Netflix Content Strategy Analysis

In [1]:

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
pio.templates.default = 'plotly_white'
```

In [2]:

```
data = pd.read_csv('netflix_content_2023.csv')
data.head()
```

Out[2]:

	Title	Available Globally?	Release Date	Hours Viewed	Language Indicator	Content Type
0	The Night Agent: Season 1	Yes	2023-03-23	81,21,00,000	English	Show
1	Ginny & Georgia: Season 2	Yes	2023-01-05	66,51,00,000	English	Show
2	The Glory: Season 1 // 더 글로리: 시즌 1	Yes	2022-12-30	62,28,00,000	Korean	Show
3	Wednesday: Season 1	Yes	2022-11-23	50,77,00,000	English	Show
4	Queen Charlotte: A Bridgerton Story	Yes	2023-05-04	50,30,00,000	English	Movie

In [3]:

```
data.shape #no of columns and rows in the data
```

Out[3]:

```
(24812, 6)
```

In [4]:

```
#Data Cleaning and preprocessing the
#Hours Viewed column to prepare it for analysis

data['Hours Viewed']=data['Hours Viewed'].replace(',','',regex=True).astype(float)

data.head()
```

Out[4]:

	Title	Available Globally?	Release Date	Hours Viewed	Language Indicator	Content Type
0	The Night Agent: Season 1	Yes	2023-03-23	812100000.0	English	Show
1	Ginny & Georgia: Season 2	Yes	2023-01-05	665100000.0	English	Show
2	The Glory: Season 1 // 더 글로리: 시즌 1	Yes	2022-12-30	622800000.0	Korean	Show
3	Wednesday: Season 1	Yes	2022-11-23	507700000.0	English	Show
4	Queen Charlotte: A Bridgerton Story	Yes	2023-05-04	503000000.0	English	Movie

In [5]:

```
data.duplicated() #checking duplicates in the data
```

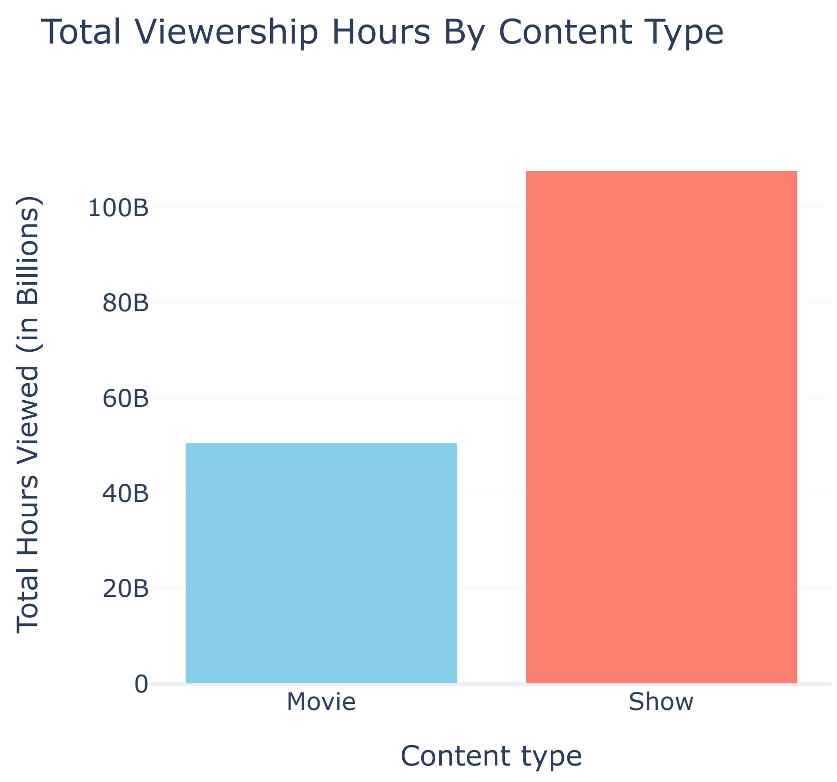
Out[5]:

```
0        False
1        False
2        False
3        False
4        False
         ...  
24807    False
24808     True
24809     True
24810     True
24811     True
Length: 24812, dtype: bool
```

Analyze trends in content type to determine whether shows or movies dominate viewership

In [6]:

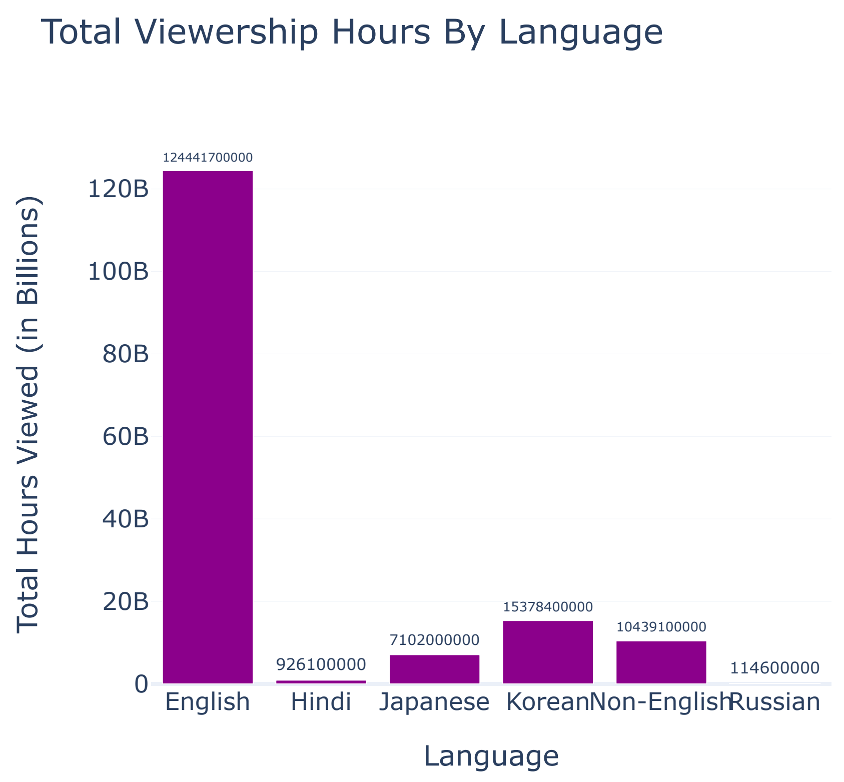
```
#aggregate viewership by content type

content_type_viewership = data.groupby('Content Type')['Hours Viewed'].sum()

fig = go.Figure(data = [
    go.Bar(
    x= content_type_viewership.index,
    y= content_type_viewership.values,
    marker_color=['skyblue', 'salmon'])
])

fig.update_layout(
    title = 'Total Viewership Hours By Content Type',
    xaxis_title = 'Content type',
    yaxis_title = 'Total Hours Viewed (in Billions)',
    xaxis_tickangle = 0,
    height = 450,
    width = 500
)

fig.show()
```



Analyze the distribution of viewership across different languages

In [7]:

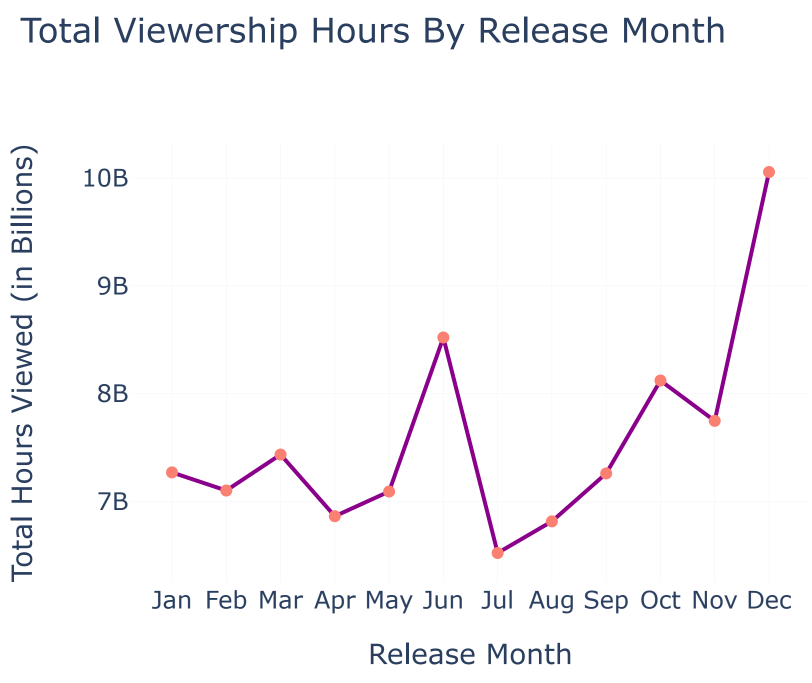
```
language_viewership = data.groupby('Language Indicator')['Hours Viewed'].sum()

fig = go.Figure(data = [
    go.Bar(
    x= language_viewership.index,
    y= language_viewership.values,
    marker_color='darkmagenta',
    text=language_viewership.values,  # Add labels here
    textposition='outside'  # Automatically places the label inside the bars
    )
])

fig.update_layout(
    title = 'Total Viewership Hours By Language',
    xaxis_title = 'Language',
    yaxis_title = 'Total Hours Viewed (in Billions)',
    xaxis_tickangle = 0,
    height = 450,
    width = 500
)

fig.show()
```



Analyze how viewership varies based on release dates to identify any trends over time, such as seasonality or patterns around specific months

In [8]:

```
data['Release Date']= pd.to_datetime(data['Release Date'])
data['Release Month']=data['Release Date'].dt.month

monthly_viewership = data.groupby('Release Month')['Hours Viewed'].sum()

fig = go.Figure(data = [
    go.Scatter(
    x= monthly_viewership.index,
    y= monthly_viewership.values,
    mode = 'lines+markers',
    marker=dict(color ='salmon'),
    line = dict(color = 'darkmagenta')
    )
])

fig.update_layout(
    title = 'Total Viewership Hours By Release Month',
    xaxis_title = 'Release Month',
    yaxis_title = 'Total Hours Viewed (in Billions)',
    xaxis_tickangle = 0,
    xaxis=dict(
        tickmode='array',
        tickvals=list(range(1, 13)),
        ticktext=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
    ),
    height = 400,
    width = 500
)

fig.show()
```



Analyze the most successful content (both shows and movies)

In [9]:

```
#Extract the top 5 Titles Based on Viewership Hours :

top_5_titles = data.nlargest(5,'Hours Viewed')

top_5_titles[['Title','Hours Viewed','Language Indicator', 'Content Type','Release Date']]
```

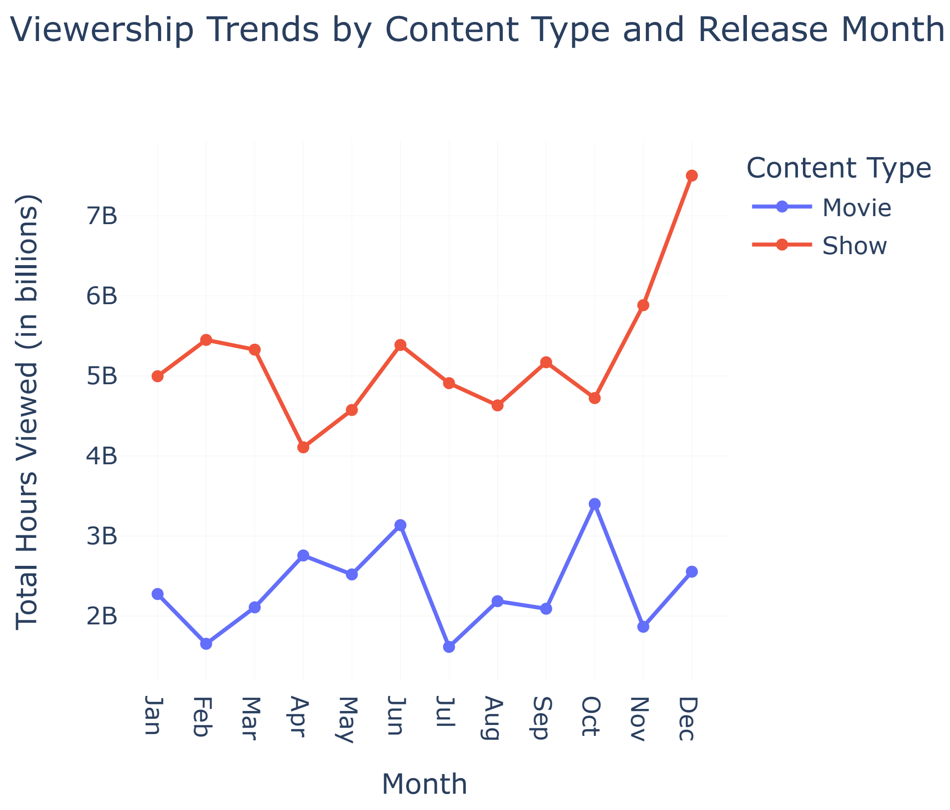
Out[9]:

	Title	Hours Viewed	Language Indicator	Content Type	Release Date
0	The Night Agent: Season 1	812100000.0	English	Show	2023-03-23
1	Ginny & Georgia: Season 2	665100000.0	English	Show	2023-01-05
18227	King the Land: Limited Series // 킹더랜드: 리미티드 시리즈	630200000.0	Korean	Movie	2023-06-17
2	The Glory: Season 1 // 더 글로리: 시즌 1	622800000.0	Korean	Show	2022-12-30
18214	ONE PIECE: Season 1	541900000.0	English	Show	2023-08-31

In [10]:

```
# aggregate viewership hours by content type and release month
monthly_viewership_by_type = data.pivot_table(index='Release Month',
                                               columns='Content Type',
                                               values='Hours Viewed',
                                               aggfunc='sum')

fig = go.Figure()
```

```python
for content_type in monthly_viewership_by_type.columns:
    fig.add_trace(
        go.Scatter(
            x=monthly_viewership_by_type.index,
            y=monthly_viewership_by_type[content_type],
            mode='lines+markers',
            name=content_type
        )
    )

fig.update_layout(
    title='Viewership Trends by Content Type and Release Month (2023)',
    xaxis_title='Month',
    yaxis_title='Total Hours Viewed (in billions)',
    xaxis=dict(
        tickmode='array',
        tickvals=list(range(1, 13)),
        ticktext=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
    ),
    height=450,
    width=500,
    legend_title='Content Type'
)

fig.show()
```

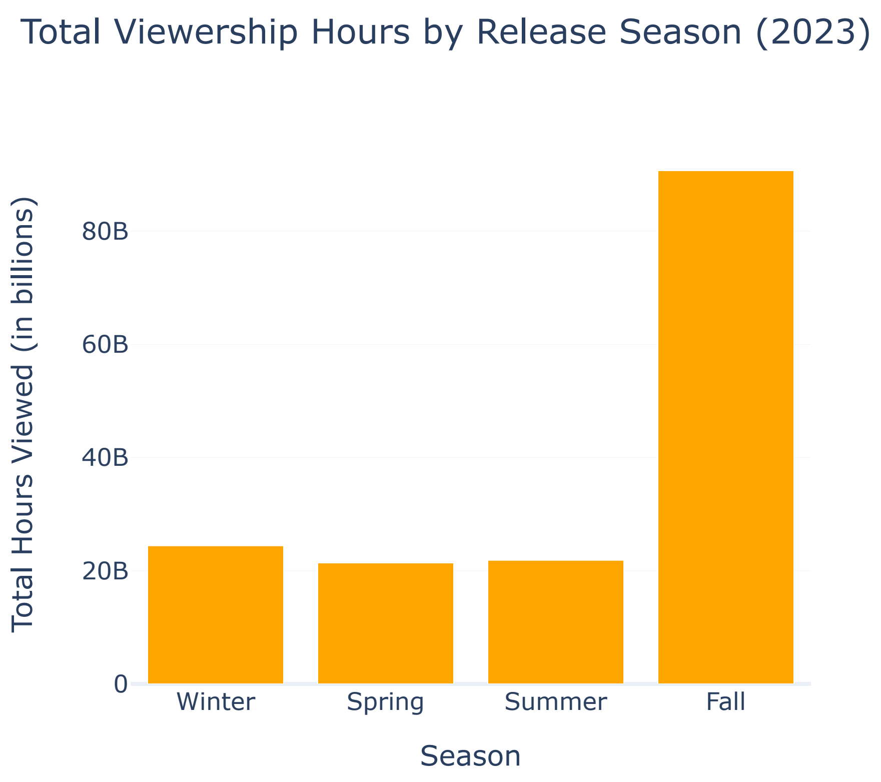


The graph compares viewership trends between movies and shows throughout 2023. It shows that shows consistently have higher viewership than movies, peaking in December. Movies have more fluctuating viewership, with notable increases in June and October. This indicates that Netflix's audience engages more with shows across the year, while movie viewership experiences occasional spikes, possibly linked to specific releases or events.

Analyze the total viewership hours distributed across different release seasons

In [11]:

```
#define Seasons Based on Release Months :

def get_season(month):
    if month in [12,1,2]:
        return 'Winter'
    elif month in [3,4,5]:
        return 'Spring'
    elif month in [6,7,8]:
        return 'Summer'
    else :
        return 'Fall'

#Apply the Season to data set
data['Release Season'] = data['Release Month'].apply(get_season)


# aggregate viewership hours by release season
seasonal_viewership = data.groupby('Release Season')['Hours Viewed'].sum()

# order the seasons as 'Winter', 'Spring', 'Summer', 'Fall'
seasons_order = ['Winter', 'Spring', 'Summer', 'Fall']
seasonal_viewership = seasonal_viewership.reindex(seasons_order)

fig = go.Figure(data=[
    go.Bar(
        x=seasonal_viewership.index,
        y=seasonal_viewership.values,
        marker_color='orange'
    )
])

fig.update_layout(
    title='Total Viewership Hours by Release Season (2023)',
    xaxis_title='Season',
    yaxis_title='Total Hours Viewed (in billions)',
    xaxis_tickangle=0,
    height=450,
    width=500,
    xaxis=dict(
        categoryorder='array',
        categoryarray=seasons_order
    )
)

fig.show()
```

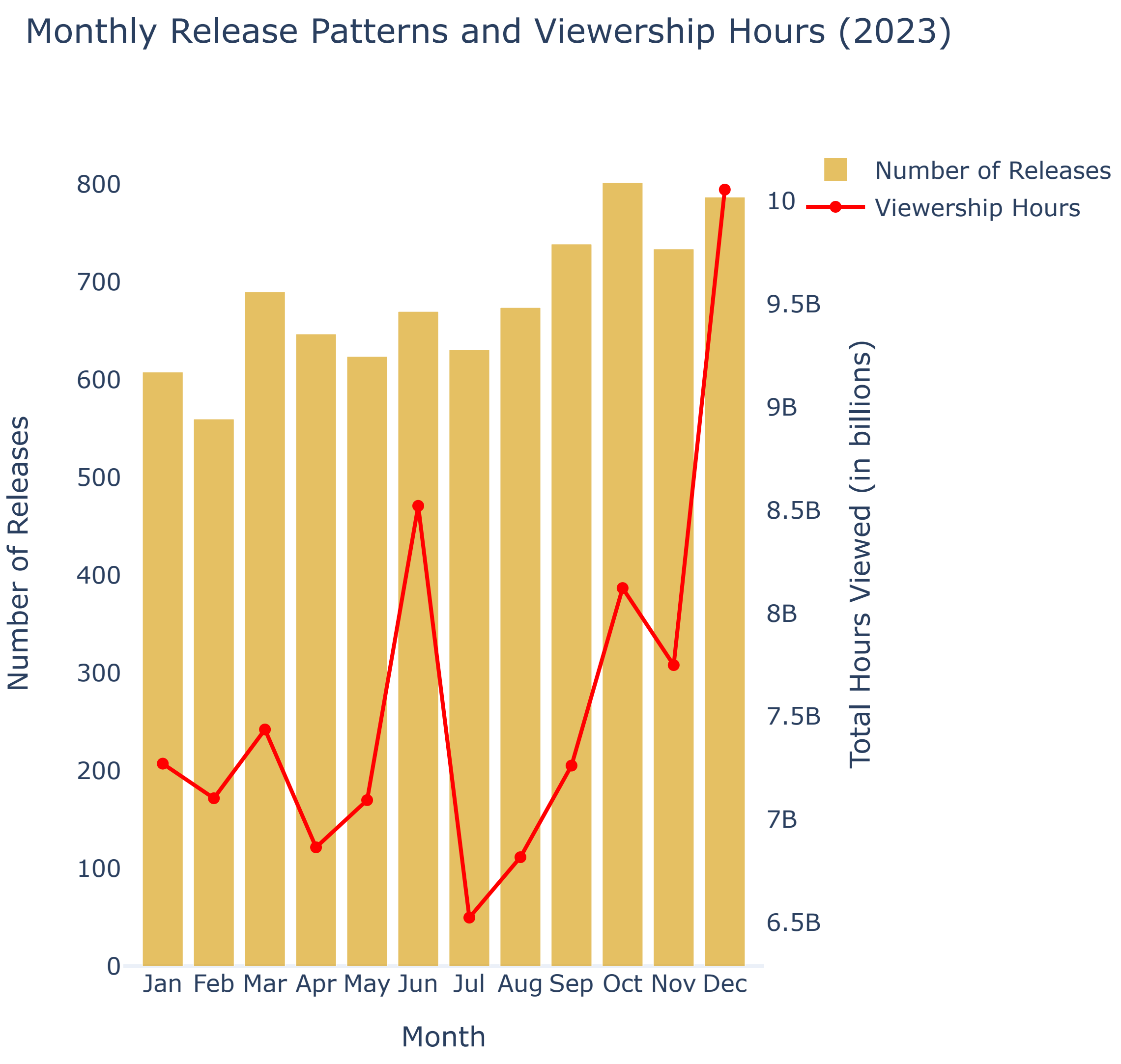


The graph indicates that viewership hours peak significantly in the Fall season, with over 80 billion hours viewed, while Winter, Spring, and Summer each have relatively stable and similar viewership around the 20 billion mark. This suggests that Netflix experiences the highest audience engagement during the Fall.

Analyze the number of content releases and their viewership hours across months

In [12]:

```
monthly_releases = data['Release Month'].value_counts().sort_index()

monthly_viewership = data.groupby('Release Month')['Hours Viewed'].sum()

fig = go.Figure()

fig.add_trace(
    go.Bar(
        x=monthly_releases.index,
        y=monthly_releases.values,
        name='Number of Releases',
        marker_color='goldenrod',
        opacity=0.7,
        yaxis='y1'
    )
)

fig.add_trace(
    go.Scatter(
        x=monthly_viewership.index,
        y=monthly_viewership.values,
        name='Viewership Hours',
        mode='lines+markers',
        marker=dict(color='red'),
        line=dict(color='red'),
        yaxis='y2'
    )
)

fig.update_layout(
    title='Monthly Release Patterns and Viewership Hours (2023)',
    xaxis=dict(
        title='Month',
        tickmode='array',
        tickvals=list(range(1, 13)),
        ticktext=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
    ),
    yaxis=dict(
        title='Number of Releases',
        showgrid=False,
        side='left'
    ),
    yaxis2=dict(
        title='Total Hours Viewed (in billions)',
        overlaying='y',
        side='right',
        showgrid=False
    ),
    legend=dict(
        x=1.05,
        y=1,
        orientation='v',
        xanchor='left'
    ),
    height=600,
    width=600
)

fig.show()
```



While the number of releases is relatively steady throughout the year, viewership hours experience a sharp increase in June and a significant rise in December, despite a stable release count. This indicates that viewership is not solely dependent on the number of releases but influenced by the timing and appeal of specific content during these months.

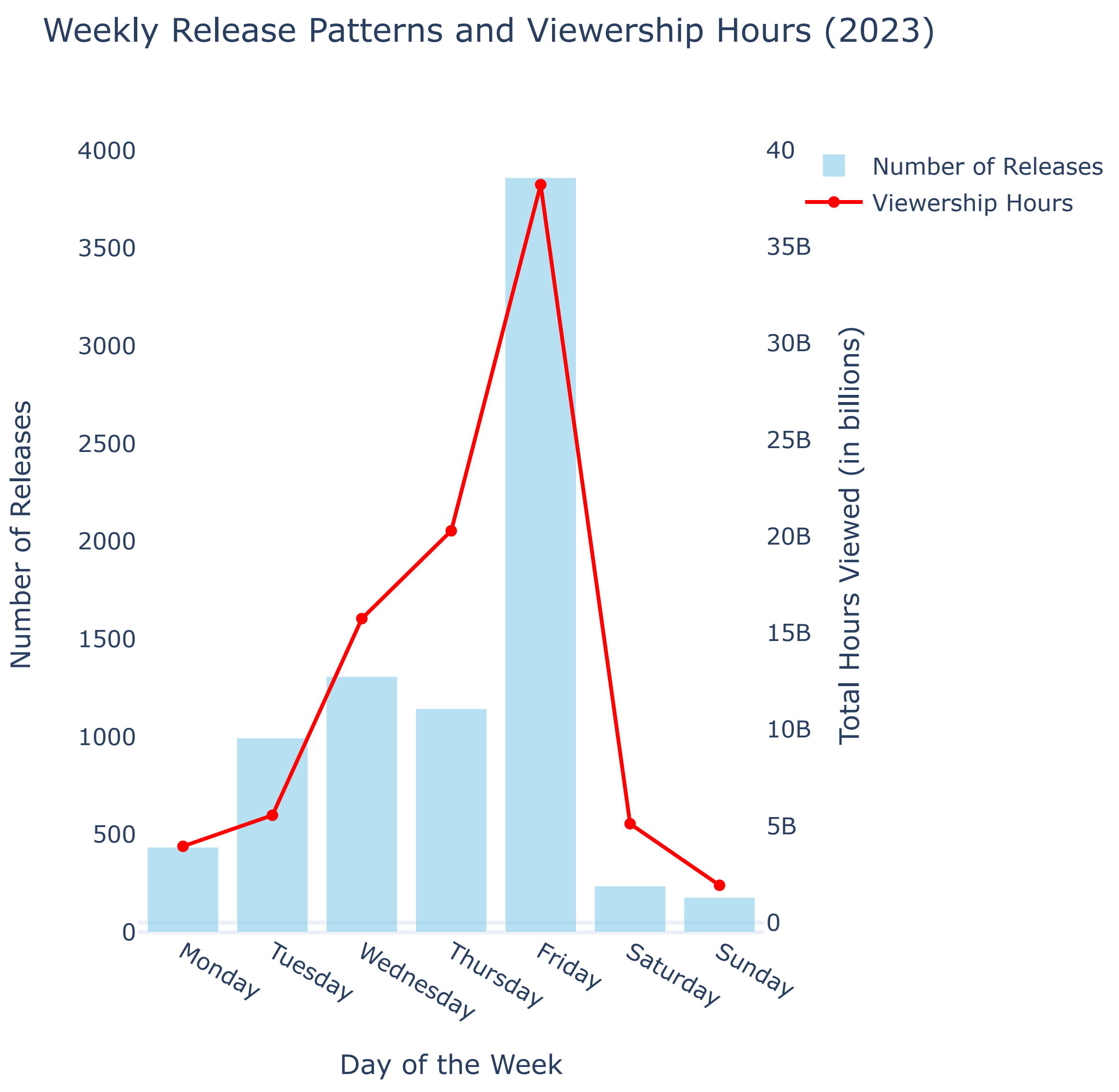
Analyze whether Netflix has a preference for releasing

content on specific weekdays and how this influences viewership patterns

In [13]:

```
data['Release Day'] = data['Release Date'].dt.day_name()

weekday_releases = data['Release Day'].value_counts().reindex(
    ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
)

# aggregate viewership hours by day of the week
weekday_viewership = data.groupby('Release Day')['Hours Viewed'].sum().reindex(
    ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
)
```

```
fig = go.Figure()

fig.add_trace(
    go.Bar(
        x=weekday_releases.index,
        y=weekday_releases.values,
        name='Number of Releases',
        marker_color='skyblue',
        opacity=0.6,
        yaxis='y1'
    )
)

fig.add_trace(
    go.Scatter(
        x=weekday_viewership.index,
        y=weekday_viewership.values,
        name='Viewership Hours',
        mode='lines+markers',
        marker=dict(color='red'),
        line=dict(color='red'),
        yaxis='y2'
    )
)

fig.update_layout(
    title='Weekly Release Patterns and Viewership Hours (2023)',
    xaxis=dict(
        title='Day of the Week',
        categoryorder='array',
        categoryarray=['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
    ),
    yaxis=dict(
        title='Number of Releases',
        showgrid=False,
        side='left'
    ),
    yaxis2=dict(
        title='Total Hours Viewed (in billions)',
        overlaying='y',
        side='right',
        showgrid=False
    ),
    legend=dict(
        x=1.05,
        y=1,
        orientation='v',
        xanchor='left'
    ),
    height=600,
    width=600
)

fig.show()
```



The graph highlights that most content releases occur on Fridays, with viewership hours also peaking significantly on that day. This suggests that Netflix strategically releases content toward the weekend to maximize audience engagement.

Analyze specific high-impact dates, such as holidays or major events, and their correlation with content releases

In [14]:

```
# define significant holidays and events in 2023
important_dates = [
    '2023-01-01',  # new year's day
    '2023-02-14',  # valentine's ay
    '2023-07-04',  # independence day (US)
    '2023-10-31',  # halloween
    '2023-12-25'   # christmas day
]

# convert to datetime
important_dates = pd.to_datetime(important_dates)
```

```
# check for content releases close to these significant holidays (within a 3-day window)
holiday_releases = data[data['Release Date'].apply(
    lambda x: any((x - date).days in range(-3, 4) for date in important_dates)
)]

# aggregate viewership hours for releases near significant holidays
holiday_viewership = holiday_releases.groupby('Release Date')['Hours Viewed'].sum()

holiday_releases[['Title', 'Release Date', 'Hours Viewed']]
```

Out[14]:

	Title	Release Date	Hours Viewed
2	The Glory: Season 1 // 더 글로리: 시즌 1	2022-12-30	622800000.0
6	La Reina del Sur: Season 3	2022-12-30	429600000.0
11	Kaleidoscope: Limited Series	2023-01-01	252500000.0
29	Perfect Match: Season 1	2023-02-14	176800000.0
124	Lady Voyeur: Limited Series // Olhar Indiscret...	2022-12-31	86000000.0
...	...	...	...
22324	The Romantics: Limited Series	2023-02-14	1000000.0
22327	Aggretsuko: Season 5 // アグレッシブ : シーズン5	2023-02-16	900000.0
22966	The Lying Life of Adults: Limited Series // La...	2023-01-04	900000.0
22985	Community Squad: Season 1 // División Palermo:...	2023-02-17	800000.0
24187	Live to Lead: Limited Series	2022-12-31	400000.0

98 rows × 3 columns

The data reveals that Netflix has strategically released content around key holidays and events. Some of the significant releases include: New Year's Period and Valentine's Day