Python-Project-Bhanu-12318805-35

April 12, 2025

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Set visual style
sns.set(style="whitegrid")
plt.rcParams['figure.figsize'] = (10, 6)

# Load dataset
df = pd.read_csv('phone_usage_india_reduced.csv')

# Basic info
print(df.head())
print(df.info())
print(df.describe())
```

```
  User ID  Age  Gender    Location Phone Brand      OS  Screen Time (hrs/day)  \
0 U12001   58 Female Hyderabad    OnePlus Android                 3.3
1 U12002   47 Female    Mumbai   Motorola     iOS                 2.9
2 U12003   58  Other   Lucknow   Realme      iOS                 1.7
3 U12004   38   Male    Mumbai   Motorola Android                 5.0
4 U12005   37   Male      Pune      Apple Android                 3.4
Data Usage (GB/month) Calls Duration (mins/day) Number of Apps Installed \
0                 44.1 35.7  39
1                 22.4 249.7 128
2                 18.8 288.1 150
3                 40.8 207.3 171
4                 30.8 23.2  165
Social Media Time (hrs/day) E-commerce Spend (INR/month) \
0                      3.6 9234
1                      4.0 6593
2                      4.3 3981
3                      5.5 154
4                      2.2 4924
       Streaming Time (hrs/day) Gaming Time (hrs/day) \
0                    1.3    3.6
```

```
1                    1.1    3.2
2                    6.4    1.4
3                    6.4    4.2
4                    2.5    4.9
      Monthly Recharge Cost (INR) Primary Use
0                      1983     Gaming
1                      233 Work
2                      646 Gaming
3                      567 Work
4                      1778 Entertainment
<class
'pandas.core.frame.DataFrame'>
RangeIndex: 5686 entries, 0 to
5685 Data columns (total 16
columns):
#   Column                    Non-Null Count Dtype
--- ------                    -------------- -----
0   User ID                   5686 non-null object
1    Age                      5686 non-null int64
2   Gender                    5686 non-null object
3   Location                  5686 non-null object
4   Phone Brand               5686 non-null object
5    OS                       5686 non-null object
6  Screen Time (hrs/day)      5686 non-null float64
7  Data Usage (GB/month)      5686 non-null float64
8  Calls Duration (mins/day)  5686 non-null float64
9  Number of Apps Installed   5686 non-null int64
 10 Social Media Time         5686 non-null float64
 (hrs/day)
 11 E-commerce Spend (INR/month) 5686 non-  int64
 null
12 Streaming Time (hrs/day)   5686 non-null float64
13 Gaming Time (hrs/day)      5686 non-null float64
14 Monthly Recharge Cost (INR)5686 non-null int64
15 Primary Use                5686 non-null object
dtypes: float64(6), int64(4), object(6)
memory usage: 710.9+ KB
None
          Age Screen Time (hrs/day) Data Usage (GB/month) \
count 5686.000000        5686.000000        5686.000000
mean    37.537988           6.505153          25.433961
std     13.406711           3.177343          14.119625
```

```
min     15.000000           1.000000            1.000000
25%     26.000000           3.700000           13.025000
50%     38.000000           6.500000           25.600000
75%     49.000000           9.200000           37.500000
max     60.000000          12.000000           50.000000
    Calls Duration (mins/day) Number of Apps Installed \

count            5686.000000           5686.000000
mean              150.445445            104.251319
std                84.890449             54.929894
min                 5.000000             10.000000
25%                77.300000             58.000000
50%               148.900000            103.000000
75%               222.375000            151.000000
max               300.000000            200.000000

      Social  Media  Time  (hrs/day)  E-commerce  Spend
      (INR/month) \
count              5686.000000              5686.000000
mean                  3.270612              5117.840661
std                   1.583853              2853.038252
min                   0.500000               103.000000
25%                   1.900000              2662.250000
50%                   3.300000              5142.000000
75%                   4.600000              7610.000000
max                   6.000000              9995.000000

      Streaming Time (hrs/day) Gaming Time (hrs/day) \
count           5686.000000         5686.000000
mean               4.263120            2.503693
std                2.154641            1.442080
min                0.500000            0.000000
25%                2.400000            1.200000
50%                4.200000            2.500000
75%                6.100000            3.800000
max                8.000000            5.000000

      Monthly Recharge Cost (INR)
count               5686.000000
mean                1036.512663
std                  550.611059
min                  100.000000
25%                  559.000000
50%                 1019.500000
75%                 1517.750000
```

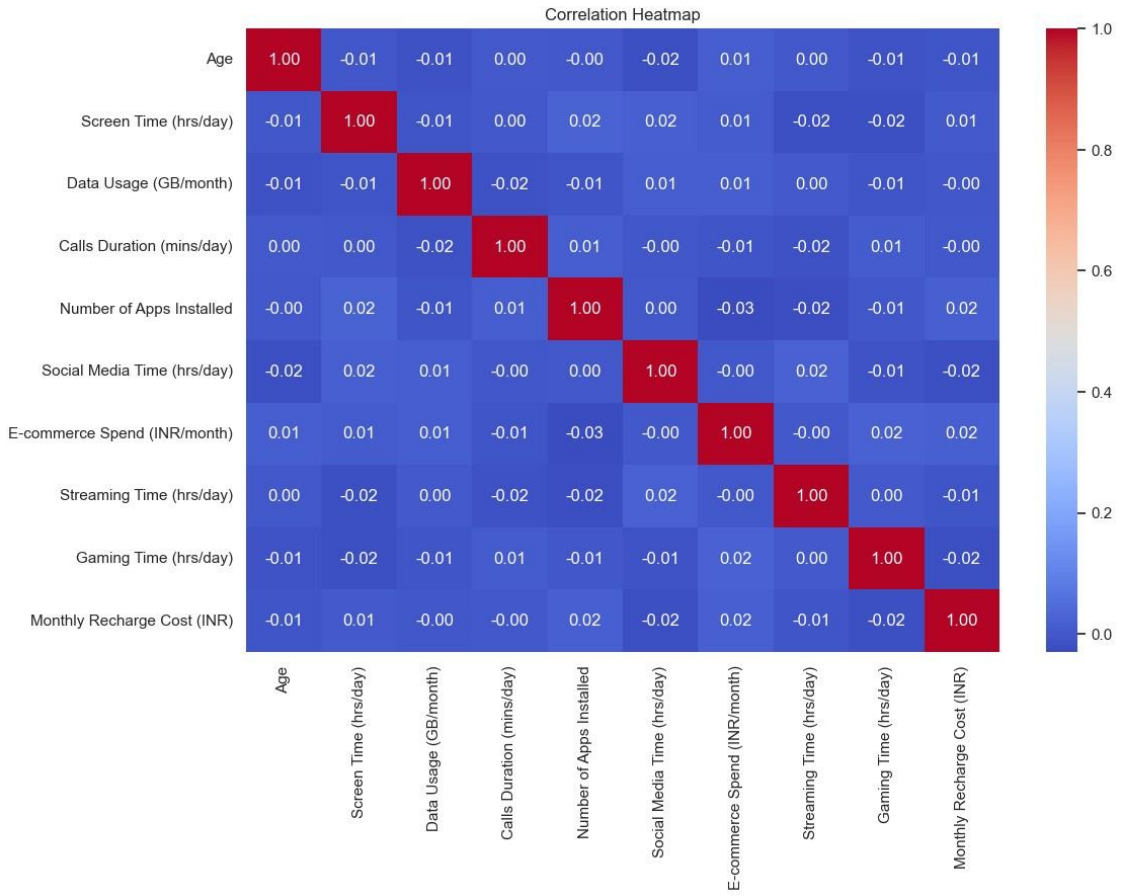
```
max                   2000.000000
```

[2]:
```
# Checking for null values
print(df.isnull().sum())

# Data types
print(df.dtypes)

# Unique values in each column
for col in df.columns:
    print(f"{col}: {df[col].nunique()} unique values")
```

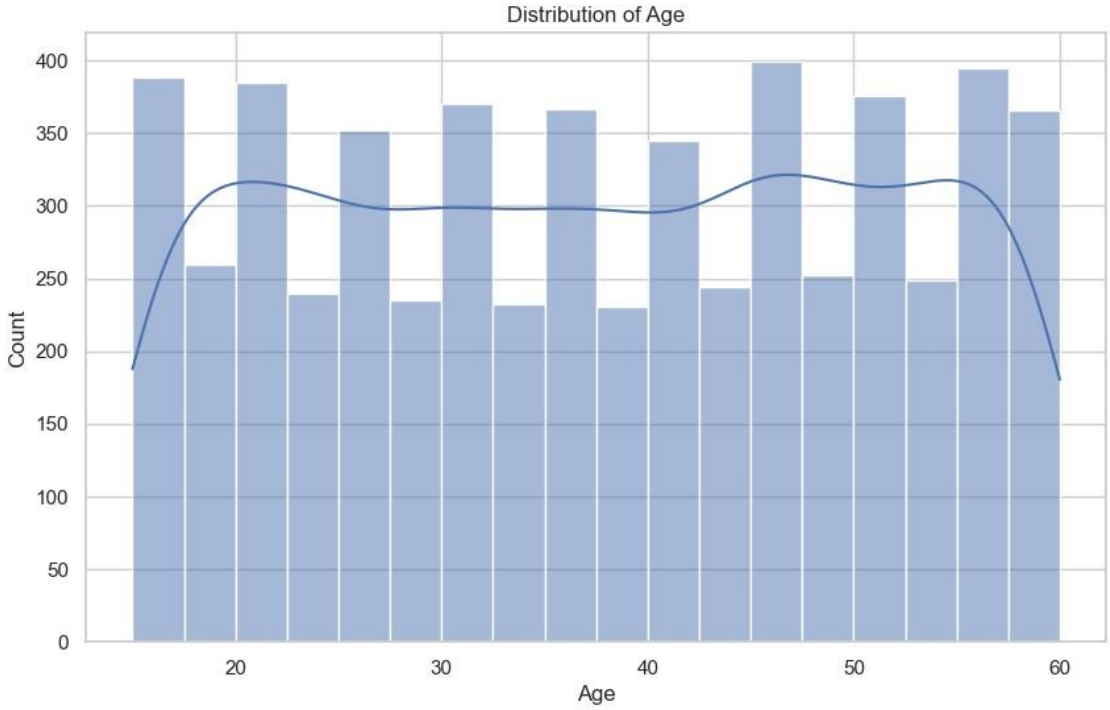
```
User ID                       0
Age                           0
Gender                        0
Location                      0
Phone Brand                   0
OS                            0
Screen Time (hrs/day)         0
Data Usage (GB/month)         0
Calls Duration (mins/day)     0
Number of Apps Installed      0
Social Media Time             0
(hrs/day)
E-commerce Spend              0
(INR/month)
Streaming Time (hrs/day)      0
Gaming Time (hrs/day)         0
Monthly Recharge Cost         0
(INR)
Primary Use                   0
dtype: int64
User ID                        object
Age                             int64
Gender                         object
Location                       object
Phone Brand                    object
OS                             object
Screen Time (hrs/day)         float64
Data Usage (GB/month)         float64
Calls Duration (mins/day)     float64
Number of Apps Installed        int64
Social Media Time             float64
(hrs/day)
E-commerce Spend                int64
(INR/month)
Streaming Time (hrs/day)      float64
Gaming Time (hrs/day)         float64
```

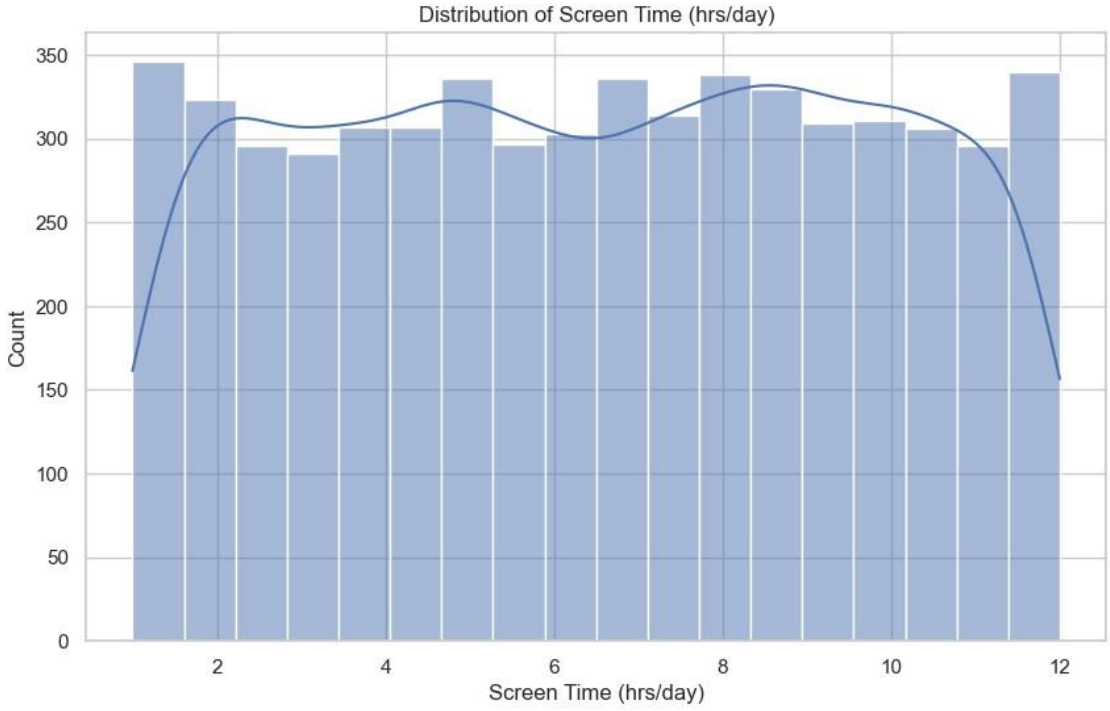
```
Monthly Recharge Cost        int64
(INR)
Primary Use                 object
dtype: object User ID:
5686 unique values
Age: 46 unique values
Gender: 3 unique values
Location: 10 unique values
Phone Brand: 10 unique values
OS: 2 unique values
Screen Time (hrs/day): 111 unique values
Data Usage (GB/month): 491 unique values
Calls Duration (mins/day): 2518 unique values
Number of Apps Installed: 191 unique values
Social Media Time (hrs/day): 56 unique values E-commerce Spend
(INR/month): 4332 unique values
Streaming Time (hrs/day): 76 unique values
Gaming Time (hrs/day): 51 unique values
Monthly Recharge Cost (INR): 1800 unique values
Primary Use: 5 unique values
```

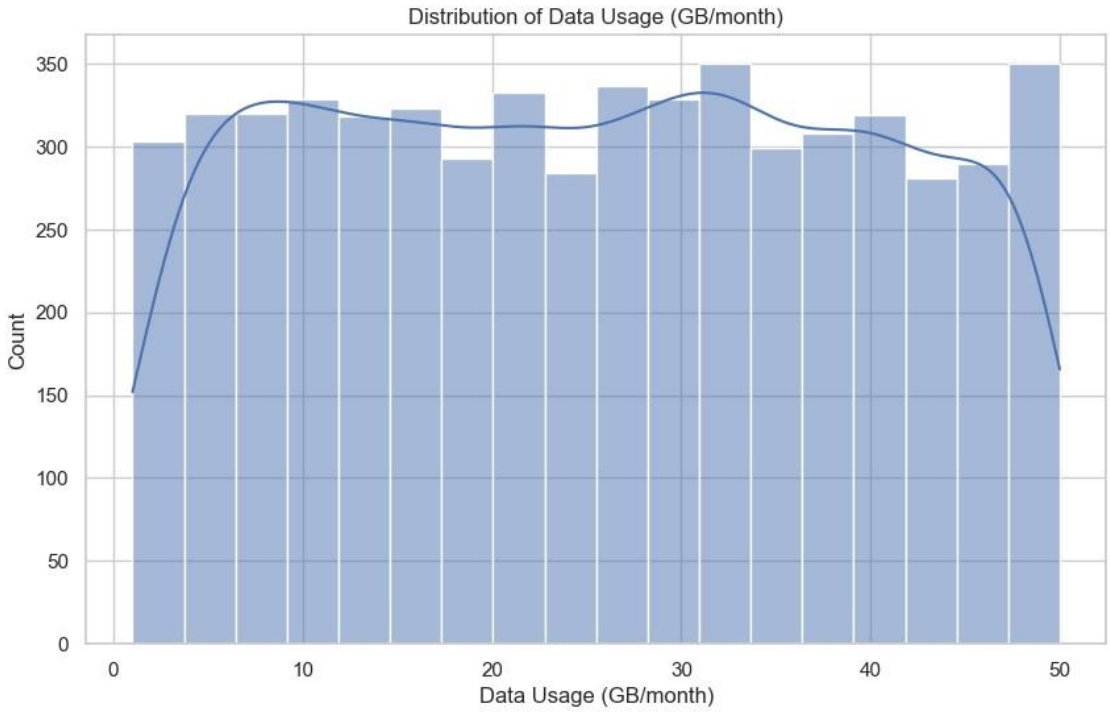
[3]:
```
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```

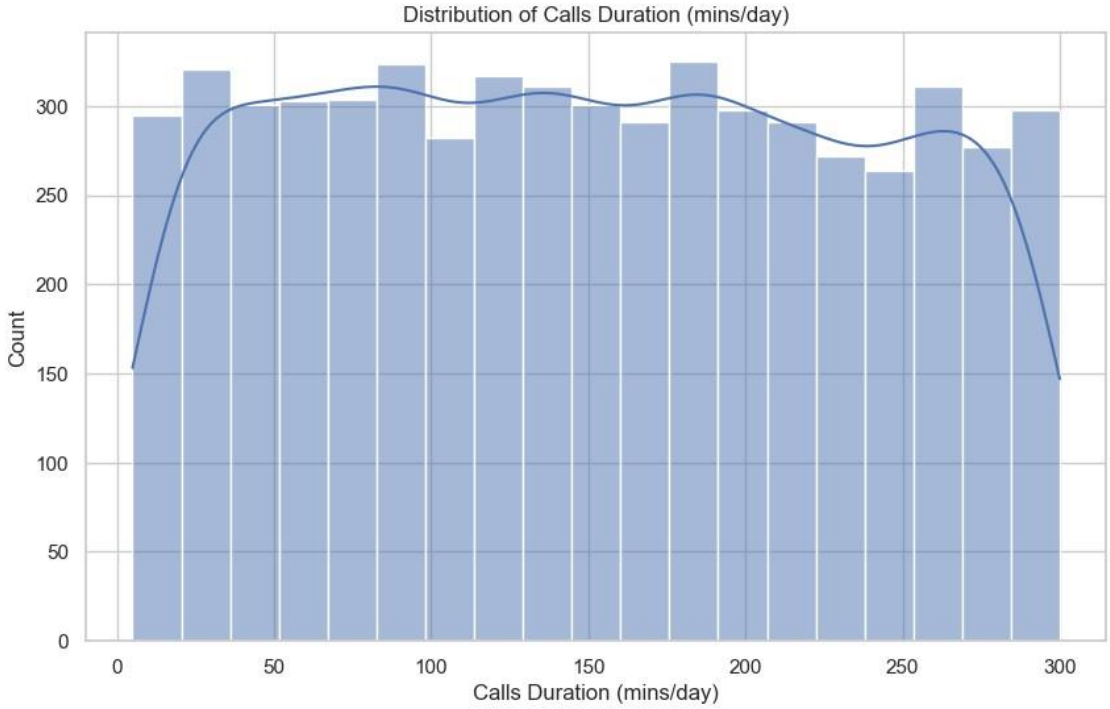


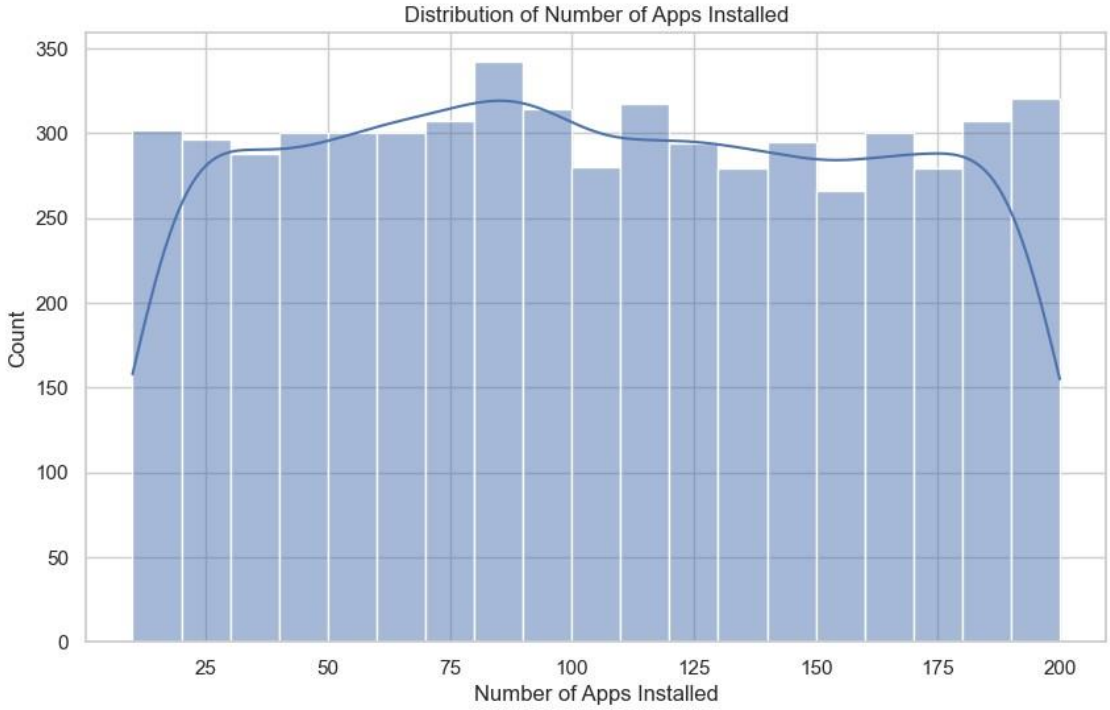
[4]:
```
for col in df.select_dtypes(include=np.number).columns:
    plt.figure()
    sns.histplot(df[col], kde=True)
    plt.title(f"Distribution of {col}")
    plt.show()
```

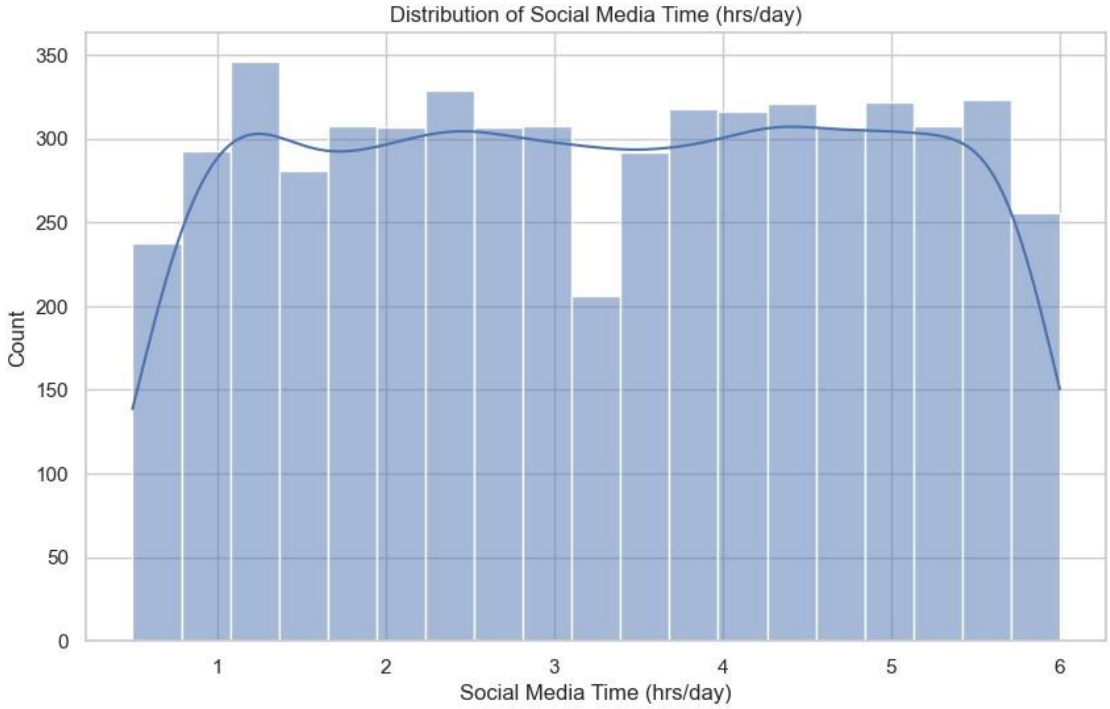

Distribution of Age
Count
Age
0
50
100
150
200
250
300
350
400
20
30
40
50
60

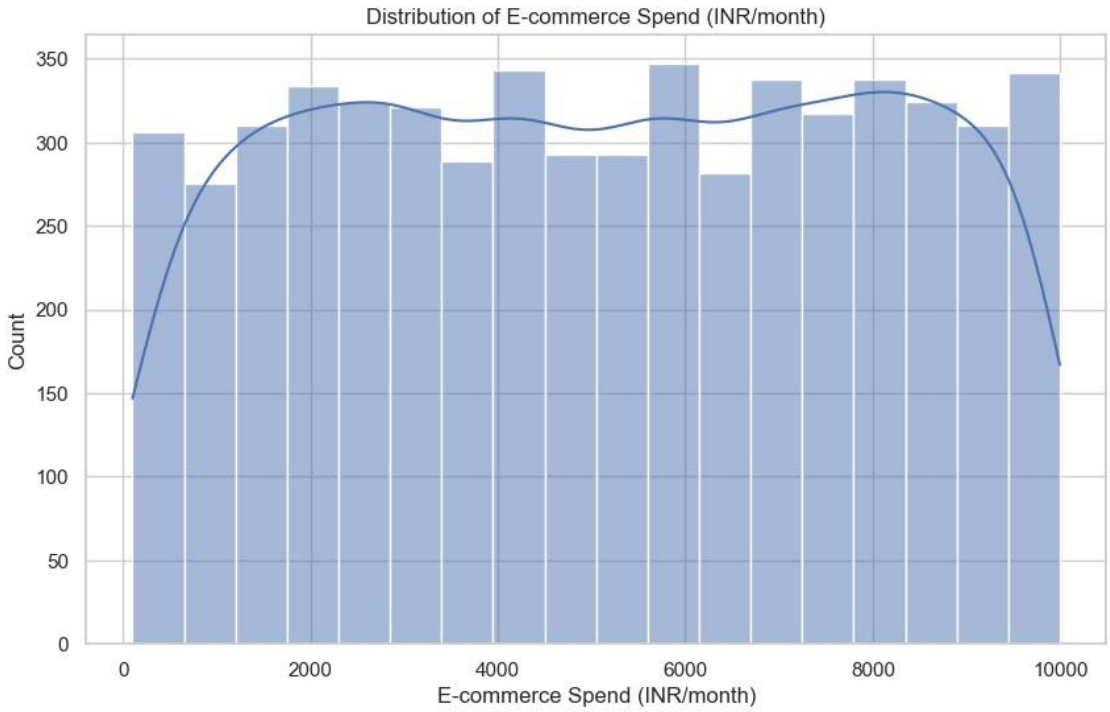

Distribution of Screen Time (hrs/day)
Count
Screen Time (hrs/day)
0
50
100
150
200
250
300
350
2
4
6
8
10
12

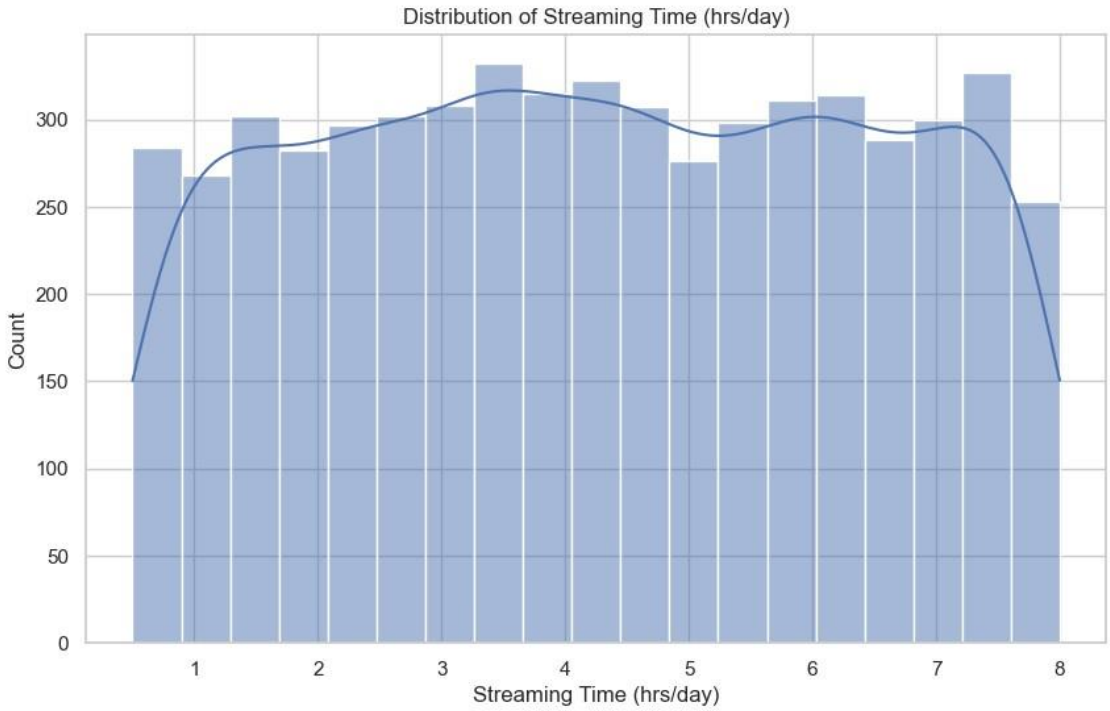

Distribution of Data Usage (GB/month)
Count
Data Usage (GB/month)
0
50
100
150
200
250
300
350
0
10
20
30
40
50

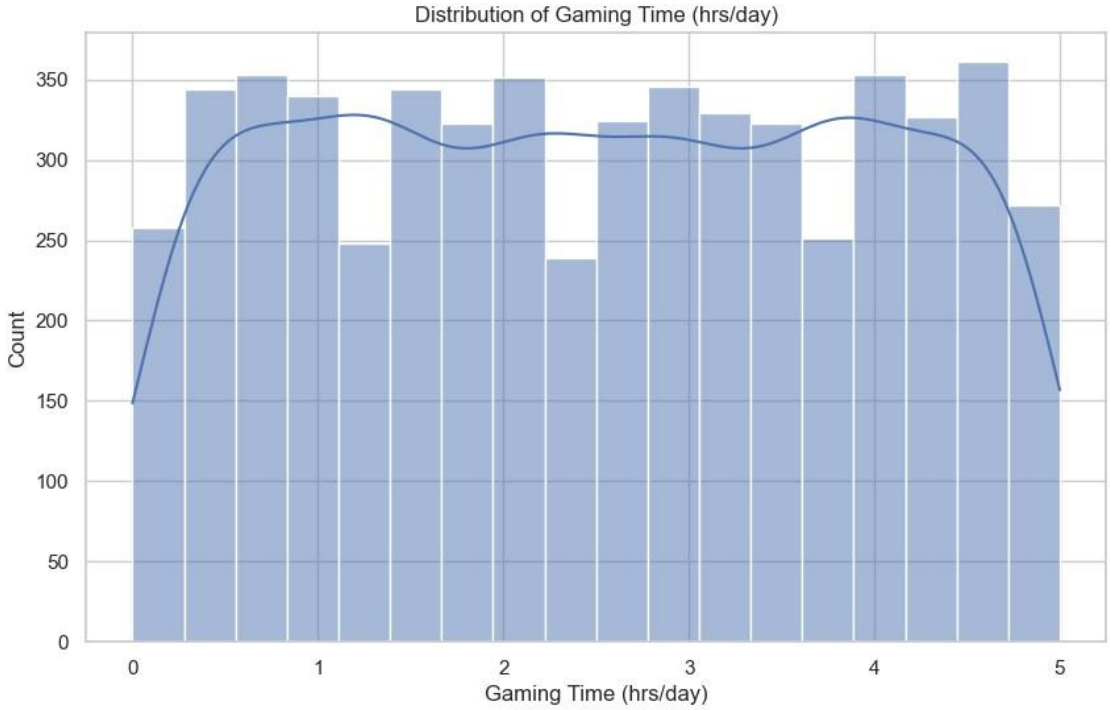

Distribution of Calls Duration (mins/day)
Count
Calls Duration (mins/day)
0
50
100
150
200
250
300
0
50
100
150
200
250
300

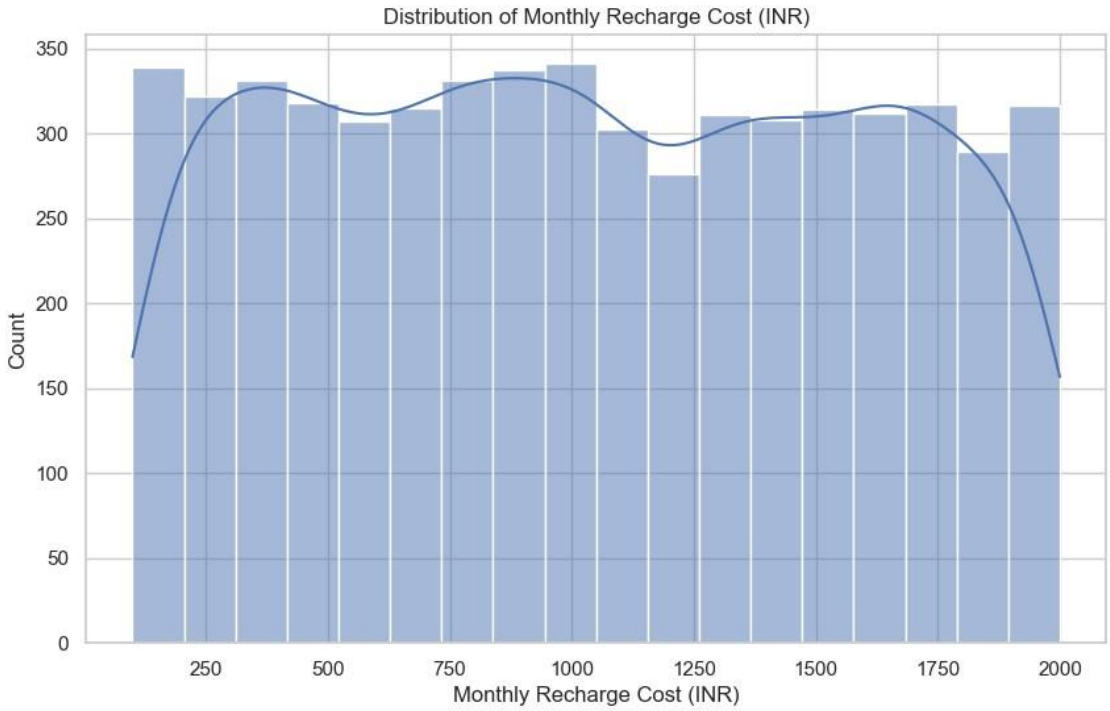

Distribution of Number of Apps Installed
Count
0
50
100
150
200
250
300
350
25
50
75
100
125
150
175
200
Number of Apps Installed


Distribution of Social Media Time (hrs/day)
Count
0
50
100
150
200
250
300
350
1
2
3
4
5
6
Social Media Time (hrs/day)

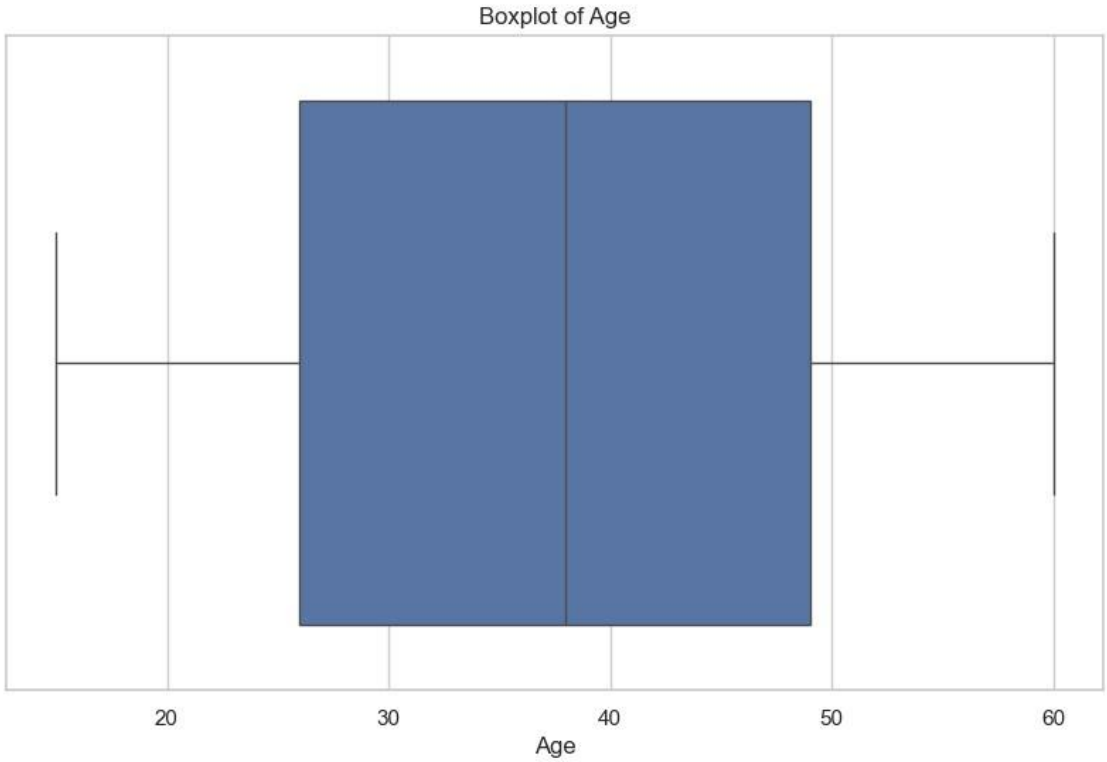

Distribution of E-commerce Spend (INR/month)
Count
E-commerce Spend (INR/month)
0
50
100
150
200
250
300
350
0
2000
4000
6000
8000
10000

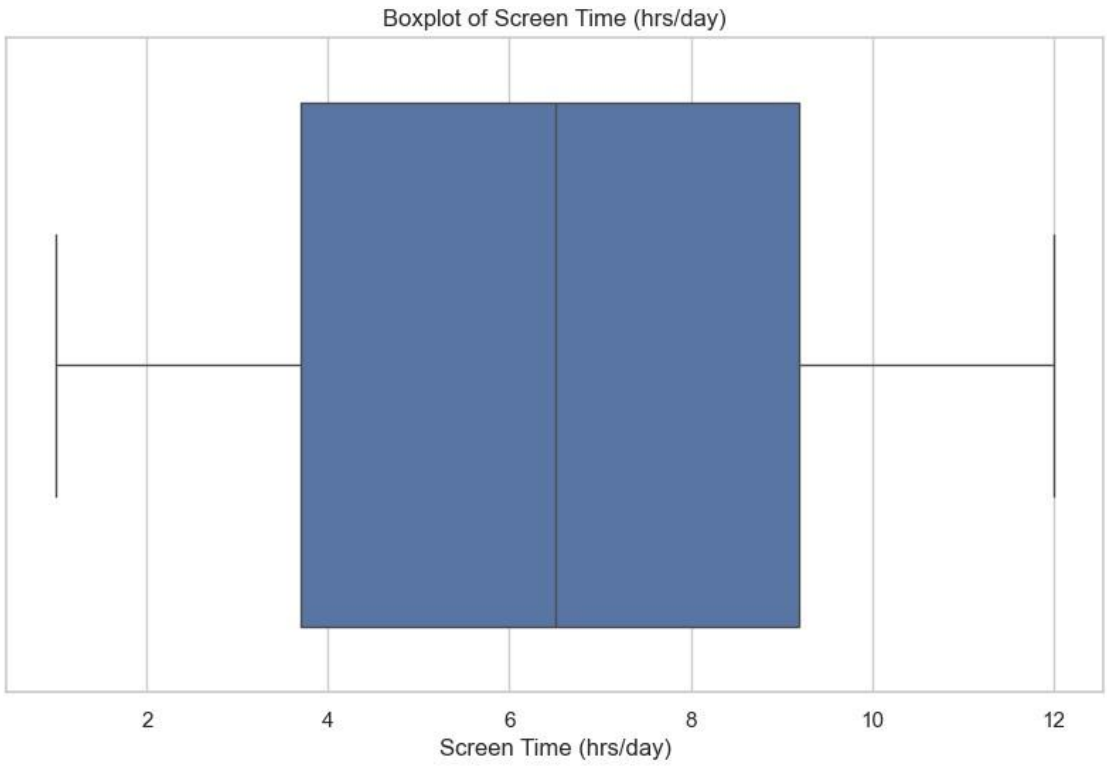

Distribution of Streaming Time (hrs/day)
Count
Streaming Time (hrs/day)
0
50
100
150
200
250
300
1
2
3
4
5
6
7
8

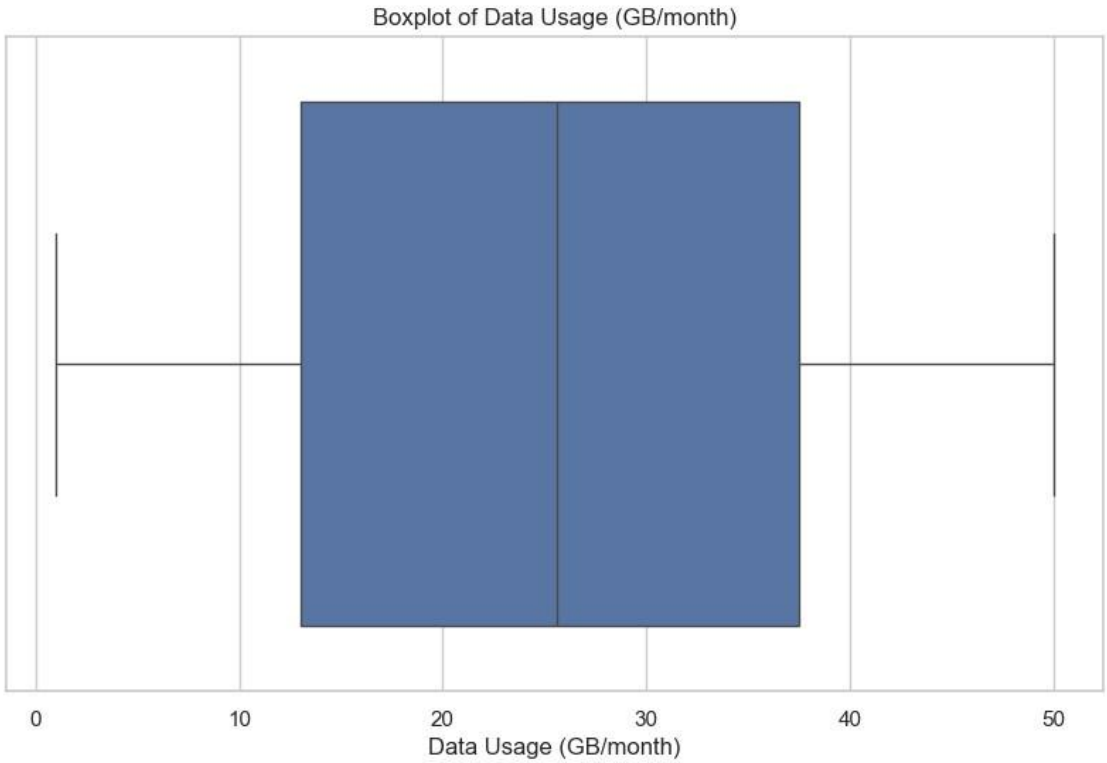

Distribution of Gaming Time (hrs/day)
Count
Gaming Time (hrs/day)
0
50
100
150
200
250
300
350
0
1
2
3
4
5

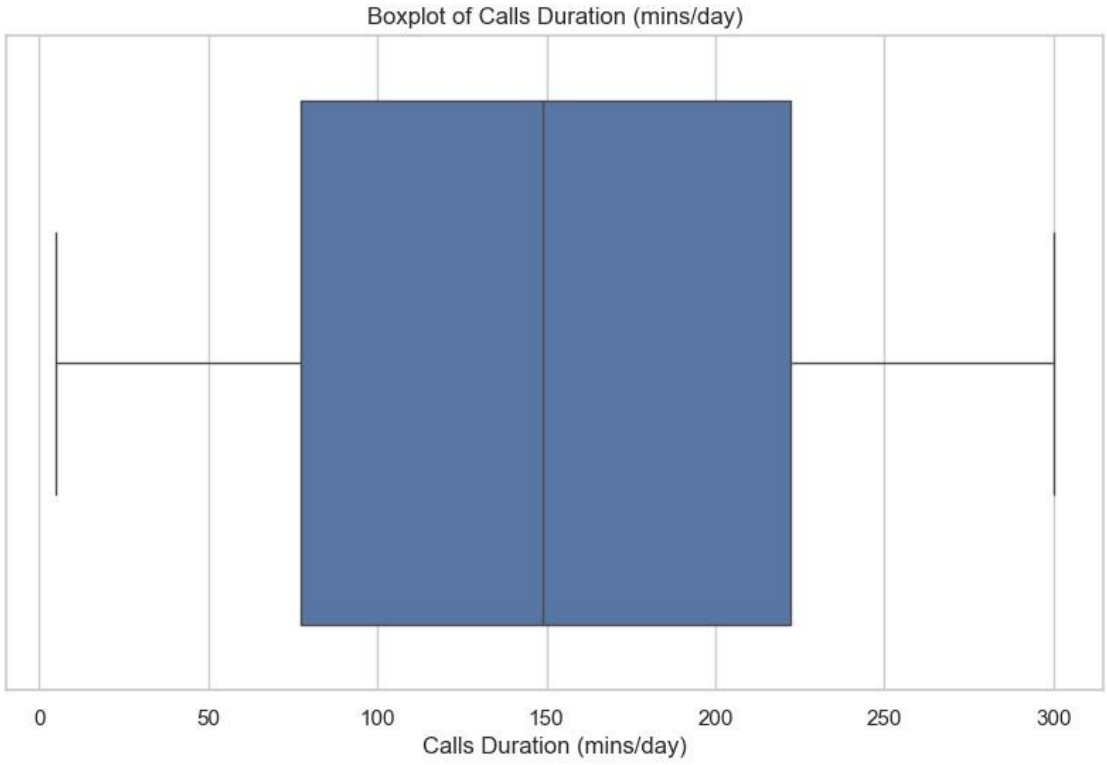

Distribution of Monthly Recharge Cost (INR)
Count
Monthly Recharge Cost (INR)
0
50
100
150
200
250
300
350
250
500
750
1000
1250
1500
1750
2000

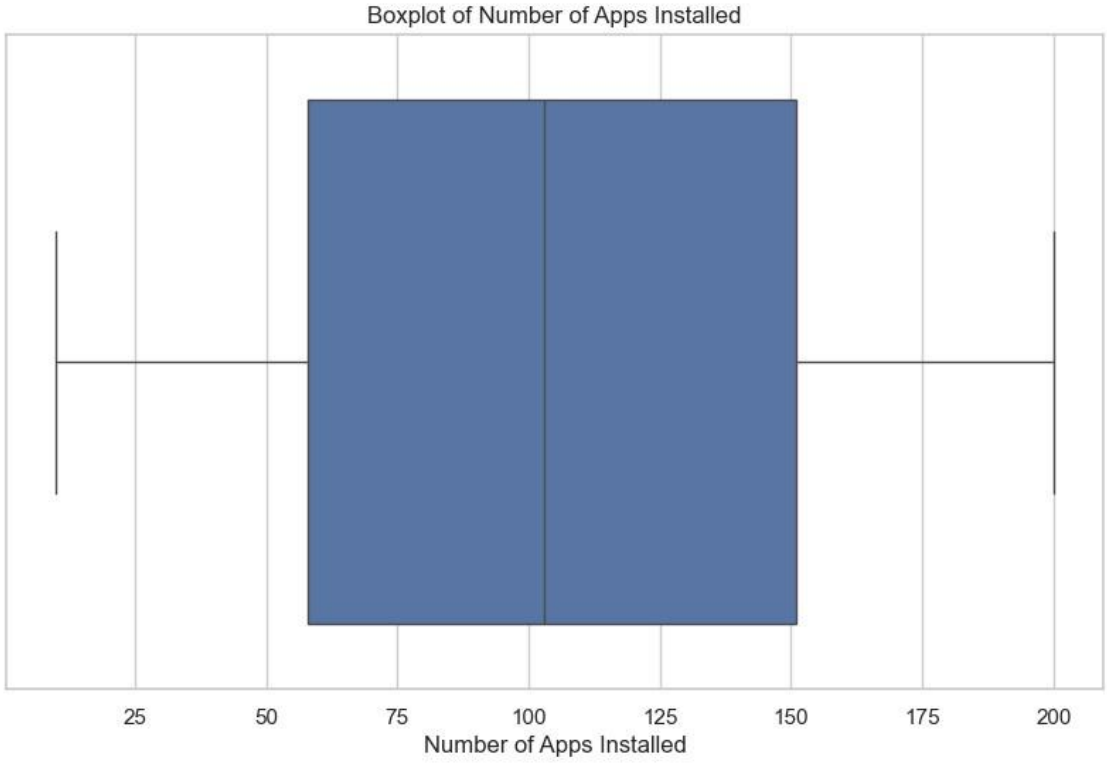
[5]:
```
for col in df.select_dtypes(include=np.number).columns:
    plt.figure()
    sns.boxplot(data=df, x=col)
    plt.title(f"Boxplot of {col}")
    plt.show()
```

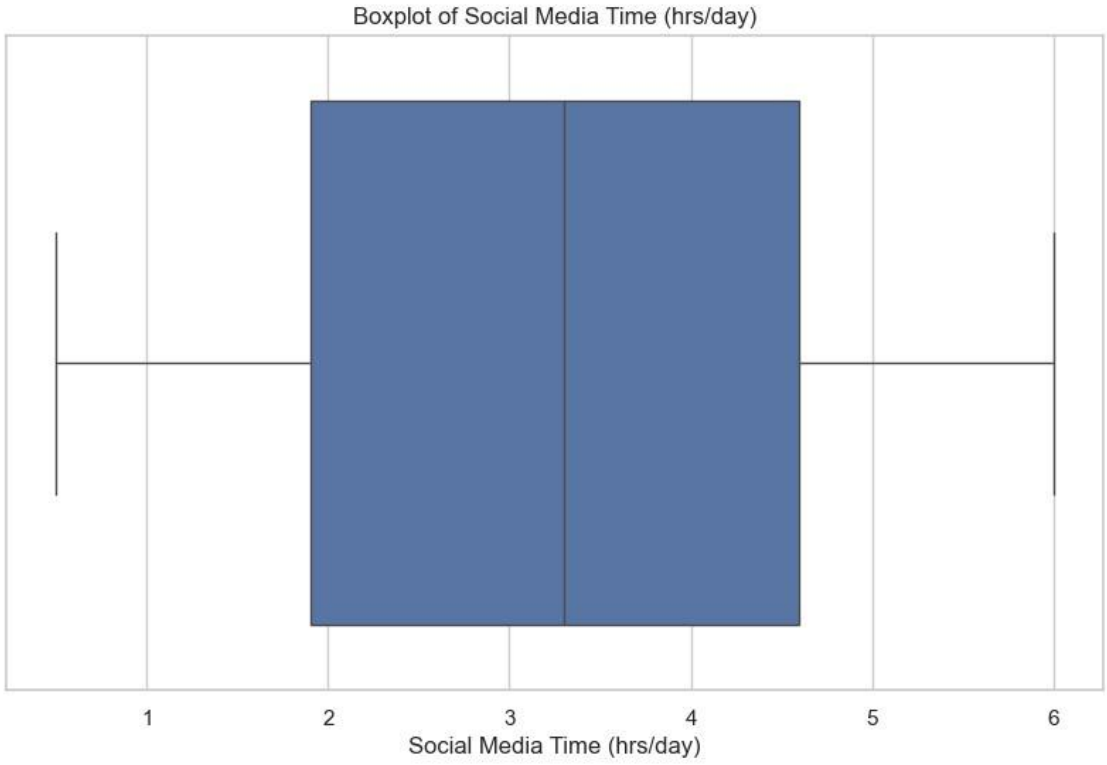


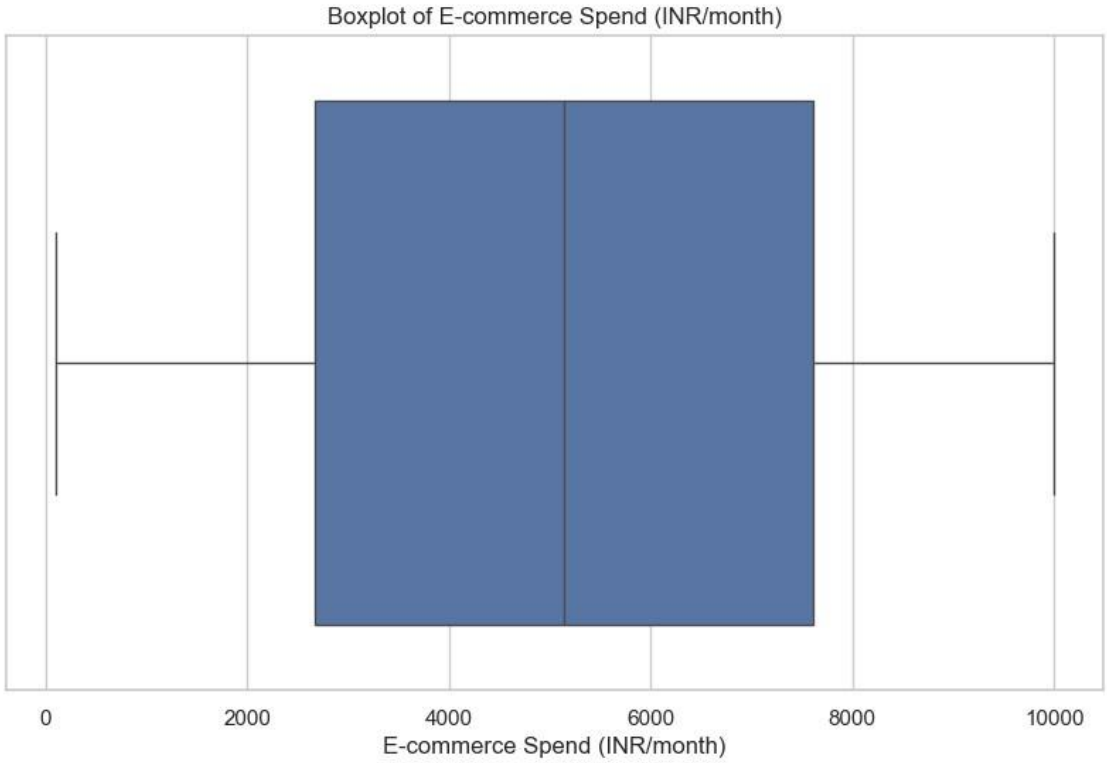

Boxplot of Screen Time (hrs/day)
2
4
6
8
10
12
Screen Time (hrs/day)

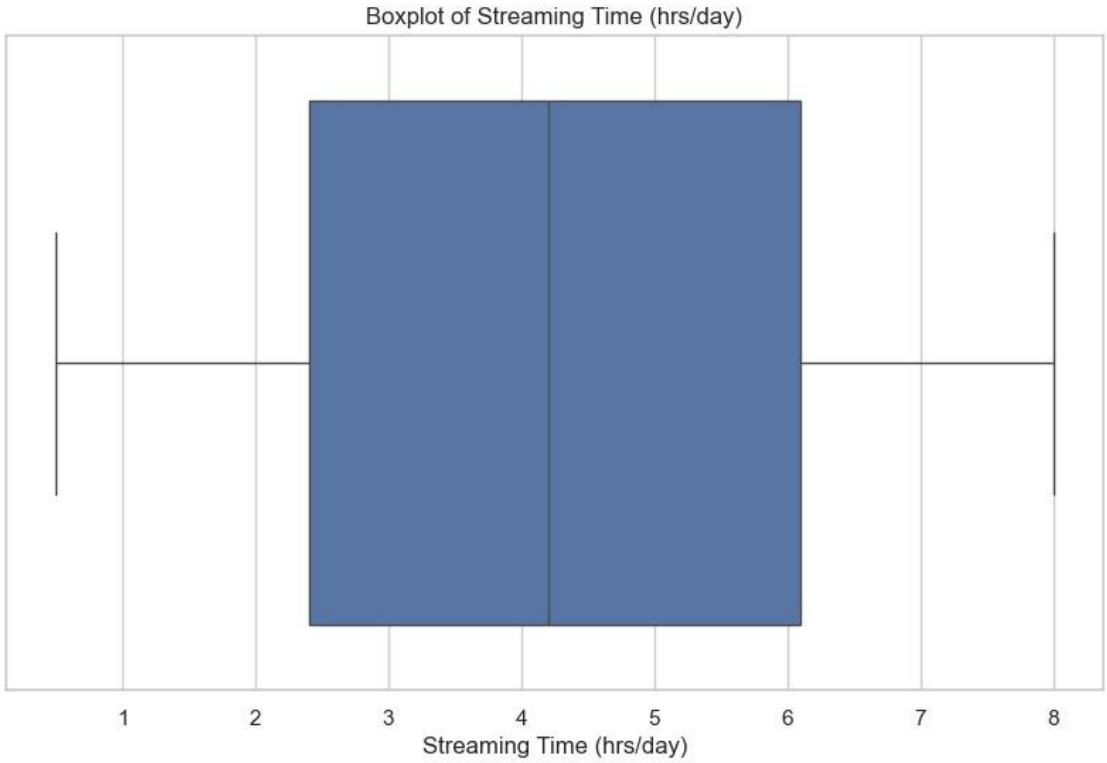

Boxplot of Data Usage (GB/month)
0
10
20
30
40
50
Data Usage (GB/month)

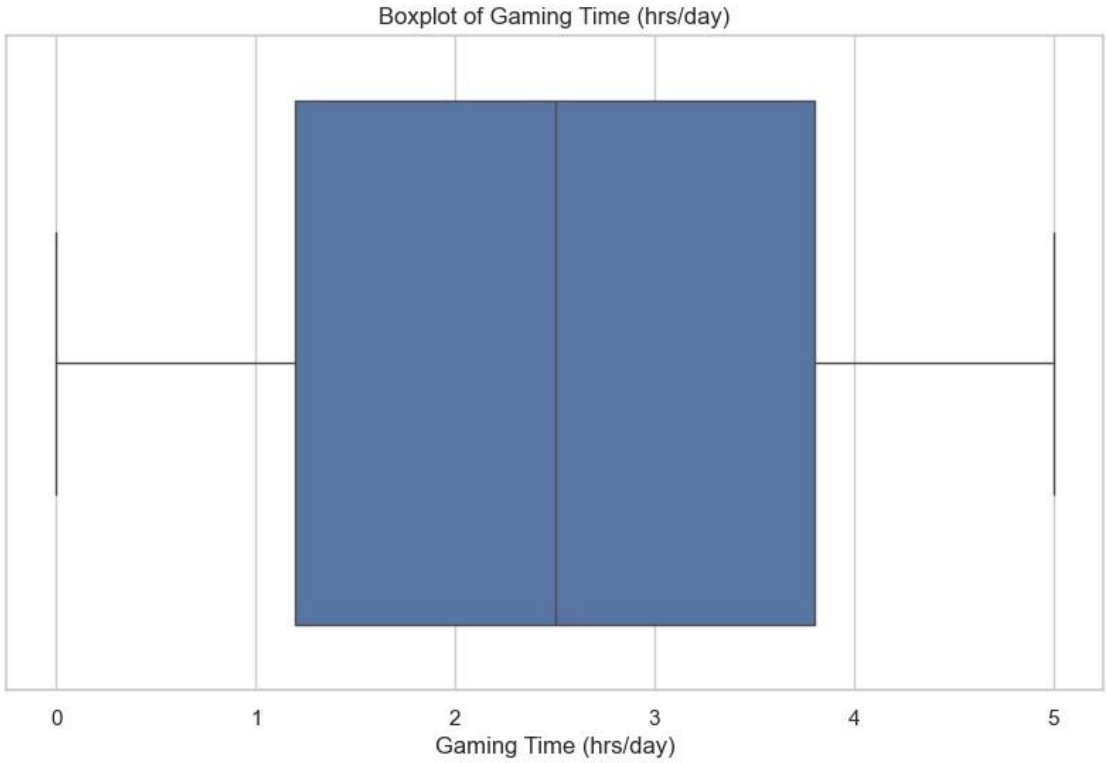

Boxplot of Calls Duration (mins/day)
0
50
100
150
200
250
300
Calls Duration (mins/day)

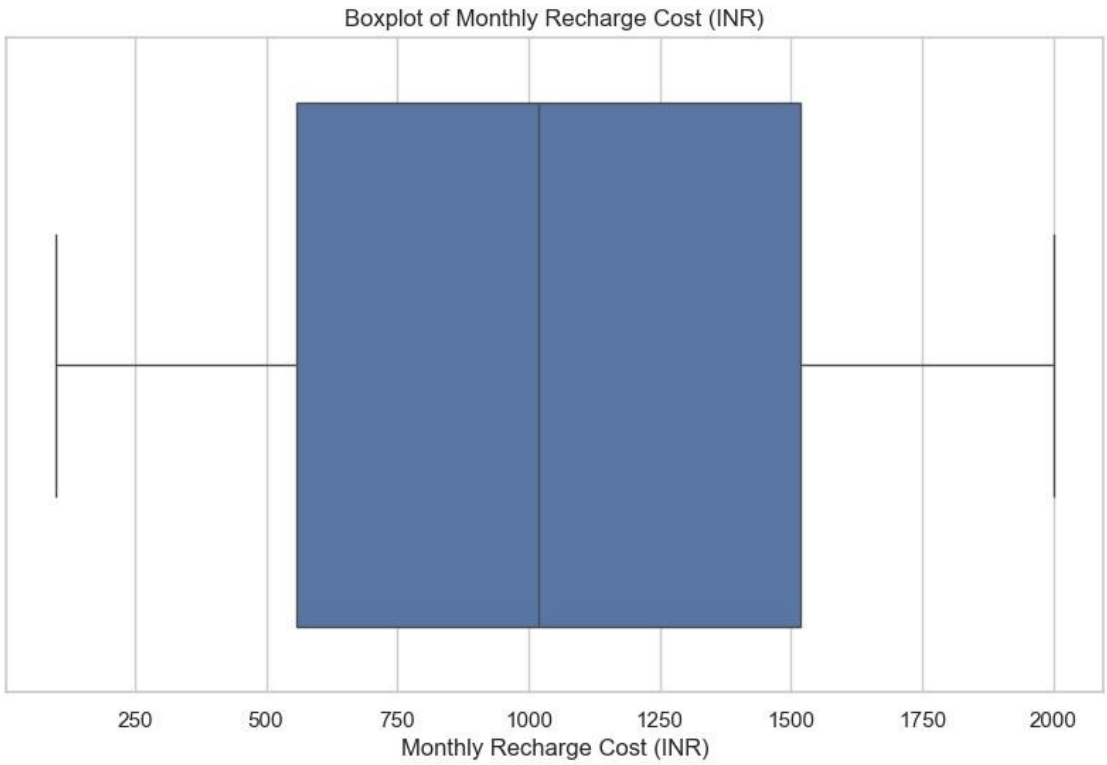

Boxplot of Number of Apps Installed
25
50
75
100
125
150
175
200
Number of Apps Installed


Boxplot of Social Media Time (hrs/day)
1
2
3
4
5
6
Social Media Time (hrs/day)


Boxplot of E-commerce Spend (INR/month)
0
2000
4000
6000
8000
10000
E-commerce Spend (INR/month)


Boxplot of Streaming Time (hrs/day)
1
2
3
4
5
6
7
8
Streaming Time (hrs/day)


Boxplot of Gaming Time (hrs/day)
0
1
2
3
4
5
Gaming Time (hrs/day)



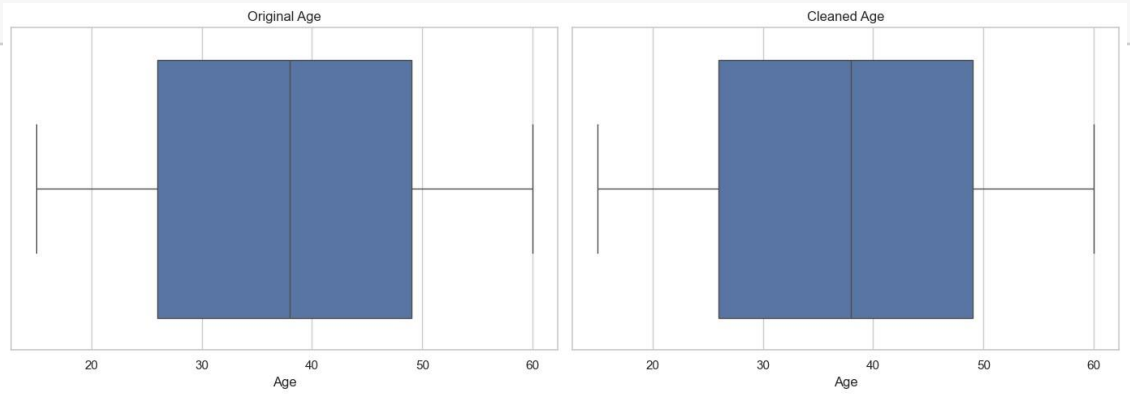
[9]:
```
if 'CategoryCol' in df.columns:
    for col in df.select_dtypes(include=np.number).columns:
        plt.figure()
        sns.boxplot(x='CategoryCol',  =col, data=df)
        plt.title(f"{col} across CategoryCol")
        plt.show()
```

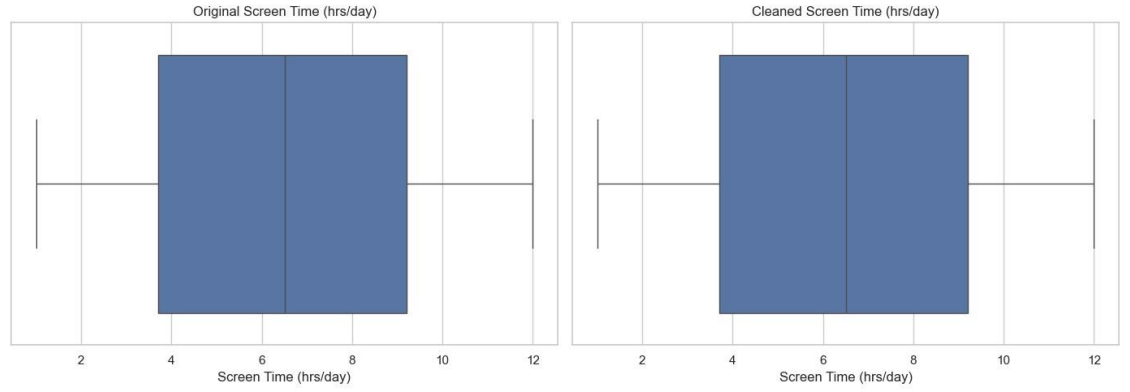
[10]:
```
def remove_outliers_iqr(data, col):
    Q1 = data[col].quantile(0.25)
    Q3 = data[col].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
    outliers = data[(data[col] < lower) | (data[col] > upper)]
    print(f"{col}: {len(outliers)} outliers")
    return data[(data[col] >= lower) & (data[col] <= upper)]

# Apply to all numerical columns
df_cleaned = df.copy()
for col in df.select_dtypes(include=np.number).columns:
    df_cleaned = remove_outliers_iqr(df_cleaned, col)
```

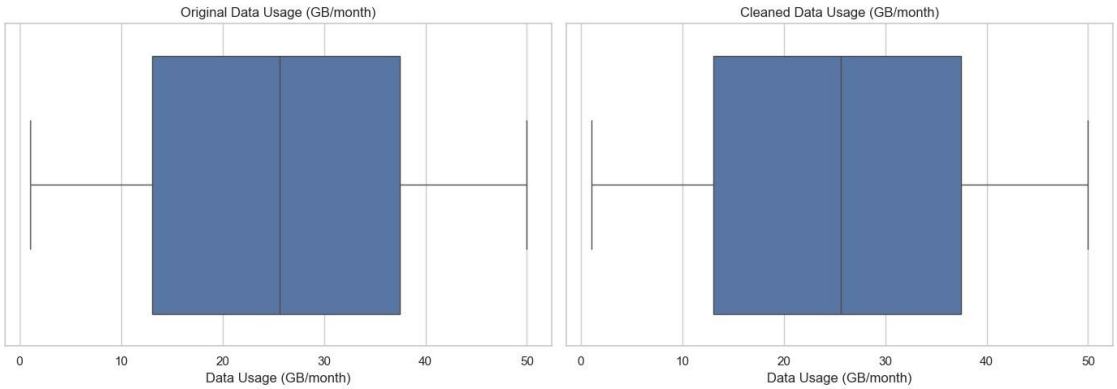
```
Age: 0 outliers
Screen Time (hrs/day): 0 outliers
Data Usage (GB/month): 0 outliers
```

```
Calls Duration (mins/day): 0 outliers
Number of Apps Installed: 0 outliers
Social Media Time (hrs/day): 0 outliers
E-commerce Spend (INR/month): 0 outliers
Streaming Time (hrs/day): 0 outliers
Gaming Time (hrs/day): 0 outliers
Monthly Recharge Cost (INR): 0 outliers
```

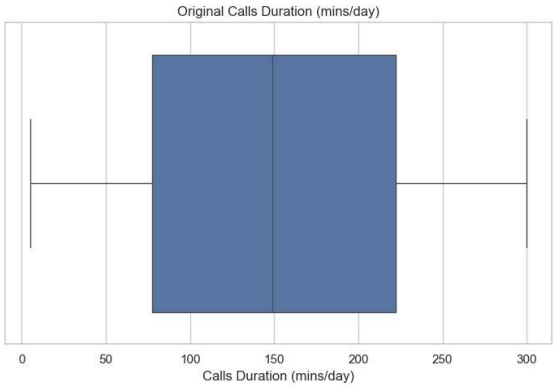
[11]:
```
for col in df.select_dtypes(include=np.number).columns:
    fig, axes = plt.subplots(1, 2,
    figsize=(14, 5)) sns.boxplot(x=df[col],
    ax=axes[0]) axes[0].set_title(f'Original
    {col}') sns.boxplot(x=df_cleaned[col],
    ax=axes[1]) axes[1].set_title(f'Cleaned
    {col}') plt.tight_layout() plt.show()
```

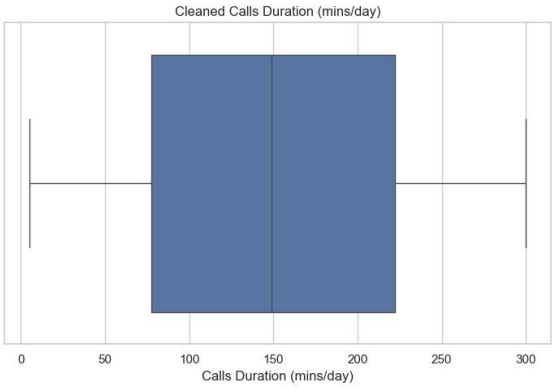


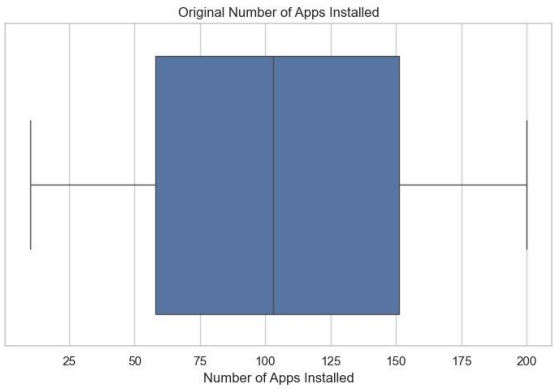


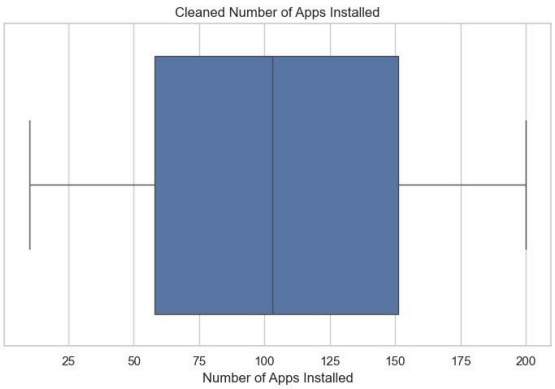

Original Data Usage (GB/month)
0
10
20
30
40
50
Data Usage (GB/month)

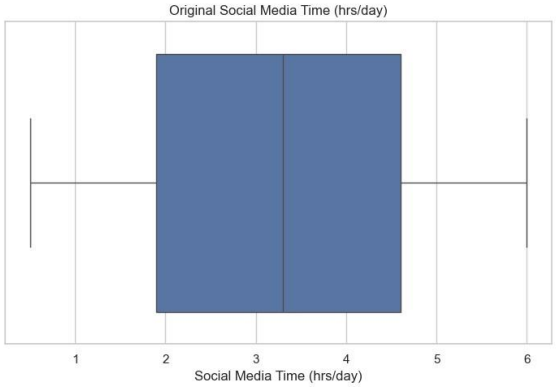
Cleaned Data Usage (GB/month)
0
10
20
30
40
50
Data Usage (GB/month)

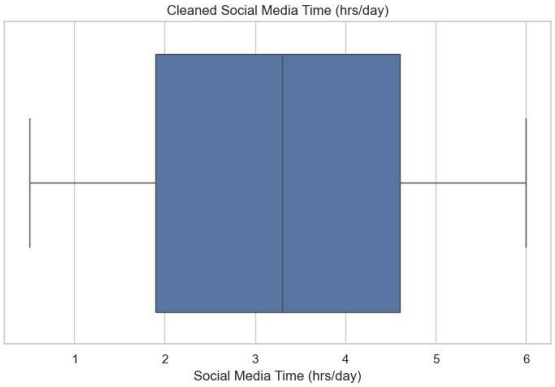

Original Calls Duration (mins/day)
0
50
100
150
200
250
300
Calls Duration (mins/day)

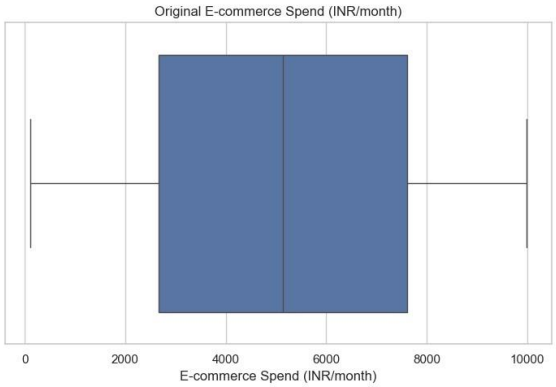

Cleaned Calls Duration (mins/day)
0
50
100
150
200
250
300
Calls Duration (mins/day)

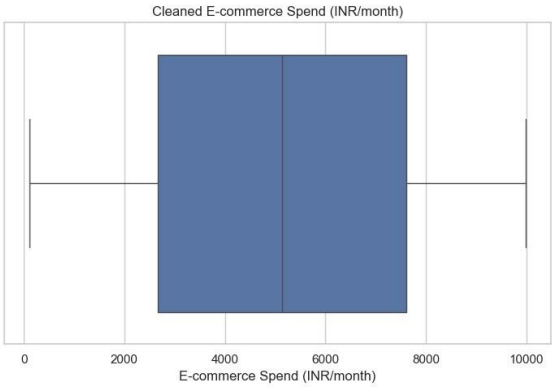

Original Number of Apps Installed
25
50
75
100
125
150
175
200
Number of Apps Installed

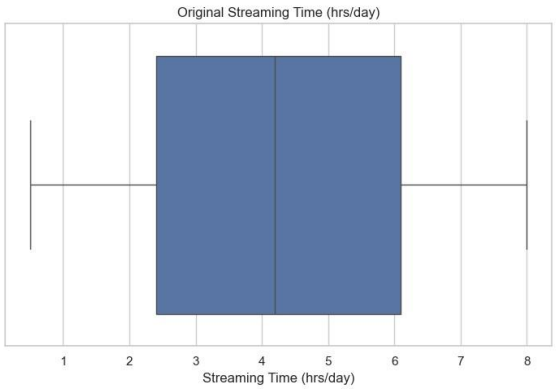

Cleaned Number of Apps Installed
25
50
75
100
125
150
175
200
Number of Apps Installed

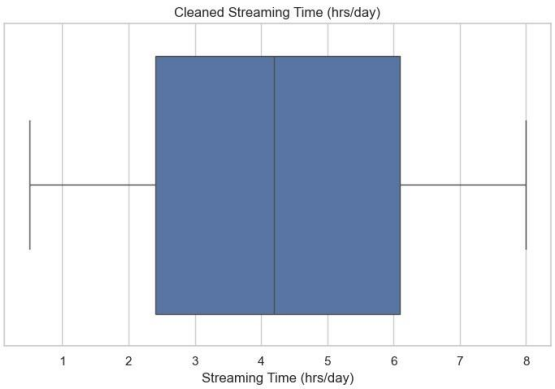

Original Social Media Time (hrs/day)
1
2
3
4
5
6
Social Media Time (hrs/day)

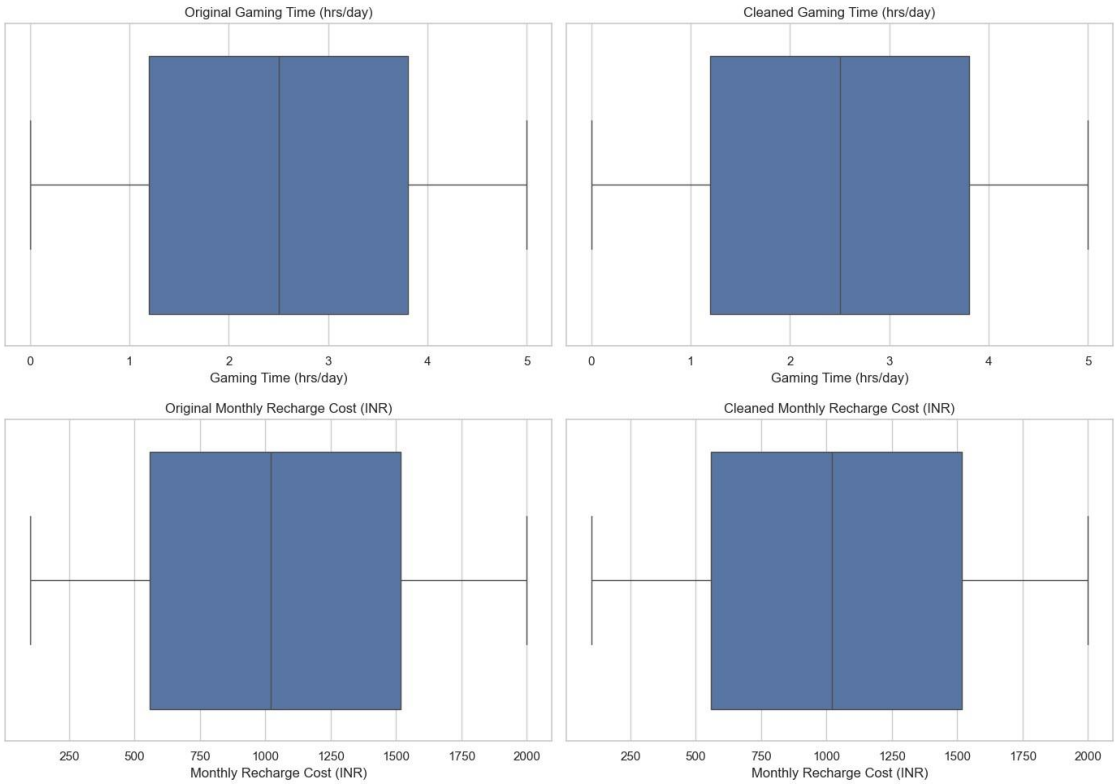

Cleaned Social Media Time (hrs/day)
1
2
3
4
5
6
Social Media Time (hrs/day)


Original E-commerce Spend (INR/month)
0
2000
4000
6000
8000
10000
E-commerce Spend (INR/month)

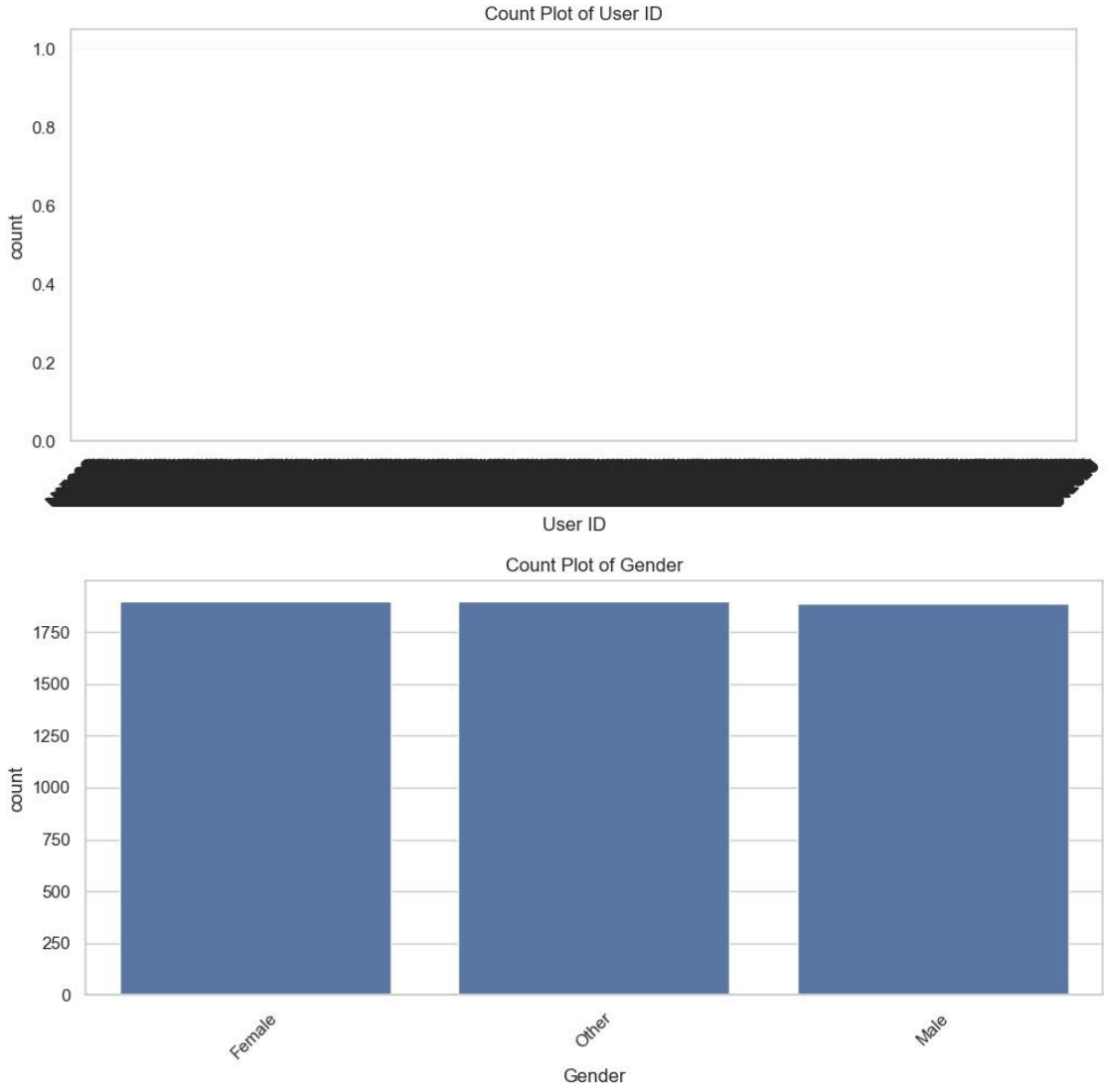

Cleaned E-commerce Spend (INR/month)
0
2000
4000
6000
8000
10000
E-commerce Spend (INR/month)

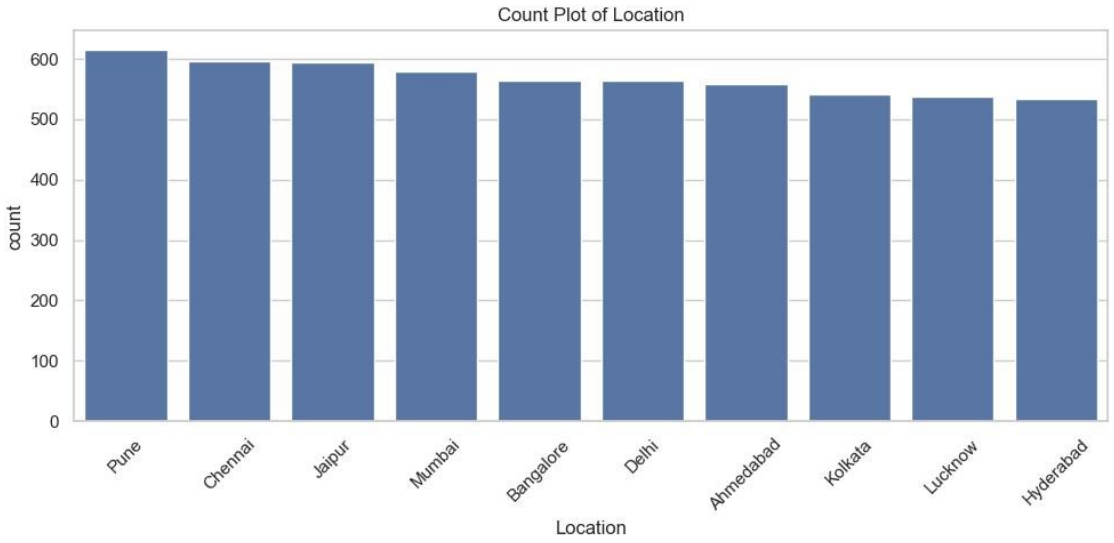

Original Streaming Time (hrs/day)
1
2
3
4
5
6
7
8
Streaming Time (hrs/day)

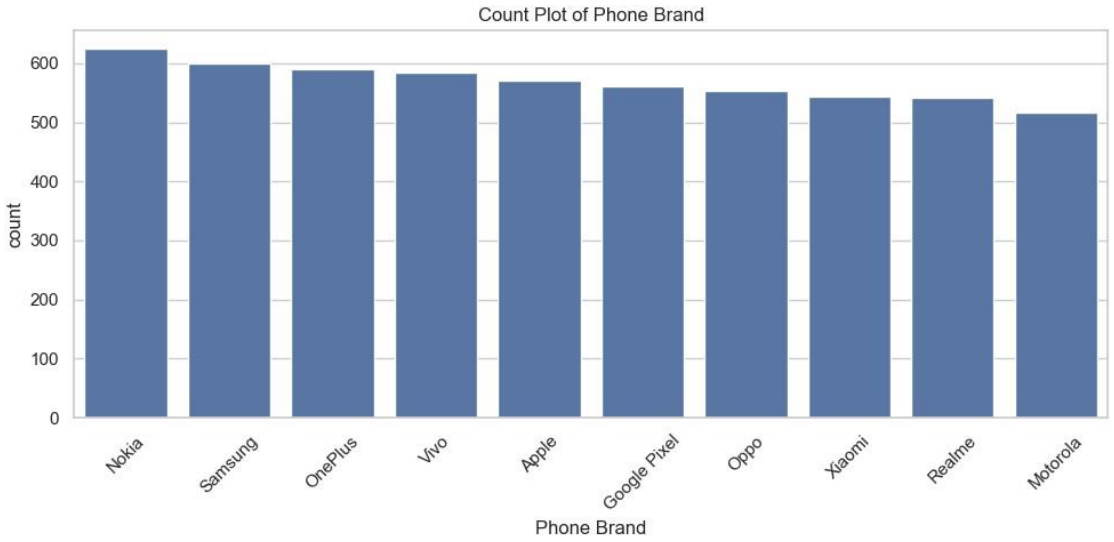

Cleaned Streaming Time (hrs/day)
1
2
3
4
5
6
7
8
Streaming Time (hrs/day)

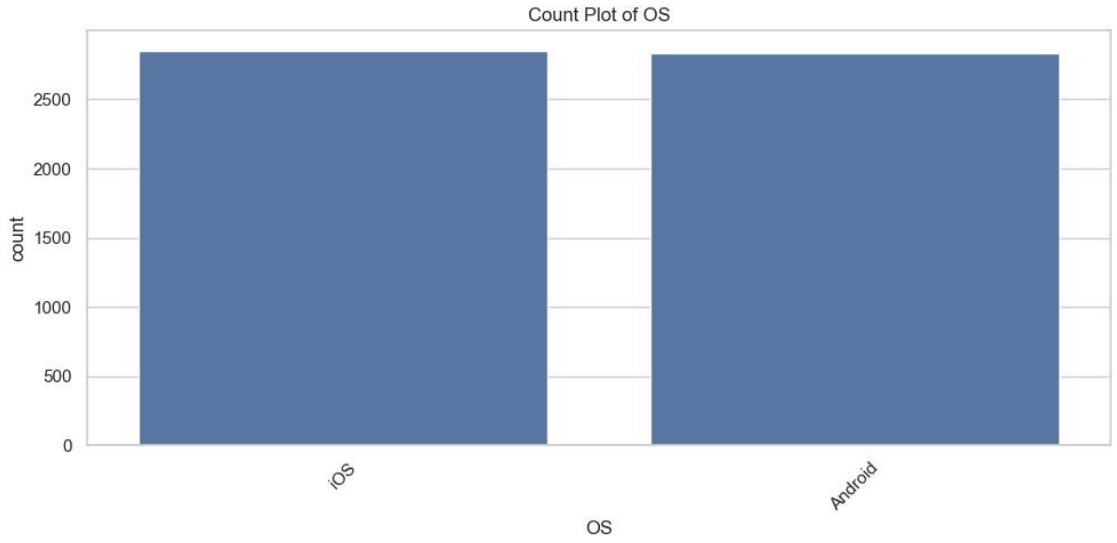

Original Gaming Time (hrs/day)
0
1
2
3
4
5
Gaming Time (hrs/day)
Cleaned Gaming Time (hrs/day)
0
1
2
3
4
5
Gaming Time (hrs/day)
Original Monthly Recharge Cost (INR)
250
500
750
1000
1250
1500
1750
2000
Monthly Recharge Cost (INR)
Cleaned Monthly Recharge Cost (INR)
250
500
750
1000
1250
1500
1750
2000
Monthly Recharge Cost (INR)

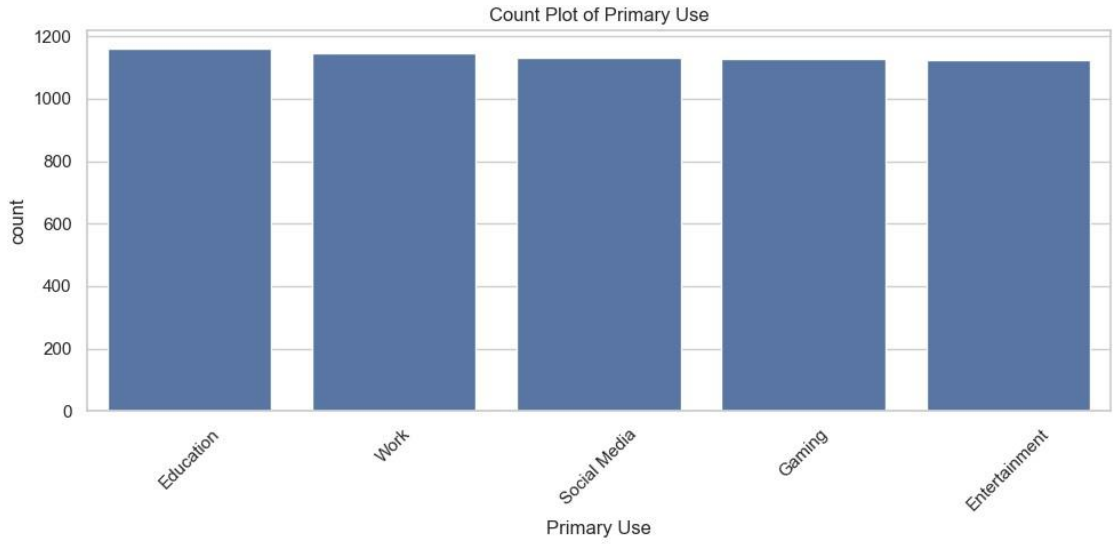
[13]:
```
cat_cols = df.select_dtypes(include='object').columns

for col in cat_cols:
    plt.figure(figsize=(10, 5))
    sns.countplot(data=df, x=col, order=df[col].value_counts().index)
    plt.title(f"Count Plot of {col}")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```



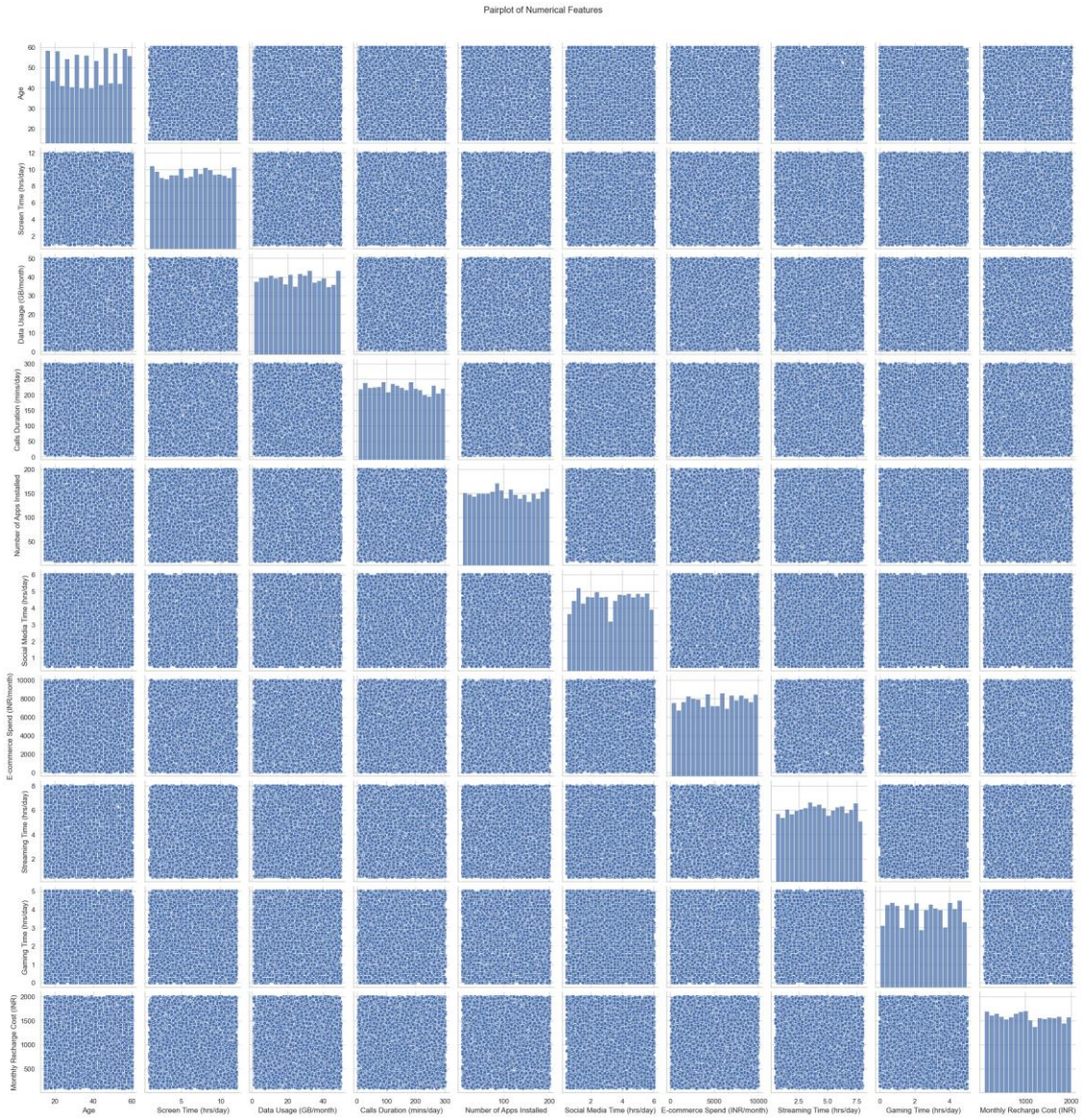

Count Plot of Location
count
0
100
200
300
400
500
600
Pune
Chennai
Jaipur
Mumbai
Bangalore
Delhi
Ahmedabad
Kolkata
Lucknow
Hyderabad
Location


Count Plot of Phone Brand
count
0
100
200
300
400
500
600
Nokia
Samsung
OnePlus
Vivo
Apple
Google Pixel
Oppo
Xiaomi
Realme
Motorola
Phone Brand

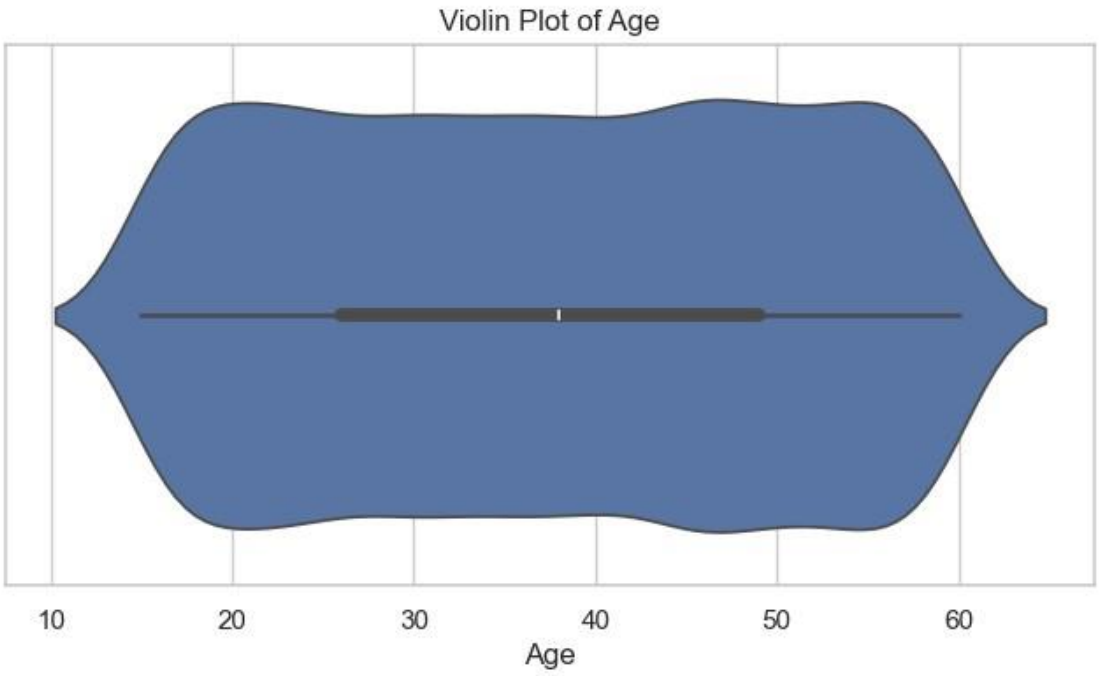




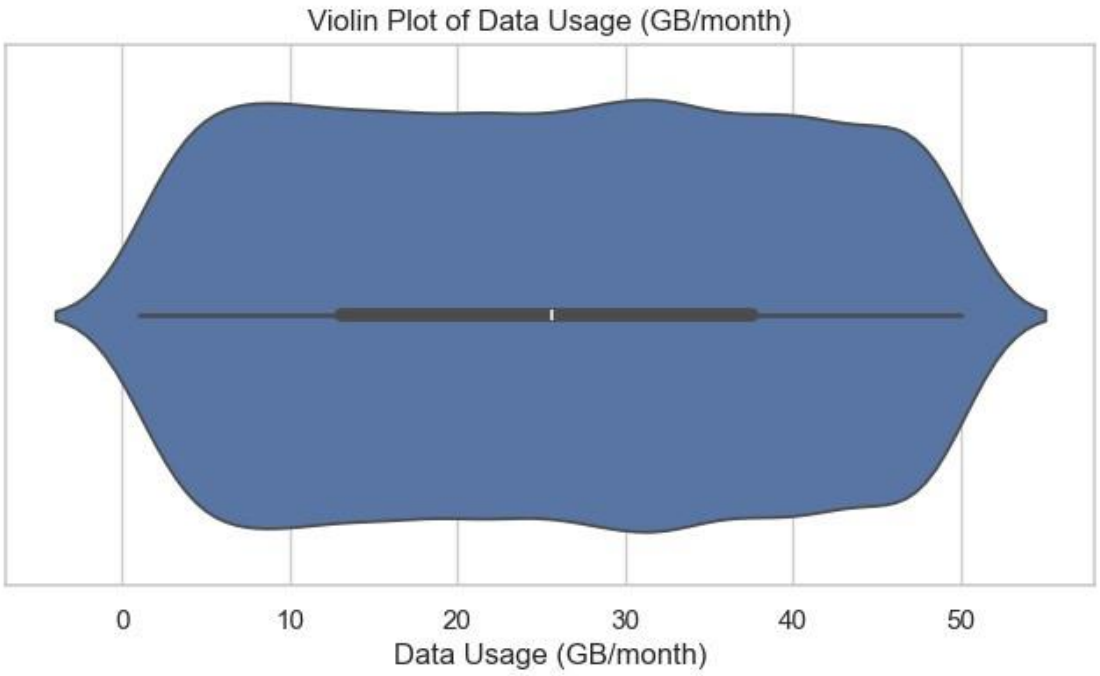
[14]:
```
sns.pairplot(df.select_dtypes(include=np.numbe
r))    plt.suptitle("Pairplot    of    Numerical
Features", y=1.02) plt.show()
```

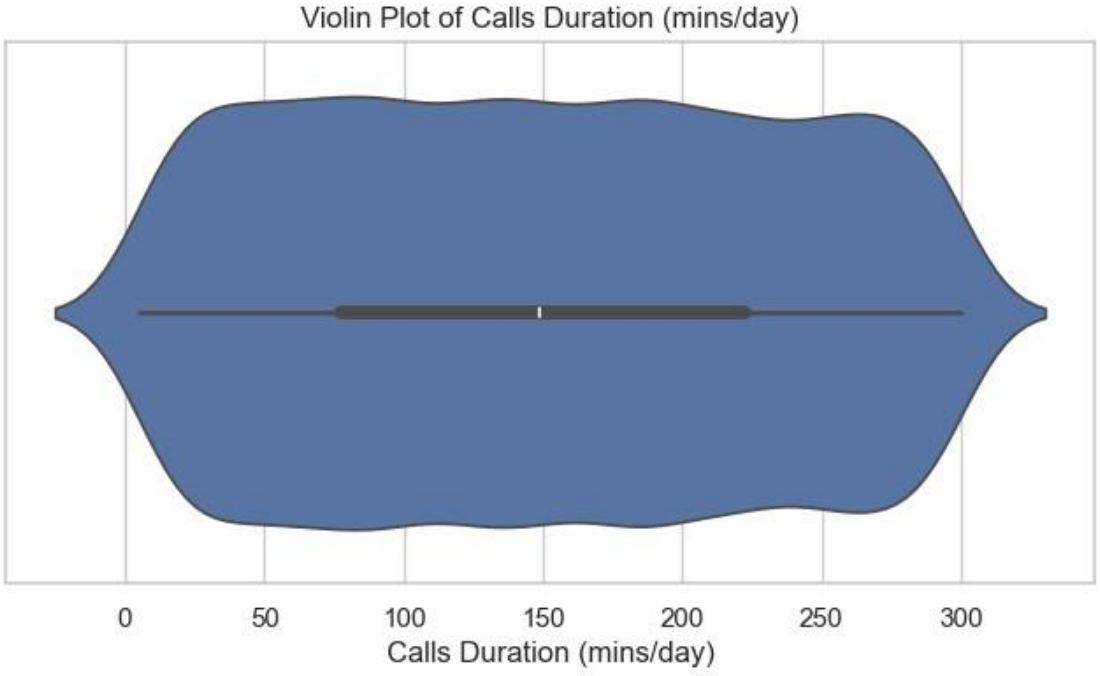


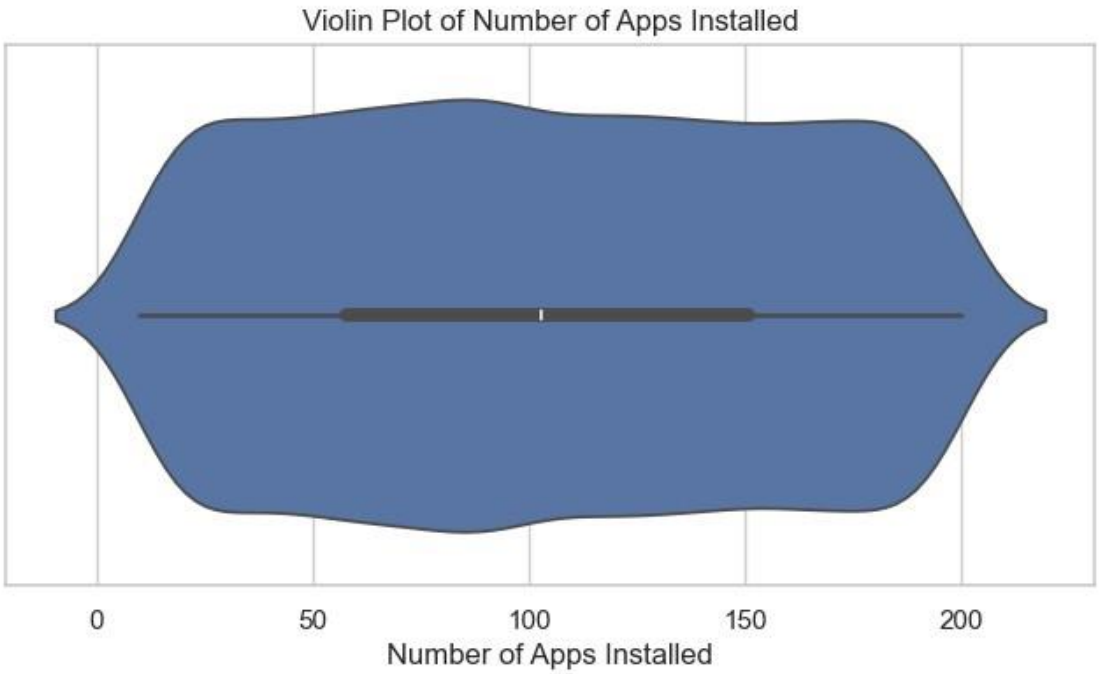
[15]:
```
for col in df.select_dtypes(include=np.number).columns:
    plt.figure(figsize=(8, 4))
    sns.violinplot(x=df[col])
    plt.title(f"Violin Plot of {col}")
    plt.show()
```

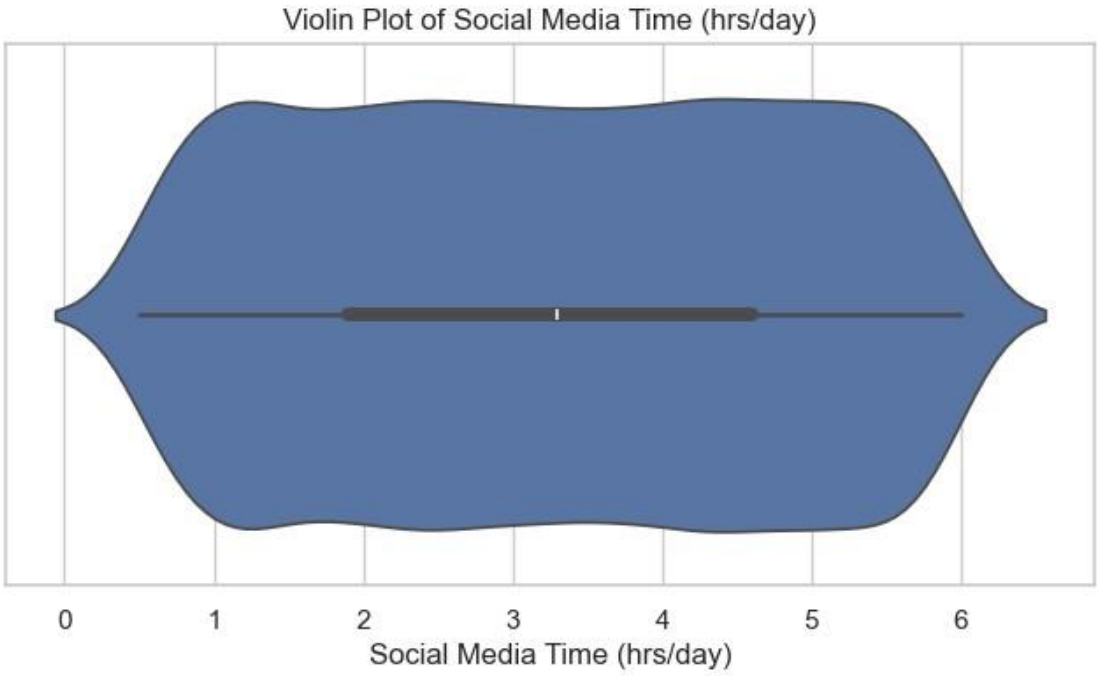

Violin Plot of Age
10
20
30
40
50
60
Age

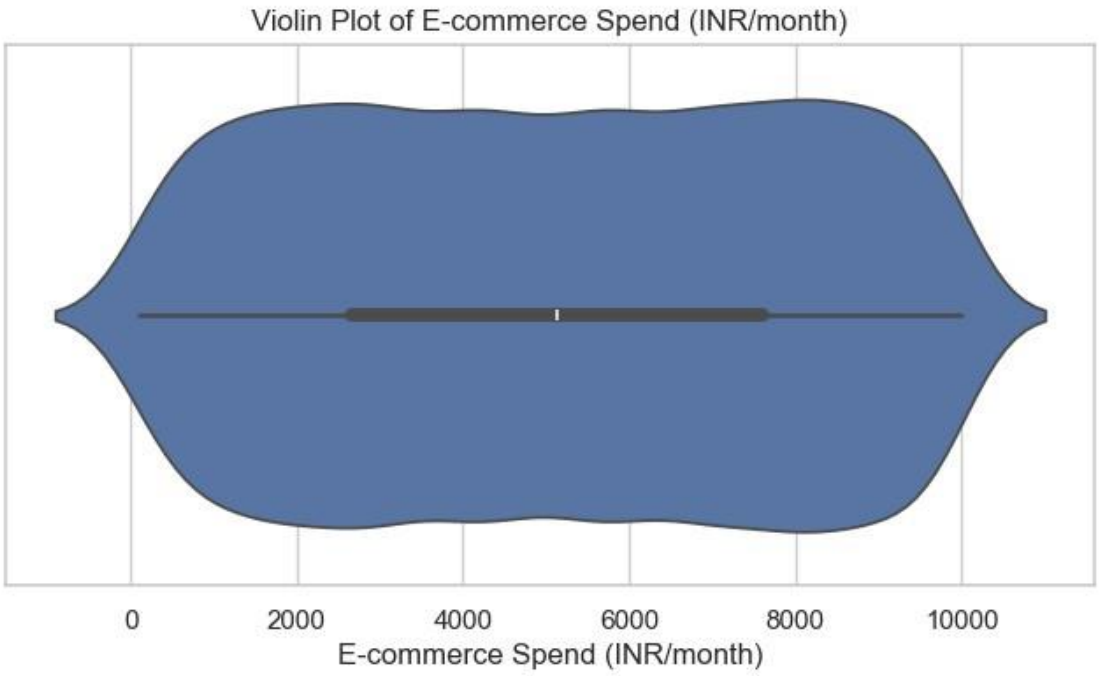

Violin Plot of Screen Time (hrs/day)
0
2
4
6
8
10
12
Screen Time (hrs/day)

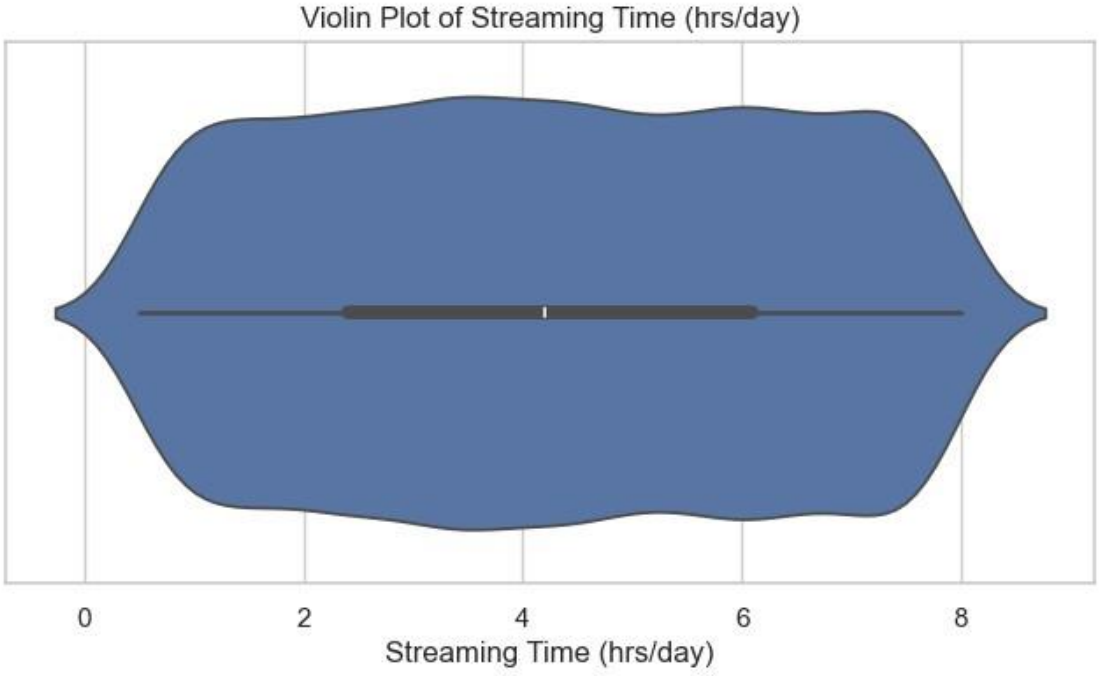

Violin Plot of Data Usage (GB/month)
0
10
20
30
40
50
Data Usage (GB/month)

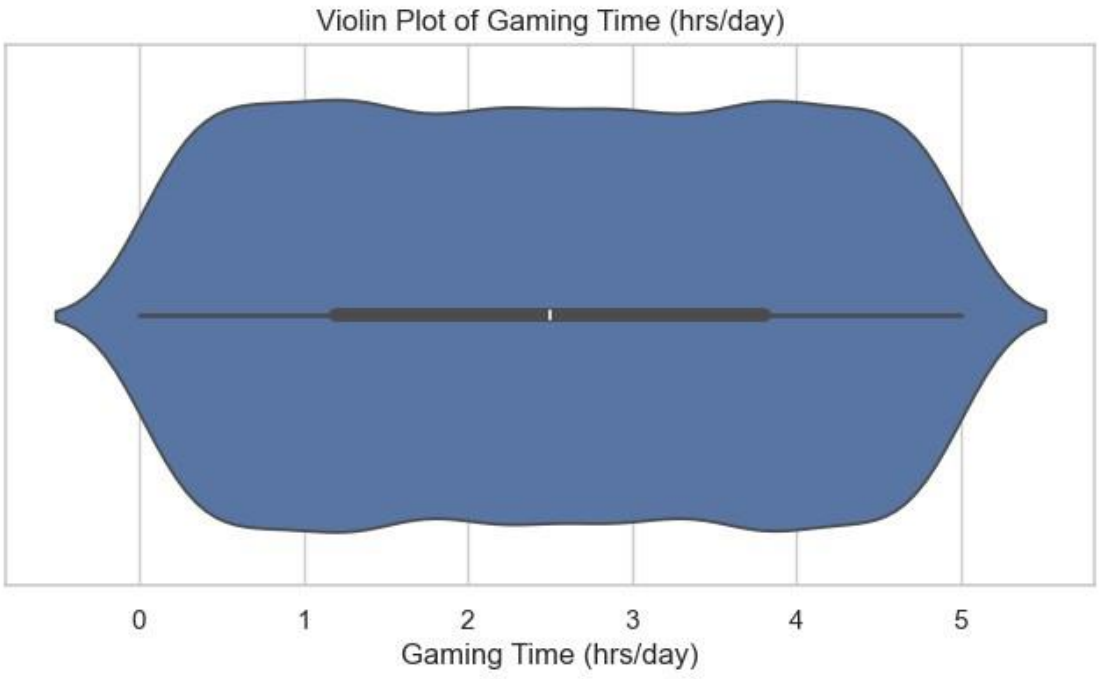

Violin Plot of Calls Duration (mins/day)
0
50
100
150
200
250
300
Calls Duration (mins/day)

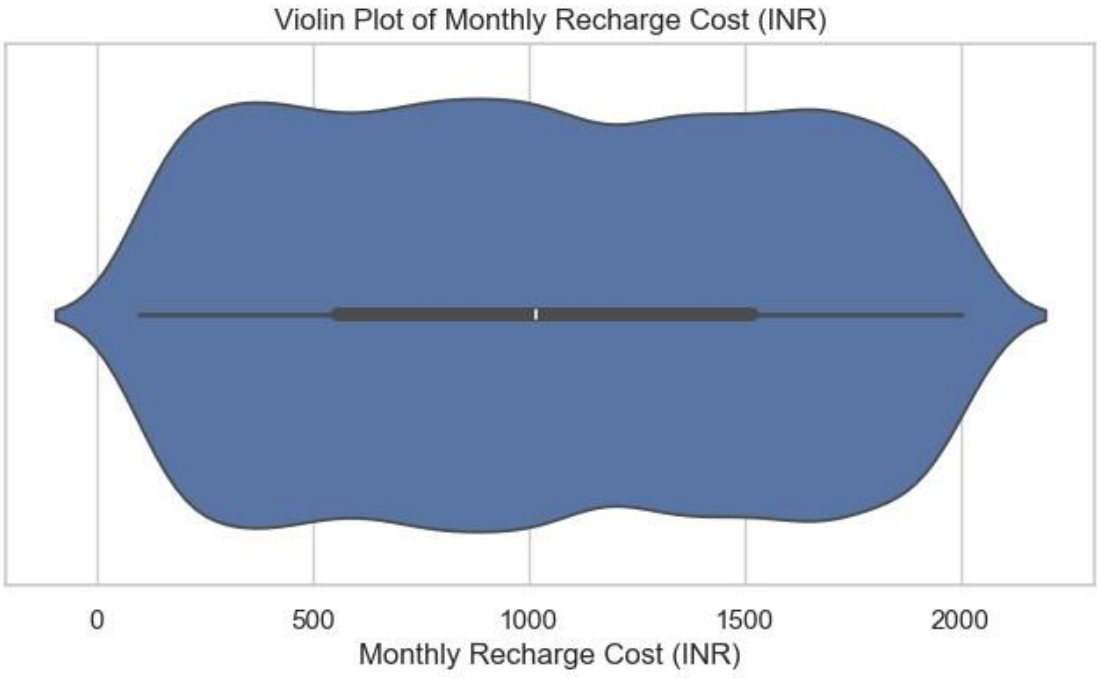

Violin Plot of Number of Apps Installed
0
50
100
150
200
Number of Apps Installed


Violin Plot of Social Media Time (hrs/day)
0
1
2
3
4
5
6
Social Media Time (hrs/day)


Violin Plot of E-commerce Spend (INR/month)
0
2000
4000
6000
8000
10000
E-commerce Spend (INR/month)


Violin Plot of Streaming Time (hrs/day)
0
2
4
6
8
Streaming Time (hrs/day)





[18]:
```
# Replace 'CategoryCol' with actual categorical
column if 'CategoryCol' in df.columns:
```

```
    for col in df.select_dtypes(include=np.number).columns:
        plt.figure(figsize=(10, 5))
        sns.swarmplot(x='CategoryCol',  =col, data=df)
        plt.title(f"{col} across CategoryCol")
        plt.xticks(rotation=45)
        plt.tight_layout()
        plt.show()
```

[ ]: