Netflix 2025:User Behavior Dataset Analysis

Dataset Source: https://www.kaggle.com/datasets/sayeeduddin/netflix-2025user-behavior-dataset-210k-records

In [ ]:
```
Name: M.Akshaya
Rollno: 2211CS010649
Section: S1 - 87
```

In [ ]:
```
1. Dataset Overview
    File Name: users.csv
    Rows: 10,300
    Columns: 14
    Category: Demographics + Subscriptions
    Purpose: Analyze user demographics, subscription patterns, spending behavior, and device usage.

2. Dataset Columns and Data Types
    i. first_name – a string column containing the first name of the user.
    ii. last_name – a string column containing the last name of the user.
    iii. age – an integer column representing the age of the user.
    iv. gender – a string column indicating the gender of the user, typically "Male" or "Female."
    v. country – a string column representing the country of the user.
    vi. state_province – a string column representing the state or province where the user resides.
    vii. city – a string column representing the city of the user.
    viii. subscription_plan – a string column indicating the subscription type, such as Basic, Standard, or Premium+.
    ix. subscription_start_date – a date column showing when the subscription started.
    x. is_active – a boolean column indicating whether the user's subscription is currently active (TRUE or FALSE).
    xi. monthly_spend – a float column representing the user's monthly spending in the subscription.
    xii. primary_device – a string column indicating the primary device used by the user, such as Laptop or Desktop.
    xiii. household_size – an integer column representing the number of people in the user's household.
    xiv. created_at – a string column containing the account creation timestamp.

3. Data Quality Notes
    Missing Values: Present in some columns (e.g., age, monthly_spend, gender).
    Duplicates: Some duplicate records may exist—need removal for accurate analysis.
    Outliers: Likely in age (very low or high values) and monthly_spend (extremely high/low spenders).

4. Categorical Column Distributions (Example)
    i. gender – The main categories are Male and Female.
    ii. subscription_plan – The primary subscription types are Basic, Standard, and Premium+.
```

```
    iii. primary_device - The most common devices used are Laptop and Desktop.
    iv. country - The top countries with the highest number of users include USA, UK, Canada, and India.
```

In [15]:
```
# --------- 1. IMPORT LIBRARIES ---------
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# --------- 2. LOAD DATA ---------
file_path = r"C:\Users\Harini\Downloads\archive\users.csv"
df = pd.read_csv(file_path)

# --------- 3. BASIC INFO ---------
print("Dataset Shape:", df.shape)
print("\nDataset Preview:\n", df.head())
print("\nDataset Info:\n")
print(df.info())
print("\nMissing Values:\n", df.isnull().sum())
```

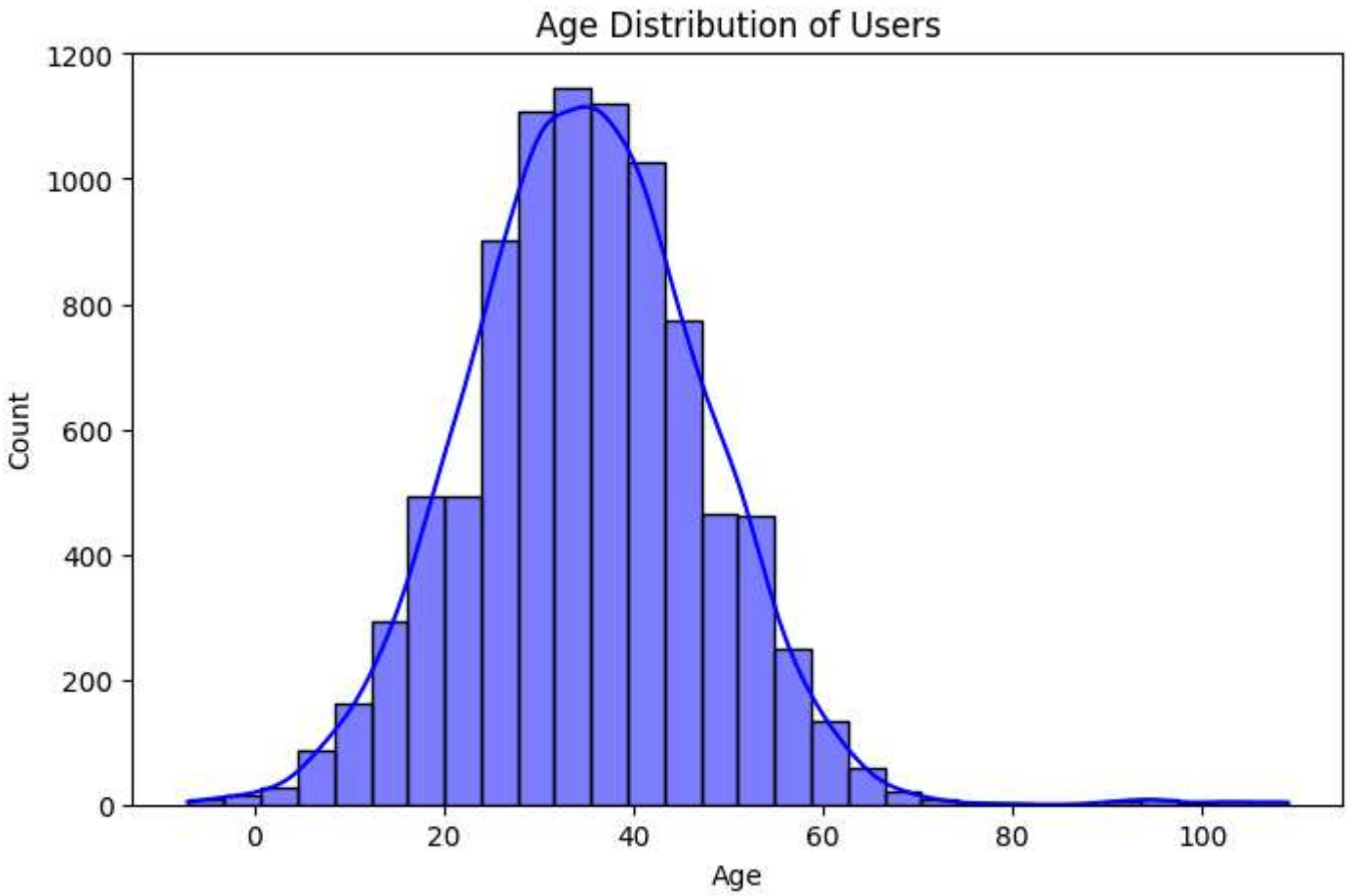
```
Dataset Shape: (10300, 16)

Dataset Preview:
       user_id                      email first_name last_name   age  gender  \
0  user_00001   figueroajohn@example.org      Erica     Garza  43.0    Male
1  user_00002      blakeerik@example.com     Joshua   Bernard  38.0    Male
2  user_00003        smiller@example.net    Barbara  Williams  32.0  Female
3  user_00004  mitchellclark@example.com    Chelsea  Ferguson  11.0    Male
4  user_00005      richard13@example.net      Jason    Foster  21.0  Female

  country state_province                city subscription_plan  \
0     USA  Massachusetts  North Jefferyhaven             Basic
1     USA          Texas      North Noahstad          Premium+
2     USA       Michigan          Traciebury          Standard
3     USA           Ohio          South Noah          Standard
4     USA        Arizona         West Donald          Standard

  subscription_start_date  is_active  monthly_spend primary_device  \
0              2024-04-08       True          36.06         Laptop
1              2024-05-24       True          14.59        Desktop
2              2023-09-22      False          11.71        Desktop
3              2024-08-21       True          28.56         Laptop
4              2024-10-28       True           9.54        Desktop

   household_size                  created_at
0             1.0  2023-04-01 14:40:50.540242
1             2.0  2024-10-10 15:39:11.030515
2             3.0  2024-06-29 14:27:49.560875
3             2.0  2023-04-11 01:01:59.614841
4             6.0  2025-04-12 19:59:30.137806

Dataset Info:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10300 entries, 0 to 10299
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   user_id                 10300 non-null  object
 1   email                   10300 non-null  object
 2   first_name              10300 non-null  object
 3   last_name               10300 non-null  object
```

```
 4   age                      9071 non-null   float64
 5   gender                   9476 non-null   object
 6   country                  10300 non-null  object
 7   state_province           10300 non-null  object
 8   city                     10300 non-null  object
 9   subscription_plan        10300 non-null  object
 10  subscription_start_date  10300 non-null  object
 11  is_active                10300 non-null  bool
 12  monthly_spend            9283 non-null   float64
 13  primary_device           10300 non-null  object
 14  household_size           8755 non-null   float64
 15  created_at               10300 non-null  object
dtypes: bool(1), float64(3), object(12)
memory usage: 1.2+ MB
None

Missing Values:
 user_id                      0
email                         0
first_name                    0
last_name                     0
age                        1229
gender                      824
country                       0
state_province                0
city                          0
subscription_plan             0
subscription_start_date       0
is_active                     0
monthly_spend              1017
primary_device                0
household_size             1545
created_at                    0
dtype: int64
```

In [3]:

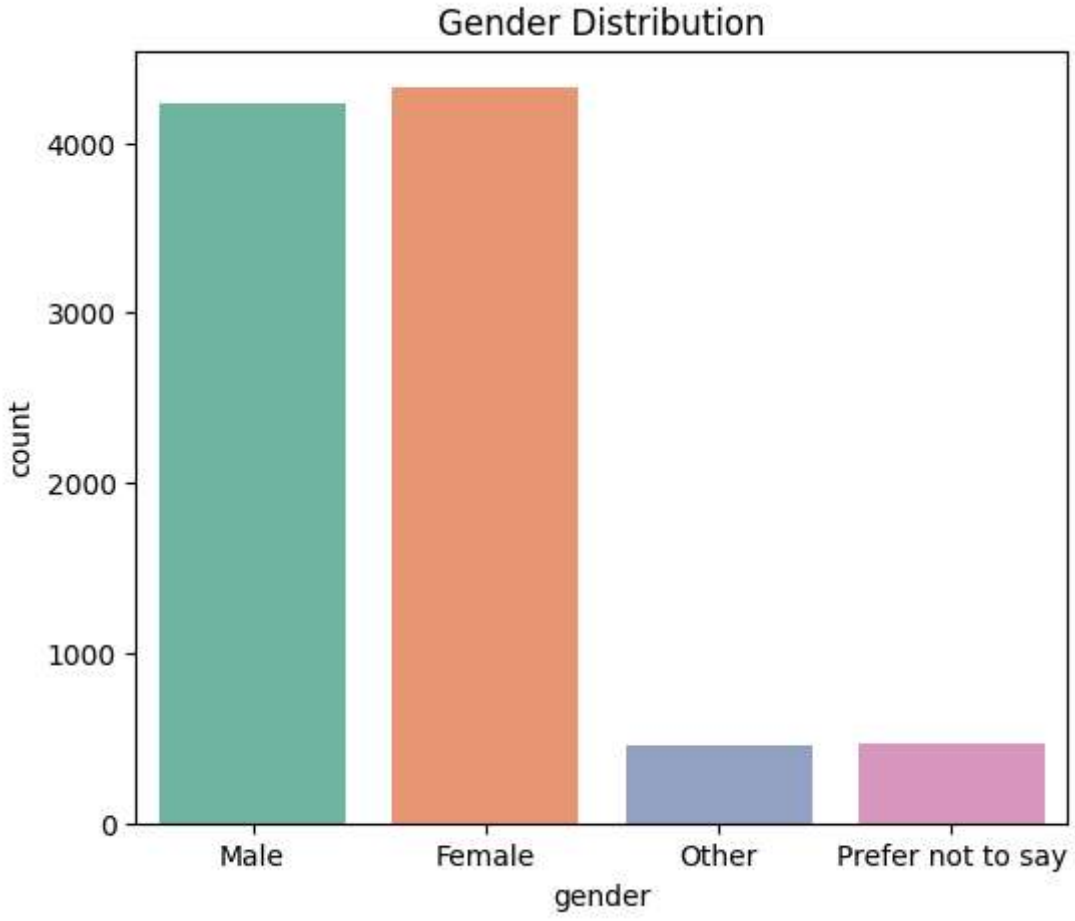
```
# --------- 4. VISUALIZATIONS ---------

# ---- Age Distribution ----
plt.figure(figsize=(8,5))
sns.histplot(df['age'], bins=30, kde=True, color='blue')
plt.title("Age Distribution of Users")
plt.xlabel("Age")
```

```
plt.ylabel("Count")
plt.show()
```

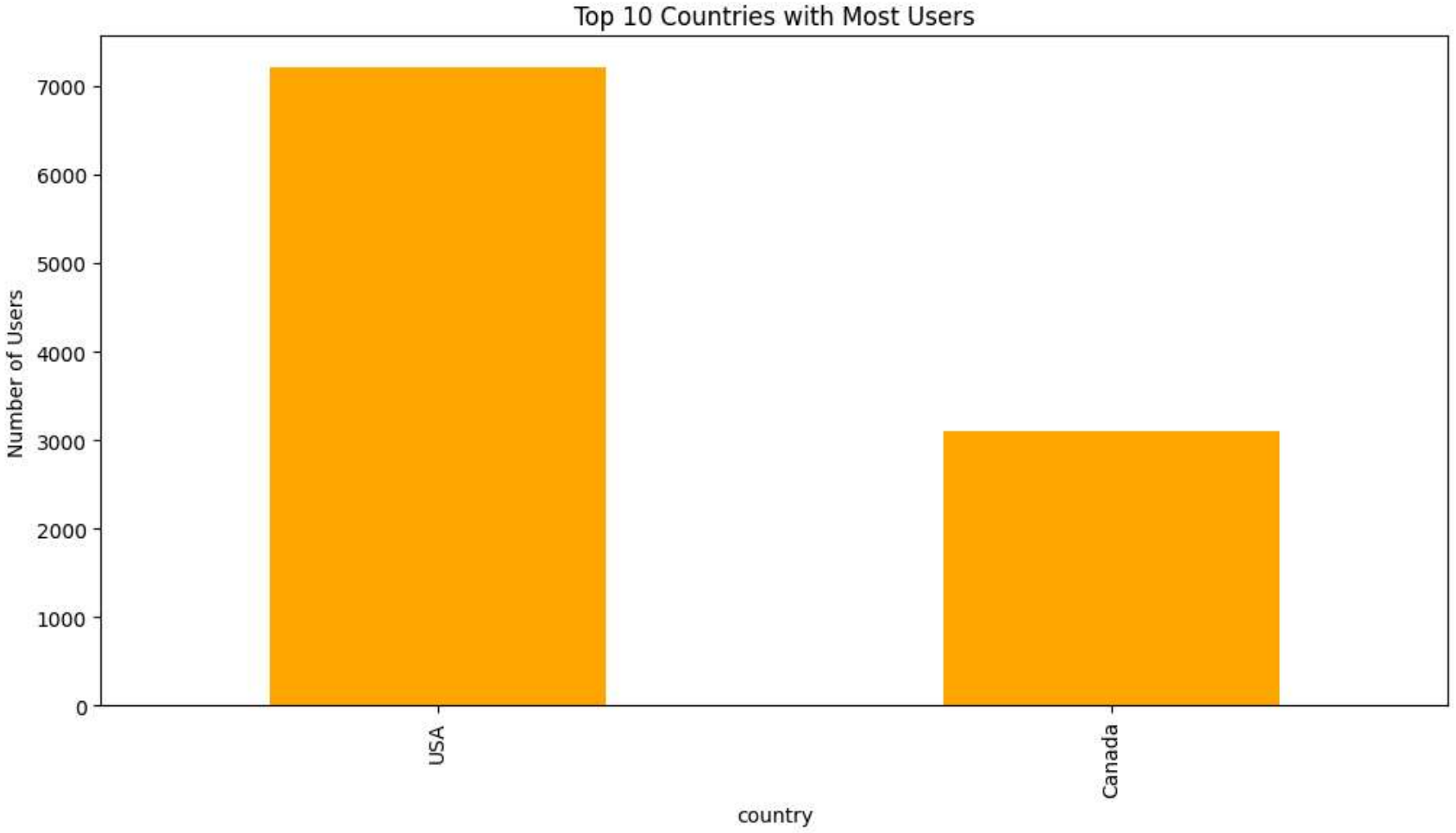


In [4]:
```
# ---- Gender Distribution ----
plt.figure(figsize=(6,5))
sns.countplot(x='gender', data=df, palette='Set2')
plt.title("Gender Distribution")
plt.show()
```

```
C:\Users\Harini\AppData\Local\Temp\ipykernel_12776\3767096763.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `h
ue` and set `legend=False` for the same effect.

  sns.countplot(x='gender', data=df, palette='Set2')
```

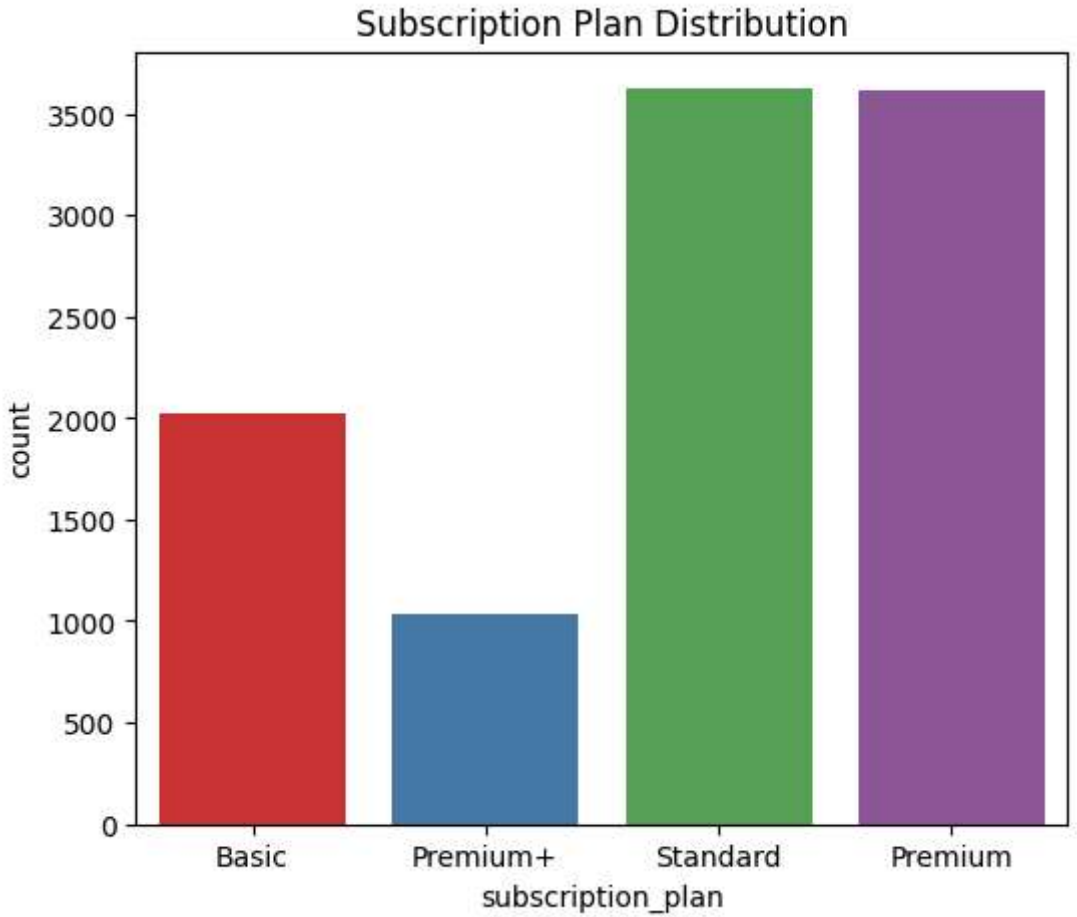


In [5]:
```
# ---- Country vs Users ----
plt.figure(figsize=(12,6))
df['country'].value_counts().head(10).plot(kind='bar', color='orange')
plt.title("Top 10 Countries with Most Users")
plt.ylabel("Number of Users")
plt.show()
```

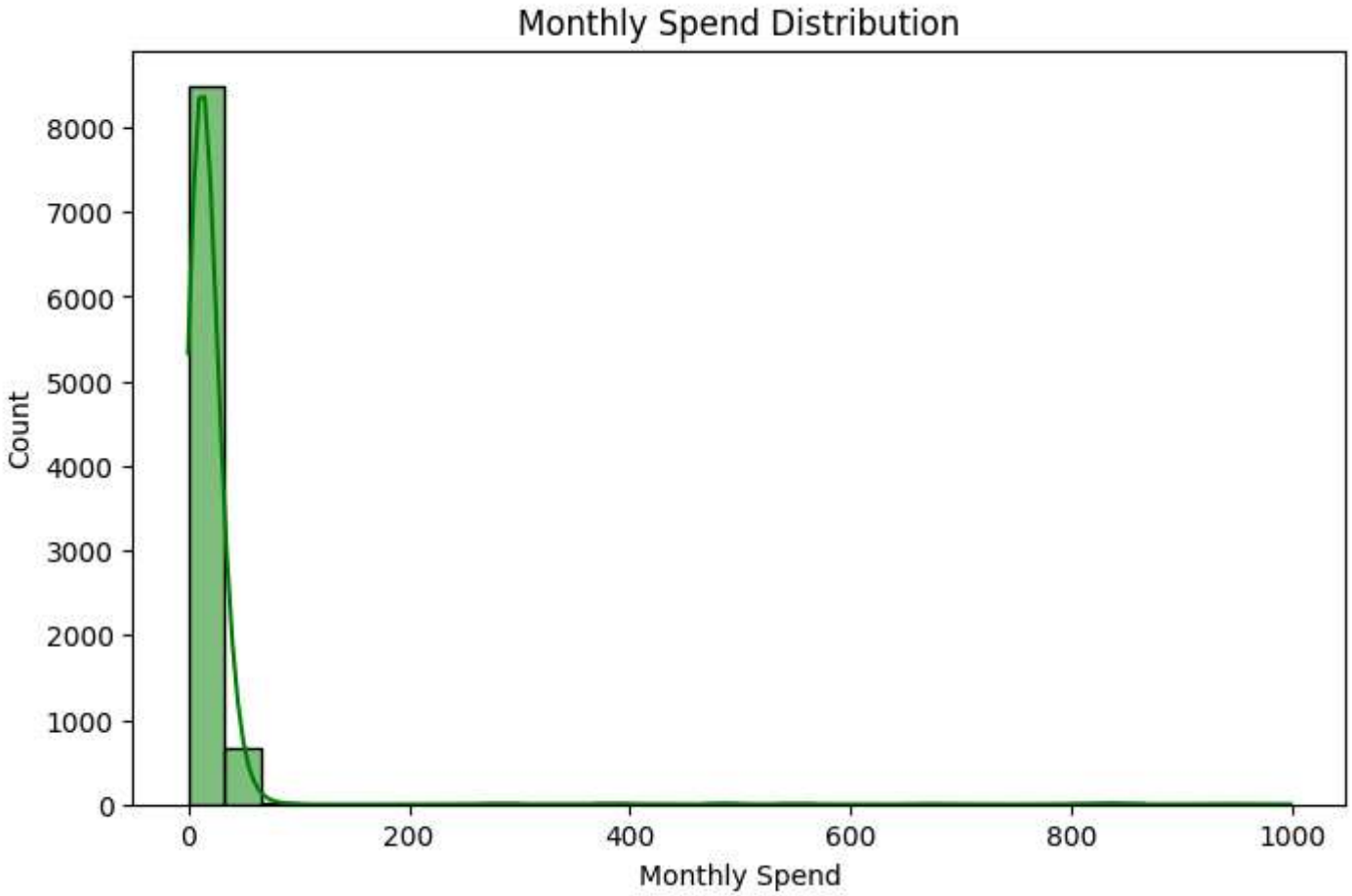


In [6]:
```
# ---- Subscription Plan Distribution ----
plt.figure(figsize=(6,5))
sns.countplot(x='subscription_plan', data=df, palette='Set1')
plt.title("Subscription Plan Distribution")
plt.show()
```

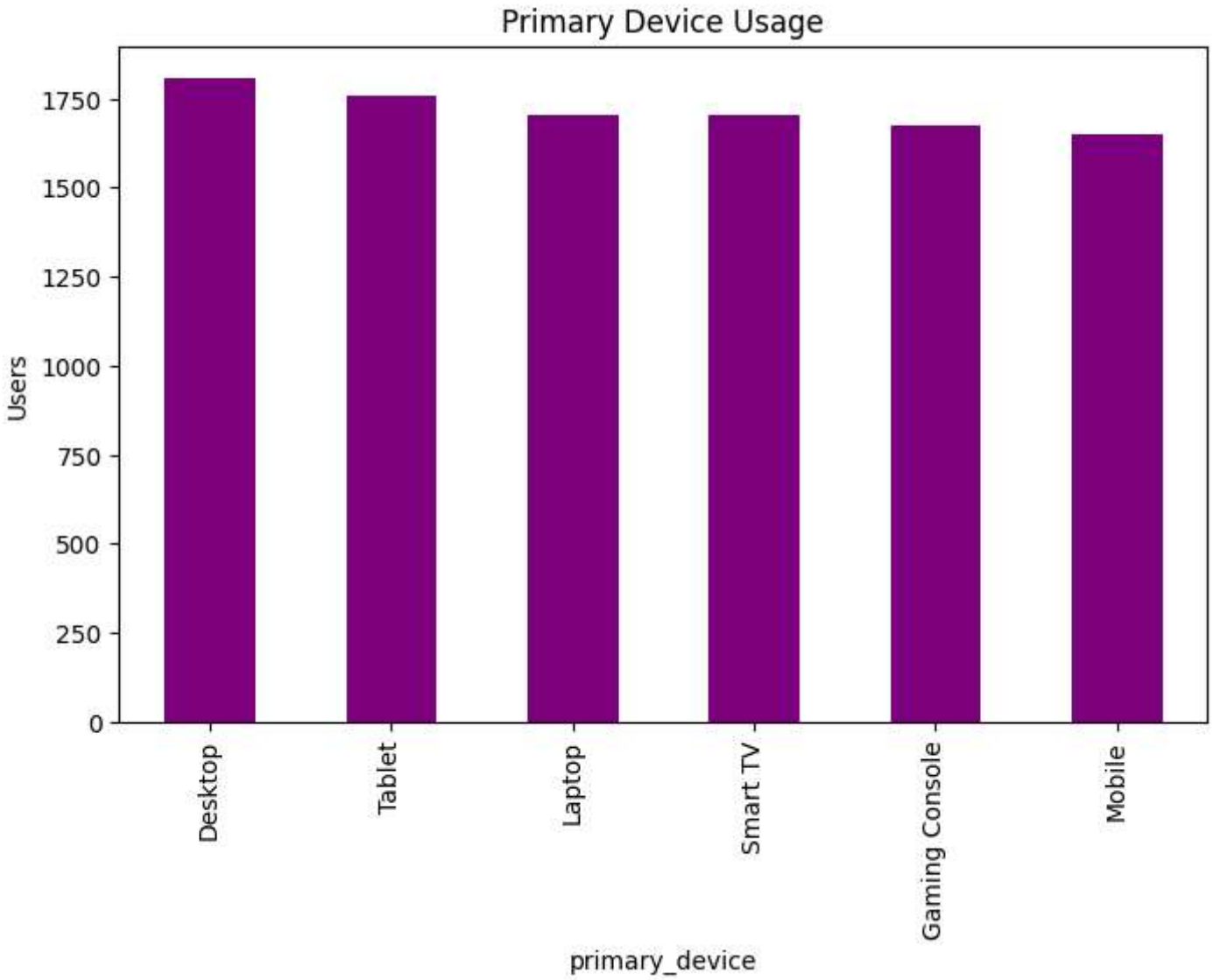
```
C:\Users\Harini\AppData\Local\Temp\ipykernel_12776\1893958190.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `h
ue` and set `legend=False` for the same effect.

  sns.countplot(x='subscription_plan', data=df, palette='Set1')
```



In [7]:
```
# ---- Monthly Spend Distribution ----
plt.figure(figsize=(8,5))
sns.histplot(df['monthly_spend'], bins=30, kde=True, color='green')
plt.title("Monthly Spend Distribution")
plt.xlabel("Monthly Spend")
plt.ylabel("Count")
plt.show()
```

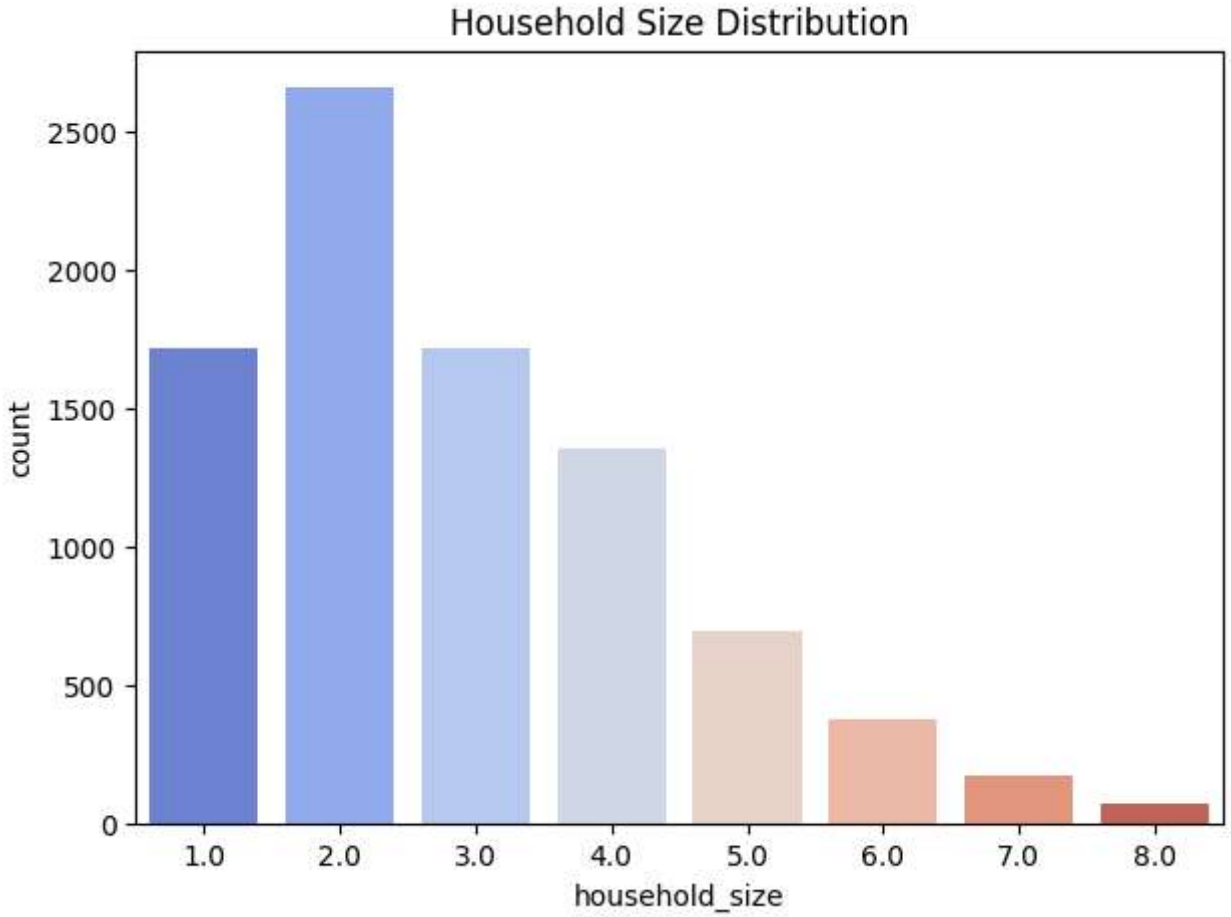


In [8]:
```
# ---- Primary Device Preference ----
plt.figure(figsize=(8,5))
df['primary_device'].value_counts().plot(kind='bar', color='purple')
plt.title("Primary Device Usage")
plt.ylabel("Users")
plt.show()
```



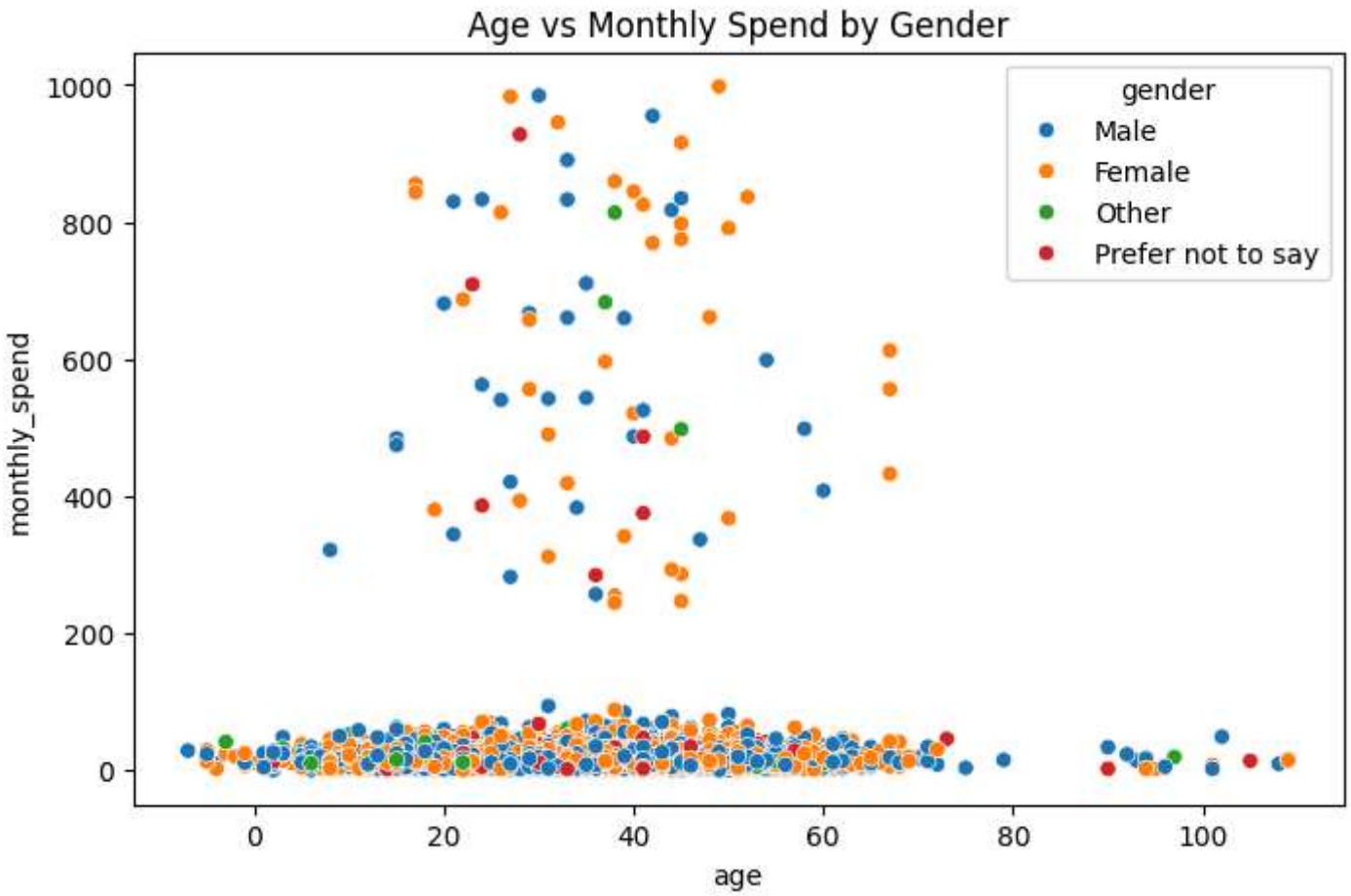
In [9]:
```
# ---- Household Size Distribution ----
plt.figure(figsize=(7,5))
sns.countplot(x='household_size', data=df, palette='coolwarm')
plt.title("Household Size Distribution")
plt.show()
```

```
C:\Users\Harini\AppData\Local\Temp\ipykernel_12776\2565489329.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `h
ue` and set `legend=False` for the same effect.

  sns.countplot(x='household_size', data=df, palette='coolwarm')
```

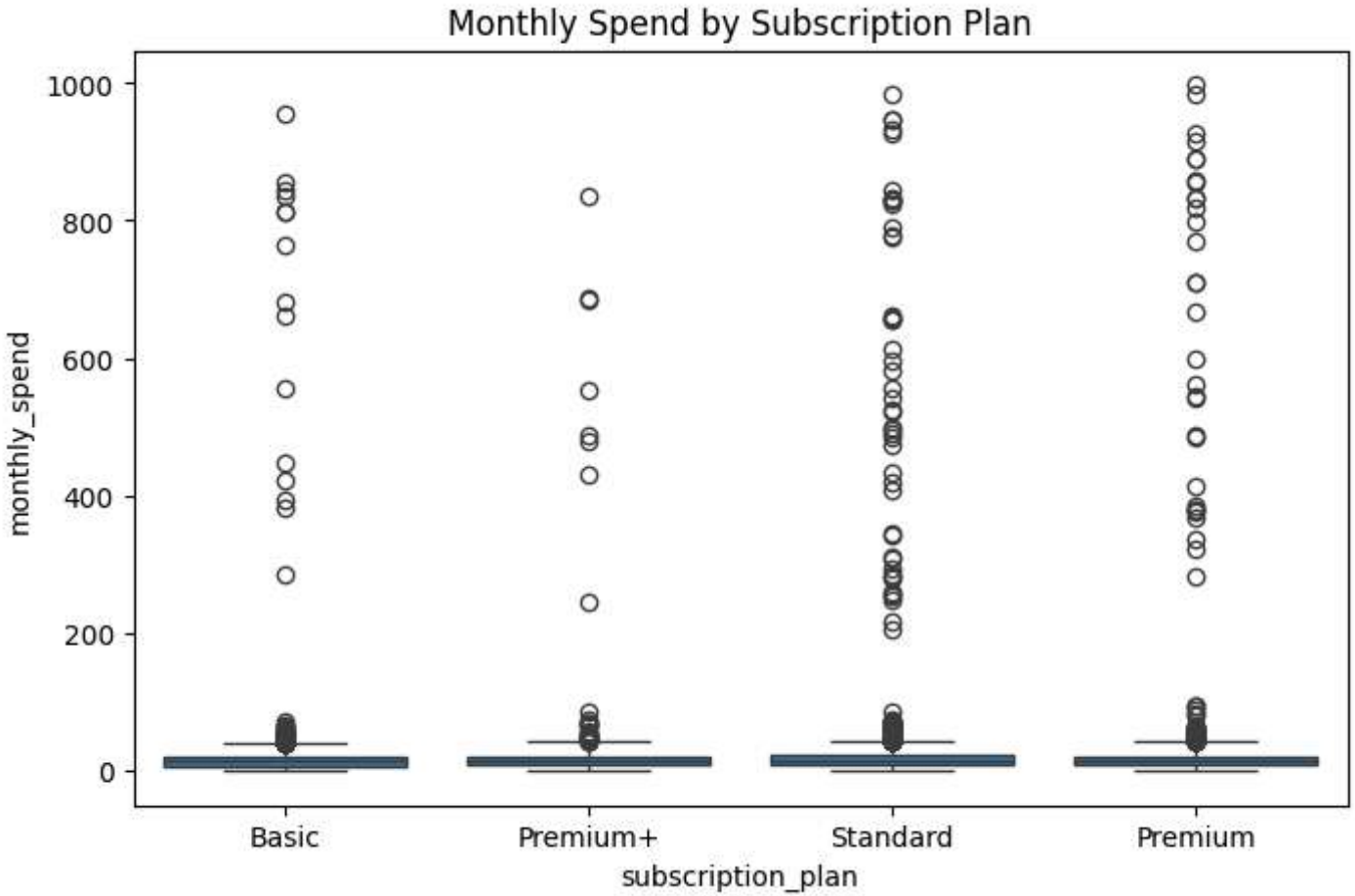


In [10]:
```
# --------- 5. RELATIONSHIPS BETWEEN COLUMNS ---------

# Age vs Monthly Spend
plt.figure(figsize=(8,5))
sns.scatterplot(x='age', y='monthly_spend', hue='gender', data=df)
```

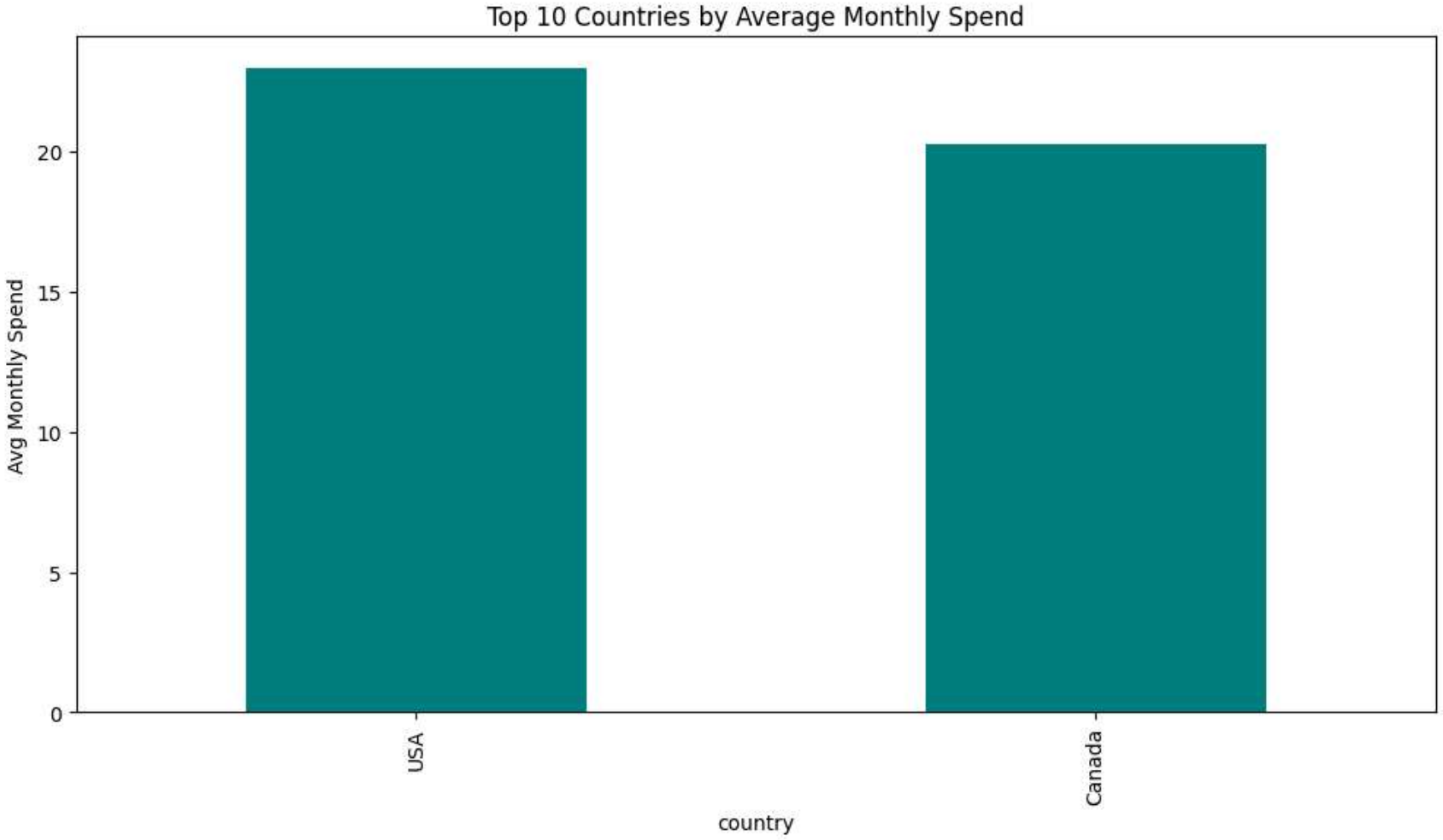
```
plt.title("Age vs Monthly Spend by Gender")
plt.show()
```



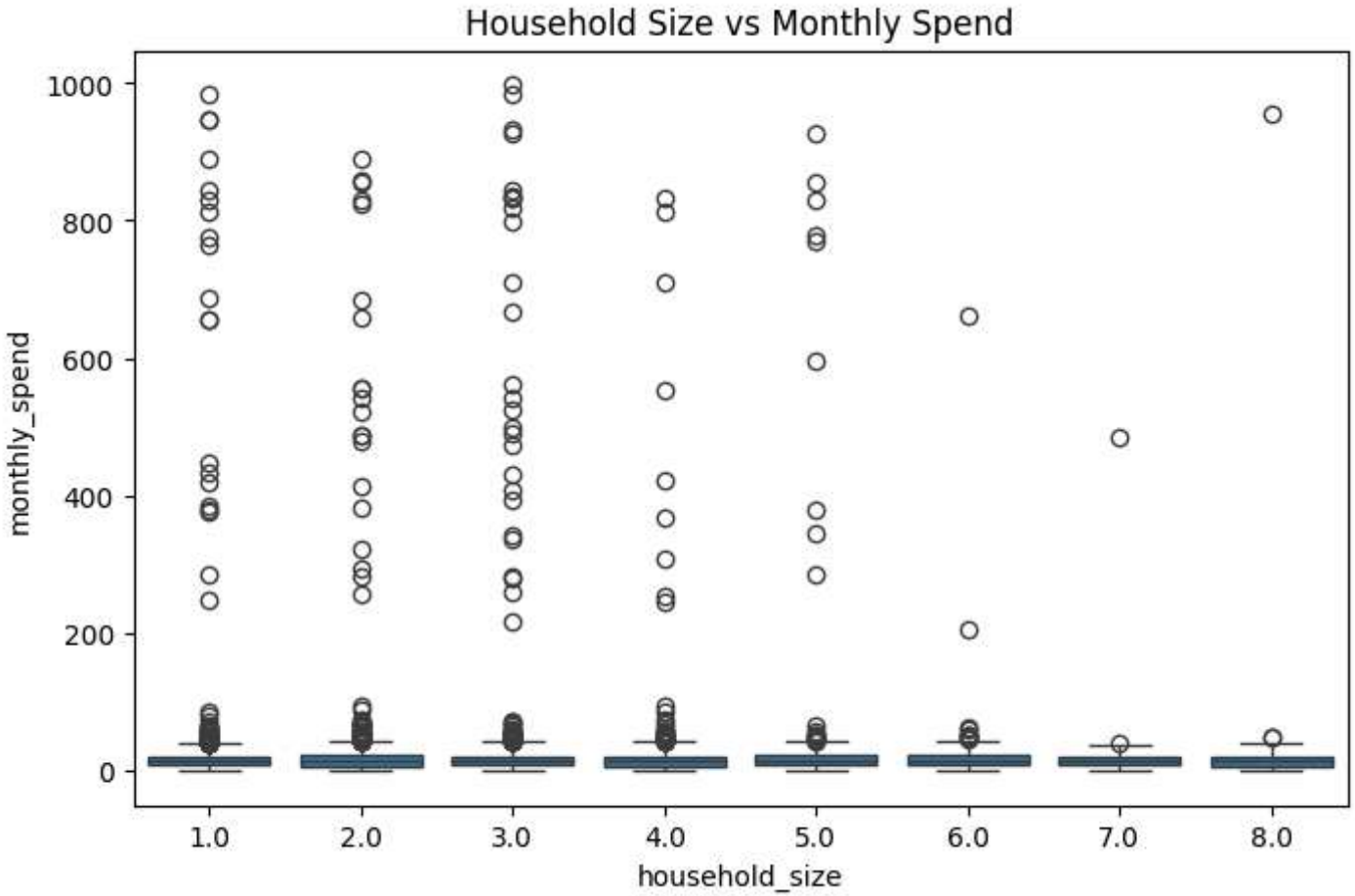
In [11]:
```
# Subscription Plan vs Monthly Spend
plt.figure(figsize=(8,5))
sns.boxplot(x='subscription_plan', y='monthly_spend', data=df)
plt.title("Monthly Spend by Subscription Plan")
plt.show()
```



In [12]:
```
# Country vs Average Spend
avg_spend_country = df.groupby('country')['monthly_spend'].mean().sort_values(ascending=False).head(10)
plt.figure(figsize=(12,6))
avg_spend_country.plot(kind='bar', color='teal')
plt.title("Top 10 Countries by Average Monthly Spend")
plt.ylabel("Avg Monthly Spend")
plt.show()
```

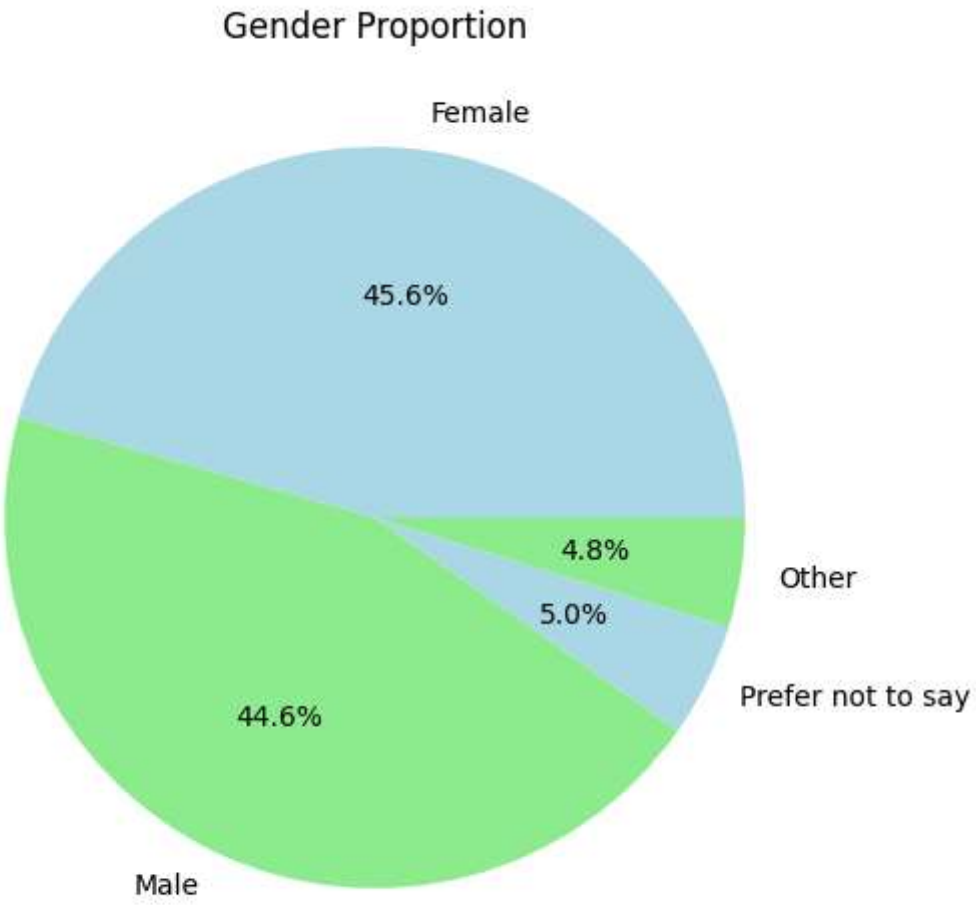


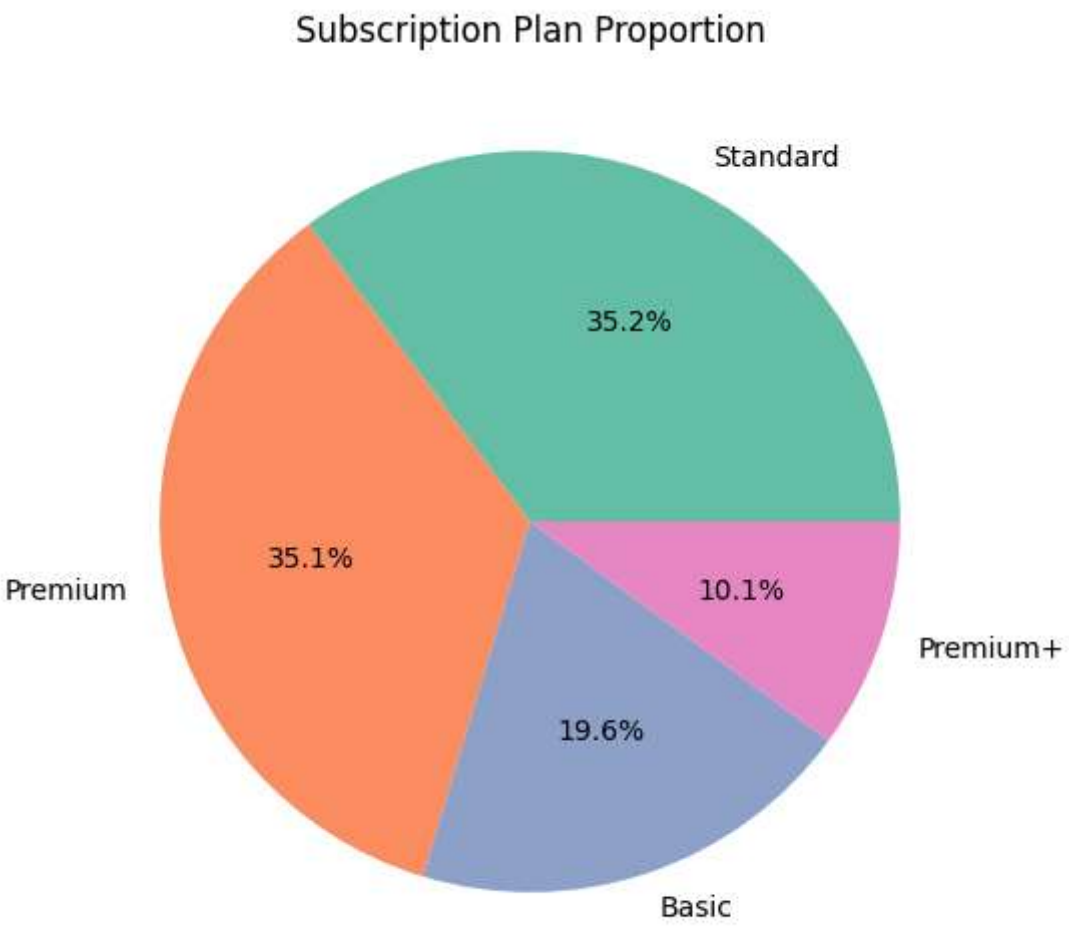
In [13]:
```
# Household Size vs Monthly Spend
plt.figure(figsize=(8,5))
sns.boxplot(x='household_size', y='monthly_spend', data=df)
plt.title("Household Size vs Monthly Spend")
plt.show()
```



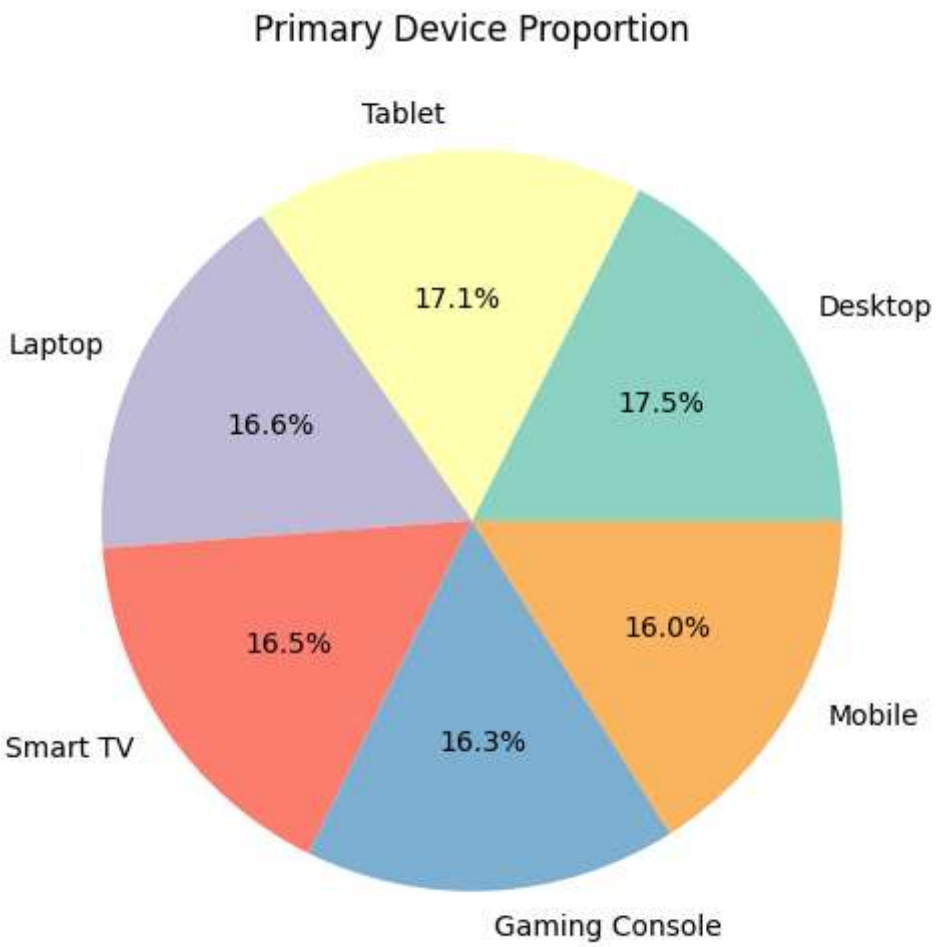
In [17]:
```
# --------- 1. PIE CHARTS FOR CATEGORICAL COLUMNS ---------
# Gender Pie Chart
plt.figure(figsize=(6,6))
df['gender'].value_counts().plot(kind='pie', autopct='%1.1f%%', colors=['lightblue','lightgreen'])
plt.title("Gender Proportion")
plt.ylabel("")
plt.show()

# Subscription Plan Pie Chart
plt.figure(figsize=(6,6))
df['subscription_plan'].value_counts().plot(kind='pie', autopct='%1.1f%%', colors=sns.color_palette('Set2'))
plt.title("Subscription Plan Proportion")
```

```
plt.ylabel("")
plt.show()
```



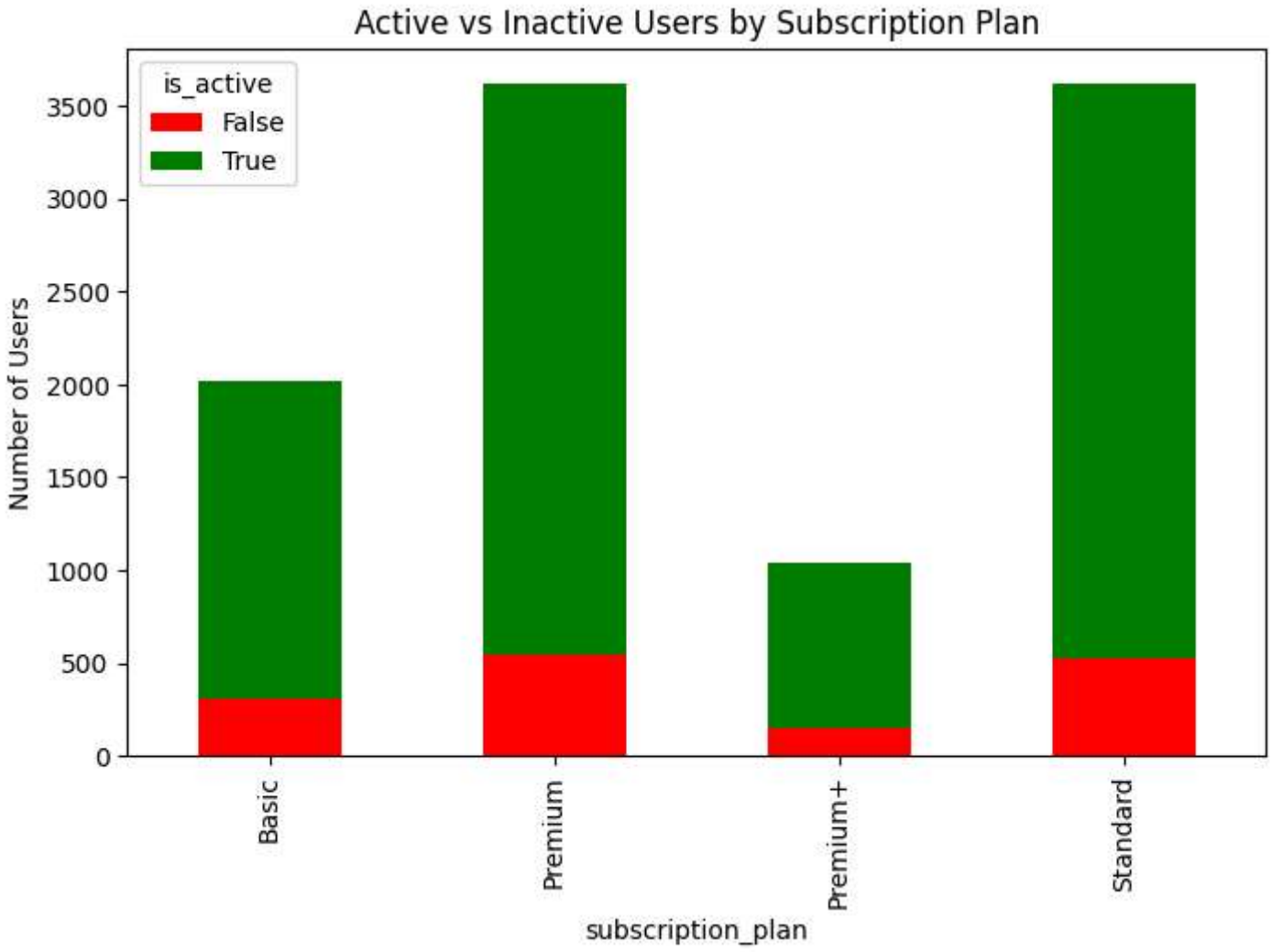


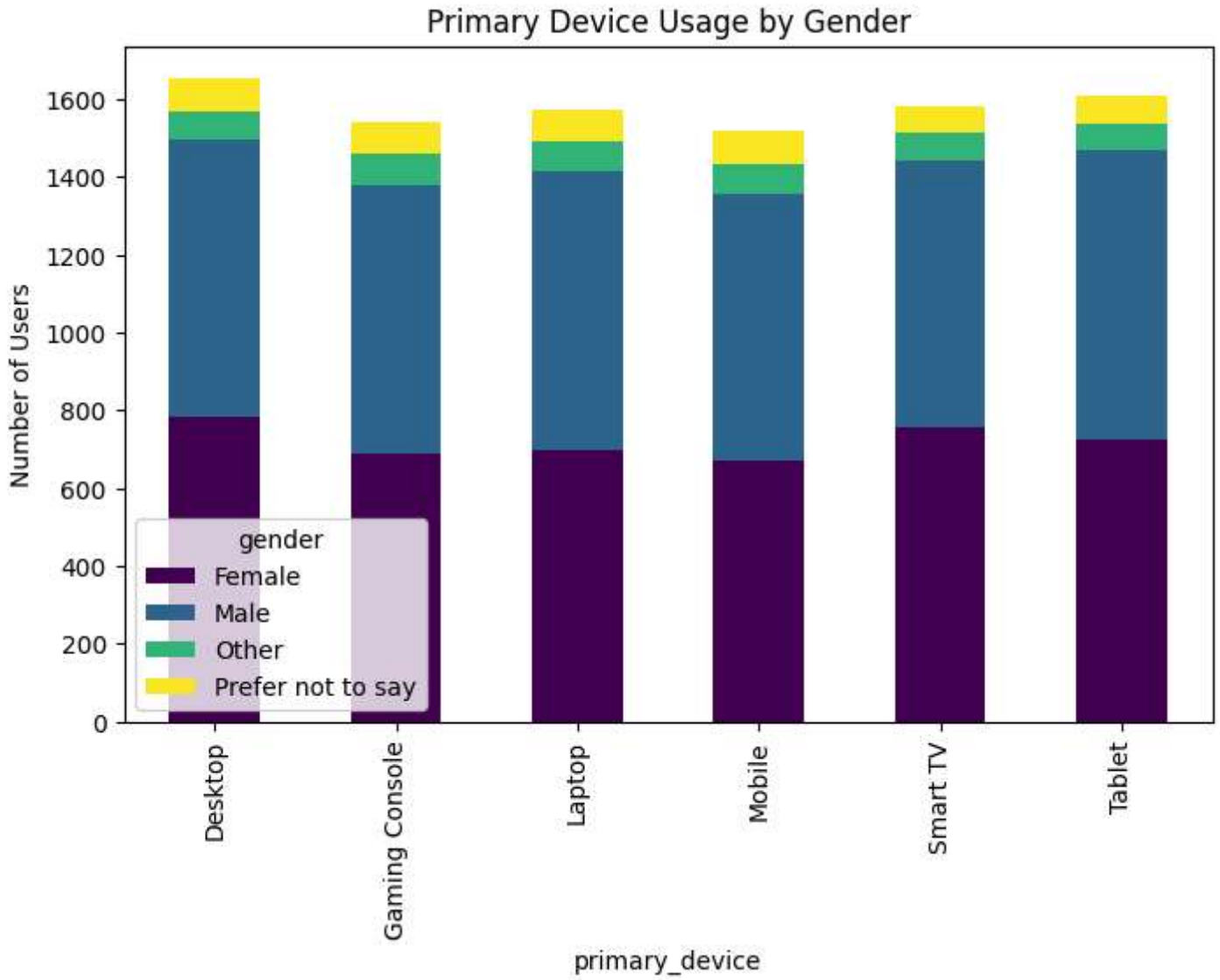
In [18]:
```
# Primary Device Pie Chart
plt.figure(figsize=(6,6))
df['primary_device'].value_counts().plot(kind='pie', autopct='%1.1f%%', colors=sns.color_palette('Set3'))
plt.title("Primary Device Proportion")
plt.ylabel("")
plt.show()
```



In [19]:
```
# --------- 2. STACKED BAR CHARTS ---------
# Active Users by Subscription Plan
active_plan = pd.crosstab(df['subscription_plan'], df['is_active'])
active_plan.plot(kind='bar', stacked=True, figsize=(8,5), color=['red','green'])
plt.title("Active vs Inactive Users by Subscription Plan")
plt.ylabel("Number of Users")
plt.show()

# Device Usage by Gender
device_gender = pd.crosstab(df['primary_device'], df['gender'])
device_gender.plot(kind='bar', stacked=True, figsize=(8,5), colormap='viridis')
```

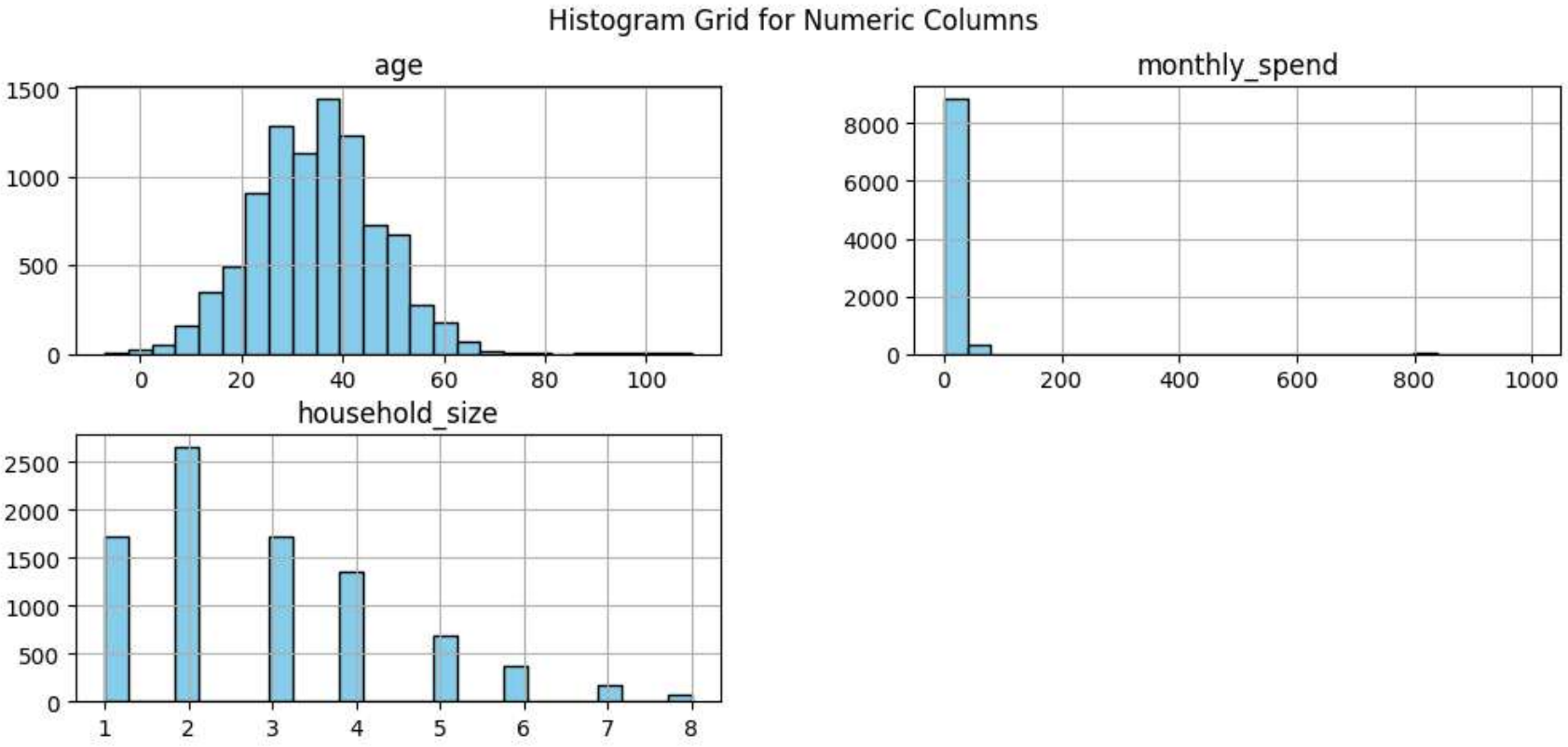
```
plt.title("Primary Device Usage by Gender")
plt.ylabel("Number of Users")
plt.show()
```

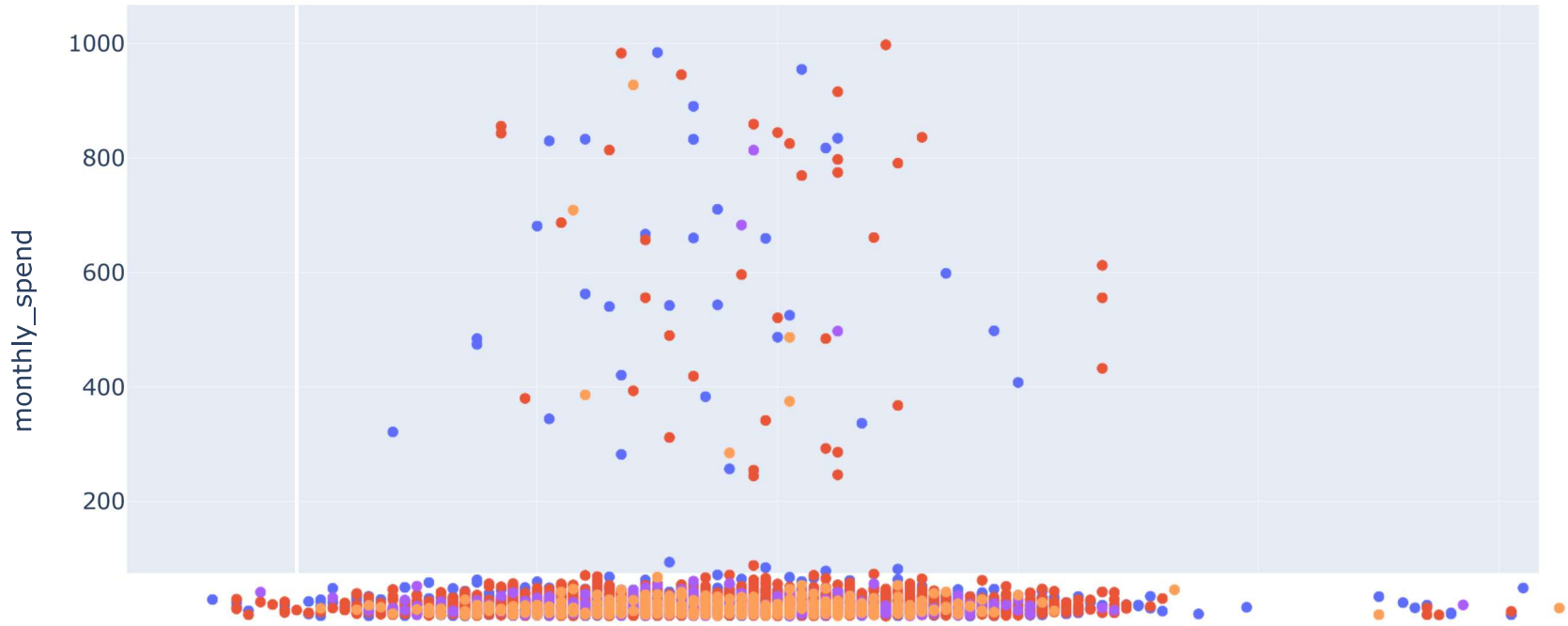




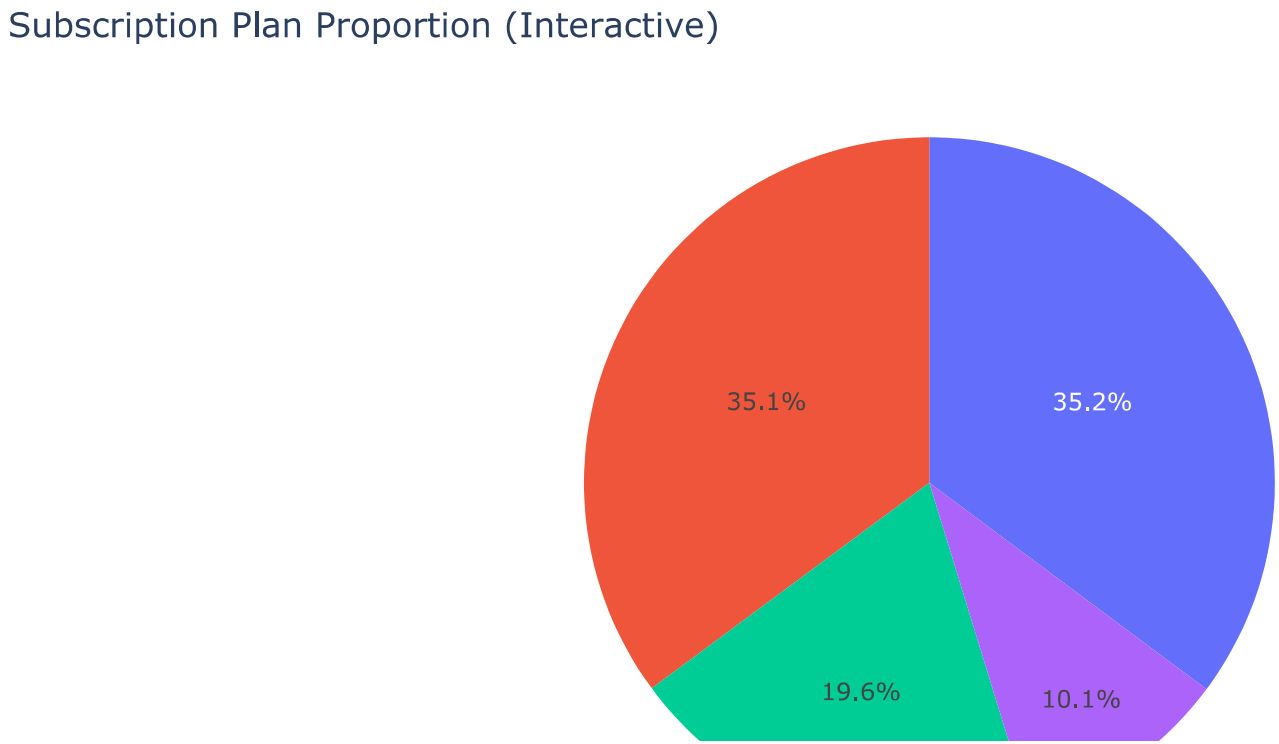
In [20]:
```
# --------- 3. HISTOGRAM GRID FOR NUMERIC COLUMNS ---------
numeric_cols = ['age','monthly_spend','household_size']
df[numeric_cols].hist(bins=25, figsize=(12,5), color='skyblue', edgecolor='black')
plt.suptitle("Histogram Grid for Numeric Columns")
plt.show()
```

```python
# --------- 4. INTERACTIVE PLOTS USING PLOTLY (OPTIONAL) ---------
import plotly.express as px

# Age vs Monthly Spend by Gender
fig = px.scatter(df, x='age', y='monthly_spend', color='gender',
                 hover_data=['subscription_plan','primary_device'],
                 title="Age vs Monthly Spend (Interactive)")
fig.show()
```

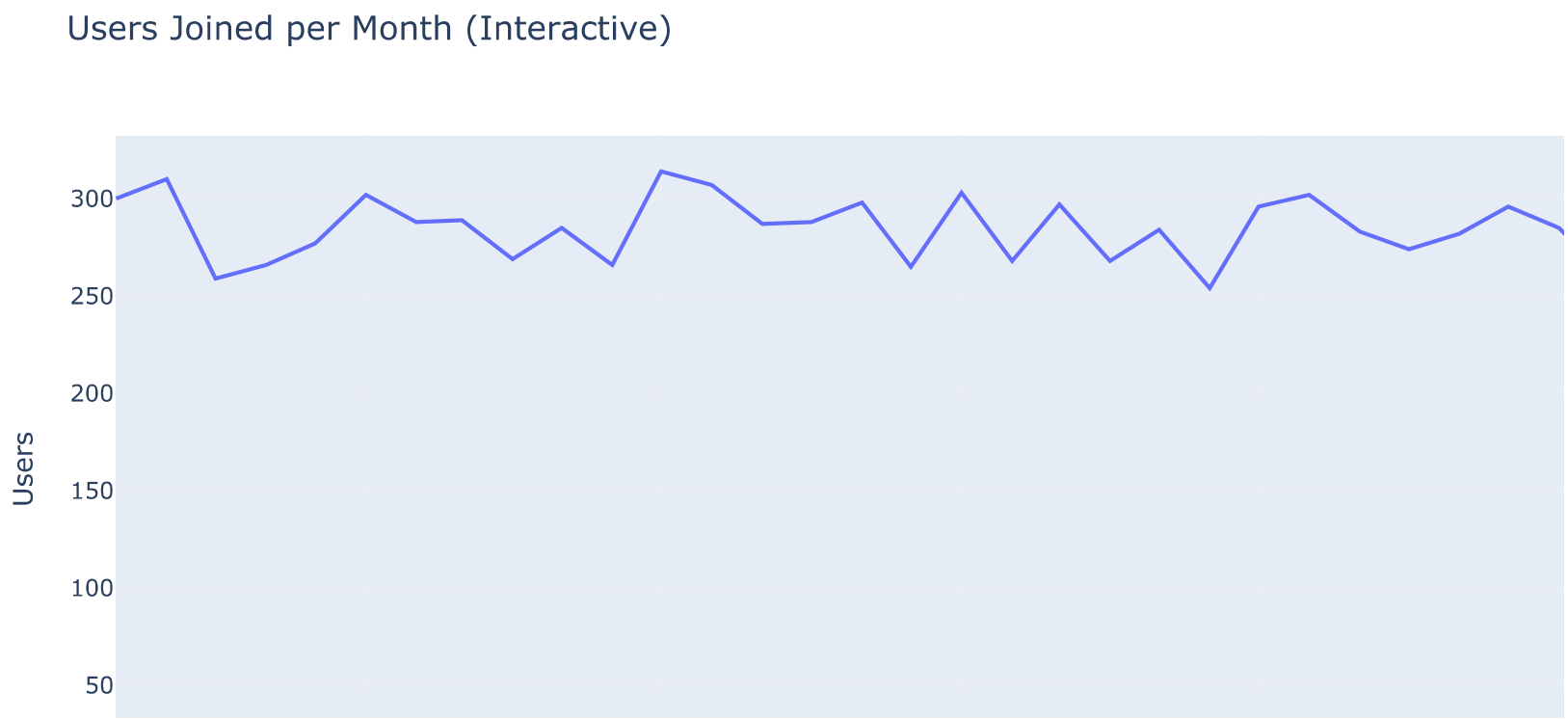




In [22]:
```
# Subscription Plan Distribution (Interactive Pie)
fig2 = px.pie(df, names='subscription_plan', title='Subscription Plan Proportion (Interactive)')
fig2.show()
```



In [23]:
```
# Users Joined Over Time (Interactive Line)
df['join_month'] = pd.to_datetime(df['subscription_start_date']).dt.to_period('M').astype(str)
monthly_users = df['join_month'].value_counts().sort_index()
fig3 = px.line(x=monthly_users.index, y=monthly_users.values, labels={'x':'Month','y':'Users'}, title='Users Joined p
fig3.show()
```



In [24]:
```python
# --------- 1. NUMERIC COLUMN STATISTICS ---------
numeric_cols = ['age', 'monthly_spend', 'household_size']
for col in numeric_cols:
    print(f"--- {col.upper()} STATISTICS ---")
    print("Count:", df[col].count())
    print("Mean:", round(df[col].mean(),2))
    print("Median:", round(df[col].median(),2))
    print("Min:", df[col].min())
    print("Max:", df[col].max())
    print("Standard Deviation:", round(df[col].std(),2))
```

```
    print("25th Percentile:", df[col].quantile(0.25))
    print("75th Percentile:", df[col].quantile(0.75))
    print("\n")
```

```
--- AGE STATISTICS ---
Count: 9071
Mean: 35.04
Median: 35.0
Min: -7.0
Max: 109.0
Standard Deviation: 12.58
25th Percentile: 27.0
75th Percentile: 43.0


--- MONTHLY_SPEND STATISTICS ---
Count: 9283
Mean: 22.15
Median: 13.53
Min: 0.11
Max: 997.8
Standard Deviation: 65.72
25th Percentile: 7.745
75th Percentile: 21.62


--- HOUSEHOLD_SIZE STATISTICS ---
Count: 8755
Mean: 2.86
Median: 2.0
Min: 1.0
Max: 8.0
Standard Deviation: 1.56
25th Percentile: 2.0
75th Percentile: 4.0
```

In [25]:
```
# --------- 2. CATEGORICAL COLUMN DISTRIBUTIONS ---------
categorical_cols = ['gender', 'subscription_plan', 'primary_device', 'is_active']
for col in categorical_cols:
    print(f"--- {col.upper()} DISTRIBUTION ---")
```

```
    counts = df[col].value_counts()
    percentages = round(df[col].value_counts(normalize=True)*100,2)
    distribution = pd.DataFrame({'Count': counts, 'Percentage (%)': percentages})
    print(distribution)
    print("\n")
```

```
--- GENDER DISTRIBUTION ---
                  Count  Percentage (%)
gender
Female             4324           45.63
Male               4228           44.62
Prefer not to say   471            4.97
Other               453            4.78


--- SUBSCRIPTION_PLAN DISTRIBUTION ---
                  Count  Percentage (%)
subscription_plan
Standard           3625           35.19
Premium            3619           35.14
Basic              2020           19.61
Premium+           1036           10.06


--- PRIMARY_DEVICE DISTRIBUTION ---
                Count  Percentage (%)
primary_device
Desktop          1807           17.54
Tablet           1759           17.08
Laptop           1706           16.56
Smart TV         1704           16.54
Gaming Console   1675           16.26
Mobile           1649           16.01


--- IS_ACTIVE DISTRIBUTION ---
           Count  Percentage (%)
is_active
True        8776            85.2
False       1524            14.8
```

In [26]:
```
# --------- 3. TOP COUNTRIES AND CITIES ---------
print("--- TOP 10 COUNTRIES BY USER COUNT ---")
top_countries = df['country'].value_counts().head(10)
print(top_countries)
print("\n")

print("--- TOP 10 CITIES BY USER COUNT ---")
top_cities = df['city'].value_counts().head(10)
print(top_cities)
print("\n")
```

```
--- TOP 10 COUNTRIES BY USER COUNT ---
country
USA       7204
Canada    3096
Name: count, dtype: int64


--- TOP 10 CITIES BY USER COUNT ---
city
North Michael        14
West Michael         10
South Christopher     9
North Brian           9
North Jennifer        9
Lake Michael          9
East Jennifer         8
Thomasmouth           8
South Robert          8
West Robert           8
Name: count, dtype: int64
```

In [27]:
```
# --------- 4. AGE GROUP DISTRIBUTION ---------
bins = [0, 18, 25, 35, 45, 60, 100]
labels = ['<18','18-25','26-35','36-45','46-60','60+']
df['age_group'] = pd.cut(df['age'], bins=bins, labels=labels)
age_group_counts = df['age_group'].value_counts().sort_index()
age_group_percent = round(df['age_group'].value_counts(normalize=True)*100,2).sort_index()
print("--- AGE GROUP DISTRIBUTION ---")
age_group_distribution = pd.DataFrame({'Count': age_group_counts, 'Percentage (%)': age_group_percent})
```

```
print(age_group_distribution)
print("\n")
```

```
--- AGE GROUP DISTRIBUTION ---
           Count  Percentage (%)
age_group
<18          782            8.65
18-25       1185           13.11
26-35       2739           30.31
36-45       2566           28.39
46-60       1609           17.80
60+          157            1.74
```

In [28]:
```
# --------- 5. SUBSCRIPTION PLAN vs ACTIVE STATUS ---------
sub_active = pd.crosstab(df['subscription_plan'], df['is_active'])
print("--- SUBSCRIPTION PLAN vs ACTIVE STATUS ---")
print(sub_active)
print("\n")
```

```
--- SUBSCRIPTION PLAN vs ACTIVE STATUS ---
is_active          False  True
subscription_plan
Basic                305   1715
Premium              543   3076
Premium+             146    890
Standard             530   3095
```

In [29]:
```
# --------- 6. PRIMARY DEVICE vs SUBSCRIPTION PLAN ---------
device_plan = pd.crosstab(df['primary_device'], df['subscription_plan'])
print("--- PRIMARY DEVICE vs SUBSCRIPTION PLAN ---")
print(device_plan)
print("\n")
```

```
--- PRIMARY DEVICE vs SUBSCRIPTION PLAN ---
subscription_plan  Basic  Premium  Premium+  Standard
primary_device
Desktop              335      623       184       665
Gaming Console       344      568       161       602
Laptop               330      596       179       601
Mobile               333      587       180       549
Smart TV             347      609       160       588
Tablet               331      636       172       620
```

In [30]:
```python
# --------- 7. MONTHLY SPEND STATISTICS BY PLAN ---------
print("--- MONTHLY SPEND STATISTICS BY SUBSCRIPTION PLAN ---")
for plan in df['subscription_plan'].unique():
    plan_data = df[df['subscription_plan']==plan]['monthly_spend']
    print(f"{plan} Plan -> Mean: {plan_data.mean():.2f}, Median: {plan_data.median():.2f}, Max: {plan_data.max()}, M
print("\n")
```

```
--- MONTHLY SPEND STATISTICS BY SUBSCRIPTION PLAN ---
Basic Plan -> Mean: 20.46, Median: 13.18, Max: 954.98, Min: 0.25
Premium+ Plan -> Mean: 20.43, Median: 13.56, Max: 836.4, Min: 0.15
Standard Plan -> Mean: 24.02, Median: 13.82, Max: 984.45, Min: 0.11
Premium Plan -> Mean: 21.70, Median: 13.39, Max: 997.8, Min: 0.11
```

In [ ]:
```python

```