Netflix Data Analysis

The aim while analyzing the data is to generate insights that can help in deciding;
-Which type of shows or movies to produce
-The growth areas of the business in different countries.

Steps

1. Importing Libraries
2. Loading the dataset:
   Load Data
   Undersranding the structure of the data
3. Data Cleaning:
   Changing data types
   Deleting redundant columns.
   Cleaning individual columns by filling NaN values and dropping.
   Some Transformations
4. Analysing and Data Visualization:
   Statistical analysis
   Visualization
5. Word Cloud:
   Country
   Title

1. Importing Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

2. Loading The Dataset

The data is converted to csv in order to read faster and then loaded. The data has 8807 rows × 12 columns.
We see there are errors in our data, such as NaN values.

The cast, country and director have more than one values within them; which are seperated by the comma delimiter.

In [2]:
```
df = pd.read_csv('Netflix data.csv')
df
```

Out[2]:

	show_id	type	title	director	cast	country	date_added	release_year	ratin
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	25-Sep-21	2020	PG-1
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	24-Sep-21	2021	TV M
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	24-Sep-21	2021	TV M
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	24-Sep-21	2021	TV M
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	24-Sep-21	2021	TV M
...	...	...	...	...	...	...	...	...	
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	20-Nov-19	2007	
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	1-Jul-19	2018	TV-Y
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	1-Nov-19	2009	
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	11-Jan-20	2006	P

	show_id	type	title	director	cast	country	date_added	release_year	ratin
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	2-Mar-19	2015	TV-1

8807 rows × 12 columns

The info() function is used to give us a snapshot of the type of data that is in our DataFrame and so that we are able to change data into proper data types and also to know the specific number of null values in each column.
Rating has 4 null values, duration has 3 null values ;
Whereas cast,country and director has significant number of cells with null values.

In [3]:
```
df.info()
```

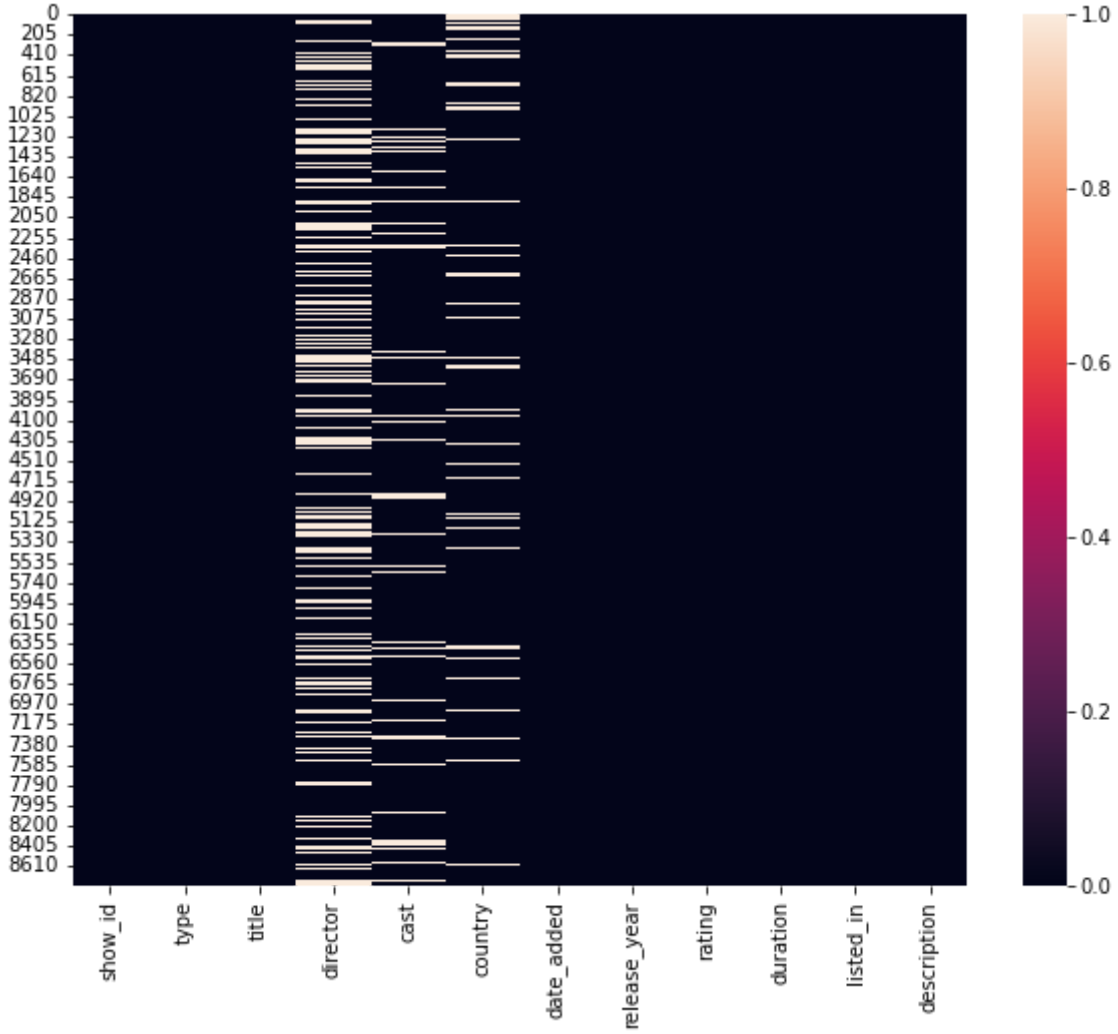
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64 
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

We plot a heatmap of the null values.

We see that the Director, cast and country columns need to be cleaned. Normally a sample size of over 100 is quite good. But in our case, Removing the NaN Values in the rows;(by drop.na) removed a significant number of Tv shows leading the data to being biased.

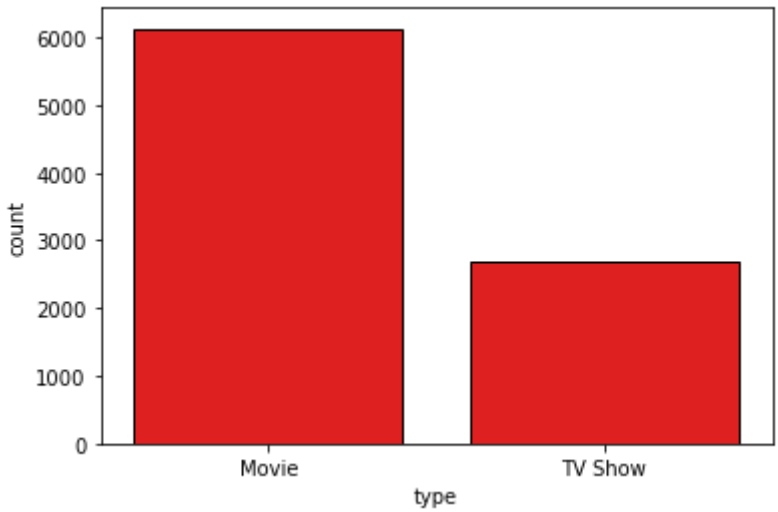
We will however drop the director and cast columns, as they are not significant in our study now. The country column which is significant, the null values shall be filled.

In [4]:
```
plt.figure(figsize=(10,8))
sns.heatmap(df.isna());
```



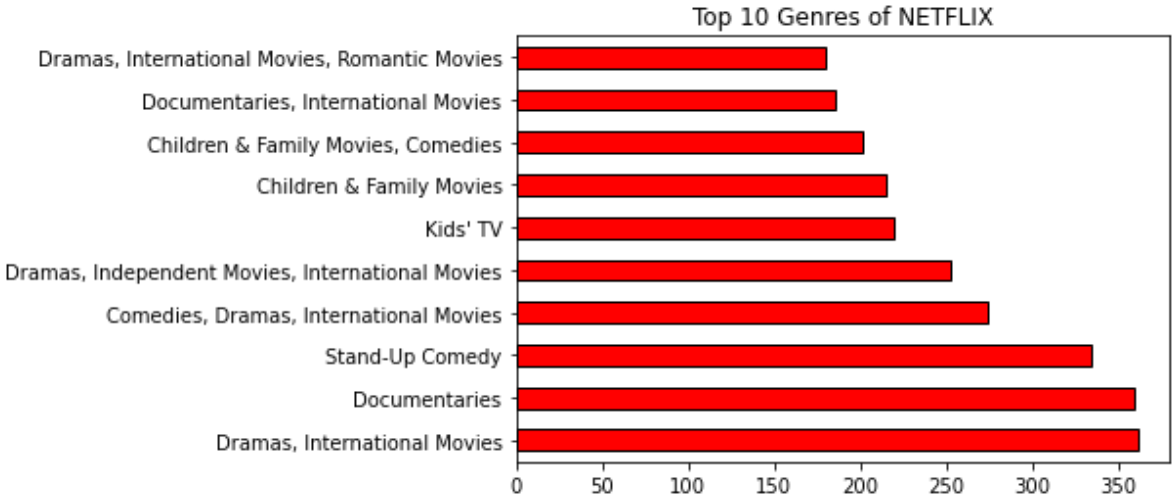
The distribution of Tv shows and Movie before manipulation of the data.

In [5]:
```
sns.countplot(x= 'type', data=df, color='Red', edgecolor='black');
```



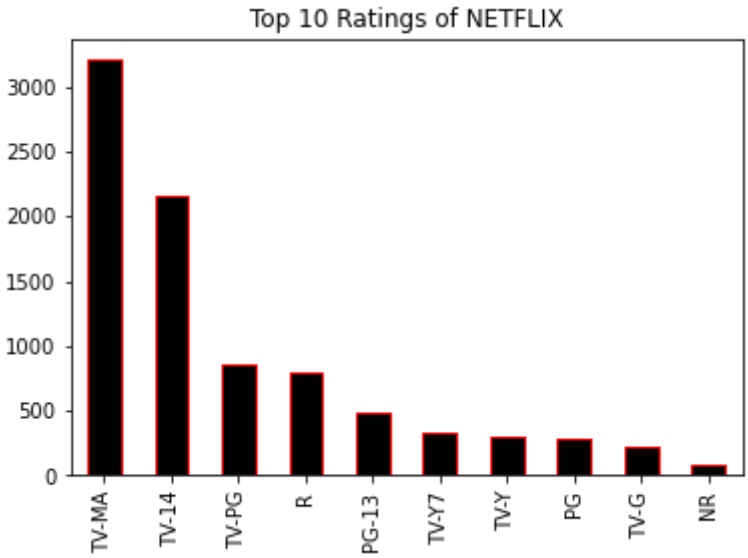
Top ten genres of the content listed, present in Netflix.

In [6]:
```
df["listed_in"].value_counts()[:10].plot(kind="barh", color="red", edgecolor='blacl
plt.title("Top 10 Genres of NETFLIX");
```



Top ten rating of the content listed, present in Netflix.

In [7]:
```
df["rating"].value_counts()[:10].plot(kind="bar", color="black", edgecolor='red')
plt.title("Top 10 Ratings of NETFLIX");
```



3. Data Cleaning

Changing the datatype of date_added to datetime

In [8]:
```
pd.to_datetime(df['date_added'])
```

Out[8]:
```
0      2021-09-25
1      2021-09-24
2      2021-09-24
3      2021-09-24
4      2021-09-24
          ...
8802   2019-11-20
8803   2019-07-01
8804   2019-11-01
8805   2020-01-11
8806   2019-03-02
Name: date_added, Length: 8807, dtype: datetime64[ns]
```

In [9]:
```
df['date_added'] = pd.to_datetime(df['date_added'])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       8807 non-null   object        
 1   type          8807 non-null   object        
 2   title         8807 non-null   object        
 3   director      6173 non-null   object        
 4   cast          7982 non-null   object        
 5   country       7976 non-null   object        
 6   date_added    8797 non-null   datetime64[ns]
 7   release_year  8807 non-null   int64         
 8   rating        8803 non-null   object        
 9   duration      8804 non-null   object        
 10  listed_in     8807 non-null   object        
 11  description   8807 non-null   object        
dtypes: datetime64[ns](1), int64(1), object(10)
memory usage: 825.8+ KB
```

#pd.to_timedelta(df['duration'])

We Drop the Cast and directors columns; as they are not significant in our study now. The country column which is significant, the null values shall be filled.

In [10]:
```
df.drop(['director', 'cast'], axis=1, inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       8807 non-null   object        
 1   type          8807 non-null   object        
 2   title         8807 non-null   object        
 3   country       7976 non-null   object        
 4   date_added    8797 non-null   datetime64[ns]
 5   release_year  8807 non-null   int64         
 6   rating        8803 non-null   object        
 7   duration      8804 non-null   object        
 8   listed_in     8807 non-null   object        
 9   description   8807 non-null   object        
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 688.2+ KB
```

To find out the number of nun unique values in each category.

From the code below; we see that there are two data types...i.e Movie and Tv shows. and from the plot of distribution of tvshows vs movies,

we saw that there is a large variance between the two, hence we divide the data into the two types for analysis to avoid bias.

In [11]:
```
df.nunique()
```

Out[11]:
```
show_id         8807
type               2
title           8804
country          748
date_added      1714
release_year      74
rating            17
duration         220
listed_in        514
description     8775
dtype: int64
```

TV Shows

We start our Data Cleaning & Analysis with Tv Shows
Make a copy of the original dataframe and call it df1

In [12]:
```
df1 = df.copy()
df1 = df1[df1['type'] == 'TV Show']

df1.head()
```

Out[12]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in
1	s2	TV Show	Blood & Water	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries
2	s3	TV Show	Ganglands	NaN	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...
3	s4	TV Show	Jailbirds New Orleans	NaN	2021-09-24	2021	TV-MA	1 Season	Docuseries, Reality TV
4	s5	TV Show	Kota Factory	India	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...
5	s6	TV Show	Midnight Mass	NaN	2021-09-24	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries

In [13]:
```
df1[df1['country'].isnull()]
```

Out[13]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_i
2	s3	TV Show	Ganglands	NaN	2021-09-24	2021	TV-MA	1 Season	Crime T Show Internation TV Show TV Act
3	s4	TV Show	Jailbirds New Orleans	NaN	2021-09-24	2021	TV-MA	1 Season	Docuserie Reality T
5	s6	TV Show	Midnight Mass	NaN	2021-09-24	2021	TV-MA	1 Season	TV Drama TV Horro TV Mysteri
10	s11	TV Show	Vendetta: Truth, Lies and The Mafia	NaN	2021-09-24	2021	TV-MA	1 Season	Crime T Show Docuserie Internation TV S
11	s12	TV Show	Bangkok Breaking	NaN	2021-09-23	2021	TV-MA	1 Season	Crime T Show Internation TV Show TV Act
...	...	...	...	...	...	...	...	...	
8679	s8680	TV Show	ViR: The Robot Boy	NaN	2018-03-31	2013	TV-Y7	2 Seasons	Kids' T
8690	s8691	TV Show	Wake Up	NaN	2018-03-31	2017	TV-14	2 Seasons	Internation TV Show TV Drama
8783	s8784	TV Show	Yoko	NaN	2018-06-23	2016	TV-Y	1 Season	Kids' T
8785	s8786	TV Show	YOM	NaN	2018-06-07	2016	TV-Y7	1 Season	Kids' T
8803	s8804	TV Show	Zombie Dumb	NaN	2019-07-01	2018	TV-Y7	2 Seasons	Kids' T Korean T Shows, T Comedi

391 rows × 10 columns

We forward fill(ffill) the na values of the country assuming that the data was acquired chronologically per country. forward fill takes the last value preceding the null value and fills it.We use axis 0 to take in te first country, for places with multiple countries in the list.

We then run the syntax is null to see if the results have been effective. And it returns an empty dataframe.

In [14]:
```
df1['country'] = df1['country'].ffill(axis=0)
df1[df1['country'].isnull()]
```

Out[14]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description

In [15]:
```
df1[df1['rating'].isnull()]
```

Out[15]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_i
6827	s6828	TV Show	Gargantia on the Verdurous Planet	Japan	2016-12-01	2013	NaN	1 Season	Anim Serie Internation TV Shov
7312	s7313	TV Show	Little Lunch	Australia	2018-02-01	2015	NaN	1 Season	Kids' TV, T Comedi

at function is similar to the iloc and loc for identiifying a particular cell and then filling in with the desired output.

In this case, since the null values are only 2, It was wise to google and manually fill in the data using the at. function.

In [17]:
```
df1.at[6827,'rating'] = 'TV-14'
df1.at[7312,'rating'] = 'TV-MA'
df1[df1['rating'].isnull()]
```

Out[17]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description

The date_added still has 10 null values.

In [16]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2676 entries, 1 to 8803
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       2676 non-null   object        
 1   type          2676 non-null   object        
 2   title         2676 non-null   object        
 3   country       2676 non-null   object        
 4   date_added    2666 non-null   datetime64[ns]
 5   release_year  2676 non-null   int64         
 6   rating        2674 non-null   object        
 7   duration      2676 non-null   object        
 8   listed_in     2676 non-null   object        
 9   description   2676 non-null   object        
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 230.0+ KB
```

We drop the null values of the specific column of date added.

In [18]:
```
df1 = df1.dropna(axis=0, subset=['date_added'])
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2666 entries, 1 to 8803
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       2666 non-null   object        
 1   type          2666 non-null   object        
 2   title         2666 non-null   object        
 3   country       2666 non-null   object        
 4   date_added    2666 non-null   datetime64[ns]
 5   release_year  2666 non-null   int64         
 6   rating        2666 non-null   object        
 7   duration      2666 non-null   object        
 8   listed_in     2666 non-null   object        
 9   description   2666 non-null   object        
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 229.1+ KB
```

Our data is pretty much clean.

The only problem we might face is the country column that has more than one value in it. Since there are about 500 columns having the country with more than one country, we split the columns, then create a new column having only one country and choosing the first country as the country_made.

Step 1. create an empty column to a dataframe

In [19]:
```
df1['country_made'] = df1.apply(lambda _: '', axis=1)
df1
```

Out[19]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed
1	s2	TV Show	Blood & Water	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Internatic TV Sho TV Dran TV Myste
2	s3	TV Show	Ganglands	South Africa	2021-09-24	2021	TV-MA	1 Season	Crime Sho Internatic TV Sho TV A
3	s4	TV Show	Jailbirds New Orleans	South Africa	2021-09-24	2021	TV-MA	1 Season	Docuser Reality
4	s5	TV Show	Kota Factory	India	2021-09-24	2021	TV-MA	2 Seasons	Internatic TV Sho Romantic Shows, T
5	s6	TV Show	Midnight Mass	India	2021-09-24	2021	TV-MA	1 Season	TV Dran TV Hor TV Myste
...	...	...	...	...	...	...	...	...	
8795	s8796	TV Show	Yu-Gi-Oh! Arc-V	Japan, Canada	2018-05-01	2015	TV-Y7	2 Seasons	Ani Series, K
8796	s8797	TV Show	Yunus Emre	Turkey	2017-01-17	2016	TV-PG	2 Seasons	Internatic TV Sho TV Dran
8797	s8798	TV Show	Zak Storm	United States, France, South Korea, Indonesia	2018-09-13	2016	TV-Y7	3 Seasons	Kids'
8800	s8801	TV Show	Zindagi Gulzar Hai	Pakistan	2016-12-15	2012	TV-PG	1 Season	Internatic TV Sho Romantic Shows, T
8803	s8804	TV Show	Zombie Dumb	Pakistan	2019-07-01	2018	TV-Y7	2 Seasons	Kids' Korean Shows, Comec

2666 rows × 11 columns

Step 2. Fill the new column with the Values from splitinng the country column by delimiter.

In [20]:
```
df1['country_made'] = [x.split(',')[-0] for x in df1['country']]
df1
```

Out[20]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed
1	s2	TV Show	Blood & Water	South Africa	2021-09-24	2021	TV-MA	2 Seasons	Internatic TV Sho TV Dran TV Myste
2	s3	TV Show	Ganglands	South Africa	2021-09-24	2021	TV-MA	1 Season	Crime Sho Internatic TV Sho TV A
3	s4	TV Show	Jailbirds New Orleans	South Africa	2021-09-24	2021	TV-MA	1 Season	Docuser Reality
4	s5	TV Show	Kota Factory	India	2021-09-24	2021	TV-MA	2 Seasons	Internatic TV Sho Romantic Shows, T
5	s6	TV Show	Midnight Mass	India	2021-09-24	2021	TV-MA	1 Season	TV Dran TV Hor TV Myste
...	...	...	...	...	...	...	...	...	
8795	s8796	TV Show	Yu-Gi-Oh! Arc-V	Japan, Canada	2018-05-01	2015	TV-Y7	2 Seasons	Ani Series, K
8796	s8797	TV Show	Yunus Emre	Turkey	2017-01-17	2016	TV-PG	2 Seasons	Internatic TV Sho TV Dran
8797	s8798	TV Show	Zak Storm	United States, France, South Korea, Indonesia	2018-09-13	2016	TV-Y7	3 Seasons	Kids'
8800	s8801	TV Show	Zindagi Gulzar Hai	Pakistan	2016-12-15	2012	TV-PG	1 Season	Internatic TV Sho Romantic Shows, T
8803	s8804	TV Show	Zombie Dumb	Pakistan	2019-07-01	2018	TV-Y7	2 Seasons	Kids' Korean Shows, Comec

2666 rows × 11 columns

Analysis and Visualization

Statistical Analysis

In [21]:
```
df1['duration'].mode()
```

Out[21]:
```
0    1 Season
Name: duration, dtype: object
```

In [22]:
```
df1['rating'].mode()
```

Out[22]:
```
0    TV-MA
Name: rating, dtype: object
```

In [23]:
```
df1['listed_in'].mode()
```

Out[23]:
```
0    Kids' TV
Name: listed_in, dtype: object
```

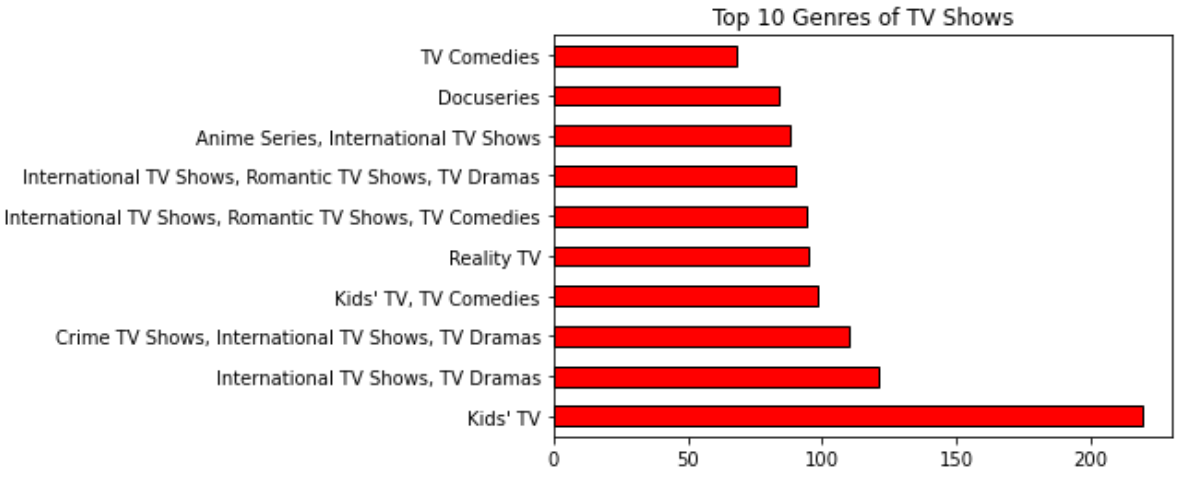
In [24]:
```
df1['country_made'].mode()
```

Out[24]:
```
0    United States
Name: country_made, dtype: object
```

Visualization

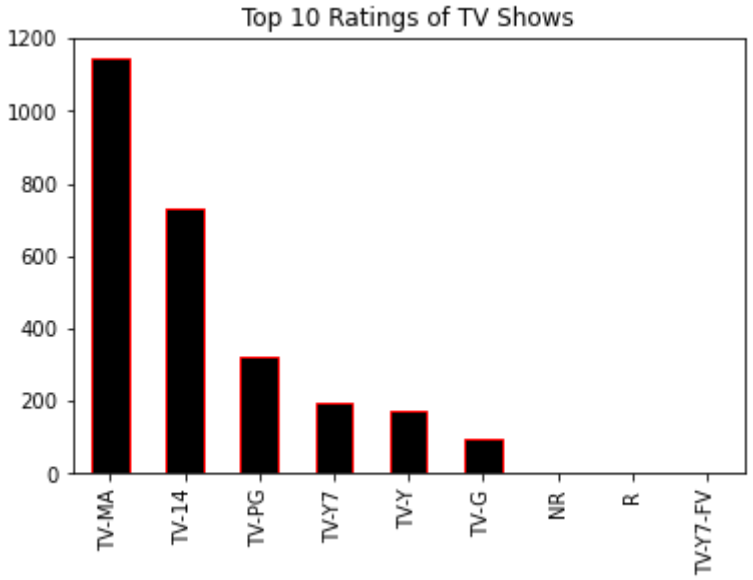
Top 10 Genres of TV Shows

In [25]:
```
df1["listed_in"].value_counts()[:10].plot(kind="barh", color="red", edgecolor='bla
plt.title("Top 10 Genres of TV Shows");
```



Top 10 Ratings of TV Shows

In [26]:
```
df1["rating"].value_counts()[:10].plot(kind="bar", color="black", edgecolor='red')
plt.title("Top 10 Ratings of TV Shows");
```

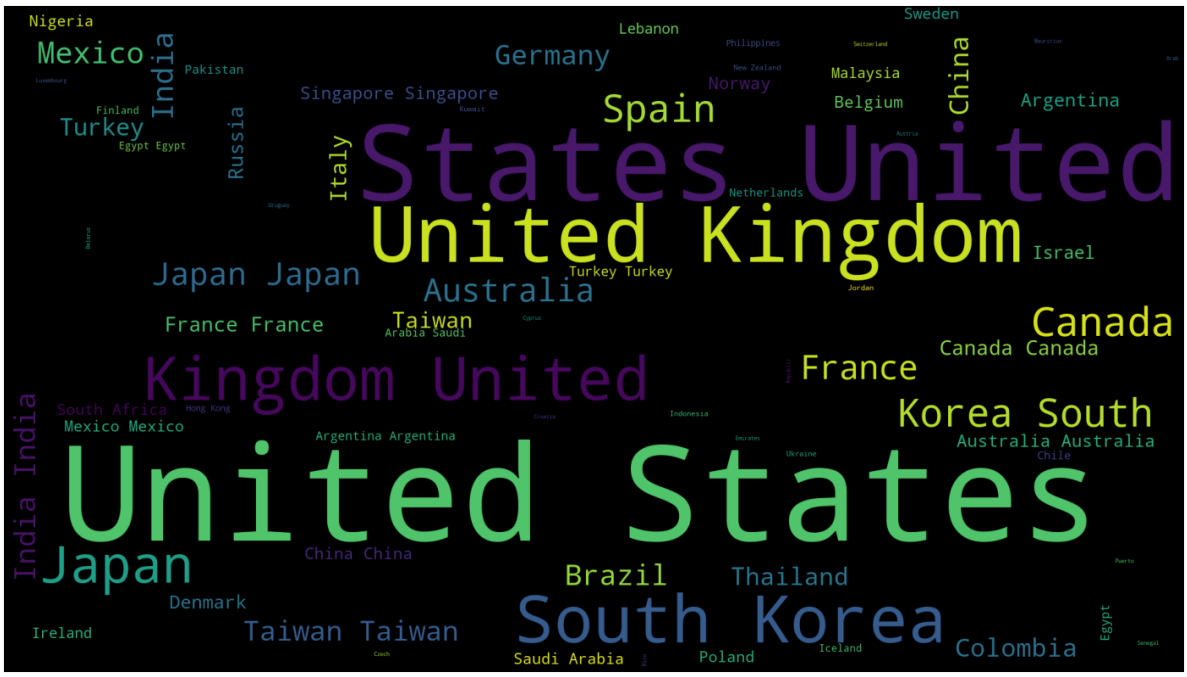


Word Cloud TV Show

In [27]:
```
from wordcloud import WordCloud
```

Word Cloud for Country

In [28]:
```
plt.subplots(figsize=(25,15))
wordcloud = WordCloud(
                          background_color='black',
                          width=1920,
                          height=1080
                         ).generate(" ".join(df1.country_made))
plt.imshow(wordcloud)
plt.axis('off')
plt.savefig('country_made.png')
plt.show()
```



Word Cloud for Title

```
In [29]: plt.subplots(figsize=(25,15))
         wordcloud = WordCloud(
                                   background_color='black',
                                   width=1920,
                                   height=1080
                                  ).generate(" ".join(df1.title))
         plt.imshow(wordcloud)
         plt.axis('off')
         plt.savefig('title.png')
         plt.show()
```


```
In [30]: df1.to_csv('TVshows.csv', index = False)
```

Movies

Data Cleaning & Analysis of Movies

```
In [31]: df2 = df.copy()
         df2 = df2[df2['type'] == 'Movie']

         df2.head()
```

Out[31]:

	show_id	type	title	country	date_added	release_year	rating	duration	liste
0	s1	Movie	Dick Johnson Is Dead	United States	2021-09-25	2020	PG-13	90 min	Documenta
6	s7	Movie	My Little Pony: A New Generation	NaN	2021-09-24	2021	PG	91 min	Childre Family Mo
7	s8	Movie	Sankofa	United States, Ghana, Burkina Faso, United Kin...	2021-09-24	1993	TV-MA	125 min	Drar Indepenc Mov Internati Mo
9	s10	Movie	The Starling	United States	2021-09-24	2021	PG-13	104 min	Comec Dra
12	s13	Movie	Je Suis Karl	Germany, Czech Republic	2021-09-23	2021	TV-MA	127 min	Drar Internati Mo

In [32]:
```
df2[df2['country'].isnull()]
```

Out[32]:

	show_id	type	title	country	date_added	release_year	rating	duration	lis
6	s7	Movie	My Little Pony: A New Generation	NaN	2021-09-24	2021	PG	91 min	Chil Family I
13	s14	Movie	Confessions of an Invisible Girl	NaN	2021-09-22	2021	TV-PG	91 min	Chil Family M Cor
16	s17	Movie	Europe's Most Dangerous Man: Otto Skorzeny in ...	NaN	2021-09-22	2020	TV-MA	67 min	Documer Intern I
18	s19	Movie	Intrusion	NaN	2021-09-22	2021	TV-14	94 min	T
22	s23	Movie	Avvai Shanmughi	NaN	2021-09-21	1996	TV-PG	161 min	Con Intern I
...	...	...	...	...	...	...	...	...	
8585	s8586	Movie	Three-Quarters Decent	NaN	2019-06-20	2010	TV-14	96 min	Con D Intern I
8602	s8603	Movie	Tom and Jerry: The Magic Ring	NaN	2019-12-15	2001	TV-Y7	60 min	Chil Family M Cor
8622	s8623	Movie	Tremors 2: Aftershocks	NaN	2020-01-01	1995	PG-13	100 min	Con Horror M S F
8718	s8719	Movie	Westside vs. the World	NaN	2019-08-09	2019	TV-MA	96 min	Documer Sports I
8759	s8760	Movie	World's Weirdest Homes	NaN	2019-02-01	2015	TV-PG	49 min	I

440 rows × 10 columns

We forward fill(ffill) the na values of the country assuming that the data was acquired chronologically per country. forward fill takes the last value preceding the null value and fills it.We use axis 0 to take in te first country, for places with multiple countries in the list.

We then run the syntax is null to see if the results have been effective. And it returns an empty dataframe.

In [33]:
```
df2['country'] = df2['country'].ffill(axis=0)
df2[df2['country'].isnull()]
```

Out[33]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description

In [34]:
```
df2[df2['rating'].isnull()]
```

Out[34]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_i
5989	s5990	Movie	13TH: A Conversation with Oprah Winfrey & Ava ...	United States	2017-01-26	2017	NaN	37 min	Movie
7537	s7538	Movie	My Honor Was Loyalty	Italy	2017-03-01	2015	NaN	115 min	Drama

at function is similar to the iloc and loc for identiifying a particular cell and then filling in with the desired output.

In this case, since the null values are only 2, It was wise to google and manually fill in the data using the at. function.

In [35]:
```
df2.at[5989,'rating'] = 'TV-PG'
df2.at[7537,'rating'] = 'TV-MA'
df2[df2['rating'].isnull()]
```

Out[35]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	description

The date_added still has 10 null values.

In [36]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6131 entries, 0 to 8806
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       6131 non-null   object        
 1   type          6131 non-null   object        
 2   title         6131 non-null   object        
 3   country       6131 non-null   object        
 4   date_added    6131 non-null   datetime64[ns]
 5   release_year  6131 non-null   int64         
 6   rating        6131 non-null   object        
 7   duration      6128 non-null   object        
 8   listed_in     6131 non-null   object        
 9   description   6131 non-null   object        
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 655.9+ KB
```

We drop the null values of the specific column of the 3 null values in duration.

In [37]:
```
df2 = df2.dropna(axis=0, subset=['duration'])
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6128 entries, 0 to 8806
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       6128 non-null   object        
 1   type          6128 non-null   object        
 2   title         6128 non-null   object        
 3   country       6128 non-null   object        
 4   date_added    6128 non-null   datetime64[ns]
 5   release_year  6128 non-null   int64         
 6   rating        6128 non-null   object        
 7   duration      6128 non-null   object        
 8   listed_in     6128 non-null   object        
 9   description   6128 non-null   object        
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 526.6+ KB
```

Our data is clean.

The only problem we might face is the country column that has more than one value in it therefore choosing the first country as the country_made.

In [38]:
```
df2['country_made'] = df2.apply(lambda _: '', axis=1)
df2
```

Out[38]:

	show_id	type	title	country	date_added	release_year	rating	duration	li
0	s1	Movie	Dick Johnson Is Dead	United States	2021-09-25	2020	PG-13	90 min	Docume
6	s7	Movie	My Little Pony: A New Generation	United States	2021-09-24	2021	PG	91 min	Chi Family
7	s8	Movie	Sankofa	United States, Ghana, Burkina Faso, United Kin...	2021-09-24	1993	TV-MA	125 min	Indep Interr
9	s10	Movie	The Starling	United States	2021-09-24	2021	PG-13	104 min	Co
12	s13	Movie	Je Suis Karl	Germany, Czech Republic	2021-09-23	2021	TV-MA	127 min	Interr
...	...	...	...	...	...	...	...	...	
8801	s8802	Movie	Zinzana	United Arab Emirates, Jordan	2016-03-09	2015	TV-MA	96 min	Interr
8802	s8803	Movie	Zodiac	United States	2019-11-20	2007	R	158 min	Cult
8804	s8805	Movie	Zombieland	United States	2019-11-01	2009	R	88 min	Co Horror
8805	s8806	Movie	Zoom	United States	2020-01-11	2006	PG	88 min	Chi Family Co
8806	s8807	Movie	Zubaan	India	2019-03-02	2015	TV-14	111 min	Interr Movies & M

6128 rows × 11 columns

In [39]:
```
df2['country_made'] = [x.split(',')[-0] for x in df2['country']]
df2
```

Out[39]:

	show_id	type	title	country	date_added	release_year	rating	duration	li
0	s1	Movie	Dick Johnson Is Dead	United States	2021-09-25	2020	PG-13	90 min	Docume
6	s7	Movie	My Little Pony: A New Generation	United States	2021-09-24	2021	PG	91 min	Chi Family
7	s8	Movie	Sankofa	United States, Ghana, Burkina Faso, United Kin...	2021-09-24	1993	TV-MA	125 min	Indep Interr
9	s10	Movie	The Starling	United States	2021-09-24	2021	PG-13	104 min	Co
12	s13	Movie	Je Suis Karl	Germany, Czech Republic	2021-09-23	2021	TV-MA	127 min	Interr
...	...	...	...	...	...	...	...	...	
8801	s8802	Movie	Zinzana	United Arab Emirates, Jordan	2016-03-09	2015	TV-MA	96 min	Interr
8802	s8803	Movie	Zodiac	United States	2019-11-20	2007	R	158 min	Cult
8804	s8805	Movie	Zombieland	United States	2019-11-01	2009	R	88 min	Co Horror
8805	s8806	Movie	Zoom	United States	2020-01-11	2006	PG	88 min	Chi Family Co
8806	s8807	Movie	Zubaan	India	2019-03-02	2015	TV-14	111 min	Interr Movies & M

6128 rows × 11 columns

Analysis and Visualization

Statistical Analysis

In [40]:
```
df2['duration'].mode()
```

Out[40]:
```
0    90 min
Name: duration, dtype: object
```

In [41]:
```
df2['rating'].mode()
```

Out[41]:
```
0    TV-MA
Name: rating, dtype: object
```

In [42]:
```
df2['listed_in'].mode()
```

Out[42]:
```
0    Dramas, International Movies
Name: listed_in, dtype: object
```

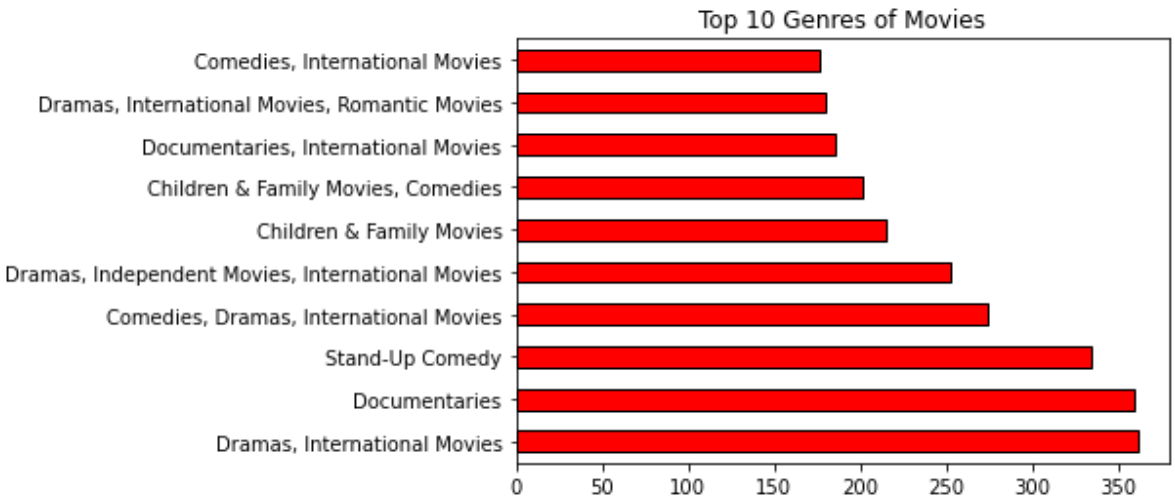
In [43]:
```
df2['country_made'].mode()
```

Out[43]:
```
0    United States
Name: country_made, dtype: object
```

Vizualization

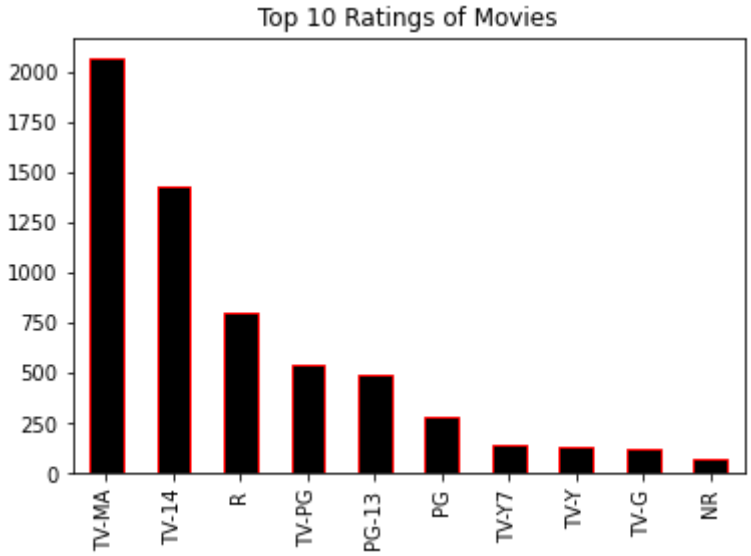
Top 10 Genres of Movies

In [44]:
```
df2["listed_in"].value_counts()[:10].plot(kind="barh", color="red", edgecolor='bla
plt.title("Top 10 Genres of Movies");
```



Top 10 Ratings of Movies

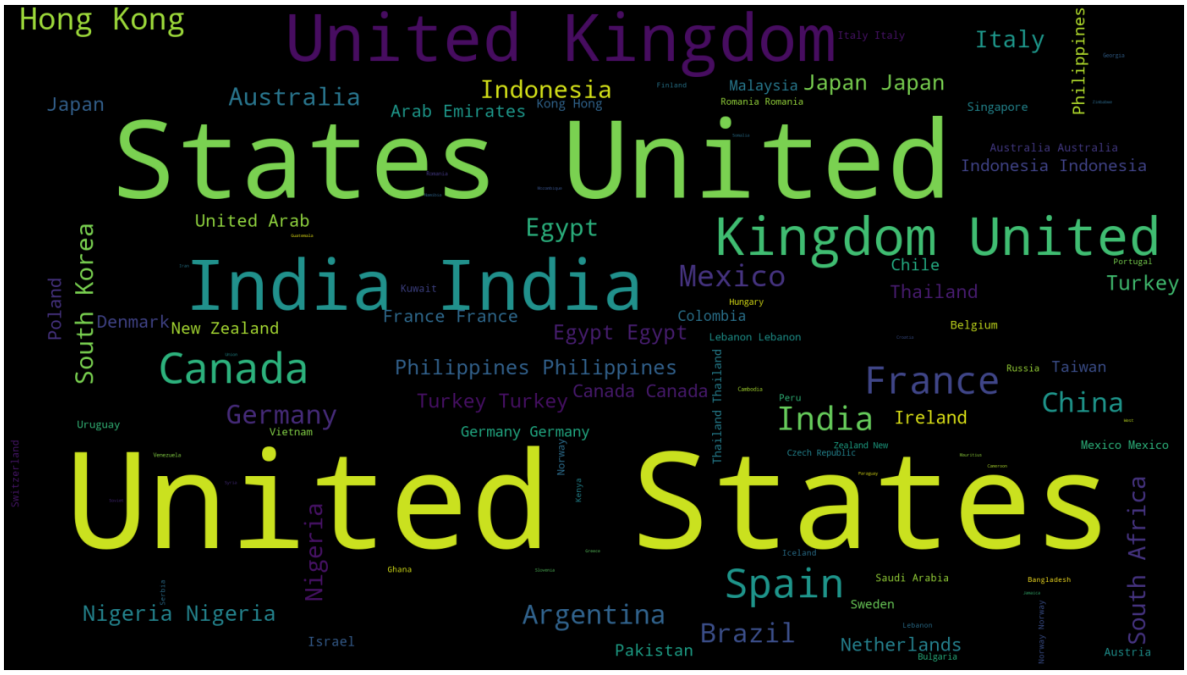
In [45]:
```
df2["rating"].value_counts()[:10].plot(kind="bar", color="black", edgecolor='red')
plt.title("Top 10 Ratings of Movies");
```



Word Cloud Movie

Word Cloud for Country

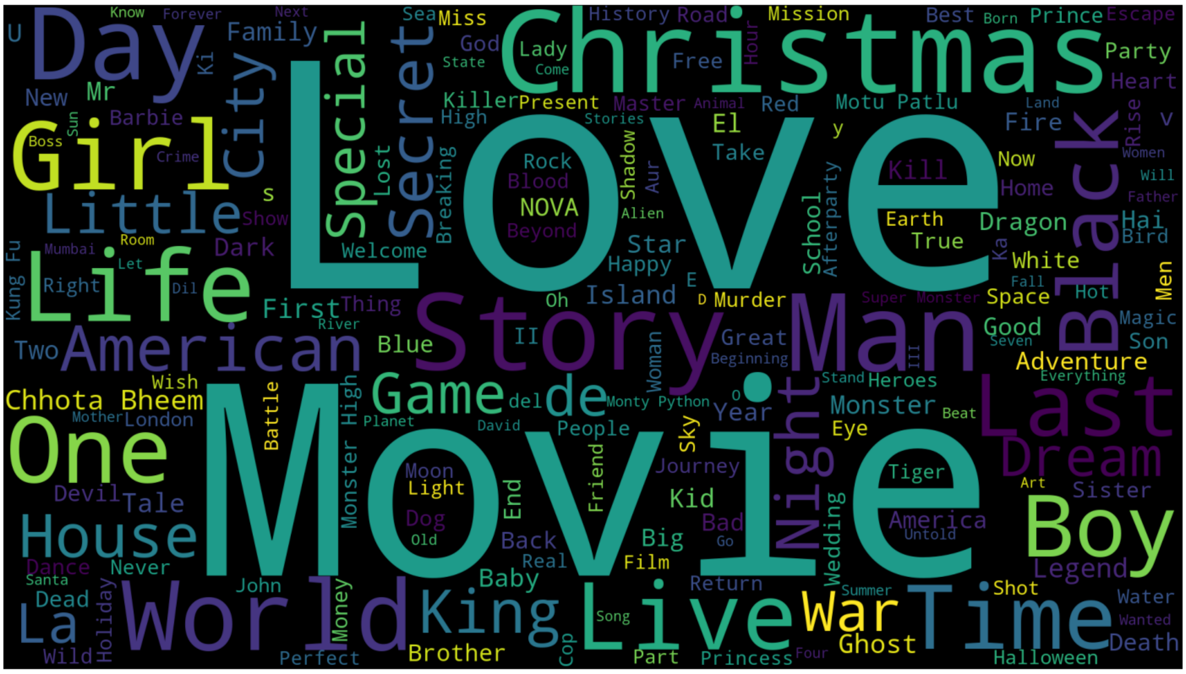
In [46]:
```
plt.subplots(figsize=(25,15))
wordcloud = WordCloud(
                          background_color='black',
                          width=1920,
                          height=1080
                         ).generate(" ".join(df2.country_made))
plt.imshow(wordcloud)
plt.axis('off')
plt.savefig('country_made.png')
plt.show()
```



Word Cloud for Title

In [47]:
```
plt.subplots(figsize=(25,15))
wordcloud = WordCloud(
```

```
                                background_color='black',
                                width=1920,
                                height=1080
                               ).generate(" ".join(df2.title))
plt.imshow(wordcloud)
plt.axis('off')
plt.savefig('title.png')
plt.show()
```



In [48]:
```
df2.to_csv('Movies.csv', index = False)
```