In [10]:

```
import numpy as np # linear algebra operation
import pandas as pd # used for data preparation
import plotly.express as plt # used for data visualization
import seaborn as sns # used for data visualization
import matplotlib.pyplot as plt # used for data visualization
```

In [3]:

```
netflix_users=pd.read_csv("Netflix Userbase.csv")
netflix_users
```

Out[3]:

	User ID	Subscription Type	Monthly Revenue	Join Date	Last Payment Date	Country	Age	Gender	Device	Plan Duration
0	1	Basic	10	15-01-22	10-06-23	United States	28	Male	Smartphone	1 Month
1	2	Premium	15	05-09-21	22-06-23	Canada	35	Female	Tablet	1 Month
2	3	Standard	12	28-02-23	27-06-23	United Kingdom	42	Male	Smart TV	1 Month
3	4	Standard	12	10-07-22	26-06-23	Australia	51	Female	Laptop	1 Month
4	5	Basic	10	01-05-23	28-06-23	Germany	33	Male	Smartphone	1 Month
...	...	...	...	...	...	...	...	...	...	...
2495	2496	Premium	14	25-07-22	12-07-23	Spain	28	Female	Smart TV	1 Month
2496	2497	Basic	15	04-08-22	14-07-23	Spain	33	Female	Smart TV	1 Month
2497	2498	Standard	12	09-08-22	15-07-23	United States	38	Male	Laptop	1 Month
2498	2499	Standard	13	12-08-22	12-07-23	Canada	48	Female	Tablet	1 Month
2499	2500	Basic	15	13-08-22	12-07-23	United States	35	Female	Smart TV	1 Month

2500 rows × 10 columns

In [5]:

```
#head function gives top 5 rows
netflix_users.head()
```

Out[5]:

	User ID	Subscription Type	Monthly Revenue	Join Date	Last Payment Date	Country	Age	Gender	Device	Plan Duration
0	1	Basic	10	15-01-22	10-06-23	United States	28	Male	Smartphone	1 Month
1	2	Premium	15	05-09-21	22-06-23	Canada	35	Female	Tablet	1 Month
2	3	Standard	12	28-02-23	27-06-23	United Kingdom	42	Male	Smart TV	1 Month
3	4	Standard	12	10-07-22	26-06-23	Australia	51	Female	Laptop	1 Month
4	5	Basic	10	01-05-23	28-06-23	Germany	33	Male	Smartphone	1 Month

In [4]:

```
netflix_users.shape
```

Out[4]:

```
(2500, 10)
```

In [21]:

```
#counts number of null values in each column
netflix_users.isnull().sum()
```

Out[21]:

```
User ID              0
Subscription Type    0
Monthly Revenue      0
Join Date            0
Last Payment Date    0
Country              0
Age                  0
Gender               0
Device               0
Plan Duration        0
Days Active          0
dtype: int64
```

In [6]:

```
netflix_users.columns
```

Out[6]:

```
Index(['User ID', 'Subscription Type', 'Monthly Revenue', 'Join Date',
       'Last Payment Date', 'Country', 'Age', 'Gender', 'Device',
       'Plan Duration'],
      dtype='object')
```

In [22]:

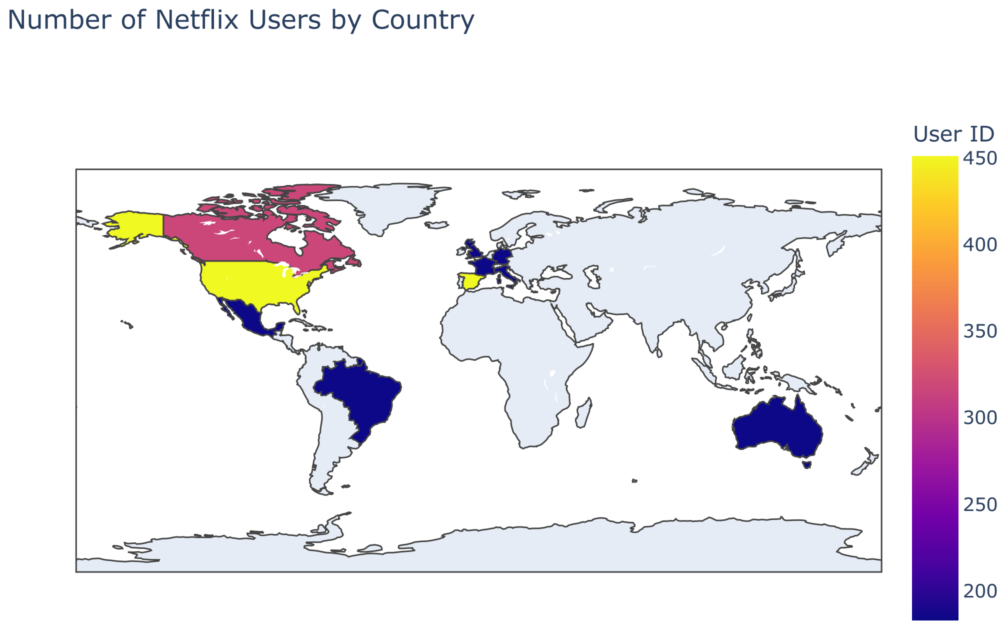
```
netflix_users.info()
```

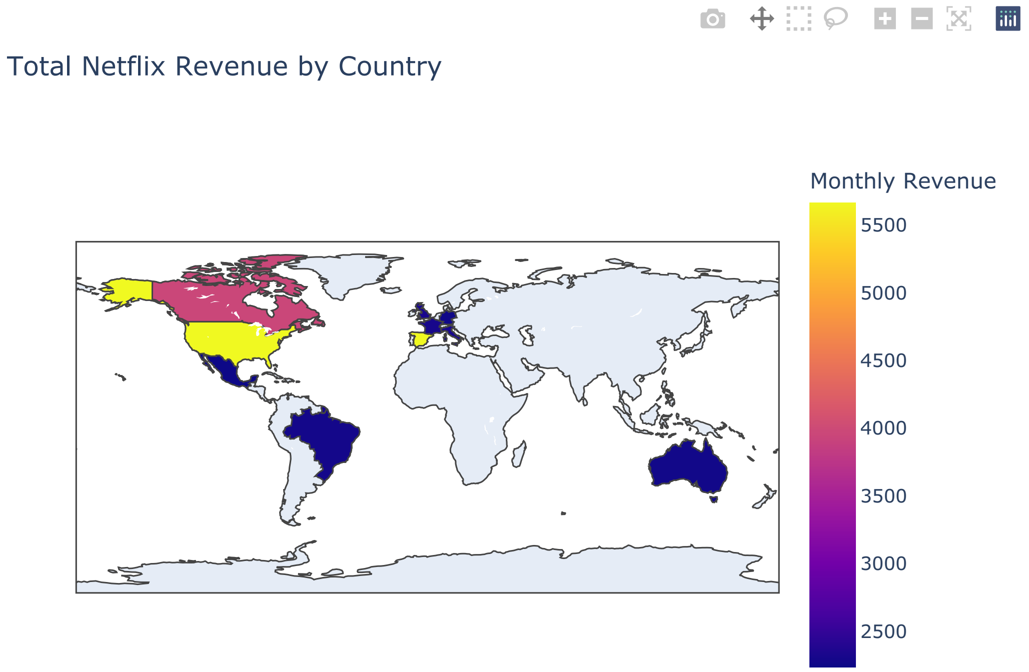
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype         
---  ------             --------------  -----         
 0   User ID            2500 non-null   int64         
 1   Subscription Type  2500 non-null   object        
 2   Monthly Revenue    2500 non-null   int64         
 3   Join Date          2500 non-null   datetime64[ns]
 4   Last Payment Date  2500 non-null   datetime64[ns]
 5   Country            2500 non-null   object        
 6   Age                2500 non-null   int64         
 7   Gender             2500 non-null   object        
 8   Device             2500 non-null   object        
 9   Plan Duration      2500 non-null   object        
 10  Days Active        2500 non-null   int64         
dtypes: datetime64[ns](2), int64(4), object(5)
memory usage: 215.0+ KB
```

NETFLIX_USERS DATA VISUALIZATION

In [8]:

```python
# Group the data by country and calculate the total number of users and total revenue from each country
country_data = netflix_users.groupby('Country').agg({'User ID': 'count', 'Monthly Revenue': 'sum'}).reset_index

# Create a choropleth map showing the number of Netflix users in each country
fig1 = px.choropleth(country_data, locations='Country', locationmode='country names',
                     color='User ID', title='Number of Netflix Users by Country',
                     hover_name='Country', color_continuous_scale='Plasma')
fig1.show()

# Create a choropleth map showing the total revenue from each country
fig2 = px.choropleth(country_data, locations='Country', locationmode='country names',
                     color='Monthly Revenue', title='Total Netflix Revenue by Country',
                     hover_name='Country', color_continuous_scale='Plasma')
fig2.show()
```

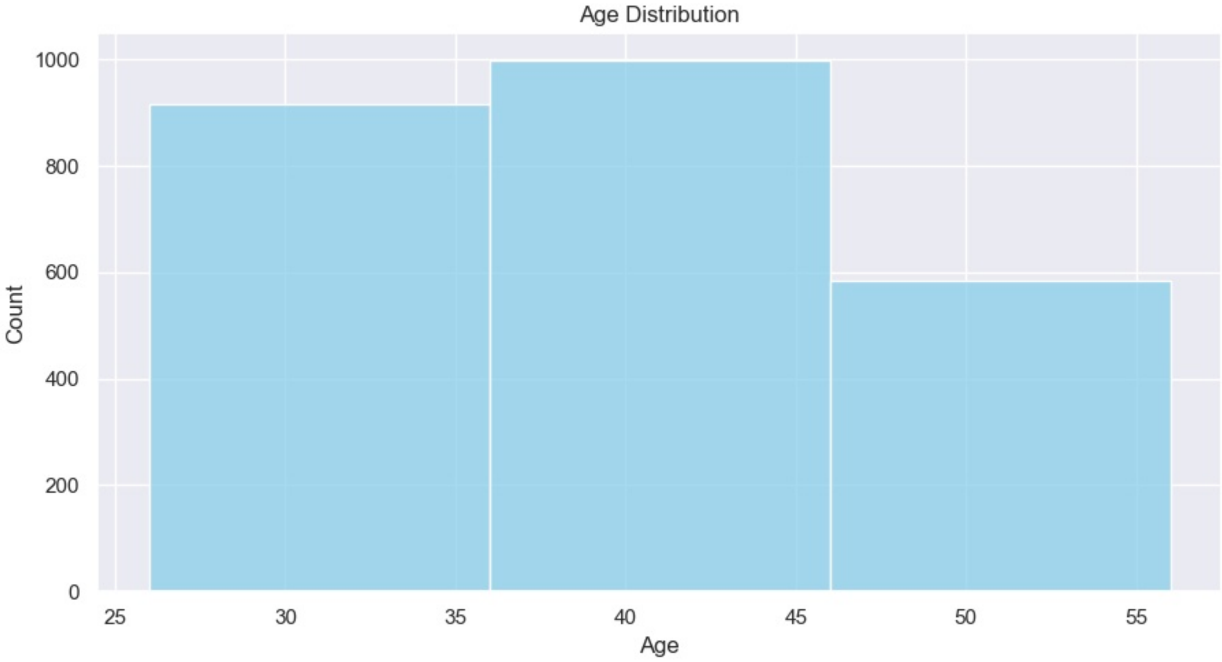


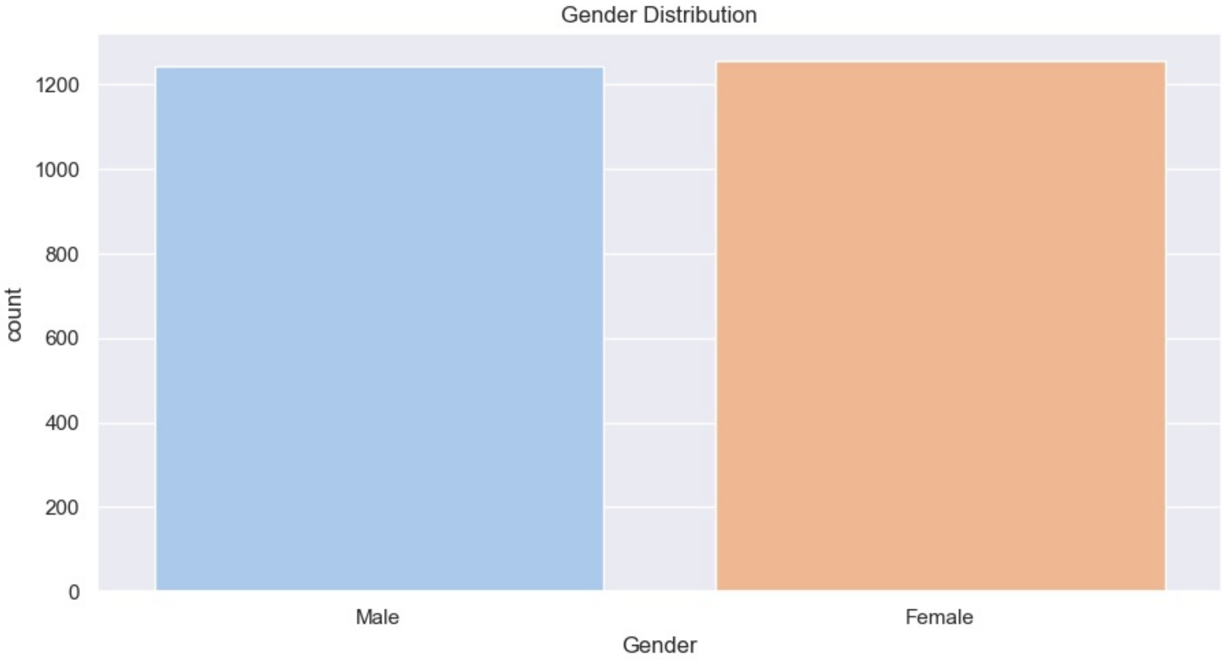


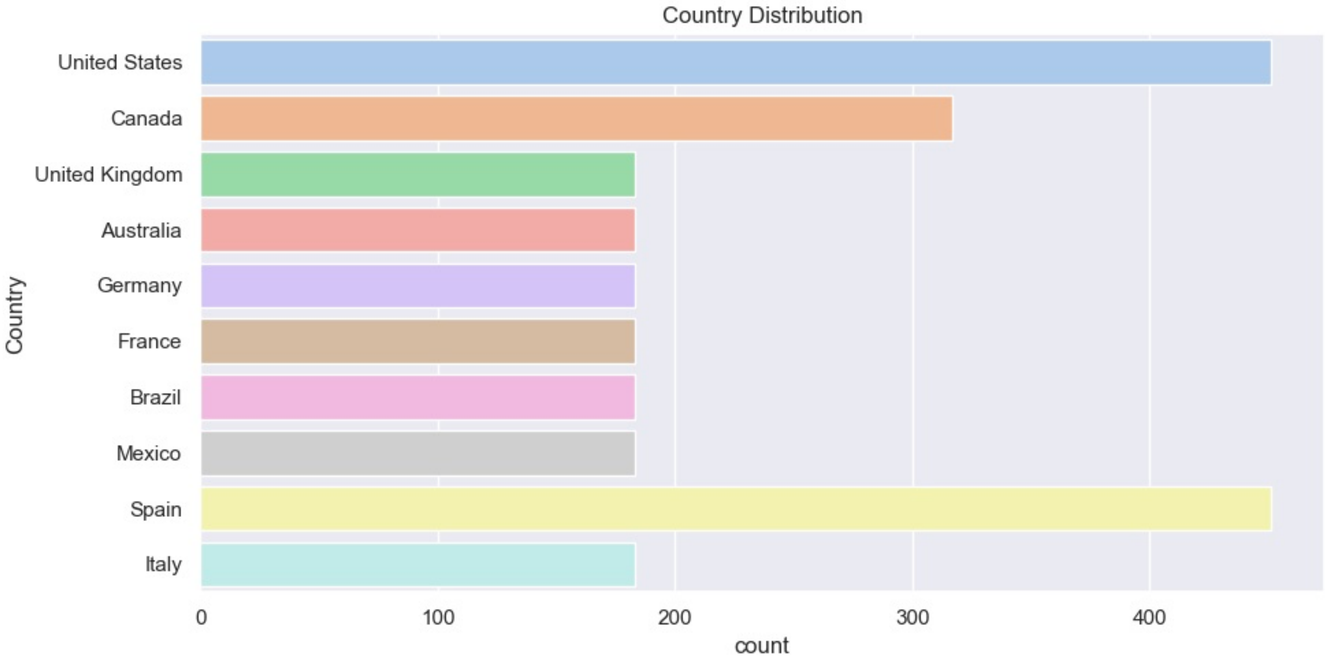
To begin, we aim to understand the demographics of our user base in terms of age gender and geography.¶

In [12]:

```
# Set the theme for the plots
sns.set_theme()

# Create a figure with three subplots
fig, axes = plt.subplots(3, 1, figsize=(10, 15))

# Plot the age distribution
sns.histplot(data=netflix_users, x="Age", binwidth=10, color='skyblue', ax=axes[0])
axes[0].set_title('Age Distribution')

# Plot the gender distribution
sns.countplot(data=netflix_users, x="Gender", palette='pastel', ax=axes[1])
axes[1].set_title('Gender Distribution')

# Plot the country distribution
sns.countplot(y="Country", data=netflix_users, palette='pastel', ax=axes[2])
axes[2].set_title('Country Distribution')

# Adjust the layout
plt.tight_layout()
plt.show()
```



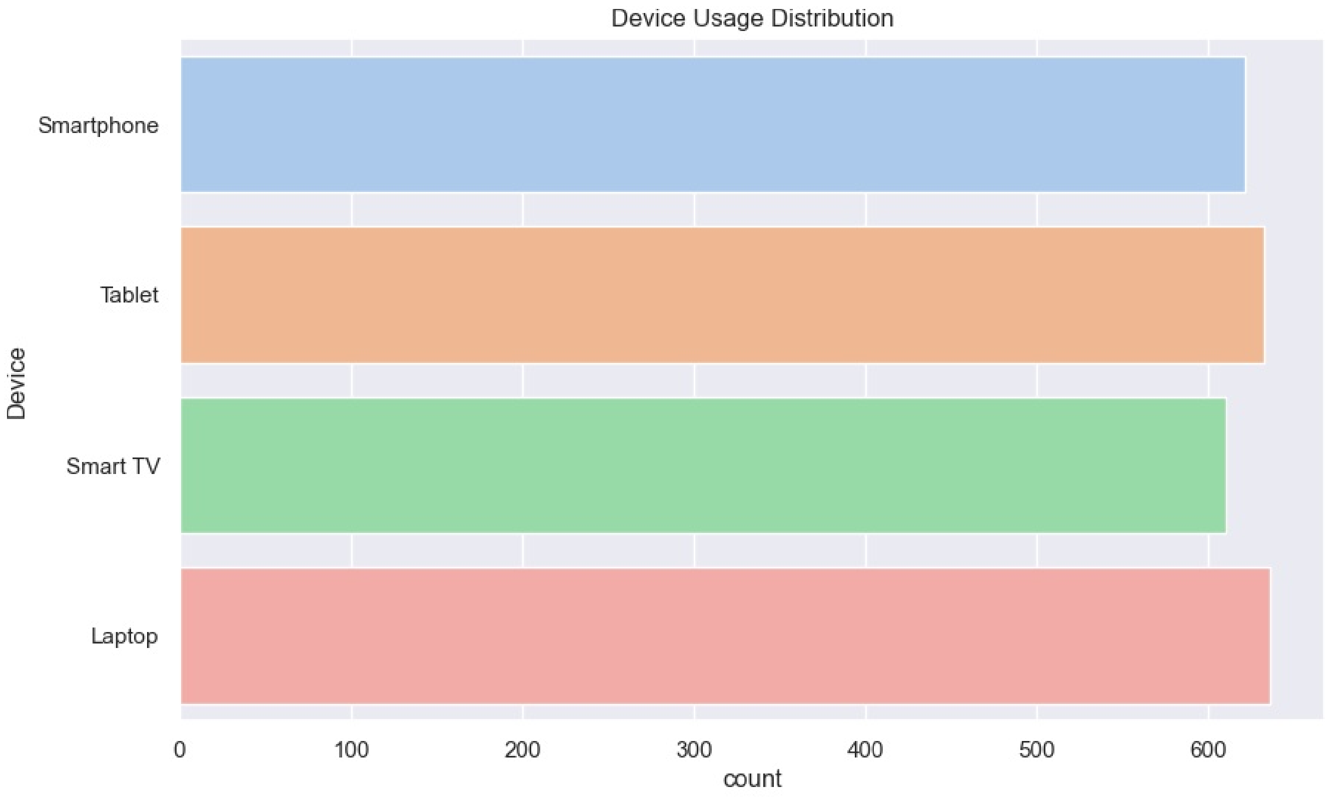




In this part, we will look at the devices our users are using to watch Netflix. This will give us insight into how our platform is being used.

In [13]:

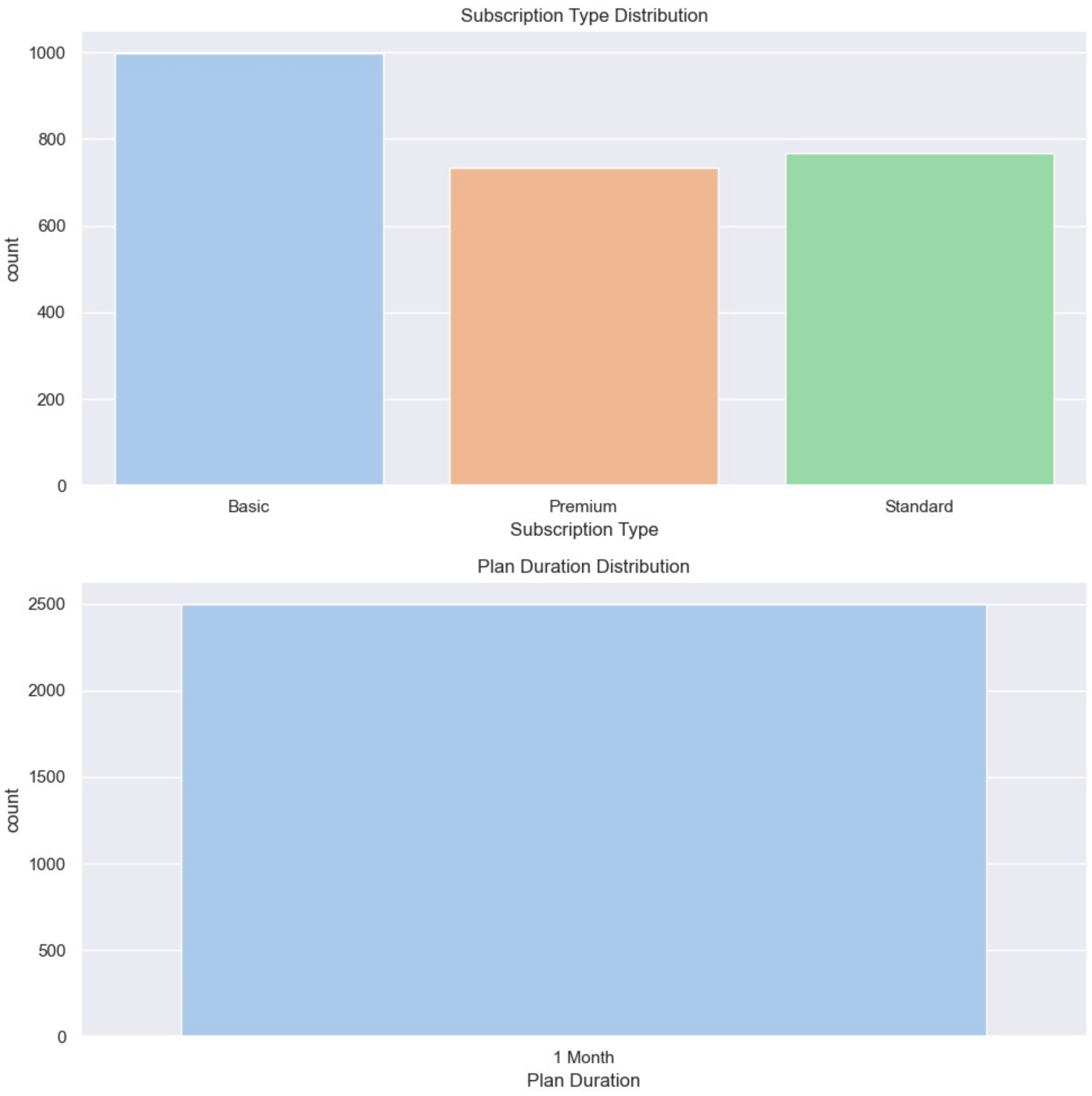
```
# Create a figure for the plot
plt.figure(figsize=(10, 6))

# Plot the device usage distribution
sns.countplot(y="Device", data=netflix_users, palette='pastel')
plt.title('Device Usage Distribution')

# Show the plot
plt.show()
```



We will examine the types of subscription plans our users have and how long they tend to keep their subscriptions. This will help us understand our revenue streams

In [14]:

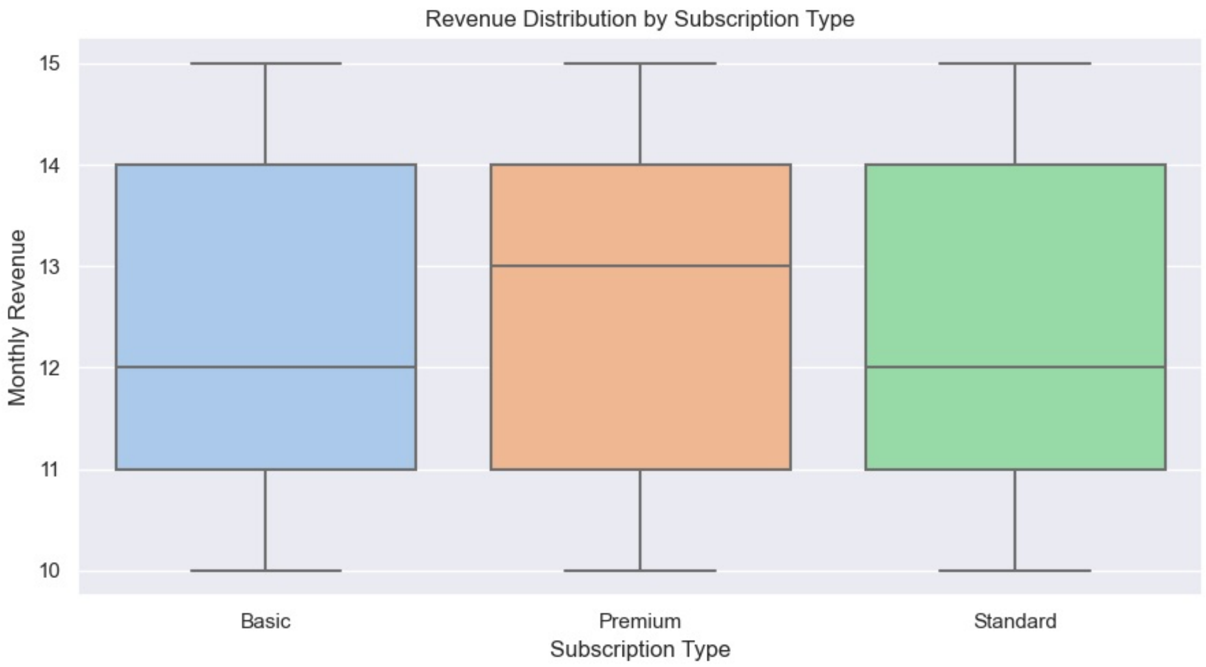
```
# Create a figure with two subplots
fig, axes = plt.subplots(2, 1, figsize=(10, 10))

# Plot the subscription type distribution
sns.countplot(data=netflix_users, x="Subscription Type", palette='pastel', ax=axes[0])
axes[0].set_title('Subscription Type Distribution')

# Plot the plan duration distribution
sns.countplot(data=netflix_users, x="Plan Duration", palette='pastel', ax=axes[1])
axes[1].set_title('Plan Duration Distribution')

# Adjust the layout
plt.tight_layout()
plt.show()
```

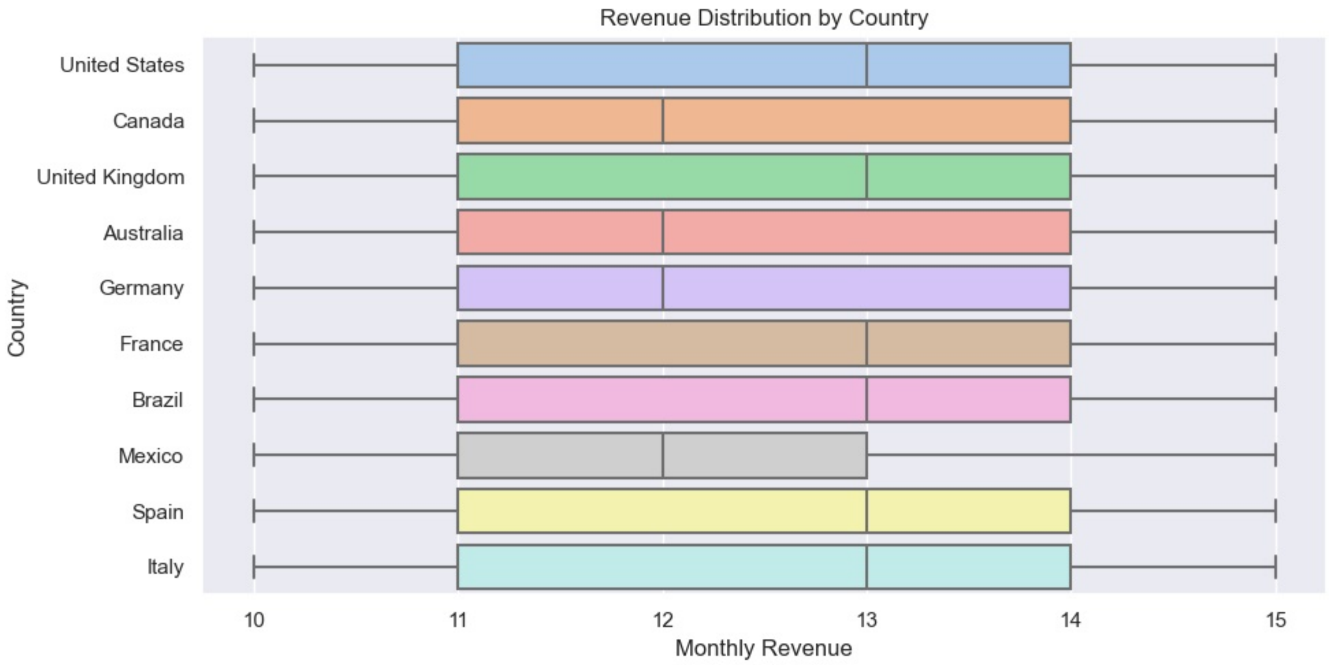


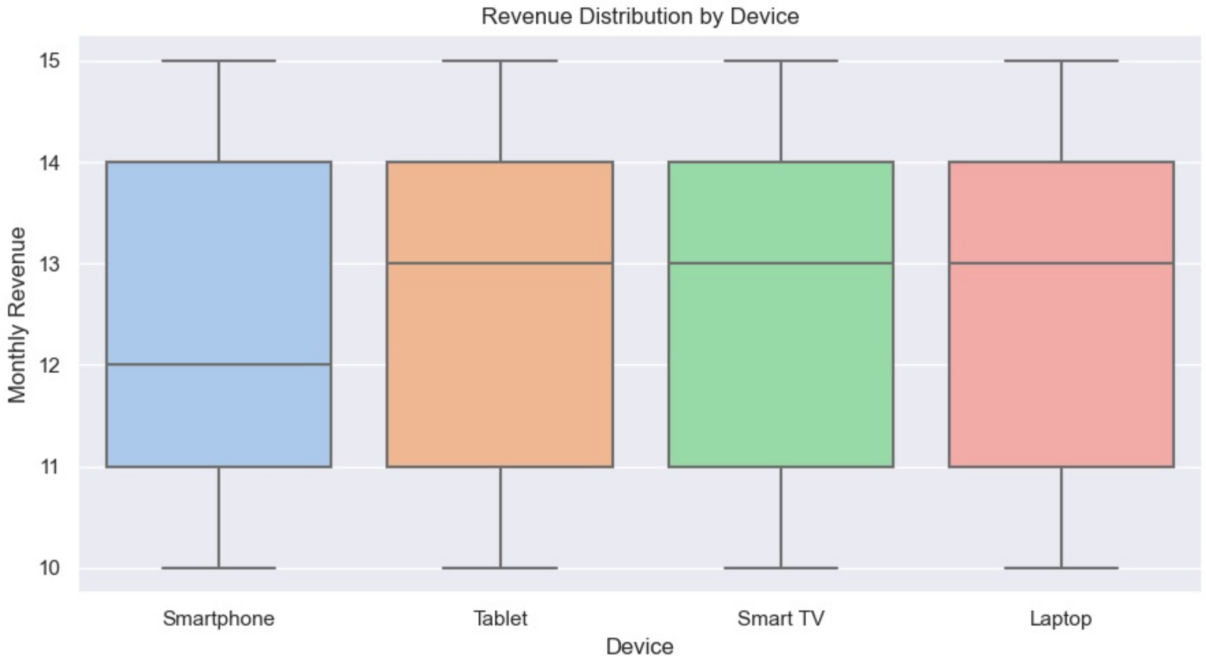
We will look at how much revenue is coming in from each user and how this relates to their subscription type, location, and device usage.

In [16]:

```python
# Create a figure with three subplots
fig, axes = plt.subplots(3, 1, figsize=(10, 15))

# Plot the revenue distribution by subscription type
sns.boxplot(x="Subscription Type", y="Monthly Revenue", data=netflix_users, palette='pastel', ax=axes[0])
axes[0].set_title('Revenue Distribution by Subscription Type')

# Plot the revenue distribution by country
sns.boxplot(y="Country", x="Monthly Revenue", data=netflix_users, palette='pastel', ax=axes[1])
axes[1].set_title('Revenue Distribution by Country')

# Plot the revenue distribution by device
sns.boxplot(x="Device", y="Monthly Revenue", data=netflix_users, palette='pastel', ax=axes[2])
axes[2].set_title('Revenue Distribution by Device')

# Adjust the layout
plt.tight_layout()
plt.show()
```







In [20]:

```
from datetime import datetime

# Convert the 'Join Date' and 'Last Payment Date' columns to datetime objects
netflix_users['Join Date'] = pd.to_datetime(netflix_users['Join Date'], format='%d-%m-%y')
netflix_users['Last Payment Date'] = pd.to_datetime(netflix_users['Last Payment Date'], format='%d-%m-%y')
```

```
# Calculate the number of days between the join date and the last payment date
netflix_users['Days Active'] = (netflix_users['Last Payment Date'] - netflix_users['Join Date']).dt.days

# Calculate the churn rate
churn_rate = (netflix_users['Days Active'] < 30).mean()

churn_rate
```

Out[20]:

```
0.0016
```

In [ ]: