DSC640-T301
Chris Kellogg
Weeks 1 & 2 Exercise

Audience
My audience is my wife. She and I have been married for over 34 years, so she's well aware of my career path, my personality, and (most importantly) our family finances. She understands what Netflix is and how top-10 lists work, and she's familiar with basic statistics. She also genuinely wants me to be happy.

Purpose
I want my wife to understand that there's something missing in my life, and that I believe that the creative outlet of developing and producing a film/movie/TV show could fill it. I also want her to believe that I've explored how my new venture might have a reasonable chance to be a financial break-even, rather than a money sink.

Medium
I'm going to write her a letter. There have been a number of times in our lives where we've huddled in front of a computer to do research on a home or car. We've had plenty of conversations, either at home or in the car, about what we want for our kids and our future. But some of the most powerful communications we've ever had are the notes and letters that we've sent each other to express feelings that we might not be able express face-to-face, mainly because they're too emotionally overwhelming.

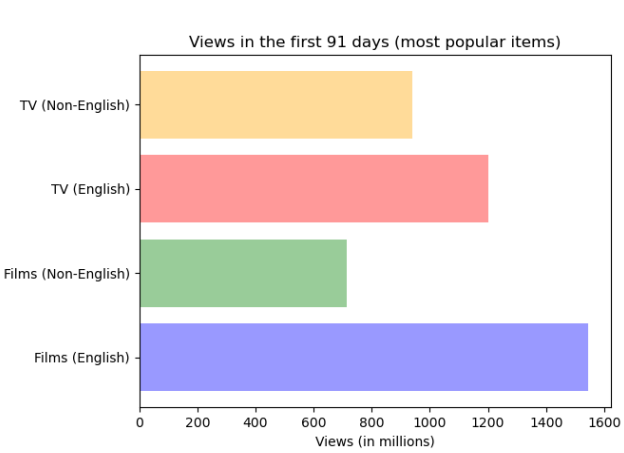
Design
It was extremely important to me that my visualizations by very clean, simple, and straightforward. I tried to use clear language, but not overly formal, and to make sure the graphics don't look too "slick." They're not supposed to seem professionally produced, just easy to understand, so they make a point but don't overwhelm the text. I was careful to use the same colors for the categories in each graphic, but apart from the minor changes to do that, the Jupyter notebook formatting defaults were generally just fine for what I wanted, although I used an alpha to "soften" the colors a bit. I consciously chose to use blue/green (cooler colors) to represent TV and red/orange (warmer colors) to suggest a grouping between them, and I also chose to use primary colors (red/blue) to represent English and secondary colors (orange/green) to represent non-English, again to suggest a grouping.

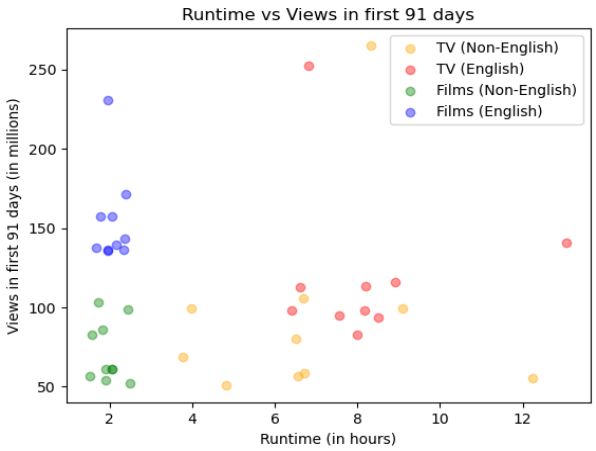
Ethical Considerations
The only changes I made were to drop a very few records that didn't have data that I needed. It seems unlikely that there are any legal, regulatory, or ethical concerns, since Netflix makes this data publicly available for download from their own web site.

Dear Mary Kay,

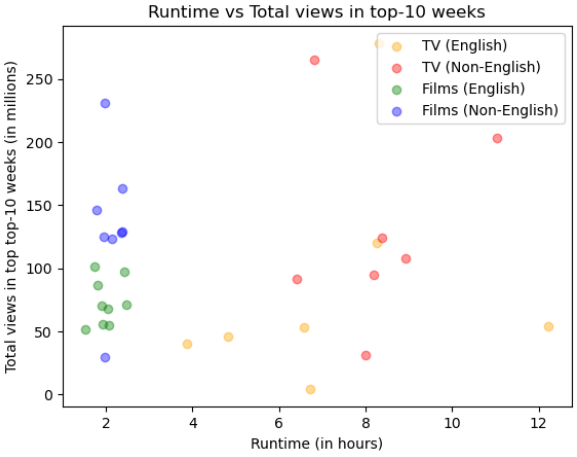
I'm looking for a new, exciting, open-ended challenge. I've been into computer programming and software developing since I was in my teens, and I've loved every minute of it. I really enjoy learning about people's workflows and innovating to solve their problems, and I wouldn't have changed any of it, because it's led to where we are now. But I feel like something is missing. Software is so regimented and structured, and I'm really missing a creative outlet, a chance to build characters, stories, and a world from scratch, a world without limits or boundaries. I want to do that, and I want to share it with people who will love it and care for it and invest themselves into it with me.

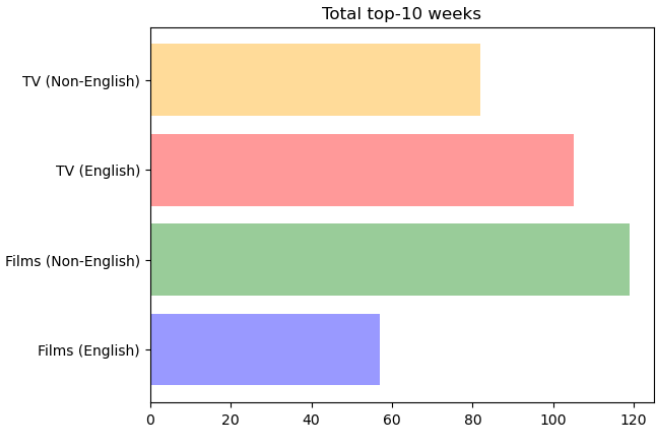
I can't help who I am, though, and I'm a data geek at heart, so I did a little bit of research to find out from Netflix what kind of stuff is most appealing to people.





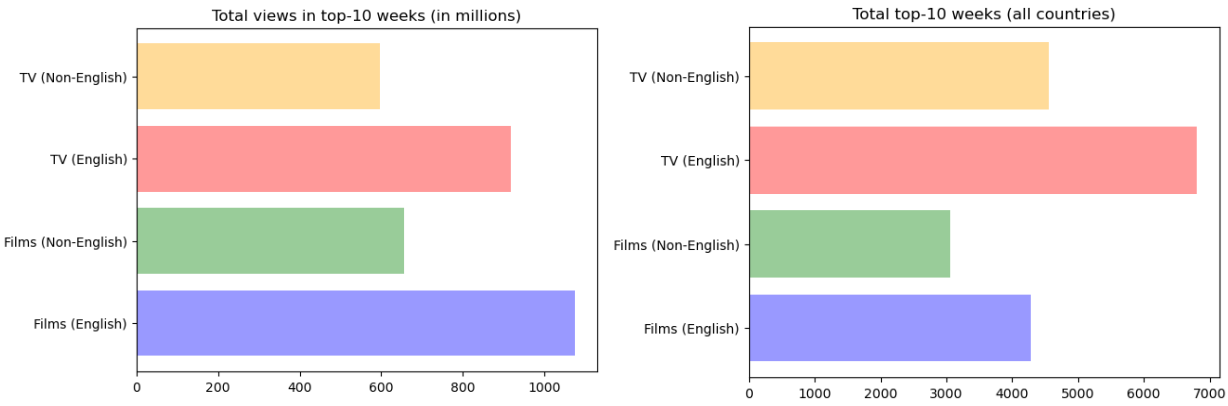
People seem to be looking for content in English. Hooray! That's my best language! And it looks like people prefer films over TV in those first 3 months, which is probably good news, too. It means that I can start with something shorter and see if I like it and if people like it. If it goes well, maybe it could turn into something bigger.





I also noticed that people seem to watch movies and then move on. I think maybe that's good news, too. We should know really quickly whether people are into it and like it, or whether they're ready to

move on from it right away. If I manage to be part of something that goes well and we can adapt for TV, it looks like good TV has excellent staying power in the top 10.



Here's just a little more evidence that people really like to jump on the bandwagon when it comes to English films. I guess it's no surprise, the way new ones come out, but the views should be really high, really fast if what I make is good. And if we can get to the TV adaptation, here's evidence that it wouldn't just be awesome here at home, but it could reach people all over the world!

I really think I have a chance to do something fun and fulfilling, but I won't take another step unless I know you're on board. I'm hoping that you can support me (or even join me) in something new, exciting, and creative. Either way, though, I really love you.

Yours for life,

Christopher

Kellogg640Week1and2

September 7, 2024

[1]:
```
## Chris Kellogg
## DSC640-T301
## Weeks 1 & 2

## ##############################################################################
## Weeks 1 & 2 Exercises
## ##############################################################################

# import and alias pandas
import pandas as pd

# import and alias numpy
import numpy as np

# import and alias the plotting package
import matplotlib.pyplot as plt

# import packages for suppress warnings
import re
import warnings
```

[2]:
```
with warnings.catch_warnings():
    warnings.filterwarnings(
        'ignore',
        category = UserWarning,
        module = re.escape('openpyxl.styles.stylesheet')
    )
    df_countries = pd.read_excel('all-weeks-countries-netflix.xlsx')
    df_global = pd.read_excel('all-weeks-global-netflix.xlsx')
    df_popular = pd.read_excel('most-popular-netflix.xlsx')
```

[3]:
```
df_countries
```

[3]:

	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
…              …           …          …      …          …
272255       Vietnam            VN  2021-07-04       TV            6
272256       Vietnam            VN  2021-07-04       TV            7
272257       Vietnam            VN  2021-07-04       TV            8
272258       Vietnam            VN  2021-07-04       TV            9
272259       Vietnam            VN  2021-07-04       TV           10

                           show_title                    season_title  \
0                        The Tearsmith                             NaN
1                               Stolen                             NaN
2                        Love, Divided                             NaN
3       Woody Woodpecker Goes to Camp                              NaN
4                        Rest In Peace                             NaN
…                                  …                             …
272255                      Reply 1988            Reply 1988: Season 1
272256                  Nevertheless,   Nevertheless,: Limited Series
272257              Too Hot to Handle     Too Hot to Handle: Season 2
272258             Record of Ragnarok    Record of Ragnarok: Season 1
272259           Crash Landing on You  Crash Landing on You: Season 1

        cumulative_weeks_in_top_10
0                                2
1                                1
2                                1
3                                1
4                                3
…                              …
272255                           1
272256                           1
272257                           1
272258                           1
272259                           1

[272260 rows x 8 columns]
```

[4]:
```
df_global
```

[4]:
```
            week          category  weekly_rank  \
0     2024-04-14   Films (English)            1
1     2024-04-14   Films (English)            2
2     2024-04-14   Films (English)            3
3     2024-04-14   Films (English)            4
4     2024-04-14   Films (English)            5
…          …               …          …
5835  2021-07-04  TV (Non-English)            6
5836  2021-07-04  TV (Non-English)            7
```

```
5837  2021-07-04  TV (Non-English)            8
5838  2021-07-04  TV (Non-English)            9
5839  2021-07-04  TV (Non-English)           10

                        show_title                  season_title  \
0                What Jennifer Did                           NaN
1     Woody Woodpecker Goes to Camp                          NaN
2                            Scoop                           NaN
3                            Glass                           NaN
4                      Megan Leavey                          NaN
…                               …                            …
5835                         Elite               Elite: Season 1
5836                         Elite               Elite: Season 3
5837                         Elite               Elite: Season 2
5838                         Katla               Katla: Season 1
5839             Record of Ragnarok  Record of Ragnarok: Season 1

      weekly_hours_viewed  runtime  weekly_views  cumulative_weeks_in_top_10  \
0                26100000   1.4500    18000000.0                           1
1                19600000   1.6667    11800000.0                           1
2                14600000   1.7167     8500000.0                           2
3                11000000   2.1500     5100000.0                           2
4                 9700000   1.9333     5000000.0                           1
…                       …        …             …                           …
5835             10530000      NaN           NaN                           1
5836             10200000      NaN           NaN                           1
5837             10140000      NaN           NaN                           1
5838              9190000      NaN           NaN                           1
5839              9140000      NaN           NaN                           1

      is_staggered_launch episode_launch_details
0                   False                    NaN
1                   False                    NaN
2                   False                    NaN
3                   False                    NaN
4                   False                    NaN
…                       …                      …
5835                False                    NaN
5836                False                    NaN
5837                False                    NaN
5838                False                    NaN
5839                False                    NaN

[5840 rows x 11 columns]
```

[5]:
```
df_popular
```

[5]:

	category	rank	show_title	season_title	hours_viewed_first_91_days
0	Films (English)	1	Red Notice	NaN	454200000
1	Films (English)	2	Don't Look Up	NaN	408600000
2	Films (English)	3	The Adam Project	NaN	281000000
3	Films (English)	4	Bird Box	NaN	325300000
4	Films (English)	5	Leave the World Behind		
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		
10	Films (Non-English)	1	Troll		
11	Films (Non-English)	2	Society of the Snow		
12	Films (Non-English)	3	Nowhere		
13	Films (Non-English)	4	The Platform		
14	Films (Non-English)	5	Through My Window		
15	Films (Non-English)	6	AKA		
16	Films (Non-English)	7	Blood Red Sky		
17	Films (Non-English)	8	My Name Is Vendetta		
18	Films (Non-English)	9	Black Crab		
19	Films (Non-English)	10	All Quiet on the Western Front		
20	TV (English)	1	Wednesday		
21	TV (English)	2	Stranger Things		
22	TV (English)	3	DAHMER		
23	TV (English)	4	Bridgerton		
24	TV (English)	5	The Queen's Gambit		
25	TV (English)	6	The Night Agent		
26	TV (English)	7	Fool Me Once		
27	TV (English)	8	Stranger Things		
28	TV (English)	9	Bridgerton		
29	TV (English)	10	The Witcher		
30	TV (Non-English)	1	Squid Game		
31	TV (Non-English)	2	Money Heist		
32	TV (Non-English)	3	Lupin		
33	TV (Non-English)	4	Money Heist		
34	TV (Non-English)	5	Money Heist		
35	TV (Non-English)	6	Lupin		
36	TV (Non-English)	7	Who Killed Sara?		
37	TV (Non-English)	8	Berlin		
38	TV (Non-English)	9	All of Us Are Dead		
39	TV (Non-English)	10	Dear Child		

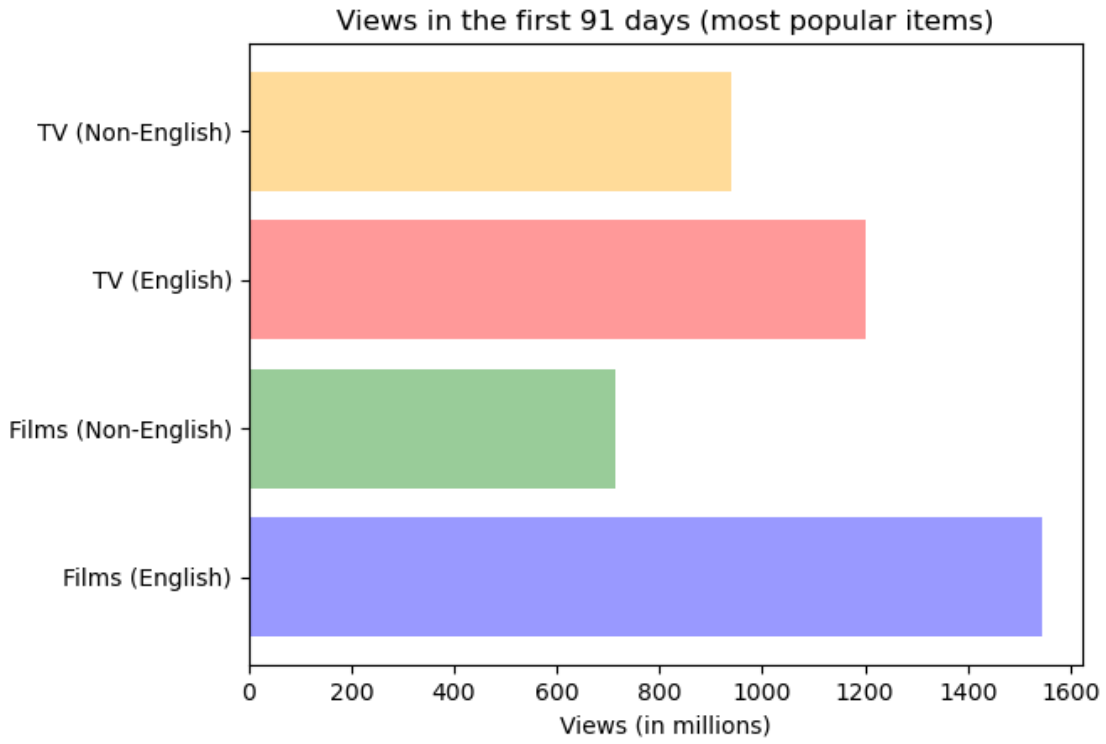
4	NaN	339300000
5	NaN	299500000
6	NaN	231200000
7	NaN	265900000
8	NaN	320300000
9	NaN	266900000
10	NaN	178600000
11	NaN	239700000
12	NaN	155600000
13	NaN	129700000
14	NaN	116000000
15	NaN	125900000
16	NaN	124800000
17	NaN	86500000
18	NaN	103300000
19	NaN	129400000
20	Wednesday: Season 1	1718800000
21	Stranger Things 4	1838000000
22	DAHMER: Monster: The Jeffrey Dahmer Story	1031100000
23	Bridgerton: Season 1	929300000
24	The Queen's Gambit: Limited Series	746400000
25	The Night Agent: Season 1	803200000
26	Fool Me Once: Limited Series	629800000
27	Stranger Things 3	716100000
28	Bridgerton: Season 2	797200000
29	The Witcher: Season 1	663600000
30	Squid Game: Season 1	2205200000
31	Money Heist: Part 4	710200000
32	Lupin: Part 1	396300000
33	Money Heist: Part 5	900700000
34	Money Heist: Part 3	519800000
35	Lupin: Part 2	258900000
36	Who Killed Sara?: Season 1	392400000
37	Berlin: Season 1	372600000
38	All of Us Are Dead: Season 1	679300000
39	Dear Child: Limited Series	245400000

	runtime	views_first_91_days
0	1.9667	230900000
1	2.3833	171400000
2	1.7833	157600000
3	2.0667	157400000
4	2.3667	143400000
5	2.1500	139300000
6	1.6833	137300000
7	1.9500	136400000
8	2.3500	136300000

```
9    1.9667            135700000
10   1.7333            103000000
11   2.4333             98500000
12   1.8167             85700000
13   1.5667             82800000
14   1.9000             61100000
15   2.0667             60900000
16   2.0500             60900000
17   1.5333             56400000
18   1.9167             53900000
19   2.4833             52100000
20   6.8167            252100000
21  13.0667            140700000
22   8.9167            115600000
23   8.2000            113300000
24   6.6167            112800000
25   8.1833             98200000
26   6.4167             98200000
27   7.5500             94800000
28   8.5000             93800000
29   8.0000             83000000
30   8.3167            265200000
31   6.7000            106000000
32   3.9833             99500000
33   9.0833             99200000
34   6.5000             80000000
35   3.7833             68400000
36   6.7167             58400000
37   6.5667             56700000
38  12.2333             55500000
39   4.8167             50900000
```

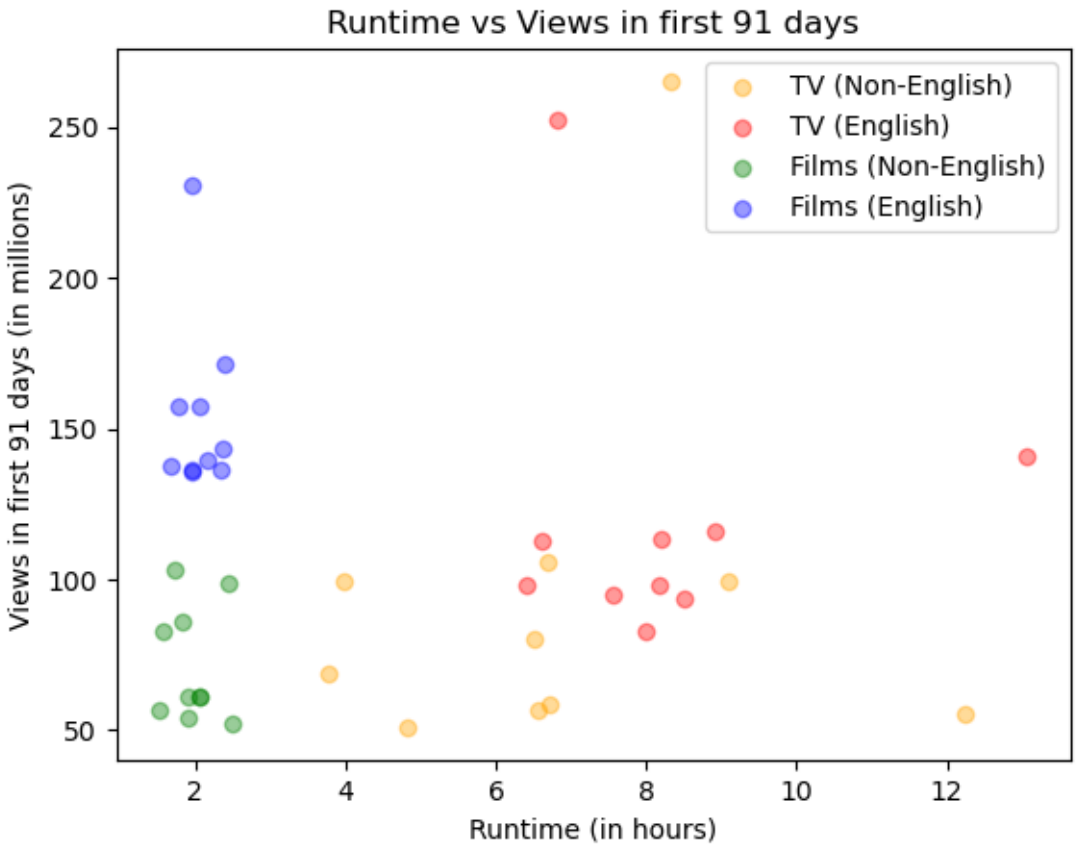
[6]:
```
##
## plot total views in the first 91 days for each category
##

# group and aggregate
df_views = df_popular \
    .groupby(['category']) \
    .agg( \
        views = ('views_first_91_days', 'sum') \
    ) \
    .reset_index()

# plot the total top-10 weeks by category
fig, ax = plt.subplots()
ax.barh(
```

```
    df_views.category,
    df_views.views / 1000000,
    color = ['blue', 'green', 'red', 'orange'],
    alpha = 0.4)

# add the title
plt.title('Views in the first 91 days (most popular items)')
plt.xlabel('Views (in millions)')

plt.show()
```



[7]:
```
##
## plot runtime v total views for each category
##

# split the data set based on the category
tv_non_english = df_popular.query('category == "TV (Non-English)"')
tv_english = df_popular.query('category == "TV (English)"')
films_non_english = df_popular.query('category == "Films (Non-English)"')
films_english = df_popular.query('category == "Films (English)"')

# plot the classes
plt.scatter(
```

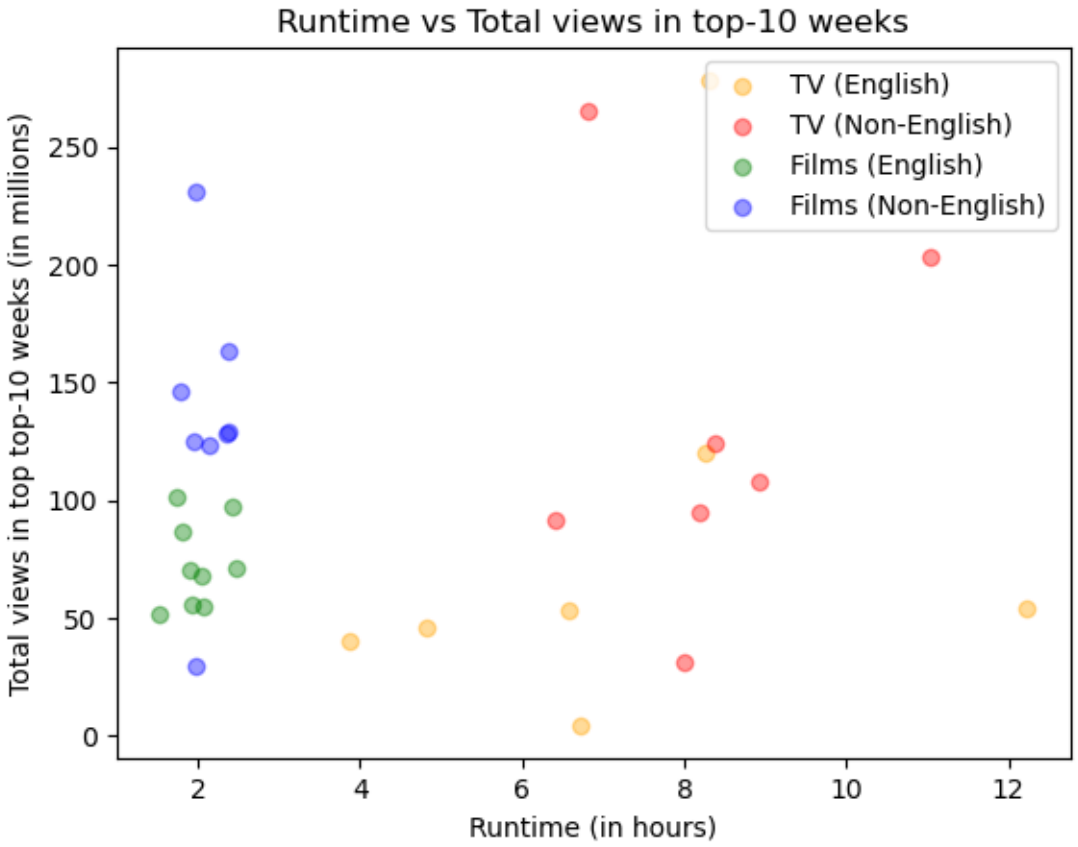
```
    tv_non_english['runtime'],
    tv_non_english['views_first_91_days'] / 1000000,
    alpha = 0.4,
    color =  'orange'
)
plt.scatter(
    tv_english['runtime'],
    tv_english['views_first_91_days'] / 1000000,
    alpha = 0.4,
    color =  'red'
)
plt.scatter(
    films_non_english['runtime'],
    films_non_english['views_first_91_days'] / 1000000,
    alpha = 0.4,
    color =  'green'
)
plt.scatter(
    films_english['runtime'],
    films_english['views_first_91_days'] / 1000000,
    alpha = 0.4,
    color =  'blue'
)

# add the title and axis labels
plt.title('Runtime vs Views in first 91 days')
plt.xlabel('Runtime (in hours)')
plt.ylabel('Views in first 91 days (in millions)')

# add a legend
plt.legend([
    'TV (Non-English)',
    'TV (English)',
    'Films (Non-English)',
    'Films (English)'
])

plt.show()
```



[8]:
```
##
## plot runtime v total views for each category
##

# merge the dataframes
df = pd.merge(
    df_popular,
    df_global,
    how = 'left',
    on = ['category', 'show_title', 'season_title']
)

# keep only the columns we want
df = pd.DataFrame(df[[
    'category',
    'show_title',
    'runtime_x',
    'views_first_91_days',
    'weekly_hours_viewed'
]])
```

```
# drop the records with no viewing data
df = df.dropna()

# group and aggregate
df = df \
    .groupby(['category', 'show_title']) \
    .agg( \
        runtime = ('runtime_x', 'mean'), \
        views_first_91_days = ('views_first_91_days', 'mean'), \
        weeks_in_top_10 = ('weekly_hours_viewed', 'count'), \
        weekly_hours_viewed = ('weekly_hours_viewed', 'sum') \
    ) \
    .reset_index()

# convert hours into views
df['views_in_top_10'] = df.weekly_hours_viewed / df.runtime

# split the data set based on the category
tv_non_english = df.query('category == "TV (Non-English)"')
tv_english = df.query('category == "TV (English)"')
films_non_english = df.query('category == "Films (Non-English)"')
films_english = df.query('category == "Films (English)"')

# plot the classes
plt.scatter(
    tv_non_english['runtime'],
    tv_non_english['views_in_top_10'] / 1000000,
    alpha = 0.4,
    color =  'orange'
)
plt.scatter(
    tv_english['runtime'],
    tv_english['views_in_top_10'] / 1000000,
    alpha = 0.4,
    color =  'red'
)
plt.scatter(
    films_non_english['runtime'],
    films_non_english['views_in_top_10'] / 1000000,
    alpha = 0.4,
    color =  'green'
)
plt.scatter(
    films_english['runtime'],
    films_english['views_in_top_10'] / 1000000,
    alpha = 0.4,
```

```
    color =  'blue'
)

# add the title and axis labels
plt.title('Runtime vs Total views in top-10 weeks')
plt.xlabel('Runtime (in hours)')
plt.ylabel('Total views in top top-10 weeks (in millions)')

# add a legend
plt.legend([
    'TV (English)',
    'TV (Non-English)',
    'Films (English)',
    'Films (Non-English)'
])

plt.show()
```

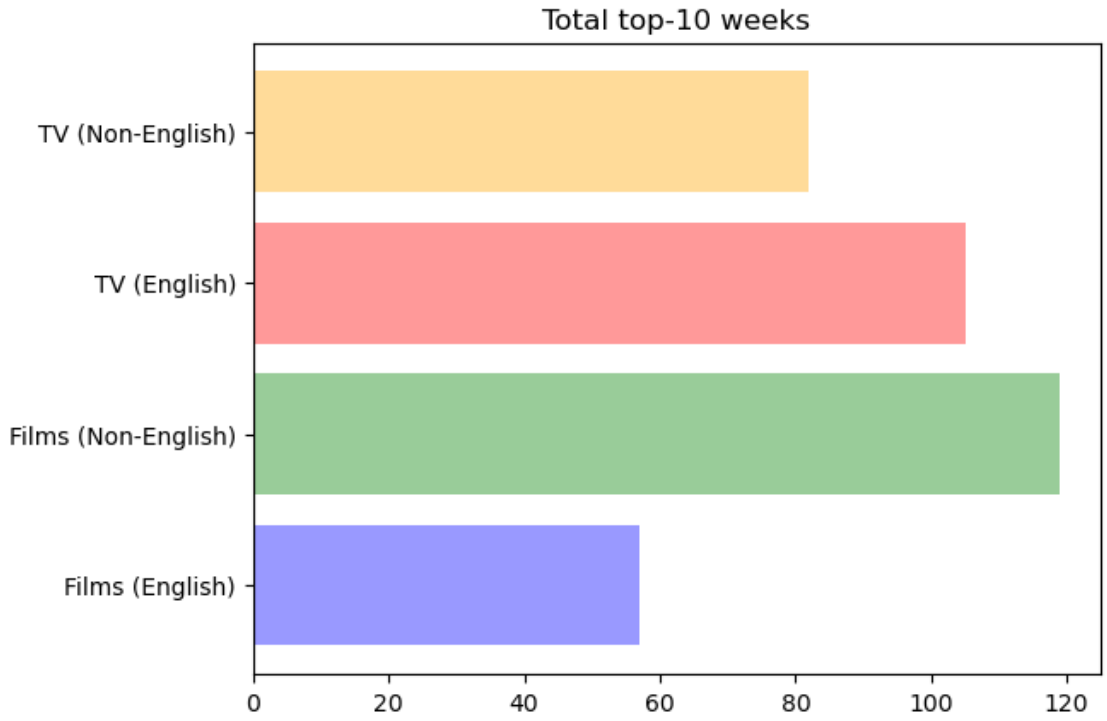


[9]:
```
# group and aggregate
df_top_10_weeks = df \
    .groupby(['category']) \
    .agg( \
        weeks_in_top_10 = ('weeks_in_top_10', 'sum'), \
        views_in_top_10 = ('views_in_top_10', 'sum') \
    ) \
    .reset_index()

df_top_10_weeks
```

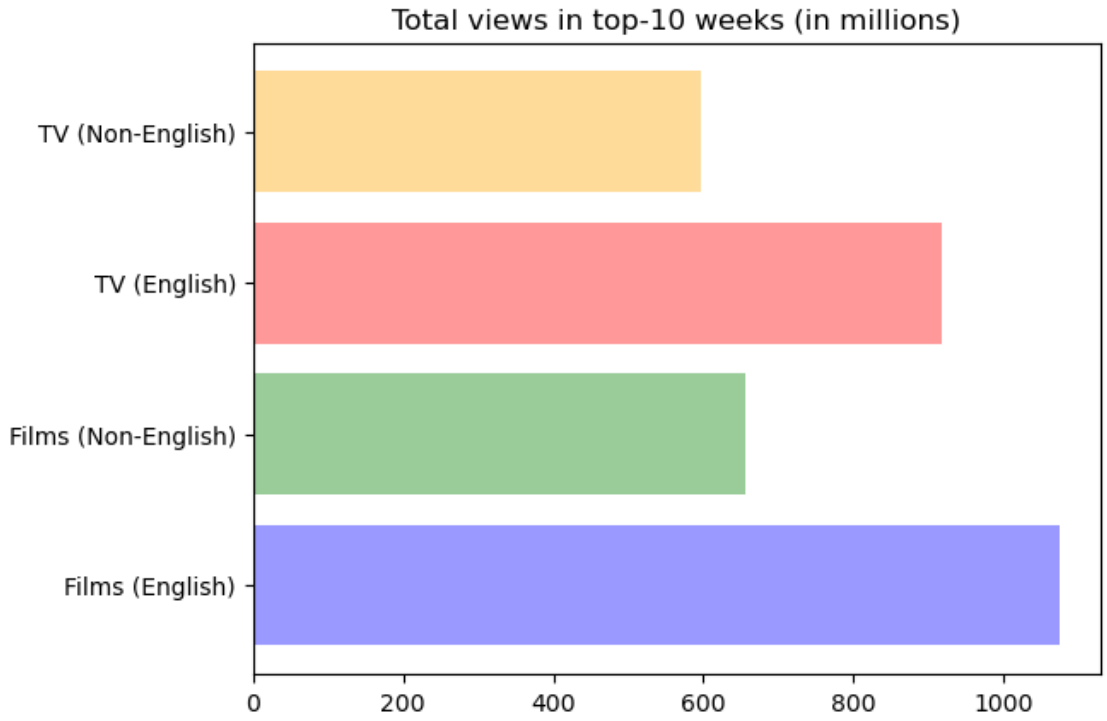
[9]:

	category	weeks_in_top_10	views_in_top_10
0	Films (English)	57	1.075362e+09
1	Films (Non-English)	119	6.560212e+08
2	TV (English)	105	9.183057e+08
3	TV (Non-English)	82	5.958430e+08

[10]:
```
##
## plot total weeks in top 10 for each category
##

# plot the weeks in the top 10 by category
fig, ax = plt.subplots()
ax.barh(
    df_top_10_weeks.category,
    df_top_10_weeks.weeks_in_top_10,
    color = ['blue', 'green', 'red', 'orange'],
    alpha = 0.4)

# add the title
plt.title('Total top-10 weeks')

plt.show()
```



[14]:
```
##
## plot total views in top-10 weeks for each category
##

# plot the total views in top-10 weeks by category
fig, ax = plt.subplots()
ax.barh(
    df_top_10_weeks.category,
    df_top_10_weeks.views_in_top_10 / 1000000,
    color = ['blue', 'green', 'red', 'orange'],
    alpha = 0.4)

# add the title
plt.title('Total views in top-10 weeks (in millions)')

plt.show()
```



[12]:
```
##
## get country viewing data for the popular titles
##

# merge the dataframes
df = pd.merge(
    df_popular,
    df_countries,
    how = 'left',
    on = ['show_title', 'season_title']
)

# rename the category column
df.rename(columns={"category_x": "category"}, inplace=True)

# keep only the columns we want
df = pd.DataFrame(df[[
    'category',
    'country_name'
]])

# drop the records with no country name
df = df.dropna()
```

```
# group and aggregate
df = df \
    .groupby(['category', 'country_name']) \
    .agg( \
        weeks = ('category', 'count') \
    ) \
    .reset_index()

df
```

[12]:

```
              category    country_name  weeks
0     Films (English)       Argentina     31
1     Films (English)       Australia     36
2     Films (English)         Austria     49
3     Films (English)         Bahamas     33
4     Films (English)         Bahrain     43
..                 …             …    …
371  TV (Non-English)  United Kingdom     30
372  TV (Non-English)   United States     28
373  TV (Non-English)         Uruguay     44
374  TV (Non-English)       Venezuela     43
375  TV (Non-English)         Vietnam     31

[376 rows x 3 columns]
```

[13]:

```
##
## plot total top-10 weeks for each category
##

# group and aggregate
df_weeks = df \
    .groupby(['category']) \
    .agg( \
        weeks = ('weeks', 'sum') \
    ) \
    .reset_index()

# plot the total top-10 weeks by category
fig, ax = plt.subplots()
ax.barh(
    df_weeks.category,
    df_weeks.weeks,
    color = ['blue', 'green', 'red', 'orange'],
    alpha = 0.4)

# add the title
```

```
plt.title('Total top-10 weeks (all countries)')

plt.show()
```