rchase-behavior-analysis-eda-final

April 5, 2025

1 Customer Purchase Behavior Analysis using Python libraries

[85]:
```
pip install -U klib
```

```
Defaulting to user installation because normal site-packages is not
writeableNote: you may need to restart the kernel to use updated packages.

Requirement already satisfied: klib in c:\users\lakshmi priya
s\appdata\roaming\python\python311\site-packages (1.3.2)
Requirement already satisfied: jinja2>=3.1.0 in
c:\programdata\anaconda3\lib\site-packages (from klib) (3.1.3)
Requirement already satisfied: matplotlib>=3.6.0 in
c:\programdata\anaconda3\lib\site-packages (from klib) (3.8.0)
Requirement already satisfied: numpy>=1.26.0 in
c:\programdata\anaconda3\lib\site-packages (from klib) (1.26.4)
Requirement already satisfied: pandas<3.0,>=1.4 in
c:\programdata\anaconda3\lib\site-packages (from klib) (2.1.4)
Requirement already satisfied: plotly>=5.11.0 in c:\users\lakshmi priya
s\appdata\roaming\python\python311\site-packages (from klib) (5.24.1)
Requirement already satisfied: scipy>=1.10.0 in
c:\programdata\anaconda3\lib\site-packages (from klib) (1.11.4)
Requirement already satisfied: screeninfo>=0.8.1 in c:\users\lakshmi priya
s\appdata\roaming\python\python311\site-packages (from klib) (0.8.1)
Requirement already satisfied: seaborn>=0.12.0 in
c:\programdata\anaconda3\lib\site-packages (from klib) (0.12.2)
Requirement already satisfied: MarkupSafe>=2.0 in
c:\programdata\anaconda3\lib\site-packages (from jinja2>=3.1.0->klib) (2.1.3)
Requirement already satisfied: contourpy>=1.0.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.6.0->klib)
(1.2.0)
Requirement already satisfied: cycler>=0.10 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.6.0->klib)
(0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.6.0->klib)
(4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.6.0->klib)
```

```
(1.4.4)
Requirement already satisfied: packaging>=20.0 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.6.0->klib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.6.0->klib)
(10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.6.0->klib)
(3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=3.6.0->klib)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in
c:\programdata\anaconda3\lib\site-packages (from pandas<3.0,>=1.4->klib)
(2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
c:\programdata\anaconda3\lib\site-packages (from pandas<3.0,>=1.4->klib)
(2023.3)
Requirement already satisfied: tenacity>=6.2.0 in
c:\programdata\anaconda3\lib\site-packages (from plotly>=5.11.0->klib) (8.2.2)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib>=3.6.0->klib) (1.16.0)
```

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import klib
warnings.filterwarnings('ignore')
```

[2]:
```
data = pd.read_csv(r"C:\Users\Lakshmi Priya␣
↪S\OneDrive\Desktop\Datasets\Ecommerce_dataset.csv")
data.head()
```

[2]:

	CID	TID	Gender	Age Group	Purchase Date	\
0	943146	5876328741	Female	25-45	30/08/2023 20:27:08	
1	180079	1018503182	Male	25-45	23/02/2024 09:33:46	
2	337580	3814082218	Other	60 and above	06/03/2022 09:09:50	
3	180333	1395204173	Other	60 and above	04/11/2020 04:41:57	
4	447553	8009390577	Male	18-25	31/05/2022 17:00:32	

	Product Category	Discount Availed	Discount Name	Discount Amount (INR)	\
0	Electronics	Yes	FESTIVE50	64.30	
1	Electronics	Yes	SEASONALOFFER21	175.19	
2	Clothing	Yes	SEASONALOFFER21	211.54	
3	Sports & Fitness	No	NaN	0.00	

```
4  Sports & Fitness              Yes         WELCOME5                  439.92

   Gross Amount   Net Amount Purchase Method   Location
0    725.304000   661.004000     Credit Card  Ahmedabad
1   4638.991875  4463.801875     Credit Card  Bangalore
2   1986.372575  1774.832575     Credit Card      Delhi
3   5695.612650  5695.612650      Debit Card      Delhi
4   2292.651500  1852.731500     Credit Card      Delhi
```

[3]:
```
data.shape
```

[3]:
```
(55000, 13)
```

[4]:
```
def column_summary(data):
    summary_data = []
    for col in data.columns:
        column_dtype = data[col].dtype
        num_of_nulls = data[col].isnull().sum()
        num_of_non_nulls = data[col].notnull().sum()
        num_of_distinct_values = data[col].nunique()
        summary_data.append({
            'col': col,
            'column_dtype': column_dtype,
            'num_of_nulls': num_of_nulls,
            'num_of_non_nulls': num_of_non_nulls,
            'num_of_distinct_values': num_of_distinct_values,
        })

    summary_df = pd.DataFrame(summary_data)
    return summary_df
```

[5]:
```
summary_df = column_summary(data)
display(summary_df)
```

```
                    col column_dtype  num_of_nulls  num_of_non_nulls  \
0                   CID        int64             0             55000
1                   TID        int64             0             55000
2                Gender       object             0             55000
3             Age Group       object             0             55000
4         Purchase Date       object             0             55000
5      Product Category       object             0             55000
6      Discount Availed       object             0             55000
7         Discount Name       object         27585             27415
8  Discount Amount (INR)      float64             0             55000
9          Gross Amount      float64             0             55000
10           Net Amount      float64             0             55000
11      Purchase Method       object             0             55000
```

```
12              Location       object             0             55000

    num_of_distinct_values
0                    29071
1                    55000
2                        3
3                        5
4                    54988
5                        9
6                        2
7                        5
8                    20578
9                    54807
10                   54936
11                       8
12                      14
```

1.1 Data Summary Description

- There are 13 features in the dataset.
- The features 'CID' and 'TID' are unique identifiers, the number of distinct values is not equal to the number of non-nulls in 'CID', which indicates same customer have repeatedly purchased manier times. But, in 'TID' the number of distinct values is equal to the number of non-nulls. So, 'TID' is the primary key.
- Both CID (Customer ID) and TID (Transaction ID) are unique identifiers. They do not carry any meaningful information that contributes to the patterns in customer behavior or transaction trends. So, lets consider droping them.
- The 'Discount name' has NaN values more that 50% of the number of observation and it has only 5 distinct values, droping this feature may not be impactful. The Discount Availed and Discount Amount (INR) columns provide enough information about discounts. Imputing that will lead to manipulation of data which is not advisilbe.
- But stil if the anlysis is around "During which discount offer the purchase was high?", 'Discount Name' must me handled.

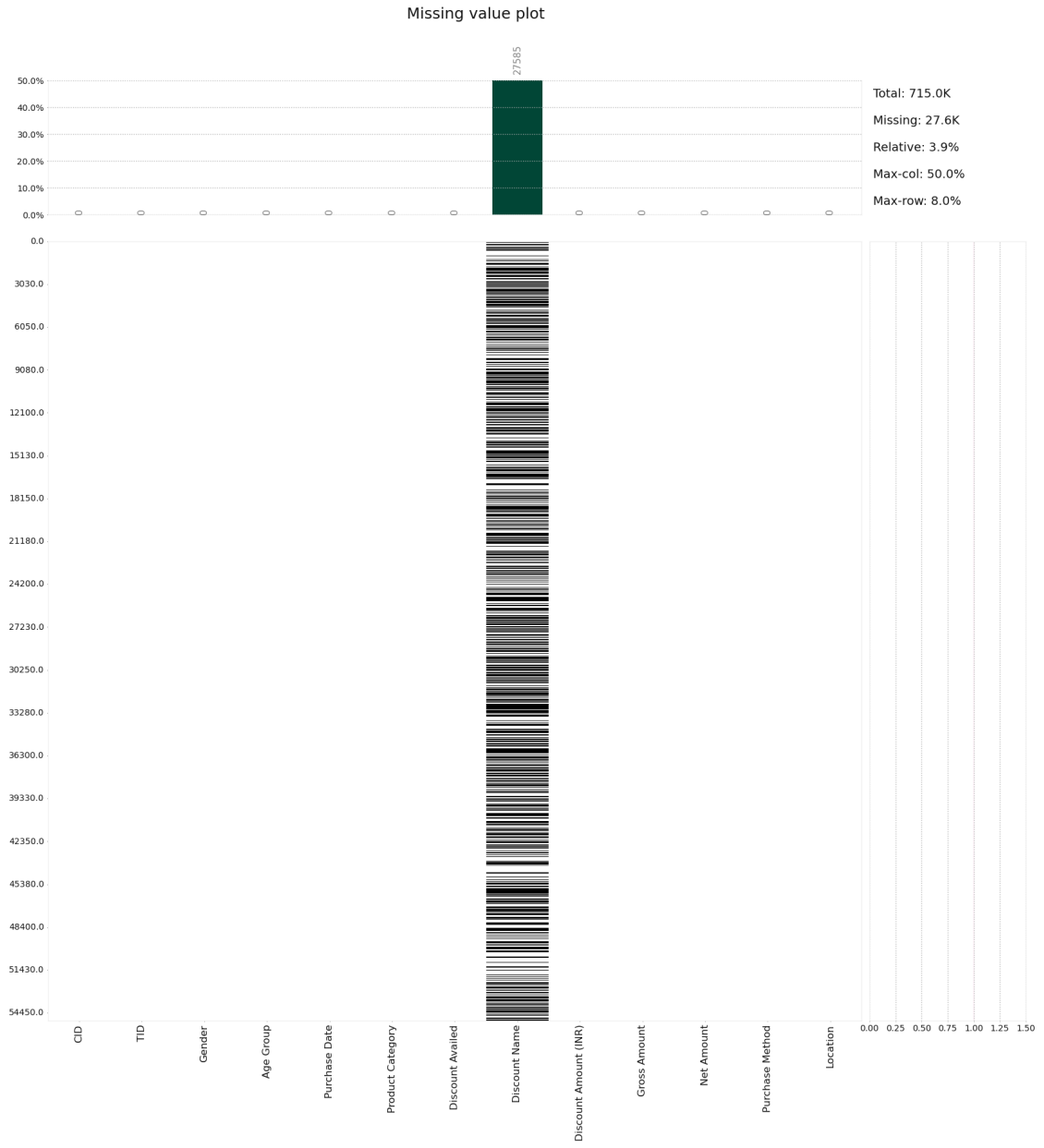
[6]:
```
data['Discount Name'].value_counts()
```

[6]:
```
Discount Name
NEWYEARS           8135
SEASONALOFFER21    6940
FESTIVE50          4115
SAVE10             4115
WELCOME5           4110
Name: count, dtype: int64
```

1.2 Missing Value Analysis

[7]:
```
klib.missingval_plot(data)
```

[7]:
```
GridSpec(6, 6)
```



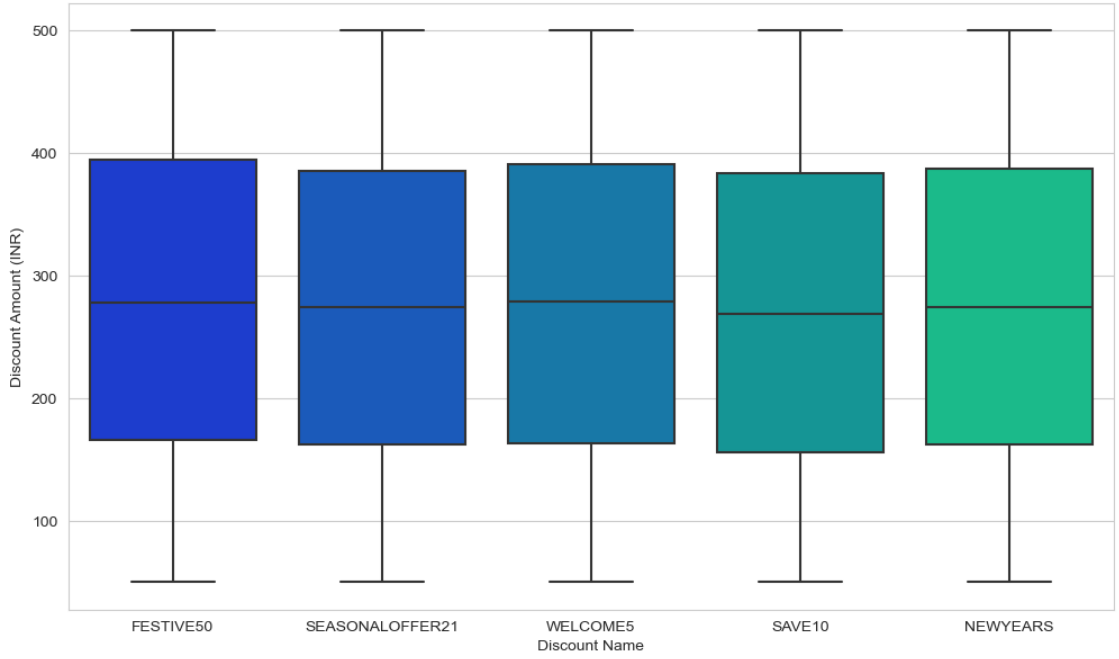
1. Total Data Points : The dataset contains 715,000 total data points.
2. Missing Data : 27.6K data points are missing, which represents 3.9% of the whole data.
3. Relative Missing : Only a small portion (3.9%) of the dataset contains missing values.
4. Missing value by Column : The missing data is concentrated in the “Discount Name” column,

where more than 50% of the values are missing.

5. Max Missing by Row: The maximum amount of missing data in any row is 8%, meaning no row has excessive missing values. This visualization suggests that most of the missing values are concentrated/accumulated in the "Discount Name" column, with the other columns having no missing values.

[8]:
```
#Analysis the distribution of 'Discount Name'
plt.figure(figsize=(12,7))
sns.set_style('whitegrid')
sns.boxplot(x='Discount Name',y='Discount Amount␣
 ↪(INR)',data=data,palette='winter')
```

[8]:
```
<Axes: xlabel='Discount Name', ylabel='Discount Amount (INR)'>
```



- The median discount amount across all discount codes(discount names) is roughly similar, which are falling between 250 to 300 INR.
- The FESTIVE50, SEASONALOFFER21, and WELCOME5 discount codes have a slightly higher median than SAVE10 and NEWYEARS.
- The range of discount amounts is similar across all the discount codes, with most values falling between 100 and 400 INR.

Summary: The discount amounts are distributed similarly across the different discount names, with no extreme variation. The medians and ranges are comparable across all codes and not much distinct.

For all the Discounts(Discount Names) provided similar amount of discount is only given. No discount offer provide much high or low discount. So, suggesting to provide different discount offers

sometimes less as 5% and somtimes more as 30% to increase curiocity in customers and engage them more in purchasing.

[9]:
```
data.duplicated().sum()
```

[9]:
```
0
```

There no duplicate values found in the dataset

1.3 Descriptive statistics for numerical features

[10]:
```
data.describe()
```

[10]:

	CID	TID	Discount Amount (INR)	Gross Amount	Net Amount
count	55000.000000	5.500000e+04	55000.000000	55000.000000	55000.000000
mean	551245.593891	5.504740e+09	136.986796	3012.936606	2875.949810
std	260603.330337	2.594534e+09	165.375502	1718.431066	1726.127778
min	100009.000000	1.000163e+09	0.000000	136.454325	-351.119775
25%	323717.000000	3.252604e+09	0.000000	1562.111325	1429.551863
50%	550088.500000	5.498383e+09	0.000000	2954.266150	2814.910875
75%	776955.750000	7.747933e+09	274.115000	4342.221675	4211.407838
max	999996.000000	9.999393e+09	500.000000	8394.825600	8394.825600

INSIGHTS: - Discount amount: 25% of transactions had no discount (25th percentile is 0), and 50% also had no discount. 75% of transactions had a discount of up to 274, which indicates a smaller number of transactions received higher discounts (top 25%). The standard deviation is relatively high 165, which represents the variability in the discount values. - Gross Amounts: The average transaction is approximately 3013, with a wide range from 136 to 8395. Half of the transactions were between 1562 and 4342, high sd also shows a wide variability in transaction amounts. - Net Amounts: After discounts, the average net amount is around 2876. Some transactions even resulted in negative net values (- 351), this may be due to refunds or over-discounting but over discounting is not possible practically. And refunds could not be more than Gross amount.

[11]:
```
# displaying the rows in the dataset that contains negative values
negative_values = data[data['Net Amount'] < 0]
print("Negative Values:\n", negative_values)
```

```
Negative Values:
```

```
           CID         TID  Gender     Age Group        Purchase Date  \
311     565676  1356242586    Male         45-60  03/05/2020 20:14:16
383     101206  4082960786  Female         18-25  05/05/2020 07:02:36
398     270557  9264884678    Male         45-60  06/03/2021 12:54:09
489     334639  1610899978    Male         18-25  18/09/2022 22:31:07
564     924584  8564435752   Other         25-45  13/02/2020 05:04:55
…        …         …     …           …                  …
54271   654558  6092629329    Male         25-45  19/02/2023 10:08:25
54280   253300  9018344109  Female  60 and above  07/03/2020 19:06:13
54378   789978  1230560432  Female         25-45  30/08/2020 05:05:20
54536   569063  8940735281    Male         45-60  20/08/2021 17:17:32
54903   940867  8469174728    Male         25-45  22/10/2019 01:21:46

        Product Category Discount Availed    Discount Name  \
311    Beauty and Health              Yes         NEWYEARS
383          Electronics              Yes         NEWYEARS
398       Home & Kitchen              Yes         WELCOME5
489       Home & Kitchen              Yes         NEWYEARS
564          Electronics              Yes           SAVE10
…                  …               …              …
54271        Electronics              Yes  SEASONALOFFER21
54280     Home & Kitchen              Yes  SEASONALOFFER21
54378              Books              Yes           SAVE10
54536        Electronics              Yes        FESTIVE50
54903           Clothing              Yes  SEASONALOFFER21

       Discount Amount (INR)  Gross Amount  Net Amount Purchase Method  \
311                   469.58    279.258000 -190.322000      Debit Card
383                   358.58    204.886500 -153.693500      Debit Card
398                   385.36    246.808100 -138.551900      Debit Card
489                   427.33    348.448275  -78.881725     Credit Card
564                   429.49    414.914325  -14.575675     Credit Card
…                      …           …         …              …
54271                 443.65    379.287000  -64.363000      Debit Card
54280                 305.25    178.402875 -126.847125       Paytm UPI
54378                 212.69    185.241000  -27.449000     PhonePe UPI
54536                 492.84    216.315000 -276.525000  Google Pay UPI
54903                 362.14    247.819000 -114.321000      Debit Card

        Location
311       Mumbai
383        Delhi
398    Bangalore
489        Delhi
564        Delhi
…          …
54271    Chennai
54280  Ahmedabad
```

```
54378     Mumbai
54536    Lucknow
54903    Kolkata

[613 rows x 13 columns]
```

[12]:
```
negative_values.shape
```

[12]:
```
(613, 13)
```

The 'Net Amount' contains negative values which is not a meaningful negative value in this senario. So, replaceing them with 0 is also not meaningful cause practically and logically no where 100% discount will be availed. So, Since the negative values in 'Net Amount' is less than 2% of the whole population these values are droped(filtered).

[13]:
```
data_f = data[data['Net Amount'] >= 0]
```

[14]:
```
# Categorical and Numerical Data
cat_cols= data_f.select_dtypes(include='object').columns
num_cols= data_f.select_dtypes(include= ['float64','int64']).columns
print(f'Categorial Columns:\n {cat_cols}')
print(f'Numerial Columns:\n {num_cols}')
```

```
Categorial Columns:
 Index(['Gender', 'Age Group', 'Purchase Date', 'Product Category',
       'Discount Availed', 'Discount Name', 'Purchase Method', 'Location'],
      dtype='object')
Numerial Columns:
 Index(['CID', 'TID', 'Discount Amount (INR)', 'Gross Amount', 'Net Amount'],
dtype='object')
```

[15]:
```
#Looking after Unique distinct values in each Categorical columns to undergo␣
 ↪efficient feature engineering..
for col in cat_cols:
    unique_cols = data_f[col].unique()
    print(f'{col}:\n {unique_cols}\n')
```

```
Gender:
 ['Female' 'Male' 'Other']

Age Group:
 ['25-45' '60 and above' '18-25' '45-60' 'under 18']

Purchase Date:
 ['30/08/2023 20:27:08' '23/02/2024 09:33:46' '06/03/2022 09:09:50' …
 '02/08/2024 09:30:44' '05/08/2020 23:57:56' '21/07/2022 09:05:18']

Product Category:
```

```
 ['Electronics' 'Clothing' 'Sports & Fitness' 'Pet Care' 'Home & Kitchen'
 'Books' 'Beauty and Health' 'Other' 'Toys & Games']

Discount Availed:
 ['Yes' 'No']

Discount Name:
 ['FESTIVE50' 'SEASONALOFFER21' nan 'WELCOME5' 'SAVE10' 'NEWYEARS']

Purchase Method:
 ['Credit Card' 'Debit Card' 'PhonePe UPI' 'Google Pay UPI' 'Net Banking'
 'Cash on Delivery' 'Paytm UPI' 'International Card']

Location:
 ['Ahmedabad' 'Bangalore' 'Delhi' 'Other' 'Chennai' 'Dehradun' 'Pune'
 'Hyderabad' 'Mumbai' 'Jaipur' 'Lucknow' 'Kolkata' 'Srinagar' 'Varanasi']
```

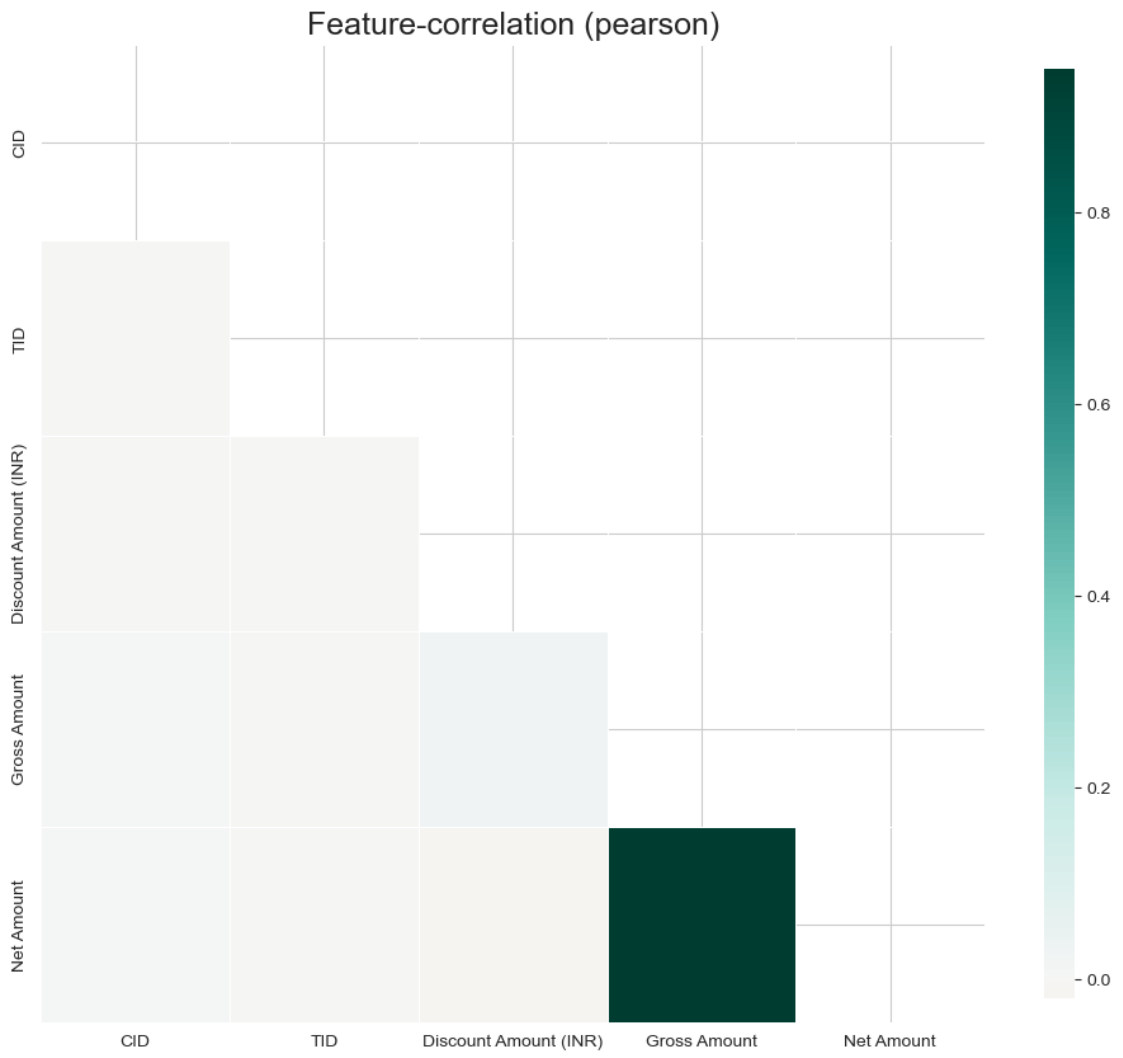
2 Visualization for Numerical Features

[16]:
```
klib.corr_mat(data_f)
```

[16]:
```
<pandas.io.formats.style.Styler at 0x1d5153fc150>
```

[17]:
```
klib.corr_plot(data_f)
```

[17]:
```
<Axes: title={'center': 'Feature-correlation (pearson)'}>
```



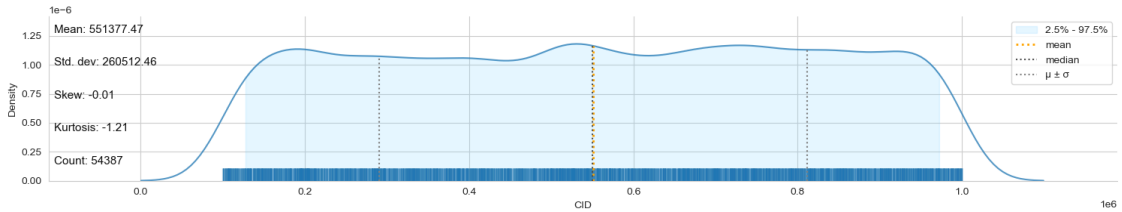
ANALYSIS: - Gross Amount and Net Amount have a strong positive correlation, meaning these two features move together closely (if one increases, the other also increases). - Discount Amount (INR) has a near-zero or very weak correlation with both Gross Amount and Net Amount, indicated by the light-coloured squares. This suggests that changes in the discount amount have very little or no direct impact on gross or net amounts.

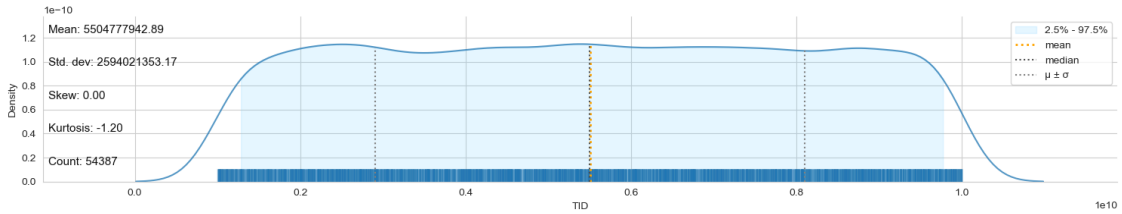
INSIGHTS: - The Gross Amount and Net Amount are highly correlated, while the Discount Amount (INR) has little correlation with the other two variables. This could mean that discounts don't significantly affect the gross or net totals in the dataset

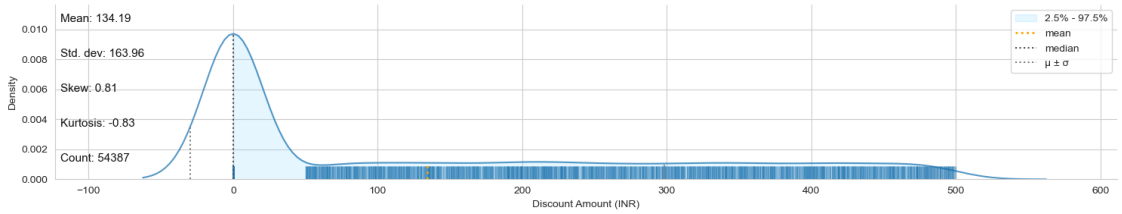
[18]:
```
klib.dist_plot(data_f)
```

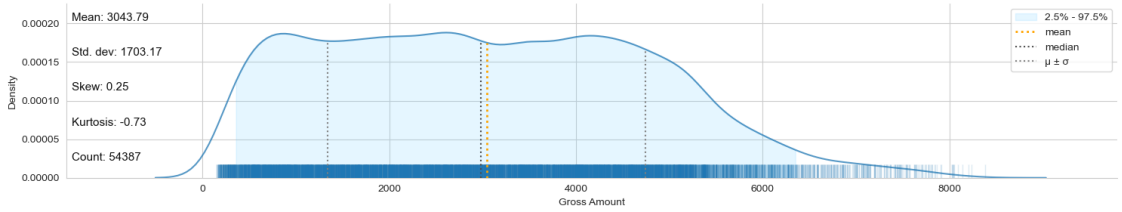
```
Large dataset detected, using 10000 random samples for the plots. Summary
statistics are still based on the entire dataset.
```

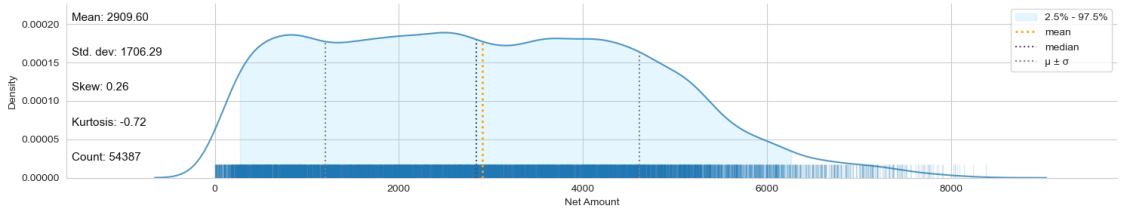
[18]: <Axes: xlabel='Net Amount', ylabel='Density'>











Most of the customers puchase productss on an average amount of 3000(mean). There is an average difference of amount 1706 for every purchase(sd). Very rare conditions customers purchase products

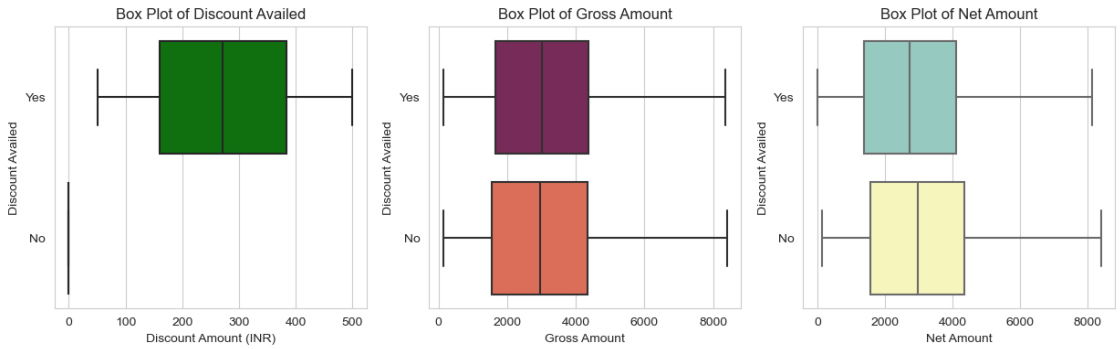
above 6000. Not heavily tailed indicating no outliers.

INFERENSE:

1. Discount Amount: Mean: 134.19 Std Dev: 163.96 Skewness: 0.81 (positively skewed, showing a longer right tail) Kurtosis: - 0.83 (light tailed compared to a normal distribution) Distribution: The plot shows positively skewed and platykurtic distributiom. Some data extend beyond 600, but those are outliers.
2. Gross Amount: Mean: 3043.79 Std Dev: 1703.17 Skewness: 0.25 (slightly positively skewed, slightly long right tailed) Kurtosis: -0.73 (light tailed) Distribution: The gross amount is more spread out, with values ranging from 0 to over 6000. It shows a moderate right tail, indicating some higher gross amounts, but most values lie between 2000 and 4000.
3. Net Amount: Mean: 2909.60 Std Dev: 1706.29 Skewness: 0.26 (slightly positively skewed) Kurtosis: -0.72 (light tailed) Distribution: Similar to the Gross Amount, this plot shows a slight positive skewed, with values concentrated between 2000 and 4000, though the tail extends up to around 6000.

INSIGHTS: - Discount Amount is negatively skewed with the majority of the data lying between 0-200. - Gross and Net Amounts are both slightly positively skewed, with similar spread and concentration of values between 2000-4000. - The distributions are fairly consistent, with no extreme skewness or heavy-tailed distributions. However, the tails indicate some outliers.

[19]:
```
plt.figure(figsize=(15, 4))
plt.subplot(1, 3, 1)
sns.boxplot(x='Discount Amount (INR)', y='Discount Availed', data=data_f,␣
 ↪color='g')
plt.title('Box Plot of Discount Availed')
plt.subplot(1, 3, 2)
sns.boxplot(x='Gross Amount', y='Discount Availed', data=data_f,␣
 ↪palette='rocket')
plt.title('Box Plot of Gross Amount')
plt.subplot(1, 3, 3)
sns.boxplot(x='Net Amount', y='Discount Availed', data=data_f,  palette='Set3')
plt.title('Box Plot of Net Amount')
plt.show()
```

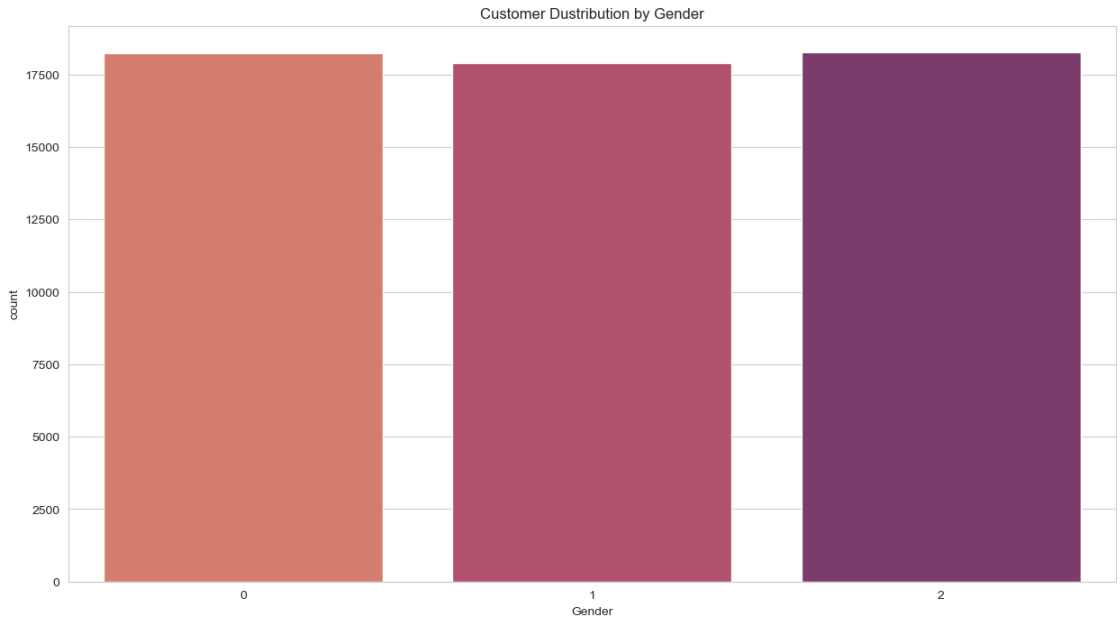


No outliers present in any of the three features. The Gross Amount and the Net Amount shows almost similar statistics. The overall discount anount ranges between 150 to 390 INR.

2.1 Visualization for CATEGORIAL COLUMNS

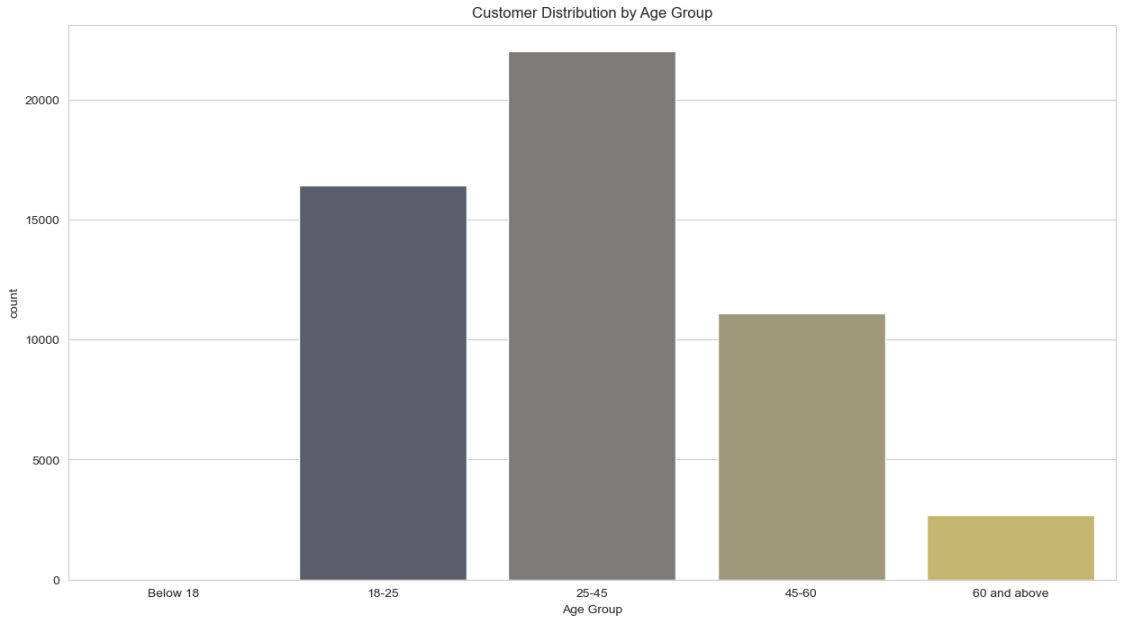
[28]:
```
# Gender distribution
plt.figure(figsize = (15,8))
sns.countplot(data=data_f, x='Gender', palette ="flare")
plt.title("Customer Dustribution by Gender")
```

[28]:
```
Text(0.5, 1.0, 'Customer Dustribution by Gender')
```



The purchasing activity across all three gender groups — Male, Female, and Other — is relatively balanced. However, female customers show a slightly higher share of purchases, indicating a marginally stronger engagement in buying behavior.

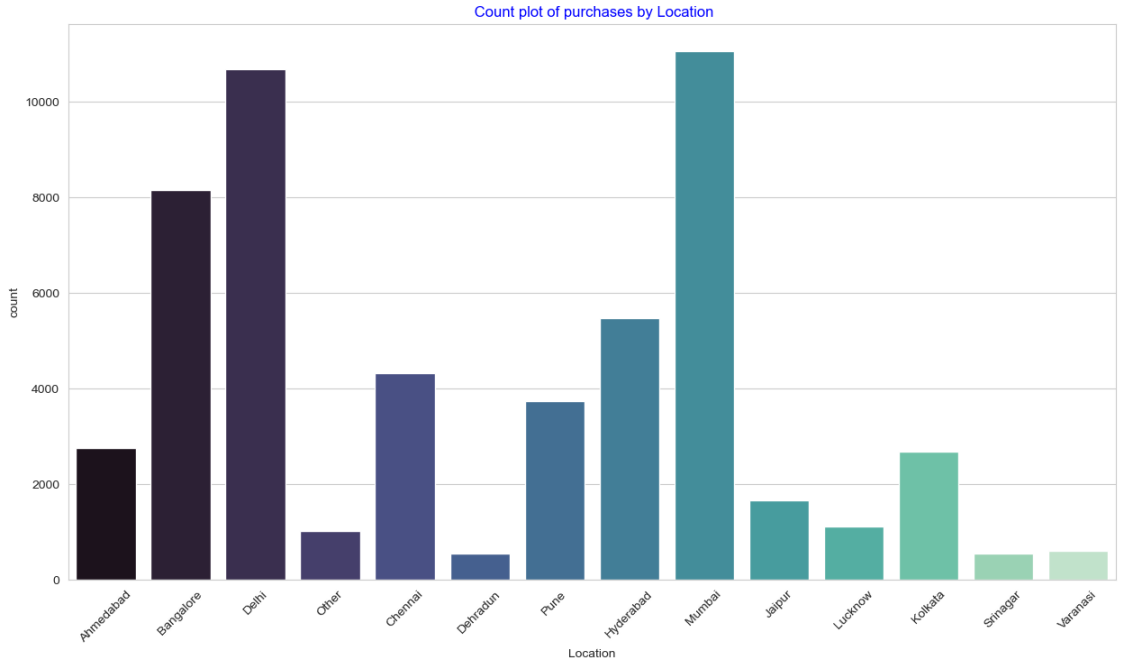
[31]:
```
# Age group distribution
plt.figure(figsize = (15,8))
sns.countplot(data, x='Age Group', order=['Below 18', '18-25', '25-45',␣
 ↪'45-60', '60 and above'], palette ='cividis')
plt.title("Customer Distribution by Age Group")
plt.show()
```



Customers in the 25-45 age group make the highest number of purchases, indicating that they are the most active and valuable segment. This group represents a strong potential target for marketing and promotional strategies.

In contrast, the Below 18 age group shows minimal or negligible purchasing activity, likely due to financial dependency on older age groups—particularly the 25-45 segment.

[32]:
```
# Top locations
plt.figure(figsize = (15,8))
sns.countplot(x='Location',data=data_f , palette='mako')
plt.title(' Count plot of purchases by Location', color ='blue')
plt.xticks(rotation=45)
plt.show()
```



The majority of purchases are concentrated in major metropolitan cities, with Mumbai, Delhi, followed by Bangalore, Hyderabad, and Chennai emerging as the top locations with the highest number of buyers.

On the other hand, cities like Srinagar, Varanasi, and Dehradun represent the lower end of the customer base, indicating minimal purchase activity from these regions.

But, the amount of transaction may be higher in any other cities? Analyse!

[33]:
```
data_f[data_f['Net Amount']>8000].Location.value_counts().reset_index() #8000␣
↪is the extreme/ highest Net amount observed
```

[33]:

	Location	count
0	Mumbai	6
1	Bangalore	6
2	Delhi	5
3	Pune	4
4	Ahmedabad	3
5	Kolkata	2
6	Chennai	2
7	Dehradun	1
8	Jaipur	1
9	Hyderabad	1
10	Srinagar	1
11	Lucknow	1

High-Value Transaction INSIGHTS:

The cities with the highest number of rare transactions—defined as purchases exceeding 8,000—are Mumbai, Bangalore, Delhi, and Pune.

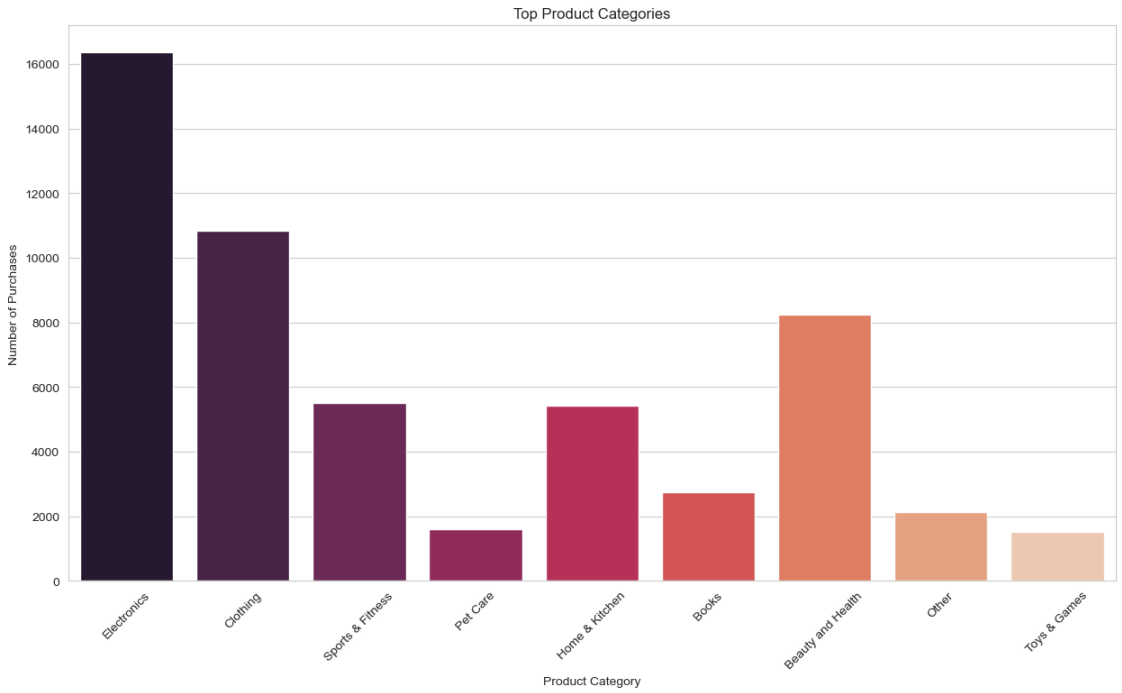
This indicates that the highest spending customers are predominantly located in these regions, making them an ideal target audience for premium product promotions and loyalty campaigns.

[34]:
```
data_f[data_f['Discount Availed']=='Yes'].Location.value_counts().reset_index()
```

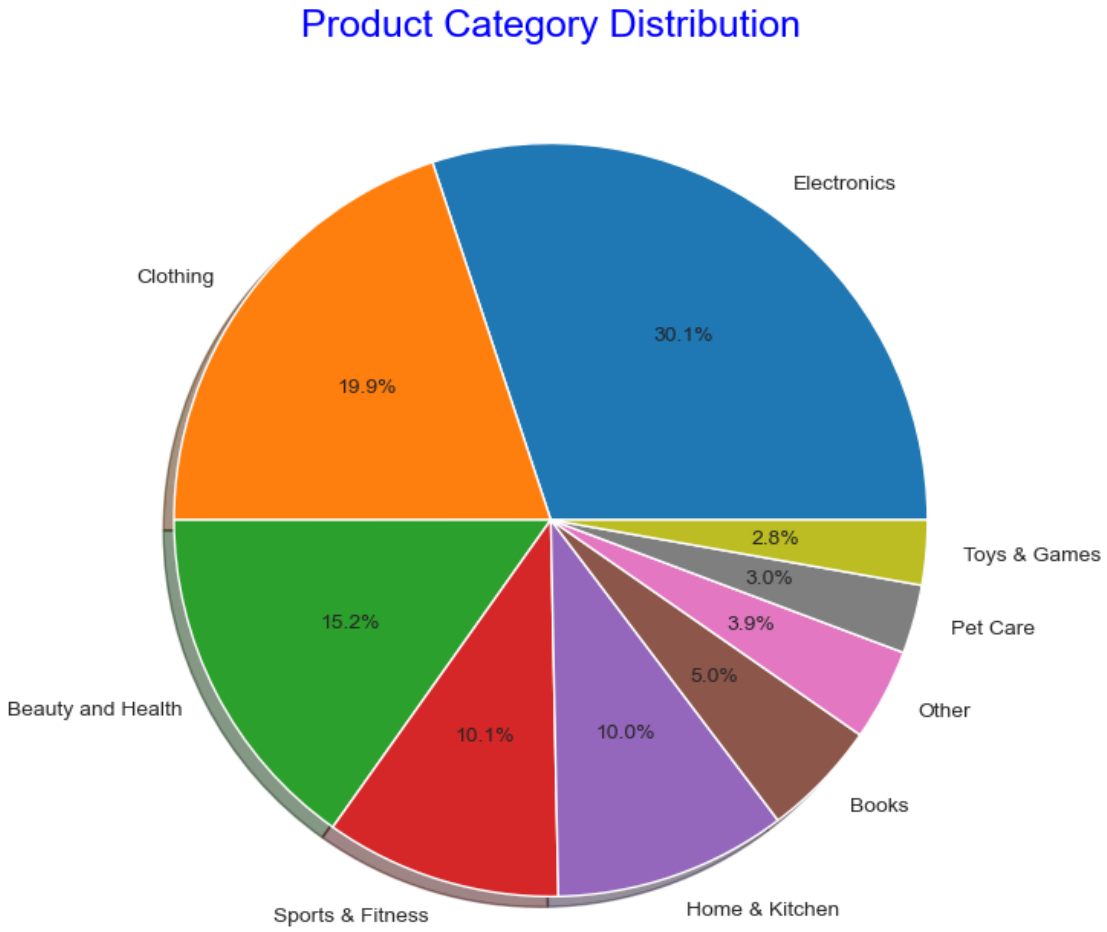
[34]:

	Location	count
0	Mumbai	5542
1	Delhi	5169
2	Bangalore	3980
3	Hyderabad	2783
4	Chennai	2141
5	Pune	1842
6	Ahmedabad	1363
7	Kolkata	1318
8	Jaipur	833
9	Lucknow	537
10	Other	505
11	Varanasi	275
12	Dehradun	268
13	Srinagar	246

[35]:
```
# Overall top product categories
plt.figure(figsize = (15,8))
sns.countplot(x='Product Category', data=data_f, palette='rocket')
plt.title("Top Product Categories")
plt.ylabel("Number of Purchases")
plt.xticks(rotation=45)
plt.show()
```



[36]:
```
category_counts = data_f['Product Category'].value_counts()
plt.figure(figsize = (15,8))
plt.pie(category_counts, labels= category_counts.index, autopct='%1.
  ↪1f%%',shadow=True)
plt.title('Product Category Distribution', color='blue', fontsize = 18)
plt.show()
```



Electronics stand out as the most popular product category, accounting for 30.1% of total purchases. This is followed by Clothing (19.9%) and Beauty & Health products (15.2%), which also demonstrate strong customer interest.

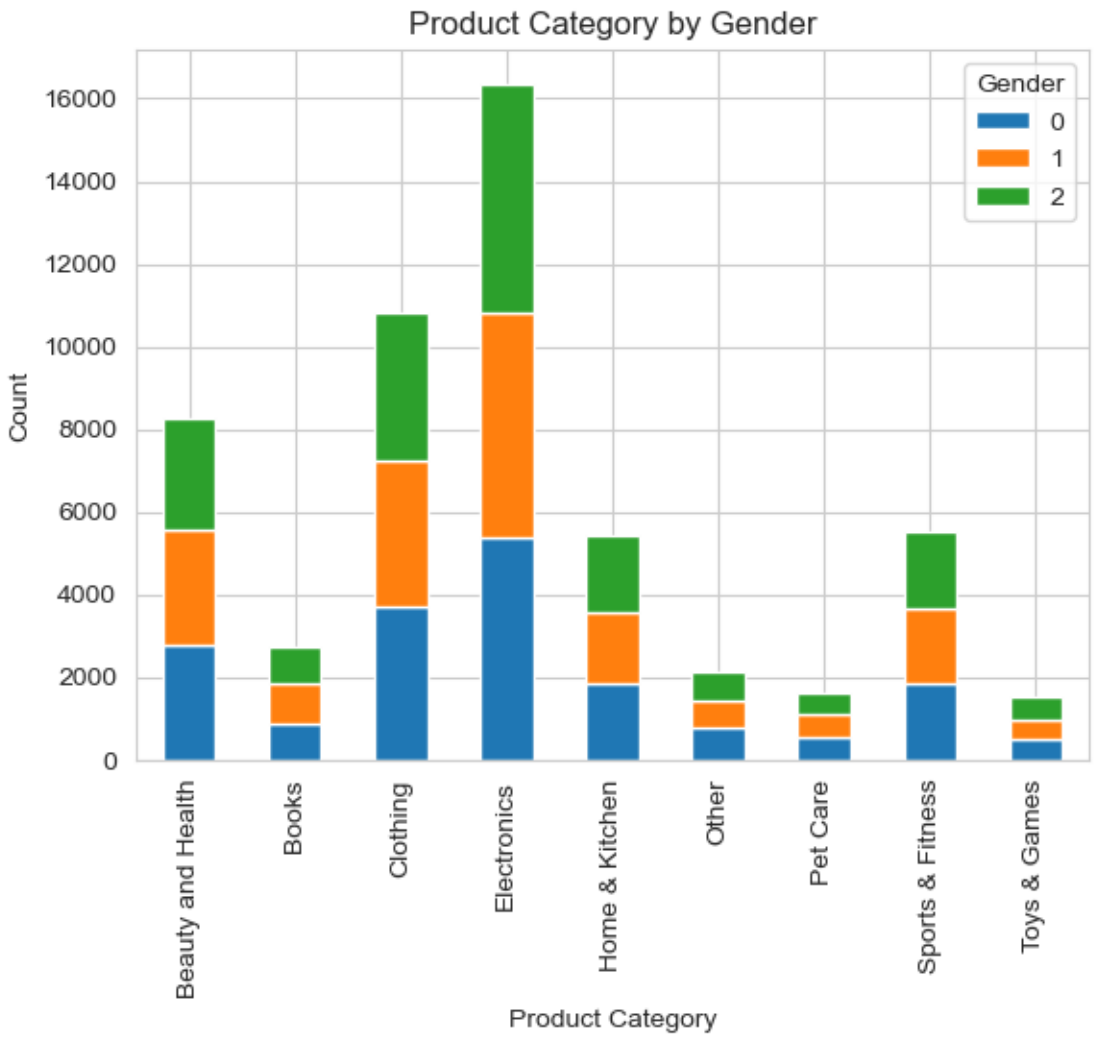
Categories such as Sports & Fitness (10.1%) and Home & Kitchen (10.0%) maintain a moderate level of engagement, indicating steady demand among shoppers.

In contrast, Games & Toys (2.8%) and Pet Care (3.0%) receive relatively lower attention, suggesting these are niche segments or areas with untapped potential within the current customer base.

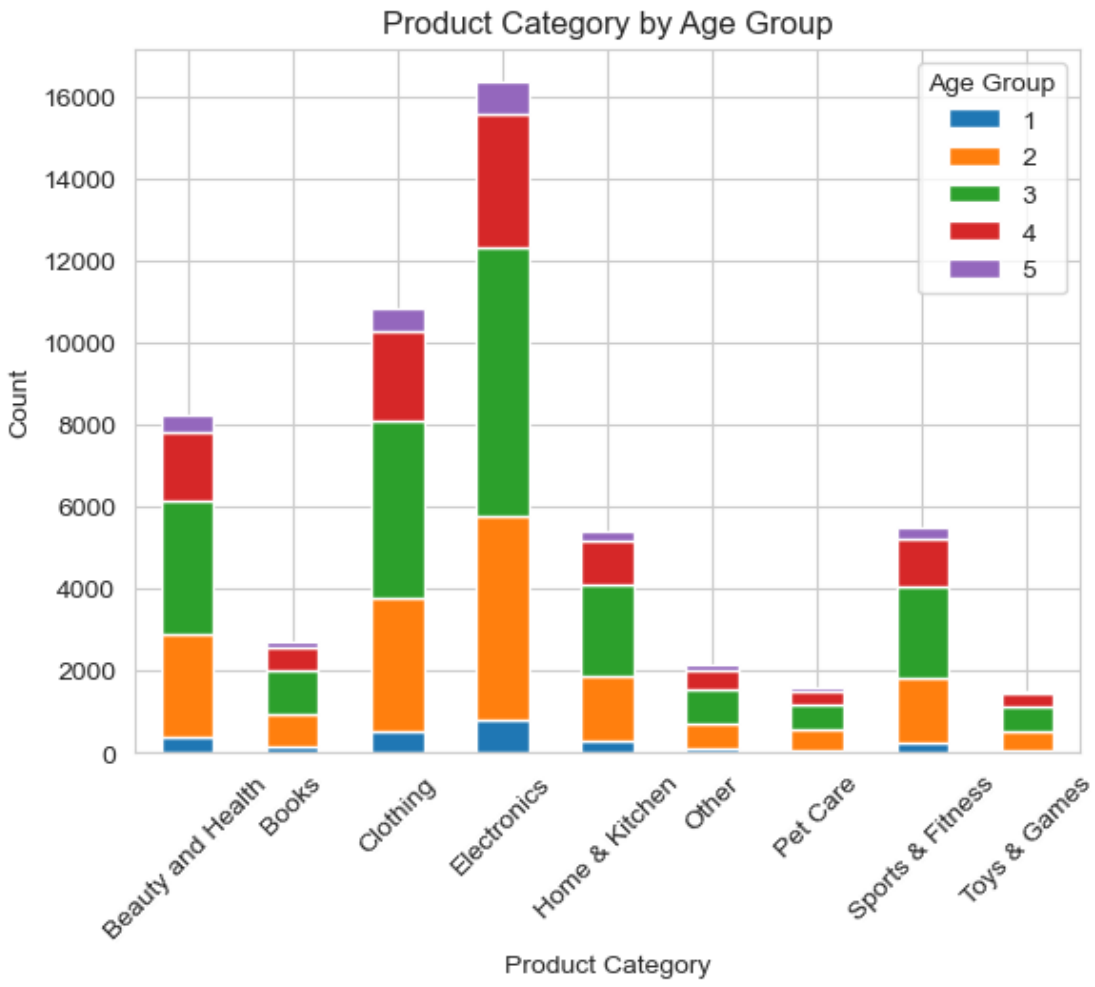
[37]:
```
# Category preference by Gender
plt.figure(figsize = (15,8))
pd.crosstab(data_f['Product Category'], data_f['Gender']).plot(kind='bar',␣
 ↪stacked=True)
plt.title("Product Category by Gender")
plt.ylabel("Count")
```

```
plt.show()
```

```
<Figure size 1500x800 with 0 Axes>
```



[38]:
```
# Category preference by Age Group
pd.crosstab(data_f['Product Category'], data_f['Age Group']).plot(kind='bar',␣
 ↪stacked=True)
plt.title("Product Category by Age Group")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.show()
```



[39]:
```
data_f['Discount Availed'].value_counts()
```

[39]:
```
Discount Availed
No      27585
Yes     26802
Name: count, dtype: int64
```

Observation: Discount is provided for only 50% of the overall purchase made

[41]:
```
#Discount Availed by Product Category
Discount_by_prod= data_f.groupby(['Product Category', 'Discount Availed']).
  ↪size().reset_index().rename(columns={0:'Discount Count'})
pd.set_option('display.max_rows',None)
Discount_by_prod
```

[41]:

	Product Category	Discount Availed	Discount Count
0	Beauty and Health	No	4244
1	Beauty and Health	Yes	4011
2	Books	No	1373
3	Books	Yes	1361
4	Clothing	No	5439
5	Clothing	Yes	5394
6	Electronics	No	8367
7	Electronics	Yes	7995
8	Home & Kitchen	No	2705
9	Home & Kitchen	Yes	2724
10	Other	No	1067
11	Other	Yes	1076
12	Pet Care	No	843
13	Pet Care	Yes	762
14	Sports & Fitness	No	2785
15	Sports & Fitness	Yes	2729
16	Toys & Games	No	762
17	Toys & Games	Yes	750

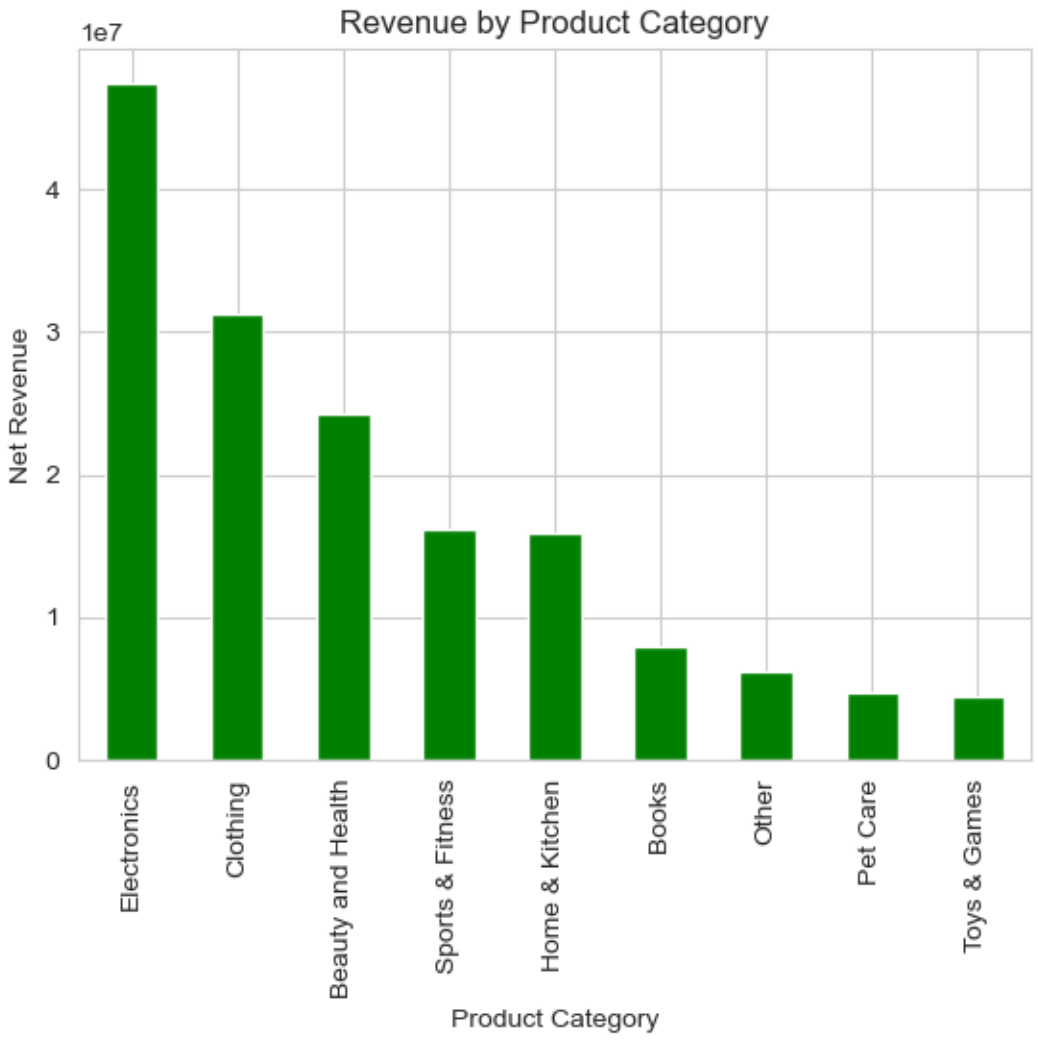
Discount Distribution & Product Category INSIGHTS

The data reveals that for each product category, approximately 50% of transactions involve discounts, indicating a balanced approach to promotional pricing across all segments.

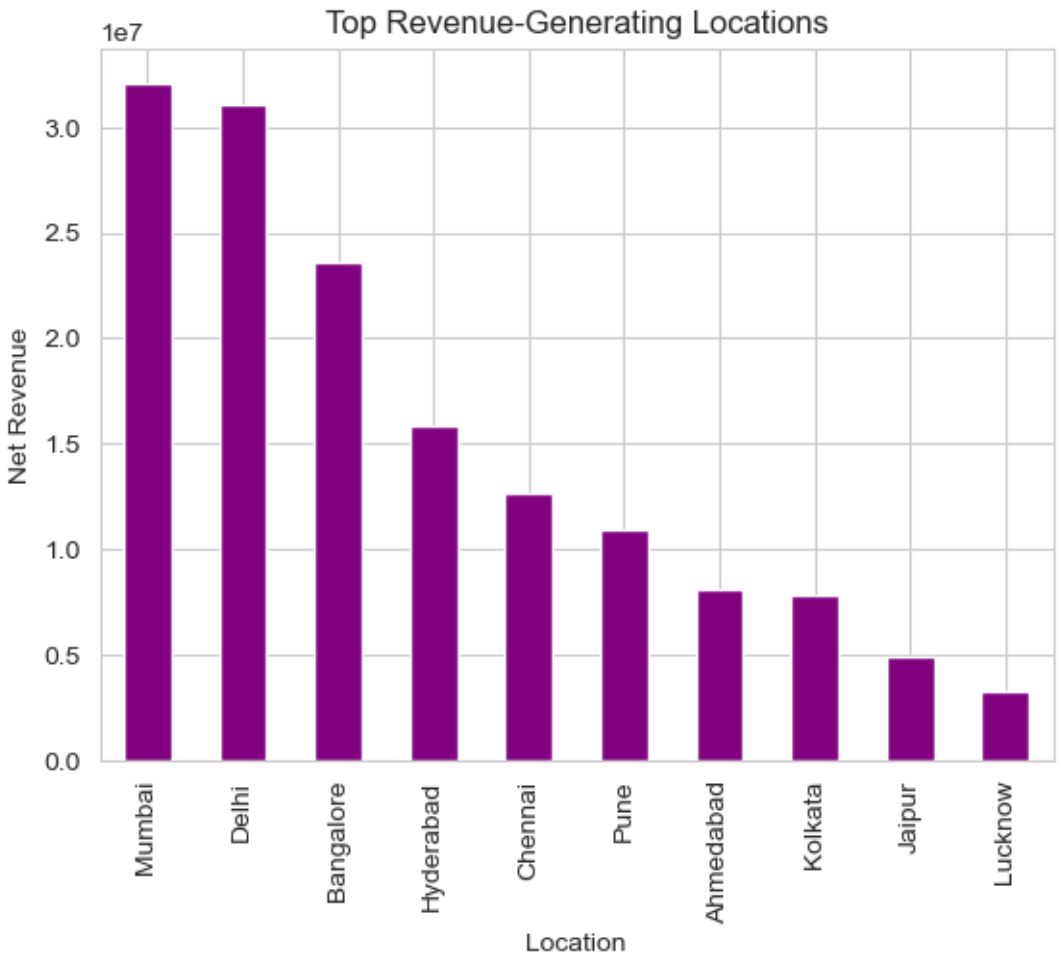
Among the categories, Electronics leads with the highest number of purchases, both with and without discounts, making it the most in-demand segment.

Conversely, Toys & Games, Pet Care, and Books have recorded the lowest purchase volumes, suggesting limited customer engagement in these categories.

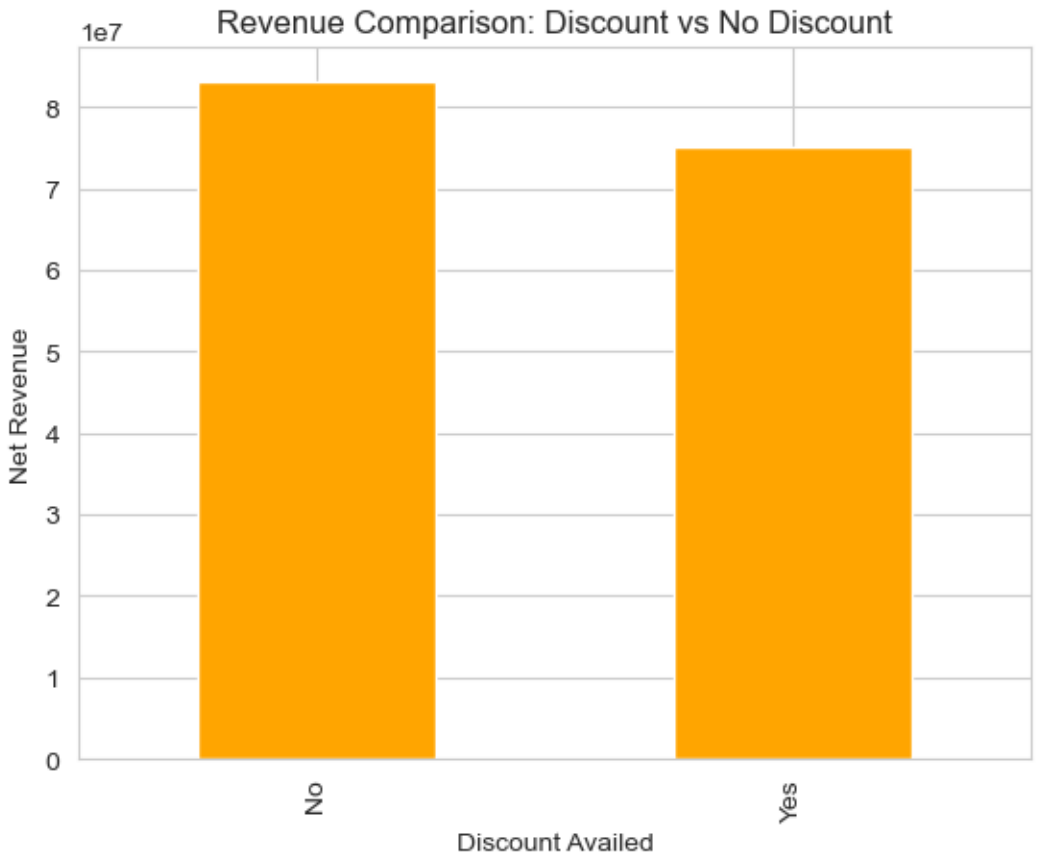
[40]:
```
# Revenue by category
category_revenue = data_f.groupby('Product Category')['Net Amount'].sum().
 ↪sort_values(ascending=False)
category_revenue.plot(kind='bar', color='green')
plt.title("Revenue by Product Category")
plt.ylabel("Net Revenue")
plt.show()
```



[42]:
```
# Revenue by location
top_rev_locations = data_f.groupby('Location')['Net Amount'].sum().
 ↪sort_values(ascending=False).head(10)
top_rev_locations.plot(kind='bar', color='purple')
plt.title("Top Revenue-Generating Locations")
plt.ylabel("Net Revenue")
plt.show()
```



[43]:
```
# Discount vs Non-discount Revenue
data_f.groupby('Discount Availed')['Net Amount'].sum().plot(kind='bar',␣
↪color='orange')
plt.title("Revenue Comparison: Discount vs No Discount")
plt.ylabel("Net Revenue")
plt.show()
```

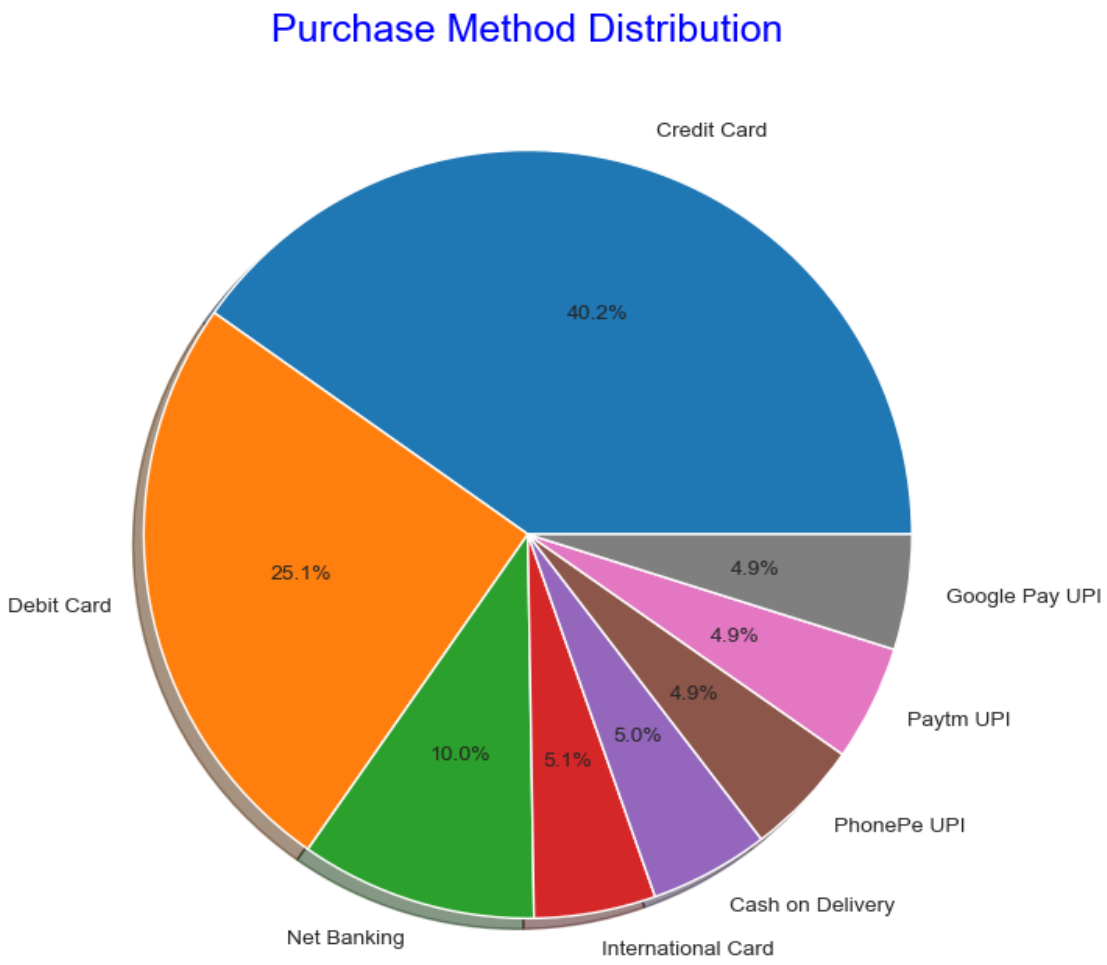


INSIGHTS: >Electronics, Clothing, and Beauty & Health are the top revenue-generating product categories, indicating strong customer demand and suggesting these segments should remain a key business focus.

Mumbai, Delhi, and Bangalore contribute the highest to overall revenue, making them ideal target markets for premium offerings, loyalty programs, and region-specific promotions.

Non-discounted purchases yield more revenue than discounted ones, highlighting that customers value product quality and are willing to pay full price—suggesting discounts should be used strategically to boost underperforming categories.

[46]:
```
payment_counts = data_f['Purchase Method'].value_counts()
plt.figure(figsize=(10, 8))
plt.pie(payment_counts, labels=payment_counts.index, autopct='%1.
↪1f%%',shadow=True)
plt.title('Purchase Method Distribution', color='blue', fontsize = 18)
plt.show()
```

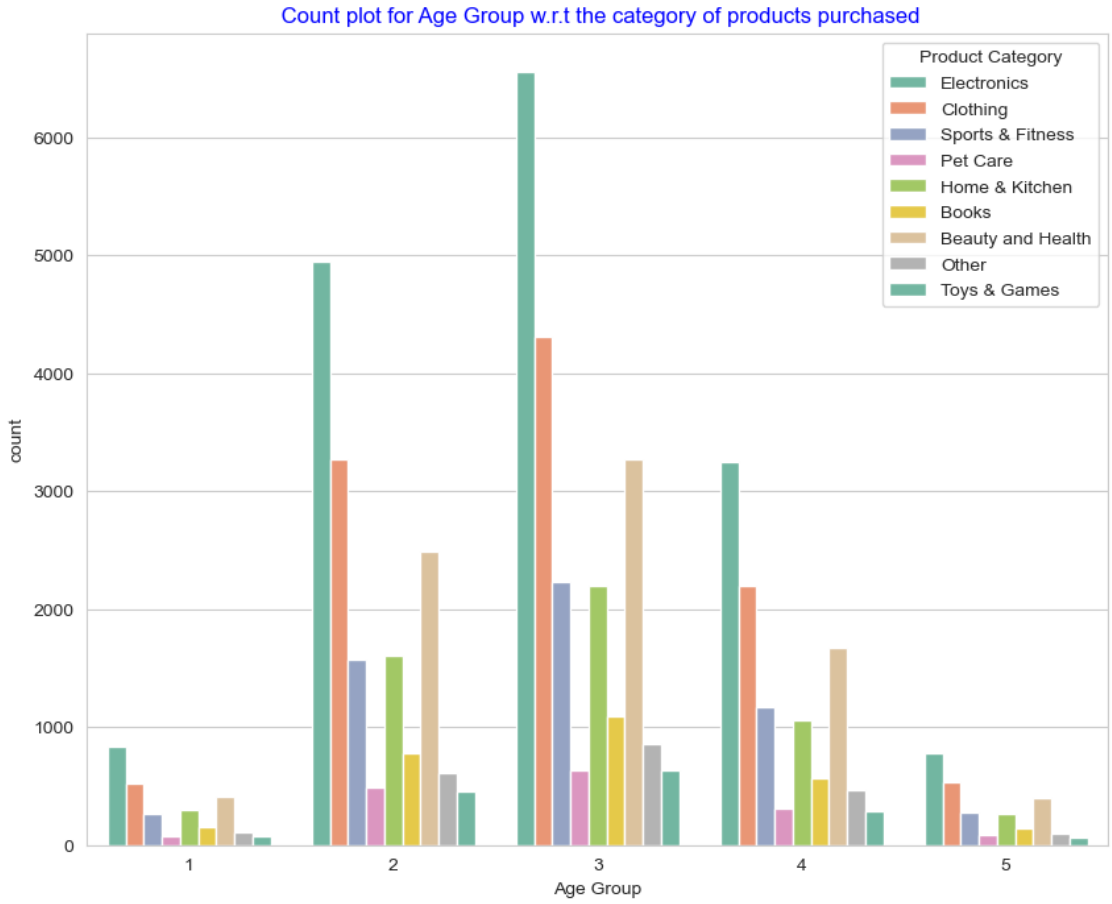


INSIGHTS:

Around 40% of the transactions are made using credit cards, while 25% are completed via debit cards. Interestingly, only 5% of purchases are made through cash on delivery. This indicates a strong customer preference for digital payments. To encourage continued usage and reward customer behavior, targeted discounts, offers, or cashback incentives can be introduced specifically for credit and debit card users. This strategy could further promote seamless transactions and enhance customer loyalty.

[49]:
```
plt.figure(figsize=(10,8))
sns.countplot(x='Age Group',hue= 'Product Category', data=data_f␣
 ↪,palette='Set2')
plt.title(' Count plot for Age Group w.r.t the category of products purchased',␣
 ↪color ='blue')
```

[49]: Text(0.5, 1.0, ' Count plot for Age Group w.r.t the category of products purchased')



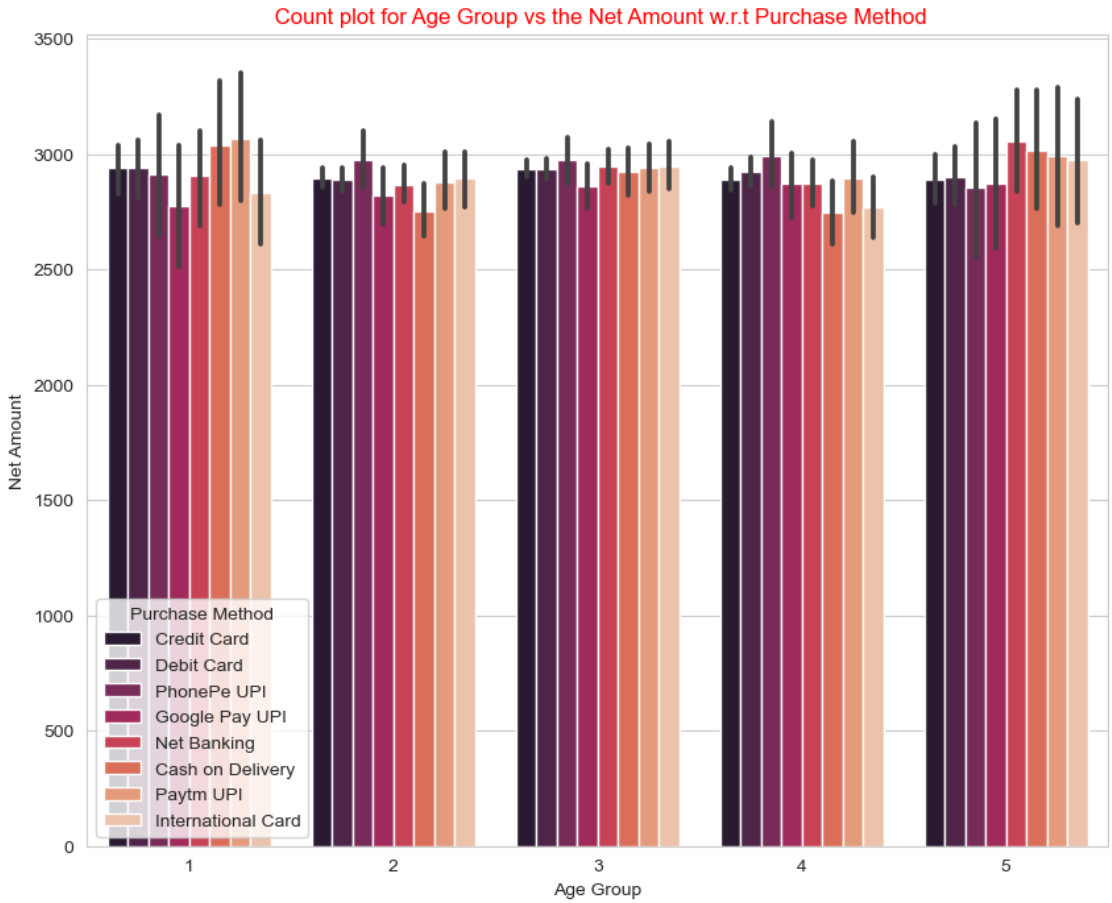
INSIGHTS:

Based on the distribution of age groups with respect to the categories of products they purchase, it is evident that individuals in the 'under 18' and 'above 60' age groups contribute the least to overall purchases. In contrast, the age group between '25–45' shows the highest purchasing activity, followed by moderate contributions from the '18–25' and '45–60' age groups.

This pattern likely reflects the income and spending capacity associated with different life stages. Individuals aged '18–25' are often at the beginning of their careers or pursuing education, resulting in average or limited spending. Similarly, those aged '45–60' may be approaching retirement, leading to more cautious or moderate expenditures. On the other hand, the '25–45' age group is typically at the peak of their earning potential and financial independence, which corresponds to their higher level of spending and purchasing activity. The minimal purchases by those 'under 18' and 'above 60' can be attributed to financial dependence or reduced income, aligning with expected trends in spending behavior across age groups.

[50]:
```
plt.figure(figsize=(10,8))
sns.barplot(x='Age Group', y='Net Amount', hue='Purchase Method', data=data_f,␣
 ↪palette='rocket')
plt.title(' Count plot for Age Group vs the Net Amount w.r.t Purchase Method',␣
 ↪color ='r')
```

[50]:
```
Text(0.5, 1.0, ' Count plot for Age Group vs the Net Amount w.r.t Purchase
Method')
```



OBSERVATION:

It is surprising to observe that while the frequency of purchases is significantly higher in the age group between 24 to 45 years, the overall amount spent is relatively consistent across all age groups. A closer look into the age-wise preference of payment methods reveals distinct patterns: - Under 18: Paytm UPI and Cash on Delivery are the most commonly used payment methods. - 18–25: PhonePe UPI is the dominant mode of payment. - 25–45: PhonePe UPI continues to be the most preferred, although other methods are also used with relatively similar frequency. - 45–60: PhonePe UPI remains the primary mode of payment. - 60 and above: Internet Banking and Cash on Delivery

are the most favored payment methods.

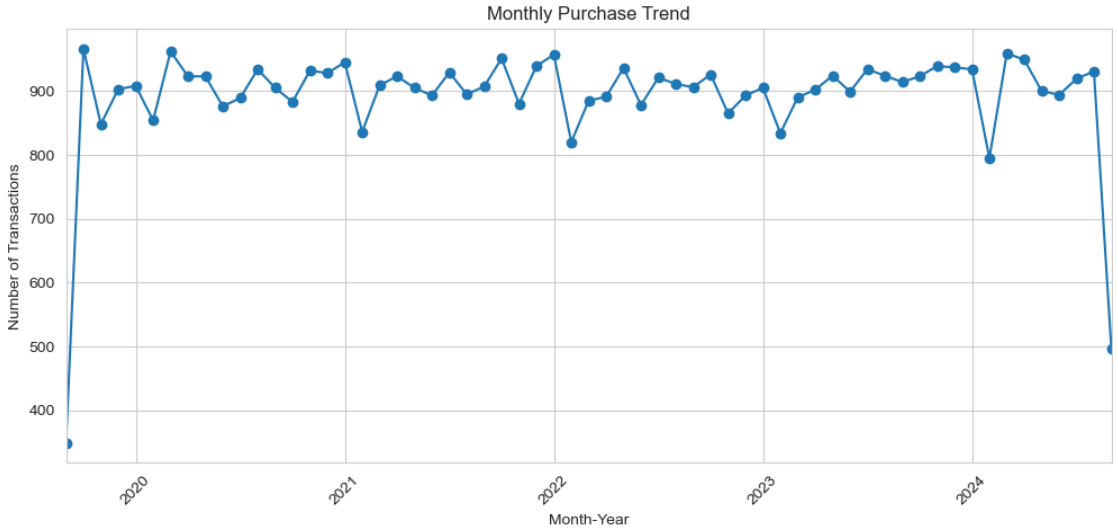
INRESTINGS INSIGHTS:

-PhonePe UPI emerges as the most widely used payment method across customers aged 18 to 60, making it a strong candidate for targeted promotions. Offering discounts or cashback on PhonePe UPI transactions could further encourage purchases within this segment. -Customers under 18 and those aged 60 and above still show a strong preference for Cash on Delivery, indicating a potential lack of trust in online transactions or comfort with traditional payment methods in these groups.

These findings from the 'Age Group vs Net Amount distribution w.r.t Payment Method' analysis are notably different from the broader 'Payment Method Distribution' results, which suggested that credit and debit cards are the most frequently used overall. This contrast highlights the importance of segment-specific analysis, as aggregate data may overlook age-based behavioral differences in payment preferences.

[44]:
```
# Convert to datetime
data_f['Purchase Date'] = pd.to_datetime(data_f['Purchase Date'], dayfirst=True)
data_f['Month-Year'] = data_f['Purchase Date'].dt.to_period('M')

# Purchases over time
monthly_purchases = data_f.groupby('Month-Year')['TID'].count()
monthly_purchases.plot(figsize=(12, 5), marker='o')
plt.title("Monthly Purchase Trend")
plt.ylabel("Number of Transactions")
plt.xticks(rotation=45)
plt.show()
```

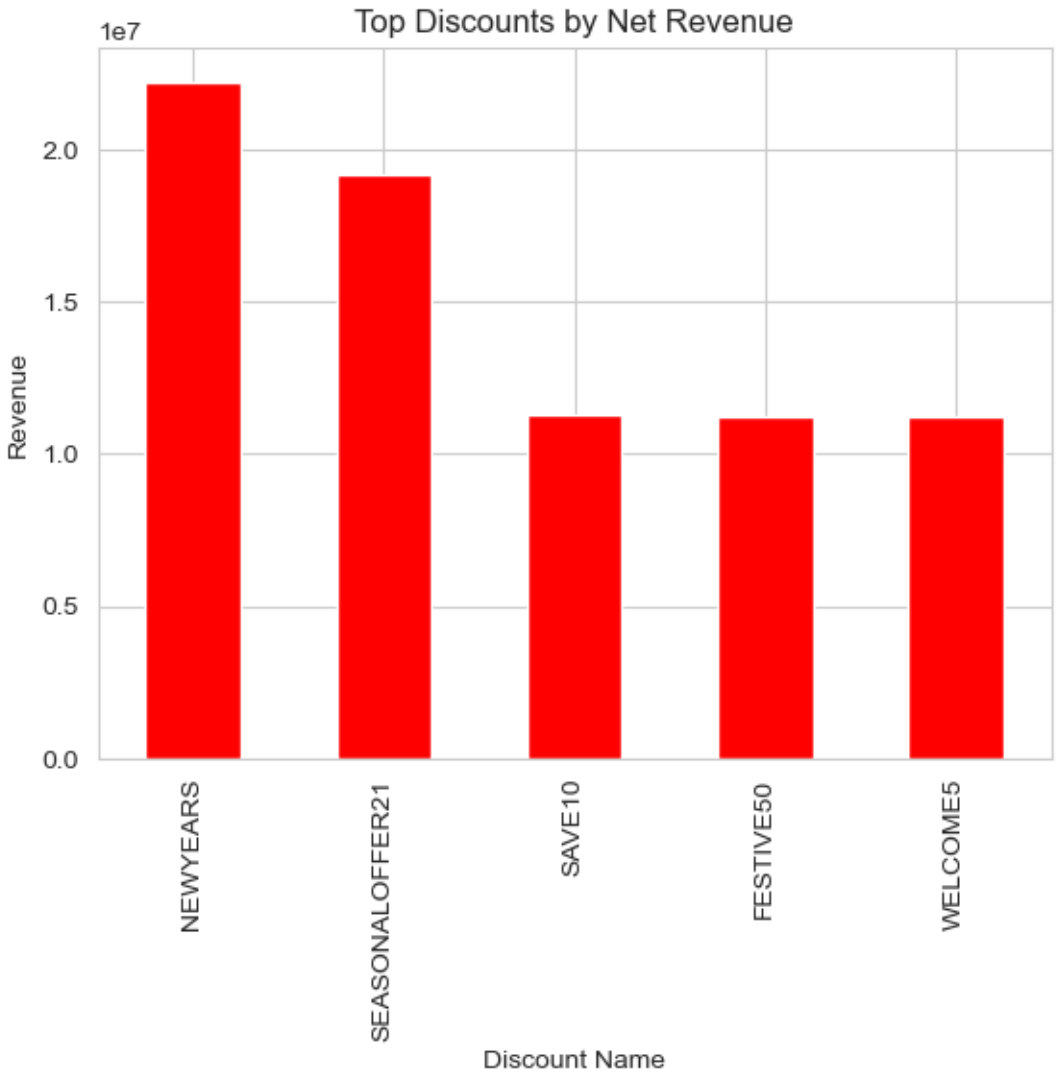


INSIGHTS:

The monthly purchase trend from 2020 to 2024 shows a stable volume of transactions,

mostly ranging between 850 and 950 per month. Minor dips are observed at the beginning and end of each year, possibly due to seasonality or data reporting lags. Despite these fluctuations, customer engagement appears strong and consistent throughout the years, indicating a stable user base and regular purchasing behavior.

[45]:
```
# Revenue by discount name
discount_impact = data_f[data_f['Discount Name'].notnull()].groupby('Discount␣
↪Name')['Net Amount'].sum().sort_values(ascending=False)
discount_impact.head(10).plot(kind='bar', color='red')
plt.title("Top Discounts by Net Revenue")
plt.ylabel("Revenue")
plt.show()
```



INSHIGHTS:

The top-performing discount campaigns in terms of net revenue are “NEWYEARS” and “SEASONALOFFER21”, which significantly outperformed others like “SAVE10,” “FESTIVE50,” and “WELCOME5”. This indicates that time-bound, event-driven promotions tend to be more effective in driving revenue compared to generic or smaller-value discounts. Leveraging this insight, businesses can strategically align future promotional campaigns around festive or seasonal events to maximize impact.

It’s important to note that 50% of the discount name data is missing. Further analysis of the available discount data shows that the median discount amount across all discount codes lies within a similar range typically between 250 to 300 INR. Codes such as FESTIVE50, SEASONALOFFER21, and WELCOME5 offer slightly higher median discounts compared to SAVE10 and NEWYEARS. Additionally, the overall discount amounts fall within a consistent range of 100–400 INR across all codes, suggesting minimal variation in discounting strategies. Given this uniformity, it’s recommended to diversify discount offerings introducing both smaller (e.g., 5%) and larger (e.g., 30%) discounts to create a sense of novelty and enhance customer engagement through psychological pricing tactics.

[ ]: