In [1]:
```
import numpy as np
import pandas as pd
pd.set_option('display.max_columns', 100)
import matplotlib.pyplot as plt
import seaborn as sns
from PIL import Image
from IPython.display import display
```

I made an exploratory data analysis for netflix dataset which I got from kaggle, here i made to answer the question which i got while looking at the dataset. Netflix is one of the international streaming service that offere wide varitey of movies, tv shows and webseries.

In [2]:
```
netflix = Image.open('netflix.png')
display(netflix)
```

Netflix_Exploratory Data Analysis

Exploring the dataset

In [3]:

```
df = pd.read_excel('NetflixOriginals.xlsx')
df
```

Out[3]:

	Title	Genre	Premiere	Runtime	IMDB Score	Language
0	Enter the Anime	Documentary	August 5, 2019	58	2.5	English/Japanese
1	Dark Forces	Thriller	August 21, 2020	81	2.6	Spanish
2	The App	Science fiction/Drama	December 26, 2019	79	2.6	Italian
3	The Open House	Horror thriller	January 19, 2018	94	3.2	English
4	Kaali Khuhi	Mystery	October 30, 2020	90	3.4	Hindi
...	...	...	...	...	...	...
579	Taylor Swift: Reputation Stadium Tour	Concert Film	December 31, 2018	125	8.4	English
580	Winter on Fire: Ukraine's Fight for Freedom	Documentary	October 9, 2015	91	8.4	English/Ukranian/Russian
581	Springsteen on Broadway	One-man show	December 16, 2018	153	8.5	English
582	Emicida: AmarElo - It's All For Yesterday	Documentary	December 8, 2020	89	8.6	Portuguese
583	David Attenborough: A Life on Our Planet	Documentary	October 4, 2020	83	9.0	English

584 rows × 6 columns

Checking datatype

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 584 entries, 0 to 583
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Title       584 non-null    object 
 1   Genre       584 non-null    object 
 2   Premiere    584 non-null    object 
 3   Runtime     584 non-null    int64  
 4   IMDB Score  584 non-null    float64
 5   Language    584 non-null    object 
dtypes: float64(1), int64(1), object(4)
memory usage: 27.5+ KB
```

converting premiere datatype to datetime datatype

In [5]:

```
df['Premiere'] = pd.to_datetime(df['Premiere'], dayfirst = True)
df
```

Out[5]:

	Title	Genre	Premiere	Runtime	IMDB Score	Language
0	Enter the Anime	Documentary	2019-08-05	58	2.5	English/Japanese
1	Dark Forces	Thriller	2020-08-21	81	2.6	Spanish
2	The App	Science fiction/Drama	2019-12-26	79	2.6	Italian
3	The Open House	Horror thriller	2018-01-19	94	3.2	English
4	Kaali Khuhi	Mystery	2020-10-30	90	3.4	Hindi
...	...	...	...	...	...	...
579	Taylor Swift: Reputation Stadium Tour	Concert Film	2018-12-31	125	8.4	English
580	Winter on Fire: Ukraine's Fight for Freedom	Documentary	2015-10-09	91	8.4	English/Ukranian/Russian
581	Springsteen on Broadway	One-man show	2018-12-16	153	8.5	English
582	Emicida: AmarElo - It's All For Yesterday	Documentary	2020-12-08	89	8.6	Portuguese
583	David Attenborough: A Life on Our Planet	Documentary	2020-10-04	83	9.0	English

584 rows × 6 columns

Adding cloumns Date, Months, Year to the dataset for future analysis

In [6]:

```
df['Day'] = df['Premiere'].apply(lambda x : x.day)
months = {1:'Jan', 2:'Feb', 3:'March', 4:'Apr', 5:'May', 6:'Jun', 7:'July', 8:'Aug', 9:'Sep', 10:'Oct', 11:'Nov', 12:'Dec'}
df['Month'] = df['Premiere'].apply(lambda x : months[x.month])
df['Year'] = df['Premiere'].apply(lambda x : x.year)
```

In [7]:
```
df.head()
```

Out[7]:

	Title	Genre	Premiere	Runtime	IMDB Score	Language	Day	Month	Year
0	Enter the Anime	Documentary	2019-08-05	58	2.5	English/Japanese	5	Aug	2019
1	Dark Forces	Thriller	2020-08-21	81	2.6	Spanish	21	Aug	2020
2	The App	Science fiction/Drama	2019-12-26	79	2.6	Italian	26	Dec	2019
3	The Open House	Horror thriller	2018-01-19	94	3.2	English	19	Jan	2018
4	Kaali Khuhi	Mystery	2020-10-30	90	3.4	Hindi	30	Oct	2020

Checking for Null

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Title         0
Genre         0
Premiere      0
Runtime       0
IMDB Score    0
Language      0
Day           0
Month         0
Year          0
dtype: int64
```

Checking for duplicates

In [9]:
```
df.duplicated().unique()
```

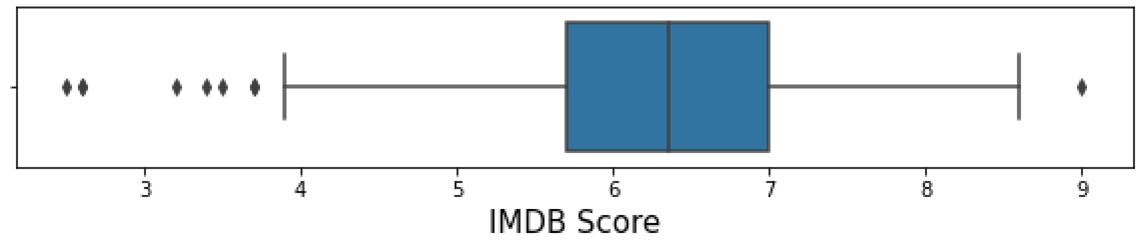
Out[9]:
```
array([False])
```

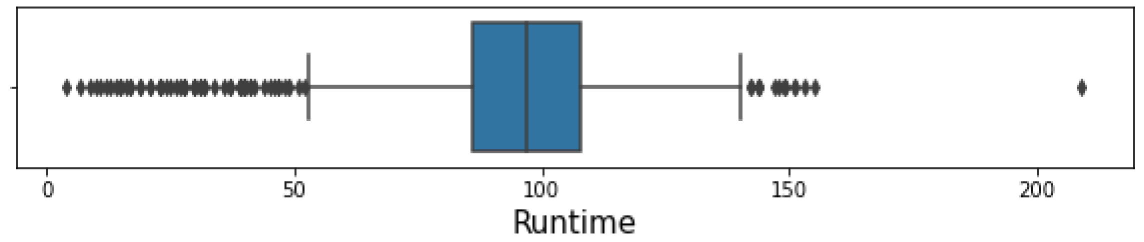
Distribution of numeric features

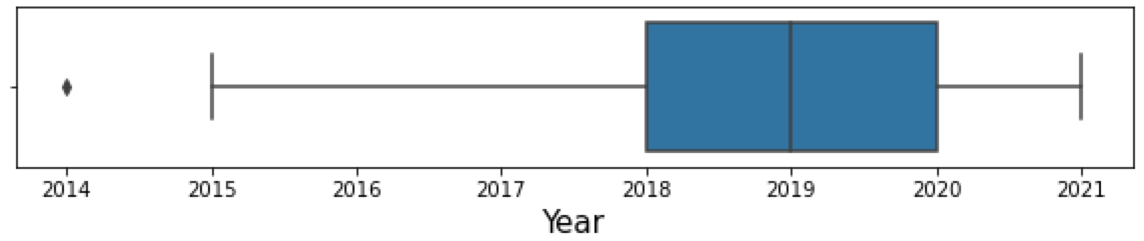
In [10]:

```
plt.figure(figsize=(10,5))
plt.subplot(3,1,1)
sns.boxplot('IMDB Score', data = df)
plt.xlabel('IMDB Score', fontsize = 15)


plt.figure(figsize=(10,5))
plt.subplot(3,1,2)
sns.boxplot('Runtime', data = df)
plt.xlabel('Runtime', fontsize = 15)


plt.figure(figsize=(10,5))
plt.subplot(3,1,3)
sns.boxplot('Year', data = df)
plt.xlabel('Year', fontsize = 15)
plt.show()
```

```
C:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(
C:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(
C:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(
```







From the above distribution,

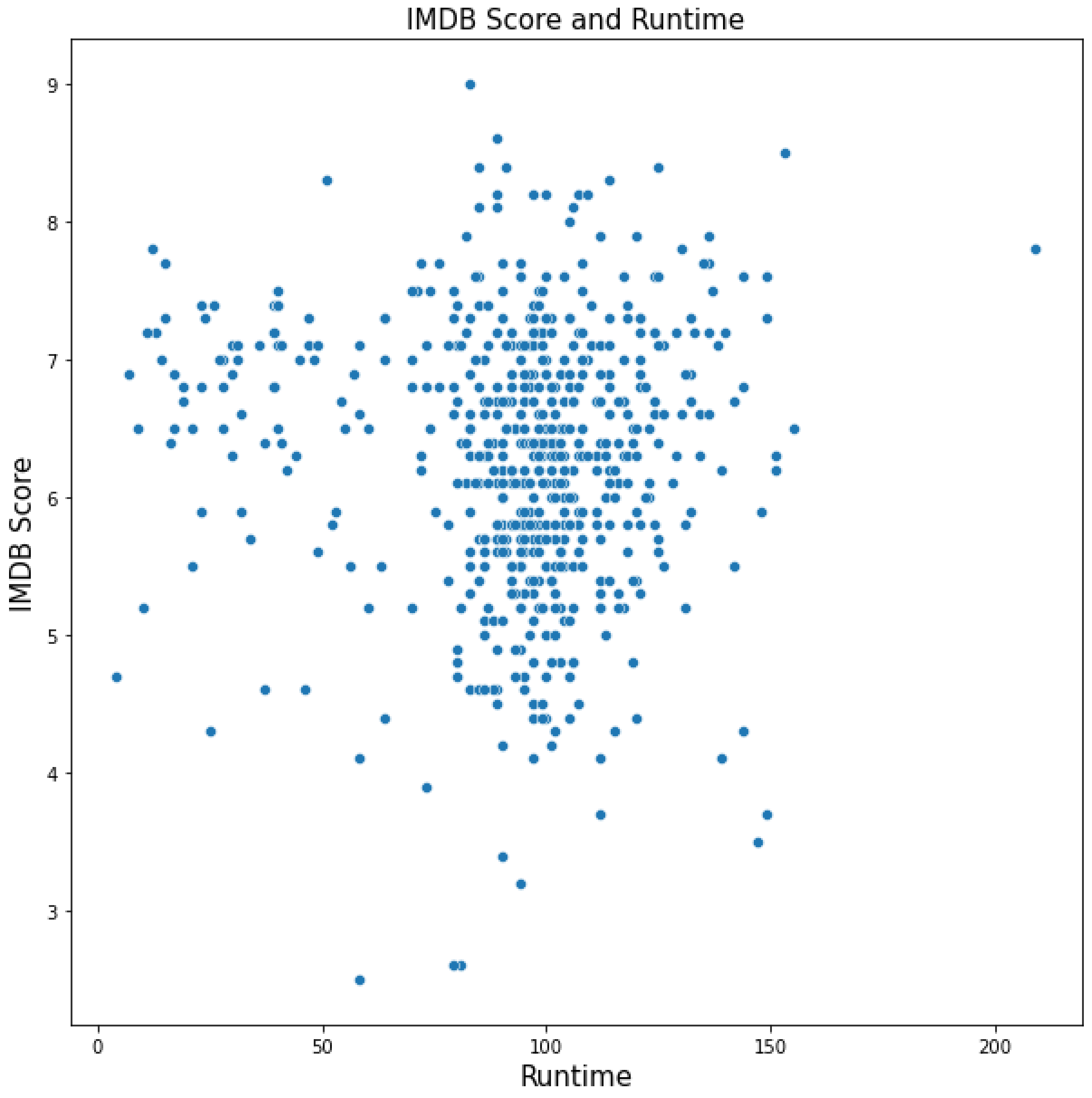
The first movie permiere in the netflix is from the year 2014,

The IMDB score minimum around 2 and maxmimum is around 9,

The maximum runtime is around 165 to 175.

Checking for correlation between IMDB and Runtime

In [11]:
```
plt.figure(figsize=(10,10))
sns.scatterplot(y = df['IMDB Score'], x = df['Runtime'])
plt.xlabel('Runtime', fontsize = 15)
plt.ylabel('IMDB Score', fontsize = 15)
plt.title('IMDB Score and Runtime', fontsize = 15)
plt.show()
```



From the above graph we found that there is no correlation between Imdb and Runtime

In [12]:
```
df.corr()
```

Out[12]:

	Runtime	IMDB Score	Day	Year
Runtime	1.000000	-0.040896	-0.024225	0.069262
IMDB Score	-0.040896	1.000000	0.057651	-0.141347
Day	-0.024225	0.057651	1.000000	-0.052283
Year	0.069262	-0.141347	-0.052283	1.000000

The correlation score:

1 : Strongly and positively correlated (one increases, other also increases and vice versa)

0 : No correlation

-1 : Strongly and negetively correlated (one increases, other also decreases and vice versa)

The value for IMDB and Runtime is -0.040896 so clear that there is no relation between these two features

seeing which genre has got high IMDB score

In [13]:
```
len(df['Genre'].unique())
```

Out[13]:
```
115
```

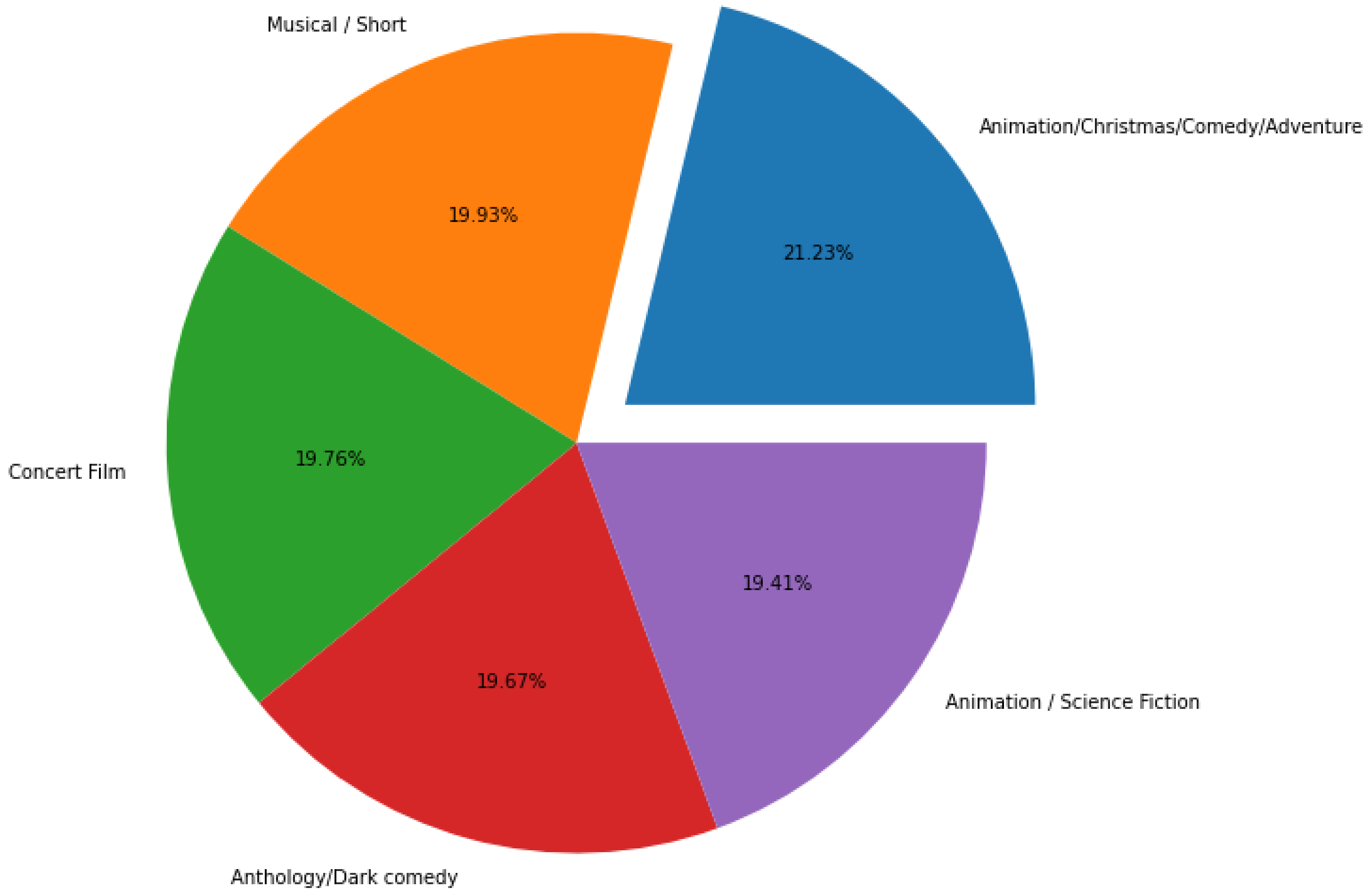
There are 115 genre are availabel so we are going see the top 5 genre based on imdb score

In [14]:
```
df_genre = df.groupby(['Genre']).mean(['IMDB Score']).sort_values(by='IMDB Score', ascending = False).reset_index().iloc[:5,:]
df_genre
```

Out[14]:

	Genre	Runtime	IMDB Score	Day	Year
0	Animation/Christmas/Comedy/Adventure	97.000000	8.200000	15.000000	2019.0
1	Musical / Short	15.000000	7.700000	27.000000	2019.0
2	Concert Film	98.666667	7.633333	21.833333	2018.5
3	Anthology/Dark comedy	149.000000	7.600000	12.000000	2020.0
4	Animation / Science Fiction	71.000000	7.500000	16.000000	2019.0

In [15]:
```
plt.figure(figsize=(5,5))
plt.pie(x = df_genre['IMDB Score'], labels = df_genre['Genre'], radius = 2, autopct = '%0.2f%%', explode = [0.3,0,0,0,0])
plt.show()
```



People like Animation/Christmas/Comedy/Adventure genre movies

In [16]:
```
df_Movies = df.sort_values(by = 'IMDB Score', ascending = False).iloc[:10,:]
df_Movies
```

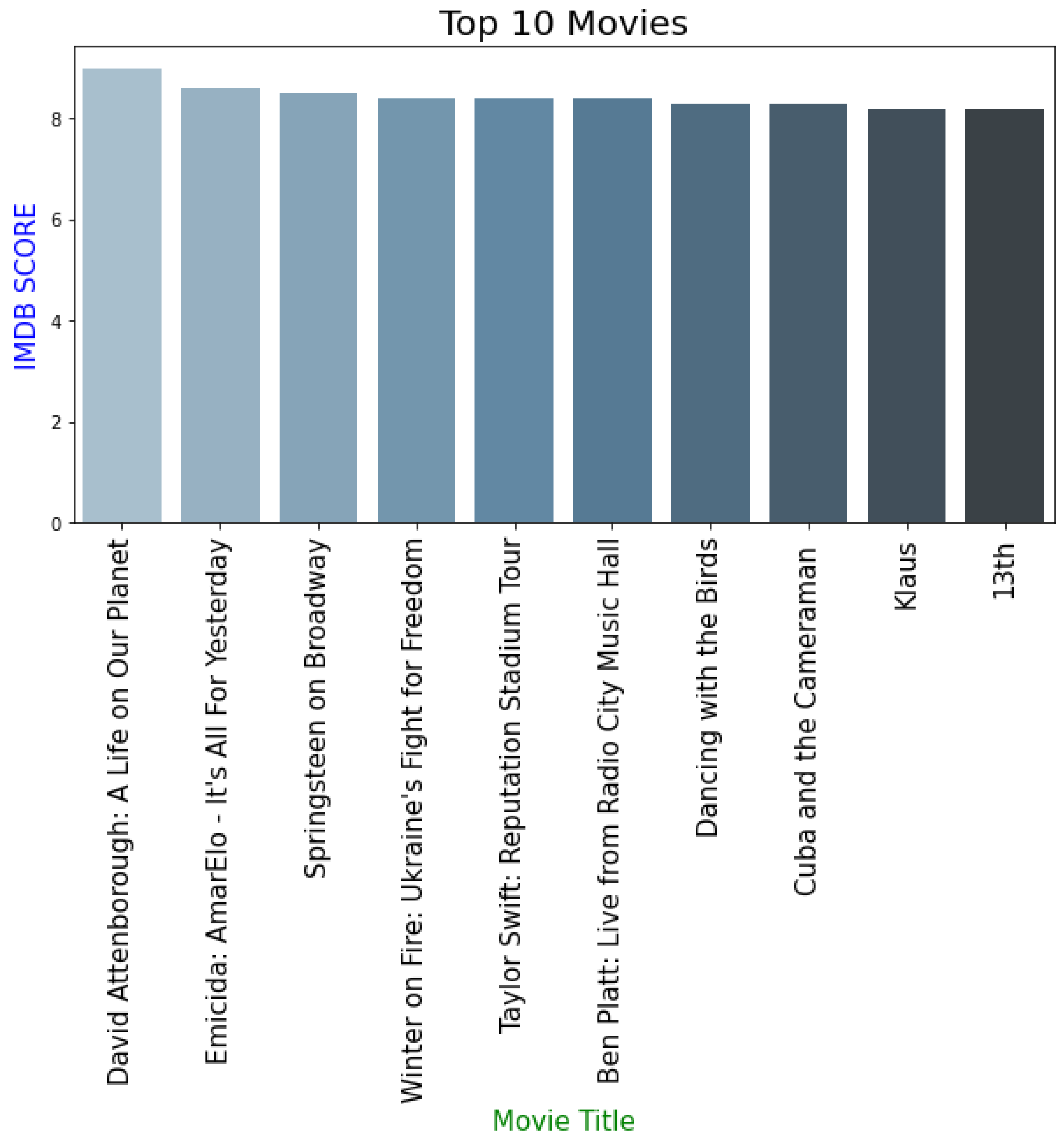
Out[16]:

	Title	Genre	Premiere	Runtime	IMDB Score	Language	Day	Month	Year
583	David Attenborough: A Life on Our Planet	Documentary	2020-10-04	83	9.0	English	4	Oct	2020
582	Emicida: AmarElo - It's All For Yesterday	Documentary	2020-12-08	89	8.6	Portuguese	8	Dec	2020
581	Springsteen on Broadway	One-man show	2018-12-16	153	8.5	English	16	Dec	2018
580	Winter on Fire: Ukraine's Fight for Freedom	Documentary	2015-10-09	91	8.4	English/Ukranian/Russian	9	Oct	2015
579	Taylor Swift: Reputation Stadium Tour	Concert Film	2018-12-31	125	8.4	English	31	Dec	2018
578	Ben Platt: Live from Radio City Music Hall	Concert Film	2020-05-20	85	8.4	English	20	May	2020
577	Dancing with the Birds	Documentary	2019-10-23	51	8.3	English	23	Oct	2019
576	Cuba and the Cameraman	Documentary	2017-11-24	114	8.3	English	24	Nov	2017
573	Klaus	Animation/Christmas/Comedy/Adventure	2019-11-15	97	8.2	English	15	Nov	2019
571	13th	Documentary	2016-10-07	100	8.2	English	7	Oct	2016

Top 10 movies based IMDB Score

In [17]:

```
plt.figure(figsize = (10,5))
x = plt.gca()
sns.barplot(x = 'Title', y = 'IMDB Score',palette="Blues_d", saturation=.5, data = df_Movies)
x.set_xticklabels(df_Movies['Title'], rotation = 90, fontsize = 15)
plt.title('Top 10 Movies', fontsize = 20)
plt.xlabel('Movie Title', fontsize = 15, color = 'g')
plt.ylabel('IMDB SCORE', fontsize = 15, color = 'b')
plt.show()
```



In [18]:

```
df_Movies['Genre'].value_counts()
```

Out[18]:

```
Documentary                           6
Concert Film                          2
One-man show                          1
Animation/Christmas/Comedy/Adventure    1
Name: Genre, dtype: int64
```

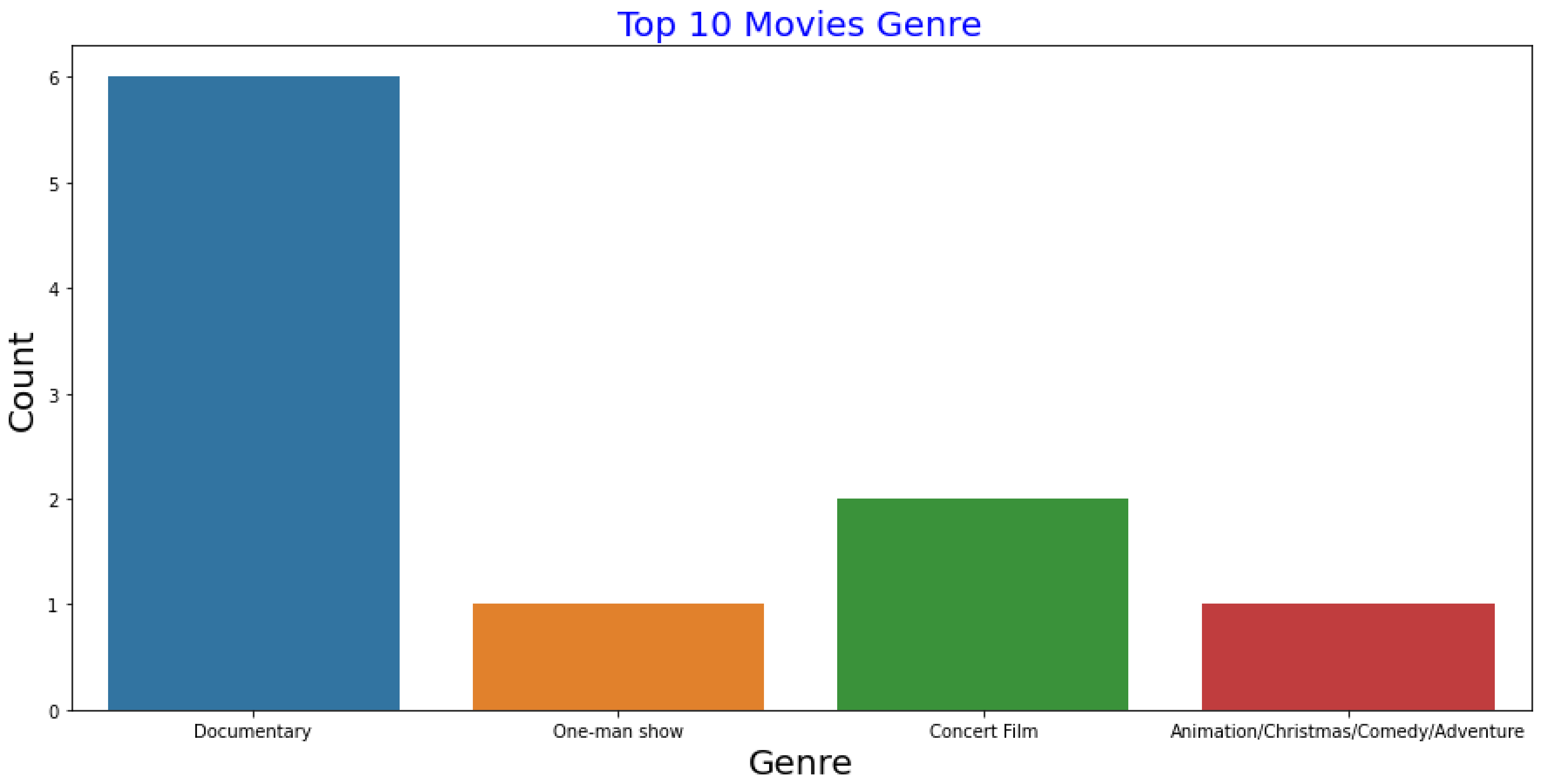
6 out 10 movies are Documentary genre movies, but this documentary genre are not listed in the top 5 genre movies.

The reason is most of documentary genre movies has very less Imdb score and only few of them got very high Imdb score.

Let's explore...

```
In [19]: plt.figure(figsize=(15,7))
         sns.countplot(df_Movies['Genre'])
         plt.title('Top 10 Movies Genre', fontsize = 20, color = 'blue')
         plt.xlabel('Genre', fontsize = 20)
         plt.ylabel('Count', fontsize = 20)
         plt.show()
```

```
C:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(
```



Let's seet how many Documentary movies having IMDB score above 8.0

In [20]:

```
df_doc_high = df[(df['Genre'] == 'Documentary') & (df['IMDB Score'] >= 8.0)]
df_doc_high.sort_values(by = 'IMDB Score', ascending = False)
```

Out[20]:

	Title	Genre	Premiere	Runtime	IMDB Score	Language	Day	Month	Year
583	David Attenborough: A Life on Our Planet	Documentary	2020-10-04	83	9.0	English	4	Oct	2020
582	Emicida: AmarElo - It's All For Yesterday	Documentary	2020-12-08	89	8.6	Portuguese	8	Dec	2020
580	Winter on Fire: Ukraine's Fight for Freedom	Documentary	2015-10-09	91	8.4	English/Ukranian/Russian	9	Oct	2015
576	Cuba and the Cameraman	Documentary	2017-11-24	114	8.3	English	24	Nov	2017
577	Dancing with the Birds	Documentary	2019-10-23	51	8.3	English	23	Oct	2019
571	13th	Documentary	2016-10-07	100	8.2	English	7	Oct	2016
572	Disclosure: Trans Lives on Screen	Documentary	2020-06-19	107	8.2	English	19	Jun	2020
574	Seaspiracy	Documentary	2021-03-24	89	8.2	English	24	March	2021
575	The Three Deaths of Marisela Escobedo	Documentary	2020-10-14	109	8.2	Spanish	14	Oct	2020
568	Chasing Coral	Documentary	2017-07-14	89	8.1	English	14	July	2017
569	My Octopus Teacher	Documentary	2020-09-07	85	8.1	English	7	Sep	2020
570	Rising Phoenix	Documentary	2020-08-26	106	8.1	English	26	Aug	2020
567	Struggle: The Life and Lost Art of Szukaiski	Documentary	2018-12-21	105	8.0	English	21	Dec	2018

In [21]:

```
len(df_doc_high)
```

Out[21]:

```
13
```

There are 13 documentary movies had got IMDB score above 8.0

In [22]:

```
df_doc_low = df[(df['Genre'] == 'Documentary') & (df['IMDB Score'] < 8.0)]
df_doc_low.sort_values(by = 'IMDB Score')
```

Out[22]:

	Title	Genre	Premiere	Runtime	IMDB Score	Language	Day	Month	Year
0	Enter the Anime	Documentary	2019-08-05	58	2.5	English/Japanese	5	Aug	2019
10	Searching for Sheela	Documentary	2021-04-22	58	4.1	English	22	Apr	2021
15	After the Raid	Documentary	2019-12-19	25	4.3	Spanish	19	Dec	2019
20	Hello Privilege. It's Me, Chelsea	Documentary	2019-09-13	64	4.4	English	13	Sep	2019
30	After Maria	Documentary	2019-05-24	37	4.6	English/Spanish	24	May	2019
...	...	...	...	...	...	...	...	...	...
555	Crip Camp: A Disability Revolution	Documentary	2020-03-25	108	7.7	English	25	March	2020
556	Jim & Andy: The Great Beyond - Featuring a Ver...	Documentary	2017-11-17	94	7.7	English	17	Nov	2017
564	Icarus	Documentary	2017-08-04	120	7.9	English	4	Aug	2017
563	A Secret Love	Documentary	2020-04-29	82	7.9	English	29	Apr	2020
566	The Ivory Game	Documentary	2016-11-04	112	7.9	English	4	Nov	2016

146 rows × 9 columns

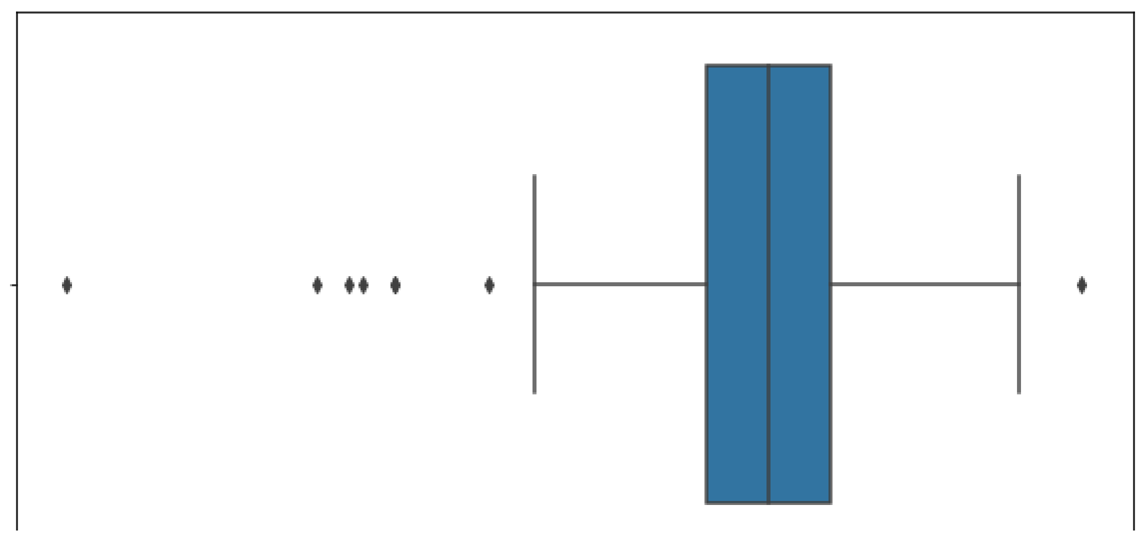
There are 146 documentary movies had IMDB Score less than 8.0

From here we can say that majority of the documentary movies had got less IMDB score, but few documentary movies had got high IMDB score between 8 to 9

Let's visualize IMDB Score for Documentary movies

In [23]:
```
df_doc = df[df['Genre']=='Documentary']

plt.figure(figsize=(10,5))
sns.boxplot('IMDB Score', data = df_doc)
plt.xlabel("IMDB Score", fontsize = 15)
plt.show()
```

```
C:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```



we can see that only few documentary movies have high IMDB score, and most of the movies having low IMDB score. This made

documentary genre taking out from the top 5 genre

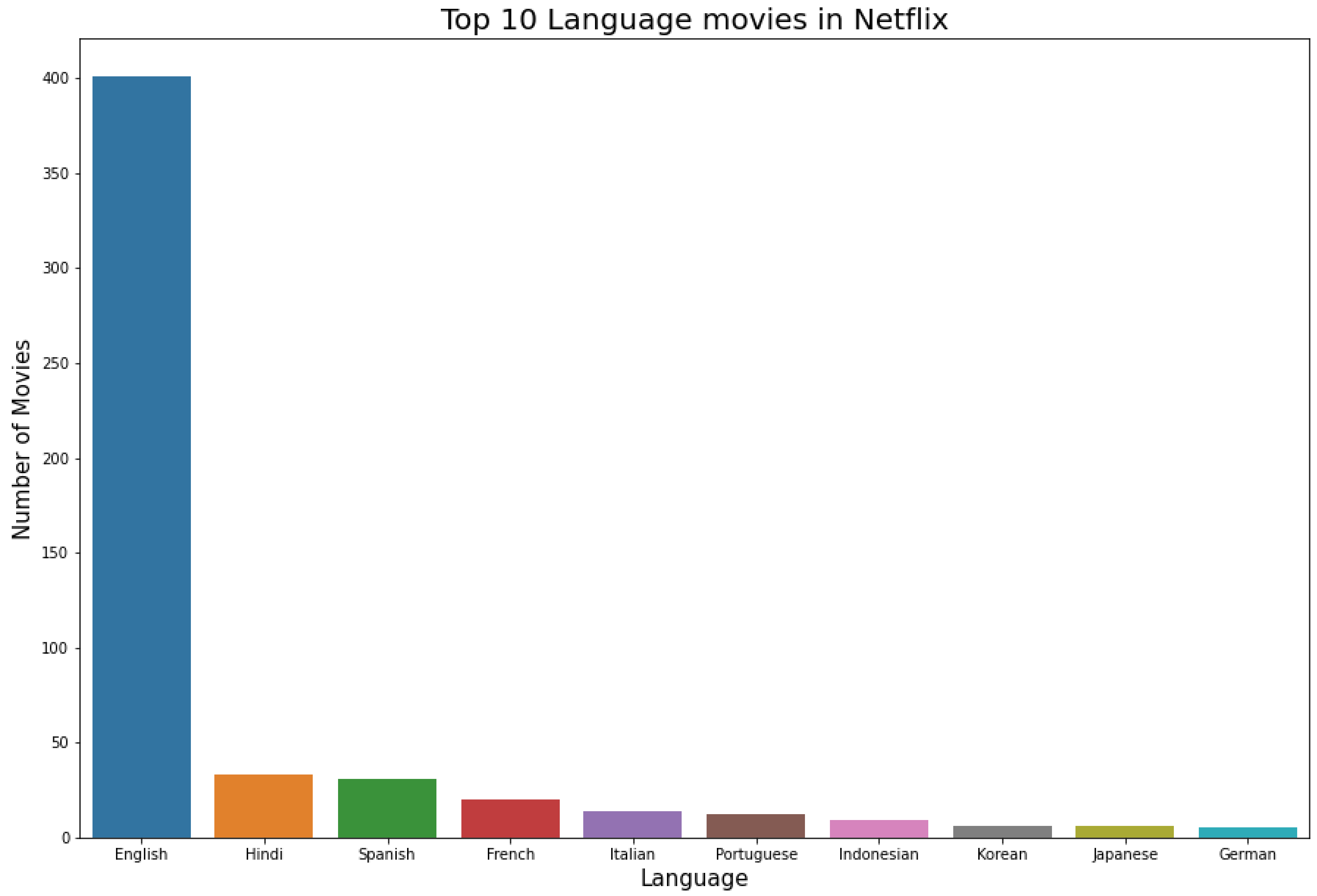
Let's explore Languages

In [24]:
```
netflix_movie_lang = len(df['Language'].unique())
print('There are ' + str(netflix_movie_lang) + ' language movies in Netflix')
```

```
There are 38 language movies in Netflix
```

The top 10 languages movies in Netflix

```
In [25]: df_lang = df.groupby('Language')['Title'].count().reset_index().sort_values(by='Title', ascending = False).iloc[:10,:]

         plt.figure(figsize=(15,10))
         sns.barplot(x = 'Language', y = 'Title', data = df_lang)
         plt.xlabel('Language', fontsize = 15)
         plt.ylabel('Number of Movies', fontsize = 15)
         plt.title('Top 10 Language movies in Netflix', fontsize = 20)
         plt.show()
```

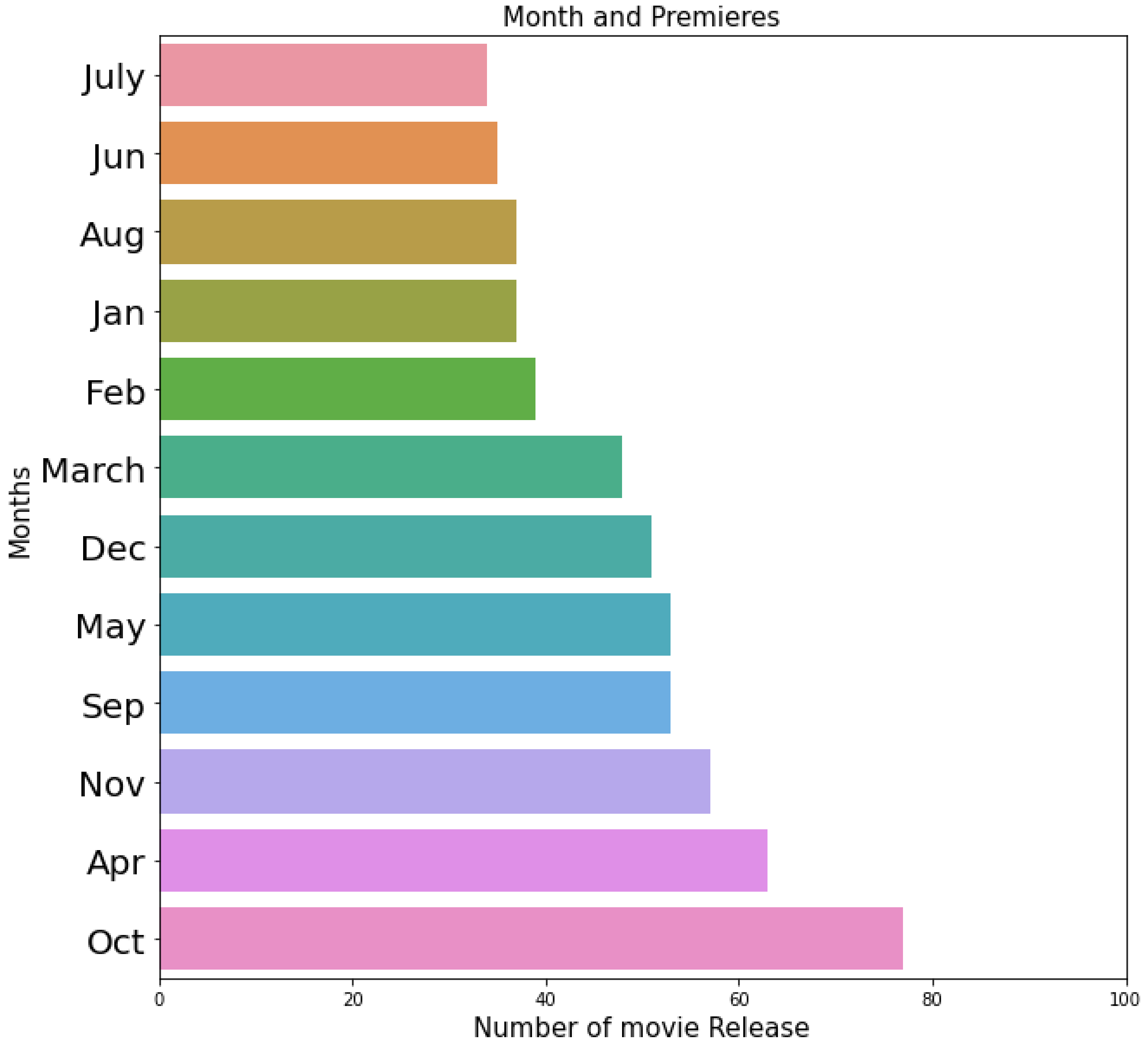


From the above graph we can see that english movies are released mostly in netflix

Let' s see in which month mostly the movie releases

In [26]:

```
df_release = df.groupby('Month')['Title'].count().reset_index().sort_values(by='Title')
x = list(np.arange(0,100,10))

plt.figure(figsize = (10,10))
x = plt.gca()
sns.barplot(x = 'Title', y = 'Month', orientation = 'horizontal', data = df_release)
x.set_yticklabels(df_release['Month'],fontsize = 20)
plt.xlim(0,100)
plt.xlabel('Number of movie Release',fontsize = 15)
plt.ylabel('Months',fontsize = 15)
plt.title('Month and Premieres',fontsize = 15)
plt.show()
```



The more movies releases are in the month october and followed by april, i think reason can be the holidays during these months.

let's see how many good movies are released in netflix

good movies should have IMDB Score 7 and above.i took based on my search, i also provided link for this

https://www.imdb.com/poll/MGIr14gyJNE/ (https://www.imdb.com/poll/MGIr14gyJNE/)

In [27]:

```
Threshold = 7

df['Best_Movie'] = df['IMDB Score'].apply(lambda x : 1 if x > Threshold else 0)
```

0 : if the IMDB score less then 7

1 : if the IMDB score is higher then 7

In [28]:

```
df
```

Out[28]:

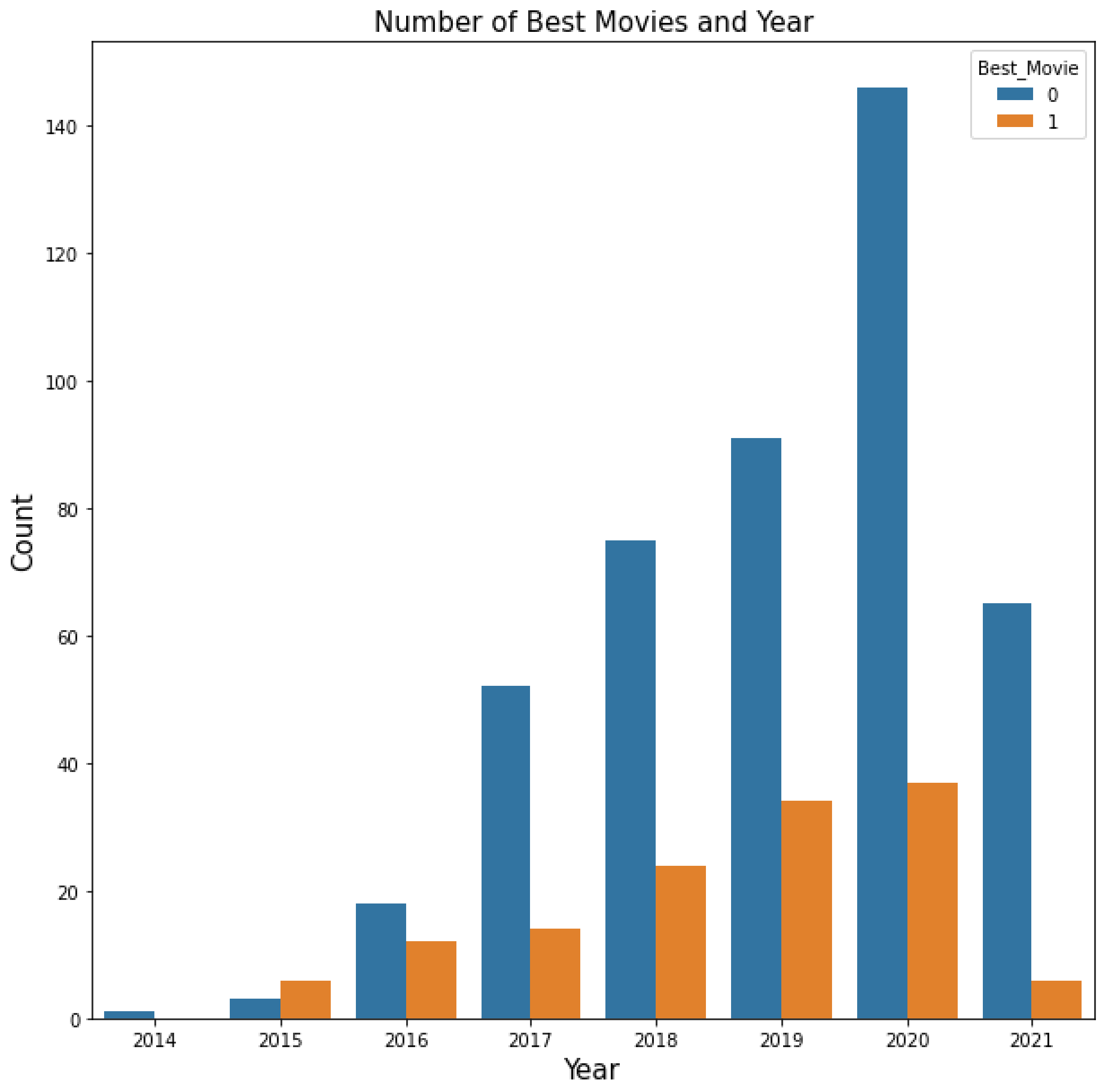
	Title	Genre	Premiere	Runtime	IMDB Score	Language	Day	Month	Year	Best_Movie
0	Enter the Anime	Documentary	2019-08-05	58	2.5	English/Japanese	5	Aug	2019	0
1	Dark Forces	Thriller	2020-08-21	81	2.6	Spanish	21	Aug	2020	0
2	The App	Science fiction/Drama	2019-12-26	79	2.6	Italian	26	Dec	2019	0
3	The Open House	Horror thriller	2018-01-19	94	3.2	English	19	Jan	2018	0
4	Kaali Khuhi	Mystery	2020-10-30	90	3.4	Hindi	30	Oct	2020	0
...	...	...	...	...	...	...	...	...	...	...
579	Taylor Swift: Reputation Stadium Tour	Concert Film	2018-12-31	125	8.4	English	31	Dec	2018	1
580	Winter on Fire: Ukraine's Fight for Freedom	Documentary	2015-10-09	91	8.4	English/Ukranian/Russian	9	Oct	2015	1
581	Springsteen on Broadway	One-man show	2018-12-16	153	8.5	English	16	Dec	2018	1
582	Emicida: AmarElo - It's All For Yesterday	Documentary	2020-12-08	89	8.6	Portuguese	8	Dec	2020	1
583	David Attenborough: A Life on Our Planet	Documentary	2020-10-04	83	9.0	English	4	Oct	2020	1

584 rows × 10 columns

Let's see how many good movies are released over the year in netflix

```
In [29]: plt.figure(figsize = (10,10))
         sns.countplot('Year',hue = 'Best_Movie', data = df)
         plt.xlabel('Year', fontsize = 15)
         plt.ylabel('Count', fontsize = 15)
         plt.title('Number of Best Movies and Year', fontsize = 15)
         plt.show()
```

```
C:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(
```



The proportion for good movies are low in the year 2021, i hope netflix will give good movies for upcoming years

I like mostly Thiller Genre movies

```
In [30]: df_fav = df[df['Genre'] == 'Thriller']
```

let's see in which language most of the tiller movies are availabel in the netflix

In [31]:

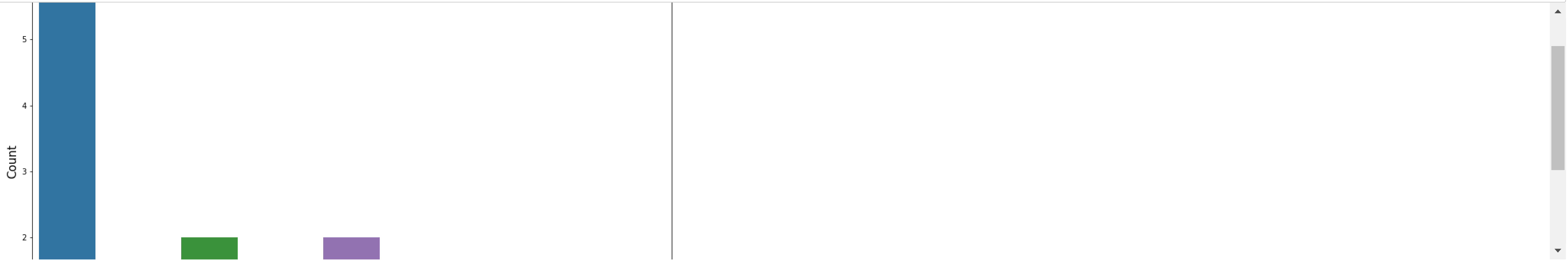
```
df_fava = df_fav.groupby(['Language', 'Year'])['Title'].count().reset_index()
df_fava
```

Out[31]:

	Language	Year	Title
0	English	2016	2
1	English	2017	1
2	English	2018	2
3	English	2019	4
4	English	2020	4
5	English	2021	1
6	Filipino	2019	1
7	French	2018	1
8	French	2020	2
9	German	2020	1
10	Hindi	2020	4
11	Hindi	2021	1
12	Korean	2020	1
13	Polish	2021	1
14	Spanish	2020	6
15	Swedish	2021	1

In [32]:

```
plt.figure(figsize = (15,10))

sns.countplot(x = 'Language', data = df_fava)
plt.xlabel('Language', fontsize = 15)
plt.ylabel('Count', fontsize = 15)

ax = plt.gca().xaxis
for i in ax.get_ticklabels():
    i.set_fontsize(15)

plt.show()
```



From the above graph we can see that tiller movies are mostly in English language...