Retail Sales Dataset✨

Dataset Overview:

This dataset is a snapshot of a fictional retail landscape, capturing essential attributes that drive retail operations and customer interactions. It includes key details such as Transaction ID, Date, Customer ID, Gender, Age, Product Category, Quantity, Price per Unit, and Total Amount.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
data = pd.read_csv("C:/Users/Dooba/Desktop/Analysis with pyhton/Retail sales/retail_sales_dataset.csv")
data.head()
```

Out[2]:

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100

Data cleaning and Manipulation

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Transaction ID    1000 non-null   int64 
 1   Date              1000 non-null   object
 2   Customer ID       1000 non-null   object
 3   Gender            1000 non-null   object
 4   Age               1000 non-null   int64 
 5   Product Category  1000 non-null   object
 6   Quantity          1000 non-null   int64 
 7   Price per Unit    1000 non-null   int64 
 8   Total Amount      1000 non-null   int64 
dtypes: int64(5), object(4)
memory usage: 70.4+ KB
```

In [4]:
```
import datetime as dt
data['Date'] = pd.to_datetime(data['Date'], errors='coerce', infer_datetime_format=True)
data["Year_Month"]=data["Date"].dt.to_period("M")
data=data.sort_values(by="Date")
```

In [5]:
```
data.isnull().sum()
```

Out[5]:
```
Transaction ID      0
Date                0
Customer ID         0
Gender              0
Age                 0
Product Category    0
Quantity            0
Price per Unit      0
Total Amount        0
Year_Month          0
dtype: int64
```

In [6]:
```
data.duplicated().sum()
```

Out[6]:
```
0
```

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

Out[7]:

	Transaction ID	Age	Quantity	Price per Unit	Total Amount
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	500.500000	41.392000	2.514000	179.890000	456.000000
std	288.819436	13.68143	1.132734	189.681356	559.997632
min	1.000000	18.000000	1.000000	25.000000	25.000000
25%	250.750000	29.000000	1.000000	30.000000	60.000000
50%	500.500000	42.000000	3.000000	50.000000	135.000000
75%	750.250000	53.000000	4.000000	300.000000	900.000000
max	1000.000000	64.000000	4.000000	500.000000	2000.000000

In [8]:
```
data =data.drop("Transaction ID", axis=1)
```

In [9]:
```
Gender_cou=data["Gender"].value_counts()
Gender_cou
```

```
Female    510
Male      490
Name: Gender, dtype: int64
```

In [10]:
```
revenue_Trend =data.groupby("Year_Month")["Total Amount"].sum().sort_values(ascending=False)
revenue_Trend
```

Out[10]:
```
Year_Month
2023-05    53150
2023-10    46580
2023-12    44690
2023-02    44060
2023-08    36960
2023-06    36715
2023-07    35465
2023-01    35450
2023-11    34920
2023-04    33870
2023-03    28990
2023-09    23620
2024-01     1530
Freq: M, Name: Total Amount, dtype: int64
```

In [11]:
```
Product_revenue =data.groupby("Product Category")["Total Amount"].sum().sort_values(ascending=False)
Product_revenue
```

Out[11]:
```
Product Category
Electronics    156905
Clothing       155580
Beauty         143515
Name: Total Amount, dtype: int64
```

In [12]:
```
Gender_revenue =data.groupby("Gender")["Total Amount"].sum().sort_values(ascending=False)
Gender_revenue
```

Out[12]:
```
Gender
Female    232840
Male      223160
Name: Total Amount, dtype: int64
```

In [13]:
```
Gender_order =data.groupby("Gender")["Quantity"].sum().sort_values(ascending=False)
Gender_order
```

Out[13]:
```
Gender
Female    1298
Male      1216
Name: Quantity, dtype: int64
```

In [15]:
```
Age_categ=[(15,25),(26,35),(36,45),(46,55),(56,65)]
data["Age_categ"] =pd.cut(data["Age"],bins=[x[0] for x in Age_categ]+ [Age_categ[-1][1]],labels=[f"{x[0]}-{x[1]}" for x in Age_categ])
data
```

Out[15]:

	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount	Year_Month	Age_categ
521	2023-01-01	CUST522	Male	46	Beauty	3	500	1500	2023-01	36-45
179	2023-01-01	CUST180	Male	41	Clothing	3	300	900	2023-01	36-45
558	2023-01-01	CUST559	Female	40	Clothing	4	300	1200	2023-01	36-45
302	2023-01-02	CUST303	Male	19	Electronics	3	30	90	2023-01	15-25
978	2023-01-02	CUST979	Female	19	Beauty	1	25	25	2023-01	15-25
...	...	...	...	...	...	...	...	...	...	...
232	2023-12-29	CUST233	Female	51	Beauty	2	300	600	2023-12	46-55
804	2023-12-29	CUST805	Female	30	Beauty	3	500	1500	2023-12	26-35
856	2023-12-31	CUST857	Male	60	Electronics	2	25	50	2023-12	56-65
210	2024-01-01	CUST211	Male	42	Beauty	3	500	1500	2024-01	36-45
649	2024-01-01	CUST650	Male	55	Electronics	1	30	30	2024-01	46-55

1000 rows × 10 columns

In [16]:
```
Age_revenue =data.groupby("Age_categ")["Total Amount"].sum().sort_values(ascending=False)
Age_revenue
```

Out[16]:
```
Age_categ
15-25    98530
46-55    97040
36-45    95855
26-35    93605
56-65    70970
Name: Total Amount, dtype: int64
```

In [17]:
```
Total_quantity = data["Quantity"].sum()
Total_quantity
```

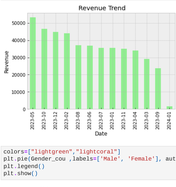
Out[17]:
```
2514
```

In [18]:
```
Total_revenue = data["Total Amount"].sum()
Total_revenue
```

Out[18]:
```
456000
```

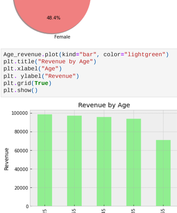
Data Visualization

In [19]:
```
from matplotlib import style
style.use("bmh")
plt.figure (facecolor="white")
revenue_Trend.plot(kind="bar", color="lightgreen")
plt.title(' Revenue Trend")
plt.xlabel("Date")
plt. ylabel("Revenue")
plt.grid(True)
plt.show()
```

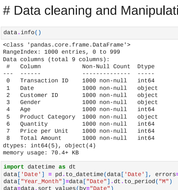


In [20]:
```
colors=["lightgreen","lightcoral"]
plt.pie(Gender_cou ,labels=['Male', 'Female'], autopct='%1.1f%%',colors=colors, explode=[0,0.1],shadow=True)
plt.legend()
plt.show()
```

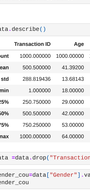
In [21]:
```
Product_revenue.plot(kind="bar", color="lightgreen")
plt.title("Revenue by Product")
plt.xlabel("Product")
plt. ylabel("Revenue")
plt.grid(True)
plt.show()
```



In [22]:
```
Gender_revenue.plot(kind="bar", color="lightgreen")
plt.title("Revenue by Gender")
plt.xlabel("Gender")
plt. ylabel("Revenue")
plt.grid(True)
plt.show()
```



In [23]:
```
colors=["lightgreen","lightcoral"]
plt.pie(Gender_order ,labels=['Male', 'Female'], autopct='%1.1f%%',colors=colors, explode=[0,0.1],shadow=True)
plt.title("Quantity by Gender")
plt.legend()
plt.show()
```



In [24]:
```
Age_revenue.plot(kind="bar", color="lightgreen")
plt.title("Revenue by Age")
plt.xlabel("Age")
plt. ylabel("Revenue")
plt.grid(True)
plt.show()
```