JacobsonDSC640Weeks1-2Exercise

June 15, 2025

[21]:
```
import pandas as pd
nation = pd.read_excel('all-weeks-countries-netflix.xlsx')
world = pd.read_excel('all-weeks-global-netflix.xlsx')
popular = pd.read_excel('most-popular-netflix.xlsx')
```

```
C:\Users\15027\anaconda3\Lib\site-packages\openpyxl\styles\stylesheet.py:237:
UserWarning: Workbook contains no default style, apply openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
C:\Users\15027\anaconda3\Lib\site-packages\openpyxl\styles\stylesheet.py:237:
UserWarning: Workbook contains no default style, apply openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
C:\Users\15027\anaconda3\Lib\site-packages\openpyxl\styles\stylesheet.py:237:
UserWarning: Workbook contains no default style, apply openpyxl's default
  warn("Workbook contains no default style, apply openpyxl's default")
```

[92]:
```
nation['title']= nation['season_title'].combine_first(nation['show_title'])
world['title']= world['season_title'].combine_first(world['show_title'])
popular['title']= popular['season_title'].combine_first(popular['show_title'])
```

[6]:
```
print(nation.shape)
nation.head()
```

```
(272260, 8)
```

[6]:

	country_name	country_iso2	week	category	weekly_rank	show_title	season_title	cumulative_weeks_in_top_10
0	Argentina	AR	2024-04-14	Films	1	The Tearsmith	NaN	2
1	Argentina	AR	2024-04-14	Films	2	Stolen	NaN	1
2	Argentina	AR	2024-04-14	Films	3	Love, Divided	NaN	1
3	Argentina	AR	2024-04-14	Films	4	Woody Woodpecker Goes to Camp	NaN	1
4	Argentina	AR	2024-04-14	Films	5	Rest In Peace	NaN	3

[5]:
```
print(world.shape)
world.head()
```

```
(5840, 11)
```

[5]:

	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

[7]:
```
print(popular.shape)
popular.head()
```

```
(40, 7)
```

[7]:

	category	rank	show_title	season_title	hours_viewed_first_91_days	runtime	views_first_91_days
0	Films (English)	1	Red Notice	NaN	454200000	1.9667	230900000
1	Films (English)	2	Don't Look Up	NaN	408600000	2.3833	171400000
2	Films (English)	3	The Adam Project	NaN	281000000	1.7833	157600000
3	Films (English)	4	Bird Box	NaN	325300000	2.0667	157400000
4	Films (English)	5	Leave the World Behind	NaN	339300000	2.3667	143400000

[8]:
```
red_world = world[world['show_title']=='Red Notice']
print(red_world.shape)
```

```
(14, 11)
```

[9]:
```
red_world
```

[9]:

	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
3569	2022-07-31	Films (English)	10	Red Notice	NaN	4090000	NaN	NaN	14	False	NaN
4449	2022-02-27	Films (English)	10	Red Notice	NaN	4970000	NaN	NaN	13	False	NaN
4605	2022-01-30	Films (English)	6	Red Notice	NaN	6540000	NaN	NaN	12	False	NaN
4647	2022-01-23	Films (English)	8	Red Notice	NaN	7310000	NaN	NaN	11	False	NaN
4685	2022-01-16	Films (English)	6	Red Notice	NaN	8710000	NaN	NaN	10	False	NaN
4725	2022-01-09	Films (English)	6	Red Notice	NaN	11090000	NaN	NaN	9	False	NaN
4764	2022-01-02	Films (English)	5	Red Notice	NaN	14540000	NaN	NaN	8	False	NaN
4806	2021-12-26	Films (English)	7	Red Notice	NaN	12140000	NaN	NaN	7	False	NaN
4843	2021-12-19	Films (English)	4	Red Notice	NaN	12710000	NaN	NaN	6	False	NaN
4881	2021-12-12	Films (English)	2	Red Notice	NaN	18010000	NaN	NaN	5	False	NaN
4922	2021-12-05	Films (English)	3	Red Notice	NaN	25400000	NaN	NaN	4	False	NaN
4960	2021-11-28	Films (English)	1	Red Notice	NaN	50650000	NaN	NaN	3	False	NaN
5000	2021-11-21	Films (English)	1	Red Notice	NaN	129110000	NaN	NaN	2	False	NaN
5040	2021-11-14	Films (English)	1	Red Notice	NaN	148720000	NaN	NaN	1	False	NaN

```
5000                False                    NaN
5040                False                    NaN
```

[10]:
```
red_world['weekly_hours_viewed'].sum()
```

[10]:
```
453990000
```

[19]:
```
red_nation = nation[(nation['show_title']=='Red Notice') &␣
 ↪(nation['week']=='2022-07-31')]
print(red_nation.shape)
```

```
(10, 8)
```

[20]:
```
red_nation.head(100)
```

[20]:

	country_name	country_iso2	week	category	weekly_rank	\
66026	Estonia	EE	2022-07-31	Films	7	
74789	Germany	DE	2022-07-31	Films	10	
92306	Hungary	HU	2022-07-31	Films	7	
127348	Latvia	LV	2022-07-31	Films	9	
156548	Morocco	MA	2022-07-31	Films	9	
212727	Serbia	RS	2022-07-31	Films	8	
218569	Slovakia	SK	2022-07-31	Films	10	
239008	Switzerland	CH	2022-07-31	Films	9	
253605	Ukraine	UA	2022-07-31	Films	6	
271127	Vietnam	VN	2022-07-31	Films	8	

	show_title	season_title	cumulative_weeks_in_top_10
66026	Red Notice	NaN	19
74789	Red Notice	NaN	9
92306	Red Notice	NaN	18
127348	Red Notice	NaN	16
156548	Red Notice	NaN	15
212727	Red Notice	NaN	22
218569	Red Notice	NaN	13
239008	Red Notice	NaN	10
253605	Red Notice	NaN	36
271127	Red Notice	NaN	22

[23]:
```
red_us = nation[(nation['show_title']=='Red Notice') &␣
 ↪(nation['country_name']=='Vietnam')]
print(red_us.shape)
red_us.head(100)
```

```
(22, 8)
```

[23]:

	country_name	country_iso2	week	category	weekly_rank	\
271127	Vietnam	VN	2022-07-31	Films	8	

271148	Vietnam	VN	2022-07-24	Films	9
271207	Vietnam	VN	2022-07-03	Films	8
271489	Vietnam	VN	2022-03-27	Films	10
271505	Vietnam	VN	2022-03-20	Films	6
271525	Vietnam	VN	2022-03-13	Films	6
271567	Vietnam	VN	2022-02-27	Films	8
271587	Vietnam	VN	2022-02-20	Films	8
271604	Vietnam	VN	2022-02-13	Films	5
271623	Vietnam	VN	2022-02-06	Films	4
271640	Vietnam	VN	2022-01-30	Films	1
271664	Vietnam	VN	2022-01-23	Films	5
271683	Vietnam	VN	2022-01-16	Films	4
271705	Vietnam	VN	2022-01-09	Films	6
271725	Vietnam	VN	2022-01-02	Films	6
271745	Vietnam	VN	2021-12-26	Films	6
271763	Vietnam	VN	2021-12-19	Films	4
271782	Vietnam	VN	2021-12-12	Films	3
271800	Vietnam	VN	2021-12-05	Films	1
271820	Vietnam	VN	2021-11-28	Films	1
271840	Vietnam	VN	2021-11-21	Films	1
271860	Vietnam	VN	2021-11-14	Films	1

	show_title	season_title	cumulative_weeks_in_top_10
271127	Red Notice	NaN	22
271148	Red Notice	NaN	21
271207	Red Notice	NaN	20
271489	Red Notice	NaN	19
271505	Red Notice	NaN	18
271525	Red Notice	NaN	17
271567	Red Notice	NaN	16
271587	Red Notice	NaN	15
271604	Red Notice	NaN	14
271623	Red Notice	NaN	13
271640	Red Notice	NaN	12
271664	Red Notice	NaN	11
271683	Red Notice	NaN	10
271705	Red Notice	NaN	9
271725	Red Notice	NaN	8
271745	Red Notice	NaN	7
271763	Red Notice	NaN	6
271782	Red Notice	NaN	5
271800	Red Notice	NaN	4
271820	Red Notice	NaN	3
271840	Red Notice	NaN	2
271860	Red Notice	NaN	1

[24]:
```
red_us = nation[(nation['show_title']=='Red Notice') &␣
 ↪(nation['country_name']=='United States')]
print(red_us.shape)
red_us.head(100)
```

```
(8, 8)
```

[24]:

	country_name	country_iso2	week	category	weekly_rank	\
262966	United States	US	2022-01-02	Films	7	
262987	United States	US	2021-12-26	Films	8	
263006	United States	US	2021-12-19	Films	7	
263023	United States	US	2021-12-12	Films	4	
263043	United States	US	2021-12-05	Films	4	
263061	United States	US	2021-11-28	Films	2	
263080	United States	US	2021-11-21	Films	1	
263100	United States	US	2021-11-14	Films	1	

	show_title	season_title	cumulative_weeks_in_top_10
262966	Red Notice	NaN	8
262987	Red Notice	NaN	7
263006	Red Notice	NaN	6
263023	Red Notice	NaN	5
263043	Red Notice	NaN	4
263061	Red Notice	NaN	3
263080	Red Notice	NaN	2
263100	Red Notice	NaN	1

0.0.1 Most popular movies in Italy that are also popular everywhere

[45]:
```
top_20 = popular.sort_values(by='views_first_91_days',ascending=False).head(20)
top_20
```

[45]:

	category	rank	show_title	\
30	TV (Non-English)	1	Squid Game	
20	TV (English)	1	Wednesday	
0	Films (English)	1	Red Notice	
1	Films (English)	2	Don't Look Up	
2	Films (English)	3	The Adam Project	
3	Films (English)	4	Bird Box	
4	Films (English)	5	Leave the World Behind	
21	TV (English)	2	Stranger Things	
5	Films (English)	6	The Gray Man	
6	Films (English)	7	We Can Be Heroes	
7	Films (English)	8	The Mother	
8	Films (English)	9	Glass Onion: A Knives Out Mystery	
9	Films (English)	10	Extraction	
22	TV (English)	3	DAHMER	
23	TV (English)	4	Bridgerton	

24	TV (English)	5	The Queen's Gambit
31	TV (Non-English)	2	Money Heist
10	Films (Non-English)	1	Troll
32	TV (Non-English)	3	Lupin
33	TV (Non-English)	4	Money Heist

	season_title	hours_viewed_first_91_days
30	Squid Game: Season 1	2205200000
20	Wednesday: Season 1	1718800000
0	NaN	454200000
1	NaN	408600000
2	NaN	281000000
3	NaN	325300000
4	NaN	339300000
21	Stranger Things 4	1838000000
5	NaN	299500000
6	NaN	231200000
7	NaN	265900000
8	NaN	320300000
9	NaN	266900000
22	DAHMER: Monster: The Jeffrey Dahmer Story	1031100000
23	Bridgerton: Season 1	929300000
24	The Queen's Gambit: Limited Series	746400000
31	Money Heist: Part 4	710200000
10	NaN	178600000
32	Lupin: Part 1	396300000
33	Money Heist: Part 5	900700000

	runtime	views_first_91_days	title
30	8.3167	265200000	Squid Game: Season 1
20	6.8167	252100000	Wednesday: Season 1
0	1.9667	230900000	Red Notice
1	2.3833	171400000	Don't Look Up
2	1.7833	157600000	The Adam Project
3	2.0667	157400000	Bird Box
4	2.3667	143400000	Leave the World Behind
21	13.0667	140700000	Stranger Things 4
5	2.1500	139300000	The Gray Man
6	1.6833	137300000	We Can Be Heroes
7	1.9500	136400000	The Mother
8	2.3500	136300000	Glass Onion: A Knives Out Mystery
9	1.9667	135700000	Extraction
22	8.9167	115600000	DAHMER: Monster: The Jeffrey Dahmer Story
23	8.2000	113300000	Bridgerton: Season 1
24	6.6167	112800000	The Queen's Gambit: Limited Series
31	6.7000	106000000	Money Heist: Part 4
10	1.7333	103000000	Troll

```
32   3.9833            99500000                          Lupin: Part 1
33   9.0833            99200000                    Money Heist: Part 5
```

[46]:
```
nation['title']= nation['season_title'].combine_first(nation['show_title'])
italy_movies = nation[nation['country_name']=='Italy']
italy_movies.sort_values(by='cumulative_weeks_in_top_10',ascending=False).head()
```

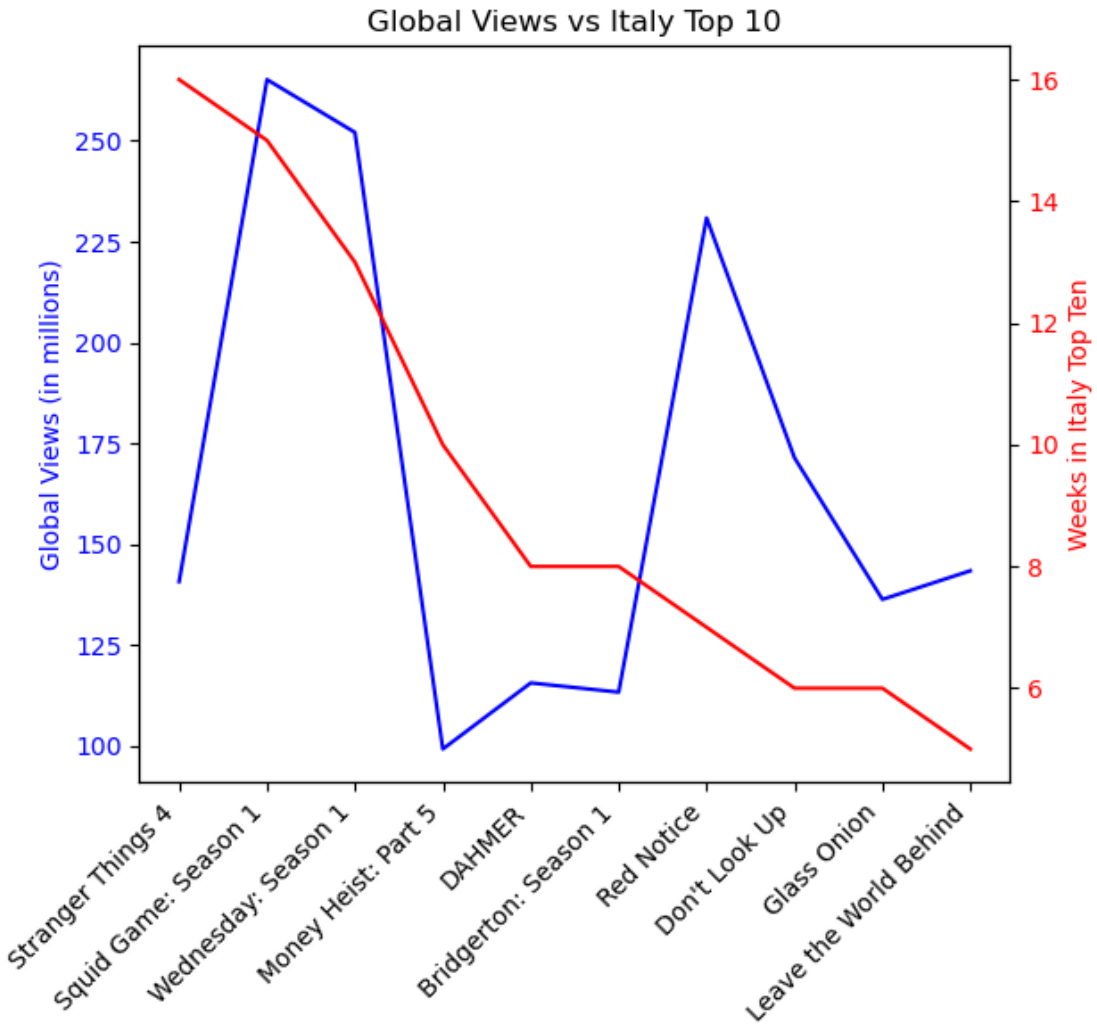
[46]:
```
       country_name country_iso2        week category  weekly_rank  \
109079        Italy           IT  2023-04-23       TV           10
108217        Italy           IT  2024-02-18       TV            8
109094        Italy           IT  2023-04-16       TV            5
108239        Italy           IT  2024-02-11       TV           10
109112        Italy           IT  2023-04-09       TV            3

            show_title              season_title  cumulative_weeks_in_top_10  \
109079  The Sea Beyond  The Sea Beyond: Season 2                          31
108217  The Sea Beyond  The Sea Beyond: Season 1                          31
109094  The Sea Beyond  The Sea Beyond: Season 2                          30
108239  The Sea Beyond  The Sea Beyond: Season 1                          30
109112  The Sea Beyond  The Sea Beyond: Season 2                          29

                         title
109079  The Sea Beyond: Season 2
108217  The Sea Beyond: Season 1
109094  The Sea Beyond: Season 2
108239  The Sea Beyond: Season 1
109112  The Sea Beyond: Season 2
```

[80]:
```
top_italy = italy_movies[['title','cumulative_weeks_in_top_10']].
 ↪groupby('title').count().reset_index()
top_italy = top_italy.
 ↪sort_values(by='cumulative_weeks_in_top_10',ascending=False)
top_italy.head(10)
```

[80]:
```
                           title  cumulative_weeks_in_top_10
1004      The Sea Beyond: Season 1                          31
1005      The Sea Beyond: Season 2                          31
906      The Good Doctor: Season 1                          16
833              Stranger Things 4                          16
822           Squid Game: Season 1                          15
545             Manifest: Season 1                          14
1140           Wednesday: Season 1                          13
946   The Lincoln Lawyer: Season 1                          11
141           Bridgerton: Season 2                          11
538           Maid: Limited Series                          10
```

[64]:
```
merged = pd.merge(top_20, top_italy, on='title', how='inner')
merged = merged.sort_values(by='cumulative_weeks_in_top_10',ascending=False).
 ↪head(10)
```

[69]:
```
import matplotlib.pyplot as plt

# Sample data
x = merged['title']
x=x.replace({
    'DAHMER: Monster: The Jeffrey Dahmer Story' : 'DAHMER',
    'Glass Onion: A Knives Out Mystery' : 'Glass Onion'
})
y1 = merged['views_first_91_days'] / 1000000
y2 = merged['cumulative_weeks_in_top_10']

# Create the main plot
fig, ax1 = plt.subplots()

# Plot the first data set on the first y-axis
ax1.plot(x, y1, color='blue', label='Data 1')
ax1.set_ylabel('Global Views (in millions)', color='blue')
ax1.tick_params(axis='y', labelcolor='blue')

# Create the second y-axis
ax2 = ax1.twinx()

# Plot the second data set on the second y-axis
ax2.plot(x, y2, color='red', label='Data 2')
ax2.set_ylabel('Weeks in Italy Top Ten', color='red')
ax2.tick_params(axis='y', labelcolor='red')

ax1.set_xticks(x)
ax1.set_xticklabels(x, rotation=45, ha='right')

# Add a title and legend
plt.title('Global Views vs Italy Top 10')
fig.set_figheight(6)
fig.tight_layout()
plt.show()
```



0.0.2 The most popular content in Italy, regardless of its popularity elsewhere

[90]:
```
world['title']= world['season_title'].combine_first(world['show_title'])
world_for_merge = world[['title','weekly_hours_viewed']].groupby('title').sum().
  ↪reset_index()
print(world_for_merge.shape)
italy_merged = pd.merge(top_italy.head(10), world_for_merge, on='title',␣
  ↪how='left')
italy_merged = italy_merged.
  ↪sort_values(by='cumulative_weeks_in_top_10',ascending=False)
italy_merged
```

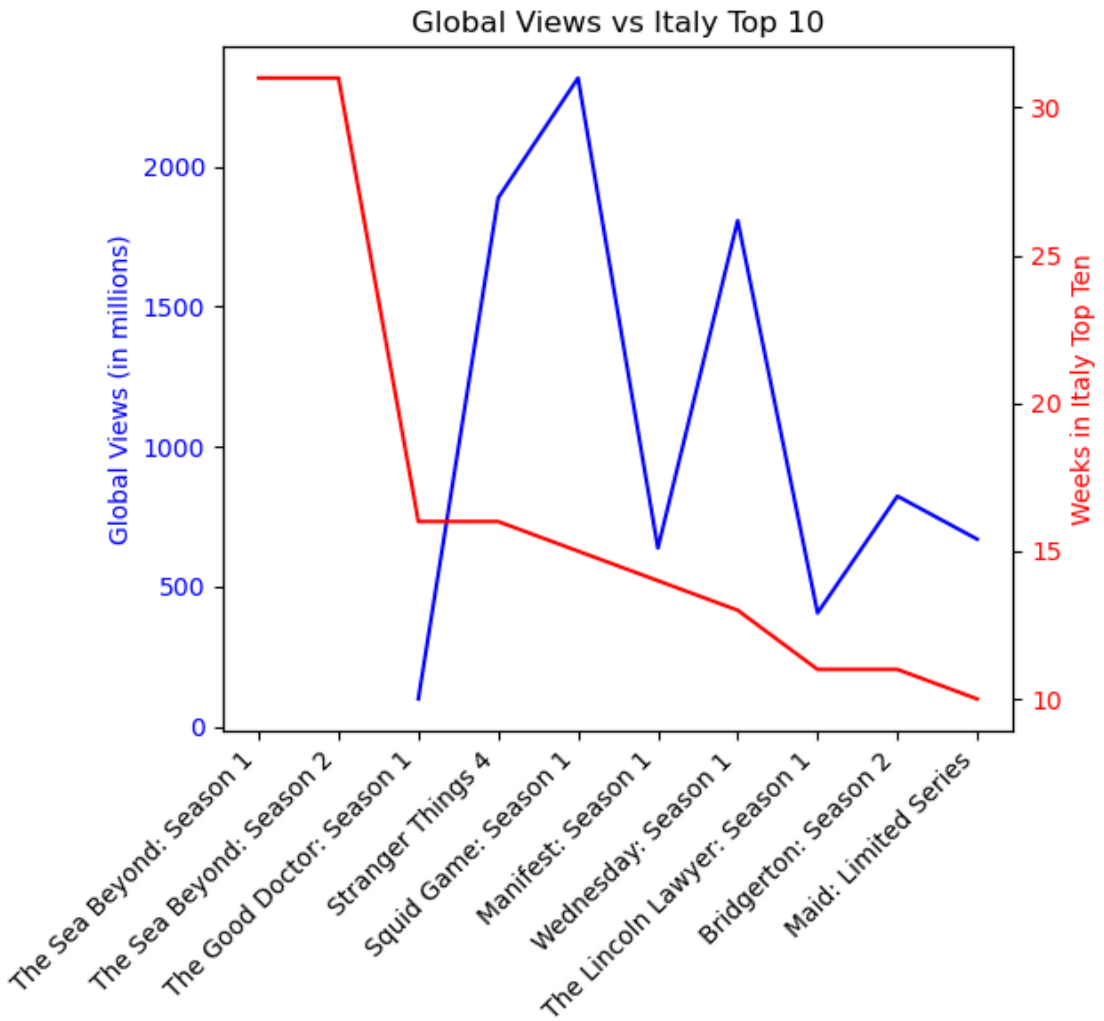
```
(2110, 2)
```

[90]:

	title	cumulative_weeks_in_top_10	weekly_hours_viewed
0	The Sea Beyond: Season 1	31	NaN
1	The Sea Beyond: Season 2	31	NaN
2	The Good Doctor: Season 1	16	9.934000e+07
3	Stranger Things 4	16	1.887310e+09
4	Squid Game: Season 1	15	2.315500e+09
5	Manifest: Season 1	14	6.374200e+08
6	Wednesday: Season 1	13	1.806850e+09
7	The Lincoln Lawyer: Season 1	11	4.061700e+08
8	Bridgerton: Season 2	11	8.232000e+08
9	Maid: Limited Series	10	6.694700e+08

[91]:
```
# Sample data
x = italy_merged['title']
x=x.replace({
    'DAHMER: Monster: The Jeffrey Dahmer Story' : 'DAHMER',
    'Glass Onion: A Knives Out Mystery' : 'Glass Onion'
})
y1 = italy_merged['weekly_hours_viewed'] / 1000000
y2 = italy_merged['cumulative_weeks_in_top_10']

# Create the main plot
fig, ax1 = plt.subplots()

# Plot the first data set on the first y-axis
ax1.plot(x, y1, color='blue', label='Data 1')
ax1.set_ylabel('Global Views (in millions)', color='blue')
ax1.tick_params(axis='y', labelcolor='blue')

# Create the second y-axis
ax2 = ax1.twinx()

# Plot the second data set on the second y-axis
ax2.plot(x, y2, color='red', label='Data 2')
ax2.set_ylabel('Weeks in Italy Top Ten', color='red')
```

```
ax2.tick_params(axis='y', labelcolor='red')

ax1.set_xticks(x)
ax1.set_xticklabels(x, rotation=45, ha='right')

# Add a title and legend
plt.title('Global Views vs Italy Top 10')
fig.set_figheight(6)
fig.tight_layout()
plt.show()
```



0.0.3 What were movies with the longest burn? weeks_on_top_10_anywhere

[93]:
```
world.head()
```

[93]:
```
         week         category  weekly_rank                    show_title  \
0  2024-04-14  Films (English)            1              What Jennifer Did
1  2024-04-14  Films (English)            2  Woody Woodpecker Goes to Camp
2  2024-04-14  Films (English)            3                          Scoop
3  2024-04-14  Films (English)            4                          Glass
4  2024-04-14  Films (English)            5                   Megan Leavey

  season_title  weekly_hours_viewed  runtime  weekly_views  \
0          NaN             26100000   1.4500    18000000.0
1          NaN             19600000   1.6667    11800000.0
2          NaN             14600000   1.7167     8500000.0
3          NaN             11000000   2.1500     5100000.0
4          NaN              9700000   1.9333     5000000.0

   cumulative_weeks_in_top_10  is_staggered_launch episode_launch_details  \
0                           1                False                     NaN
1                           1                False                     NaN
2                           2                False                     NaN
3                           2                False                     NaN
4                           1                False                     NaN

                         title
0              What Jennifer Did
1  Woody Woodpecker Goes to Camp
2                          Scoop
3                          Glass
4                   Megan Leavey
```

[165]:
```
longest_views = world[['title','weekly_hours_viewed']].groupby('title').sum().
 ↪reset_index()
```

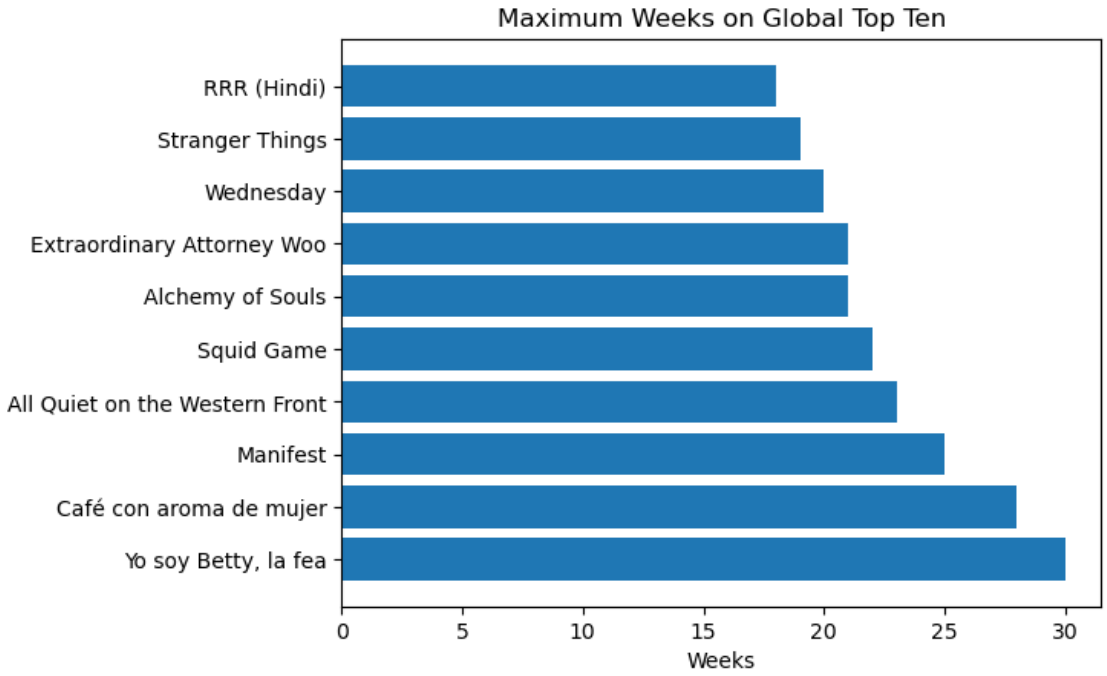
[166]:
```
longest_burn = world[['title','cumulative_weeks_in_top_10','show_title']].
 ↪groupby(['title','show_title']).max().reset_index()
longest_burn = longest_burn.
 ↪sort_values(by='cumulative_weeks_in_top_10',ascending=False).head(10)
longest_burn = pd.merge(longest_burn.head(10), longest_views, on='title',␣
 ↪how='left')
longest_burn
```

[166]:
```
                                title                    show_title  \
0       Yo soy Betty, la fea: Season 1          Yo soy Betty, la fea
1    Café con aroma de mujer: Season 1       Café con aroma de mujer
2                  Manifest: Season 1                      Manifest
```

3	All Quiet on the Western Front	All Quiet on the Western Front
4	Squid Game: Season 1	Squid Game
5	Alchemy of Souls: Part 1	Alchemy of Souls
6	Extraordinary Attorney Woo: Season 1	Extraordinary Attorney Woo
7	Wednesday: Season 1	Wednesday
8	Stranger Things 4	Stranger Things
9	RRR (Hindi)	RRR (Hindi)

	cumulative_weeks_in_top_10	weekly_hours_viewed
0	30	297560000
1	28	813480000
2	25	637420000
3	23	176490000
4	22	2315500000
5	21	303210000
6	21	662090000
7	20	1806850000
8	19	1887310000
9	18	79780000

[161]:
```
import matplotlib.pyplot as plt

categories = longest_burn['show_title']
values = longest_burn['cumulative_weeks_in_top_10']

plt.barh(categories, values)
plt.xlabel("Weeks")
plt.title("Maximum Weeks on Global Top Ten")
plt.show()
```

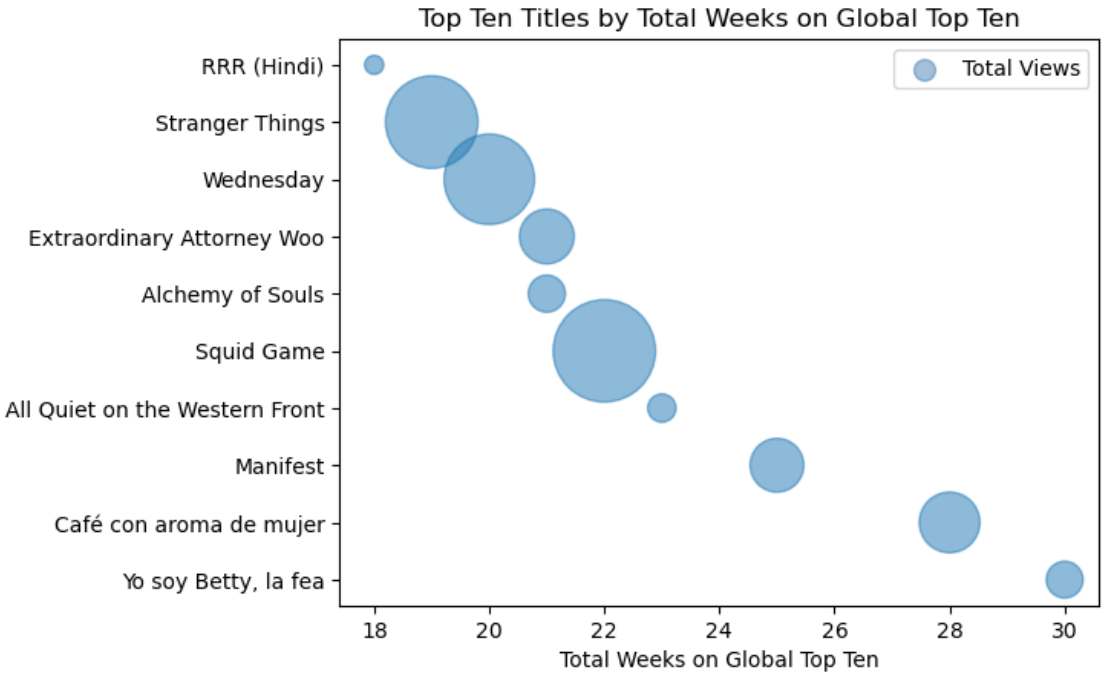


[171]:
```
scatter.legend_elements(num=3)
```

[171]:
```
([], [])
```

[180]:
```
# Sample data
x = longest_burn['cumulative_weeks_in_top_10']
y = longest_burn['show_title']
sizes = longest_burn['weekly_hours_viewed']/1000000 # Size of the bubbles

# Create the scatter plot (bubble chart)
scatter = plt.scatter(x, y, s=sizes, alpha=0.5) # alpha sets transparency

sizes_legend = [100]
labels_legend = ["Total Views"]
handles = []
for size, label in zip(sizes_legend, labels_legend):
  handle = plt.scatter([], [], s=size, alpha=0.5, label=label, color='#4682B4')
  handles.append(handle)
plt.legend(handles=handles, title="Bubble Size")

# Add labels and title
plt.xlabel("Total Weeks on Global Top Ten")
plt.title("Top Ten Titles by Total Weeks on Global Top Ten")

# Show the plot
```

```
plt.legend()
plt.show()
```



Most views by being popular in first 91 days vs longest burn

[112]:
```
print(world[(world['title']=='Red Notice')].groupby('title').
 ↪max()['cumulative_weeks_in_top_10'])
popular.head(1)
```

```
title
Red Notice    14
Name: cumulative_weeks_in_top_10, dtype: int64
```

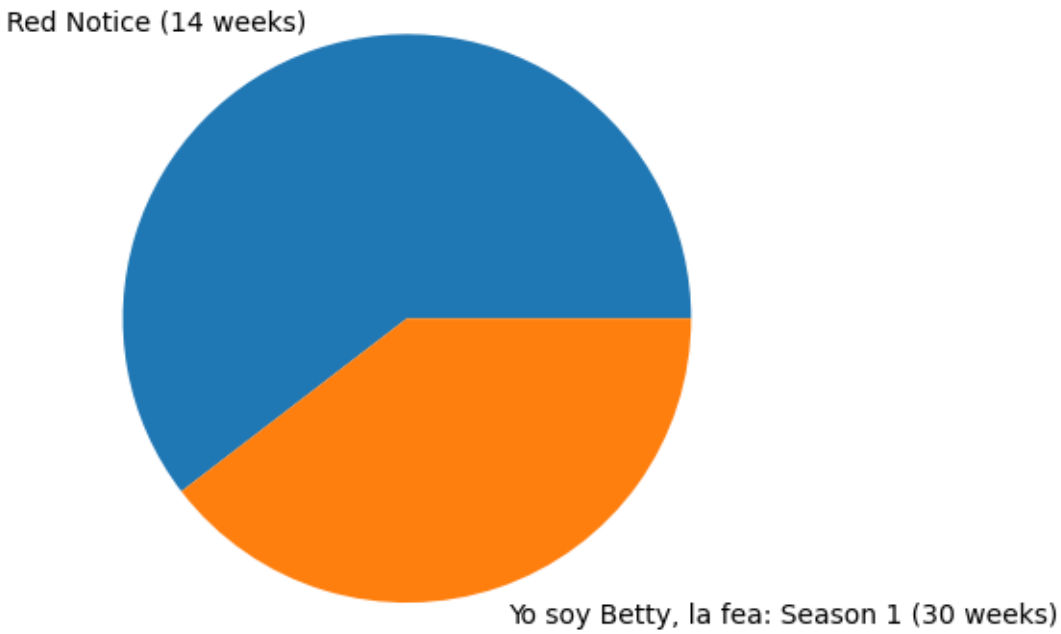
[112]:

	category	rank	show_title	season_title	hours_viewed_first_91_days	runtime	views_first_91_days	title
0	Films (English)	1	Red Notice	NaN	454200000	1.9667	230900000	Red Notice

[115]:
```
versus = world[(world['title']=='Red Notice') | (world['title']=='Yo soy Betty,␣
 ↪la fea: Season 1')][['title','weekly_hours_viewed']]
versus = versus.groupby('title').sum().reset_index()
versus = versus.replace({
    'Red Notice' :'Red Notice (14 weeks)',
    'Yo soy Betty, la fea: Season 1' : 'Yo soy Betty, la fea: Season 1 (30␣
 ↪weeks)'
```

```
})
```

[116]:
```
plt.pie(versus['weekly_hours_viewed'], labels=versus['title'])
plt.show()
```



[124]:
```
random_week = world.sample(n=1)['week'].values[0]
random_week
```

[124]:
```
'2023-05-14'
```

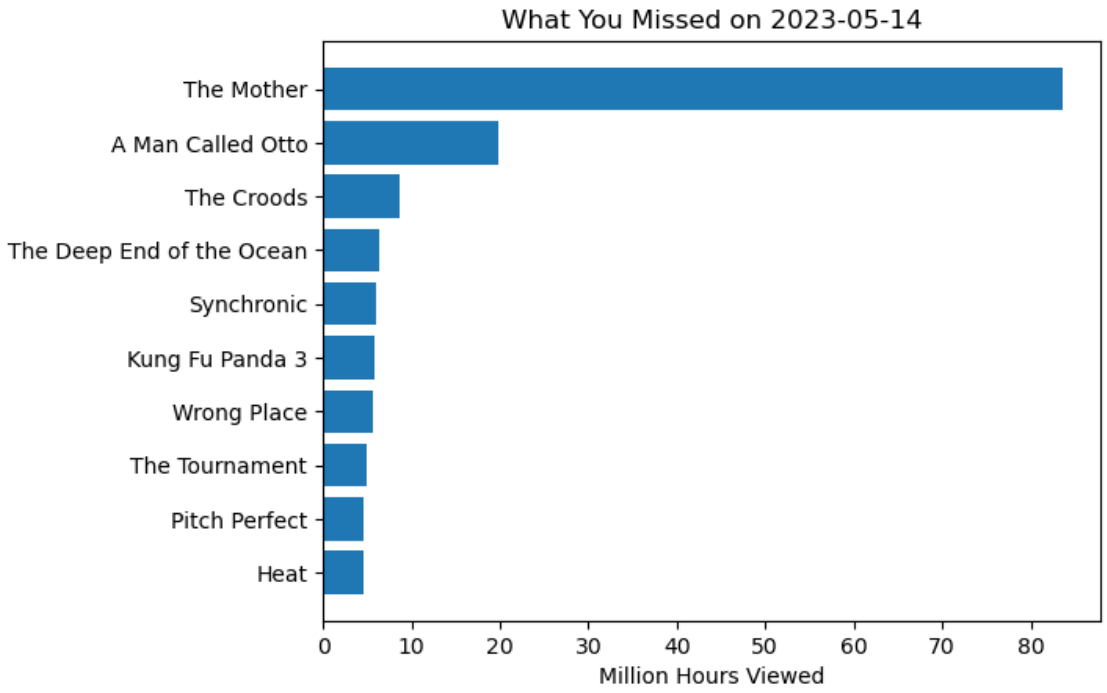
[135]:
```
missed = world[(world['week']==random_week)&(world['category']=='Films␣
↪(English)')][['title','weekly_hours_viewed','weekly_rank']]
missed = missed.sort_values(by='weekly_hours_viewed')
missed
```

[135]:

	title	weekly_hours_viewed	weekly_rank
1929	Heat	4520000	10
1928	Pitch Perfect	4580000	9
1927	The Tournament	4880000	8
1926	Wrong Place	5580000	7
1925	Kung Fu Panda 3	5750000	6
1924	Synchronic	5930000	5
1923	The Deep End of the Ocean	6420000	4

```
1922                The Croods              8700000            3
1921          A Man Called Otto             19880000            2
1920                The Mother              83710000            1
```

[139]:
```
    categories = missed['title']
    values = missed['weekly_hours_viewed'] / 1000000

    plt.barh(categories, values)
    plt.xlabel("Million Hours Viewed")
    plt.title("What You Missed on "+random_week)
    plt.show()
```



Most popular movies outside of the United States

[150]:
```
non_us = nation[nation['country_name'] != 'United␣
 ↪States'][['show_title','cumulative_weeks_in_top_10','country_name']]
non_us = non_us.groupby(['show_title','country_name']).max().
 ↪reset_index()[['show_title','cumulative_weeks_in_top_10']]
non_us = non_us.groupby('show_title').mean().reset_index()
non_us = non_us.sort_values(by='cumulative_weeks_in_top_10',ascending=False).
 ↪head(10)
non_us
```

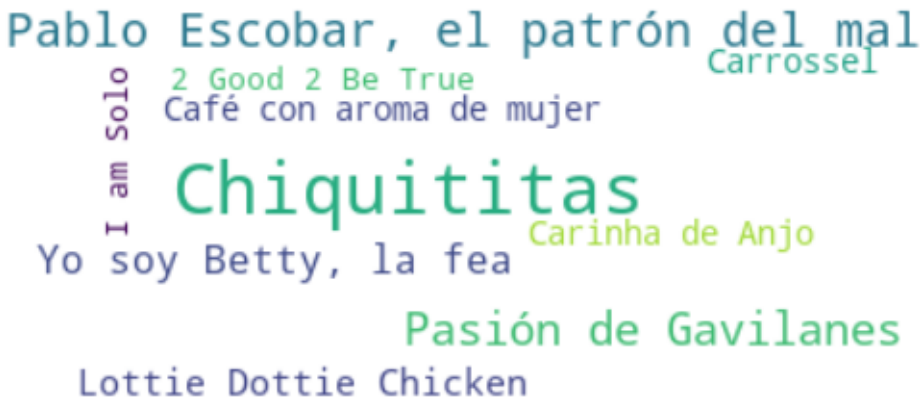
[150]:

	show_title	cumulative_weeks_in_top_10
1078	Chiquititas	93.000000
4114	Pablo Escobar, el patrón del mal	79.941176
4160	Pasión de Gavilanes	70.000000
6956	Yo soy Betty, la fea	49.187500
3223	Lottie Dottie Chicken	44.000000
965	Café con aroma de mujer	41.421053
998	Carinha de Anjo	39.000000
1002	Carrossel	39.000000
2533	I am Solo	32.000000
45	2 Good 2 Be True	32.000000

[152]:
```
frequencies = dict(zip(non_us['show_title'],␣
↪non_us['cumulative_weeks_in_top_10'].astype(int)))
frequencies
```

[152]:
```
{'Chiquititas': 93,
 'Pablo Escobar, el patrón del mal': 79,
 'Pasión de Gavilanes': 70,
 'Yo soy Betty, la fea': 49,
 'Lottie Dottie Chicken': 44,
 'Café con aroma de mujer': 41,
 'Carinha de Anjo': 39,
 'Carrossel': 39,
 'I am Solo': 32,
 '2 Good 2 Be True': 32}
```

[159]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

wordcloud = WordCloud(        background_color='white').
↪generate_from_frequencies(frequencies)

plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```


Pablo Escobar, el patrón del mal
Carrossel
2 Good 2 Be True
Café con aroma de mujer
I am Solo
Chiquititas
Carinha de Anjo
Yo soy Betty, la fea
Pasión de Gavilanes
Lottie Dottie Chicken