In [71]:

```
import pandas as pd
import numpy as np
import seaborn as sbn
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import ttest_ind # T-test for independent samples
from scipy.stats import shapiro # Shapiro-Wilk’s test for Normality
from scipy.stats import levene # Levene's test for Equality of Variance
from scipy.stats import f_oneway # One-way ANOVA
from scipy.stats import chi2_contingency # Chi-square test of independence
```

In [72]:

```
df = pd.read_excel("C:/Users/Hp/Downloads/Netflix.xlsx")
```

In [73]:

```
df.head()
```

Out[73]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020.0	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021.0	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021.0	TV-MA
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021.0	TV-MA
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021.0	TV-MA

In [74]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8809 entries, 0 to 8808
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8809 non-null   object
 1   type          8808 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7983 non-null   object
 5   country       7976 non-null   object
 6   date_added    8796 non-null   datetime64[ns]
 7   release_year  8806 non-null   float64
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8806 non-null   object
 11  description   8806 non-null   object
dtypes: datetime64[ns](1), float64(1), object(10)
memory usage: 826.0+ KB
```

In [75]:
```
df.shape
```

Out[75]:
```
(8809, 12)
```

In [76]:
```
df.dtypes
```

Out[76]:
```
show_id                 object
type                    object
title                   object
director                object
cast                    object
country                 object
date_added      datetime64[ns]
release_year           float64
rating                  object
duration                object
listed_in               object
description             object
dtype: object
```

In [77]:
```
#All the variables are categorical and no need to convert them to numerical.
```

In [78]:

```
#checking for null values
df.isna().sum()
```

Out[78]:

```
show_id            0
type               1
title              2
director        2636
cast             826
country          833
date_added        13
release_year       3
rating             6
duration           5
listed_in          3
description        3
dtype: int64
```

In [79]:

```
# how many percentage of data is missing in each column

missing_value = pd.DataFrame({
    'Missing Value': df.isnull().sum(),
    'Percentage': (df.isnull().sum() / len(df))*100
})
missing_value.sort_values(by='Percentage', ascending=False)
```

Out[79]:

	Missing Value	Percentage
director	2636	29.923941
country	833	9.456238
cast	826	9.376774
date_added	13	0.147576
rating	6	0.068112
duration	5	0.056760
release_year	3	0.034056
listed_in	3	0.034056
description	3	0.034056
title	2	0.022704
type	1	0.011352
show_id	0	0.000000

In [80]:

```
#Numerical variable missing value can be handled by imputation of the missir
#There are no numerical variable.
# Categorical variable can be treated by imputng mode
```

In [81]:
```
df["director"].value_counts()
```

Out[81]:
```
director
Rajiv Chilaka                     19
RaÃºl Campos, Jan Suter           18
Marcus Raboy                      16
Suhas Kadav                       16
Jay Karas                         14
                                  ..
Raymie Muzquiz, Stu Livingston     1
Joe Menendez                       1
Eric Bross                         1
Will Eisenberg                     1
Mozez Singh                        1
Name: count, Length: 4528, dtype: int64
```

In [82]:
```
df["director"].isna().sum()
```

Out[82]:
```
2636
```

In [ ]:
```

```

#There are 2363 missing values in director category and we cant just drop the missing values(almost 30%). Hence in order to treat the missing value, mode has to understood.

In [83]:
```
df['director'].mode()
```

Out[83]:
```
0    Rajiv Chilaka
Name: director, dtype: object
```

In [84]:
```
#Instead of taking mode we can take other in order to avoid biasness towards
#Hence missing directors can be filled with others
```

In [85]:
```
df["director"]=df["director"].fillna('Others')
df["director"].value_counts()
```

Out[85]:
```
director
Others                            2636
Rajiv Chilaka                       19
RaÃºl Campos, Jan Suter             18
Suhas Kadav                         16
Marcus Raboy                        16
                                  ...
Raymie Muzquiz, Stu Livingston       1
Joe Menendez                         1
Eric Bross                           1
Will Eisenberg                       1
Mozez Singh                          1
Name: count, Length: 4529, dtype: int64
```

In [86]:

```
df["country"].value_counts()
```

Out[86]:

```
country
United States                               2817
India                                        972
United Kingdom                               419
Japan                                        245
South Korea                                  199
                                            ... 
Romania, Bulgaria, Hungary                     1
Uruguay, Guatemala                             1
France, Senegal, Belgium                       1
Mexico, United States, Spain, Colombia         1
United Arab Emirates, Jordan                   1
Name: count, Length: 749, dtype: int64
```

In [87]:

```
df["country"].isna().sum()
```

Out[87]:

```
833
```

In [88]:

```
#Country variable has 833 null values and can be imputed with others insteac
```

In [89]:

```
df["country"]=df["country"].fillna('Others')
df["country"].value_counts()
```

Out[89]:

```
country
United States                               2817
India                                        972
Others                                       833
United Kingdom                               419
Japan                                        245
                                            ... 
Romania, Bulgaria, Hungary                     1
Uruguay, Guatemala                             1
France, Senegal, Belgium                       1
Mexico, United States, Spain, Colombia         1
United Arab Emirates, Jordan                   1
Name: count, Length: 750, dtype: int64
```

In [90]:
```
df["cast"].value_counts()
```

Out[90]:
```
cast
David Attenborough
19
Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava, Mo
usam, Swapnil
14
Samuel West
10
Jeff Dunham
7
David Spade, London Hughes, Fortune Feimster
6

..
Michael PeÃ±a, Diego Luna, Tenoch Huerta, Joaquin Cosio, JosÃ© MarÃa Yazpi
k, Matt Letscher, Alyssa Diaz
1
Nick Lachey, Vanessa Lachey
1
Takeru Sato, Kasumi Arimura, Haru, Kentaro Sakaguchi, Takayuki Yamada, Ken
do Kobayashi, Ken Yasuda, Arata Furuta, Suzuki Matsuo, Koichi Yamadera, Ar
ata Iura, Chikako Kaku, Kotaro Yoshida     1
Toyin Abraham, Sambasa Nzeribe, Chioma Chukwuka Akpotha, Chioma Omeruah, C
hiwetalu Agu, Dele Odule, Femi Adebayo, Bayray McNwizu, Biodun Stephen
1
Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna
Malik, Malkeet Rauni, Anita Shabdish, Chittaranjan Tripathy
1
Name: count, Length: 7693, dtype: int64
```

In [91]:
```
df["cast"].isna().sum()
```

Out[91]:
```
826
```

In [92]:
```
#Cast variable has 833 null values and can be imputed with others instead of
```

In [93]:
```
df["cast"]=df["cast"].fillna('Others')
df["cast"].value_counts()
```

Out[93]:
```
cast
Others
826
David Attenborough
19
Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava, Mo
usam, Swapnil
14
Samuel West
10
Jeff Dunham
7

...
Nick Lachey, Vanessa Lachey
1
Takeru Sato, Kasumi Arimura, Haru, Kentaro Sakaguchi, Takayuki Yamada, Ken
do Kobayashi, Ken Yasuda, Arata Furuta, Suzuki Matsuo, Koichi Yamadera, Ar
ata Iura, Chikako Kaku, Kotaro Yoshida      1
Toyin Abraham, Sambasa Nzeribe, Chioma Chukwuka Akpotha, Chioma Omeruah, C
hiwetalu Agu, Dele Odule, Femi Adebayo, Bayray McNwizu, Biodun Stephen
1
Neeraj Kabi, Geetanjali Kulkarni, Danish Husain, Sheeba Chaddha, Paras Pri
yadarshan, Anshul Chauhan, Anud Singh Dhaka, Shirin Sewani, Mihir Ahuja, V
asundhara Rajput                            1
Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna
Malik, Malkeet Rauni, Anita Shabdish, Chittaranjan Tripathy
1
Name: count, Length: 7694, dtype: int64
```

Rest of the variables' missing value can be treated with mode since the percentage of missing value is very less or we can just drop them

In [94]:
```
#checking for null values
df.isna().sum()
```

Out[94]:
```
show_id          0
type             1
title            2
director         0
cast             0
country          0
date_added      13
release_year     3
rating           6
duration         5
listed_in        3
description      3
dtype: int64
```

In [95]:
```
df.shape
```

Out[95]:
```
(8809, 12)
```

In [96]:
```
#description, listed_in, duration, type is imputed with mode in order to tre
#create a list of columns and create an instance of the class "SimpleImputer
```

In [97]:
```
from sklearn.impute import SimpleImputer
```

In [98]:
```
cat_missing = ['description', 'listed_in','duration','type']

freq_imputer = SimpleImputer(strategy = 'most_frequent') # mode
for col in cat_missing:
    df[col] = pd.DataFrame(freq_imputer.fit_transform(pd.DataFrame(df[col]))
```

In [99]:
```
#checking for null values
df.isna().sum()
```

Out[99]:
```
show_id          0
type             0
title            2
director         0
cast             0
country          0
date_added      13
release_year     3
rating           6
duration         0
listed_in        0
description      0
dtype: int64
```

In [100]:
```
#drop date_added and title misisng values
```

In [123]:
```
df.dropna(inplace=True)
```

In [125]:
```
# Checking for duplicate rows -
dup_rows = df[df.duplicated()]
print("No. of duplicate rows: ", dup_rows.shape[0])
```

```
No. of duplicate rows:  0
```

In [126]:

```
#removing mins from data
df['duration']=df['duration'].str.replace(" min","")
df.head()
```

Out[126]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Others	United States	2021-09-25	2020.0	PG-13
1	s2	TV Show	Blood & Water	Others	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021.0	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Others	2021-09-24	2021.0	TV-MA
3	s4	TV Show	Jailbirds New Orleans	Others	Others	Others	2021-09-24	2021.0	TV-MA
4	s5	TV Show	Kota Factory	Others	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021.0	TV-MA

In [127]:
```
df['duration'].unique()
```

Out[127]:
```
array(['90', '2 Seasons', '1 Season', '91', '125', '9 Seasons', '104',
       '127', '4 Seasons', '67', '94', '5 Seasons', '161', '61', '166',
       '147', '103', '97', '106', '111', '3 Seasons', '110', '105', '96',
       '124', '116', '98', '23', '115', '122', '99', '88', '100',
       '6 Seasons', '102', '93', '95', '85', '83', '113', '13', '182',
       '48', '145', '87', '92', '80', '117', '128', '119', '143', '114',
       '118', '108', '63', '121', '142', '154', '120', '82', '109', '101',
       '86', '229', '76', '89', '156', '112', '107', '129', '135', '136',
       '165', '150', '133', '70', '84', '140', '78', '7 Seasons', '64',
       '59', '139', '69', '148', '189', '141', '130', '138', '81', '132',
       '10 Seasons', '123', '65', '68', '66', '62', '74', '131', '39',
       '46', '38', '8 Seasons', '17 Seasons', '126', '155', '159', '137',
       '12', '273', '36', '34', '77', '60', '49', '58', '72', '204',
       '212', '25', '73', '29', '47', '32', '35', '71', '149', '33', '15',
       '54', '224', '162', '37', '75', '79', '55', '158', '164', '173',
       '181', '185', '21', '24', '51', '151', '42', '22', '134', '177',
       '13 Seasons', '52', '14', '53', '8', '57', '28', '50', '9', '26',
       '45', '171', '27', '44', '146', '20', '157', '17', '203', '41',
       '30', '194', '15 Seasons', '233', '237', '230', '195', '253',
       '152', '190', '160', '208', '180', '144', '5', '174', '170', '192',
       '209', '187', '172', '16', '186', '11', '193', '176', '56', '169',
       '40', '10', '3', '168', '312', '153', '214', '31', '163', '19',
       '12 Seasons', '179', '11 Seasons', '43', '200', '196', '167',
       '178', '228', '18', '205', '201', '191'], dtype=object)
```

In [128]:
```
#Description is given in duration column
```

In [134]:
```
df.isna().sum()
```

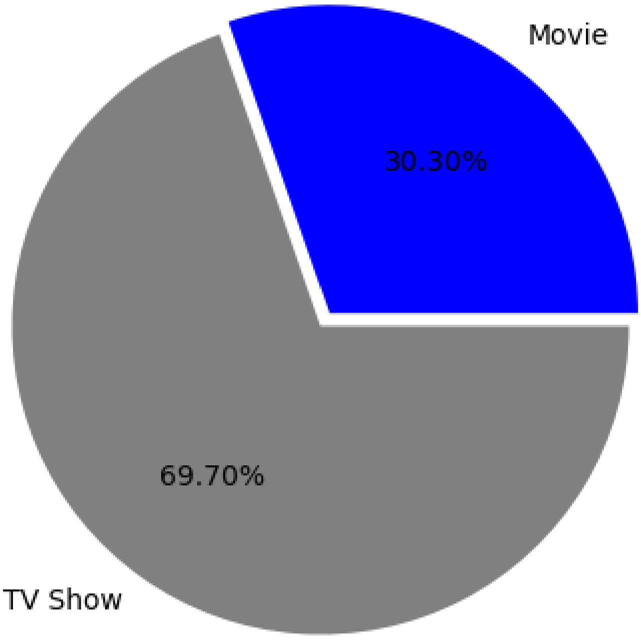
Out[134]:
```
show_id          0
type             0
title            0
director         0
cast             0
country          0
date_added       0
release_year     0
rating           0
duration         0
listed_in        0
description      0
duration_copy    0
dtype: int64
```

In [ ]:
```
#There are no missing values
```

Content type on Netflix

In [136]:

```
plt.figure(figsize=(10,5))
plt.pie(df['type'].value_counts().sort_values(),labels=df['type'].value_cour
          autopct='%1.2f%%',colors=['Blue','grey'])
plt.show()
```



Nearly 2/3rd of the content on netflix are movies and remaining 1/3rd of them are TV Show

Contents added over the year:

In [148]:
```
df.head()
```

Out[148]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Others	United States	2021-09-25	2020.0	PG-13
1	s2	TV Show	Blood & Water	Others	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021.0	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Others	2021-09-24	2021.0	TV-MA
3	s4	TV Show	Jailbirds New Orleans	Others	Others	Others	2021-09-24	2021.0	TV-MA
4	s5	TV Show	Kota Factory	Others	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021.0	TV-MA

In [151]:
```
df_tv = df[df["type"] == "TV Show"]
df_movies = df[df["type"] == "Movie"]
```

In [156]:
```
#number of distinct titles on the basis of type
df.groupby(['type']).agg({"title":"nunique"})
```

Out[156]:

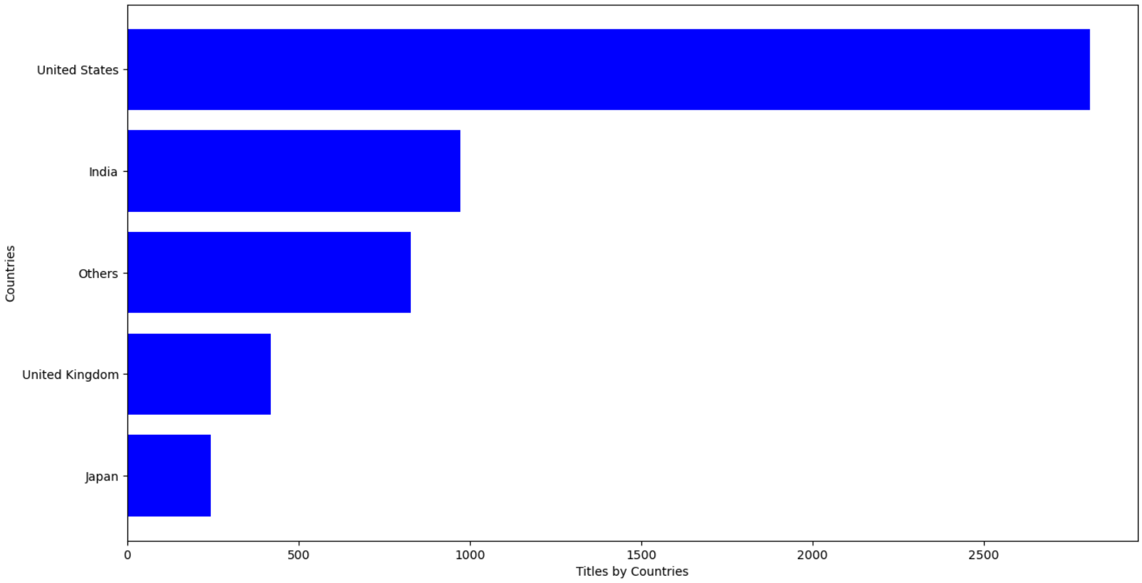
	title
type	
Movie	6128
TV Show	2664

In [158]:
```
#number of distinct titles on the basis of country
df_country=df.groupby(['country']).agg({"title":"nunique"})
```

plt.figure(figsize=(15,8)) plt.barh(df_country[::-1]['country'], df_country[::-1]['title'],color=

In [159]:

```
df_country=df.groupby(['country']).agg({"title":"nunique"}).reset_index().sc
plt.figure(figsize=(15,8))
plt.barh(df_country[::-1]['country'], df_country[::-1]['title'],color=['blue
plt.xlabel('Titles by Countries')
plt.ylabel('Countries')
plt.show()
```



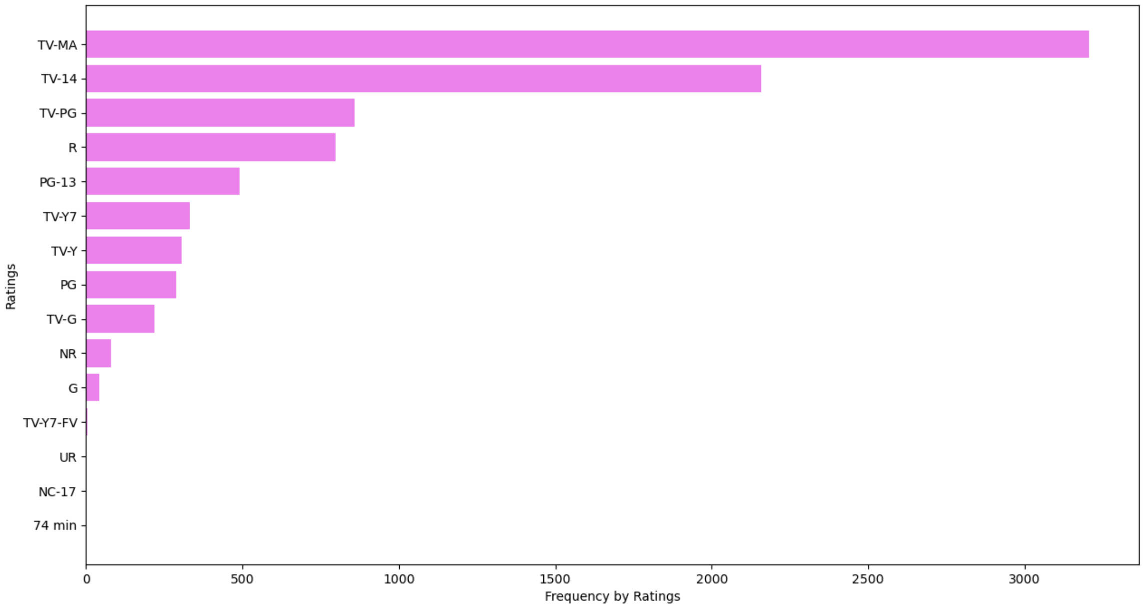
US,India,UK,Canada and France are leading countries in Content Creation on Netflix

In [161]:

```
#number of distinct titles on the basis of rating
df_rating=df.groupby(['rating']).agg({"title":"nunique"})
```

In [163]:

```
df_rating=df.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



Most of the highly rated content on Netflix is intended for Mature Audiences, R Rated, content not intended for audience under 14 and those which require Parental Guidance

```
In [164]: #number of distinct titles on the basis of duration
          df.groupby(['duration']).agg({"title":"nunique"})
```

```
Out[164]:
                   title
           duration
          1 Season  1794
                10     1
        10 Seasons     6
               100   108
               101   116
               ...   ...
                95   137
                96   130
                97   146
                98   120
                99   118

220 rows × 1 columns
```

The duration of Most Watched content in our whole data is 80-100 mins.These must be movies and Shows having only 1 Season.

In [168]:

```
#number of distinct titles on the basis of Actors
df.groupby(['cast']).agg({"title":"nunique"})
```

Out[168]:

	title
cast	
'Najite Dede, Jude Chukwuka, Taiwo Arimoro, Odenike Odetola, Funmi Eko, Keppy Ekpenyong	1
4Minute, B1A4, BtoB, ELSIE, EXID, EXO, Got7, INFINITE, KARA, Shinee, Sistar, VIXX, Nine Muses, BTS, Secret, Topp Dogg	1
50 Cent, Ryan Phillippe, Bruce Willis, Rory Markham, Jenna Dewan, Brett Granstaff, Randy Couture, Susie Abromeit, Ron Turner, James Remar	1
A.J. LoCascio, Sendhil Ramamurthy, Fred Tatasciore, Jake Johnson, Lauren Lapkus, Zachary Levi, BD Wong, David Gunning	1
A.R. Rahman	1
...	...
Ä°brahim BÃ¼yÃ¼kak, Zeynep KoÃ§ak, Gupse Ã–zay, Cengiz Bozkurt	1
Ä°brahim Ã‡elikkol, BelÃ§im Bilgin, Alican YÃ¼cesoy, Teoman KumbaracÄ±baÅŸÄ±, Serdar YeÅŸin, TÃ¼lay GÃ¼nal, GÃ¶zde CÄ±ÄŸacÄ±, Ferit AktuÄŸ, Rojda Demirer, Aybars Kartal Ã–zson	1
Åžahin Irmak, Ä°rem Sak, Gonca Vuslateri, Emre Karayel, Duygu YetiÅŸ, Onur Buldu, Salih Kalyon, Bilge Åžen, NilgÃ¼n BelgÃ¼n, Hakan AkÄ±n	1
ÅžÃ¼krÃ¼ Ã–zyÄ±ldÄ±z, AslÄ± Enver, Åženay GÃ¼rler, BaÅŸak Parlak, Mahir GÃ¼nÅŸiray, Hakan Boyav, Hakan GerÃ§ek, Berrak KuÅŸ, Gamze SÃ¼ner Atay, Mehmet Esen	1
á¹¢á»□páº¹Ì□ DÃ¬rÃsÃ¹, Wunmi Mosaku, Matt Smith, Malaika Wakoli-Abigaba	1

7680 rows × 1 columns

In [172]:

```
df_actors=df.groupby(['cast']).agg({"title":"nunique"}).reset_index().sort_
df_actors
```

Out[172]:

	cast	title
5469	Others	824
1696	David Attenborough	19
7269	Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jig...	14
6304	Samuel West	10
3147	Jeff Dunham	7
1730	David Spade, London Hughes, Fortune Feimster	6
4937	Michela Luci, Jamie Watson, Eric Peterson, Ann...	6
1529	Craig Sechler	6
3927	Kevin Hart	6
3281	Jim Gaffigan	5
2794	Iliza Shlesinger	5
969	Bill Burr	5
975	Bill Hicks	4
226	Aishwarya Rajesh, Vidhu, Surya Ganapathy, Madh...	4
4963	Mike Birbiglia	4
7112	Tom Segura	4
3214	Jerry Seinfeld	4
5683	Prabhas, Rana Daggubati, Anushka Shetty, Taman...	4
1577	Damandeep Singh Baggan, Smita Malhotra, Baba S...	4
3108	Jay O. Sanders	4
1685	Dave Chappelle	4
3286	Jim Jefferies	4
7358	Vir Das	4
4935	Michela Luci, Jamie Watson, Anna Claire Bartla...	4
6260	Sam Kinison	4
6680	Sonal Kaushal, Rupa Bhimani, Julie Tejwani, Sa...	4
1509	Colin Quinn	3
6749	Stephen Fry, Alex Marty	3
5094	Morgan Freeman	3
3416	John Mulaney	3
6441	Sebastian Maniscalco	3

In [174]:
```
#number of distinct titles on the basis of Actors
df.groupby(['director']).agg({"title":"nunique"})
```

Out[174]:
```
                              title
                    director
---------------------------------------
                  A. L. Vijay     2
                A. Raajdheep      1
                    A. Salaam     1
               A.R. Murugadoss    2
               Aadish Keluskar    1
                          ...   ...
             Ã–mer Faruk Sorak    2
           Ã“skar ThÃ³r Axelsson  1
                  Ã‡agan Irmak    1
     Ã€lex Pastor, David Pastor   2
              Åženol SÃ¶nmez      2

4527 rows × 1 columns
```

In [175]:
```
#number of distinct titles on the basis of Actors
df_director = df.groupby(['director']).agg({"title":"nunique"}).reset_index(
df_director
```

Out[175]:

	director	title
3124	Others	2621
3392	Rajiv Chilaka	19
3443	RaÃºl Campos, Jan Suter	18
4046	Suhas Kadav	16
2597	Marcus Raboy	16
1789	Jay Karas	14
684	Cathy Garcia-Molina	13
4479	Youssef Chahine	12
1786	Jay Chapman	12
2670	Martin Scorsese	12
4020	Steven Spielberg	11
1104	Don Michael Paul	10
972	David Dhawan	9
3848	Shannon Hartman	8
1506	Hakan AlgÃ¼l	8
4279	Troy Miller	8
1996	Johnnie To	8
1281	Fernando AyllÃ³n	8
3559	Robert Rodriguez	8
2346	Lance Bangs	8
2324	Kunle Afolayan	8
3345	Quentin Tarantino	8
3649	Ryan Polito	8
4490	YÄ±lmaz ErdoÄŸan	8
1585	Hidenori Inoue	7
3408	Ram Gopal Varma	7
3617	Ron Howard	7
827	Clint Eastwood	7
2748	McG	7
4258	Toshiya Shinohara	7
3655	S.S. Rajamouli	7

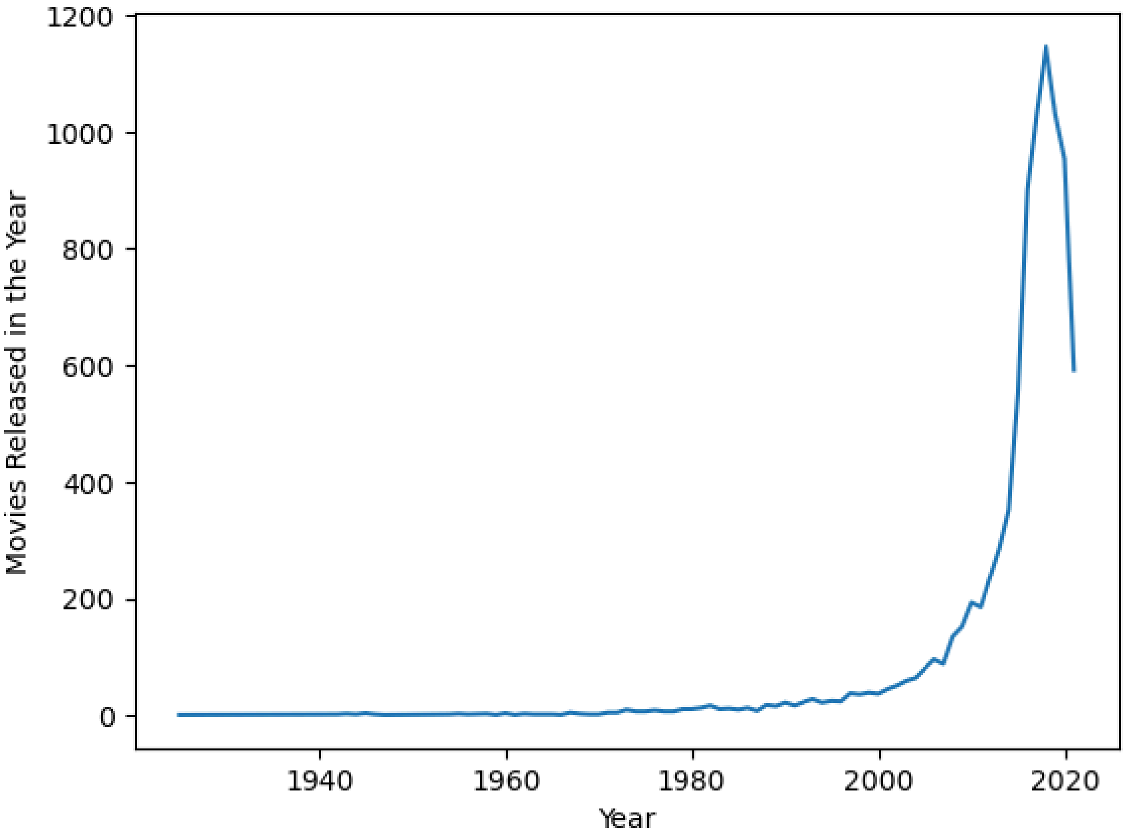
In [ ]:
```
#Rajiv Chilaka,RaÃºl Campos, Jan Suter,Suhas Kadav are most popular director
```

In [178]:
```
#number of distinct titles on the basis of year
df.groupby(['release_year']).agg({"title":"nunique"}).reset_index().sort_val
```

Out[178]:

	release_year	title
73	2021.0	592
72	2020.0	953
71	2019.0	1030
70	2018.0	1146
69	2017.0	1031
68	2016.0	901
67	2015.0	556
66	2014.0	352
65	2013.0	286
64	2012.0	236
63	2011.0	185
62	2010.0	193
61	2009.0	152
60	2008.0	135
59	2007.0	88
58	2006.0	96
57	2005.0	80
56	2004.0	64
55	2003.0	59
54	2002.0	51
53	2001.0	45
52	2000.0	37
51	1999.0	39
50	1998.0	36
49	1997.0	38
48	1996.0	24
47	1995.0	25
46	1994.0	22
45	1993.0	28
44	1992.0	23
43	1991.0	17

In [179]:
```
df_year=df.groupby(['release_year']).agg({"title":"nunique"}).reset_index()
sns.lineplot(data=df_year, x='release_year', y='title')
plt.ylabel("Movies Released in the Year")
plt.xlabel("Year")
plt.show()
```



The Amount of Content across Netflix has increased from 2008 continuously till 2019. Then started decreasing from here(probably due to Covid)

In [180]:
```
df.head(5)
```

Out[180]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Others	United States	2021-09-25	2020.0	PG-13
1	s2	TV Show	Blood & Water	Others	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021.0	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	Others	2021-09-24	2021.0	TV-MA
3	s4	TV Show	Jailbirds New Orleans	Others	Others	Others	2021-09-24	2021.0	TV-MA
4	s5	TV Show	Kota Factory	Others	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021.0	TV-MA

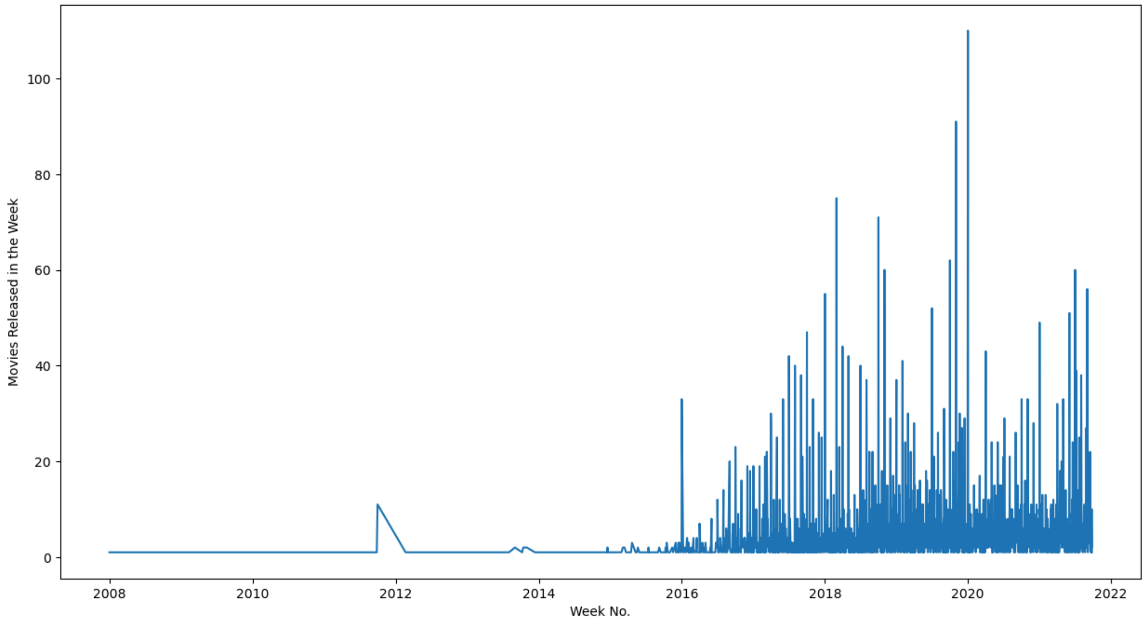
In [181]:
```
#number of distinct titles on the basis of week
df.groupby(['date_added']).agg({"title":"nunique"})
```

Out[181]:

	title
date_added	
2008-01-01	1
2008-02-04	1
2009-05-05	1
2009-11-18	1
2010-11-01	1
...	...
2021-09-21	5
2021-09-22	9
2021-09-23	2
2021-09-24	10
2021-09-25	1

1713 rows × 1 columns

In [182]:
```
df_week=df.groupby(['date_added']).agg({"title":"nunique"}).reset_index()
plt.figure(figsize=(15,8))
sns.lineplot(data=df_week, x='date_added', y='title')
plt.ylabel("Movies Released in the Week")
plt.xlabel("Week No.")
plt.show()
```

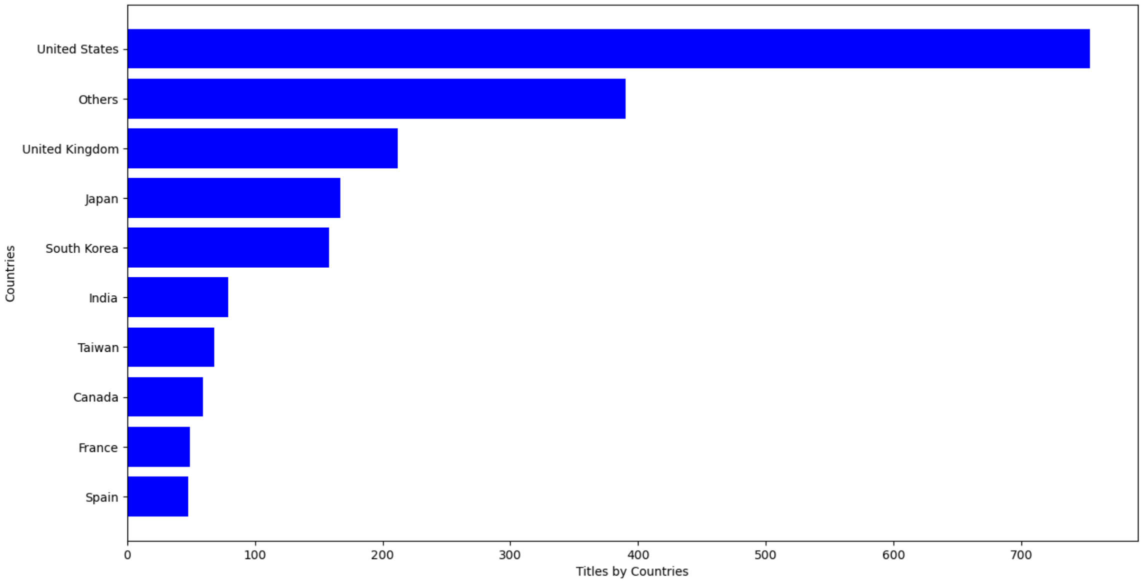


Net content release which are later uploaded to Netflix has increased since 1980 till 2020 though later reduced certainly due to COVID-19

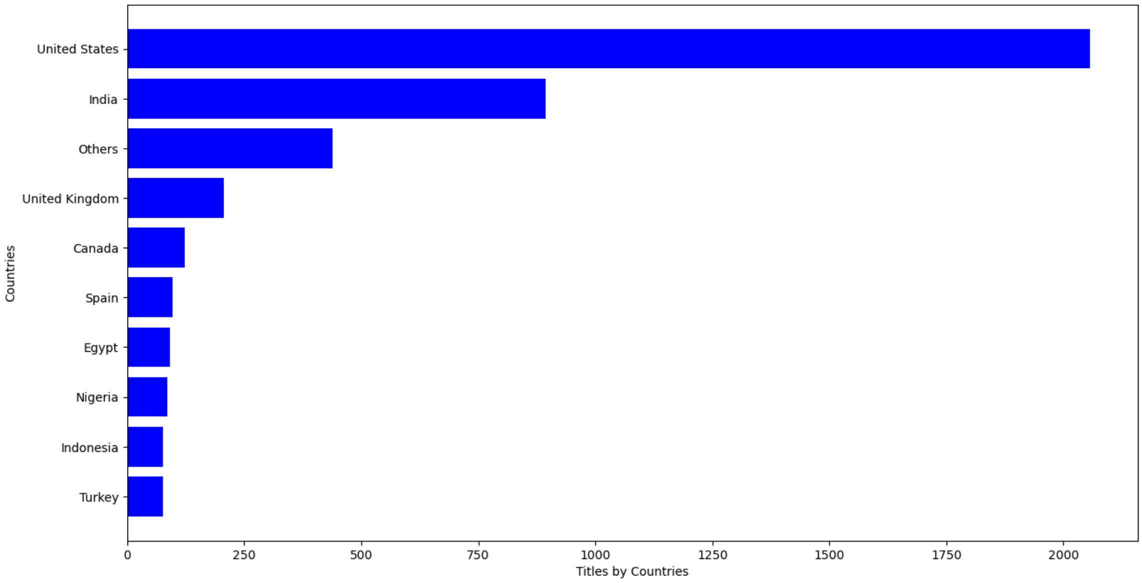
Univariate Analysis

In [183]:
```
df_shows=df[df['type']=='TV Show']
df_movies=df[df['type']=='Movie']
```

```
In [186]: df_country=df_shows.groupby(['country']).agg({"title":"nunique"}).reset_inde
          plt.figure(figsize=(15,8))
          plt.barh(df_country[::-1]['country'], df_country[::-1]['title'],color=['blue
          plt.xlabel('Titles by Countries')
          plt.ylabel('Countries')
          plt.show()
```



```
In [187]: df_country=df_movies.groupby(['country']).agg({"title":"nunique"}).reset_inc
          plt.figure(figsize=(15,8))
          plt.barh(df_country[::-1]['country'], df_country[::-1]['title'],color=['blue
          plt.xlabel('Titles by Countries')
          plt.ylabel('Countries')
          plt.show()
```

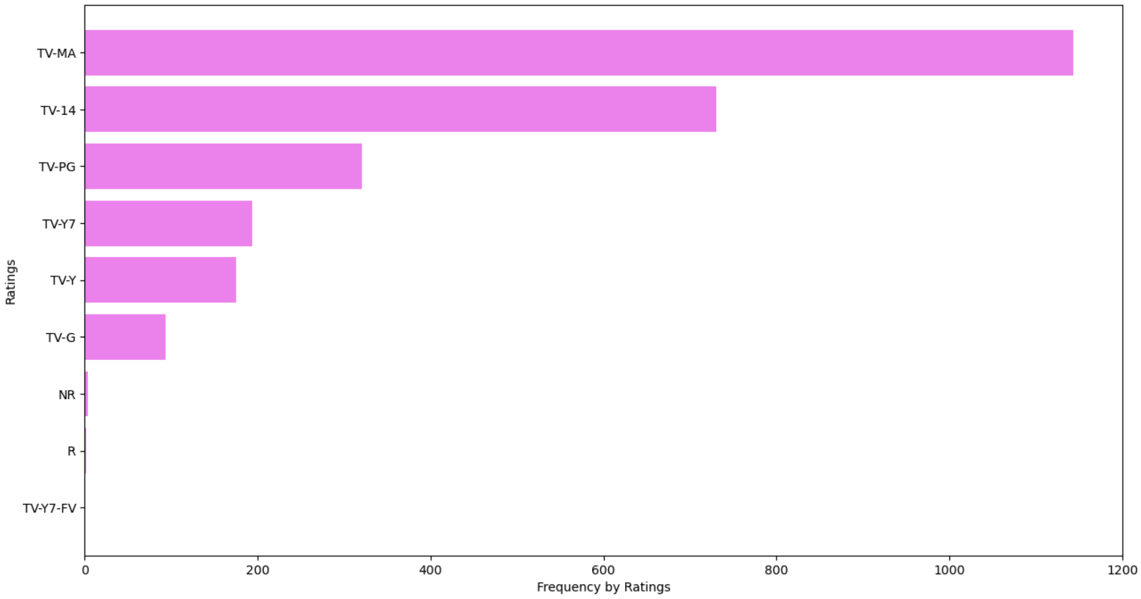


United States is leading across both TV Shows and Movies, UK also provides great content across TV Shows and Movies. Surprisingly India is much more prevalent in Movies as compared TV Shows.

Moreover the number of Movies created in India outweigh the sum of TV Shows and Movies across UK since India was rated as second in net sum of whole content across Netflix.

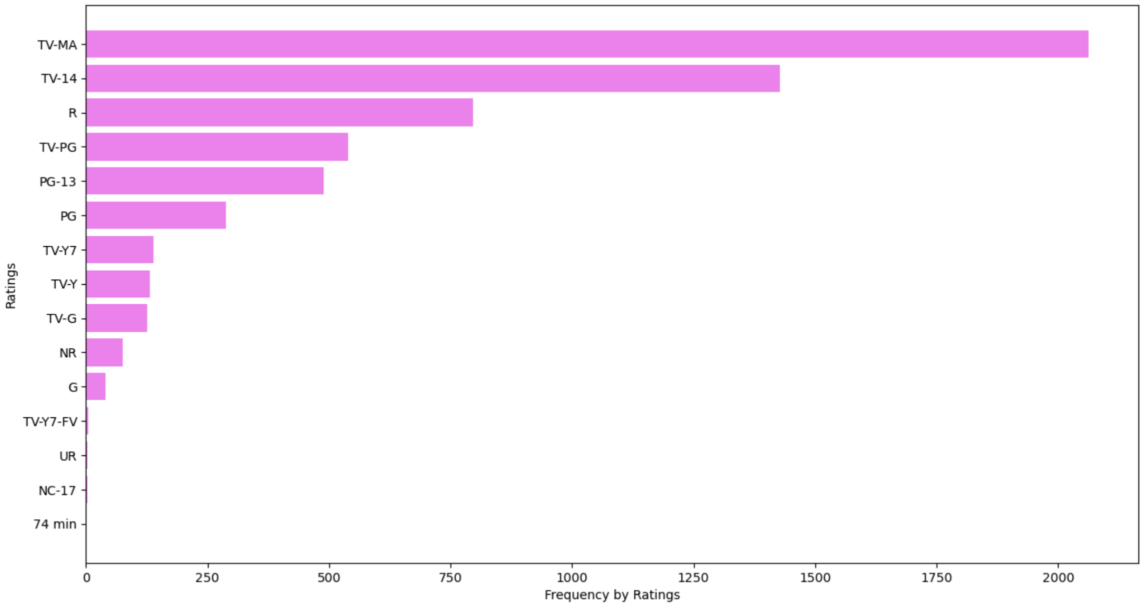
In [188]:

```
df_rating=df_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index(
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



In [189]:

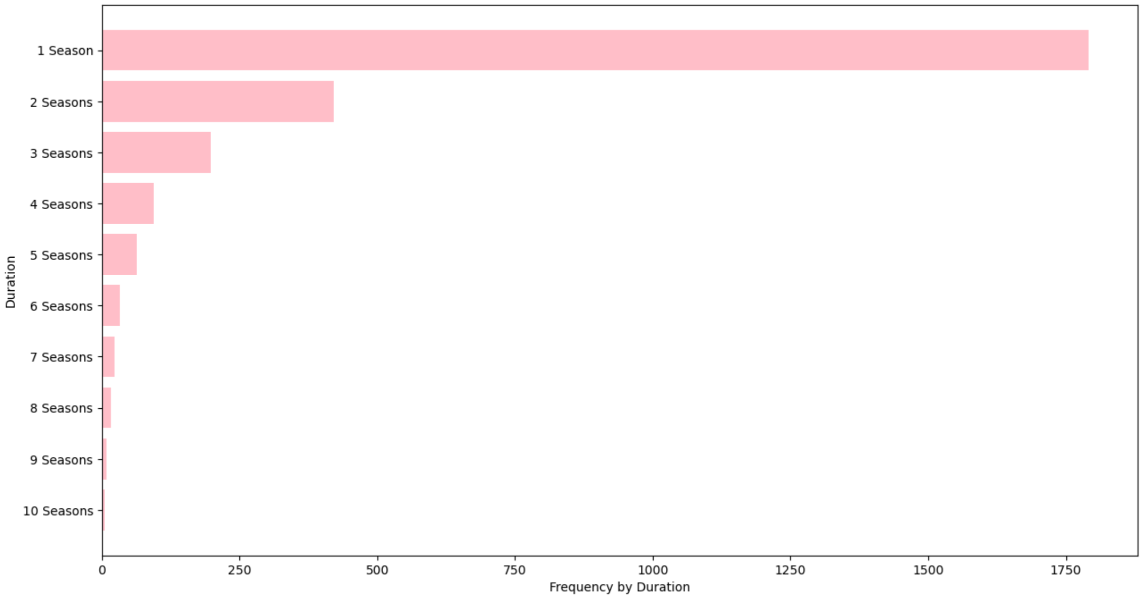
```
df_rating=df_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



So it seems plaussible to conclude that the popular ratings across Netflix includes Mature Audiences and those appropriate for over 14/over 17 ages.

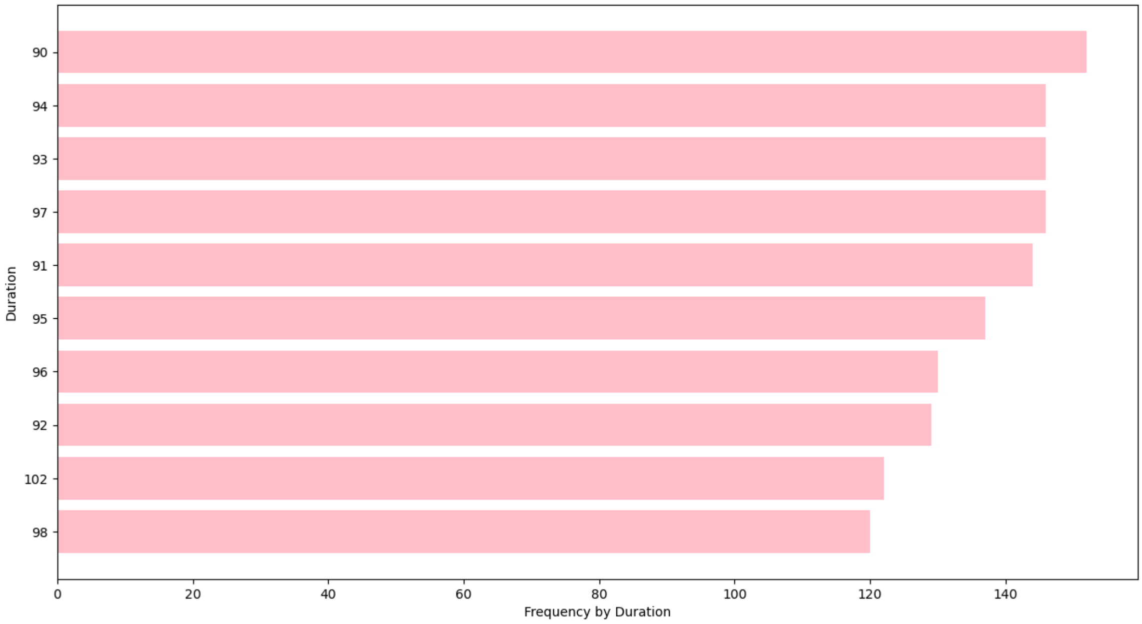
Moreover there are no TV Shows having a rating of R

In [190]:
```
df_duration=df_shows.groupby(['duration']).agg({"title":"nunique"}).reset_i
plt.figure(figsize=(15,8))
plt.barh(df_duration[::-1]['duration'], df_duration[::-1]['title'],color=['
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



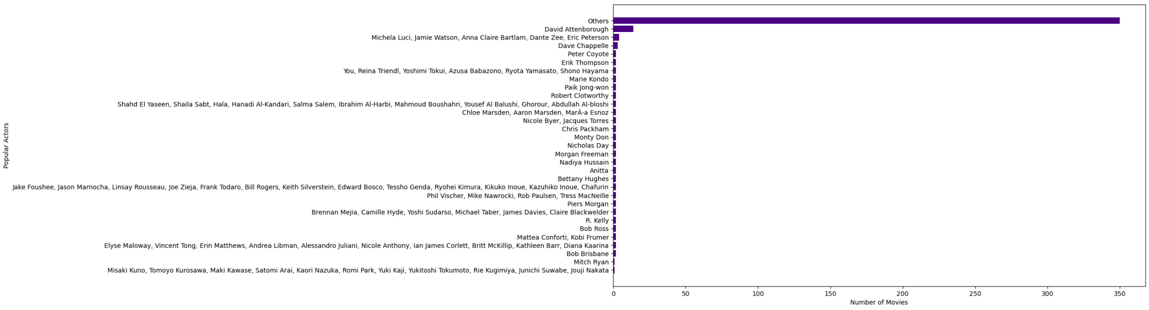
Across TV Shows, shows having only 1 Season are common as soon as the season length increases, the number of shows decrease and this definitely sounds as expected

In [191]:
```
df_duration=df_movies.groupby(['duration']).agg({"title":"nunique"}).reset_i
plt.figure(figsize=(15,8))
plt.barh(df_duration[::-1]['duration'], df_duration[::-1]['title'],color=['
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```

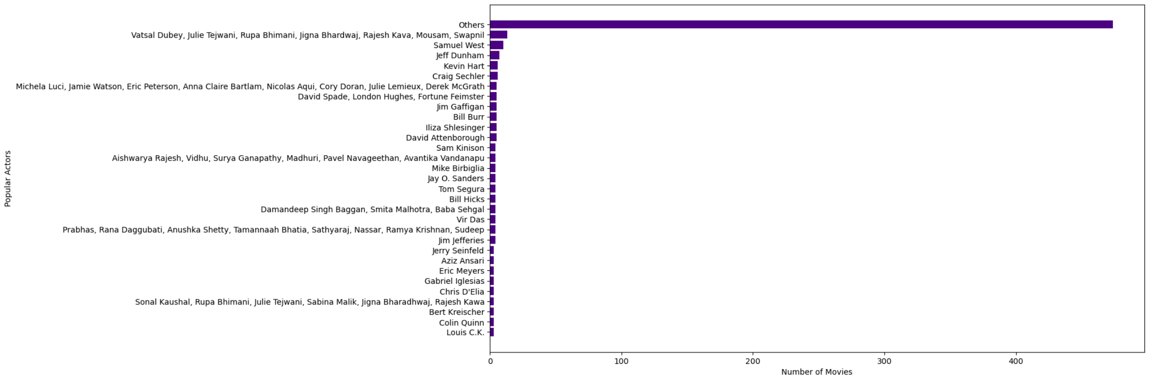


Across movies 80-100 is the ranges of minutes for which most movies lie.

In [193]:
```
df_actors=df_shows.groupby(['cast']).agg({"title":"nunique"}).reset_index().
df_actors=df_actors[df_actors['cast']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['cast'], df_actors[::-1]['title'],color=['indigo']
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actors')
plt.show()
```



In [194]:
```
df_actors=df_movies.groupby(['cast']).agg({"title":"nunique"}).reset_index()
df_actors=df_actors[df_actors['cast']!='Unknown Actor']
plt.figure(figsize=(15,8))
plt.barh(df_actors[::-1]['cast'], df_actors[::-1]['title'],color=['indigo']
plt.xlabel('Number of Movies')
plt.ylabel('Popular Actors')
plt.show()
```



**How has the number of movies released per year changed over the last 20-30 years

Comparison of tv shows vs. movies.

What is the best time to launch a TV show?

Analysis of actors/directors of different types of shows/movies.

Does Netflix has more focus on TV Shows than movies in recent years

Understanding what content is available in different countries

*For USA audience 80-120 mins is the recommended length for movies and Kids TV Shows are also popular along with the genres in first point, hence recommended.

*For UK audience, recommended length for movies is same as that of USA (80-120 mins)

*The target audience in USA and India is recommended to be 14+ and above ratings while for UK, its recommended to be completely Mature/R content . *Add movies for Indian Audience, it has been declining since 2018.

While creating content, take into consideration the popular actors/directors for that country. Also take into account the director-actor combination which is highly recommended.