""" Exploratory Data Analysis

1. Dataset Overview:
    - The dataset comprises 14,260 entries across 10 columns, representing various attributes related to product sales.
2. Sales Metrics:
    - The dataset includes two primary sales metrics: Quantity (QTY) and Value (VALUE), providing insights into the volume and monetary value of products sold.
    - Minimum quantity recorded is 0 units, indicating instances of no sales, while the maximum quantity is 641 units for a single transaction.
3. Temporal Distribution:
    - Sales data is categorized by month (MONTH), with the majority of entries distributed across three months: M1, M2, and M3.
4. Store Variation:
    - Sales transactions are recorded across multiple stores (STORECODE), with varying levels of sales activity observed across different store codes.
5. Product Categorization:
    - Products are classified into hierarchical categories, including Group Level (GRP), Subgroup Level (SGRP), and Sub-Subgroup Level (SSGRP), providing a structured taxonomy for analysis.
    - The top-selling product group is "CONFECTIONERY - TOTAL GUM," indicating high demand in this category.
6. Brand Information:
    - Each product is associated with company (CMP), mother brand (MBRD), and brand (BRD) attributes, enabling analysis of brand-level performance and market share.
7. Sales Patterns:
    - Analysis of sales metrics reveals diverse sales patterns, with a wide range of quantities and values recorded across different product categories and stores.
    - The top-selling product, "CENTER FRUIT," recorded a maximum quantity of 641 units in a single transaction.
8. Variability and Trends:
    - Examination of sales data reveals variability in sales metrics across different temporal periods, stores, and product categories, as well as potential trends or seasonality.
9. Insights for Decision-Making:
    - The dataset offers valuable insights for decision-making, including identifying high-performing products, optimizing inventory management, and targeting marketing efforts based on sales patterns and trends.
10. Opportunities for Improvement:
    - Analysis of sales data highlights opportunities for improvement, such as identifying underperforming products or product groups for strategic adjustments and enhancing sales strategies to capitalize on emerging trends. """

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
url="/content/Hackathon_Ideal_Data.csv.zip"
```

```
df=pd.read_csv(url)
```

```
df.columns
```

```
Index(['MONTH', 'STORECODE', 'QTY', 'VALUE', 'GRP', 'SGRP', 'SSGRP', 'CMP',
       'MBRD', 'BRD'],
      dtype='object')
```

GRP - Group Level

SGRP - Sub Group Level

SSGRP - Sub Sub Group Level

CMP - Company

MBRD - Mother Brand Level

BRD - Brand Level

```
df.head()
```

```
df.tail()
```

	MONTH	STORECODE	QTY	VALUE	GRP	SGRP	SSGRP	CMP	I
14255	M3	P10	0	0	SUGAR SUBSTITUTE (11/05)	POWDER (SUGAR SUBST)	POWDER (SUGAR SUBST)	ZYDUS WELLNESS LTD	SU F
14256	M3	P8	1	62	SUGAR SUBSTITUTE (11/05)	PELLETS (SUGAR SUBST)	PELLETS (SUGAR SUBST)	ZYDUS WELLNESS LTD	SU F
14257	M1	P6	0	0	SUGAR SUBSTITUTE (11/05)	PELLETS (SUGAR SUBST)	PELLETS (SUGAR SUBST)	ZYDUS WELLNESS LTD	SU F
14258	M1	P10	0	0	SUGAR SUBSTITUTE	POWDER (SUGAR	POWDER (SUGAR	ZYDUS WELLNESS	SU

```
df.sort_values(by=["MONTH"])
```

	MONTH	STORECODE	QTY	VALUE	GRP	SGRP	
0	M1	P1	25	83	HAIR CONDITIONERS	HAIR CONDITIONERS	COND
7106	M1	P3	1	0	AGARBATTI & DHOOPBATTI	DHOOPBATTI	DH
7105	M1	P3	16	196	AGARBATTI & DHOOPBATTI	AGARBATTI	A
7104	M1	P3	0	0	AGARBATTI & DHOOPBATTI	AGARBATTI	A
7103	M1	P3	4	36	AGARBATTI & DHOOPBATTI	AGARBATTI	A
...	...	...	...	...	...	...	
1758	M3	P7	1	0	SHAMPOO - BY SEGMENTS	BOTTLES / TUBES	BOTTLE
1759	M3	P7	0	0	SHAMPOO - BY SEGMENTS	BOTTLES / TUBES	BOTTLE
1760	M3	P7	13	822	SHAMPOO - BY SEGMENTS	BOTTLES / TUBES	BOTTLE

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14260 entries, 0 to 14259
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   MONTH      14260 non-null  object
 1   STORECODE  14260 non-null  object
 2   QTY        14260 non-null  int64 
 3   VALUE      14260 non-null  int64 
 4   GRP        14260 non-null  object
 5   SGRP       14260 non-null  object
 6   SSGRP      14260 non-null  object
 7   CMP        14260 non-null  object
 8   MBRD       14260 non-null  object
 9   BRD        14260 non-null  object
dtypes: int64(2), object(8)
memory usage: 1.1+ MB
```

```
df.isnull().sum()
```

```
MONTH        0
STORECODE    0
QTY          0
```

```
VALUE        0
GRP          0
SGRP         0
SSGRP        0
CMP          0
MBRD         0
BRD          0
dtype: int64
```

```
df.shape
```

```
(14260, 10)
```

```
df.describe()
```

	QTY	VALUE
count	14260.000000	14260.000000
mean	16.354488	294.455330
std	34.365583	760.129558
min	0.000000	0.000000
25%	1.000000	10.000000
50%	4.000000	99.000000
75%	16.000000	283.000000
max	641.000000	24185.000000

```
df["MONTH"].unique()
```

```
array(['M1', 'M3', 'M2'], dtype=object)
```

```
df["MONTH"].value_counts()
```

```
M2    4816
M1    4804
M3    4640
Name: MONTH, dtype: int64
```

```
df["STORECODE"].unique()
```

```
array(['P1', 'P2', 'P3', 'P4', 'P5', 'P6', 'P7', 'P8', 'P9', 'P10'],
      dtype=object)
```

```
df["STORECODE"].value_counts().sort_values(ascending=False)
```

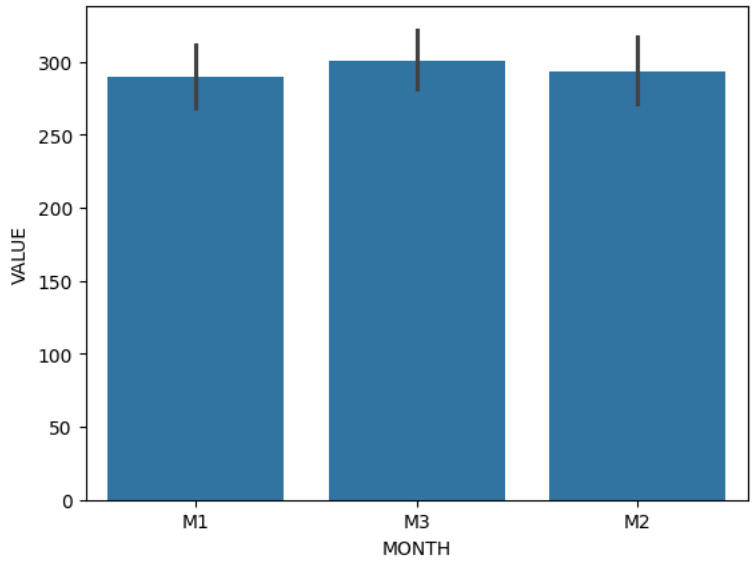
```
P8     2642
P6     2097
P10    1508
P2     1471
P4     1454
P3     1362
P9     1188
P1     1061
P7      850
P5      627
Name: STORECODE, dtype: int64
```

```
df[["GRP","SGRP","SSGRP","CMP","MBRD","BRD"]].nunique().sort_values(ascending=False)
```

```
BRD      1613
MBRD      818
CMP       512
SSGRP     242
SGRP      177
GRP        80
dtype: int64
```

```
sns.barplot(data = df,x="MONTH",y="VALUE")
```

```
<Axes: xlabel='MONTH', ylabel='VALUE'>
```

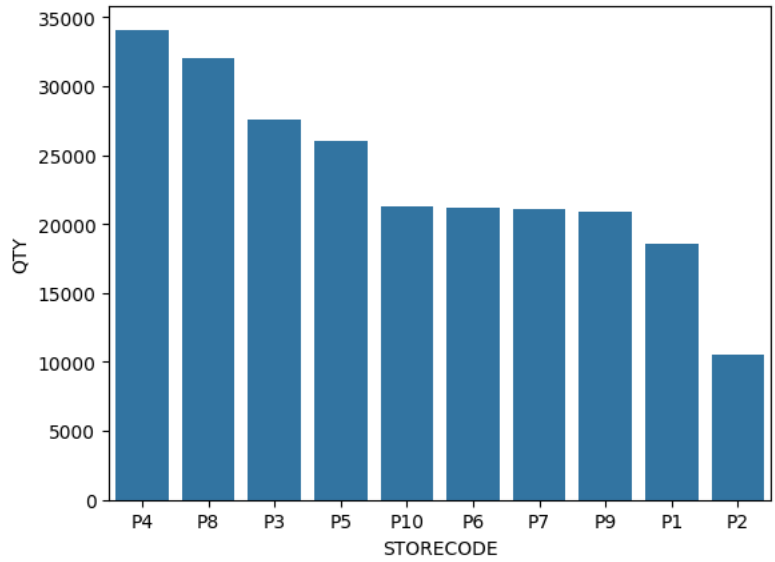



```
qty_storecode = df.groupby("STORECODE")["QTY"].sum().sort_values(ascending=False)
qty_storecode
```

```
STORECODE
P4     34089
P8     32003
P3     27602
P5     25993
P10    21323
P6     21178
P7     21114
P9     20904
P1     18526
P2     10483
Name: QTY, dtype: int64
```

```
sns.barplot(data=qty_storecode)
```

```
<Axes: xlabel='STORECODE', ylabel='QTY'>
```



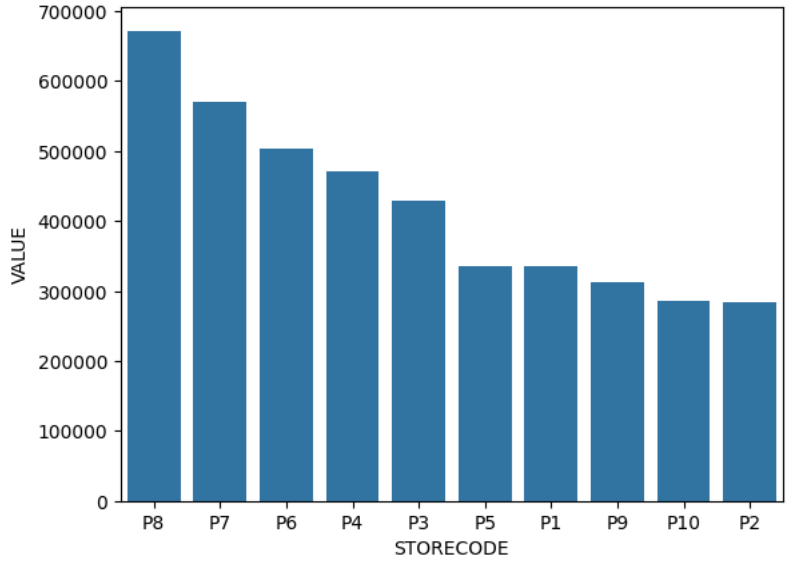

```
value_storecode= df.groupby("STORECODE")["VALUE"].sum().sort_values(ascending=False)
value_storecode
```

```
STORECODE
P8     671988
P7     571136
P6     502627
P4     471566
P3     428550
P5     335086
P1     334710
P9     313426
P10    285820
```

```
P2     284024
Name: VALUE, dtype: int64
```

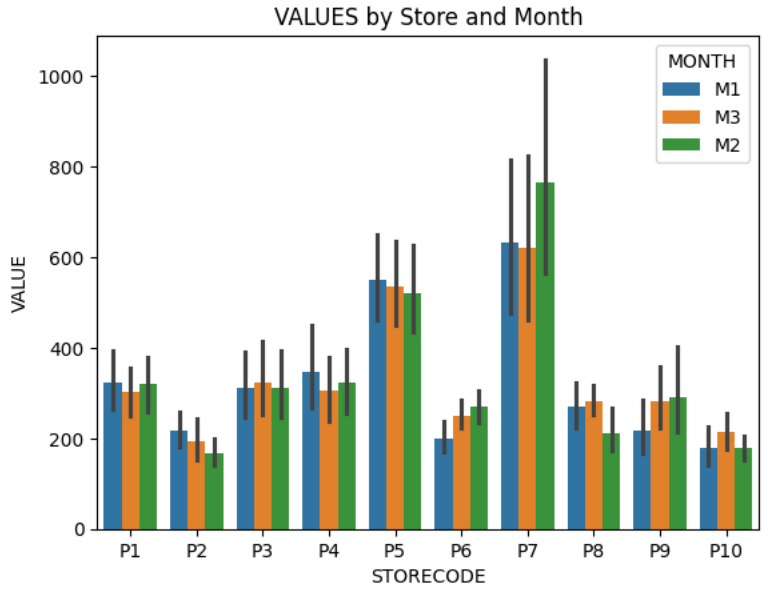
```
sns.barplot(data=value_storecode)
```

```
<Axes: xlabel='STORECODE', ylabel='VALUE'>
```



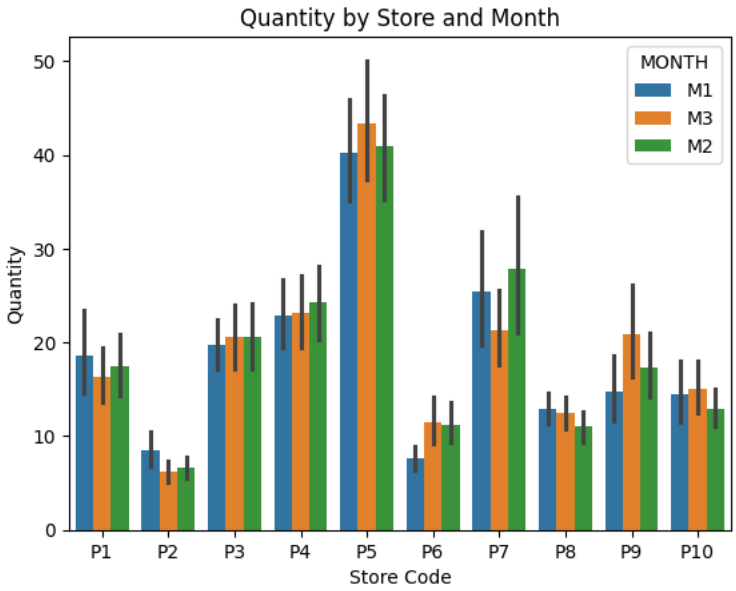

```
sns.barplot(data=df,x="STORECODE",y="VALUE",hue="MONTH")
plt.title("VALUES by Store and Month")
```

```
Text(0.5, 1.0, 'VALUES by Store and Month')
```




```
sns.barplot(data=df,x="STORECODE",y="QTY",hue="MONTH")
plt.title("Quantity by Store and Month")
plt.xlabel("Store Code")
plt.ylabel("Quantity")
```

```
Text(0, 0.5, 'Quantity')
```




```
#top ten
qty_sort = df.sort_values(by="QTY",ascending=False).head(10)
```

```
qty_sort
```

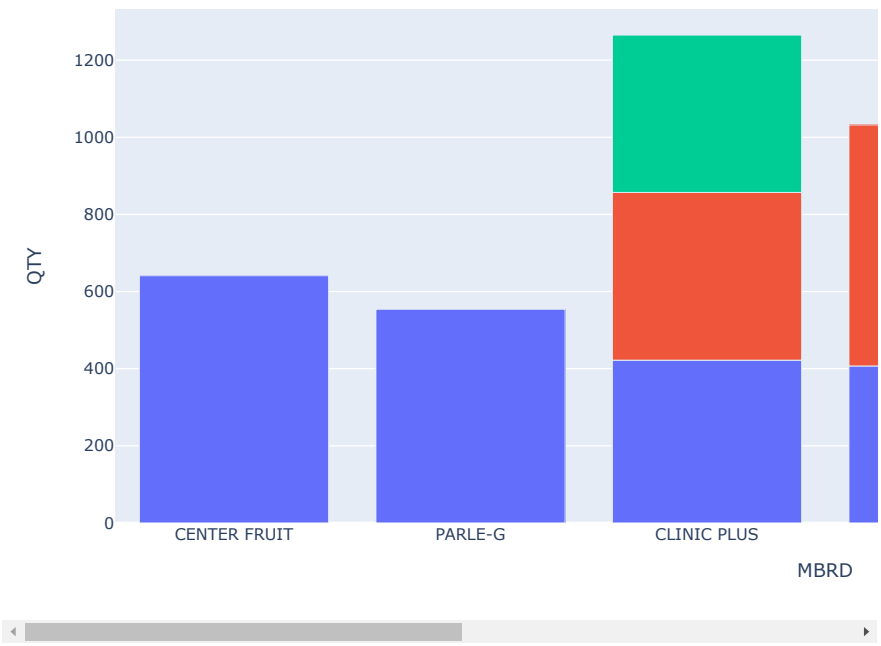
	MONTH	STORECODE	QTY	VALUE	GRP	SGRP	SSGRP	
9602	M1	P10	641	641	CONFECTIONERY - TOTAL GUM	BUBBLE GUMS	BUBBLE GUMS	PE VAI
4595	M3	P9	625	5126	PACKAGED TEA	MAIN PACKS	MAIN PACKS	HASM
4779	M3	P6	605	775	COFFEE	INSTANT COFFEE	INSTANT COFFEE	HIN UI
12861	M1	P1	554	2009	BISCUITS - CORE & NON CORE	GLUCOSE	GLUCOSE	
4750	M2	P6	541	990	COFFEE	INSTANT COFFEE	INSTANT COFFEE	HIN UI
1836	M3	P9	435	435	SHAMPOO - BY SEGMENTS	SACHETS	SACHETS	HIN UI
1579	M1	P7	422	422	SHAMPOO - BY SEGMENTS	SACHETS	SACHETS	HIN UI
7557	M2	P7	416	5383	ALL IODISED SALT	POWDERED SALT	POWDERED SALT	DIV P
								HIN

Next steps: Generate code with qty_sort View recommended plots

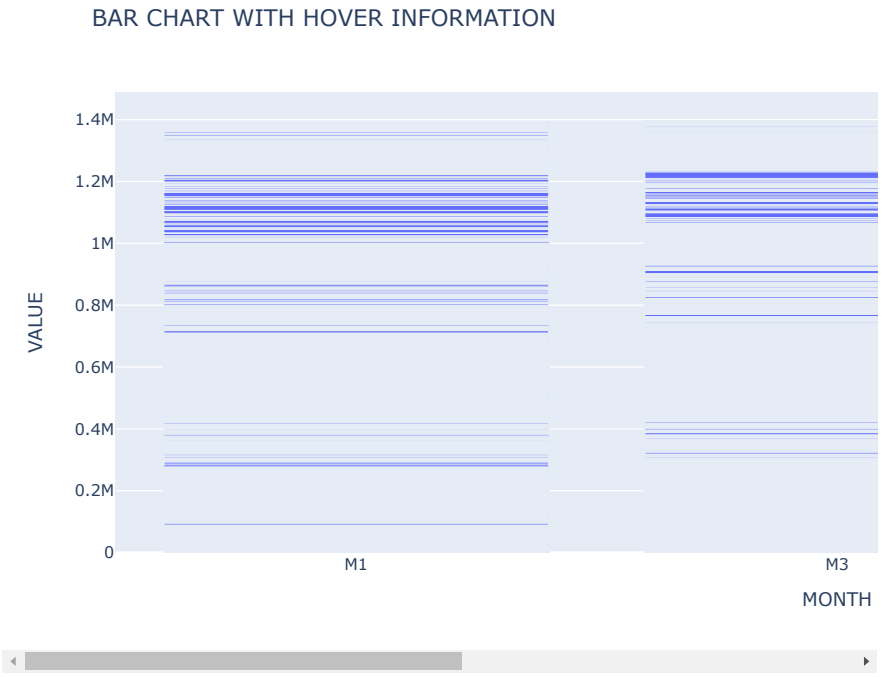
```
qty_sort
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7d6887e87760>
```

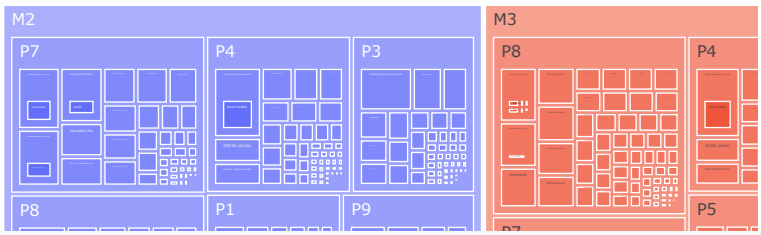
```
px.bar(qty_sort,x="MBRD",y="QTY",color="MONTH")
```




```
px.bar(df,x="MONTH",y="VALUE",hover_data={"STORECODE":True,"QTY":True,"GRP":True,"CMP":True,"MBRD":True,"BRD":True},title="BAR CHART WI
```




```
fig=px.treemap(df,path=["MONTH","STORECODE","GRP","SGRP"],values="VALUE")
fig.show()
```




```
fig=px.sunburst(df,path=["MONTH","STORECODE","GRP","SGRP"],values="VALUE")
fig.show()
```