ARTIFICIAL INTELLIGENCE PRESENTATION

COURSE TEACHER : **MISS AQSA UMAR**

RUBA INAM QURESHI

BCB-23S-010

UMM E AIMON

BCB-23S-028

ANAMTA IRFAN MANSOORI

BCB-23S-036

ANALYZING SALES DATA

SALES DATA:

EDA of Supermarket Performance

Focuses on analyzing the performance of supermarket sales through EDA.

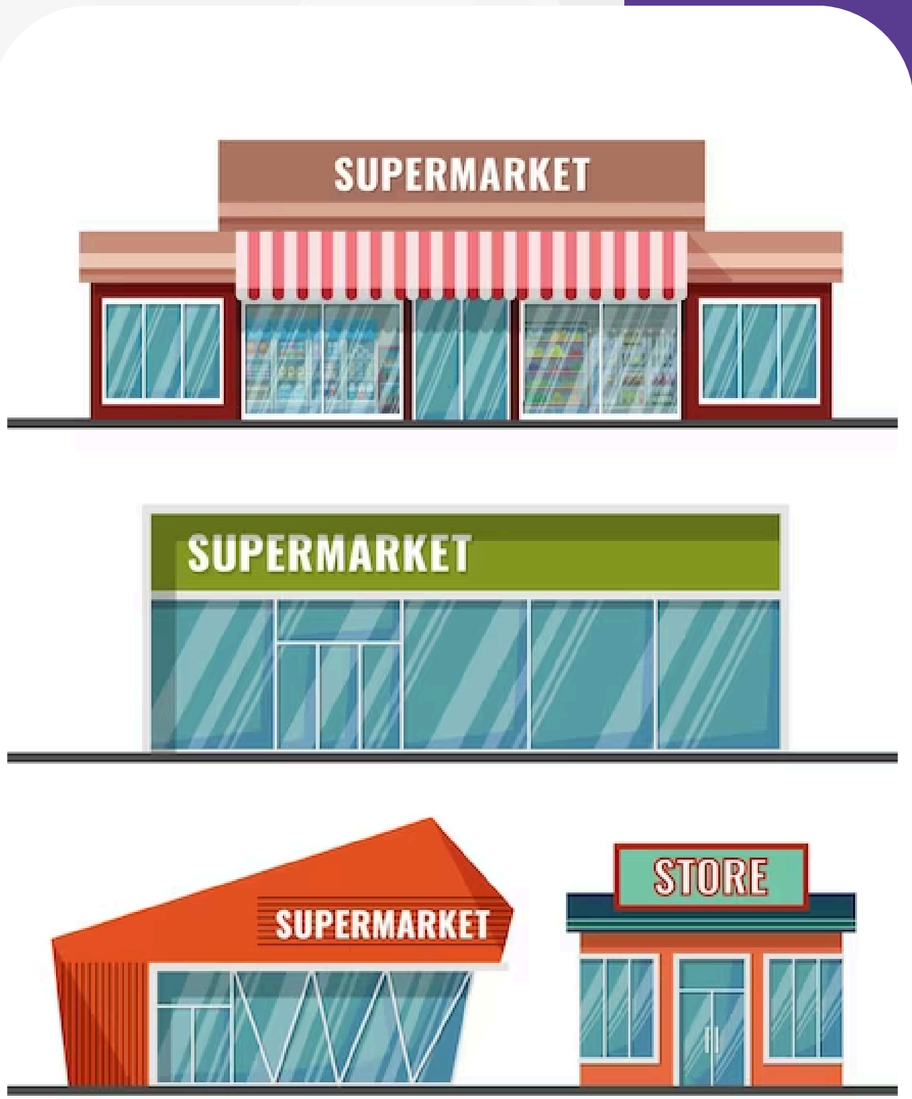
INTROCUTION TO EDA

What is EDA?

- Exploratory Data Analysis (EDA) is a technique used to analyze datasets to summarize their main characteristics, often using visual methods.

Why is EDA Important?

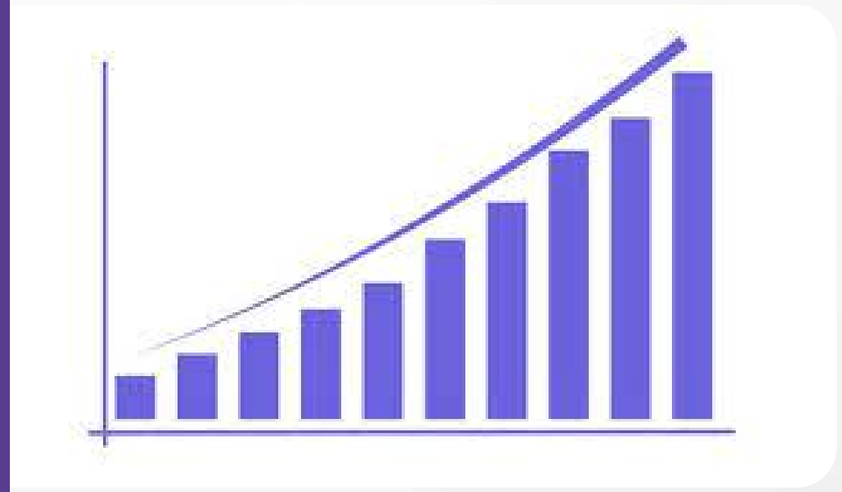
- Understanding the Dataset: EDA helps in gaining a comprehensive understanding of the dataset, its structure, and its variables.
- Identifying Data Issues: It identifies missing values, outliers, and errors, ensuring data quality before moving to modeling.
- Guiding Further Analysis: EDA guides the next steps in data analysis, including hypothesis testing and model selection.
- Providing Insights: It provides valuable insights that can drive business decisions and strategies.



INTRODUCTION TO EDA

Objectives of Our EDA

- To explore and analyze the supermarket sales dataset.
- To uncover sales patterns, customer behavior, and product performance.
- To identify key factors affecting sales.



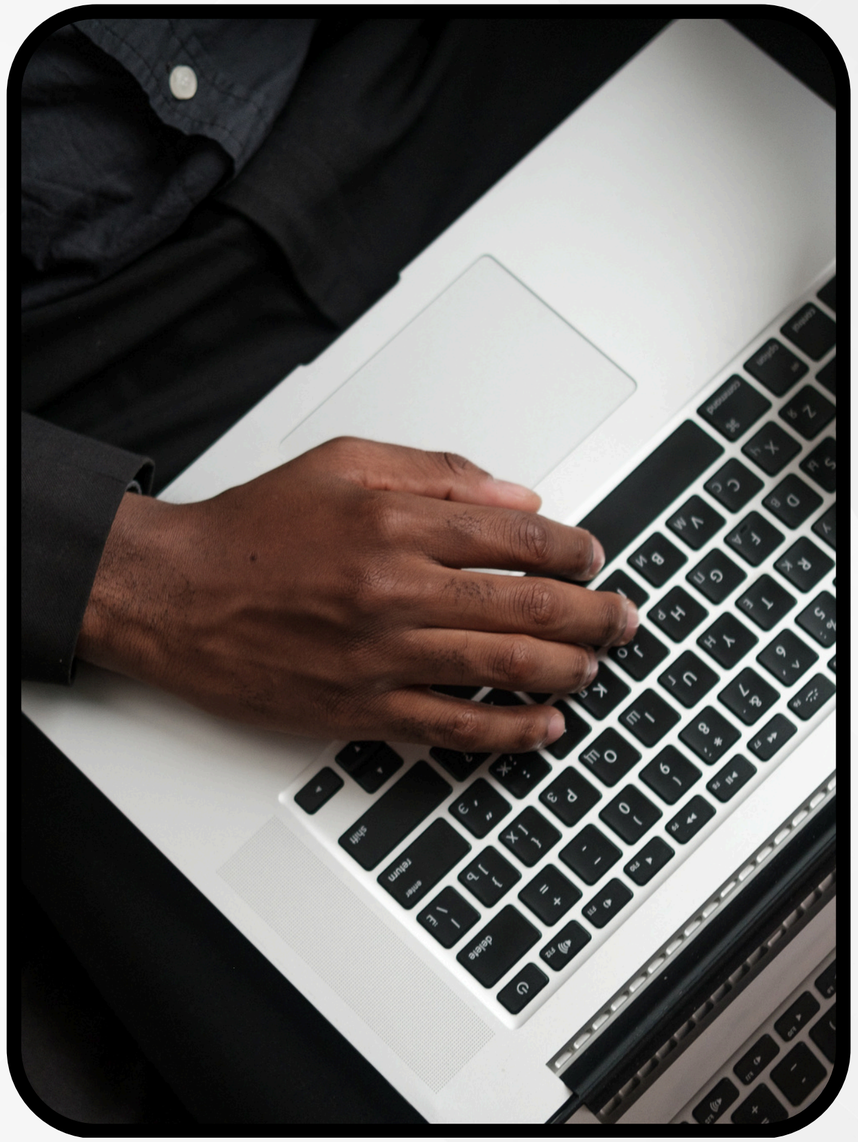
Visuals:

Include a simple diagram or flowchart that illustrates the steps of EDA: Data Cleaning, Data Visualization, and Summary Statistics.

WORKING ON EDA

Dataset Overview

- We analyzed a supermarket sales dataset with several key attributes including sales amount, date, product category, and customer details.
- The dataset contains [1000 records] and [17] columns.



WORKING ON EDA

Data Cleaning:

- Handling Missing Values: We addressed missing values by either removing incomplete rows or imputing with appropriate values.
- Removing Duplicates: We ensured no duplicate records were present to maintain data integrity.
- Outlier Treatment: We identified and handled outliers to avoid skewed analysis results.

Data Visualization:

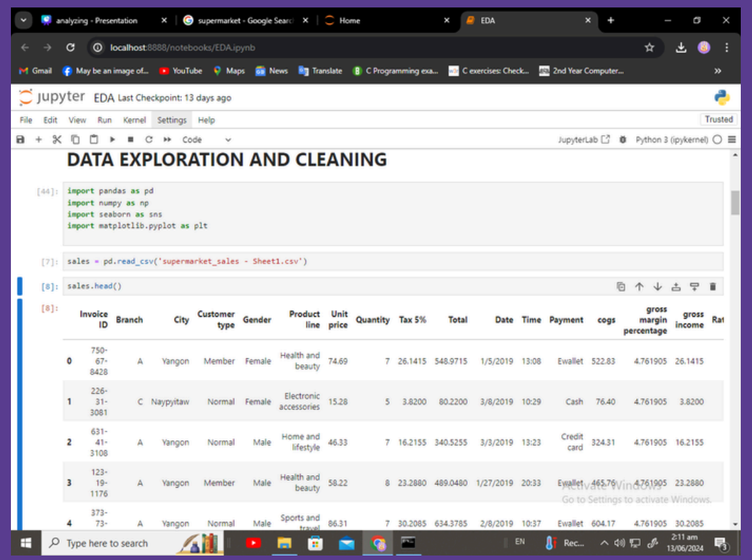
- Histograms and Bar Charts: To visualize the distribution of sales across different categories.
- Scatter Plots: To examine the relationships between sales and other numerical variables.
- Box Plots: To identify the spread and potential outliers in the data.

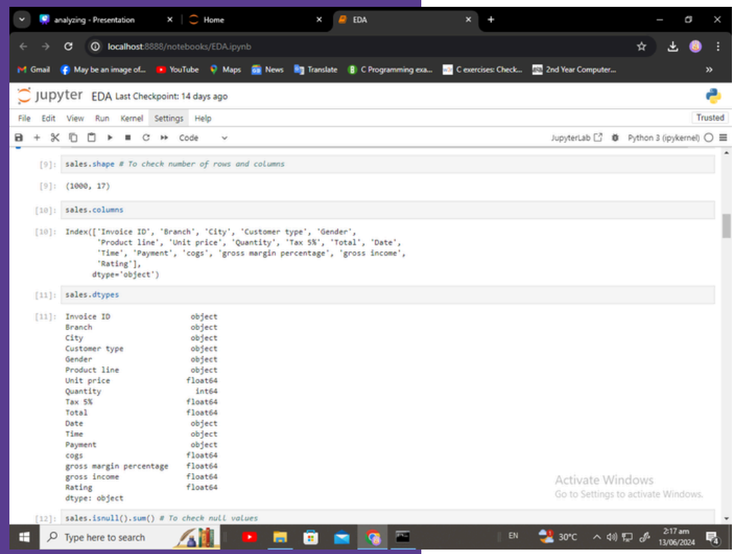
WORKING ON EDA

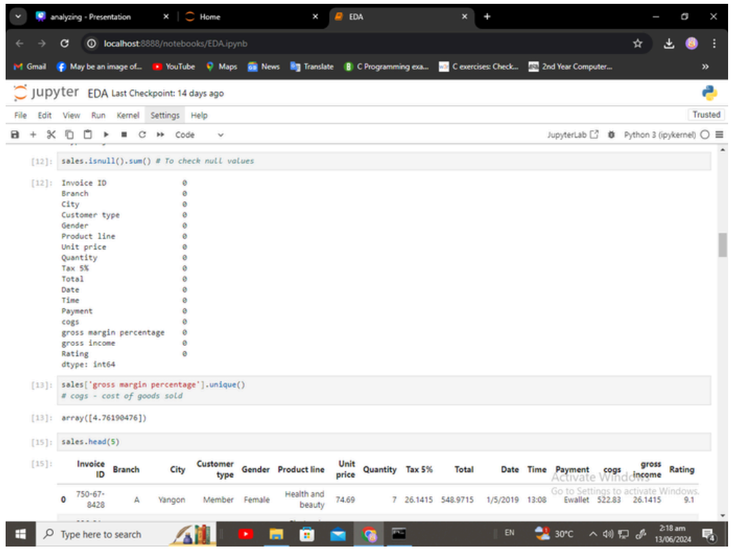
Statistical Analysis:

- Summary Statistics: Calculated mean, median, mode, and standard deviation to summarize data

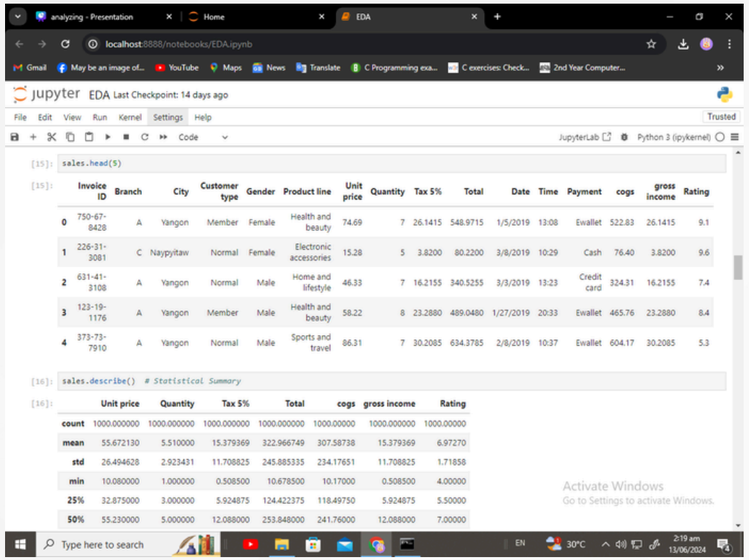
CODE


DATA EXPLORATION AND CLEANING
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sales = pd.read_csv('supermarket_sales - Sheet1.csv')
sales.head()


sales.shape # To check number of rows and columns
(1000, 17)
sales.columns
Index(['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender',
'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Total', 'Date',
'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income',
'Rating'],
dtype='object')
sales.dtypes
Invoice ID object
Branch object
City object
Customer type object
Gender object
Product line object
Unit price float64
Quantity int64
Tax 5% float64
Total float64
Date object
Time object
Payment object
cogs float64
gross margin percentage float64
gross income float64
Rating float64
dtype: object
sales.isnull().sum() # To check null values


sales.isnull().sum() # To check null values
Invoice ID 0
Branch 0
City 0
Customer type 0
Gender 0
Product line 0
Unit price 0
Quantity 0
Tax 5% 0
Total 0
Date 0
Time 0
Payment 0
cogs 0
gross margin percentage 0
gross income 0
Rating 0
dtype: int64
sales['gross margin percentage'].unique()
# cogs - cost of goods sold
array([4.76190476])
sales.head(5)

CODE

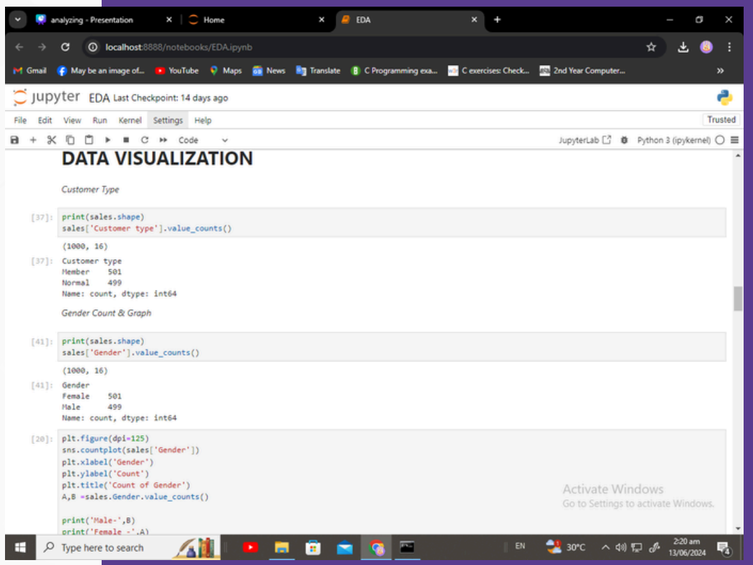


```
[15]: sales.head(5)
```

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross income	Rating
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	13:08	Ewallet	522.83	26.1415	9.1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29	Cash	76.40	3.8200	9.6
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	13:23	Credit card	324.31	16.2155	7.4
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	20:33	Ewallet	465.76	23.2880	8.4
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ewallet	604.17	30.2085	5.3

```
[16]: sales.describe()  # Statistical Summary
```

	Unit price	Quantity	Tax 5%	Total	cogs	gross income	Rating
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1000.000000	1000.00000
mean	55.672130	5.510000	15.379369	322.966749	307.58738	15.379369	6.97270
std	26.494628	2.923431	11.708825	245.885335	234.17651	11.708825	1.71858
min	10.080000	1.000000	0.508500	10.678500	10.17000	0.508500	4.00000
25%	32.875000	3.000000	5.924875	124.422375	118.49750	5.924875	5.50000
50%	55.230000	5.000000	12.088000	253.848000	241.76000	12.088000	7.00000



DATA VISUALIZATION

Customer Type

```
[37]: print(sales.shape)
      sales['Customer type'].value_counts()
```

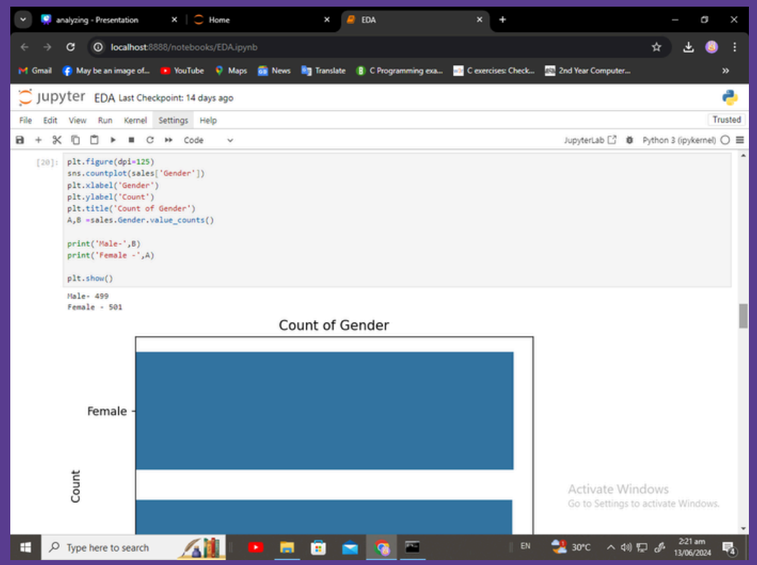
```
(1000, 16)
Customer type
Member    501
Normal    499
Name: count, dtype: int64
```

Gender Count & Graph

```
[41]: print(sales.shape)
      sales['Gender'].value_counts()
```

```
(1000, 16)
Gender
Female    501
Male      499
Name: count, dtype: int64
```

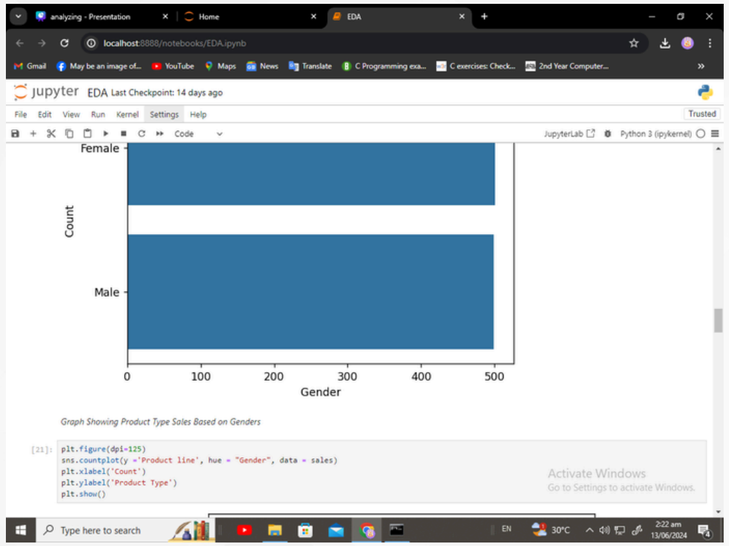
```
[20]: plt.figure(dpi=125)
      sns.countplot(sales['Gender'])
      plt.xlabel('Gender')
      plt.ylabel('Count')
      plt.title('Count of Gender')
      A,B =sales.Gender.value_counts()

      print('Male-',B)
      print('Female -',A)
```

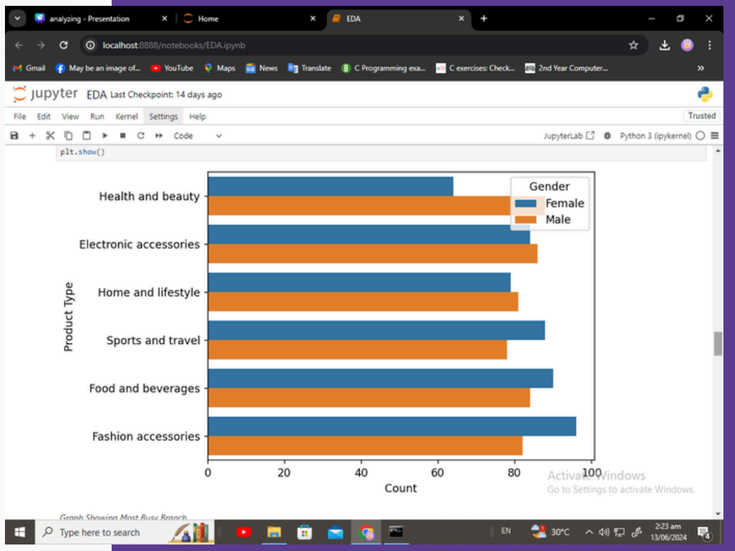


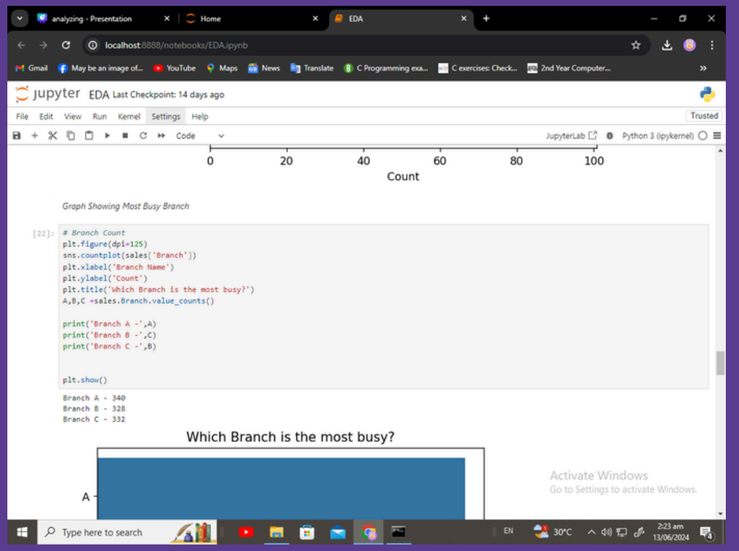
```
[20]: plt.figure(dpi=125)
      sns.countplot(sales['Gender'])
      plt.xlabel('Gender')
      plt.ylabel('Count')
      plt.title('Count of Gender')
      A,B =sales.Gender.value_counts()

      print('Male-',B)
      print('Female -',A)

      plt.show()
```

```
Male- 499
Female - 501
```

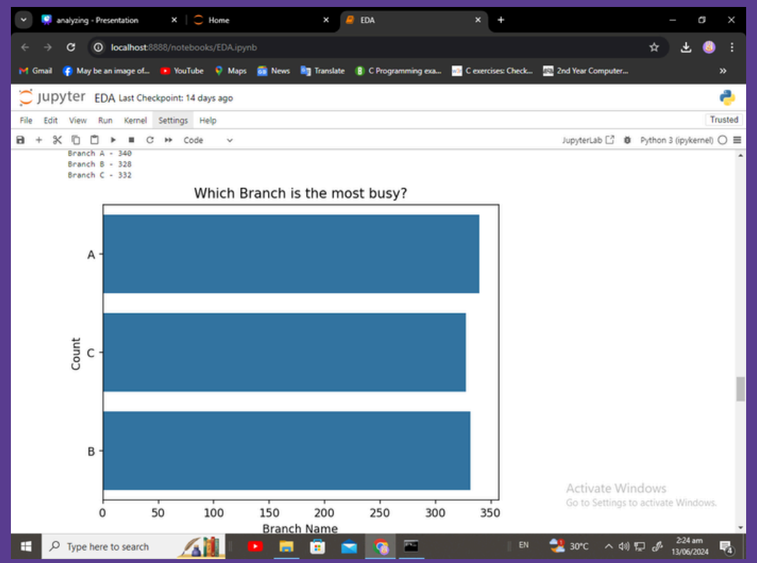
CODE

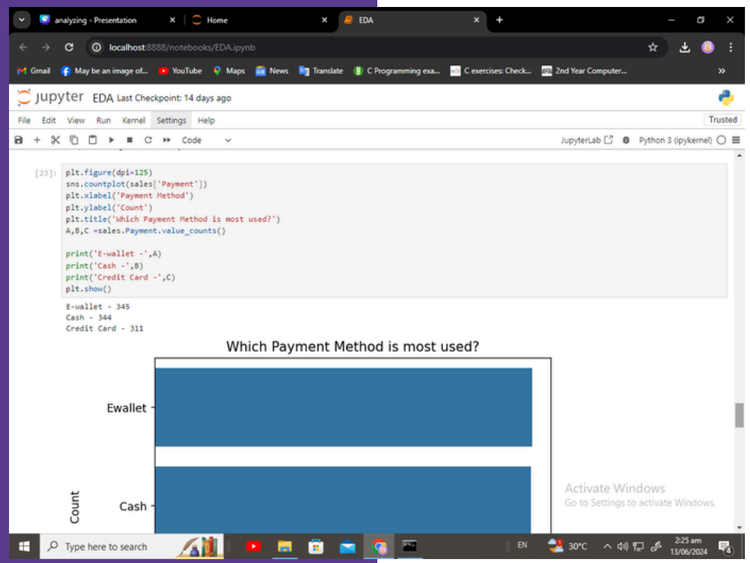

Female
Male
Count
0
100
200
300
400
500
Gender
Graph Showing Product Type Sales Based on Genders
[21]:
plt.figure(dpi=125)
sns.countplot(y ='Product line', hue = "Gender", data = sales)
plt.xlabel('Count')
plt.ylabel('Product Type')
plt.show()

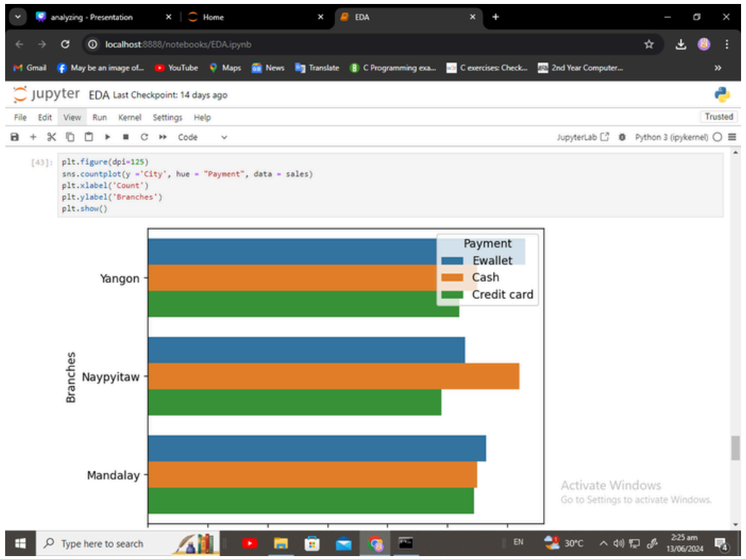

plt.show()
Health and beauty
Electronic accessories
Home and lifestyle
Sports and travel
Food and beverages
Fashion accessories
Product Type
Gender
Female
Male
0
20
40
60
80
100
Count


0
20
40
60
80
100
Count
Graph Showing Most Busy Branch
[22]:
# Branch Count
plt.figure(dpi=125)
sns.countplot(sales['Branch'])
plt.xlabel('Branch Name')
plt.ylabel('Count')
plt.title('Which Branch is the most busy?')
A,B,C =sales.Branch.value_counts()
print('Branch A -',A)
print('Branch B -',C)
print('Branch C -',B)
plt.show()
Branch A - 340
Branch B - 328
Branch C - 332
Which Branch is the most busy?
A

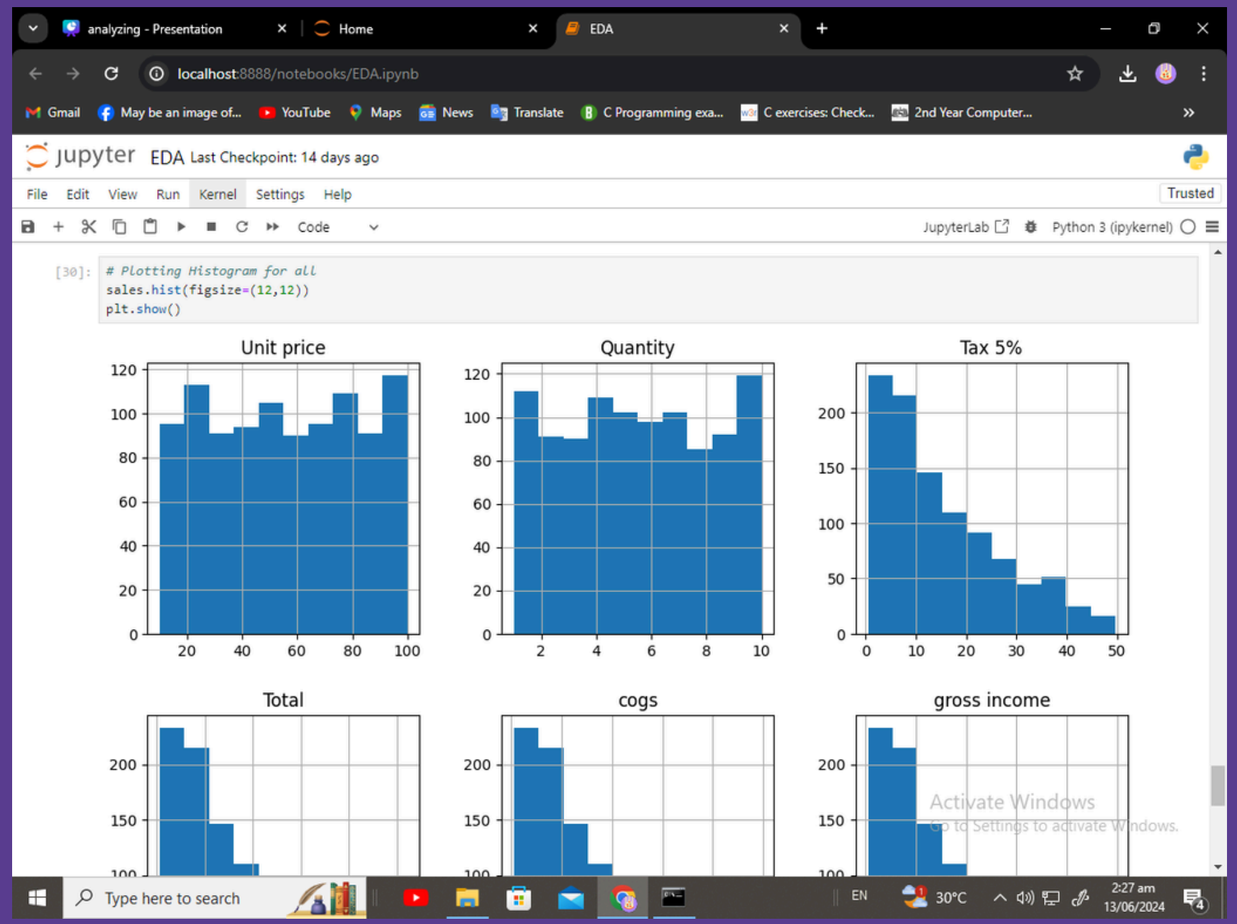
CODE

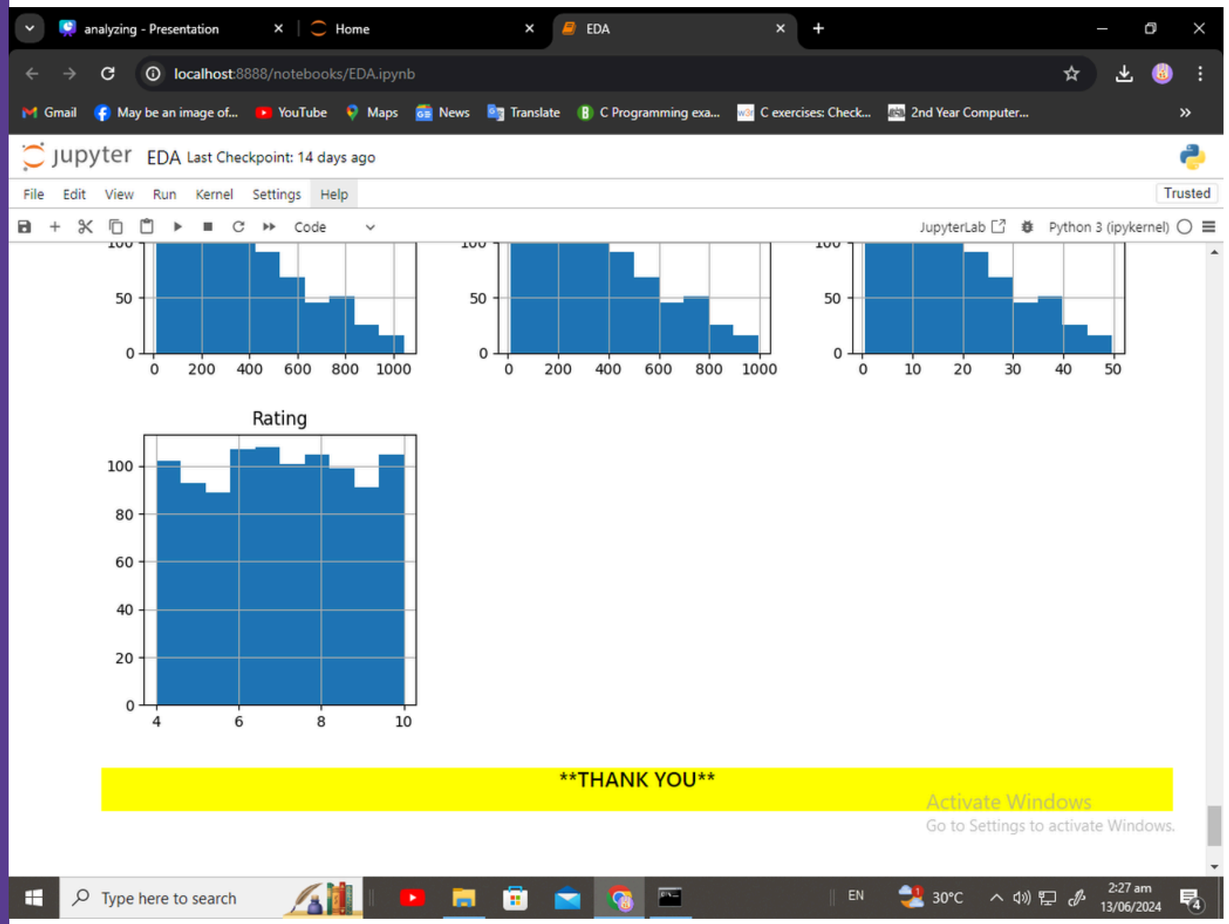

Branch A - 340
Branch B - 328
Branch C - 332
Which Branch is the most busy?
A
C
B
Count
0
50
100
150
200
250
300
350
Branch Name


[23]:
plt.figure(dpi=125)
sns.countplot(sales['Payment'])
plt.xlabel('Payment Method')
plt.ylabel('Count')
plt.title('Which Payment Method is most used?')
A,B,C =sales.Payment.value_counts()
print('E-wallet -',A)
print('Cash -',B)
print('Credit Card -',C)
plt.show()
E-wallet - 345
Cash - 344
Credit Card - 311
Which Payment Method is most used?
Ewallet
Cash
Count


[43]:
plt.figure(dpi=125)
sns.countplot(y ='City', hue = "Payment", data = sales)
plt.xlabel('Count')
plt.ylabel('Branches')
plt.show()
Payment
Ewallet
Cash
Credit card
Yangon
Naypyitaw
Mandalay
Branches

CODE


analyzing - Presentation
Home
EDA
localhost:8888/notebooks/EDA.ipynb
jupyter EDA Last Checkpoint: 14 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[30]:
# Plotting Histogram for all
sales.hist(figsize=(12,12))
plt.show()
Unit price
Quantity
Tax 5%
Total
cogs
gross income
Activate Windows
Go to Settings to activate Windows.
Type here to search
EN
30°C
2:27 am
13/06/2024


analyzing - Presentation
Home
EDA
localhost:8888/notebooks/EDA.ipynb
jupyter EDA Last Checkpoint: 14 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Rating
THANK YOU
Activate Windows
Go to Settings to activate Windows.
Type here to search
EN
30°C
2:27 am
13/06/2024

CONTRIBUTION OF MEMBERS

Ruba Inam Qureshi:

- Role: Data Collection and Initial Cleaning.
- Contribution: Gathered the dataset and performed initial data cleaning to ensure data quality.

Umm e Aimon:

- Role: Data Visualization Expert.
- Contribution: Created detailed visual plots and graphs to represent the data insights visually.

Anamta Irfan Mansoori:

- Role: Code Integration and Documentation.
- Contribution: Integrated all code, ensured consistency, and prepared the final notebook for presentation.

THANK YOU