```
# Import necessary libraries
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import dates
from datetime import datetime
```

Business Understanding

The objective of this analysis is to provide insights into the sales performance of Walmart across different cities and branches, understand the average price of items sold at each branch, and identify focus areas to improve sales for April 2019..

In this Assignment, we focused to answer the following questions:

A. Analyze the performance of sales and revenue at the city and branch level .

B. What is the average price of an item sold at each branch of the city.

C. Analyze the performance of sales and revenue, Month over Month across the Product line, Gender, and Payment Method, and identify the focus areas to get better sales for April 2019.

Data Understanding

The Walmart sales dataset comprises information on sales transactions recorded over a period of time. Key aspects of understanding the dataset include:

1. Data Composition:

- The dataset likely consists of rows and columns, with each row representing a unique sales transaction and each column representing a different attribute or feature of the transaction.
- Common columns may include:
- Invoice ID: Unique identifier for each transaction.
- Branch: Identifier for the branch of Walmart where the transaction occurred.
- City: The city where the Walmart branch is located.
- Customer type: Type of customer (e.g., member, non-member).
- Gender: Gender of the customer.
- Product line: Category or type of product purchased. - Unit price: Price of a single unit of the product.
- Quantity: Number of units of the product purchased.
- Date: Date of the transaction.
- Time: Time of the transaction.
- Payment: Method of payment used for the transaction.
- Rating: Customer rating or satisfaction score for the transaction.

1. Data Quality Issues:

- Missing Values: Check for missing values in any of the columns, which could impact the analysis.
- Data Consistency: Ensure consistency in data formats across columns (e.g., date format, gender categories).
- Outliers: Identify any unusual or extreme values that may skew the analysis.
- Data Errors: Look for any errors or inconsistencies in the data entries (e.g., negative quantities, invalid payment methods).

```
# Load dataset
df = pd.read_excel("Walmart_Sales.xlsx")
df['Date'] = pd.to_datetime(df['Date'])
```

Data Preparation

```
## Convert date to datetime format and show dataset information
df['Date'] =  pd.to_datetime(df['Date'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Invoice ID     1000 non-null   object
```

```
 1   Branch          1000 non-null   object
 2   City            1000 non-null   object
 3   Customer type   1000 non-null   object
 4   Gender          1000 non-null   object
 5   Product line    1000 non-null   object
 6   Unit price      1000 non-null   float64
 7   Quantity        1000 non-null   int64
 8   Date            1000 non-null   datetime64[ns]
 9   Time            1000 non-null   object
 10  Payment         1000 non-null   object
 11  Rating          1000 non-null   float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(8)
memory usage: 93.9+ KB
```

```
# checking for missing values
df.isnull().sum()
```

```
Invoice ID       0
Branch           0
City             0
Customer type    0
Gender           0
Product line     0
Unit price       0
Quantity         0
Date             0
Time             0
Payment          0
Rating           0
dtype: int64
```

```
# Splitting Date and create new columns (Day, Month, and Year)
df["Day"]= pd.DatetimeIndex(df['Date']).day
df['Month'] = pd.DatetimeIndex(df['Date']).month
df['Year'] = pd.DatetimeIndex(df['Date']).year
df
```

```
      Invoice ID Branch       City Customer type  Gender  \
0    750-67-8428      A     Yangon        Member  Female
1    226-31-3081      A  Naypyitaw        Normal  Female
2    631-41-3108      A     Yangon        Normal    Male
3    123-19-1176      B     Yangon        Member    Male
4    373-73-7910      C     Yangon        Normal    Male
..           ...    ...        ...           ...     ...
995  233-67-5758      A  Naypyitaw        Normal    Male
996  303-96-2227      A   Mandalay        Normal  Female
997  727-02-1313      A     Yangon        Member    Male
998  347-56-2442      B     Yangon        Normal    Male
999  849-09-3807      C     Yangon        Member  Female
```

```
                Product line  Unit price  Quantity       Date      Time
\
0          Health and beauty       74.69         7 2019-01-05  13:08:00

1    Electronic accessories       15.28         5 2019-03-08  10:29:00

2         Home and lifestyle       46.33         7 2019-03-03  13:23:00

3          Health and beauty       58.22         8 2019-01-27  20:33:00

4          Sports and travel       86.31         7 2019-02-08  10:37:00

..                       ...         ...       ...        ...       ...

995        Health and beauty       40.35         1 2019-01-29  13:46:00

996       Home and lifestyle       97.38        10 2019-03-02  17:16:00

997       Food and beverages       31.84         1 2019-02-09  13:22:00

998       Home and lifestyle       65.82         1 2019-02-22  15:33:00

999      Fashion accessories       88.34         7 2019-02-18  13:28:00


          Payment  Rating  Day  Month  Year
0         Ewallet     9.1    5      1  2019
1            Cash     9.6    8      3  2019
2    Credit card     7.4    3      3  2019
3         Ewallet     8.4   27      1  2019
4         Ewallet     5.3    8      2  2019
..            ...     ...  ...    ...   ...
995       Ewallet     6.2   29      1  2019
996       Ewallet     4.4    2      3  2019
997          Cash     7.7    9      2  2019
998          Cash     4.1   22      2  2019
999          Cash     6.6   18      2  2019

[1000 rows x 15 columns]
```

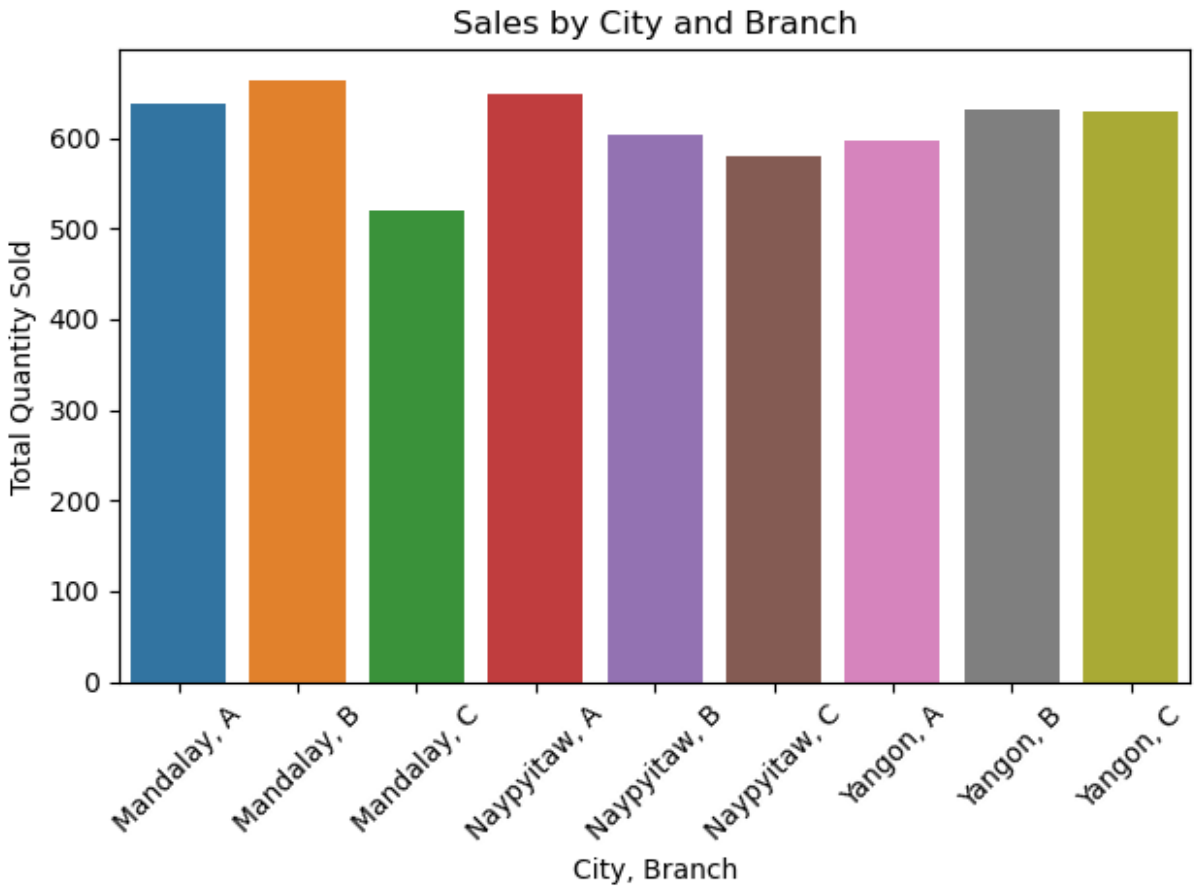
A. Analyze the performance of sales and revenue at the city and branch level

```
sales_revenue_city_branch = df.groupby(['City',
'Branch']).agg({'Quantity': 'sum', 'Unit price': 'sum'})
print("Performance of sales and revenue at the city and branch
level:")
print(sales_revenue_city_branch)
```

```
Performance of sales and revenue at the city and branch level:
                 Quantity  Unit price
City      Branch
Mandalay  A            637     6349.11
          B            664     6623.73
          C            519     5506.04
Naypyitaw A            648     5953.55
          B            604     6298.64
          C            579     6315.57
Yangon    A            598     6342.88
          B            631     6329.25
          C            630     5953.36
```

```
# Combine 'City' and 'Branch' into a single column for x-axis
sales_revenue_city_branch.reset_index(inplace=True)
sales_revenue_city_branch['City_Branch'] =
sales_revenue_city_branch['City'] + ', ' +
sales_revenue_city_branch['Branch']

# Plot
sns.barplot(x='City_Branch', y='Quantity',
data=sales_revenue_city_branch)
plt.title('Sales by City and Branch')
plt.xlabel('City, Branch')
plt.ylabel('Total Quantity Sold')
plt.xticks(rotation=45)
plt.tight_layout()  # Adjust layout to prevent overlapping labels
plt.show()
```



B. What is the average price of an item sold at each branch of the city

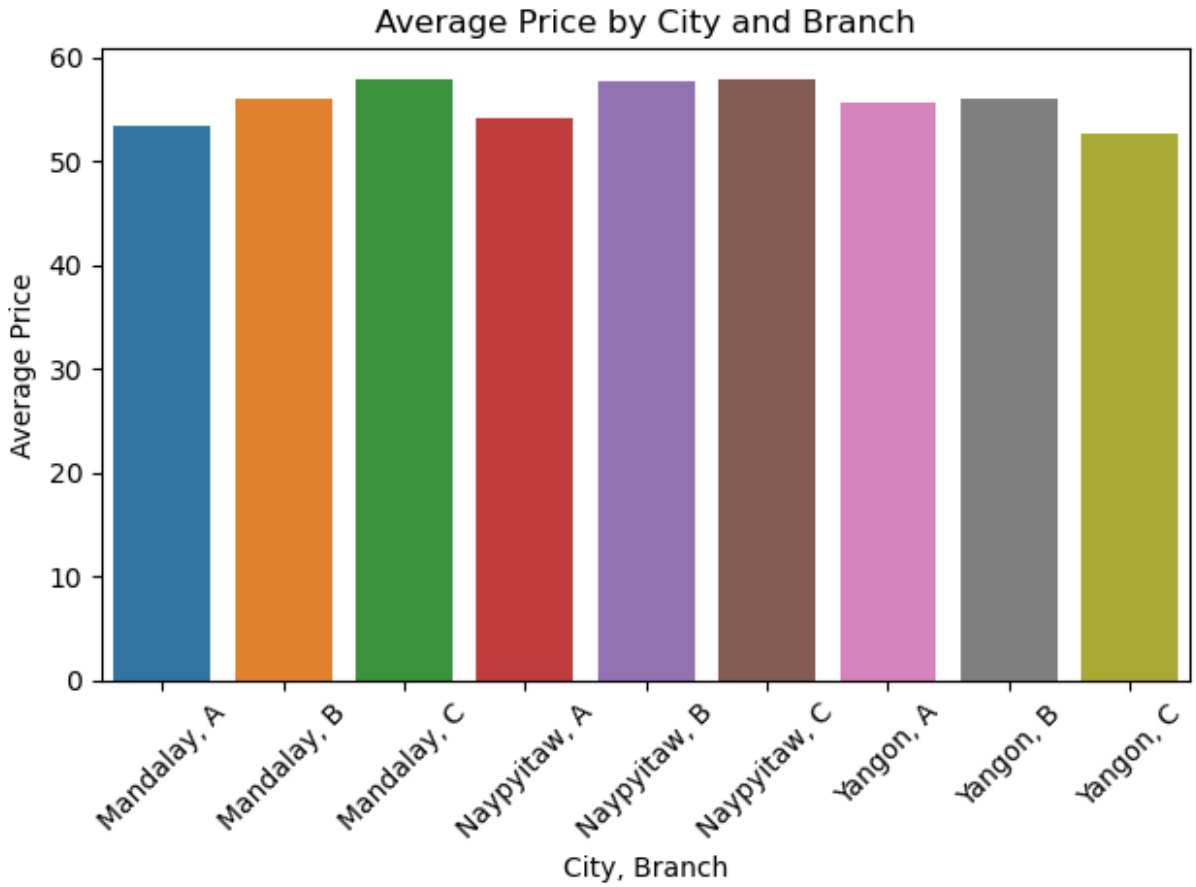
```
avg_price_branch_city = df.groupby(['City', 'Branch']).agg({'Unit
price': 'mean'})
print("\nAverage price of an item sold at each branch of the city:")
print(avg_price_branch_city)


Average price of an item sold at each branch of the city:
                 Unit price
City      Branch
Mandalay  A         53.353866
          B         56.133305
          C         57.958316
Naypyitaw A         54.123182
          B         57.785688
          C         57.941009
Yangon    A         55.639298
          B         56.011062
          C         52.684602

# Combine 'City' and 'Branch' into a single column for x-axis
avg_price_branch_city.reset_index(inplace=True)
```

```
avg_price_branch_city['City_Branch'] = avg_price_branch_city['City'] +
', ' + avg_price_branch_city['Branch']

# Plot
sns.barplot(x='City_Branch', y='Unit price',
data=avg_price_branch_city)
plt.title('Average Price by City and Branch')
plt.xlabel('City, Branch')
plt.ylabel('Average Price')
plt.xticks(rotation=45)
plt.tight_layout()  # Adjust layout to prevent overlapping labels
plt.show()
```




```
### C. Analyze the performance of sales and revenue, Month over Month
across Product line, Gender, and Payment Method
###     Focus areas to improve sales for April 2019.

# C. Analyze the performance of sales and revenue, Month over Month
across Product line, Gender, and Payment Method
#     Focus areas to improve sales for April 2019

# Extract month and year from the 'Date' column
```

```
df['Month'] = pd.to_datetime(df['Date']).dt.month
df['Year'] = pd.to_datetime(df['Date']).dt.year

# Filter data for April 2019
april_2019_data = df[(df['Month'] == 4) & (df['Year'] == 2019)]

# Group by Product line, Gender, and Payment Method
april_sales_analysis = april_2019_data.groupby(['Product line',
'Gender', 'Payment']).agg({'Quantity': 'sum', 'Unit price': 'sum'})
print("\nAnalysis of sales and revenue for April 2019:")
print(april_sales_analysis)
```

```
Analysis of sales and revenue for April 2019:
Empty DataFrame
Columns: [Quantity, Unit price]
Index: []
```

2. App Exploration:

Explore the features and user experience of the Jar app. Identify two aspects that you think could be significantly improved and explain your reasoning behind each suggestion.

Explanation For App Exploration

""" Explore the features and user experience of the Jar app:

The Jar app is designed to simplify expense and savings management by allowing users to allocate funds into different "jars" or categories. Users can set budgets, track spending, and visualize financial goals. However, there are two key areas where the app could significantly improve:

1. Streamlined User Onboarding:
   - The current onboarding process may overwhelm new users, potentially leading to drop-offs. To enhance user experience, the app could streamline onboarding by breaking it into smaller, manageable steps. Providing clear instructions and guidance at each stage can help users set up their accounts and navigate the app efficiently. Additionally, integrating interactive tutorials or tooltips can further assist users in understanding the app's features.
2. Enhanced Budgeting Features:
   - While the Jar app offers budgeting functionality, there's room for improvement to make it more versatile and personalized. Introducing customizable budget categories would empower users to create specific budgets for various expenses or savings objectives. Moreover, incorporating predictive budgeting, where the app analyzes past spending patterns to propose future budgets, can facilitate proactive financial management. Furthermore, integrating notifications or alerts to notify users when they approach or exceed budget limits can encourage better financial decision-making.

Addressing these aspects will enhance user satisfaction, retention, and overall usability of the Jar app, resulting in a more positive user experience. """

3. Product Optimisation:

The Jar app has an engagement feature called 'Spin to Win'. Right now, if 100 people come to the app each day, only 23 of them try out this spinning game. But, we know that people who spin are more likely to retain on the app and do transactions. Now, we want to get more people to play the game. So, the question is, how can we make sure that at least 50 people out of every 100 who visit the app each day will play 'Spin to Win'? What can we do to get more people interested in spinning the wheel?

Explanation For Product Optimisation:

""" Product Optimization:

The Jar app's 'Spin to Win' feature currently has a participation rate of only 23% out of 100 daily visitors. However, engaging with this feature has shown to increase user retention and transactions. To boost participation to at least 50%, the following strategies can be implemented:

1. Enhance Rewards and Incentives:
   - Offer more enticing rewards such as discounts, exclusive deals, loyalty points, or big prizes. Valuable rewards will motivate users to play the 'Spin to Win' game.
2. Promotional Campaigns:
   - Launch targeted marketing campaigns via email, push notifications, and social media to raise awareness about the 'Spin to Win' feature. Emphasize the benefits and rewards to attract more users.
3. Improve User Interface and Experience:
   - Enhance the visual appeal and interactivity of the 'Spin to Win' feature with smooth spinning mechanisms, attractive graphics, and clear instructions. A seamless user experience will encourage more users to participate.
4. Personalization and Customization:
   - Personalize the 'Spin to Win' experience based on user preferences, behavior, and demographics. Offer customized rewards and allow users to tailor their spinning experience.
5. Limited-Time Offers and Events:
   - Introduce time-sensitive promotions, themed events, or flash sales associated with the 'Spin to Win' game to create a sense of urgency and exclusivity. Limited-time offers can drive higher participation rates.
6. Social Sharing and Virality:
   - Enable social sharing functionalities to encourage users to share their winnings and experiences on social media platforms. Incentivize sharing and leverage user-generated content to amplify visibility.

Implementing these strategies will help the Jar app achieve its goal of increasing participation in the 'Spin to Win' game, leading to greater user engagement and retention.

"""