Exploratory Data Analysis (EDA) Report

Overview

This project provides a detailed **Exploratory Data Analysis (EDA)** of the `customer_sales_data` dataset. The goal is to understand the company's financial performance, product category distribution, and customer behavior based on gender, age, and payment methods.

Dataset

- **Table name:** customer_sales_data
- **Number of rows:** (99,457)
- **Number of columns:**(11)
- **Key columns:** customer_id, gender, age, payment_method, category, invoice_date, invoice_no, price, quantity, shopping_mall, total_sales

Analysis Sections

1. **Total Revenue by Year:** Analyze yearly revenue trends to identify growth or decline periods.
2. **Quantity by Product Category:** Identify top-selling and low-selling product categories.
3. **Top 5 Shopping Malls by Revenue:** Find the shopping malls generating the highest revenue.
4. **Gender Distribution by Product Categories:** Examine which gender prefers which product categories.
5. **Age Distribution by Payment Method:** Explore the relationship between age groups and preferred payment methods.

Importing Data from SQL Server

In this section, we connect to the SQL Server database and load the **customer_sales_data** table for further analysis.

In [1]:
```
pip install ipython-sql sqlalchemy pyodbc
```

```
Requirement already satisfied: ipython-sql in f:\program files\anaconda\lib\site-packages (0.5.0)
Requirement already satisfied: sqlalchemy in f:\program files\anaconda\lib\site-packages (2.0.39)
Requirement already satisfied: pyodbc in f:\program files\anaconda\lib\site-packages (5.2.0)
Requirement already satisfied: prettytable in f:\program files\anaconda\lib\site-packages (from ipython-sql) (3.17.0)
Requirement already satisfied: ipython in f:\program files\anaconda\lib\site-packages (from ipython-sql) (8.30.0)
Requirement already satisfied: sqlparse in f:\program files\anaconda\lib\site-packages (from ipython-sql) (0.5.4)
Requirement already satisfied: six in f:\program files\anaconda\lib\site-packages (from ipython-sql) (1.17.0)
Requirement already satisfied: ipython-genutils in f:\program files\anaconda\lib\site-packages (from ipython-sql) (0.
2.0)
Requirement already satisfied: greenlet!=0.4.17 in f:\program files\anaconda\lib\site-packages (from sqlalchemy) (3.
1.1)
Requirement already satisfied: typing-extensions>=4.6.0 in f:\program files\anaconda\lib\site-packages (from sqlalche
my) (4.12.2)
Requirement already satisfied: decorator in f:\program files\anaconda\lib\site-packages (from ipython->ipython-sql)
(5.1.1)
Requirement already satisfied: jedi>=0.16 in f:\program files\anaconda\lib\site-packages (from ipython->ipython-sql)
(0.19.2)
Requirement already satisfied: matplotlib-inline in f:\program files\anaconda\lib\site-packages (from ipython->ipytho
n-sql) (0.1.6)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in f:\program files\anaconda\lib\site-packages (from ipy
thon->ipython-sql) (3.0.43)
Requirement already satisfied: pygments>=2.4.0 in f:\program files\anaconda\lib\site-packages (from ipython->ipython-
sql) (2.19.1)
Requirement already satisfied: stack-data in f:\program files\anaconda\lib\site-packages (from ipython->ipython-sql)
(0.2.0)
Requirement already satisfied: traitlets>=5.13.0 in f:\program files\anaconda\lib\site-packages (from ipython->ipytho
n-sql) (5.14.3)
Requirement already satisfied: colorama in f:\program files\anaconda\lib\site-packages (from ipython->ipython-sql)
(0.4.6)
Requirement already satisfied: wcwidth in f:\program files\anaconda\lib\site-packages (from prompt-toolkit<3.1.0,>=3.
0.41->ipython->ipython-sql) (0.2.5)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in f:\program files\anaconda\lib\site-packages (from jedi>=0.16->i
python->ipython-sql) (0.8.4)
Requirement already satisfied: executing in f:\program files\anaconda\lib\site-packages (from stack-data->ipython->ip
ython-sql) (0.8.3)
Requirement already satisfied: asttokens in f:\program files\anaconda\lib\site-packages (from stack-data->ipython->ip
ython-sql) (3.0.0)
Requirement already satisfied: pure-eval in f:\program files\anaconda\lib\site-packages (from stack-data->ipython->ip
ython-sql) (0.2.2)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
%load_ext sql
```

In [4]:
```
%sql mssql+pyodbc://DESKTOP-K8SDHTT/sales?driver=ODBC+Driver+17+for+SQL+Server&trusted_connection=yes
```

In [6]:
```
pip install pandas sqlalchemy pyodbc
```

```
Requirement already satisfied: pandas in f:\program files\anaconda\envs\notebook\lib\site-packages (2.3.3)
Requirement already satisfied: sqlalchemy in f:\program files\anaconda\envs\notebook\lib\site-packages (2.0.44)
Requirement already satisfied: pyodbc in f:\program files\anaconda\envs\notebook\lib\site-packages (5.3.0)
Requirement already satisfied: numpy>=1.26.0 in f:\program files\anaconda\envs\notebook\lib\site-packages (from panda
s) (2.3.5)
Requirement already satisfied: python-dateutil>=2.8.2 in f:\program files\anaconda\envs\notebook\lib\site-packages (f
rom pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in f:\program files\anaconda\envs\notebook\lib\site-packages (from panda
s) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in f:\program files\anaconda\envs\notebook\lib\site-packages (from pand
as) (2025.2)
Requirement already satisfied: greenlet>=1 in f:\program files\anaconda\envs\notebook\lib\site-packages (from sqlalch
emy) (3.2.4)
Requirement already satisfied: typing-extensions>=4.6.0 in f:\program files\anaconda\envs\notebook\lib\site-packages
(from sqlalchemy) (4.15.0)
Requirement already satisfied: six>=1.5 in f:\program files\anaconda\envs\notebook\lib\site-packages (from python-dat
eutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
import pandas as pd
from sqlalchemy import create_engine

# معلومات الاتصال
server = 'DESKTOP-K8SDHTT'
database = 'sales'

# الاتصال باستخدام Windows Authentication
engine = create_engine(
    f"mssql+pyodbc://@{server}/{database}?trusted_connection=yes&driver=ODBC+Driver+17+for+SQL+Server"
)

# اختبار الاتصال: قراءة أول 10 صفوف من جدول sales_data
df = pd.read_sql("SELECT TOP (10) * FROM [dbo].[sales_data]", engine)
df
```

Out[4]:

	invoice_no	customer_id	category	quantity	price	invoice_date	shopping_mall	total_sales
0	I138884	C241288	Clothing	5	1500.400024	2022-08-05	Kanyon	7502.000000
1	I317333	C111565	Shoes	3	1800.510010	2021-12-12	Forum Istanbul	5401.529785
2	I127801	C266599	Clothing	1	300.079987	2021-11-09	Metrocity	300.079987
3	I173702	C988172	Shoes	5	3000.850098	2021-05-16	Metropol AVM	15004.250000
4	I337046	C189076	Books	4	60.599998	2021-10-24	Kanyon	242.399994
5	I227836	C657758	Clothing	5	1500.400024	2022-05-24	Forum Istanbul	7502.000000
6	I121056	C151197	Cosmetics	1	40.660000	2022-03-13	Istinye Park	40.660000
7	I293112	C176086	Clothing	2	600.159973	2021-01-13	Mall of Istanbul	1200.319946
8	I293455	C159642	Clothing	3	900.239990	2021-11-04	Metrocity	2700.719971
9	I326945	C283361	Clothing	2	600.159973	2021-08-22	Kanyon	1200.319946

Testing Cleaning

In [5]:
```
# Load dataset from SQL Server
query = "SELECT * FROM customer_sales_data"
df = pd.read_sql(query, engine)

df.head()
```

Out[5]:

	customer_id	gender	age	payment_method	category	invoice_date	invoice_no	price	quantity	shopping_mall	tot
0	C100005	Male	34.0	Cash	Shoes	2023-03-03	I158163	1200.339966	2	Kanyon	2400
1	C100006	Male	44.0	Credit Card	Toys	2022-12-01	I262373	107.519997	3	Cevahir AVM	322
2	C100012	Male	25.0	Cash	Food & Beverage	2021-08-15	I334895	26.150000	5	Kanyon	130
3	C100019	Female	21.0	Credit Card	Toys	2021-07-25	I202043	35.840000	1	Metrocity	35
4	C100025	Male	55.0	Debit Card	Toys	2021-06-03	I303349	71.680000	2	Metrocity	143

EDA

1- TOTAL REVENUE BY YEAR

2- CATEGORY BY QUANTITY

3- TOP 5 SHOPPING MALL BY REVENUE

4- GENDER DISTRIBUTION BY PRODUCT CATEGORIES

5- AGE DISTRIBUTION BY PAYMENT METHOD

1- TOTAL REVENUE BY YEAR

In [53]:
```
query = """SELECT YEAR(invoice_date) AS year, ROUND(SUM(total_sales), 0) AS total_sales_rounded
           FROM customer_sales_data
           GROUP BY YEAR(invoice_date)
           order by SUM(total_sales) desc"""
df = pd.read_sql(query, engine)
df
```

Out[53]:

	year	total_sales_rounded
0	2022	115436813.0
1	2021	114560569.0
2	2023	21508409.0

2- CATEGORY BY QUANTITY

In [52]:
```
query = """select category, count(quantity) AS quantity_count
            from customer_sales_data
            group by category
            order by quantity_count desc"""
df = pd.read_sql(query,engine)
df
```

Out[52]:

	category	quantity_count
0	Clothing	34487
1	Cosmetics	15097
2	Food & Beverage	14776
3	Toys	10087
4	Shoes	10034
5	Souvenir	4999
6	Technology	4996
7	Books	4981

3- TOP 5 SHOPPING MALL BY REVENUE

In [51]:
```
query = """ select top(5) shopping_mall , round(sum(total_sales),0) as revenue
from customer_sales_data
```

```
group by shopping_mall
order by revenue desc"""
df = pd.read_sql(query,engine)
df
```

Out[51]:

	shopping_mall	revenue
0	Mall of Istanbul	50872481.0
1	Kanyon	50554230.0
2	Metrocity	37302787.0
3	Metropol AVM	25379913.0
4	Istinye Park	24618827.0

4- GENDER DISTRIBUTION BY PRODUCT CATEGORIES

In [54]:

```
query = """ select category ,gender,count(*) as count
from customer_sales_data
group by gender, category
order by count desc"""
df = pd.read_sql(query,engine)
df
pivot_df = df.pivot_table(
    index='category',
    columns='gender',
    values='count',
    aggfunc='sum',
    fill_value=0
)

pivot_df
```

Out[54]:

gender category	Female	Male
Books	2906	2075
Clothing	20652	13835
Cosmetics	9070	6027
Food & Beverage	8804	5972
Shoes	5967	4067
Souvenir	3017	1982
Technology	2981	2015
Toys	6085	4002

5 - AGE DISTRIBUTION BY PAYMENT METHOD

In [68]:
```
query = """
SELECT
    CASE
        WHEN age BETWEEN 0 AND 25 THEN '0-25'
        WHEN age BETWEEN 26 AND 50 THEN '26-50'
        WHEN age BETWEEN 51 AND 75 THEN '51-75'
        WHEN age BETWEEN 76 AND 100 THEN '76-100'
        ELSE 'Other'
    END AS age_range,
    payment_method,
    COUNT(*) AS count
FROM customer_sales_data
GROUP BY
    CASE
        WHEN age BETWEEN 0 AND 25 THEN '0-25'
        WHEN age BETWEEN 26 AND 50 THEN '26-50'
        WHEN age BETWEEN 51 AND 75 THEN '51-75'
        WHEN age BETWEEN 76 AND 100 THEN '76-100'
        ELSE 'Other'
```

```
    END,
    payment_method
    order by count desc
"""

df = pd.read_sql(query, engine)
df
```

Out[68]:

	age_range	payment_method	count
0	26-50	Cash	21395
1	26-50	Credit Card	16819
2	51-75	Cash	16169
3	51-75	Credit Card	12660
4	26-50	Debit Card	9727
5	51-75	Debit Card	7225
6	0-25	Cash	6833
7	0-25	Credit Card	5419
8	0-25	Debit Card	3091
9	Other	Cash	50
10	Other	Debit Card	36
11	Other	Credit Card	33

In [ ]: