548
Toys                 107.733185
Name: price, dtype: float64
```

[22]:
```
df["category"].value_counts()
```

[22]:
```
category
Clothing           34487
Cosmetics          15097
Food & Beverage    14776
Toys               10087
Shoes              10034
Souvenir            4999
Technology          4996
Books               4981
Name: count, dtype: int64
```

[ ]:
```

```