Data Loading and Cleaning

```
# importing essential libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from pandas.plotting import autocorrelation_plot

data=pd.read_csv('/content/retail_sales_dataset_oasis.csv')

data
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 1000,\n  \"fields\":
[\n    {\n      \"column\": \"Transaction ID\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 288,\n
\"min\": 1,\n        \"max\": 1000,\n        \"num_unique_values\":
1000,\n        \"samples\": [\n          522,\n          738,\n
741\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Date\",\n      \"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 345,\n        \"samples\": [\n          \"2023-
04-04\",\n          \"2023-04-13\",\n          \"2023-04-15\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Customer ID\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 1000,\n        \"samples\": [\n
\"CUST522\",\n          \"CUST738\",\n          \"CUST741\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Female\",\n          \"Male\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 13,\n        \"min\": 18,\n
\"max\": 64,\n        \"num_unique_values\": 47,\n        \"samples\":
[\n          48,\n          61\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Product Category\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"Beauty\",\n          \"Clothing\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Quantity\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 1,\n        \"max\": 4,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          2,\n
```

```
4\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Price per Unit\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 189,\n        \"min\": 25,\n
\"max\": 500,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          500,\n          300\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Total Amount\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
559,\n        \"min\": 25,\n        \"max\": 2000,\n
\"num_unique_values\": 18,\n        \"samples\": [\n          150,\n
1000\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"data"}
```


```
#displaying first 4 cutomer from data
data.head()
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 1000,\n  \"fields\":
[\n    {\n      \"column\": \"Transaction ID\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 288,\n
\"min\": 1,\n        \"max\": 1000,\n        \"num_unique_values\":
1000,\n        \"samples\": [\n          522,\n          738,\n
741\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Date\",\n      \"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 345,\n        \"samples\": [\n          \"2023-
04-04\",\n          \"2023-04-13\",\n          \"2023-04-15\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Customer ID\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 1000,\n        \"samples\": [\n
\"CUST522\",\n          \"CUST738\",\n          \"CUST741\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Female\",\n          \"Male\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 13,\n        \"min\": 18,\n
\"max\": 64,\n        \"num_unique_values\": 47,\n        \"samples\":
[\n          48,\n          61\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Product Category\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"Beauty\",\n          \"Clothing\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Quantity\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
```

```
1,\n        \"min\": 1,\n        \"max\": 4,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          2,\n
4\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Price per Unit\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 189,\n        \"min\": 25,\n
\"max\": 500,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          500,\n          300\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Total Amount\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
559,\n        \"min\": 25,\n        \"max\": 2000,\n
\"num_unique_values\": 18,\n        \"samples\": [\n          150,\n
1000\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"data"}
```


```
#displaying last 4 cutomer from data
data.tail()
```


```
{"repr_error":"0","type":"dataframe"}
```


```
#Getting unique value from category column
data['Product Category'].unique()
```

```
array(['Beauty', 'Clothing', 'Electronics'], dtype=object)
```

```
data['Product Category'].value_counts()
```

```
Product Category
Clothing       351
Electronics    342
Beauty         307
Name: count, dtype: int64
```

```
data.isnull().sum()
```

```
Transaction ID      0
Date                0
Customer ID         0
Gender              0
Age                 0
Product Category    0
Quantity            0
Price per Unit      0
Total Amount        0
dtype: int64
```

```
data.head()
```

{"summary":"{\n \"name\": \"data\",\n \"rows\": 1000,\n \"fields\":
[\n   {\n   \"column\": \"Transaction ID\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 288,\n
\"min\": 1,\n   \"max\": 1000,\n   \"num_unique_values\":
1000,\n   \"samples\": [\n   522,\n   738,\n
741\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Date\",\n   \"properties\": {\n   \"dtype\": \"object\",\n
\"num_unique_values\": 345,\n   \"samples\": [\n   \"2023-
04-04\",\n   \"2023-04-13\",\n   \"2023-04-15\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"Customer ID\",\n
\"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 1000,\n   \"samples\": [\n
\"CUST522\",\n   \"CUST738\",\n   \"CUST741\"\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Gender\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 2,\n   \"samples\":
[\n   \"Female\",\n   \"Male\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Age\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 13,\n   \"min\": 18,\n
\"max\": 64,\n   \"num_unique_values\": 47,\n   \"samples\":
[\n   48,\n   61\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"Product Category\",\n   \"properties\": {\n
\"dtype\": \"category\",\n   \"num_unique_values\": 3,\n
\"samples\": [\n   \"Beauty\",\n   \"Clothing\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"Quantity\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1,\n   \"min\": 1,\n   \"max\": 4,\n
\"num_unique_values\": 4,\n   \"samples\": [\n   2,\n
4\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Price per Unit\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 189,\n   \"min\": 25,\n
\"max\": 500,\n   \"num_unique_values\": 5,\n   \"samples\":
[\n   500,\n   300\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Total Amount\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
559,\n   \"min\": 25,\n   \"max\": 2000,\n
\"num_unique_values\": 18,\n   \"samples\": [\n   150,\n
1000\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"data"}

since data is cleaned we moved toward next step that is observing descriptive statistics.

Descriptive Statistics

```
data.describe()
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 8,\n  \"fields\": [\n    {\n      \"column\": \"Transaction ID\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 360.1000917722167,\n        \"min\": 1.0,\n        \"max\": 1000.0,\n        \"num_unique_values\": 6,\n        \"samples\": [\n          1000.0,\n          500.5,\n          750.25\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 340.78610904107535,\n        \"min\": 13.681429659122532,\n        \"max\": 1000.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          41.392,\n          42.0,\n          1000.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Quantity\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 352.71483432640656,\n        \"min\": 1.0,\n        \"max\": 1000.0,\n        \"num_unique_values\": 6,\n        \"samples\": [\n          1000.0,\n          2.514,\n          4.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Price per Unit\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 330.7254087976672,\n        \"min\": 25.0,\n        \"max\": 1000.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          179.89,\n          50.0,\n          1000.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Total Amount\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 660.8084213867855,\n        \"min\": 25.0,\n        \"max\": 2000.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          456.0,\n          135.0,\n          1000.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```


```
np.median(data['Age'])
```

```
42.0
```

```
np.median(data['Price per Unit'])
```

```
50.0
```

```
np.median(data['Total Amount'])
```

```
135.0
```

```
stats.mode(data['Age'])
```

```
ModeResult(mode=43, count=31)
```

```
stats.mode(data['Price per Unit'])
```

```
ModeResult(mode=50, count=211)
```

```
stats.mode(data['Total Amount'])
```

```
ModeResult(mode=50, count=115)
```

Time Series Analysis

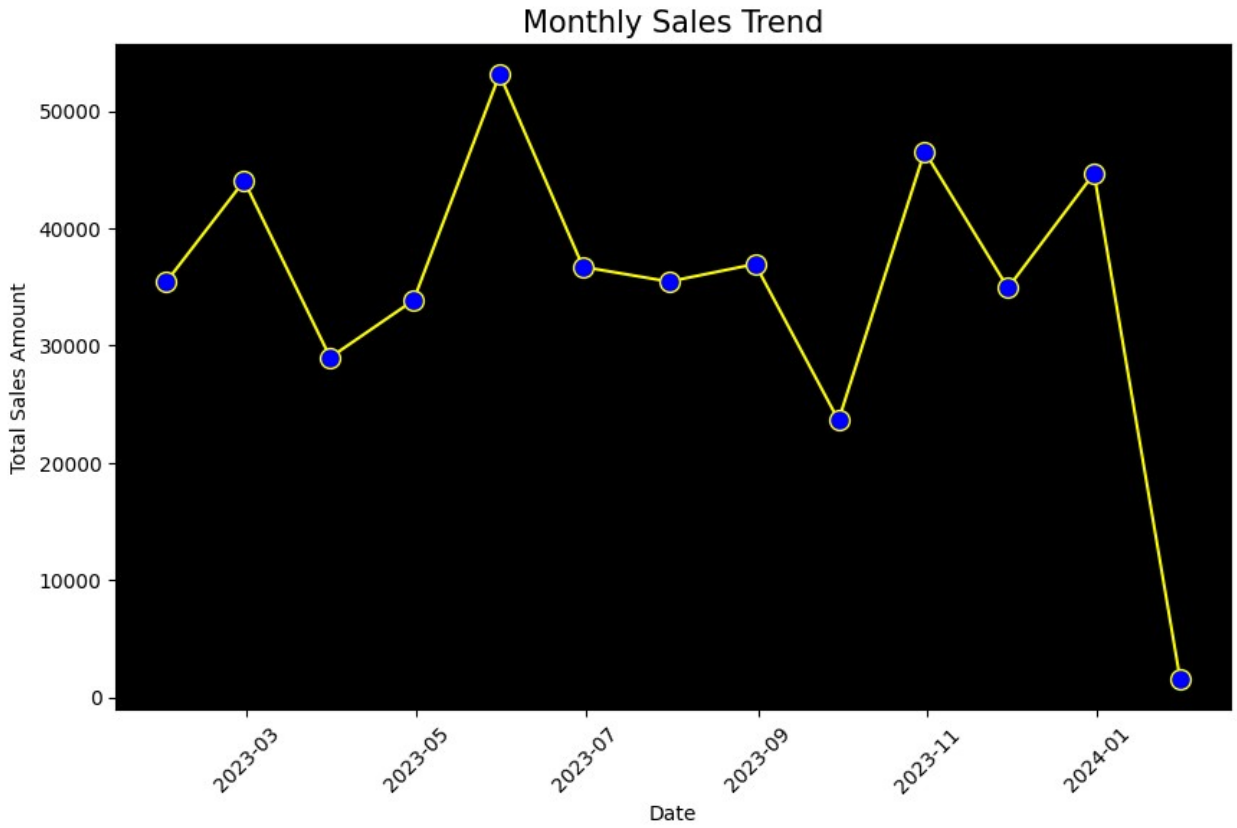
```
data.head()
```


```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 1000,\n  \"fields\":
[\n    {\n      \"column\": \"Transaction ID\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 288,\n
\"min\": 1,\n        \"max\": 1000,\n        \"num_unique_values\":
1000,\n        \"samples\": [\n          522,\n          738,\n
741\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Date\",\n      \"properties\": {\n        \"dtype\": \"date\",\n
\"min\": \"2023-01-01 00:00:00\",\n        \"max\": \"2024-01-01
00:00:00\",\n        \"num_unique_values\": 345,\n        \"samples\":
[\n          \"2023-04-04 00:00:00\",\n          \"2023-04-13
00:00:00\",\n          \"2023-04-15 00:00:00\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Customer ID\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 1000,\n        \"samples\": [\n
\"CUST522\",\n          \"CUST738\",\n          \"CUST741\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Female\",\n          \"Male\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 13,\n        \"min\": 18,\n
\"max\": 64,\n        \"num_unique_values\": 47,\n        \"samples\":
[\n          48,\n          61\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Product Category\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"Beauty\",\n          \"Clothing\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Quantity\",\n
```

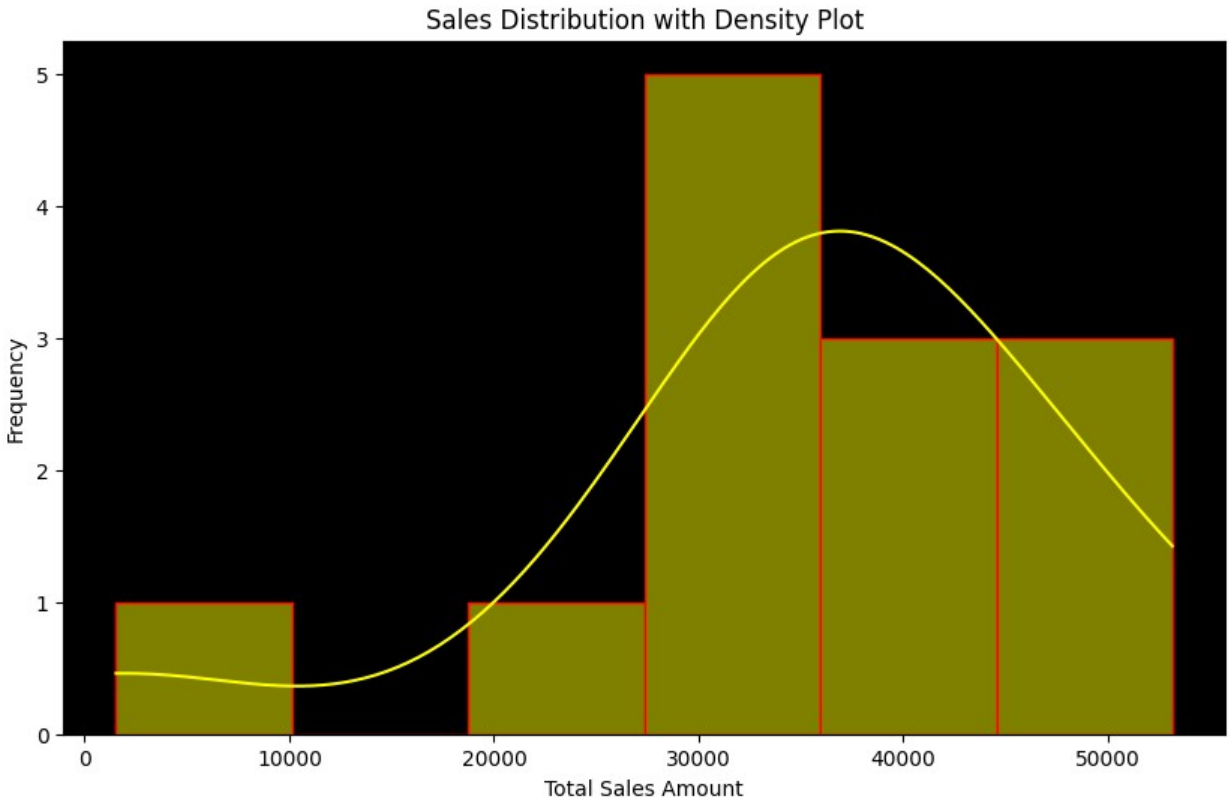
```
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 1,\n        \"max\": 4,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          2,\n
4\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Price per Unit\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 189,\n        \"min\": 25,\n
\"max\": 500,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          500,\n          300\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Total Amount\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
559,\n        \"min\": 25,\n        \"max\": 2000,\n
\"num_unique_values\": 18,\n        \"samples\": [\n          150,\n
1000\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"data"}
```

```
data['Date'] = pd.to_datetime(data['Date'])

monthly_sales = data.resample('M', on='Date')['Total Amount'].sum()

plt.figure(figsize=(10, 6))
plt.gca().set_facecolor('black')
plt.plot(monthly_sales.index, monthly_sales.values, marker='o',
color='yellow',markerfacecolor='blue',markersize=10)
plt.title('Monthly Sales Trend',fontsize=15)
plt.xlabel('Date',fontsize=10)
plt.ylabel('Total Sales Amount',fontsize=10)
plt.grid(False)
plt.xticks(rotation=45)
plt.show()
```

```
<ipython-input-44-68cdc1dfbe03>:3: FutureWarning: 'M' is deprecated
and will be removed in a future version, please use 'ME' instead.
  monthly_sales = data.resample('M', on='Date')['Total Amount'].sum()
```




```
plt.figure(figsize=(10, 6))
plt.gca().set_facecolor('black')
sns.histplot(monthly_sales, kde=True, color='yellow',edgecolor='red')
plt.title('Sales Distribution with Density Plot')
plt.xlabel('Total Sales Amount')
plt.ylabel('Frequency')
plt.show()
```

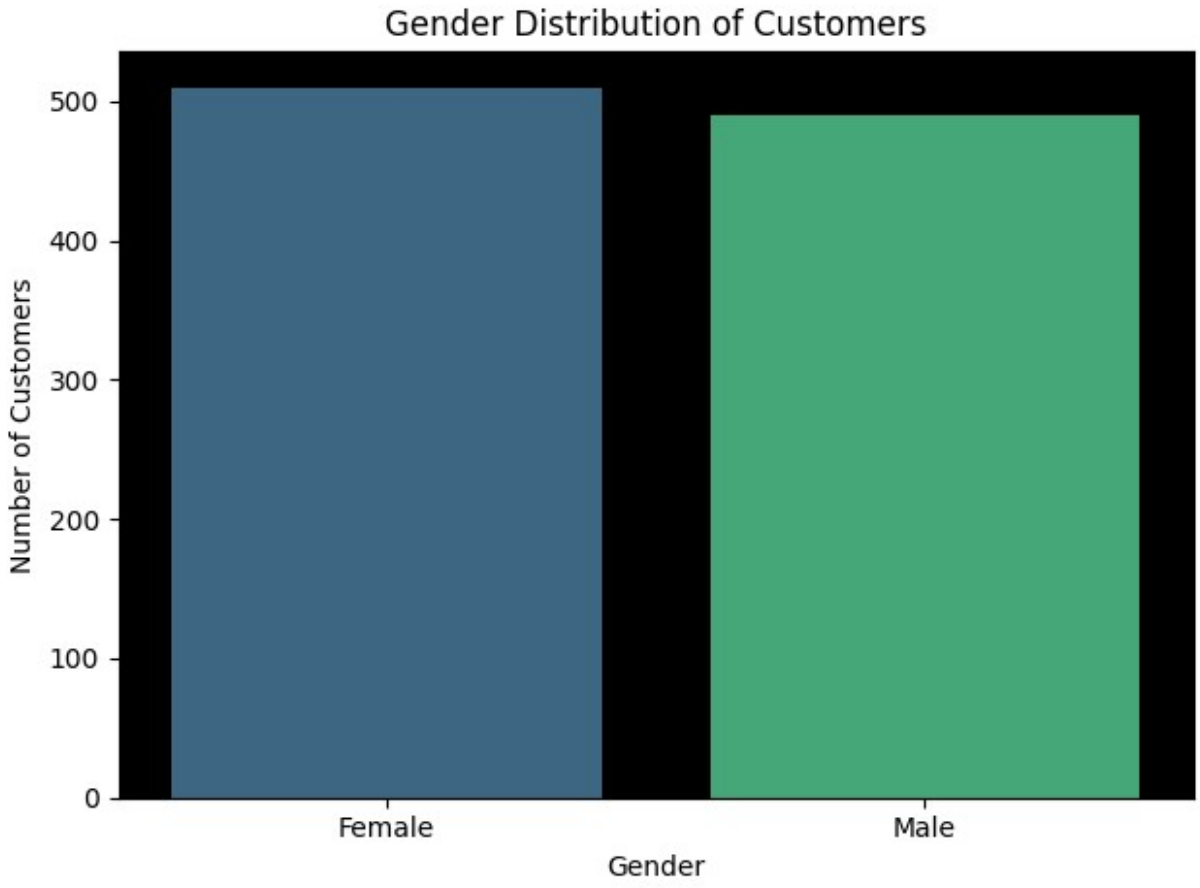


Customer and Product Analysis

```
gen_count = data['Gender'].value_counts()

sns.barplot(x=gen_count.index, y=gen_count.values, palette='viridis')
plt.gca().set_facecolor('black')
plt.title('Gender Distribution of Customers')
plt.xlabel('Gender')
plt.ylabel('Number of Customers')
plt.tight_layout()
plt.show()
```

```
<ipython-input-56-5241929ffbe0>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x=gen_count.index, y=gen_count.values,
palette='viridis')
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
```

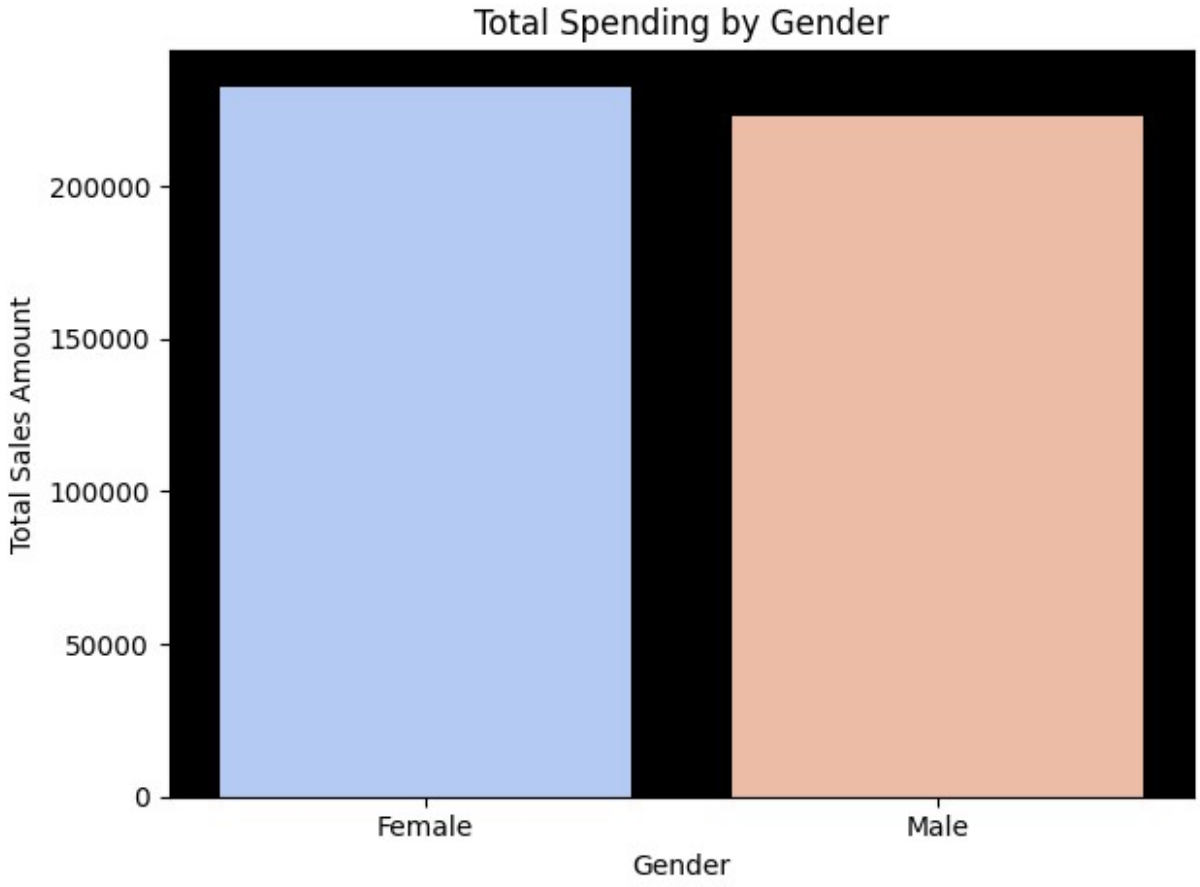
```
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
```




```
gen_spend = data.groupby('Gender')['Total Amount'].sum()

sns.barplot(x=gen_spend.index, y=gen_spend.values, palette='coolwarm')
plt.gca().set_facecolor('black')
plt.title('Total Spending by Gender')
plt.xlabel('Gender')
plt.ylabel('Total Sales Amount')
plt.tight_layout()
plt.show()
```

```
<ipython-input-57-7dd62dbaaac2>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.
```

```
  sns.barplot(x=gen_spend.index, y=gen_spend.values,
palette='coolwarm')
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
```

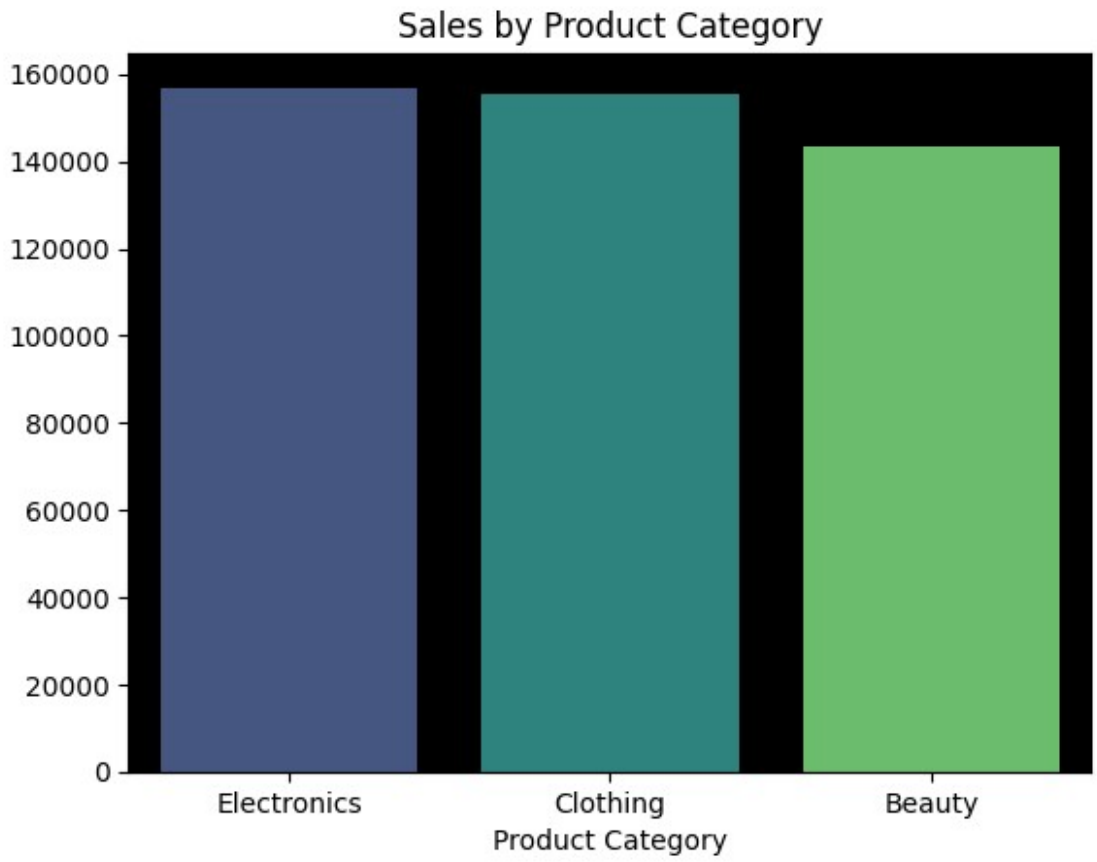



```
cat_sales = data.groupby('Product Category')['Total
Amount'].sum().sort_values(ascending=False)

sns.barplot(x=cat_sales.index, y=cat_sales.values, palette='viridis')
plt.gca().set_facecolor('black')
plt.title('Sales by Product Category')
```

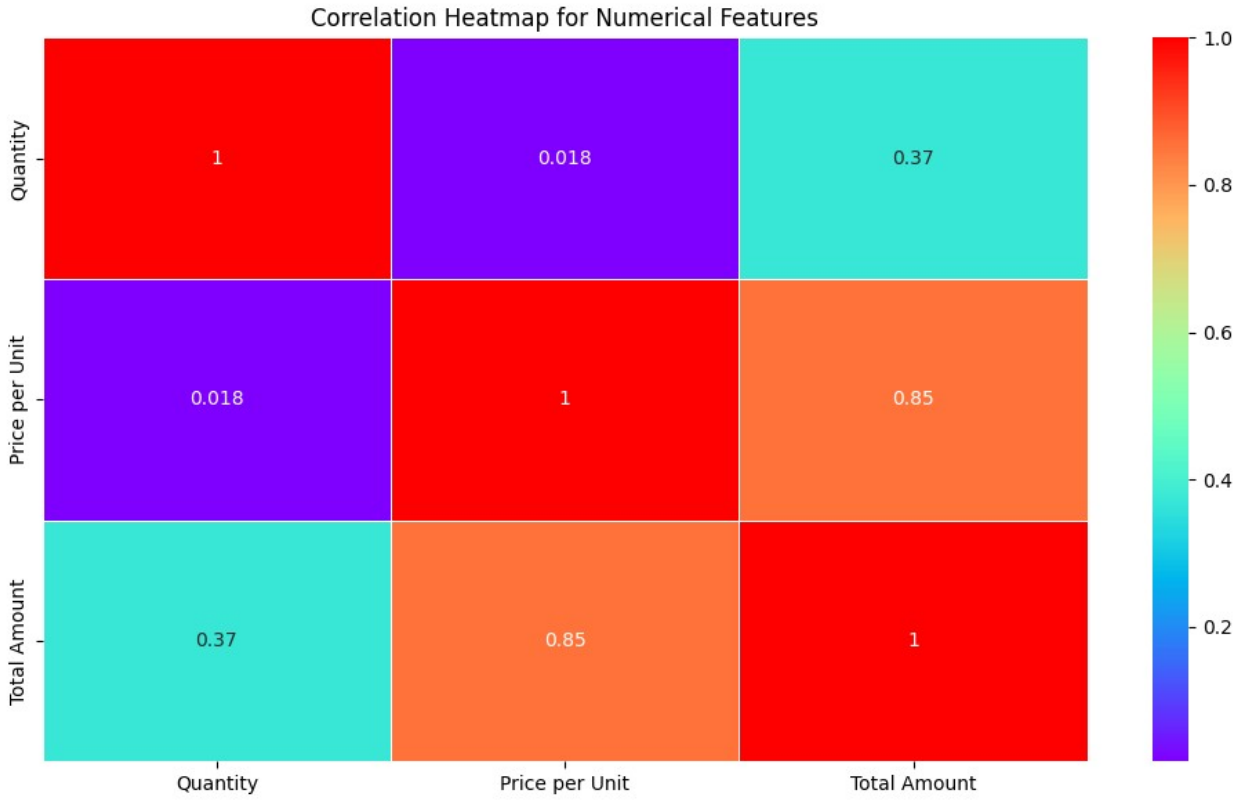
```
plt.xlabel('Product Category')
plt
```

```
<ipython-input-58-788b2efab6c4>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x=cat_sales.index, y=cat_sales.values,
palette='viridis')
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
/usr/local/lib/python3.10/dist-packages/seaborn/_base.py:949:
FutureWarning: When grouping with a length-1 list-like, you will need
to pass a length-1 tuple to get_group in a future version of pandas.
Pass `(name,)` instead of `name` to silence this warning.
  data_subset = grouped_data.get_group(pd_key)
```

```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.10/dist-
packages/matplotlib/pyplot.py'>
```




```
numerical_columns = data[['Quantity', 'Price per Unit', 'Total
Amount']]

corr_mat = numerical_columns.corr()

plt.figure(figsize=(10,6))
sns.heatmap(corr_mat, annot=True, cmap='rainbow', linewidths=0.5)
plt.title('Correlation Heatmap for Numerical Features')
plt.tight_layout()
plt.show()
```



Recommendations

- Focus marketing campaigns on the age group that spends the most. For example, if the 30-40 age group has the highest total spending, launch targeted promotions (e.g., personalized ads, discounts) toward this demographic
- Increase stock for the most popular categories to avoid shortages, especially during high-demand periods (e.g., Festivals, special promotions)
- If a strong correlation exists between higher unit prices and higher total amounts, introduce premium pricing strategies for high-demand or high-value products.
- Adjust product offerings in your store to reflect demographic preferences. For example, if men spend more on electronics, ensure that the store features a prominent electronics section for male shoppers.