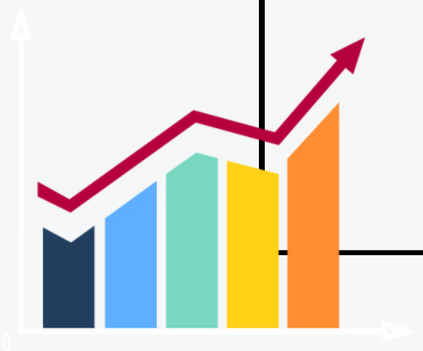
OASIS INFOBYTE

Biswarup Neogi

EXPLORATORY DATA ANALYSIS (EDA) ON RETAIL SALES DATA



ABOUT

Retail sales analysis involves the meticulous examination of sales data within the retail sector to derive valuable insights. Through this process, businesses gain a comprehensive understanding of their sales performance, customer behavior, market trends, and areas for improvement.

retail sales analysis involves gathering and analyzing data from various sources such as point-of-sale systems, customer transactions, online platforms, and market research reports. By leveraging advanced analytics techniques, businesses can uncover patterns in their sales data. This allows them to make their sales strategies,marketing efforts

LEARNING OBJECTIVES

Objective 1

Gain hands-on experience in data cleaning and exploratory data analysis.

Objective 2

Develop skills in interpreting descriptive statistics and time series analysis.

Objective 3

Learn to use data visualization for effective communication of insights.

OASIS INFOBYTE

DATASET DETAILS

Column Name

- invoice_no
- customer_id
- gender
- age
- category
- quantity
- total_price
- payment_method
- Order_Date
- Month
- Year
- shopping_mall

1. IMPORT DATASET & PYTHON LIBRARIES

Import Libraries

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plot
import seaborn as sns
```

Import Dataset

[2]:
```
data = pd.read_csv("F:/data analyst roadmap/OASIS/customer_shopping_data.csv")
```

Read Data

[3]:
```
data.head()
```

[3]:

	invoice_no	customer_id	gender	age	category	quantity	total_price	payment_method	Order Date	Month	Year	shopping_mall
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	5-Sep-19	September	2019	Kanyon
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	5-Sep-19	September	2019	Forum Istanbul
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	17-Jun-21	June	2021	Metrocity
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	15-Jul-21	July	2021	Metropol AVM
4	I337046	C189076	Female	53	Books	4	60.60	Cash	15-Jul-21	July	2021	Kanyon

In [6]:
```
round(data.describe())
```

Out[6]:

	age	quantity	total_price	Year
count	2538.0	2538.0	2538.0	2538.0
mean	44.0	3.0	690.0	2022.0
std	15.0	1.0	977.0	2.0
min	18.0	1.0	5.0	2019.0
25%	30.0	2.0	41.0	2021.0
50%	43.0	3.0	203.0	2021.0
75%	56.0	4.0	1200.0	2023.0
max	69.0	5.0	5250.0	2024.0

In [7]:
```
data.nunique()
```

Out[7]:
```
invoice_no        2538
customer_id       2538
gender               2
age                 52
category             8
quantity             5
total_price         40
payment_method       3
Order Date        1864
Month               12
Year                 6
shopping_mall       10
dtype: int64
```

2. DATA CLEANING

Cleaning The Data

In [9]:
```
data.isnull().sum()
```

Out[9]:
```
invoice_no        0
customer_id       0
gender            0
age               0
category          0
quantity          0
total_price       0
payment_method    0
Order Date        0
Month             0
Year              0
shopping_mall     0
dtype: int64
```

In [10]:
```
data.head()
```

Out[10]:

	invoice_no	customer_id	gender	age	category	quantity	total_price	payment_method	Order Date	Month	Year	shopping_mall
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	5-Sep-19	September	2019	Kanyon
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	5-Sep-19	September	2019	Forum Istanbul
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	17-Jun-21	June	2021	Metrocity
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	15-Jul-21	July	2021	Metropol AVM
4	I337046	C189076	Female	53	Books	4	60.60	Cash	15-Jul-21	July	2021	Kanyon

Delete a Column

In [11]:
```
newdata = data.drop(['Order Date'],axis=1)
newdata.head()
```

Out[11]:

	invoice_no	customer_id	gender	age	category	quantity	total_price	payment_method	Month	Year	shopping_mall
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	September	2019	Kanyon
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	September	2019	Forum Istanbul
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	June	2021	Metrocity
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	July	2021	Metropol AVM
4	I337046	C189076	Female	53	Books	4	60.60	Cash	July	2021	Kanyon

New Updated column

In [12]:
```
newdata.head()
```

Out[12]:

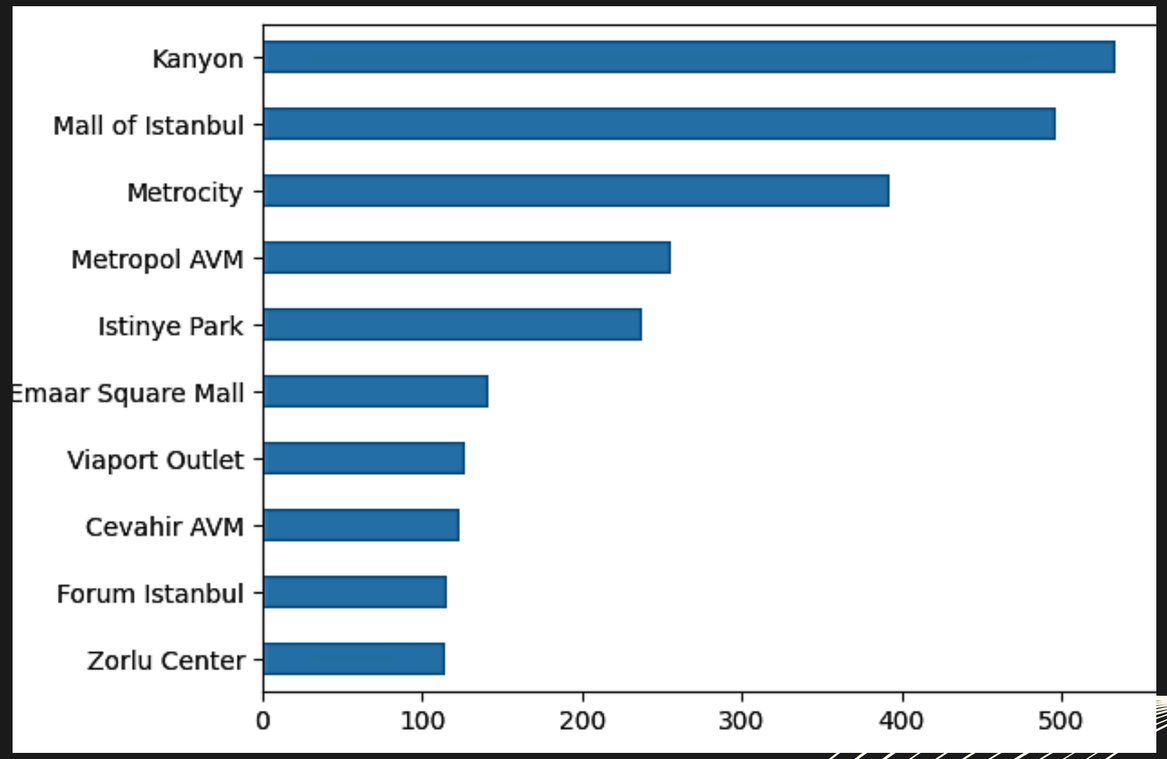
	invoice_no	customer_id	gender	age	category	quantity	total_price	payment_method	Month	Year	shopping_mall
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	September	2019	Kanyon
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	September	2019	Forum Istanbul
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	June	2021	Metrocity
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	July	2021	Metropol AVM
4	I337046	C189076	Female	53	Books	4	60.60	Cash	July	2021	Kanyon

Q1. WHAT IS THE NUMBER OF CUSTOMERS WHO SHOPPED AT EACH MALL?

```
In [28]: mall_customer_count = newdata['shopping_mall'].value_counts().sort_values()
         print(mall_customer_count)
         mall_customer_count.plot(kind='barh')
```

RESULT -->

```
shopping_mall
Kanyon                534
Mall of Istanbul      496
Metrocity             392
Metropol AVM          256
Istinye Park          238
Emaar Square Mall     141
Viaport Outlet        127
Cevahir AVM           124
Forum Istanbul        116
Zorlu Center          114
Name: count, dtype: int64
```



Q2. WHAT IS THE AVERAGE AMOUNT SPENT BY CUSTOMERS AT EACH MALL?

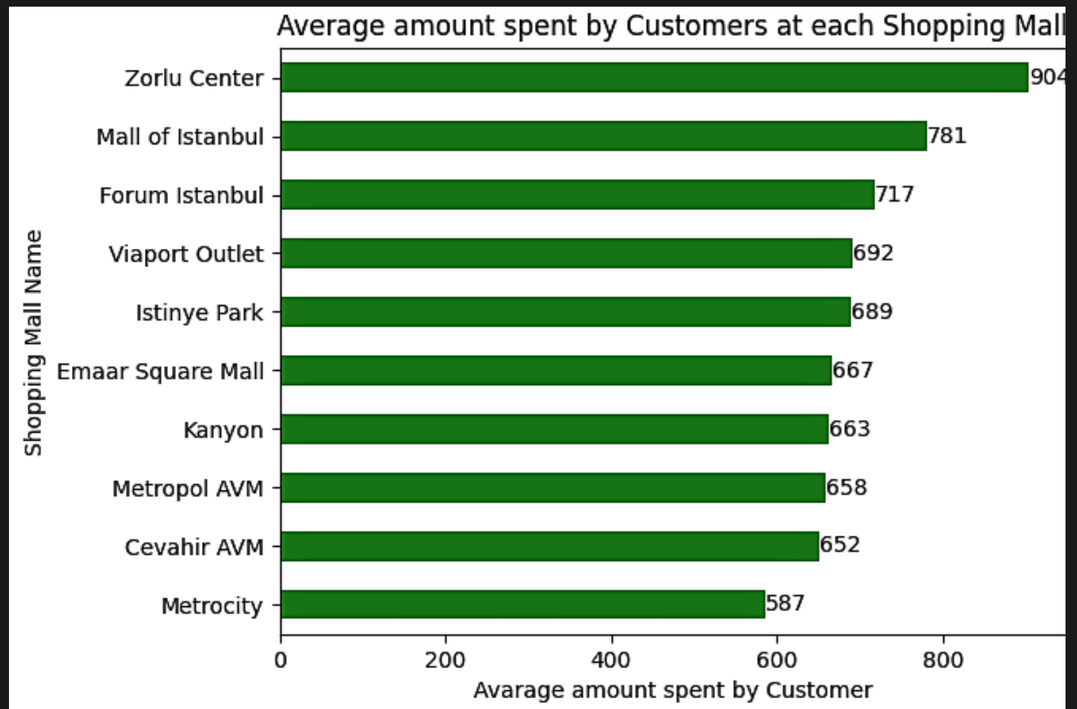
```
mall_avg_amount_spent = newdata.groupby('shopping_mall')['total_price'].mean().sort_values()

ax = mall_avg_amount_spent.plot(kind='barh', color='green')

# show the values inside the bars
for i, v in enumerate(mall_avg_amount_spent):
    ax.text(v + 0.1, i, str(round(v)), va='center')

plot.title('Average amount spent by Customers at each Shopping Mall')
plot.xlabel('Avarage amount spent by Customer')
plot.ylabel('Shopping Mall Name')
plot.show()
```

RESULT -->



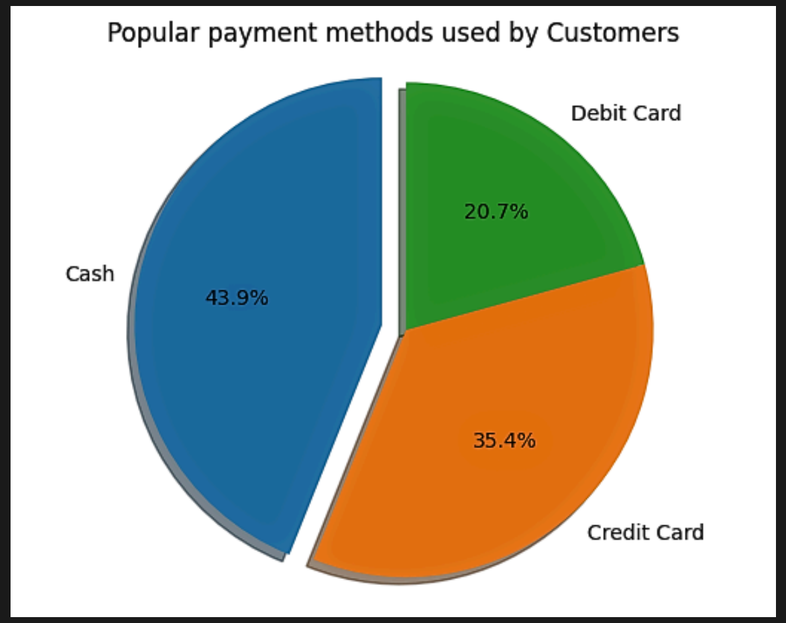
Q3. WHAT ARE THE MOST POPULAR PAYMENT METHODS USED BY CUSTOMERS?

```
popular_payment_method = newdata['payment_method'].value_counts()
print(popular_payment_method)
popular_method = popular_payment_method.idxmax()
print('Popular Payment Method is :',popular_method)


explode = [0.1 if i == popular_method else 0 for i in popular_payment_method.index]
plot.pie(popular_payment_method,labels= popular_payment_method.index, autopct='%1.1f%%', explode=explode, startangle=90,shadow=True)

plot.title('Popular payment methods used by Customers')
plot.axis('equal')
plot.show()
```

RESULT -->



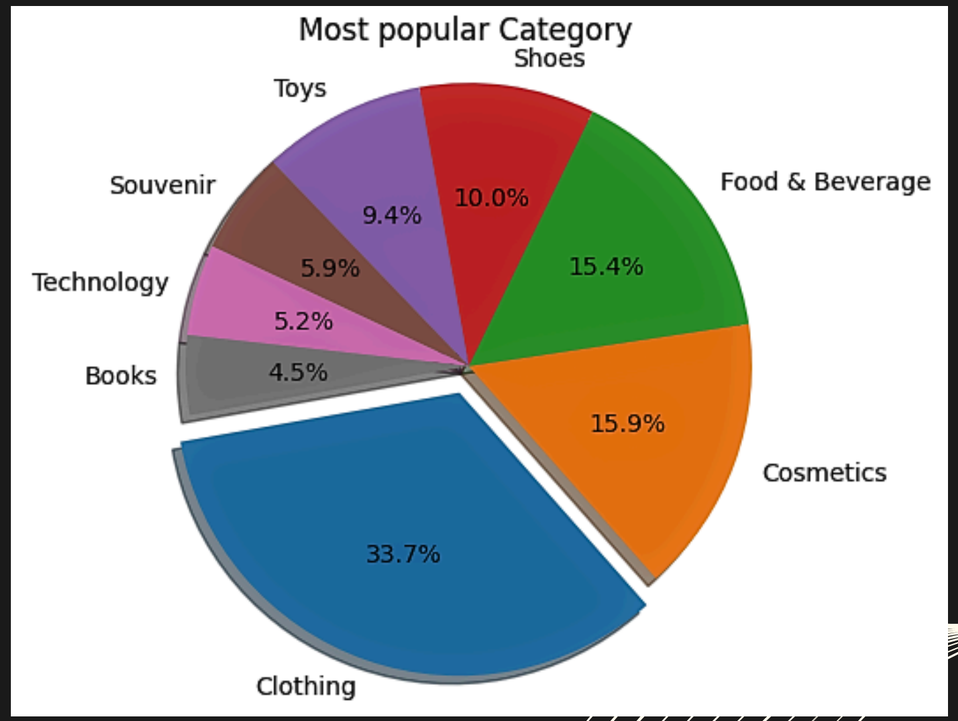
Q4. WHAT IS THE MOST POPULAR CATEGORY OF ITEMS PURCHASED BY CUSTOMERS?

```
popular_category = newdata['category'].value_counts()
print(popular_category)
popular_category_id=popular_category.idxmax()
print('Most popular category of items purchased by customers is : ', popular_category_id)

explode=[0.1 if i == popular_category_id else 0 for i in popular_category.index]
plot.pie(popular_category, labels=popular_category.index, autopct='%1.1f%%' ,explode=explode, shadow=True, startangle=190)
plot.title('Most popular Category')
plot.axis('equal')
plot.show()
```

RESULT -->

```
category
Clothing           856
Cosmetics          403
Food & Beverage    391
Shoes              254
Toys               238
Souvenir           149
Technology         132
Books              115
Name: count, dtype: int64
Most popular category of items purchased by customers is :  Clothing
```



Q5. WHAT IS THE TOTAL AMOUNT SPENT BY A CUSTOMER AT A PARTICULAR MALL?

```
customer_mall_spent = newdata.groupby(['customer_id','shopping_mall'])['total_price'].sum()
print(customer_mall_spent)
```

RESULT -->

```
customer_id  shopping_mall
C100299      Metrocity              15.15
C100322      Viaport Outlet       5250.00
C100484      Metrocity             143.36
C100507      Mall of Istanbul     1500.40
C100684      Kanyon                 35.84
                                    ...
C995929      Mall of Istanbul      121.98
C996668      Emaar Square Mall     121.98
C997380      Metrocity             162.64
C997987      Emaar Square Mall    1200.32
C999586      Zorlu Center          107.52
Name: total_price, Length: 2538, dtype: float64
```

Q6. WHAT IS THE TOTAL AMOUNT OF SALES OF EACH SHOPPING MALL BY THE YEAR?

```
shopping_mall_sales = round(newdata.groupby(['Year','shopping_mall'])['total_price'].sum().unstack(level=0))
print(shopping_mall_sales)
```

RESULT -->

Year	2019	2020	2021	2022	2023	2024
shopping_mall						
Cevahir AVM	11257.0	29295.0	13218.0	16215.0	5700.0	5111.0
Emaar Square Mall	13998.0	6888.0	31069.0	25590.0	6890.0	9578.0
Forum Istanbul	11762.0	3529.0	30454.0	7952.0	20438.0	9090.0
Istinye Park	15766.0	15344.0	43685.0	30262.0	16834.0	42048.0
Kanyon	38225.0	41608.0	98267.0	84003.0	40516.0	51175.0
Mall of Istanbul	34555.0	57823.0	94316.0	73353.0	68153.0	58955.0
Metrocity	18086.0	22147.0	76141.0	44438.0	40152.0	28965.0
Metropol AVM	12217.0	20388.0	47161.0	29055.0	30756.0	28815.0
Viaport Outlet	8860.0	15078.0	24471.0	16660.0	9431.0	13347.0
Zorlu Center	12333.0	14758.0	22644.0	26101.0	7722.0	19457.0

Q7. WHICH MONTH HAS MAXIMUM SALES OF EACH SHOPPING MALL?

```
max_sales_month = newdata.groupby(['shopping_mall','Month'])['total_price'].sum().unstack(level=0)
info = max_sales_month.idxmax()

print("{:<30} {:<30} {:<30}".format("Shopping Mall Name","Maximum Sales Month","Maximum sales Amount"))
for i in info.index :
    max_month = info.loc[i]
    max_sales = max_sales_month.loc[max_month,i]
    print("{:<30} {:<30} {:<30}".format(i,max_month,round(max_sales)))
```

RESULT -->

Shopping Mall Name	Maximum Sales Month	Maximum sales Amount
Cevahir AVM	November	14056
Emaar Square Mall	March	15248
Forum Istanbul	December	14726
Istinye Park	November	18671
Kanyon	May	43551
Mall of Istanbul	September	45570
Metrocity	December	30461
Metropol AVM	August	23902
Viaport Outlet	April	12627
Zorlu Center	February	16280

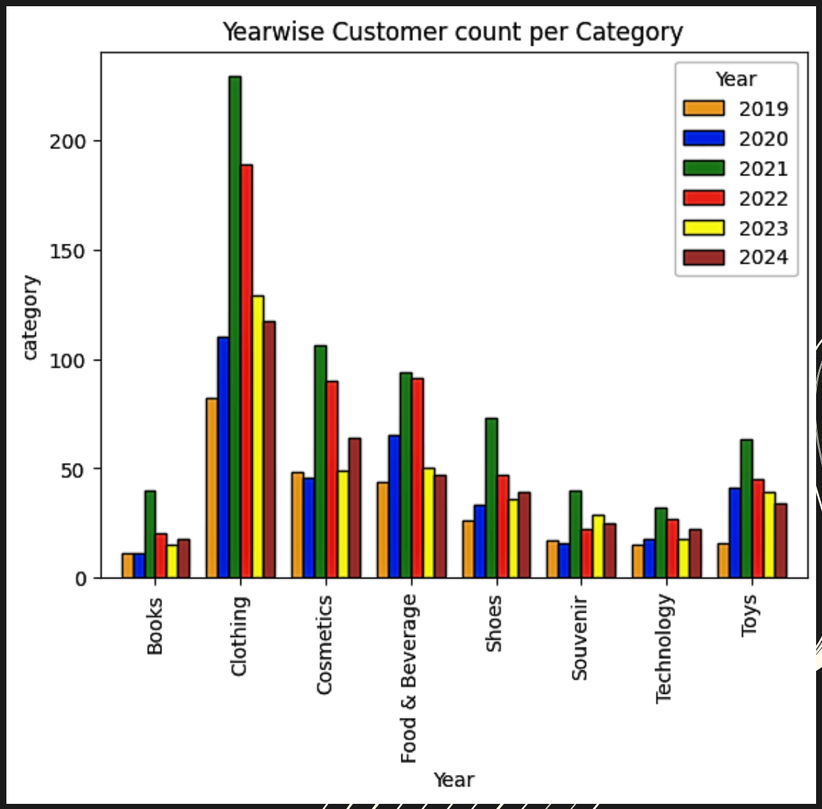
Q8. YEARWISE CUSTOMER COUNTS OF EACH CATEGORY?

```
customer_per_category = newdata.groupby(['Year','category'])['customer_id'].nunique().unstack(level=0)
print(customer_per_category)

customer_per_category.plot(kind='bar' , color=['orange' , 'blue','green','red','yellow','brown'], width=0.8, edgecolor='black' )
plot.title('Yearwise Customer count per Category')
plot.xlabel('Year')
plot.ylabel('category')
plot.xticks(rotation=90)
plot.show()
```

RESULT -->

Year	2019	2020	2021	2022	2023	2024
category						
Books	11	11	40	20	15	18
Clothing	82	110	229	189	129	117
Cosmetics	48	46	106	90	49	64
Food & Beverage	44	65	94	91	50	47
Shoes	26	33	73	47	36	39
Souvenir	17	16	40	22	29	25
Technology	15	18	32	27	18	22
Toys	16	41	63	45	39	34



Q9. WHAT IS THE AVERAGE PRICE OF AN ITEM IN EACH CATEGORY?

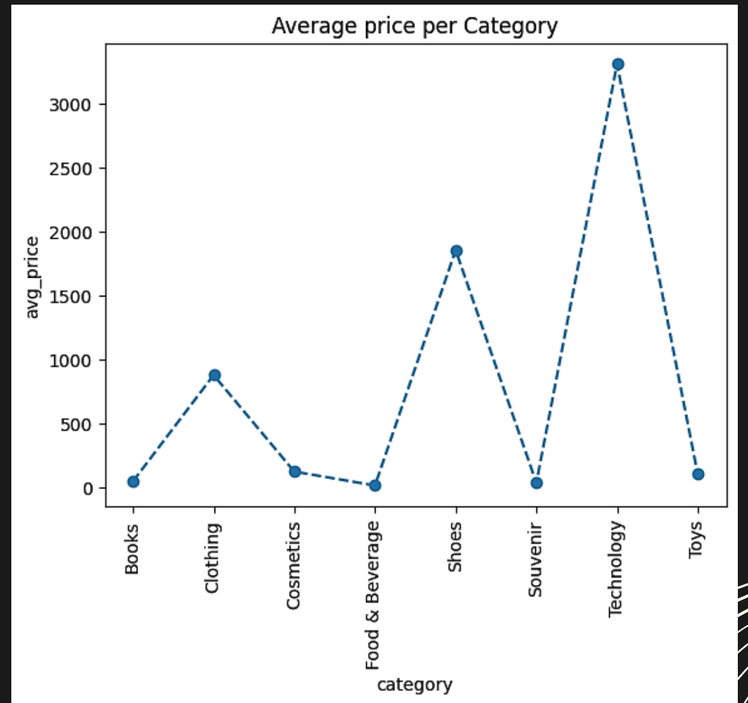
```
avg_price = newdata.groupby(['category'])['total_price'].mean()
print(round(avg_price).sort_values(ascending=False))

avg_price.plot( kind='line', marker='o', linestyle='--')

plot.title('Average price per Category')
plot.xlabel('category')
plot.ylabel('avg_price')
plot.xticks(rotation=90)
plot.show()
```

RESULT -->

```
category
Technology          3309.0
Shoes               1855.0
Clothing             879.0
Cosmetics            124.0
Toys                 105.0
Books                 48.0
Souvenir              36.0
Food & Beverage       16.0
Name: total_price, dtype: float64
```



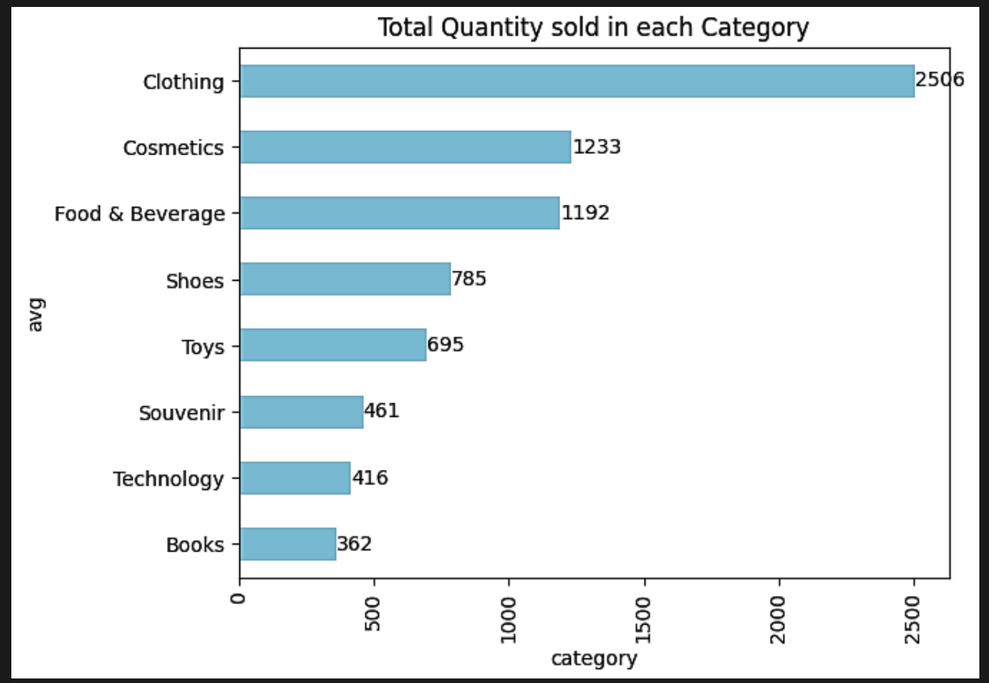
Q10. TOTAL QUANTITY SOLD IN EACH CATEGORY

```
avg = newdata.groupby(['category'])['quantity'].sum().sort_values()

ax = avg.plot(kind = 'barh', color = 'skyblue')
for i,v in enumerate(avg):
    ax.text(v+0.1,i,str(v),va='center')

plot.title('Total Quantity sold in each Category')
plot.xlabel('category')
plot.ylabel('avg')
plot.xticks(rotation=90)
plot.show()
```

RESULT -->



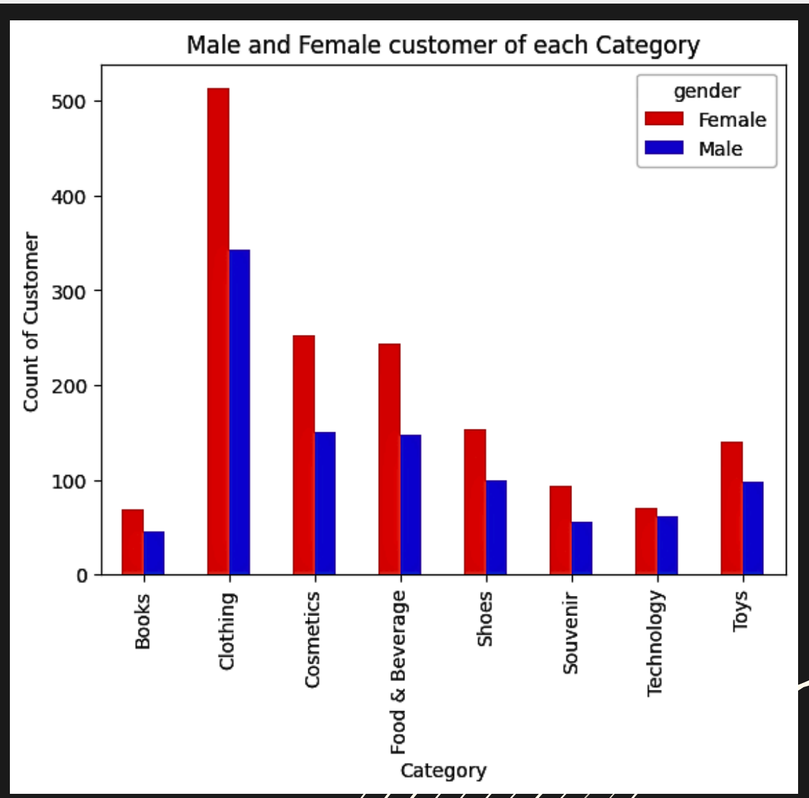
Q11. COUNT THE MALE AND FEMALE CUSTOMER OF EACH CATEGORY

```
customer_gender = newdata.groupby(['category','gender'])['customer_id'].count().unstack()
print(customer_gender)

customer_gender.plot(kind='bar',color=['red','blue'])
plot.xlabel('Category')
plot.ylabel('Count of Customer')
plot.title('Male and Female customer of each Category')
plot.show()
```

RESULT -->

gender	Female	Male
category		
Books	69	46
Clothing	513	343
Cosmetics	252	151
Food & Beverage	244	147
Shoes	154	100
Souvenir	94	55
Technology	71	61
Toys	140	98



Q12. WHICH AGE CATEGORY IS MOST FREQUENT OF ITEM PURCHASED OF EACH CATEGORY

```
customer_age = newdata.groupby(['category','age'])['customer_id'].count().unstack(level=0)
print(customer_age.idxmax())
```

RESULT -->

```
category
Books                22
Clothing             30
Cosmetics            26
Food & Beverage      26
Shoes                66
Souvenir             41
Technology           43
Toys                 38
dtype: int64
```

STRATEGIES

Focus on their most profitable customers

The data might reveal which customer segments are most profitable for the business. The company could then focus its marketing and sales efforts on attracting and retaining these customers.

STRATEGY 1

Invest in new products

The data may reveal areas where the business can improve its existing products or services. Businesses can then invest in research and development to make their offerings more appealing to customers.

STRATEGY 2

Expand into new markets

The data may show that there is an opportunity for the business to expand into new markets. Businesses can research new markets and develop strategies for entering them.

STRATEGY 3

OASIS INFOBYTE

THANK'S FOR WATCHING

Biswarup Neogi