DSC640_KristaKnuckey_Week1&2

April 5, 2025

[1]:
```
#Krista Knuckey
#Netflix Viwership
```

[2]:
```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[3]:
```
#import Netflix datasets
```

[4]:
```
global_data = pd.read_csv('all-weeks-global-netflix.csv')
global_data.head()
```

[4]:

	week	category	weekly_rank	show_title	season_title	weekly_hours_viewed	runtime	weekly_views	cumulative_weeks_in_top_10	is_staggered_launch	episode_launch_details
0	2024-04-14	Films (English)	1	What Jennifer Did	NaN	26100000	1.4500	18000000.0	1	False	NaN
1	2024-04-14	Films (English)	2	Woody Woodpecker Goes to Camp	NaN	19600000	1.6667	11800000.0	1	False	NaN
2	2024-04-14	Films (English)	3	Scoop	NaN	14600000	1.7167	8500000.0	2	False	NaN
3	2024-04-14	Films (English)	4	Glass	NaN	11000000	2.1500	5100000.0	2	False	NaN
4	2024-04-14	Films (English)	5	Megan Leavey	NaN	9700000	1.9333	5000000.0	1	False	NaN

[5]:
```
countries_data = pd.read_csv('all-weeks-countries-netflix.csv')
countries_data.head()
```

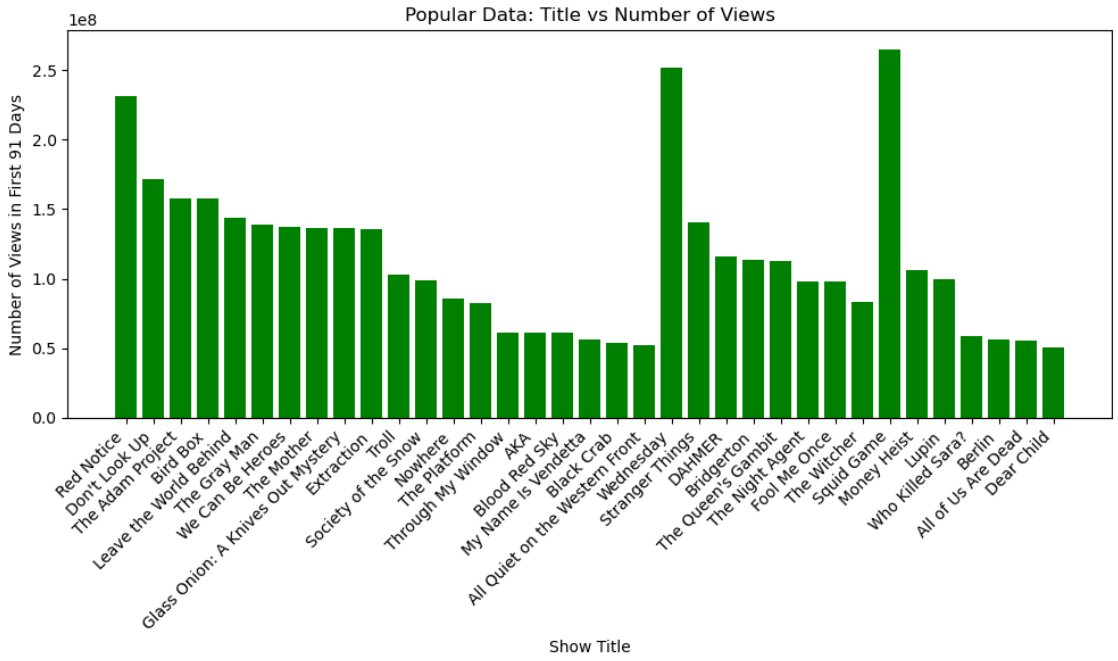
[5]:
```
  country_name country_iso2        week category  weekly_rank  \
0    Argentina           AR  2024-04-14    Films            1
1    Argentina           AR  2024-04-14    Films            2
2    Argentina           AR  2024-04-14    Films            3
3    Argentina           AR  2024-04-14    Films            4
4    Argentina           AR  2024-04-14    Films            5

                     show_title season_title  cumulative_weeks_in_top_10
0                 The Tearsmith          NaN                           2
1                        Stolen          NaN                           1
2                 Love, Divided          NaN                           1
3  Woody Woodpecker Goes to Camp          NaN                           1
4                 Rest In Peace          NaN                           3
```

[6]:
```
popular_data = pd.read_csv('most-popular-netflix.csv')
popular_data.head()
```

[6]:
```
          category  rank              show_title season_title  \
0  Films (English)     1              Red Notice          NaN
1  Films (English)     2          Don't Look Up          NaN
2  Films (English)     3       The Adam Project          NaN
3  Films (English)     4              Bird Box          NaN
4  Films (English)     5  Leave the World Behind          NaN

   hours_viewed_first_91_days  runtime  views_first_91_days
0                   454200000   1.9667            230900000
1                   408600000   2.3833            171400000
2                   281000000   1.7833            157600000
3                   325300000   2.0667            157400000
4                   339300000   2.3667            143400000
```

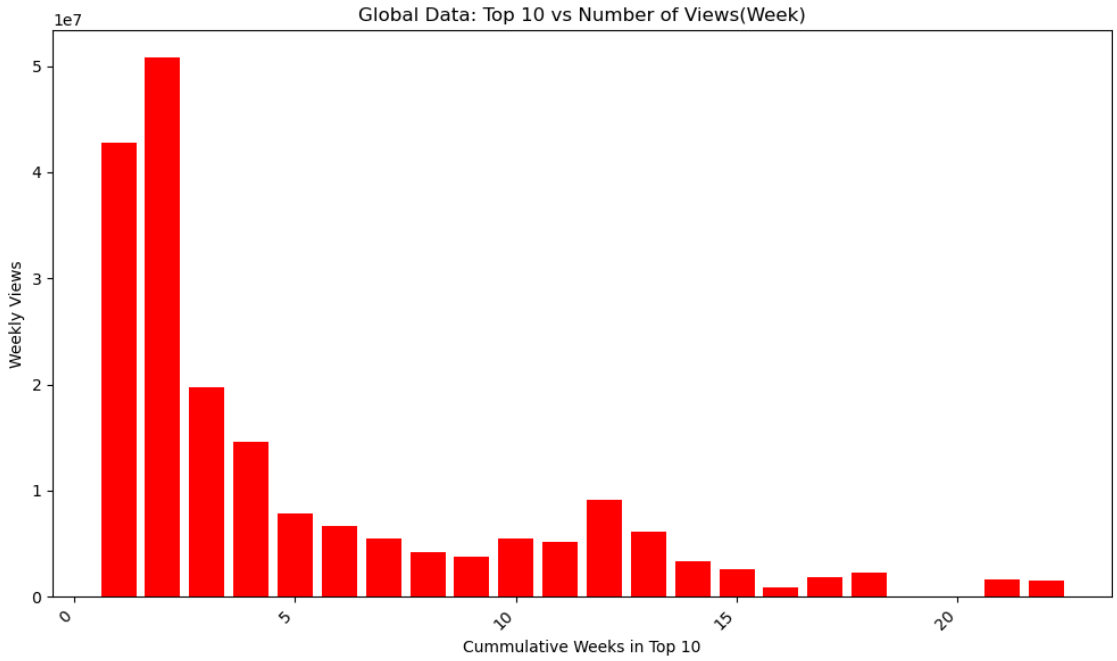
[7]:
```
#Exploratory Analysis- creating visuals to guide the story for explanatory␣
 ↪analysis
```

[8]:
```
plt.figure(figsize=(10,6))
plt.bar(popular_data['show_title'], popular_data['views_first_91_days'],␣
 ↪color='green')

plt.title('Popular Data: Title vs Number of Views')
plt.xlabel('Show Title')
plt.ylabel('Number of Views in First 91 Days')
plt.xticks(rotation=45, ha='right')

plt.tight_layout()
plt.show()
```



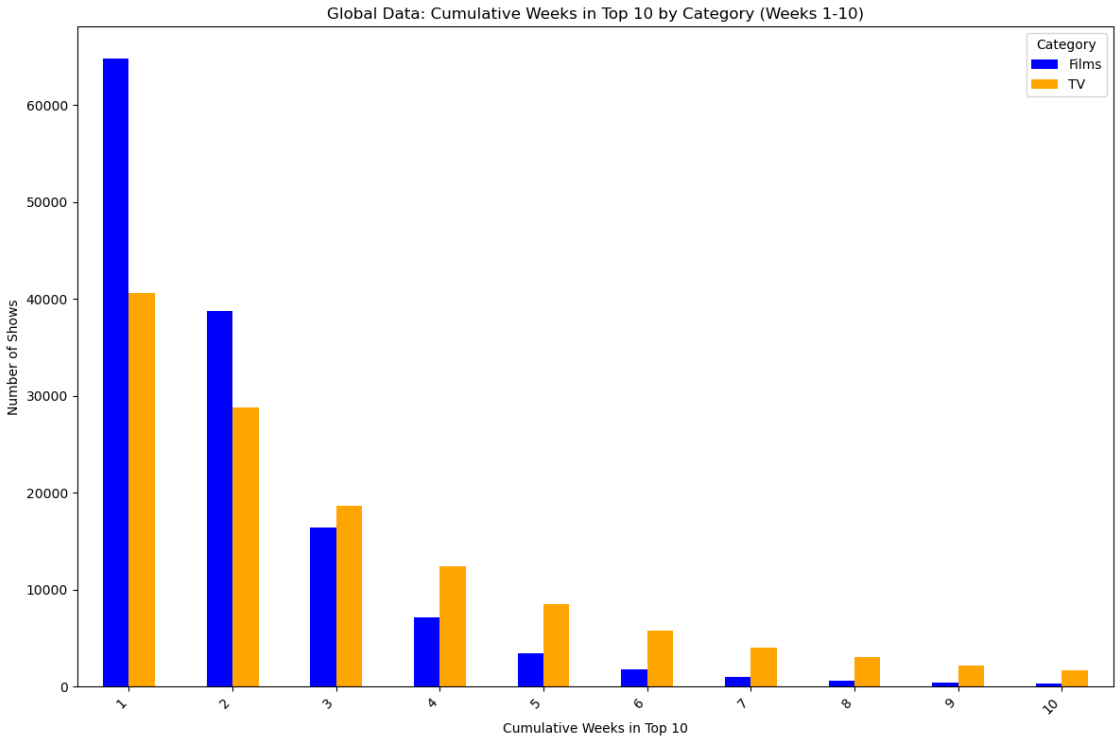
[9]:
```
#Observation 1: Top 2 titles "Wednesday" "Red Notice" "Squid Game" are in the␣
↪thriller/ horror genre
```

[10]:
```
plt.figure(figsize=(10,6))
plt.bar(global_data['cumulative_weeks_in_top_10'], global_data['weekly_views'],␣
↪color='red')

plt.title('Global Data: Top 10 vs Number of Views(Week)')
plt.xlabel('Cummulative Weeks in Top 10')
plt.ylabel('Weekly Views')
plt.xticks(rotation=45, ha='right')

plt.tight_layout()
plt.show()
```



[11]:
```
#Observation: Titles that are in the top ten for the longest have the lowest␣
↪weekly views.
```

[12]:
```
filtered_data = countries_data[countries_data['cumulative_weeks_in_top_10'].
↪between(1, 10)]

grouped_filtered_data = filtered_data.groupby(['cumulative_weeks_in_top_10',␣
↪'category']).size().unstack().fillna(0)

grouped_filtered_data.plot(kind='bar', figsize=(12, 8), color=['blue',␣
↪'orange'])

plt.title('Global Data: Cumulative Weeks in Top 10 by Category (Weeks 1-10)')
plt.xlabel('Cumulative Weeks in Top 10')
plt.ylabel('Number of Shows')

plt.xticks(ticks=range(10), labels=range(1, 11), rotation=45, ha='right')

plt.legend(title='Category')
plt.tight_layout()
plt.show()
```



[13]:
```
#Observation: It is interesting to see that films are positively skeweed and␣
↪slowly tv shows become more popular.
```

[14]:
```
#Explanatory Analysis- Financial Analyst proposing a boost in horror/ thriller␣
↪TV production to gain new subscribers and more revenue
```

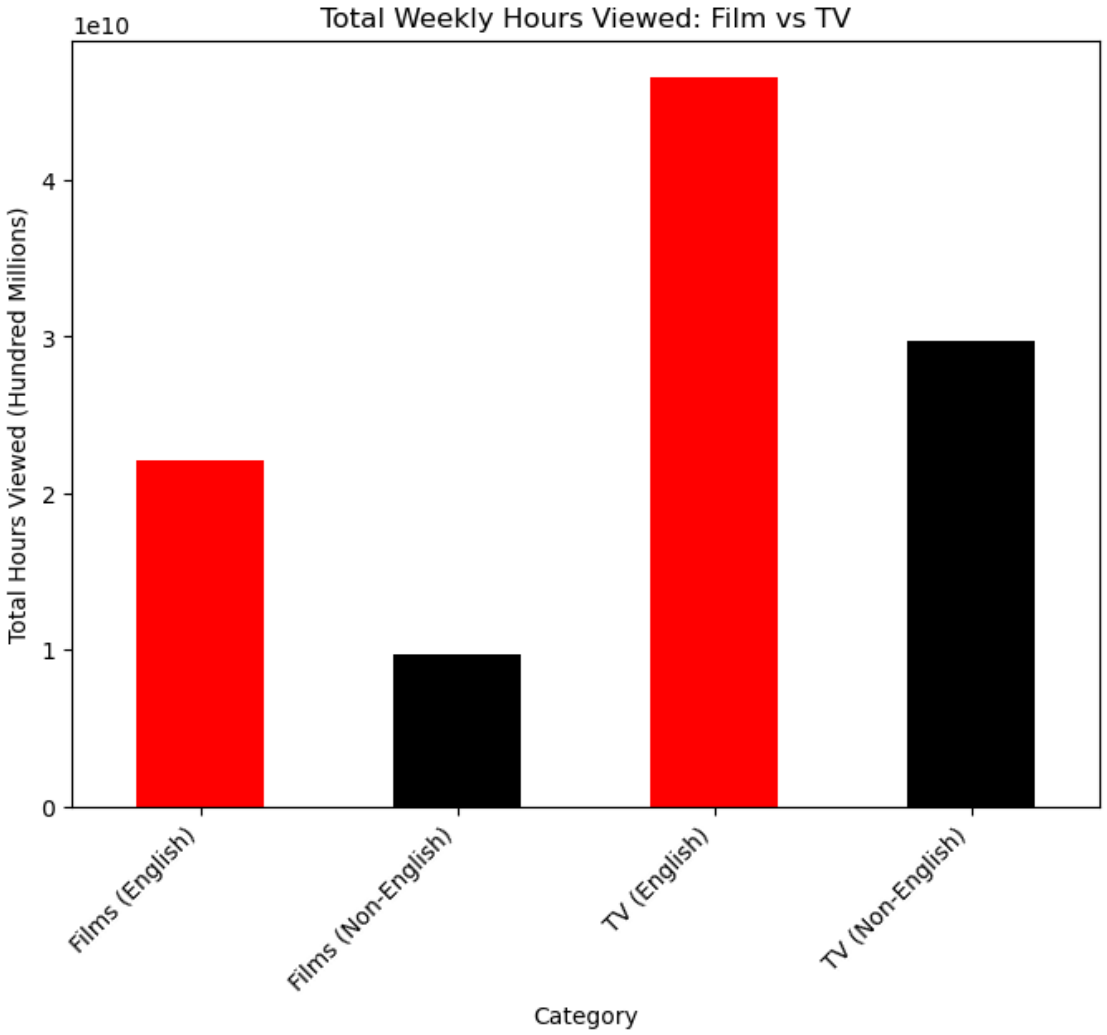
[15]:
```
#Visualization 1: Simple text created in Canva to emphasize the amount of views␣
↪"Wednesday" and "Squid Games" had (See powerpoint presentation for visual 1)
```

[16]:
```
#Visualization 2- another simple text too emphasize the amount of time␣
↪customers are using- this will be a great talking point during the␣
↪presentation.
#Both visualizations I created in Canva
```

[17]:
```
#Visualization 3: bar graph to compare English vs. non English Tv/ Film␣
↪popularity
```

[18]:
```
category_hours = global_data.groupby('category')['weekly_hours_viewed'].sum()

plt.figure(figsize=(8,6))
category_hours.plot(kind='bar', color=['red', 'black'])

plt.title('Total Weekly Hours Viewed: Film vs TV')
```

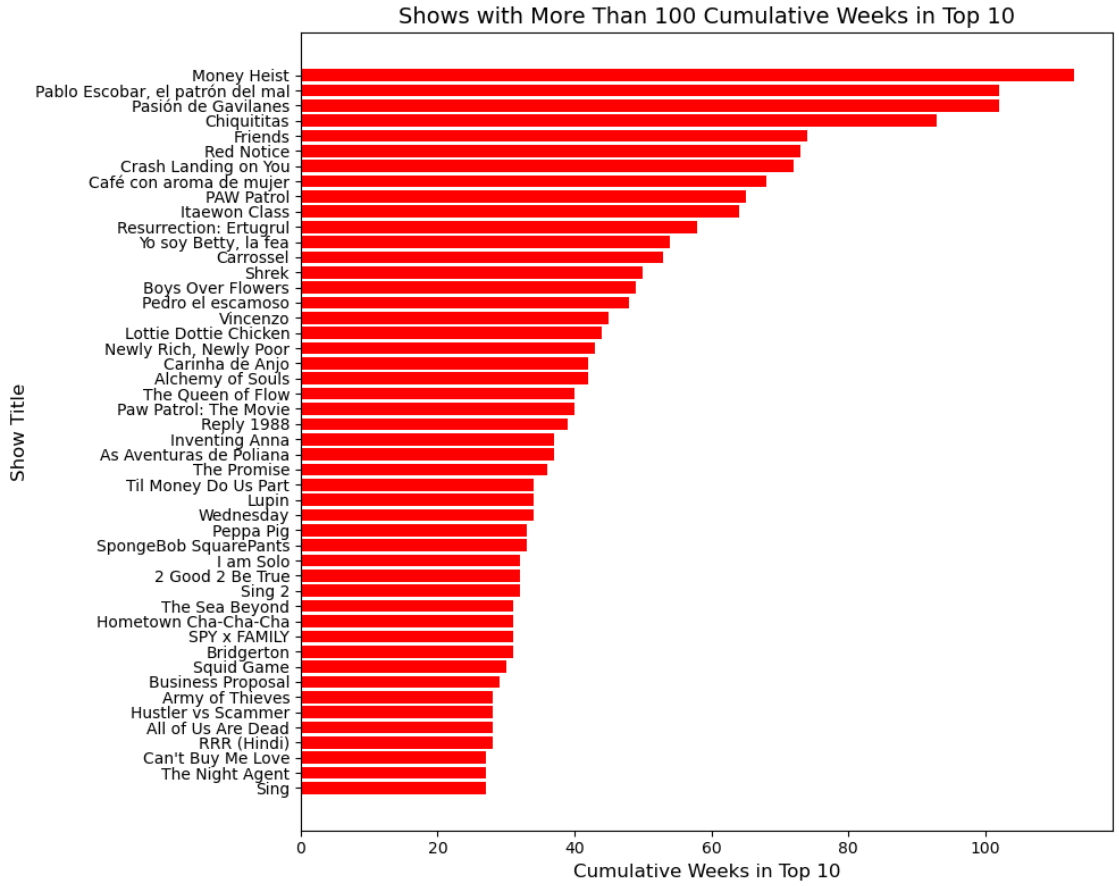
```
plt.ylabel('Total Hours Viewed (Hundred Millions)')
plt.xlabel('Category')
plt.xticks(rotation=45, ha='right')

plt.show()
```



[19]:
```
#Visualization 4: horizontal bar chart to find the top 10 at over 26 weeks- as this is half the year we can see that the content captivates the audience.
```

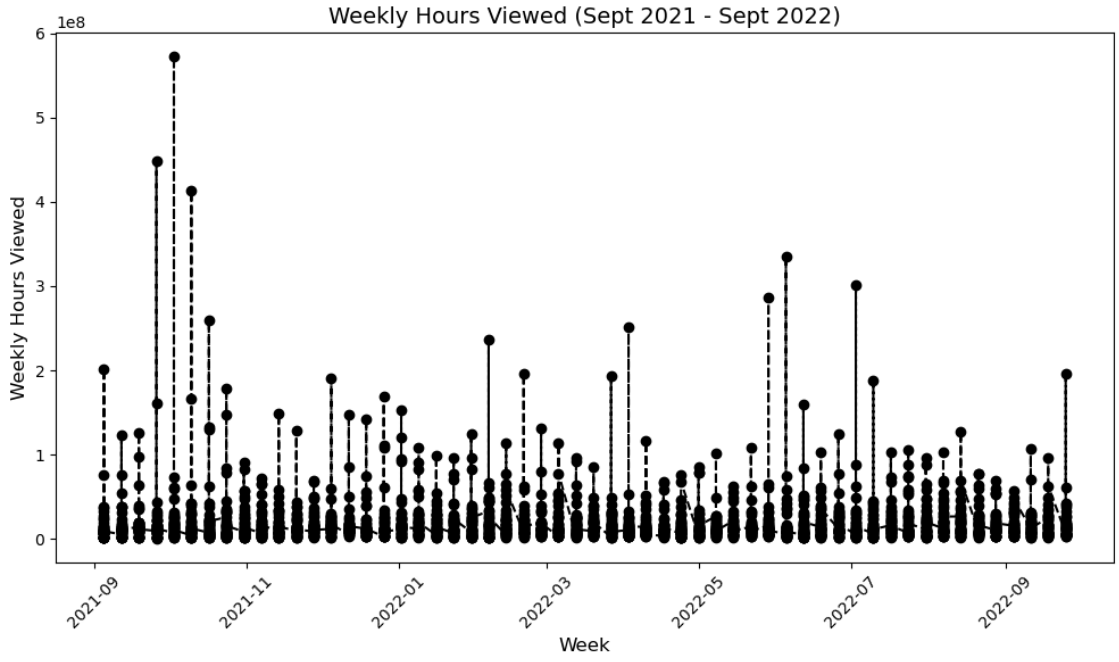
[20]:
```
countries_data['cumulative_weeks_in_top_10'] = pd.to_numeric(countries_data['cumulative_weeks_in_top_10'], errors='coerce')

df_filtered = countries_data[countries_data['cumulative_weeks_in_top_10'] > 26]
```

```
df_filtered = df_filtered.sort_values(by='cumulative_weeks_in_top_10',␣
 ↪ascending=False)

plt.figure(figsize=(10, 8))
plt.barh(df_filtered['show_title'], df_filtered['cumulative_weeks_in_top_10'],␣
 ↪color='red')

plt.title('Shows with More Than 100 Cumulative Weeks in Top 10', fontsize=14)
plt.xlabel('Cumulative Weeks in Top 10', fontsize=12)
plt.ylabel('Show Title', fontsize=12)
plt.gca().invert_yaxis()

plt.tight_layout()
plt.show()
```

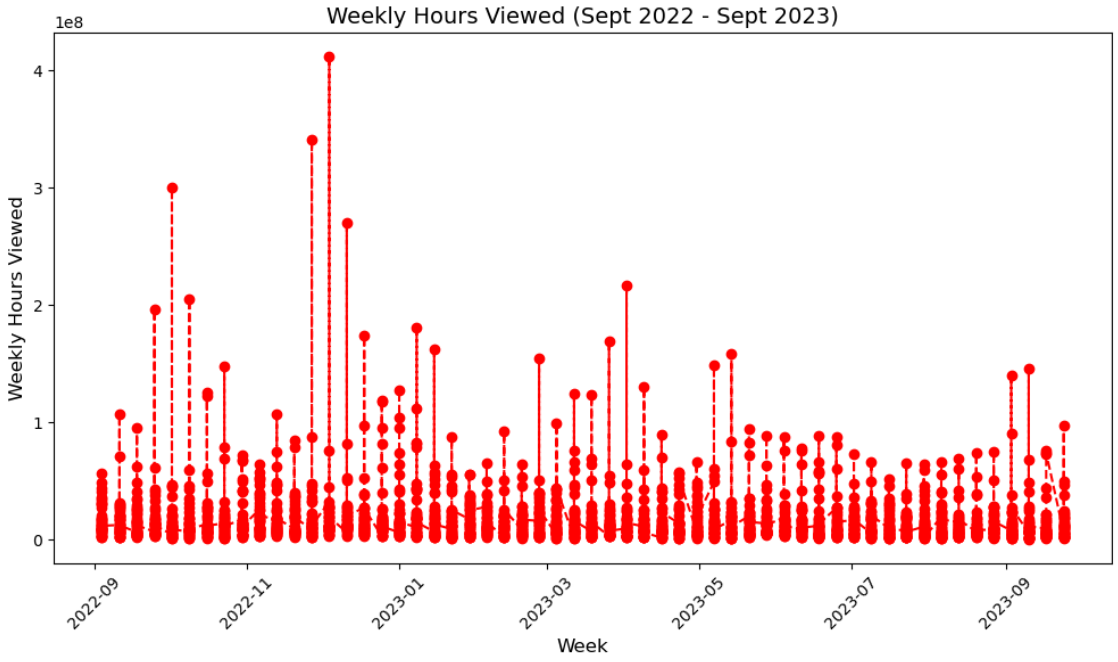


[21]:
```
#Visulization 5: Line chart filtered for one fiscal year to see the highs/ lows␣
 ↪of viewing. We can see that this relates back to the top 10 because in␣
 ↪September 2021 "Squid Games" was released
```

```
global_data['week'] = pd.to_datetime(global_data['week'])
start_date = '2021-09-01'
end_date = '2022-09-30'
filtered_data = global_data[(global_data['week'] >= start_date) &␣
 ↪(global_data['week'] <= end_date)]

filtered_data = filtered_data.sort_values(by='week')


plt.figure(figsize=(10, 6))
plt.plot(filtered_data['week'], filtered_data['weekly_hours_viewed'],␣
 ↪marker='o', color='black', linestyle='--')


plt.title('Weekly Hours Viewed (Sept 2021 - Sept 2022)', fontsize=14)
plt.xlabel('Week', fontsize=12)
plt.ylabel('Weekly Hours Viewed', fontsize=12)

plt.xticks(rotation=45)


plt.tight_layout()
plt.show()
```



[22]:
```
#Visualization 6: Line chart of a different fiscal year to have consistency,␣
 ↪but be able to show stakeholders consistent peaks/ lows within content. This␣
 ↪also matches the previous top 10 since "Wednesday" was relased in November␣
 ↪2022

global_data['week'] = pd.to_datetime(global_data['week'])
start_date = '2022-09-01'
end_date = '2023-09-30'
filtered_data = global_data[(global_data['week'] >= start_date) &␣
 ↪(global_data['week'] <= end_date)]

filtered_data = filtered_data.sort_values(by='week')

plt.figure(figsize=(10, 6))
plt.plot(filtered_data['week'], filtered_data['weekly_hours_viewed'],␣
 ↪marker='o', color='red', linestyle='--')


plt.title('Weekly Hours Viewed (Sept 2022 - Sept 2023)', fontsize=14)
plt.xlabel('Week', fontsize=12)
plt.ylabel('Weekly Hours Viewed', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



[23]:
```
#Analysis- After discovering peak times I analyzed to see what countries had␣
 ↪the most viewership. I filtered for the peaks months, and grouped country␣
 ↪with the weekly hours viewed to discover Argentina had the most viewership

countries_data['week'] = pd.to_datetime(countries_data['week'])
global_data['week'] = pd.to_datetime(global_data['week'])

merged_data = pd.merge(global_data, countries_data, on='week', how='inner')
september_2021_data = merged_data[(merged_data['week'] >= '2021-09-01') &␣
 ↪(merged_data['week'] <= '2021-09-30')]
november_2022_data = merged_data[(merged_data['week'] >= '2022-11-01') &␣
 ↪(merged_data['week'] <= '2022-11-30')]

september_2021_sum = september_2021_data.
 ↪groupby('country_name')['weekly_hours_viewed'].sum().reset_index()
november_2022_sum = november_2022_data.
 ↪groupby('country_name')['weekly_hours_viewed'].sum().reset_index()

top_country_september_2021 = september_2021_sum.
 ↪loc[september_2021_sum['weekly_hours_viewed'].idxmax()]
top_country_november_2022 = november_2022_sum.
 ↪loc[november_2022_sum['weekly_hours_viewed'].idxmax()]


print("Country that watched the most in September 2021:
 ↪",top_country_september_2021)
print("Country that watched the most in November 2022:
 ↪",top_country_november_2022)
```

```
Country that watched the most in September 2021: country_name
Argentina
weekly_hours_viewed    65997200000
Name: 0, dtype: object
Country that watched the most in November 2022: country_name
Argentina
weekly_hours_viewed    69435000000
Name: 0, dtype: object
```

[24]:
```
#Visualization 7: Pie chart to compare other countries to Argentina, which is␣
 ↪impactful based on the population of Argentina vs viewership

argentina_sept = september_2021_sum[september_2021_sum['country_name'] ==␣
 ↪'Argentina']['weekly_hours_viewed'].values[0]
other_countries_sept = september_2021_sum[september_2021_sum['country_name'] !=␣
 ↪'Argentina']['weekly_hours_viewed'].sum()
```

```
argentina_nov = november_2022_sum[november_2022_sum['country_name'] ==␣
 ↪'Argentina']['weekly_hours_viewed'].values[0]
other_countries_nov = november_2022_sum[november_2022_sum['country_name'] !=␣
 ↪'Argentina']['weekly_hours_viewed'].sum()
labels = ['Argentina', 'Other Countries']

fig, axes = plt.subplots(1, 2, figsize=(12, 6))

axes[0].pie([argentina_sept, other_countries_sept], labels=labels, autopct='%1.
 ↪1f%%', colors=['red', 'lightblue'], startangle=90)
axes[0].set_title('September 2021: Argentina vs Other Countries')

axes[1].pie([argentina_nov, other_countries_nov], labels=labels, autopct='%1.
 ↪1f%%', colors=['green', 'lightgray'], startangle=90)
axes[1].set_title('November 2022: Argentina vs Other Countries')

plt.tight_layout()
plt.show()
```