Analysis of Sales Dataset

Grace Imanuela Ruttun Karoma'

Analysis Background (1)

Introduction Sales is an exchange process of goods or services for currency. This analyst explores the sales patterns and behavior of various customers.

Contextual Framework Sales historically involved trading a thing for another thing, known as “bartering.” But nowadays we use money to get things or services.

Rationale This analyst aims to gain knowledge about the sales patterns and behaviors of the customers

Analysis Background (2)

Object and Research Questions

- How is the sales pattern based on the days?
- Which gender has the biggest sales?
- Which quarter has the biggest sales?
- Which generation has the biggest sales, and what product categories do they spend the most on?
- Which product category has the biggest sales?
- How is the sales growth?

Dataset

The dataset is from **Kaggle.com**.

There is a sales dataset table with 8 columns, there are:

- Unnamed: 0 = Auto index
- Date = The date the transaction occurred
- Gender = The gender of the customers
- Age = The age of the customers
- Product Category = The category of the products
- Quantity = The number of products purchased
- Price per unit = Price of the single products
- Total Amount = Total sales of the products (quantity x price per unit)

There are **1000 rows** and **8 columns**

Dataset

Dataset table preview

	Unnamed: 0	Date	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	0	2023-11-24	Male	34	Beauty	3	50	150
1	1	2023-02-27	Female	26	Clothing	2	500	1000
2	2	2023-01-13	Male	50	Electronics	1	30	30
3	3	2023-05-21	Male	37	Clothing	1	500	500
4	4	2023-05-06	Male	30	Beauty	2	50	100
5	5	2023-04-25	Female	45	Beauty	1	30	30
6	6	2023-03-13	Male	46	Clothing	2	25	50
7	7	2023-02-22	Male	30	Electronics	4	25	100
8	8	2023-12-13	Male	63	Electronics	2	300	600
9	9	2023-10-07	Female	52	Clothing	4	50	200
10	10	2023-02-14	Male	23	Clothing	2	50	100

Tools & Libraries

Tools

Libraries

Data Exploration

Data Anomalies

Column Names

There are 8 columns in this table. And there is one column that has an unnamed name.

```
# See the columns name
df.columns
```

```
Index(['Unnamed: 0', 'Date', 'Gender', 'Age', 'Product Category', 'Quantity',
       'Price per Unit', 'Total Amount'],
      dtype='object')
```

That is why the unnamed column name changed to the **"Row Number"** column.

```
# Changing unknown column's name
df_sales.rename(columns={'Unnamed: 0' : 'Row Number'}, inplace=True)
```

Column Datatypes

The 8 columns of data types have one column with incorrect datatypes.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Row Number        1000 non-null   int64
 1   Date              1000 non-null   object
 2   Gender            1000 non-null   object
 3   Age               1000 non-null   int64
 4   Product Category  1000 non-null   object
 5   Quantity          1000 non-null   int64
 6   Price per Unit    1000 non-null   int64
 7   Total Amount      1000 non-null   int64
dtypes: int64(5), object(3)
memory usage: 62.6+ KB
```

The column with incorrect datatypes is the **Date** column. The datatype should be a datetime datatype.

```
# Change Date column datatype from object to datetime
df_sales['Date'] = pd.to_datetime(df_sales['Date'])
```

```
1   Date              1000 non-null   datetime64[ns]
```

Column Values

The **Row Number** column values start with 0. Because there are 1000 rows, the last row number will be 999.

	Row Number	Date	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	0	2023-11-24	Male	34	Beauty	3	50	150
1	1	2023-02-27	Female	26	Clothing	2	500	1000
2	2	2023-01-13	Male	50	Electronics	1	30	30
3	3	2023-05-21	Male	37	Clothing	1	500	500
4	4	2023-05-06	Male	30	Beauty	2	50	100

Instead of starting with 0 and ending with 999. It will be better if the row number starts with 1 and ends with 1000

```
# Change row number values
df_sales['Row Number'] = df_sales['Row Number'] + 1
```

Null Values

Based on the data preview, there are no null values.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Row Number        1000 non-null   int64
 1   Date              1000 non-null   object
 2   Gender            1000 non-null   object
 3   Age               1000 non-null   int64
 4   Product Category  1000 non-null   object
 5   Quantity          1000 non-null   int64
 6   Price per Unit    1000 non-null   int64
 7   Total Amount      1000 non-null   int64
dtypes: int64(5), object(3)
memory usage: 62.6+ KB
```

```
# Count null values
df.isna().sum()
```

✓ 0.0s

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

Data Duplication

The duplication can occur for many reasons, like human error, system integration issues, data entry variations, merging data from many sources, etc.

Instead of dropping the duplicates immediately, it is better to show the duplicates to know the reason for the duplicates.

```
# Show duplicate
df_sales_duplicate = df_sales.duplicated()
```

```
# Count duplications
df_sales_duplicate.value_counts()
```

✓ 0.0s

```
False    1000
Name: count, dtype: int64
```

```
# Show all duplicates
print(df_sales_duplicate)
```

```
0      False
1      False
2      False
3      False
4      False
       ...
995    False
996    False
997    False
998    False
999    False
Length: 1000, dtype: bool
```

Data Anomalies Conclusions

- The unnamed column changed to the **Row Number** column
- The date datatype from the object datatype changed to the **datetime datatype**
- The row number starting with 0 and ending with 999 changed to **1 to 1000**
- There are **no null values**
- There are **no data duplications**

Data Before and After

Before dealing with data anomalies

	Unnamed: 0	Date	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	0	2023-11-24	Male	34	Beauty	3	50	150
1	1	2023-02-27	Female	26	Clothing	2	500	1000
2	2	2023-01-13	Male	50	Electronics	1	30	30
3	3	2023-05-21	Male	37	Clothing	1	500	500
4	4	2023-05-06	Male	30	Beauty	2	50	100
5	5	2023-04-25	Female	45	Beauty	1	30	30
6	6	2023-03-13	Male	46	Clothing	2	25	50
7	7	2023-02-22	Male	30	Electronics	4	25	100
8	8	2023-12-13	Male	63	Electronics	2	300	600
9	9	2023-10-07	Female	52	Clothing	4	50	200

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Unnamed: 0        1000 non-null   int64
 1   Date              1000 non-null   object
 2   Gender            1000 non-null   object
 3   Age               1000 non-null   int64
 4   Product Category  1000 non-null   object
 5   Quantity          1000 non-null   int64
 6   Price per Unit    1000 non-null   int64
 7   Total Amount      1000 non-null   int64
dtypes: int64(5), object(3)
memory usage: 62.6+ KB
```

- Unnamed column **"Unnamed: 0"**
- Wrong datatype for Date column **"Object"**

Data Before and After

After dealing with data anomalies

	Row Number	Date	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	Male	34	Beauty	3	50	150
1	2	2023-02-27	Female	26	Clothing	2	500	1000
2	3	2023-01-13	Male	50	Electronics	1	30	30
3	4	2023-05-21	Male	37	Clothing	1	500	500
4	5	2023-05-06	Male	30	Beauty	2	50	100
5	6	2023-04-25	Female	45	Beauty	1	30	30
6	7	2023-03-13	Male	46	Clothing	2	25	50
7	8	2023-02-22	Male	30	Electronics	4	25	100
8	9	2023-12-13	Male	63	Electronics	2	300	600
9	10	2023-10-07	Female	52	Clothing	4	50	200

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Row Number        1000 non-null   int64
 1   Date              1000 non-null   datetime64[ns]
 2   Gender            1000 non-null   object
 3   Age               1000 non-null   int64
 4   Product Category  1000 non-null   object
 5   Quantity          1000 non-null   int64
 6   Price per Unit    1000 non-null   int64
 7   Total Amount      1000 non-null   int64
dtypes: datetime64[ns](1), int64(5), object(2)
memory usage: 62.6+ KB
```

- Change “Unnamed: 0” to “Row Number”
- Change date datatype from “Object” to “Datetime”

Data Exploration

Data Transformation

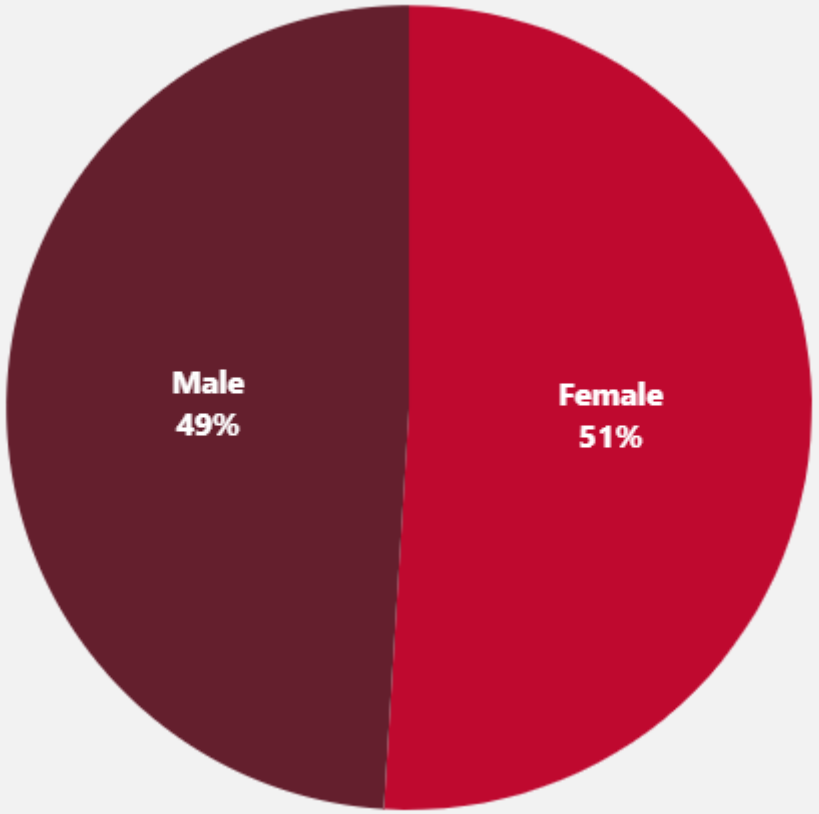
Days Sales

	sum	count	mean
	Quantity	Quantity	Quantity
Day			
Monday	385	146	2.636986
Tuesday	397	161	2.465839
Wednesday	356	139	2.561151
Thursday	301	123	2.447154
Friday	373	143	2.608392
Saturday	373	150	2.486667
Sunday	329	138	2.384058
Total	2514	1000	2.514000

Monday and Tuesday are the biggest sales by quantity, contributing **31%** of the total quantity.

It happened because there was an early product promotion, restock cycles, customers' weekly needs reset, customers' salary effect, and many more.

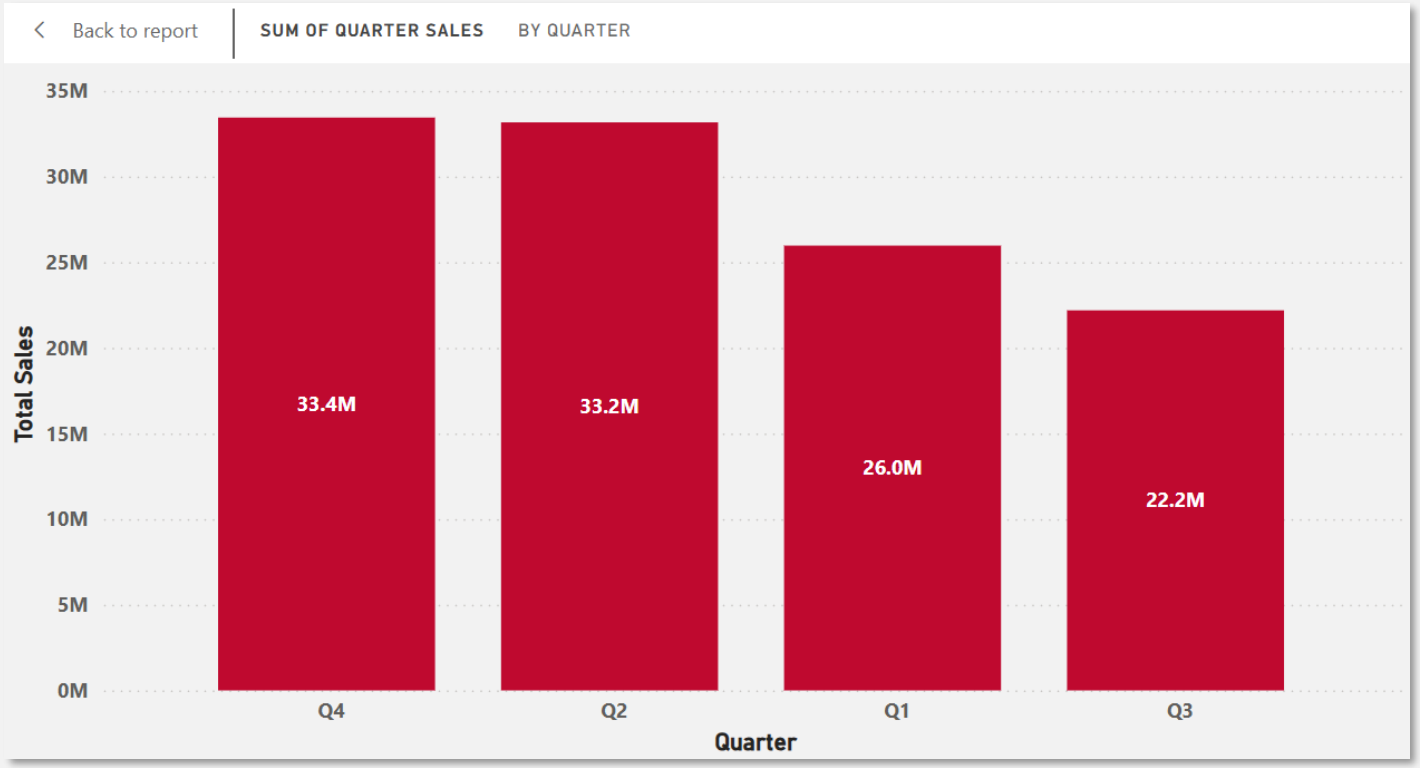
Gender Sales



Female customers contributed **51%** of the total customers, while male customers contributed **49%** of the total customers.

Gender	Count of Gender
Female	510
Male	490
Total	**1000**

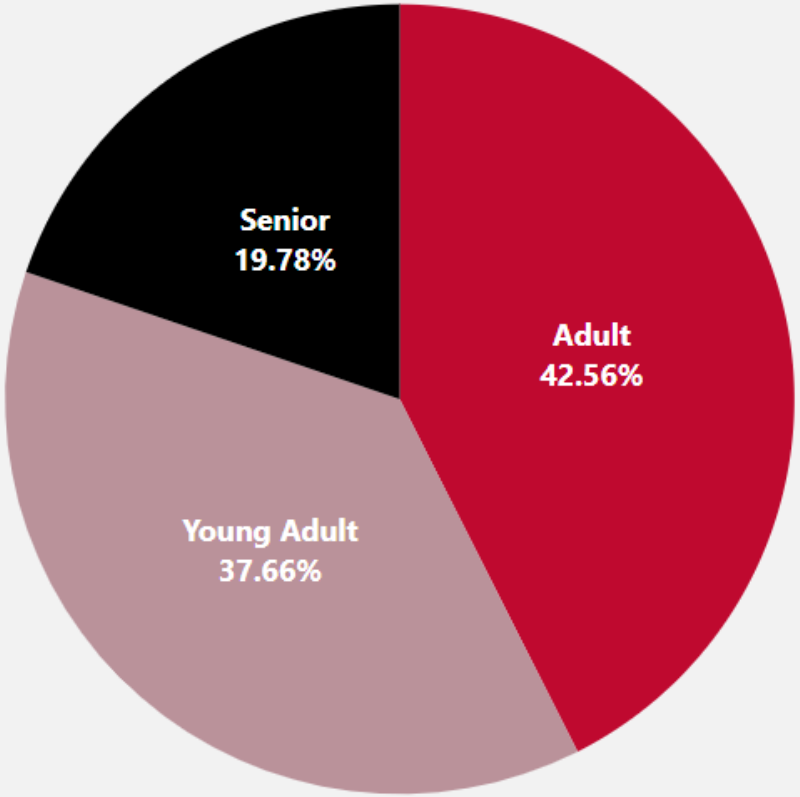
Quarter Sales



Q4 is the highest sales period, contributing **29%** of the total sales due to big holidays like Halloween, Thanksgiving, Black Friday, Christmas, and New Year's Eve.

Q2 contributed **28%**, also due to big holidays like Easter, Mother's Day, and Father's Day

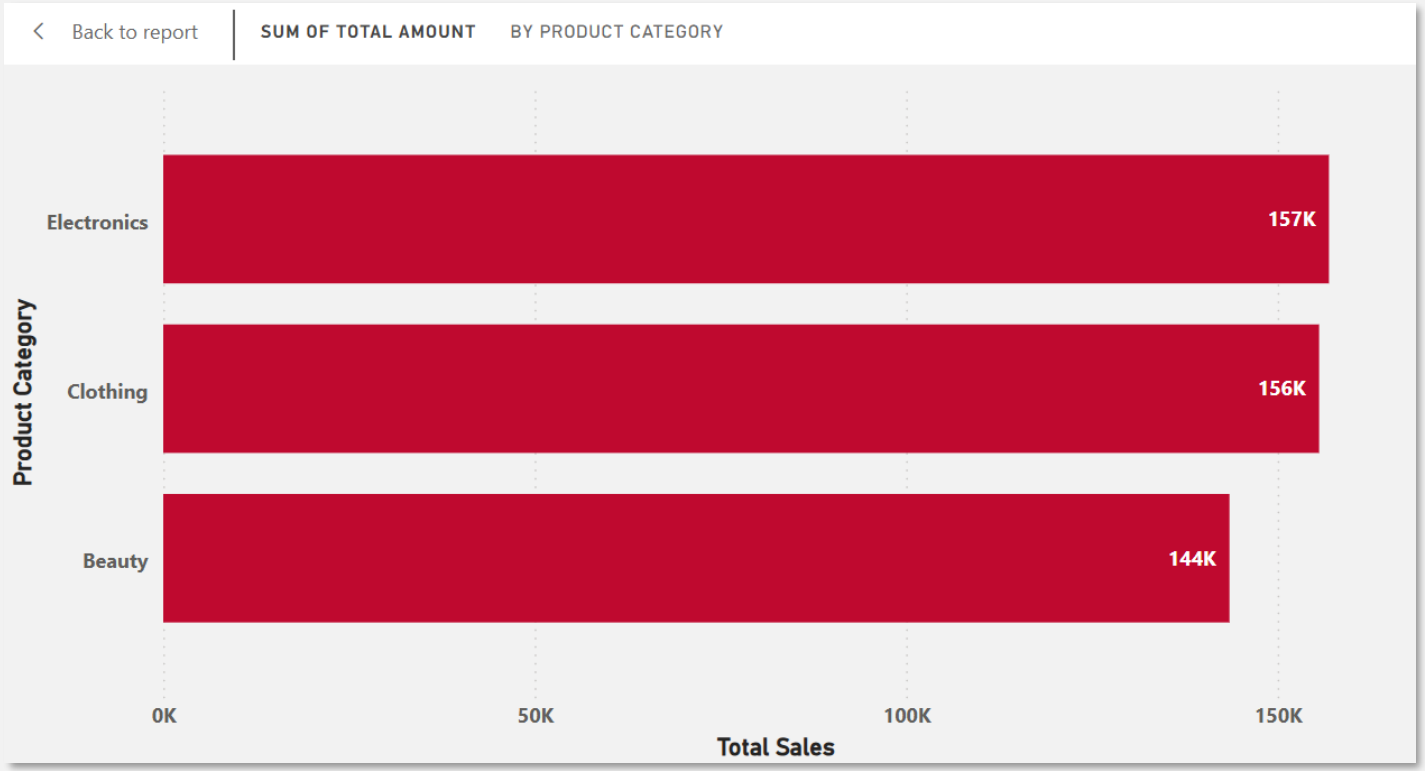
Age Segment



Adults are the highest spending age category, contributing **42.56%** of the total sales.

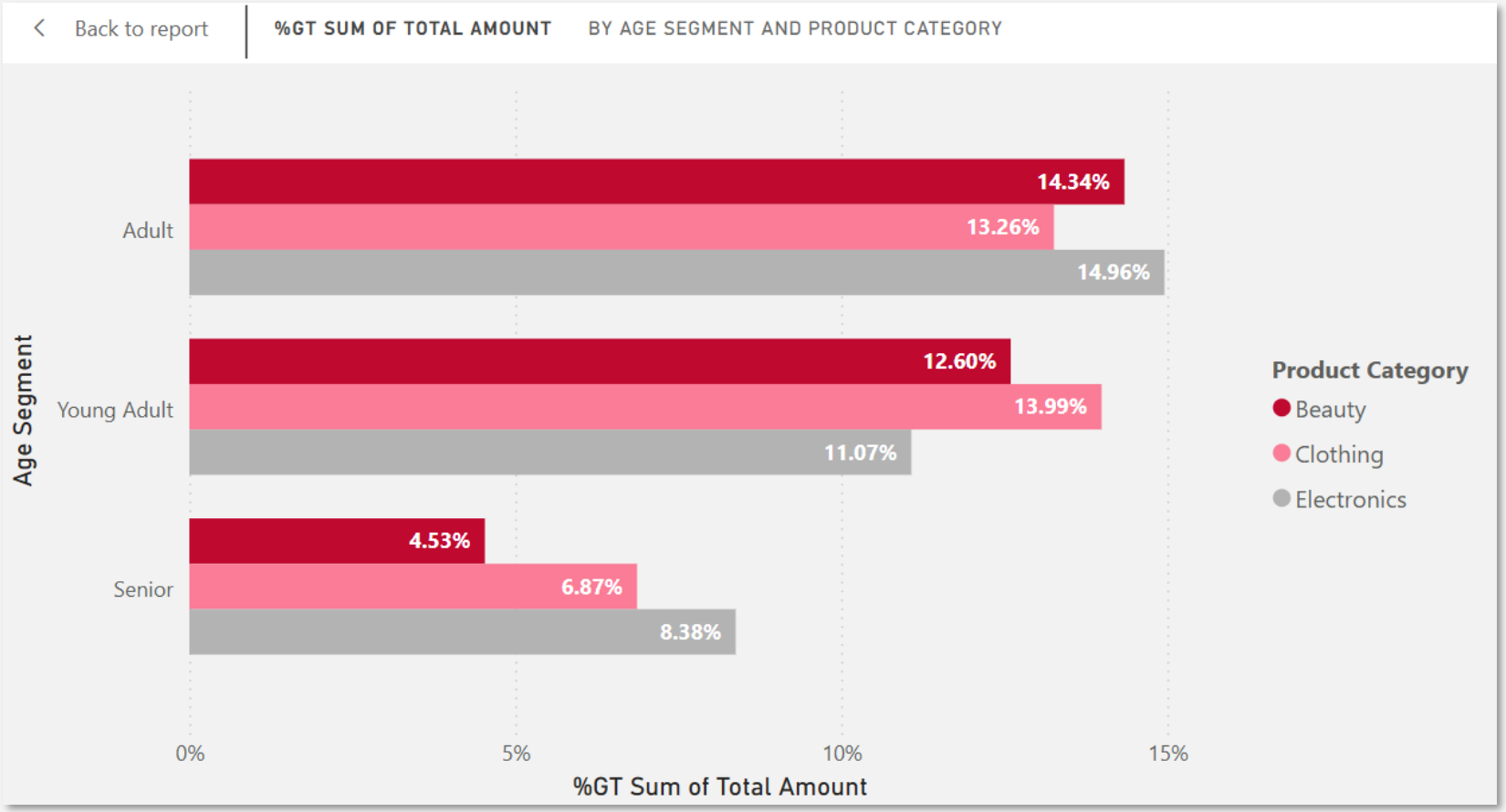
Adults are spending more because due to more responsibilities, such as supporting the household, children's lifestyle, children's education, and their own lifestyle.

Product Category



Electronics contributed **34%** of the total sales, making it the highest product category sales. It happened because of household expenses. And consistent with adults' spending.

Age Segment by Product Category



Adults tend to spend more on **Electronics** due to household needs.

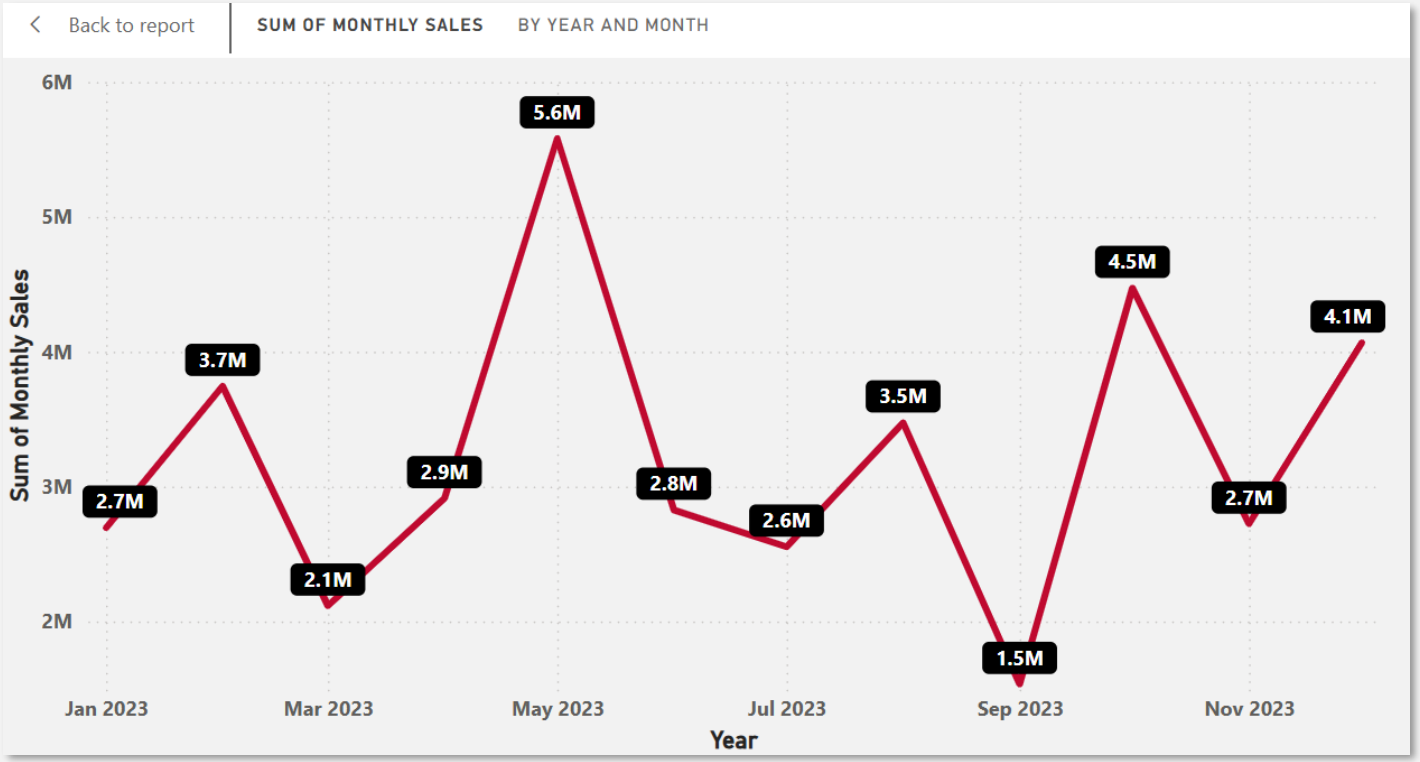
Young adults tend to spend more on **clothing** due to the influence of fashion trends.

Seniors tend to spend more on **electronics** for health tools and entertainment.

Monthly Growth (1)

May 2023 contributed **14.42%** of the yearly total sales.

It happened because online sales increased to **2.7%**, motor demand increased to **1.4%**, and demand for summer clothes and outdoor goods increased due to the summer season.

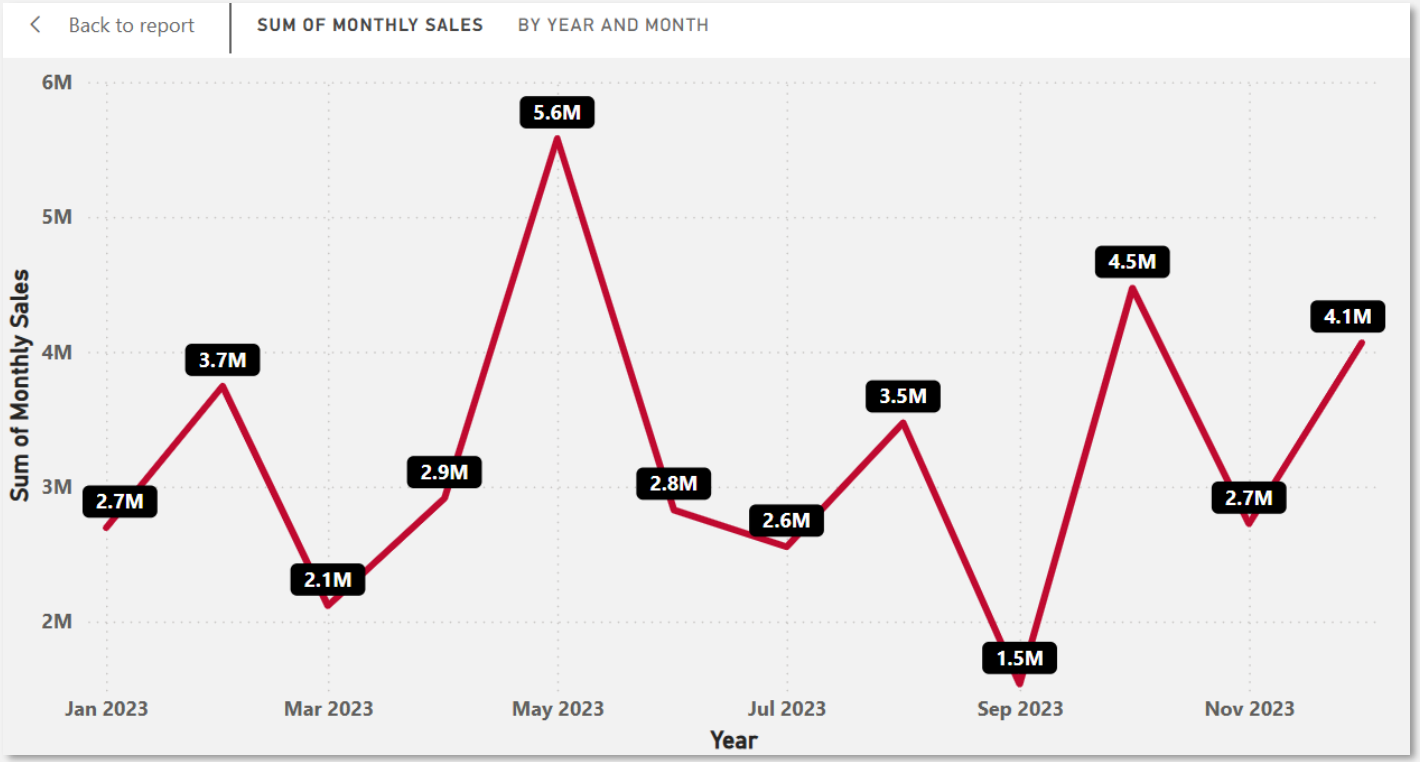


Monthly Growth (2)

In September 2023, sales dropped contributed only **3%** for the yearly sales.

There are a few categories were falling **0.9%** falling retail sales, **2.2%** online store falling, and **1.9%** falling non-food sales.

It happened due to the continuous cost-of-living pressure from autumn to winter.



Data Exploration

Data Dashboard

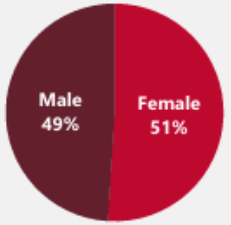
Sales Dashboard

456K
Sum of Total Amount

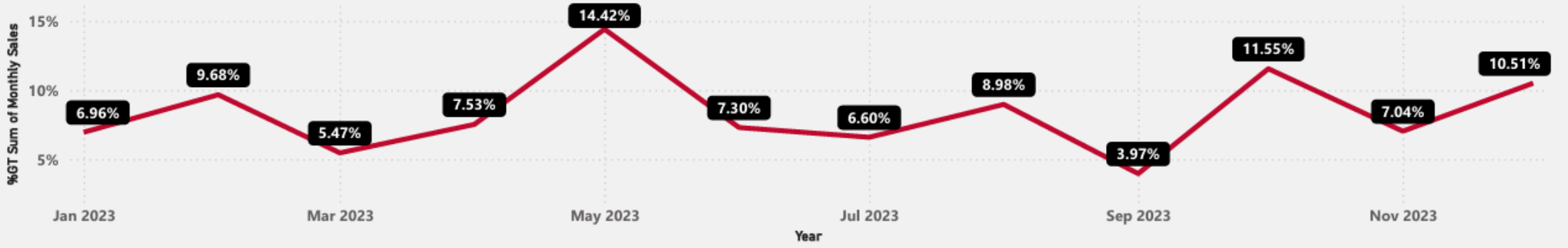
1000
Count of Row Number

3
Count of Product Category

Count of Row Number by Gender

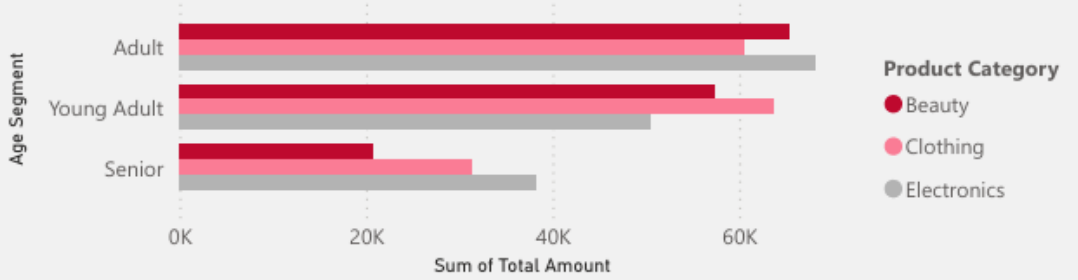


%GT Sum of Monthly Sales by Year and Month



Sum of Total Amount by Product Category

Sum of Total Amount by Age Segment and Product Category



Summary

- **Monday and Tuesday** share the biggest sales of the week
- **Females** contributed most of the sales, even though there isn't a big difference from males.
- **Q4** shares the biggest sales of the year
- **Adults (35 -54 years old)** share the biggest sales, and **electronics** are the most popular for adults
- **Electronics** share the biggest sales
- **May 2023** is the highest sales of the month, and **September 2023** is the lowest sales of the month

Thank You