Task: Analyze customer behavior based on transaction data

Develop a data analysis project that focuses on understanding and analyzing customer behavior based on transactional data. The goal is to perform Exploratory Data Analysis (EDA) to derive valuable insights into customer preferences, purchasing patterns, and overall behavior.

```
# Import Necessary Libraries:

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from mlxtend.frequent_patterns import apriori, association_rules


#Load the Transaction Data:
data = pd.read_csv("/content/Hackathon_Working_Data.csv")
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
  and should_run_async(code)
```

```
print('There are {} distinct data at Group level'.format(data['GRP'].nunique()))
print('There are {} distinct data at sub-Group level'.format(data['SGRP'].nunique()))
print('There are {} distinct data at sub-sub-Group level'.format(data['SSGRP'].nunique()))
print('There are {} distinct data at Company level'.format(data['CMP'].nunique()))
print('There are {} distinct data at Mother-Brand level'.format(data['MBRD'].nunique()))
print('There are {} distinct data at Brand level'.format(data['BRD'].nunique()))
```

```
There are 80 distinct data at Group level
There are 174 distinct data at sub-Group level
There are 232 distinct data at sub-sub-Group level
There are 354 distinct data at Company level
There are 643 distinct data at Mother-Brand level
There are 1315 distinct data at Brand level
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
  and should_run_async(code)
```

```
# Initial Data Exploration:

print(data.head())

# Get descriptive statistics
print(data.describe())

# Check for missing values
print(data.isnull().sum())
```

```
  MONTH STORECODE  DAY BILL_ID  BILL_AMT  QTY  VALUE  PRICE  \
0    M1        N1    4    T375     225.0  1.0  225.0  225.0
1    M1        N1    4    T379      95.0  1.0   95.0   95.0
2    M1        N1    4    T381      10.0  1.0   10.0   10.0
3    M1        N1    4    T382     108.0  1.0  108.0  108.0
4    M1        N1    4    T384      19.0  1.0   19.0   19.0

                      GRP                    SGRP                   SSGRP  \
0     BUTTER MARGR  (4/94)                  BUTTER                  SALTED
1  CONFECTIONERY - ECLAIRS  CONFECTIONERY - ECLAIRS  CONFECTIONERY - ECLAIRS
2                CHOCOLATE         CHOCOLATE PANNED         CHOCOLATE PANNED
```

```
3             PACKAGED TEA              MAIN PACKS              MAIN PACKS
4         ALL IODISED SALT            POWDERED SALT            POWDERED SALT

                      CMP           MBRD                BRD
0               G C M M F           AMUL                AMUL
1             PARLE PRODS         MELODY    MELODY CHOCOLATY
2  MONDELEZ INTERNATIONAL  CADBURY SHOTS       CADBURY SHOTS
3      GUJ TEA PROCESSORS     WAGH BAKRI  WAGH BAKRI INSTANT
4               TATA CHEM           TATA           TATA SALT
                    DAY      BILL_AMT           QTY         VALUE         PRICE
count  26985.000000  26985.000000  26985.000000  26985.000000  26985.000000
mean      15.167019    278.754206      4.105021     67.808551     52.812982
std        8.956057    541.398504     95.666947    118.005978     84.987730
min        1.000000      0.000000      0.500000      0.000000      0.000000
25%        7.000000     40.000000      1.000000     10.000000     10.000000
50%       14.000000    111.000000      1.000000     30.000000     22.000000
75%       23.000000    280.000000      2.000000     80.000000     64.000000
max       31.000000   7292.000000  12000.000000   3150.000000   3150.000000
MONTH        0
STORECODE    0
DAY          0
BILL_ID      0
BILL_AMT     0
QTY          0
VALUE        0
PRICE        0
GRP          0
SGRP         0
SSGRP        0
CMP          0
MBRD         0
BRD          0
dtype: int64
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
  and should_run_async(code)
```

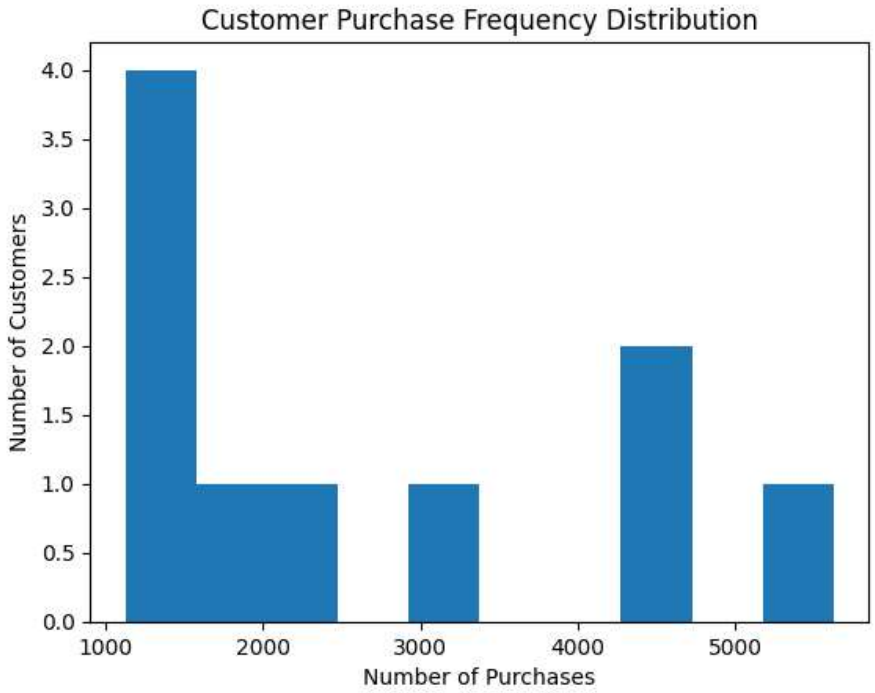
```
# Data Cleaning and Preprocessing:
print(data.dropna);
```

```
<bound method DataFrame.dropna of       MONTH STORECODE  DAY BILL_ID  BILL_AMT  QTY  VALUE  PRICE  \
0        M1        N1    4    T375     225.0  1.0  225.0  225.0
1        M1        N1    4    T379      95.0  1.0   95.0   95.0
2        M1        N1    4    T381      10.0  1.0   10.0   10.0
3        M1        N1    4    T382     108.0  1.0  108.0  108.0
4        M1        N1    4    T384      19.0  1.0   19.0   19.0
...     ...       ...  ...     ...       ...  ...    ...    ...
26980    M3       N10   31   T5999      10.0  1.0   10.0   10.0
26981    M3       N10   31   T6007      88.0  1.0   10.0   10.0
26982    M3       N10   31   T6007      88.0  2.0   20.0   10.0
26983    M3       N10   31   T6007      88.0  1.0   58.0   58.0
26984    M3       N10   31   T6013      15.0  1.0   15.0   15.0

                            GRP                    SGRP  \
0            BUTTER MARGR  (4/94)                  BUTTER
1         CONFECTIONERY - ECLAIRS  CONFECTIONERY - ECLAIRS
2                       CHOCOLATE         CHOCOLATE PANNED
3                   PACKAGED TEA               MAIN PACKS
4                ALL IODISED SALT            POWDERED SALT
...                           ...                     ...
26980  BISCUITS - CORE & NON CORE         NON-SALT CRACKER
26981  BISCUITS - CORE & NON CORE                   MARIE
26982  BISCUITS - CORE & NON CORE                 GLUCOSE
26983           CLEANERS - GLASS                  LIQUIDS
26984  BISCUITS - CORE & NON CORE            SWEET/COOKIES

                        SSGRP                    CMP             MBRD  \
0                      SALTED              G C M M F             AMUL
1      CONFECTIONERY - ECLAIRS             PARLE PRODS           MELODY
2            CHOCOLATE PANNED  MONDELEZ INTERNATIONAL     CADBURY SHOTS
3                  MAIN PACKS      GUJ TEA PROCESSORS        WAGH BAKRI
```

```
4                POWDERED SALT              TATA CHEM              TATA
...                        ...                    ...                ...
26980         NON-SALT CRACKER             PARLE PRODS  PARLE KRACK JACK
26981                   MARIE             PARLE PRODS       PARLE MARIE
26982                 GLUCOSE             PARLE PRODS           PARLE-G
26983                 LIQUIDS       RECKITT BENCKISER             COLIN
26984           SWEET/COOKIES                SAJ INDS         BISK FARM

                         BRD
0                      AMUL
1           MELODY CHOCOLATY
2              CADBURY SHOTS
3         WAGH BAKRI INSTANT
4                  TATA SALT
...                      ...
26980       PARLE KRACK JACK
26981            PARLE MARIE
26982                PARLE-G
26983                  COLIN
26984  BISK FARM JUST GINGER

[26985 rows x 14 columns]>
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
  and should_run_async(code)
```

```
# Exploratory Data Analysis (EDA):

# Customer Purchase Patterns:
plt.hist(data["STORECODE"].value_counts())
plt.xlabel("Number of Purchases")
plt.ylabel("Number of Customers")
plt.title("Customer Purchase Frequency Distribution")
```

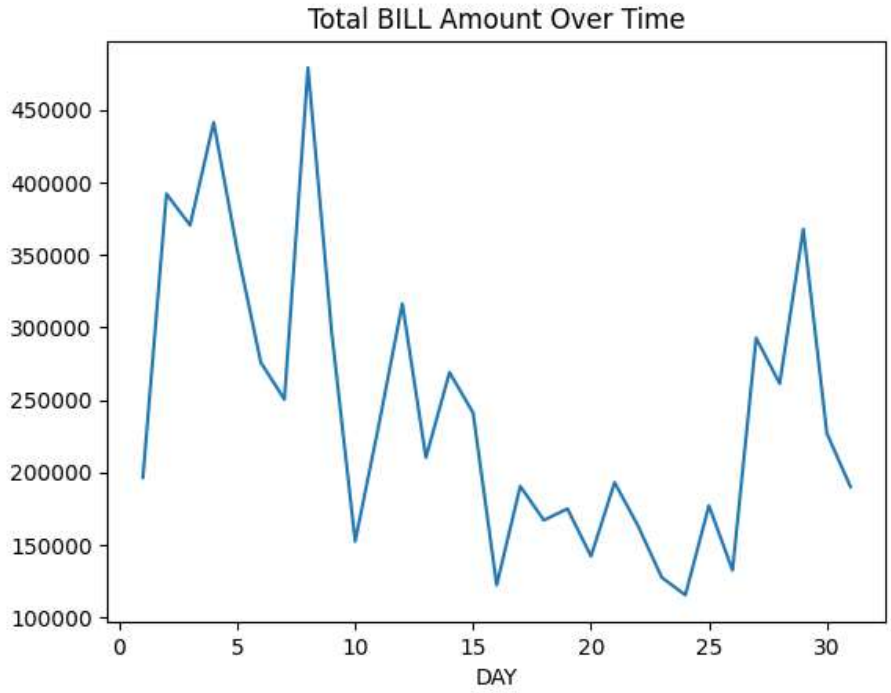
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarnir
  and should_run_async(code)
Text(0.5, 1.0, 'Customer Purchase Frequency Distribution')
```




```
#Average BILL amount
print("BILL_AMT:", data["BILL_AMT"].mean())
```

```
#Purchase patterns over time (visualize trends, seasonality)

data.groupby("DAY")["BILL_AMT"].sum().plot()
plt.title("Total BILL Amount Over Time")
```

```
BILL_AMT: 278.75420604039283
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
  and should_run_async(code)
Text(0.5, 1.0, 'Total BILL Amount Over Time')
```



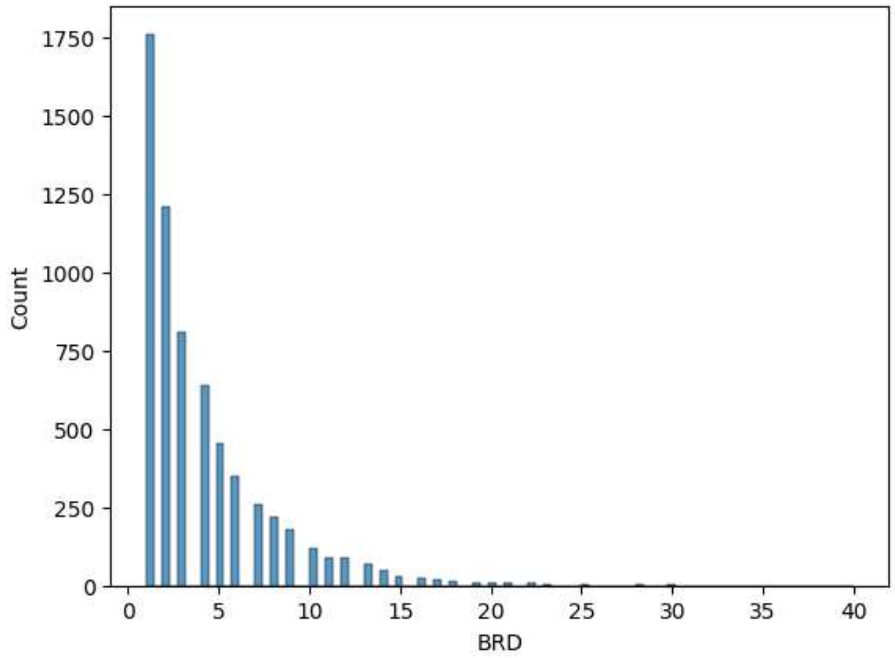

```
#Customer Preferences:

#Most popular products
print(data["GRP"].value_counts().head(10))
```

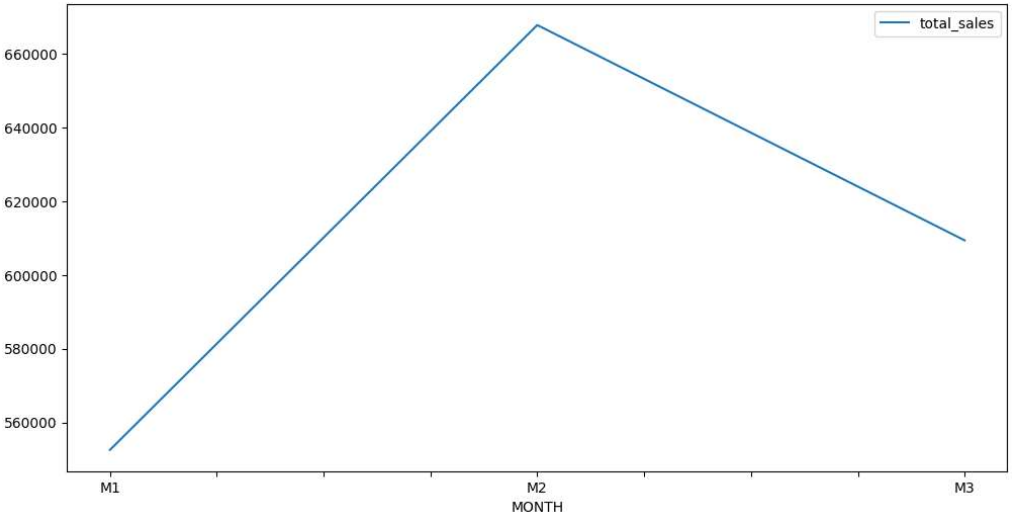
```
BISCUITS - CORE & NON CORE       6677
SALTY SNACKS (2/97)              1721
CHOCOLATE                        1408
SPICES (03/04)                   1347
VERMICELLI & NOODLE              1330
WASHING POWDERS/LIQUIDS          1208
PACKAGED TEA                      983
TOILET SOAPS (04/00)              933
REFINED EDIBLE OILS-ALL PACKS     851
CLEANERS - UTENSIL                666
Name: GRP, dtype: int64
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
  and should_run_async(code)
```

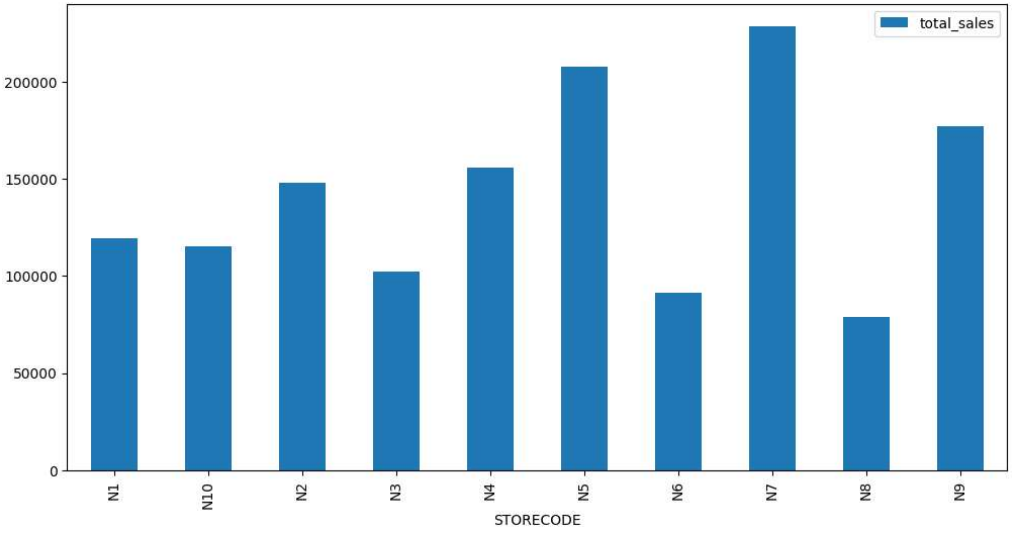
```
# Analyze purchase frequency
purchase_frequency = data.groupby("BILL_ID")["BRD"].nunique()
sns.histplot(purchase_frequency)
plt.show()

# Analyze purchase trends over time
data.groupby("MONTH").agg(total_sales=("VALUE", "sum")).plot(figsize=(12, 6))
```

```
data.groupby('MONTH').agg(total_sales=('VALUE', 'sum')).plot(figsize=(12, 6))
plt.show()

# Analyze purchase patterns by product category
data.groupby("STORECODE").agg(total_sales=("PRICE", "sum")).plot.bar(figsize=(12, 6))
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarr
  and should_run_async(code)
```







```
data.info()
```

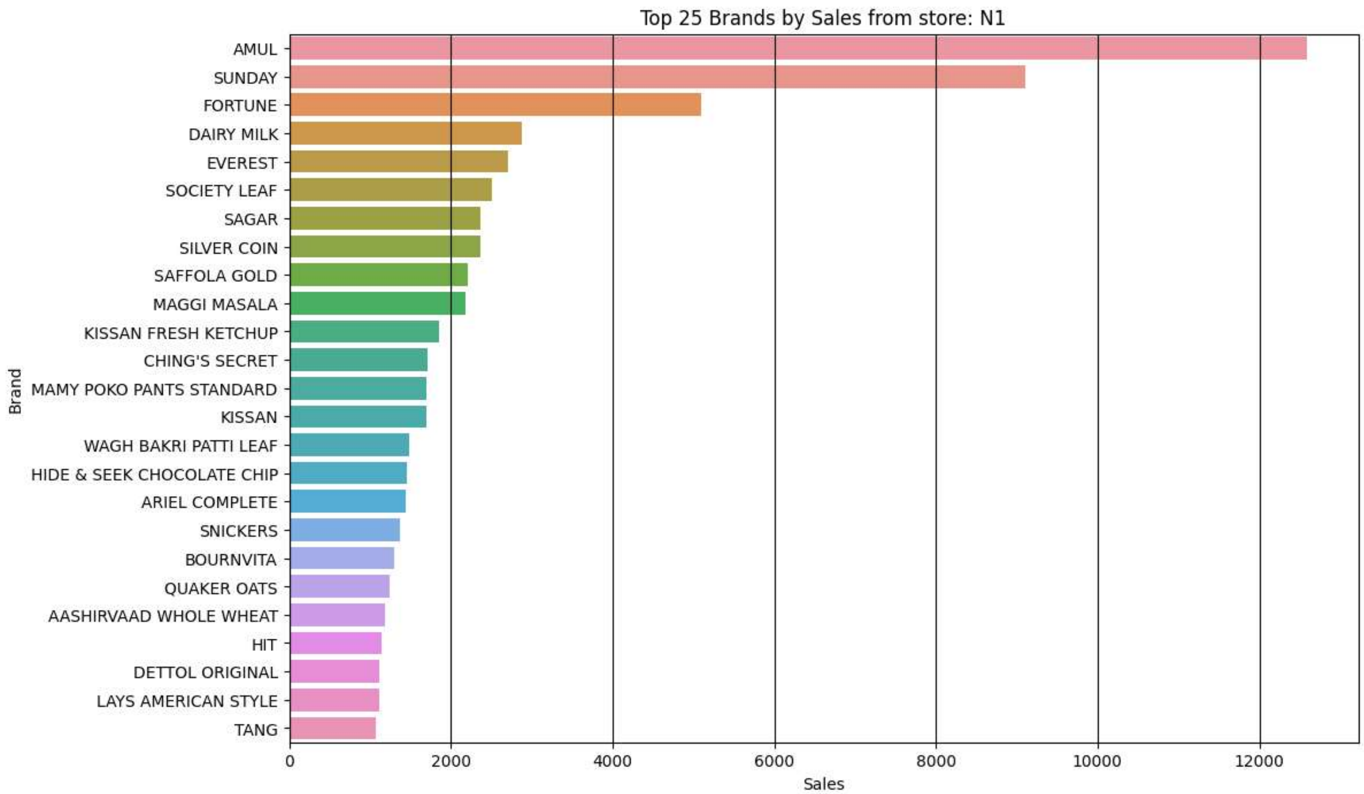
```
data['UNIQUE_ID'] = data['STORECODE'].str.cat(data['BILL_ID'],sep="_")
```

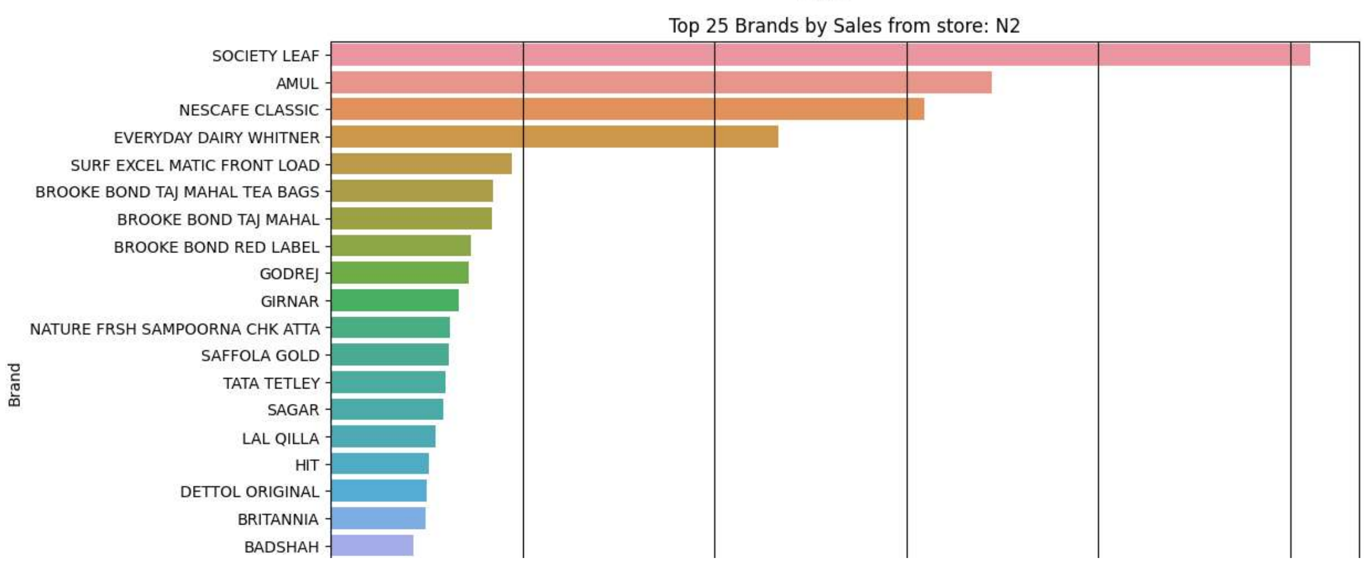
```
by_store = data.groupby("UNIQUE_ID").mean()[["DAY","BILL_AMT","QTY"]]
```

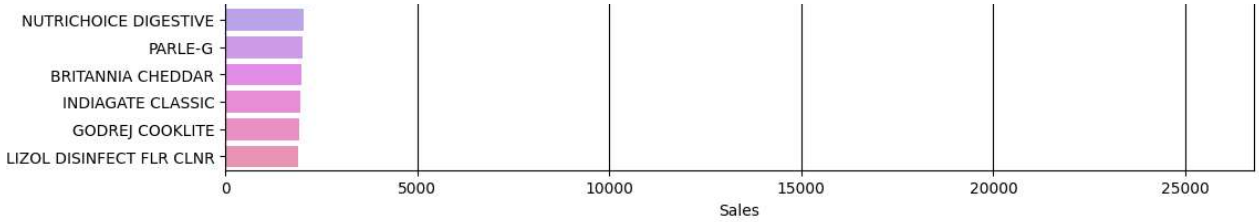
```
merged_data = pd.merge(data,by_store,on="UNIQUE_ID")
```

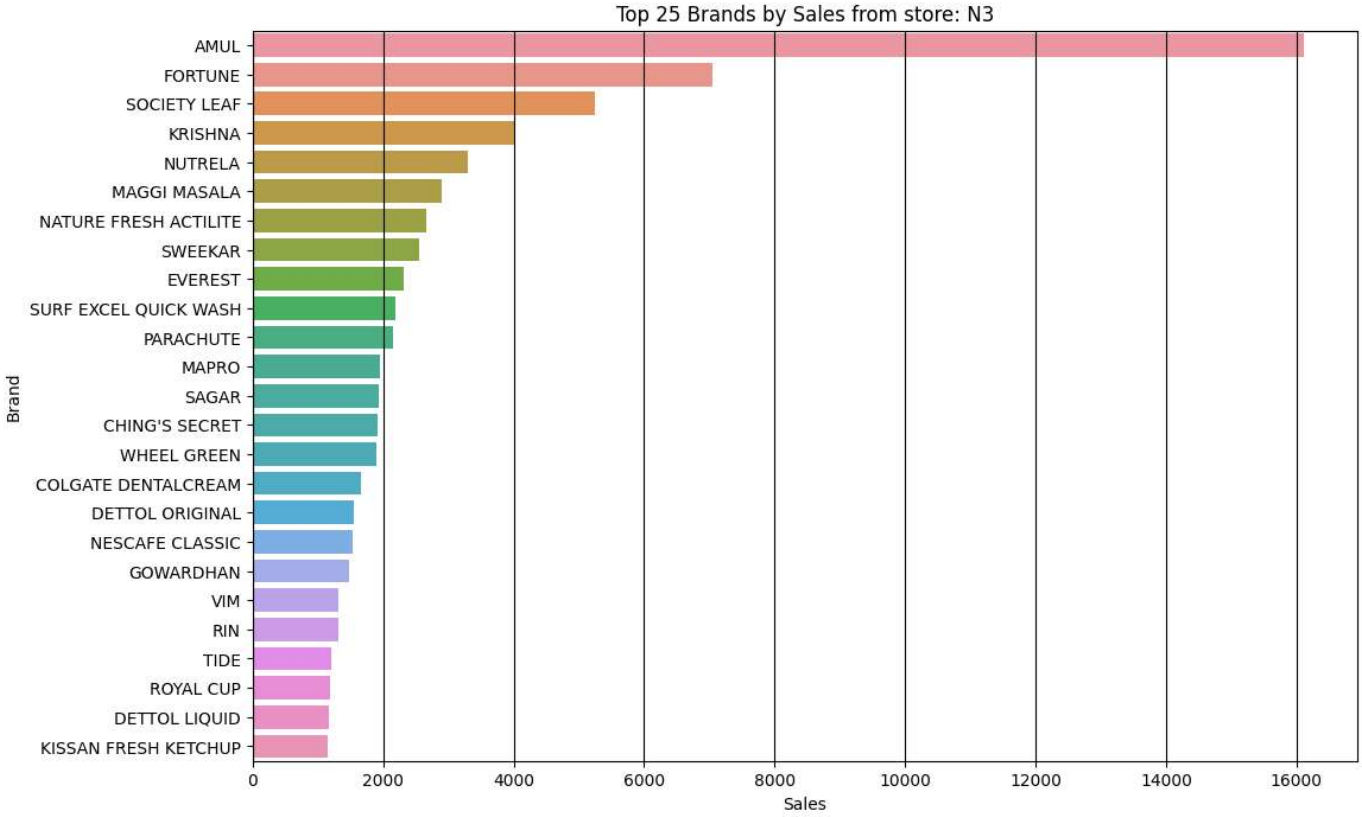
```
for i in merged_data.STORECODE.unique():
    brd_st = merged_data.loc[merged_data.STORECODE == i]
    brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
    plt.figure(figsize=(12,8))
    sns.barplot(x='VALUE',y=brd.index,data=brd)
    plt.title("Top 25 Brands by Sales from store: " + i)
    plt.xlabel("Sales")
    plt.ylabel("Brand")
    plt.grid(axis='x',color='black')
```

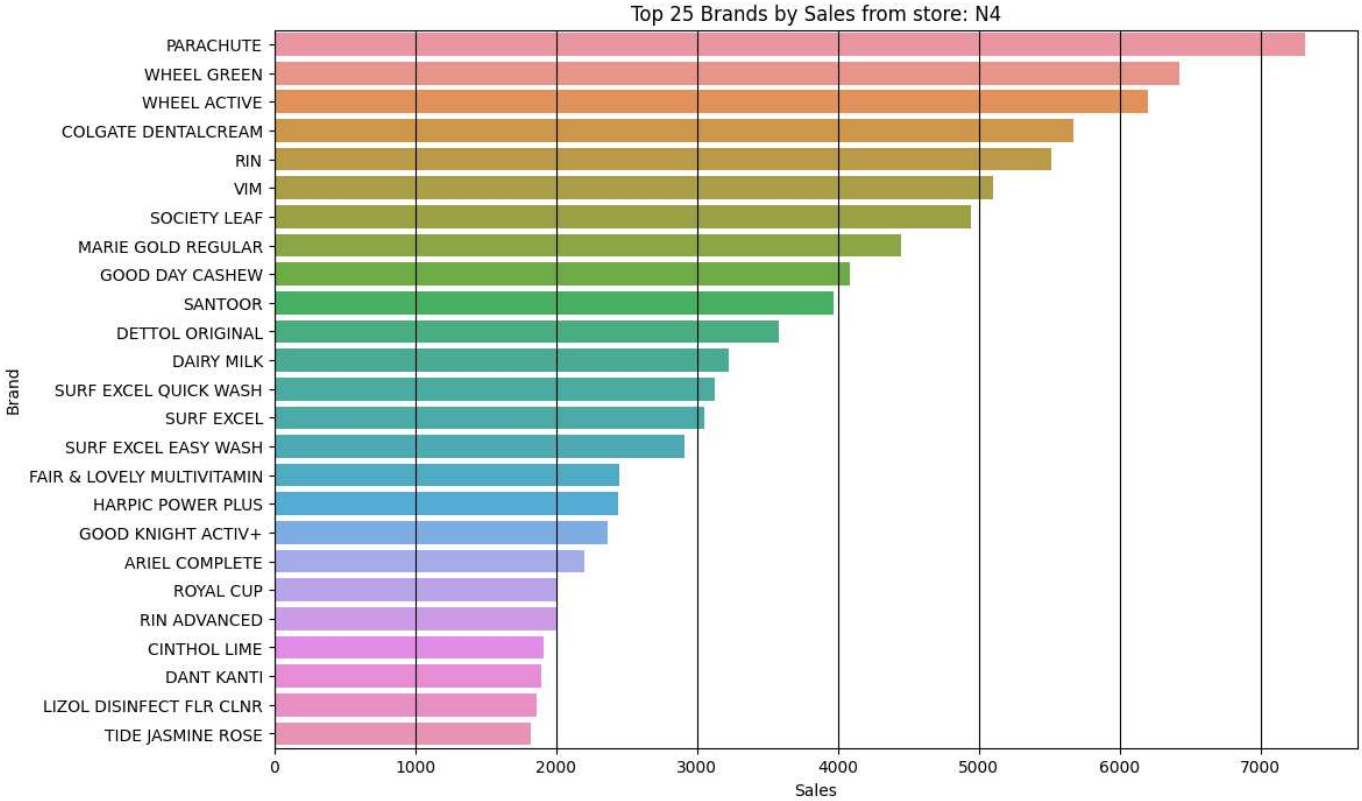
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will
  and should_run_async(code)
<ipython-input-19-53c40ccd4bf6>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
  brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
<ipython-input-19-53c40ccd4bf6>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
  brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
<ipython-input-19-53c40ccd4bf6>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
  brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
<ipython-input-19-53c40ccd4bf6>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
  brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
<ipython-input-19-53c40ccd4bf6>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
  brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
<ipython-input-19-53c40ccd4bf6>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
  brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
<ipython-input-19-53c40ccd4bf6>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
  brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
<ipython-input-19-53c40ccd4bf6>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
  brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
<ipython-input-19-53c40ccd4bf6>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
  brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
<ipython-input-19-53c40ccd4bf6>:3: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
  brd = brd_st.groupby('BRD').sum().sort_values('VALUE',ascending=False)[['VALUE','QTY_x']][0:25]
```

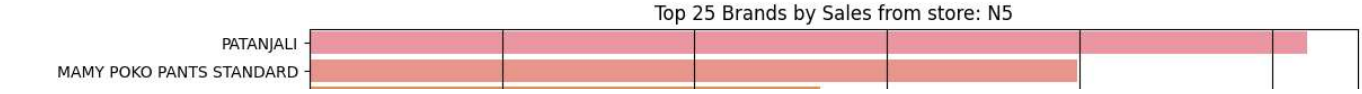


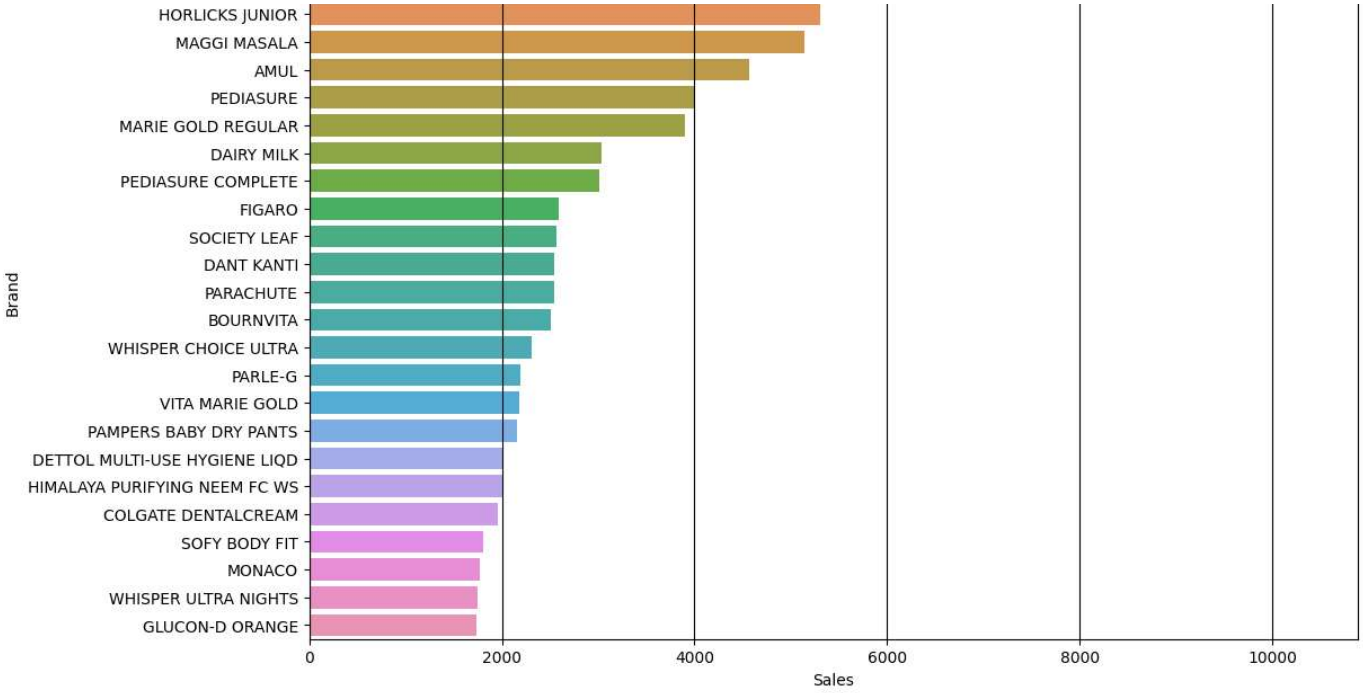


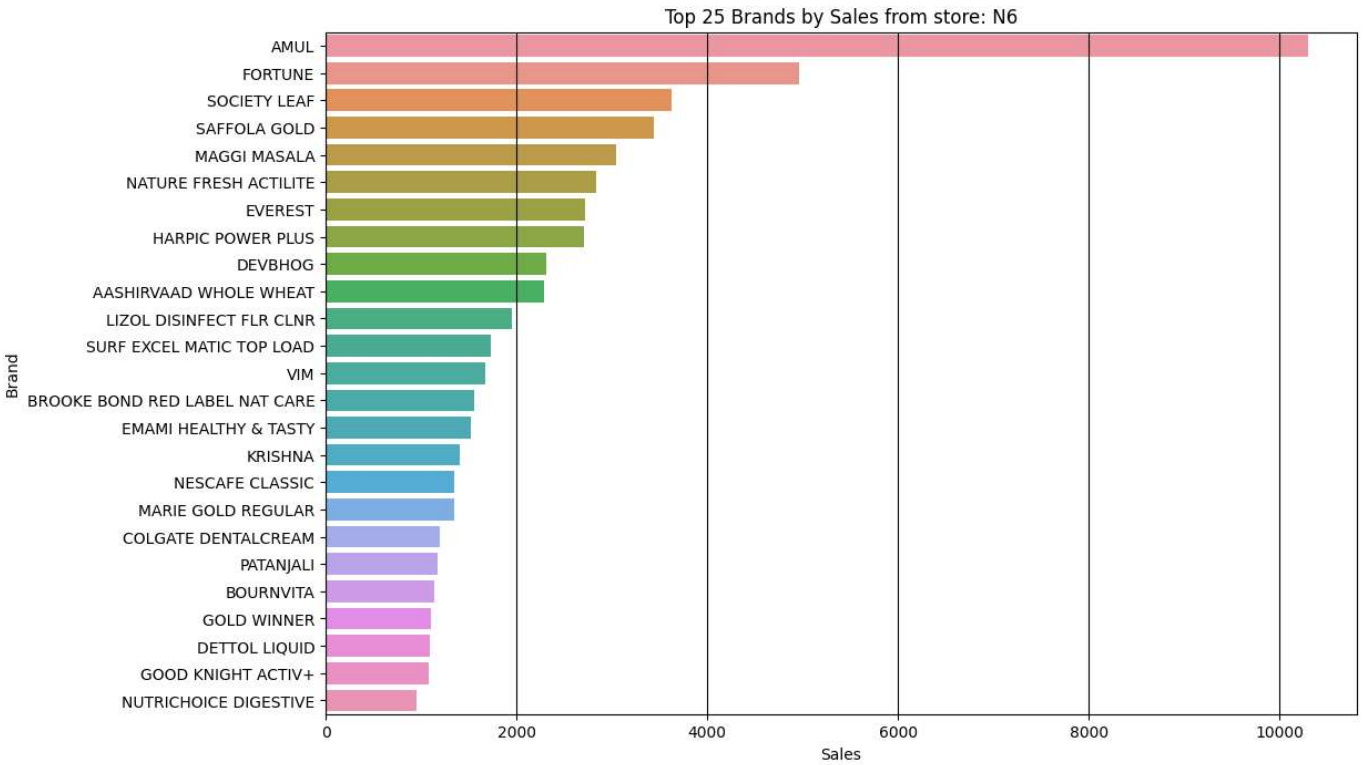

NUTRICHOICE DIGESTIVE
PARLE-G
BRITANNIA CHEDDAR
INDIAGATE CLASSIC
GODREJ COOKLITE
LIZOL DISINFECT FLR CLNR
0
5000
10000
15000
20000
25000
Sales

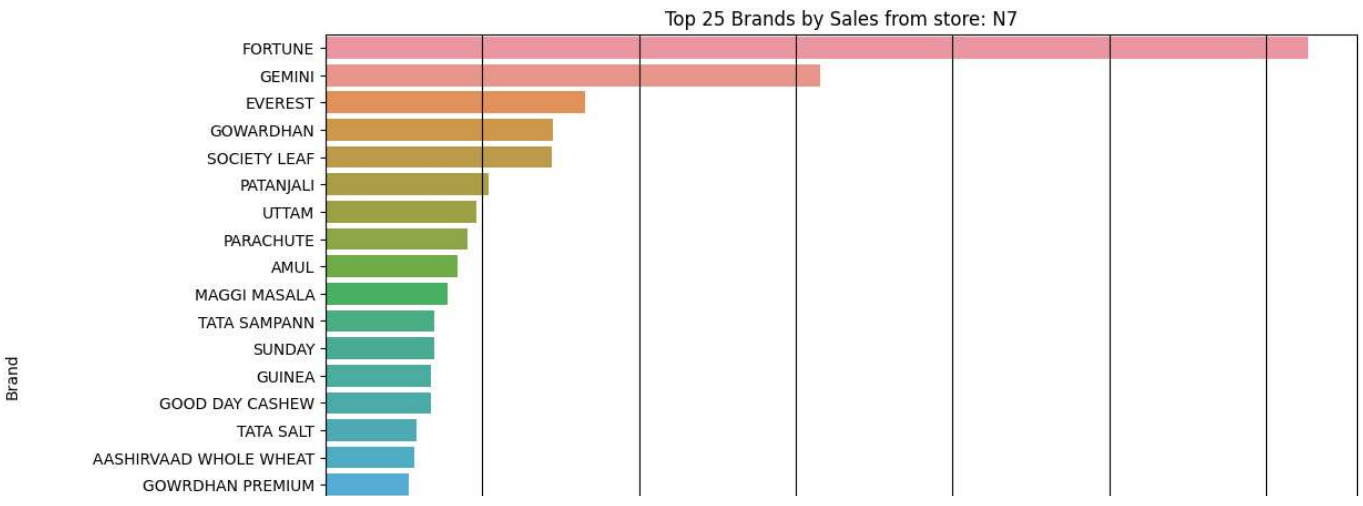

Top 25 Brands by Sales from store: N3
AMUL
FORTUNE
SOCIETY LEAF
KRISHNA
NUTRELA
MAGGI MASALA
NATURE FRESH ACTILITE
SWEEKAR
EVEREST
SURF EXCEL QUICK WASH
PARACHUTE
MAPRO
SAGAR
CHING'S SECRET
WHEEL GREEN
COLGATE DENTALCREAM
DETTOL ORIGINAL
NESCAFE CLASSIC
GOWARDHAN
VIM
RIN
TIDE
ROYAL CUP
DETTOL LIQUID
KISSAN FRESH KETCHUP
Brand
0
2000
4000
6000
8000
10000
12000
14000
16000
Sales


Top 25 Brands by Sales from store: N4
PARACHUTE
WHEEL GREEN
WHEEL ACTIVE
COLGATE DENTALCREAM
RIN
VIM
SOCIETY LEAF
MARIE GOLD REGULAR
GOOD DAY CASHEW
SANTOOR
DETTOL ORIGINAL
DAIRY MILK
SURF EXCEL QUICK WASH
SURF EXCEL
SURF EXCEL EASY WASH
FAIR & LOVELY MULTIVITAMIN
HARPIC POWER PLUS
GOOD KNIGHT ACTIV+
ARIEL COMPLETE
ROYAL CUP
RIN ADVANCED
CINTHOL LIME
DANT KANTI
LIZOL DISINFECT FLR CLNR
TIDE JASMINE ROSE
Brand
0
1000
2000
3000
4000
5000
6000
7000
Sales


Top 25 Brands by Sales from store: N5
PATANJALI
MAMY POKO PANTS STANDARD


HORLICKS JUNIOR
MAGGI MASALA
AMUL
PEDIASURE
MARIE GOLD REGULAR
DAIRY MILK
PEDIASURE COMPLETE
FIGARO
SOCIETY LEAF
DANT KANTI
PARACHUTE
BOURNVITA
WHISPER CHOICE ULTRA
PARLE-G
VITA MARIE GOLD
PAMPERS BABY DRY PANTS
DETTOL MULTI-USE HYGIENE LIQD
HIMALAYA PURIFYING NEEM FC WS
COLGATE DENTALCREAM
SOFY BODY FIT
MONACO
WHISPER ULTRA NIGHTS
GLUCON-D ORANGE
Brand
0
2000
4000
6000
8000
10000
Sales


Top 25 Brands by Sales from store: N6
AMUL
FORTUNE
SOCIETY LEAF
SAFFOLA GOLD
MAGGI MASALA
NATURE FRESH ACTILITE
EVEREST
HARPIC POWER PLUS
DEVBHOG
AASHIRVAAD WHOLE WHEAT
LIZOL DISINFECT FLR CLNR
SURF EXCEL MATIC TOP LOAD
VIM
BROOKE BOND RED LABEL NAT CARE
EMAMI HEALTHY & TASTY
KRISHNA
NESCAFE CLASSIC
MARIE GOLD REGULAR
COLGATE DENTALCREAM
PATANJALI
BOURNVITA
GOLD WINNER
DETTOL LIQUID
GOOD KNIGHT ACTIV+
NUTRICHOICE DIGESTIVE
Brand
0
2000
4000
6000
8000
10000
Sales


Top 25 Brands by Sales from store: N7
FORTUNE
GEMINI
EVEREST
GOWARDHAN
SOCIETY LEAF
PATANJALI
UTTAM
PARACHUTE
AMUL
MAGGI MASALA
TATA SAMPANN
SUNDAY
GUINEA
GOOD DAY CASHEW
TATA SALT
AASHIRVAAD WHOLE WHEAT
GOWRDHAN PREMIUM
Brand