ujjwal-singh-project-1

June 12, 2024

[1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[2]:
```
df=pd.read_csv('ottdataset.csv')
df.head()
```

[2]:

	Unnamed: 0	show_id	OTT Platform	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	1	S0002	Amazon	Movie	Take Care Good Night	Girish Joshi	Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar	India	March 30, 2021	2018	13+	110 min	Drama, International	A Metro Family decides to fight a Cyber Crimin…
1	9670	S9671	Disney	Movie	Ice Age: A Mammoth Christmas	Karen Disher	Raymond Albert Romano, John Leguizamo, Denis L…	United States	November 26, 2021	2011	TV-G	23 min	Animation, Comedy, Family	
2	9673	S9674	Disney	Movie	Becoming Cousteau	Liz Garbus	Jacques Yves Cousteau, Vincent Cassel	United States	November 24, 2021	2021	PG-13	94 min	Biographical, Documentary	
3	9677	S9678	Disney	Movie	A Muppets Christmas: Letters To Santa	Kirk R. Thatcher	Steve Whitmire, Dave Goelz, Bill Barretta, Eri…	United States	November 19, 2021	2008	G	45 min	Comedy, Family, Musical	
4	9680	S9681	Disney	Movie	The Pixar Story	Leslie Iwerks	Stacy Keach, John Lasseter, Brad Bird, John Mu…	United States	November 19, 2021	2007	G	91 min	Documentary, Family	

```
1          Sid the Sloth is on Santa's naughty list.
2  An inside look at the legendary life of advent…
3  Celebrate the holiday season with all your fav…
4  A groundbreaking company forever changes the f…
```

[3]:
```
df.columns
```

[3]:
```
Index(['Unnamed: 0', 'show_id', 'OTT Platform', 'type', 'title', 'director',
       'cast', 'country', 'date_added', 'release_year', 'rating', 'duration',
       'listed_in', 'description'],
      dtype='object')
```

[4]:
```
df.drop(['Unnamed: 0'],axis=1,inplace=True)
```

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

[5]:

	show_id	OTT Platform	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	S0002	Amazon	Movie	Take Care Good Night	Girish Joshi	Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar	India	March 30, 2021	2018	13+	110 min	Drama, International	A Metro Family decides to fight a Cyber Crimin…
1	S9671	Disney	Movie	Ice Age: A Mammoth Christmas	Karen Disher	Raymond Albert Romano, John Leguizamo, Denis L…	United States	November 26, 2021	2011	TV-G	23 min	Animation, Comedy, Family	Sid the Sloth is on Santa's naughty list.
2	S9674	Disney	Movie	Becoming Cousteau	Liz Garbus	Jacques Yves Cousteau, Vincent Cassel	United States	November 24, 2021	2021	PG-13	94 min	Biographical, Documentary	An inside look at the legendary life of advent…
3	S9678	Disney	Movie	A Muppets Christmas: Letters To Santa	Kirk R. Thatcher	Steve Whitmire, Dave Goelz, Bill Barretta, Eri…	United States	November 19, 2021	2008	G	45 min	Comedy, Family, Musical	Celebrate the holiday season with all your fav…
4	S9681	Disney	Movie	The Pixar Story	Leslie Iwerks	Stacy Keach, John Lasseter, Brad Bird, John Mu…	United States	November 19, 2021	2007	G	91 min	Documentary, Family	A groundbreaking company forever changes the f…

```
2  An inside look at the legendary life of advent…
3  Celebrate the holiday season with all your fav…
4  A groundbreaking company forever changes the f…
```

[6]:
```
df['OTT Platform'].value_counts()
```

[6]:
```
OTT Platform
Netflix    5332
Disney      818
Amazon        1
Name: count, dtype: int64
```

[7]:
```
df['type'].value_counts()
```

[7]:
```
type
Movie      6004
TV Show     147
Name: count, dtype: int64
```

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6151 entries, 0 to 6150
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       6151 non-null   object
 1   OTT Platform  6151 non-null   object
 2   type          6151 non-null   object
 3   title         6151 non-null   object
 4   director      6151 non-null   object
 5   cast          6151 non-null   object
 6   country       6151 non-null   object
 7   date_added    6151 non-null   object
 8   release_year  6151 non-null   int64
 9   rating        6151 non-null   object
 10  duration      6151 non-null   object
 11  listed_in     6151 non-null   object
 12  description   6151 non-null   object
dtypes: int64(1), object(12)
memory usage: 624.8+ KB
```

[9]:
```
def duration_to_min(duration):
    if 'Season' in duration:
        return int(duration.split()[0]) * 22* 30
    else:
        return int(duration.split()[0])
```

```
df['duration_min'] = df['duration'].map(duration_to_min)
df.drop(['duration'],axis=1,inplace=True)
df.head()
```

[9]:

```
  show_id OTT Platform   type                                    title  \
0   S0002       Amazon  Movie                   Take Care Good Night
1   S9671       Disney  Movie           Ice Age: A Mammoth Christmas
2   S9674       Disney  Movie                        Becoming Cousteau
3   S9678       Disney  Movie  A Muppets Christmas: Letters To Santa
4   S9681       Disney  Movie                          The Pixar Story

           director                                                cast  \
0      Girish Joshi   Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar
1      Karen Disher  Raymond Albert Romano, John Leguizamo, Denis L…
2        Liz Garbus              Jacques Yves Cousteau, Vincent Cassel
3  Kirk R. Thatcher  Steve Whitmire, Dave Goelz, Bill Barretta, Eri…
4     Leslie Iwerks  Stacy Keach, John Lasseter, Brad Bird, John Mu…

         country         date_added  release_year rating  \
0          India     March 30, 2021          2018    13+
1  United States  November 26, 2021          2011   TV-G
2  United States  November 24, 2021          2021  PG-13
3  United States  November 19, 2021          2008      G
4  United States  November 19, 2021          2007      G

                   listed_in  \
0       Drama, International
1  Animation, Comedy, Family
2  Biographical, Documentary
3    Comedy, Family, Musical
4        Documentary, Family

                                        description  duration_min
0  A Metro Family decides to fight a Cyber Crimin…           110
1          Sid the Sloth is on Santa's naughty list.            23
2  An inside look at the legendary life of advent…            94
3  Celebrate the holiday season with all your fav…            45
4  A groundbreaking company forever changes the f…            91
```

[10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6151 entries, 0 to 6150
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
```

```
 0   show_id       6151 non-null   object
 1   OTT Platform  6151 non-null   object
 2   type          6151 non-null   object
 3   title         6151 non-null   object
 4   director      6151 non-null   object
 5   cast          6151 non-null   object
 6   country       6151 non-null   object
 7   date_added    6151 non-null   object
 8   release_year  6151 non-null   int64
 9   rating        6151 non-null   object
 10  listed_in     6151 non-null   object
 11  description   6151 non-null   object
 12  duration_min  6151 non-null   int64
dtypes: int64(2), object(11)
memory usage: 624.8+ KB
```

[11]:
```
# Data type change for all columns
datatype_map = {
    'show_id' : 'str',
    'OTT Platform' : 'category',
    'type' : 'category',
    'title' : 'str',
    'director' : 'category',
    'cast' : 'category',
    'country' : 'category',
    'rating' : 'category',
    'duration_min' : 'int',
    'date_added' : 'datetime64',
    'listed_in' : 'category',
    'description' : 'str'
}
# apply mapped category
df = df.astype(datatype_map,errors='ignore')
```

[12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6151 entries, 0 to 6150
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       6151 non-null   object
 1   OTT Platform  6151 non-null   category
 2   type          6151 non-null   category
 3   title         6151 non-null   object
 4   director      6151 non-null   category
 5   cast          6151 non-null   category
 6   country       6151 non-null   category
```

```
 7   date_added    6151 non-null   object
 8   release_year  6151 non-null   int64
 9   rating        6151 non-null   category
 10  listed_in     6151 non-null   category
 11  description   6151 non-null   object
 12  duration_min  6151 non-null   int32
dtypes: category(7), int32(1), int64(1), object(4)
memory usage: 711.8+ KB
```

[13]:
```
df.head()
```

[13]:

	show_id	OTT Platform	type	title	director	cast	country	date_added	release_year	rating	listed_in	description	duration_min
0	S0002	Amazon	Movie	Take Care Good Night	Girish Joshi	Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar	India	March 30, 2021	2018	13+	Drama, International	A Metro Family decides to fight a Cyber Crimin…	110
1	S9671	Disney	Movie	Ice Age: A Mammoth Christmas	Karen Disher	Raymond Albert Romano, John Leguizamo, Denis L…	United States	November 26, 2021	2011	TV-G	Animation, Comedy, Family	Sid the Sloth is on Santa's naughty list.	23
2	S9674	Disney	Movie	Becoming Cousteau	Liz Garbus	Jacques Yves Cousteau, Vincent Cassel	United States	November 24, 2021	2021	PG-13	Biographical, Documentary	An inside look at the legendary life of advent…	94
3	S9678	Disney	Movie	A Muppets Christmas: Letters To Santa	Kirk R. Thatcher	Steve Whitmire, Dave Goelz, Bill Barretta, Eri…	United States	November 19, 2021	2008	G	Comedy, Family, Musical	Celebrate the holiday season with all your fav…	45
4	S9681	Disney	Movie	The Pixar Story	Leslie Iwerks	Stacy Keach, John Lasseter, Brad Bird, John Mu…	United States	November 19, 2021	2007	G	Documentary, Family	A groundbreaking company forever changes the f…	91

[14]:
```
ratings=df['rating'].unique()
ratings
```

[14]:
```
['13+', 'TV-G', 'PG-13', 'G', 'PG', …, 'TV-MA', 'R', 'NC-17', 'NR', 'UR']
Length: 15
Categories (15, object): ['13+', 'G', 'NC-17', 'NR', …, 'TV-Y', 'TV-Y7',
'TV-Y7-FV', 'UR']
```

[15]:
```
ratings_order = ['TV-Y', 'TV-Y7', 'TV-Y7-FV', 'G', 'PG', 'TV-G', 'TV-PG',␣
↪'PG-13', 'TV-14', 'R', 'NC-17', 'NR', 'UR', 'TV-MA','13+']
```

[16]:
```
# Reorder the 'rating' column
# df['rating'] = pd.Categorical(df['rating'], categories=ratings_order,␣
↪ordered=True)
df['rating'].value_counts()
```

[16]:
```
rating
TV-MA       1822
TV-14       1226
R            778
PG-13        536
TV-PG        535
PG           498
G            271
TV-G         241
TV-Y7         97
TV-Y          77
NR            58
TV-Y7-FV       6
UR             3
NC-17          2
13+            1
Name: count, dtype: int64
```

[44]:
```
# Generating new features : `delay_in_release` In years (int) we can take the␣
↪difference between `date_added` and `released_date`
df['delay_in_release'] = df['date_added'].dt.year - df['release_year']
df.head()
```

[44]:
```
  show_id OTT Platform   type                                    title  \
0   S0002       Amazon  Movie                   Take Care Good Night
1   S9671       Disney  Movie           Ice Age: A Mammoth Christmas
2   S9674       Disney  Movie                      Becoming Cousteau
3   S9678       Disney  Movie  A Muppets Christmas: Letters To Santa
4   S9681       Disney  Movie                        The Pixar Story

           director                                                cast  \
```

```
0      Girish Joshi   Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar
1      Karen Disher  Raymond Albert Romano, John Leguizamo, Denis L…
2        Liz Garbus              Jacques Yves Cousteau, Vincent Cassel
3  Kirk R. Thatcher  Steve Whitmire, Dave Goelz, Bill Barretta, Eri…
4     Leslie Iwerks  Stacy Keach, John Lasseter, Brad Bird, John Mu…

         country date_added  release_year rating                 listed_in  \
0          India 2021-03-30          2018    13+       Drama, International
1  United States 2021-11-26          2011   TV-G  Animation, Comedy, Family
2  United States 2021-11-24          2021  PG-13  Biographical, Documentary
3  United States 2021-11-19          2008      G    Comedy, Family, Musical
4  United States 2021-11-19          2007      G        Documentary, Family

                                      description  duration_min  \
0  A Metro Family decides to fight a Cyber Crimin…           110
1          Sid the Sloth is on Santa's naughty list.            23
2  An inside look at the legendary life of advent…            94
3  Celebrate the holiday season with all your fav…            45
4  A groundbreaking company forever changes the f…            91

   delay_in_release
0               3.0
1              10.0
2               0.0
3              13.0
4              14.0
```

[39]:
```
director_counts = df['director'].str.split(', ').explode().value_counts().head()
# Print the most prolific directors
print("Most Prolific Directors:")
print(director_counts)
```

```
Most Prolific Directors:
director
Jan Suter       21
Raúl Campos     19
Jay Karas       16
Paul Hoen       16
Marcus Raboy    15
Name: count, dtype: int64
```

[40]:
```
director_counts.index
```

[40]:
```
Index(['Jan Suter', 'Raúl Campos', 'Jay Karas', 'Paul Hoen', 'Marcus Raboy'],
dtype='object', name='director')
```

[41]:
```
director_counts.values
```

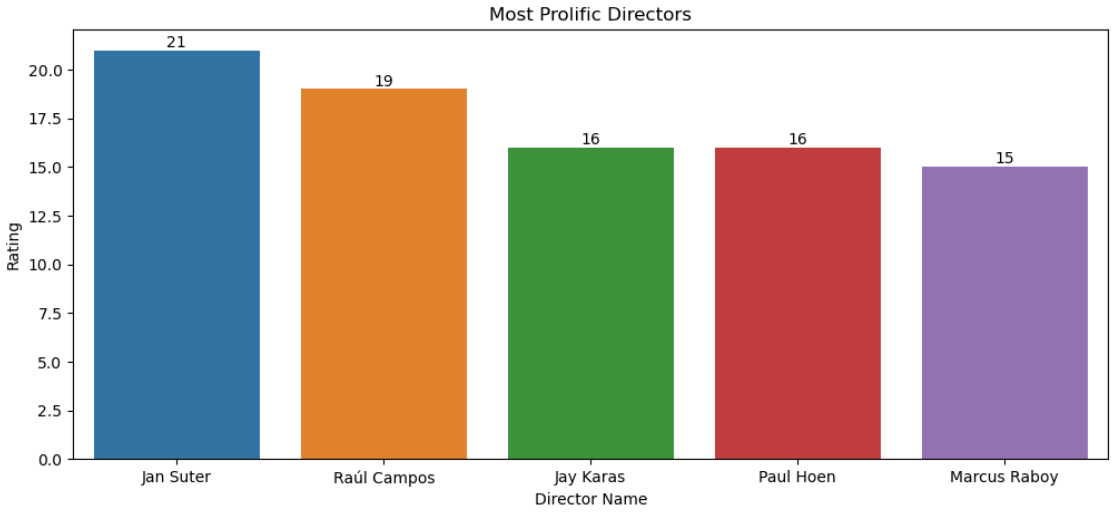
[41]:
```
array([21, 19, 16, 16, 15], dtype=int64)
```

[42]:
```
# Converting series to DataFrame
data={
    'Director Name': director_counts.index,
    'Rating' : director_counts.values
}
director_df=pd.DataFrame(data)
director_df
```

[42]:

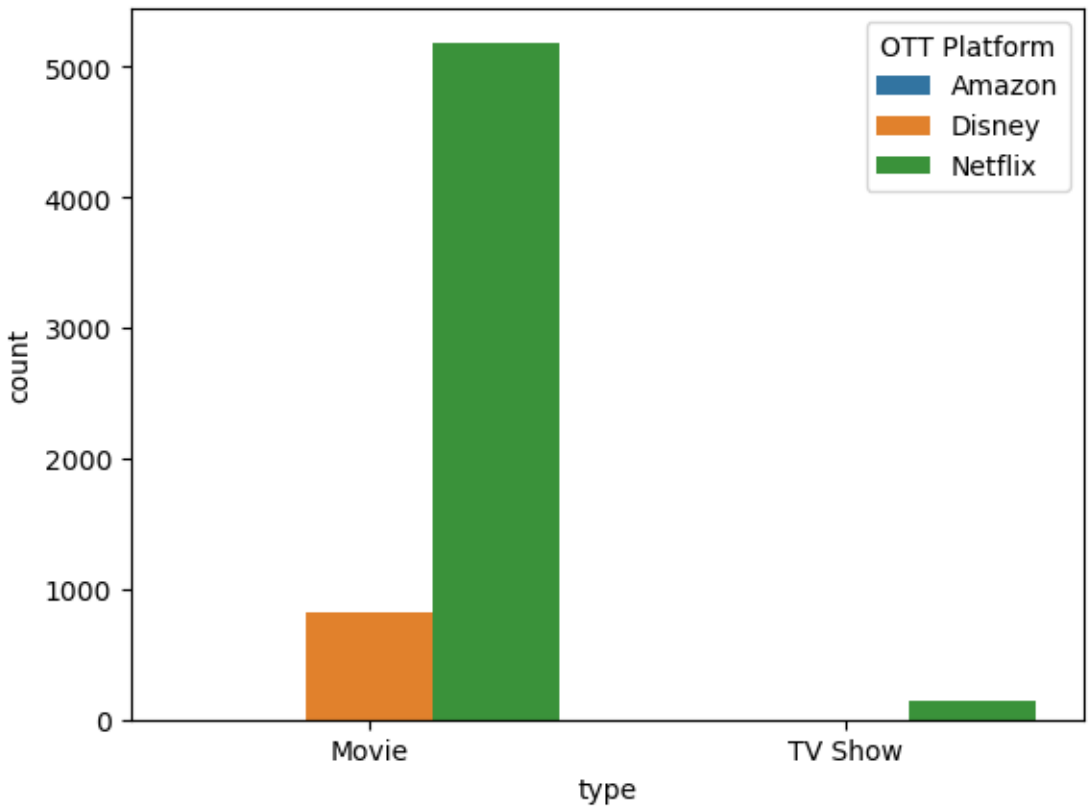
	Director Name	Rating
0	Jan Suter	21
1	Raúl Campos	19
2	Jay Karas	16
3	Paul Hoen	16
4	Marcus Raboy	15

[43]:
```
plt.figure(figsize=(12,5))
plt.title('Most Prolific Directors')
chart=sns.barplot(x='Director Name',y='Rating',data=director_df)
for v in chart.containers: chart.bar_label(v)
```



[38]:
```
sns.countplot(x='type',data=df,hue='OTT Platform')
```

[38]:
```
<Axes: xlabel='type', ylabel='count'>
```

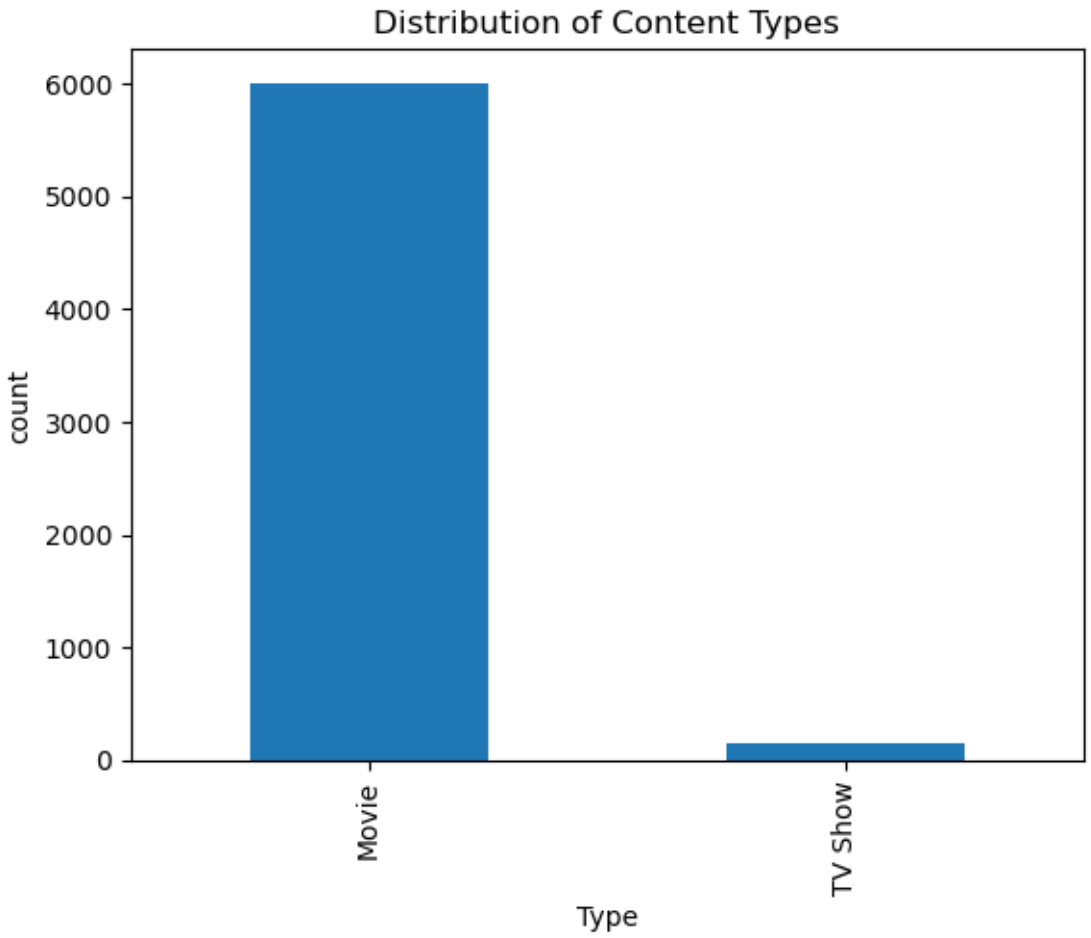


1 Distribution of content types(Movies vs TV Shows)

In this we specify that how many entries are of movies and TV Shows

[20]:
```
distribution= df['type'].value_counts()
distribution.plot(kind='bar',title='Distribution of Content Types',␣
 ↪xlabel='Type' , ylabel='count')
```

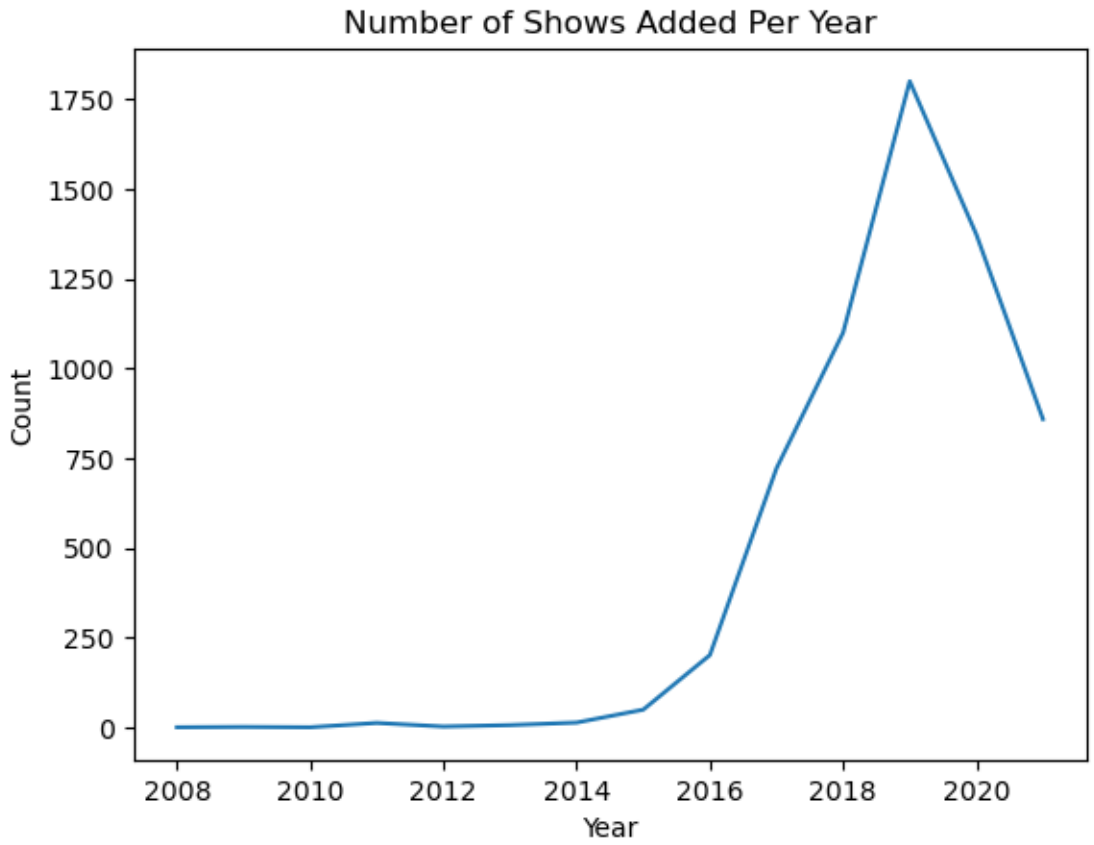
[20]:
```
<Axes: title={'center': 'Distribution of Content Types'}, xlabel='Type',
ylabel='count'>
```



2 Number of Shows Added Per year

[25]:
```
df['date_added'] = pd.to_datetime(df['date_added'], format='%B %d, %Y',␣
 ↪errors='coerce')
shows_per_year=df['date_added'].dt.year.value_counts().sort_index()
shows_per_year.plot(kind='line', title='Number of Shows Added Per Year',␣
 ↪xlabel= 'Year' , ylabel= 'Count')
```

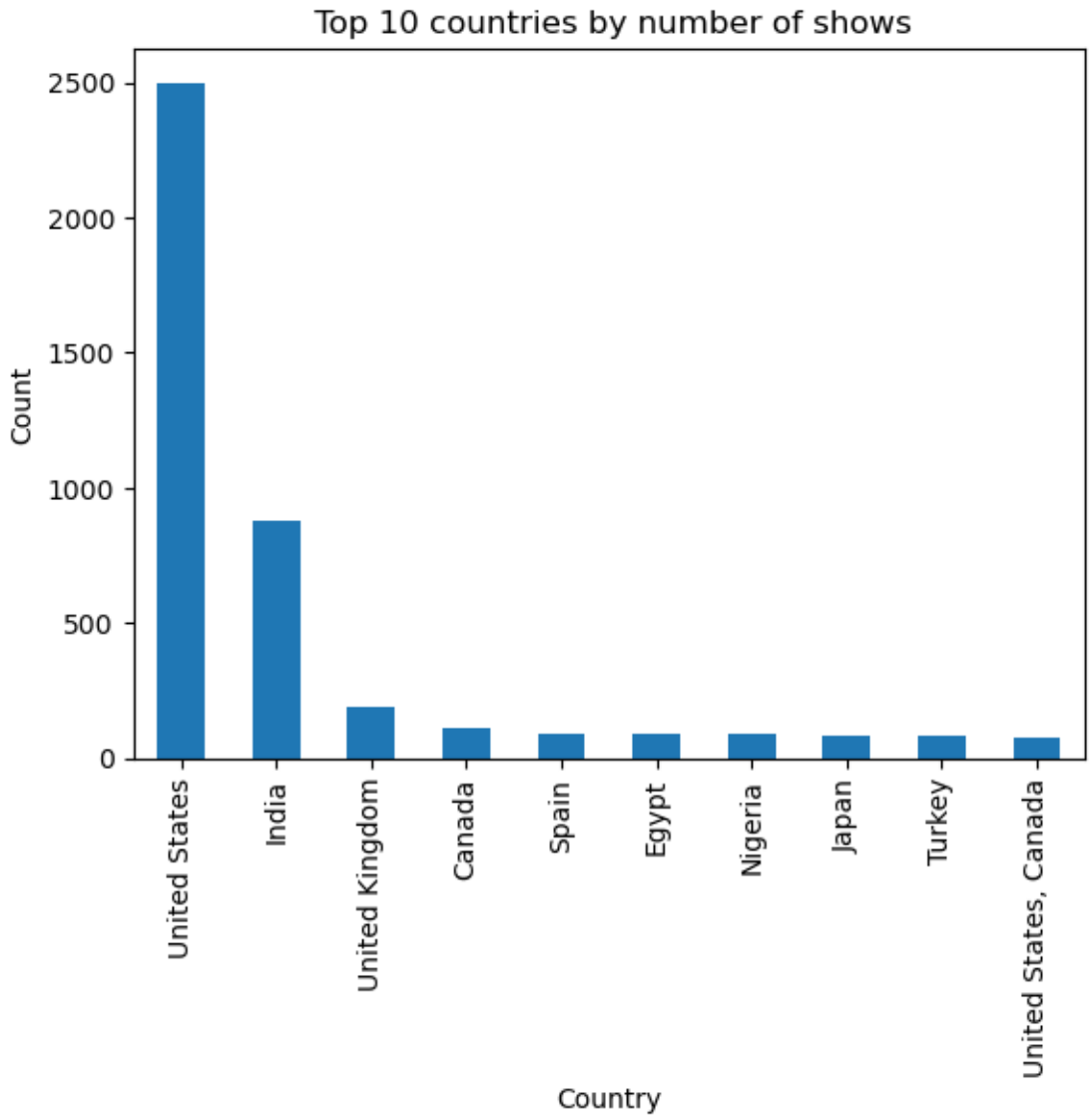
[25]:
```
<Axes: title={'center': 'Number of Shows Added Per Year'}, xlabel='Year',
ylabel='Count'>
```



3 Top 10 countries by number of shows

[29]:
```
top_countries=df['country'].value_counts().head(10)
#value_counts().head(10): Counts the unique values in the 'country' column and␣
 ↪selects the top 10.
top_countries.plot(kind='bar', title=' Top 10 countries by number of shows',␣
 ↪xlabel= 'Country',ylabel='Count')
```

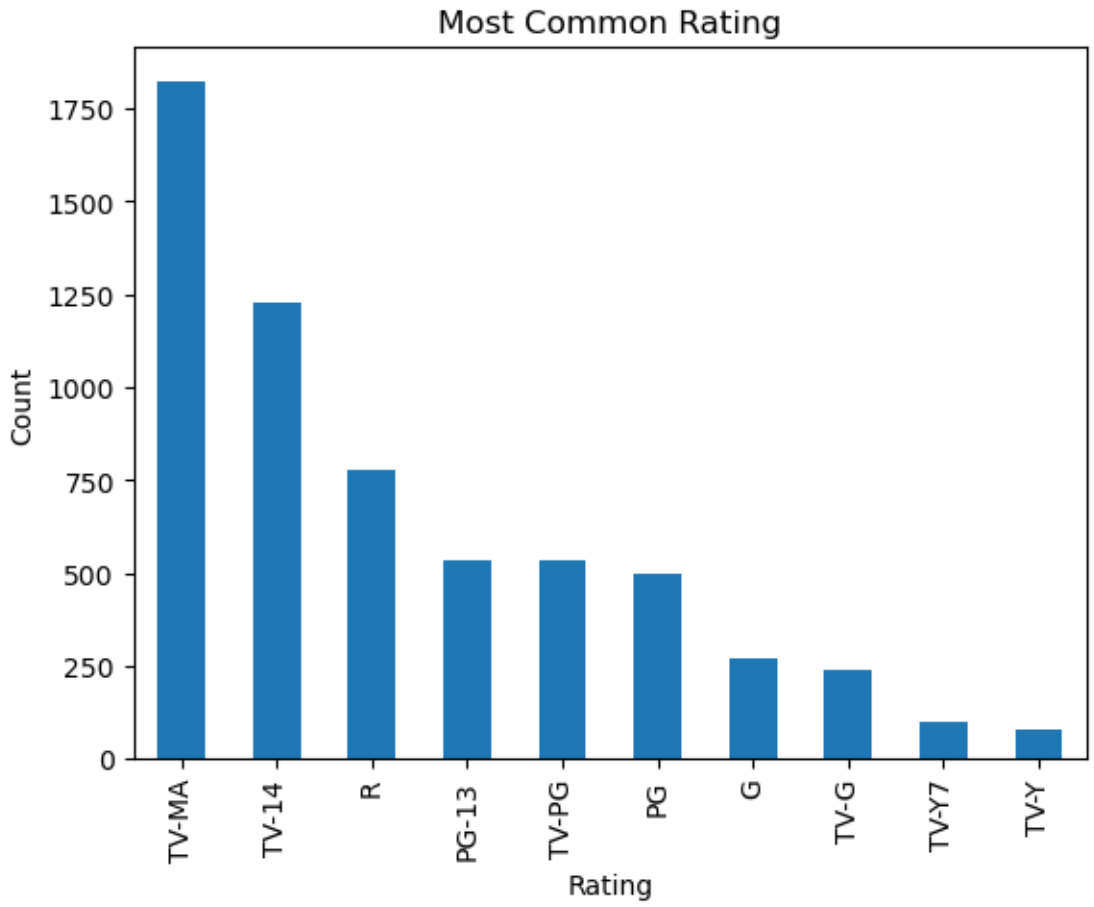
[29]:
```
<Axes: title={'center': ' Top 10 countries by number of shows'},
xlabel='Country', ylabel='Count'>
```



4 Most common content ratings

[30]:
```
count_ratings= df['rating'].value_counts().head(10)
count_ratings.plot(kind='bar', title='Most Common␣
↪Rating',xlabel='Rating',ylabel='Count')
```

[30]: <Axes: title={'center': 'Most Common Rating'}, xlabel='Rating', ylabel='Count'>



5 Average duration of movies and TV shows

6 Extract numerical duration for movies

df[‘duration_minutes’] = df[‘duration’].apply(lambda x: int(x.split()[0]) if ‘min’ in x else 0) average_duration = df.groupby(‘type’)[‘duration_minutes’].mean() average_duration.plot(kind=‘bar’, title=‘Average Duration of Movies and TV Shows’, xlabel=‘Type’, ylabel=‘Average Duration (minutes)’)

print(“Columns in dataframe:”, df.columns)

7 Display the first few rows to inspect the data

print(“First few rows of the dataframe:”) print(df.head())

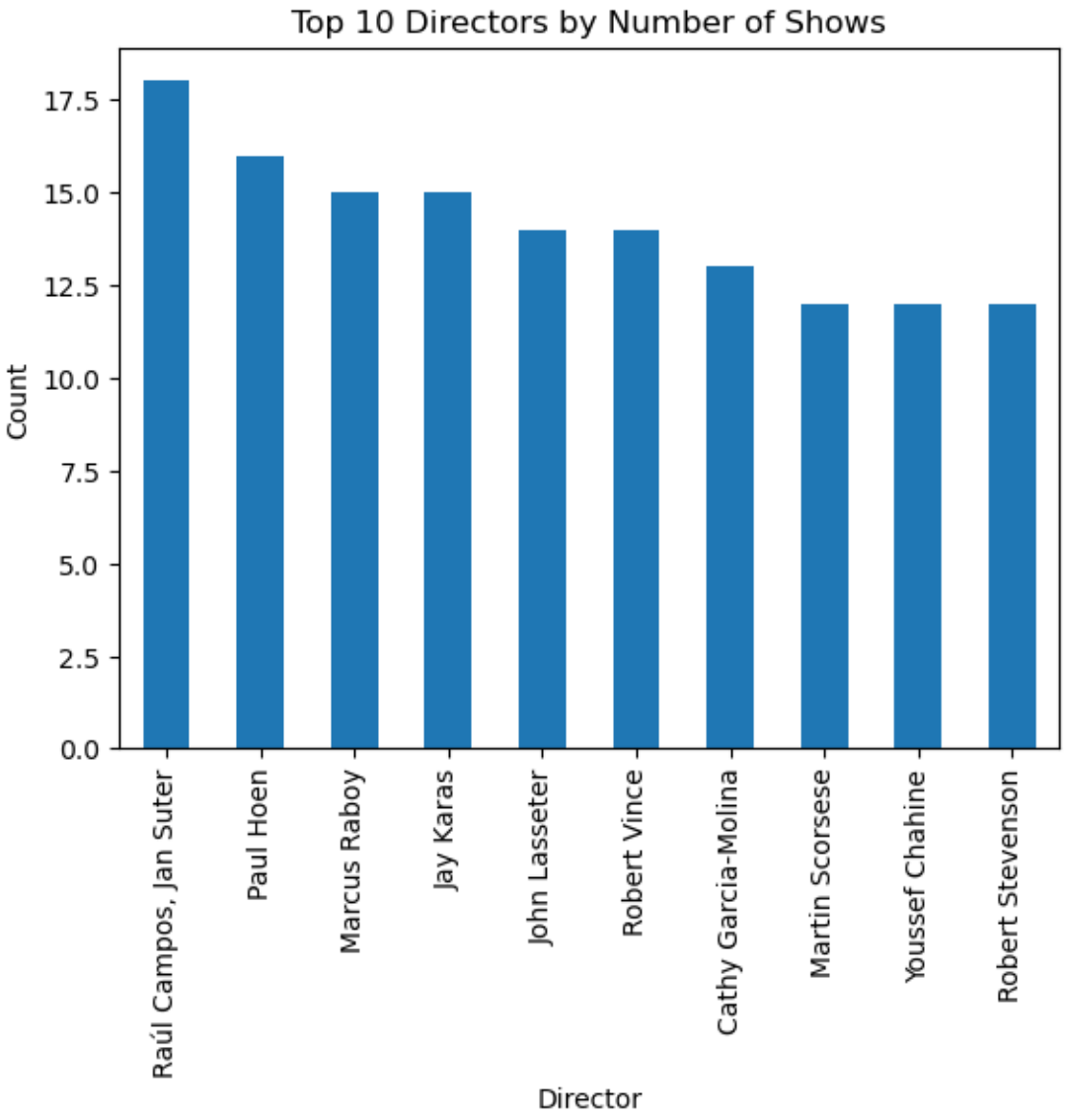
8 Check if ‘duration’ column exists

if ‘duration’ in df.columns: # Extract numerical duration for movies df[‘duration_minutes’] = df[‘duration’].apply(lambda x: int(x.split()[0]) if ‘min’ in x else 0) average_duration = df.groupby(‘type’)[‘duration_minutes’].mean() average_duration.plot(kind=‘bar’, title=‘Average Duration of Movies and TV Shows’, xlabel=‘Type’, ylabel=‘Average Duration (minutes)’)

9 Top 10 directors by number of shows

[47]:
```
top_directors = df['director'].value_counts().head(10)
top_directors.plot(kind='bar', title='Top 10 Directors by Number of Shows',␣
↪xlabel='Director', ylabel='Count')
```

[47]:
```
<Axes: title={'center': 'Top 10 Directors by Number of Shows'},
xlabel='Director', ylabel='Count'>
```



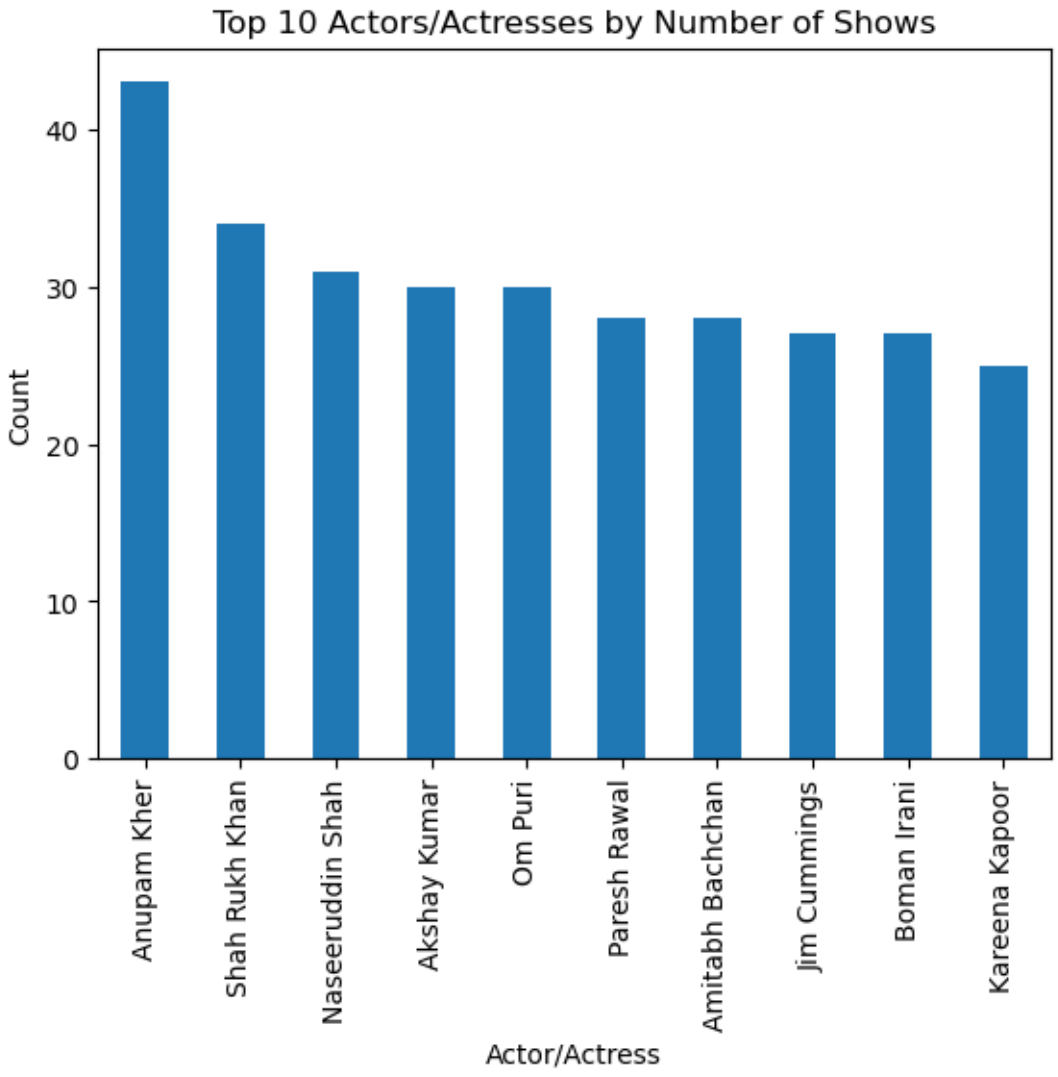
10 Top 10 actors/actresses by number of shows

For calculating the actors/actresses we have to first split the 'cast' column into a list of actors. then we have to count the occurences of each actor by 'counter(flag_cast)

[50]:
```
from collections import Counter

cast_members = df['cast'].str.split(', ')
flat_cast = [item for sublist in cast_members.dropna() for item in sublist]
top_cast = pd.Series(Counter(flat_cast)).sort_values(ascending=False).head(10)
```

```
top_cast.plot(kind='bar', title='Top 10 Actors/Actresses by Number of Shows',␣
 ↪xlabel='Actor/Actress', ylabel='Count')
```

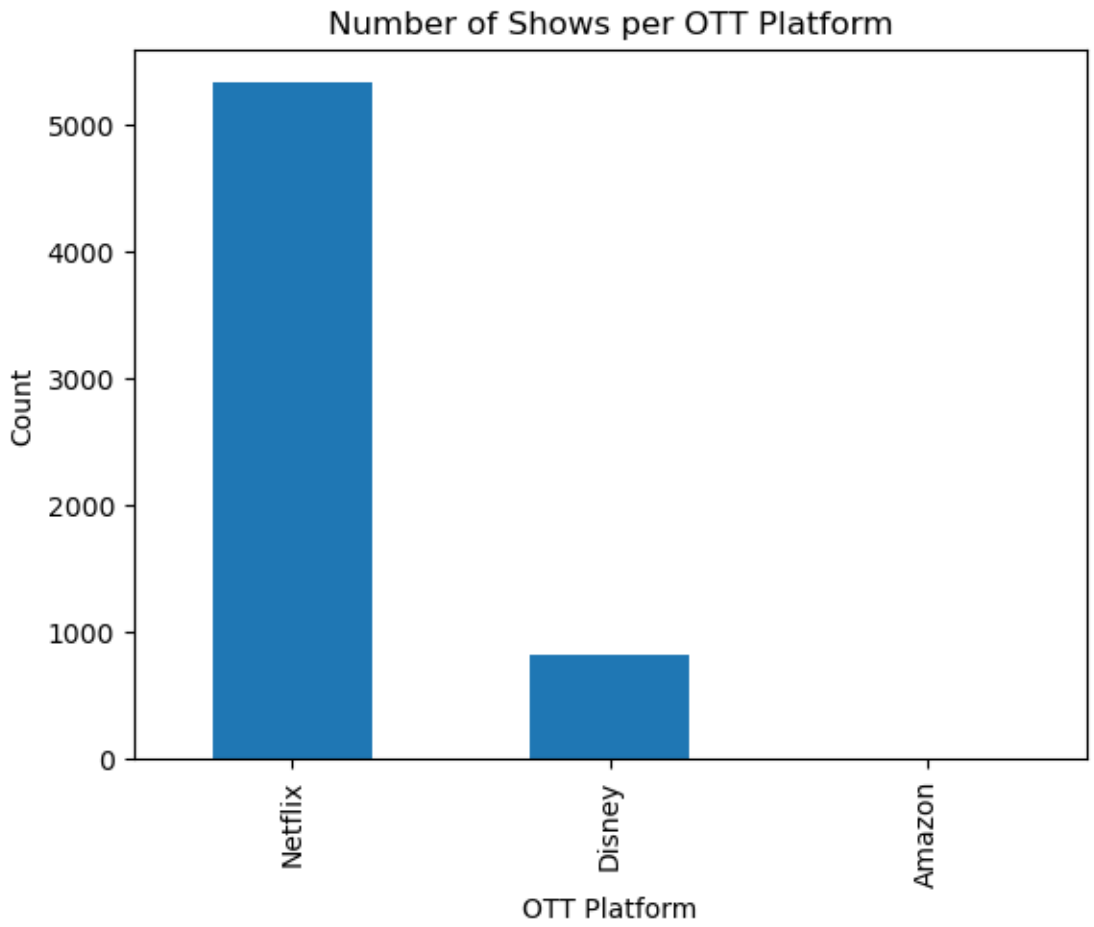
[50]:
```
<Axes: title={'center': 'Top 10 Actors/Actresses by Number of Shows'},
xlabel='Actor/Actress', ylabel='Count'>
```



11 Number of shows per OTT platform

[51]:
```
shows_per_platform = df['OTT Platform'].value_counts()
shows_per_platform.plot(kind='bar', title='Number of Shows per OTT Platform',␣
 ↪xlabel='OTT Platform', ylabel='Count')
```

[51]: <Axes: title={'center': 'Number of Shows per OTT Platform'}, xlabel='OTT Platform', ylabel='Count'>

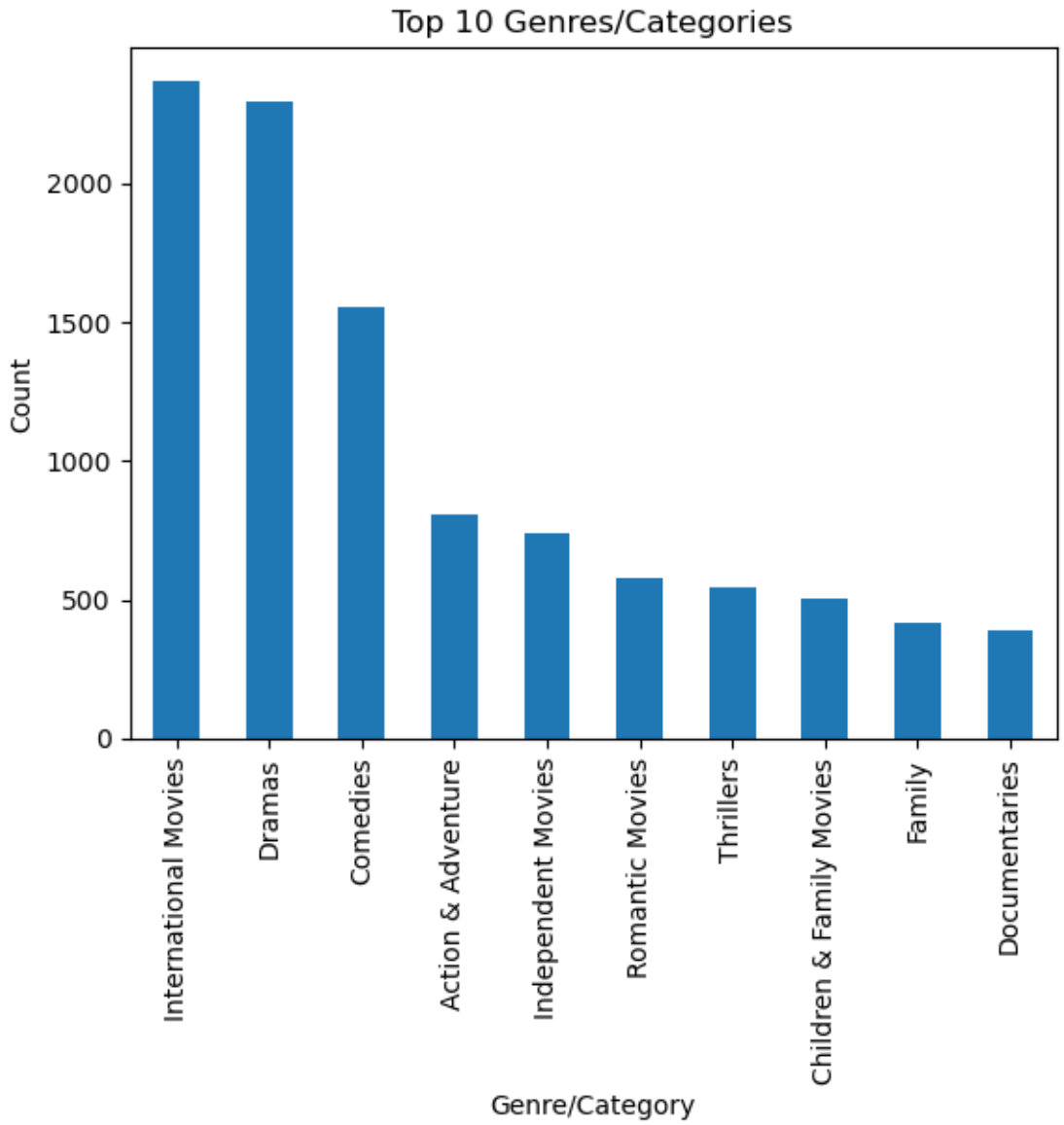


12 Distribution of genres/categories

[53]:
```
df['listed_in'] = df['listed_in'].astype(str).fillna('')

# Split the 'listed_in' column and explode it into separate rows
#Split and Explode: Split the 'listed_in' column on ', ' and use the explode␣
 ↪method to transform each list element into a separate row.
df_exploded = df.assign(listed_in=df['listed_in'].str.split(', ')).
 ↪explode('listed_in')

# Get the top 10 genres/categories
category_distribution = df_exploded['listed_in'].value_counts().head(10)
```

```
# Plot the distribution
category_distribution.plot(kind='bar', title='Top 10 Genres/Categories',␣
↪xlabel='Genre/Category', ylabel='Count')
```

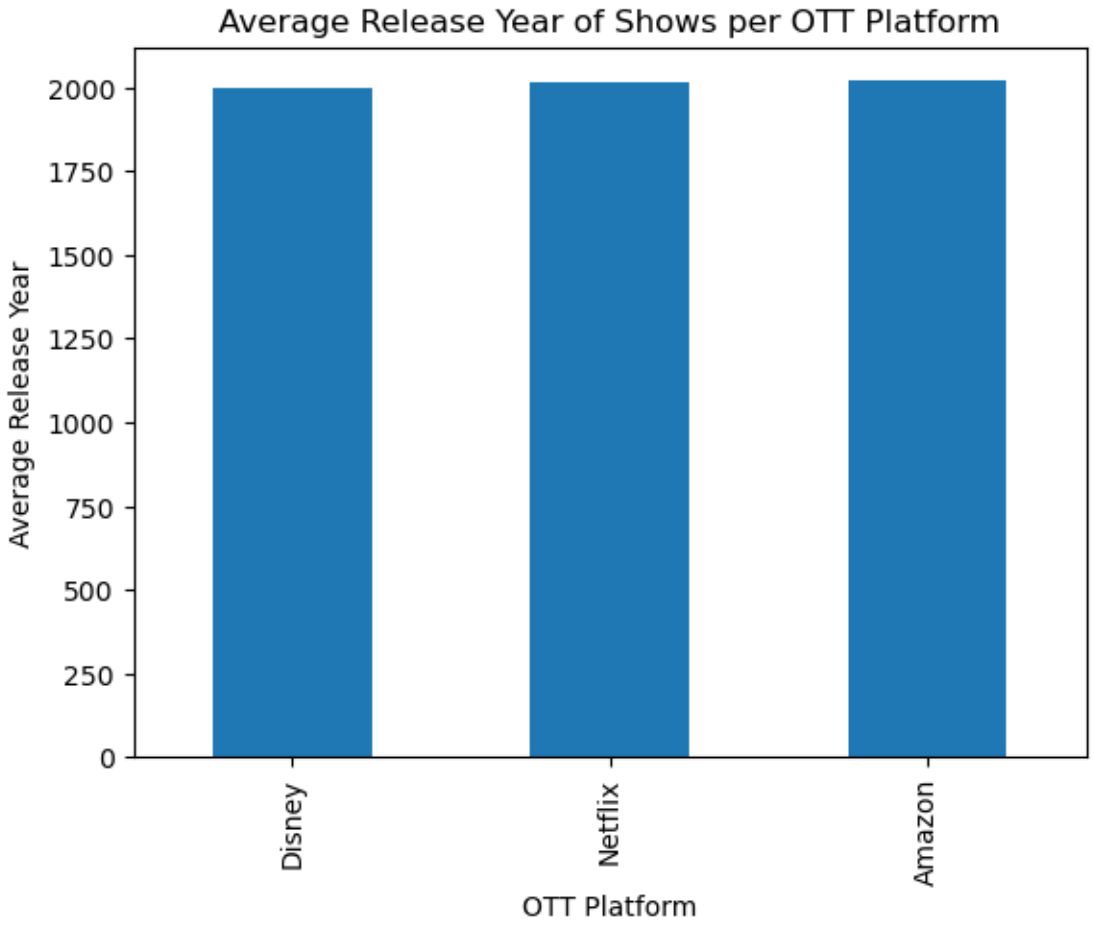
[53]: <Axes: title={'center': 'Top 10 Genres/Categories'}, xlabel='Genre/Category', ylabel='Count'>



13 Average release year of shows per OTT platform

[54]:
```
average_release_year = df.groupby('OTT Platform')['release_year'].mean().
↪sort_values()
average_release_year.plot(kind='bar', title='Average Release Year of Shows per␣
↪OTT Platform', xlabel='OTT Platform', ylabel='Average Release Year')
```

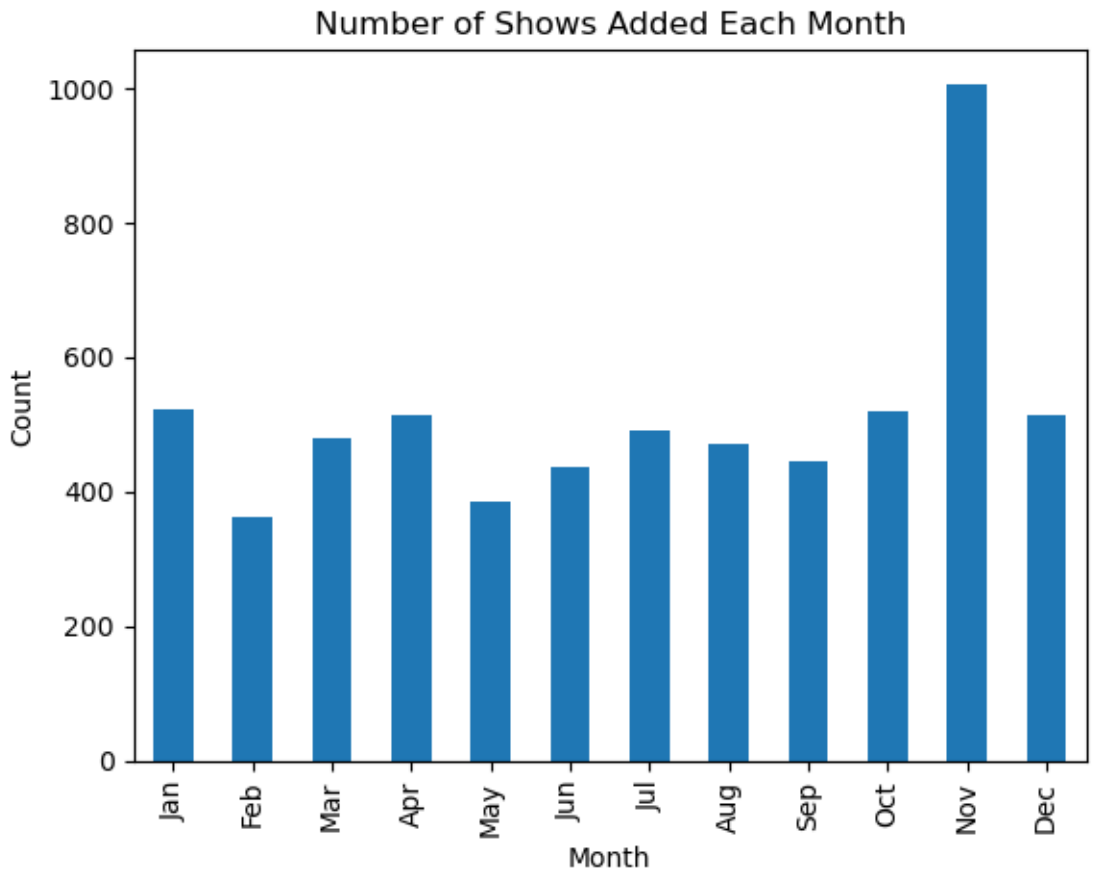
[54]:
```
<Axes: title={'center': 'Average Release Year of Shows per OTT Platform'},
xlabel='OTT Platform', ylabel='Average Release Year'>
```



14 Number of shows added each month

[56]:
```
shows_per_month = df['date_added'].dt.month.value_counts().sort_index()
shows_per_month.index = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul',␣
↪'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
shows_per_month.plot(kind='bar', title='Number of Shows Added Each Month',␣
↪xlabel='Month', ylabel='Count')
```

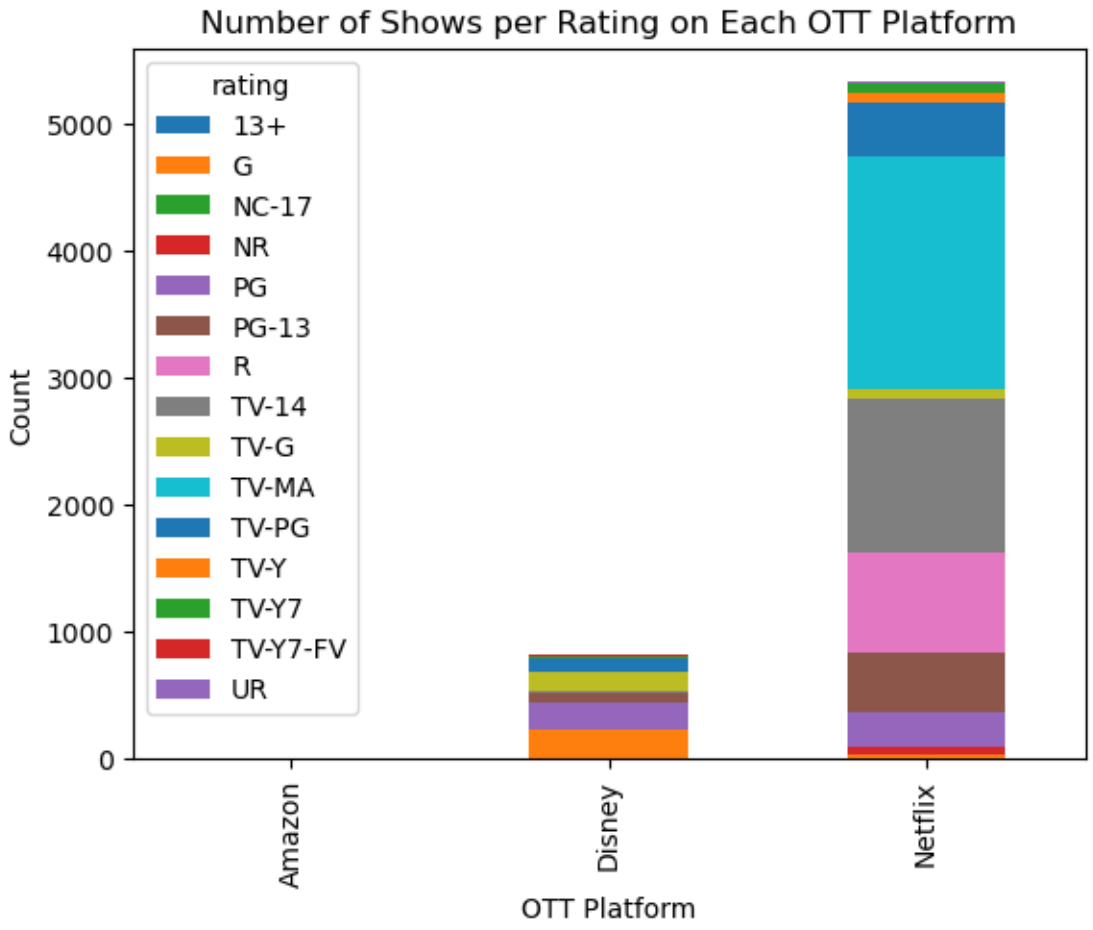
[56]: <Axes: title={'center': 'Number of Shows Added Each Month'}, xlabel='Month', ylabel='Count'>



15 Number of shows per rating on each OTT platform

[57]:
```
platform_rating_distribution = df.groupby(['OTT Platform', 'rating']).size().
  ↪unstack().fillna(0)
platform_rating_distribution.plot(kind='bar', stacked=True, title='Number of␣
  ↪Shows per Rating on Each OTT Platform', xlabel='OTT Platform',␣
  ↪ylabel='Count')
```

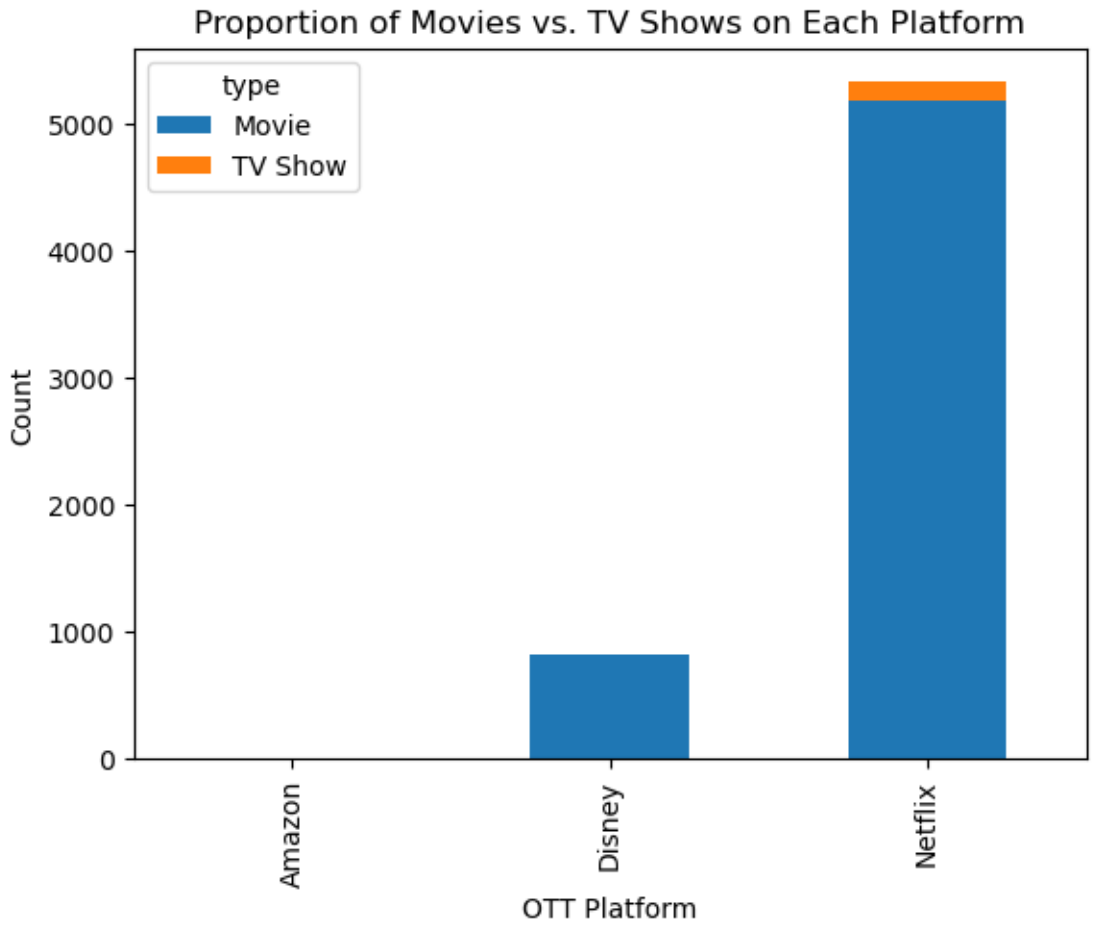
[57]: <Axes: title={'center': 'Number of Shows per Rating on Each OTT Platform'}, xlabel='OTT Platform', ylabel='Count'>



16 Proportion of movies vs. TV shows on each platform

[58]:
```
platform_type_distribution = df.groupby(['OTT Platform', 'type']).size().
↪unstack().fillna(0)
platform_type_distribution.plot(kind='bar', stacked=True, title='Proportion of␣
↪Movies vs. TV Shows on Each Platform', xlabel='OTT Platform', ylabel='Count')
```

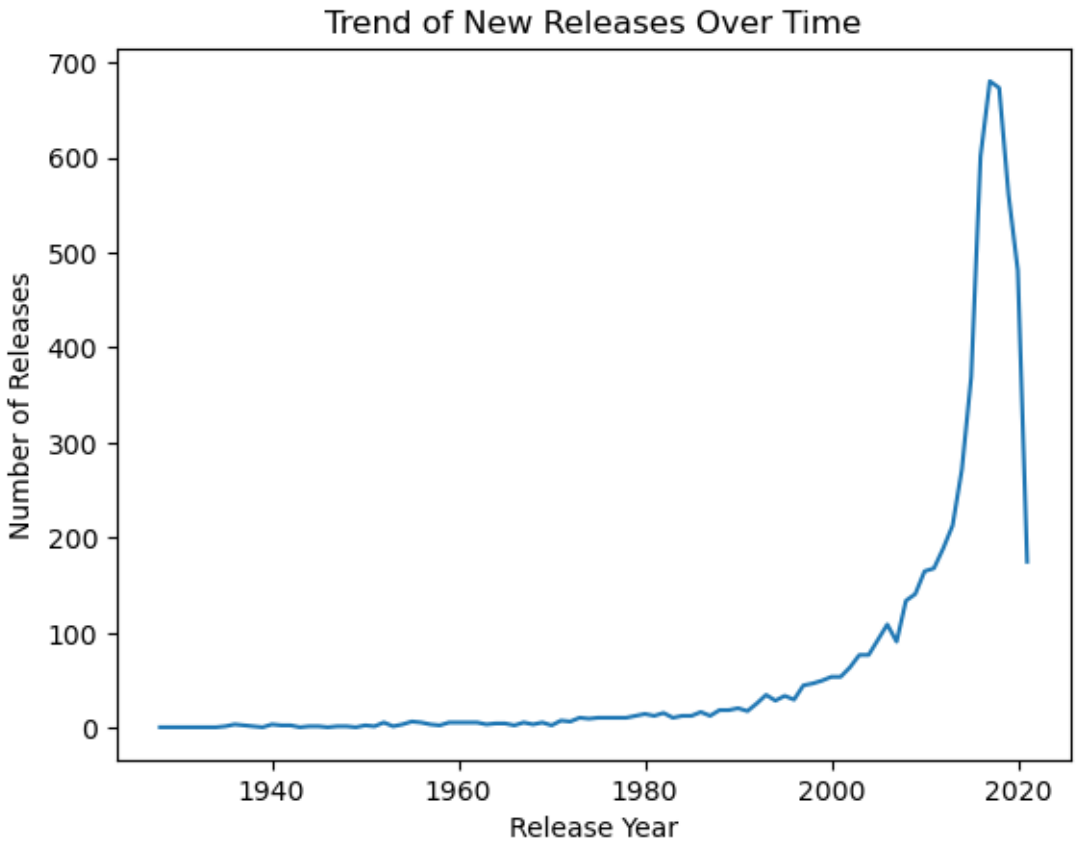
[58]:
```
<Axes: title={'center': 'Proportion of Movies vs. TV Shows on Each Platform'},
xlabel='OTT Platform', ylabel='Count'>
```



17 Trend of new releases over time

[59]:
```
new_releases_trend = df['release_year'].value_counts().sort_index()
new_releases_trend.plot(kind='line', title='Trend of New Releases Over Time',␣
↪xlabel='Release Year', ylabel='Number of Releases')
```

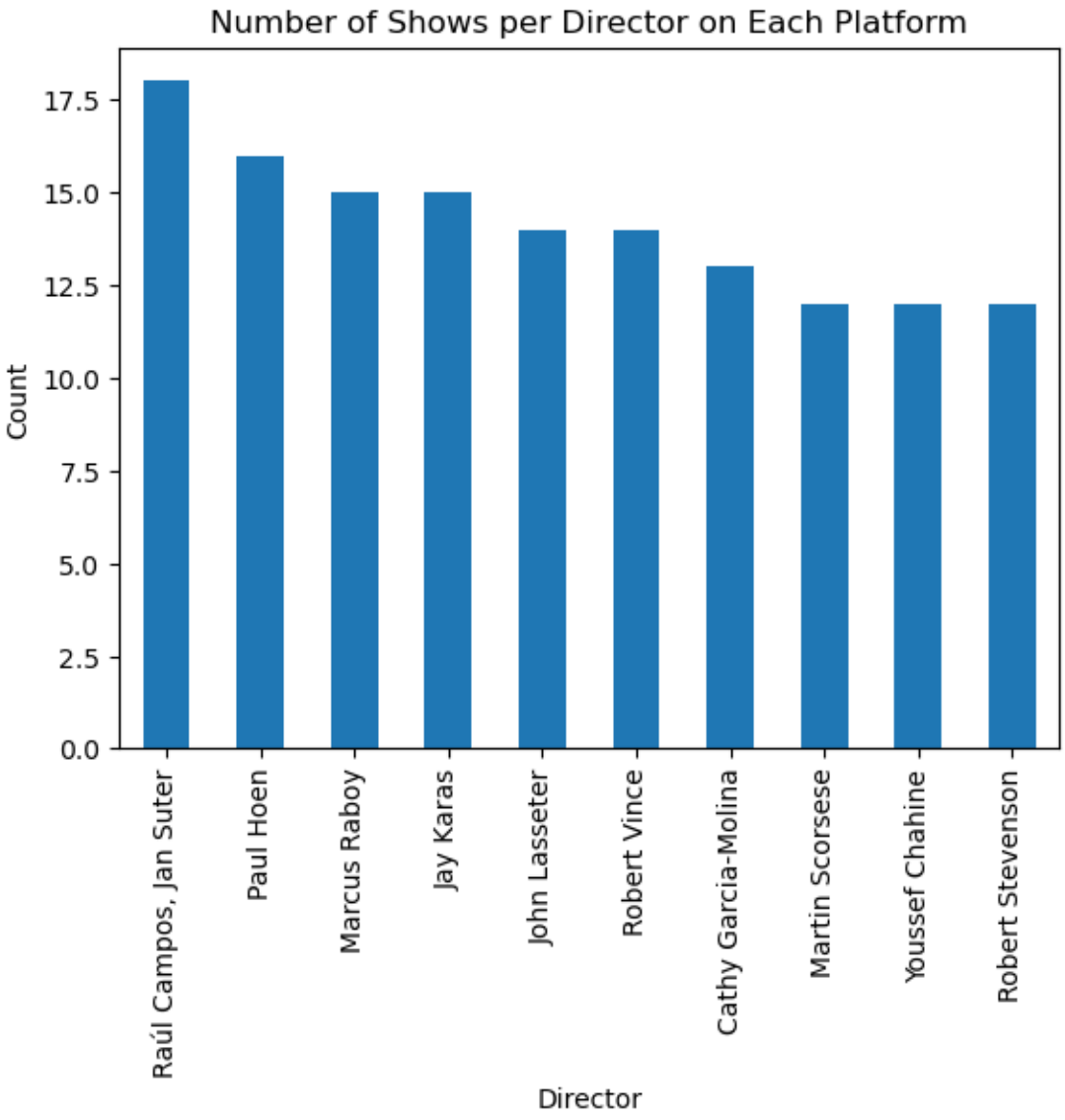
[59]:
```
<Axes: title={'center': 'Trend of New Releases Over Time'}, xlabel='Release
Year', ylabel='Number of Releases'>
```



18 Number of shows per director on each platform

[60]:
```
director_platform_distribution = df.groupby(['director', 'OTT Platform']).
  ↪size().unstack().fillna(0).sum(axis=1).sort_values(ascending=False).head(10)
director_platform_distribution.plot(kind='bar', title='Number of Shows per␣
  ↪Director on Each Platform', xlabel='Director', ylabel='Count')
```

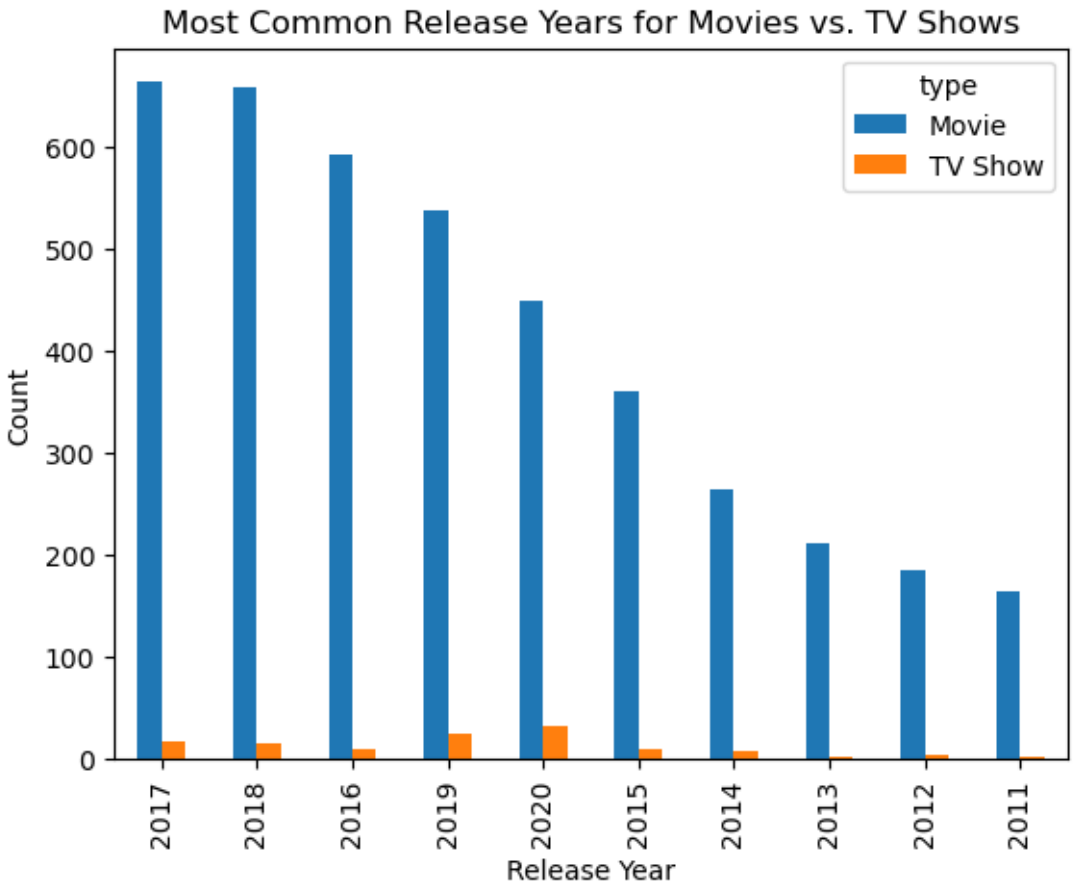
[60]:
```
<Axes: title={'center': 'Number of Shows per Director on Each Platform'},
xlabel='Director', ylabel='Count'>
```



19 Most common release years for movies vs. TV shows

[61]:
```
common_release_years = df.groupby(['release_year', 'type']).size().unstack().
↪fillna(0).sort_values(by='Movie', ascending=False).head(10)
common_release_years.plot(kind='bar', title='Most Common Release Years for␣
↪Movies vs. TV Shows', xlabel='Release Year', ylabel='Count')
```

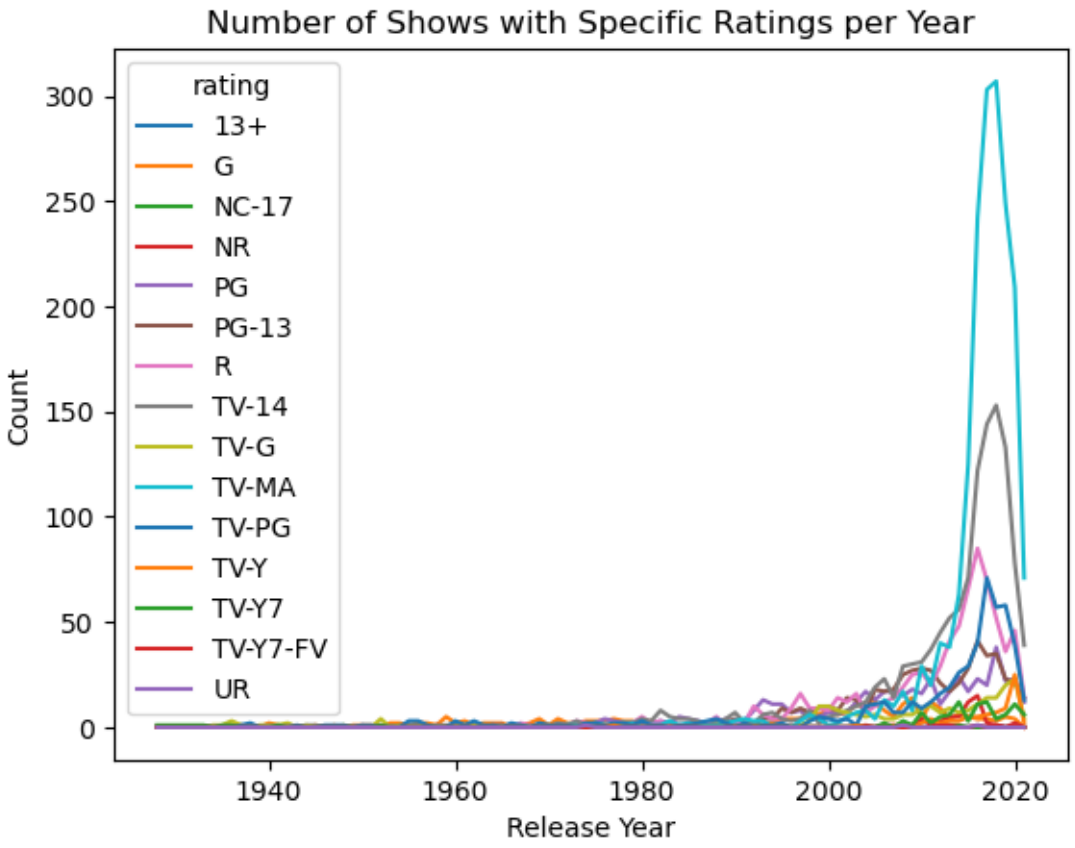
[61]:
```
<Axes: title={'center': 'Most Common Release Years for Movies vs. TV Shows'},
xlabel='Release Year', ylabel='Count'>
```



20 Number of shows with specific ratings per year

[64]:
```
ratings_per_year = df.groupby(['release_year', 'rating']).size().unstack().
↪fillna(0)
ratings_per_year.plot(kind='line', title='Number of Shows with Specific Ratings␣
↪per Year', xlabel='Release Year', ylabel='Count')
```

[64]:
```
<Axes: title={'center': 'Number of Shows with Specific Ratings per Year'},
xlabel='Release Year', ylabel='Count'>
```



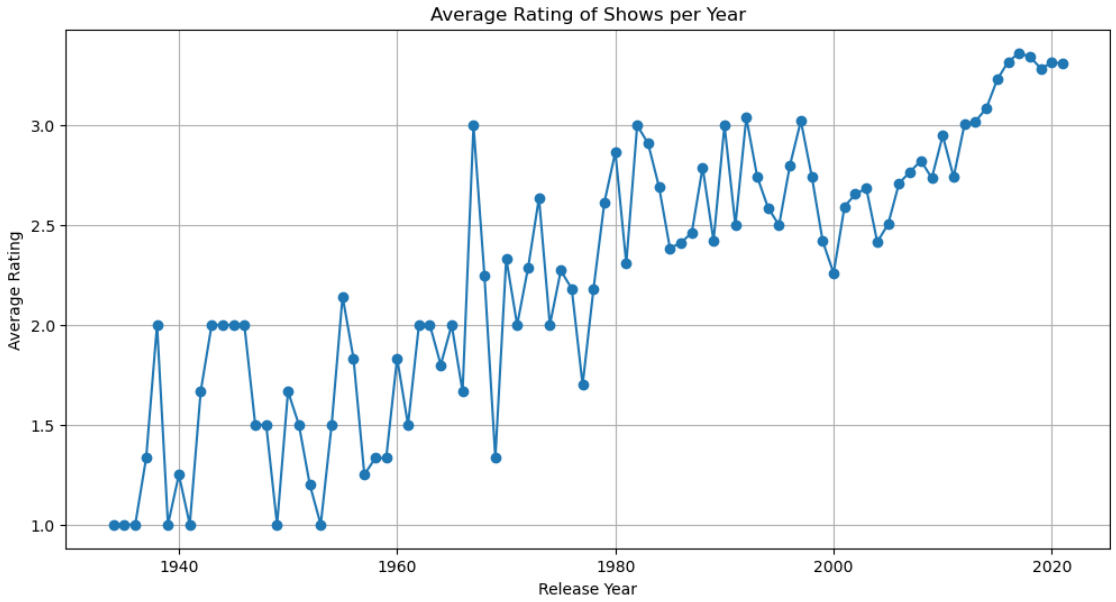
21 Average rating of shows per year

[66]:
```
print(df['rating'].unique())
```

```
['13+', 'TV-G', 'PG-13', 'G', 'PG', …, 'TV-MA', 'R', 'NC-17', 'NR', 'UR']
Length: 15
Categories (15, object): ['13+', 'G', 'NC-17', 'NR', …, 'TV-Y', 'TV-Y7',
'TV-Y7-FV', 'UR']
```

[70]:
```
import pandas as pd
import matplotlib.pyplot as plt


# Define a mapping for the ratings (example mapping)
rating_map = {
    'G': 1,
    'TV-G': 1,
    'PG': 2,
    'TV-PG': 2,
```

```
    'PG-13': 3,
    'TV-14': 3,
    'R': 4,
    'TV-MA': 4,
    'NC-17': 5,
    'NR'  : 0,
    'UR' : 0,

}

# Convert 'rating' using the mapping
df['numeric_rating'] = df['rating'].map(rating_map)

# Drop rows with NaN values in 'numeric_rating' or 'release_year'
df_clean = df.dropna(subset=['numeric_rating', 'release_year'])

# Calculate the average rating per year manually
average_rating_per_year = df_clean.groupby('release_year')['numeric_rating'].
 ↪mean().reset_index()

# Plot the results using matplotlib
plt.figure(figsize=(12, 6))
plt.plot(average_rating_per_year['release_year'],␣
 ↪average_rating_per_year['numeric_rating'], marker='o')
plt.title('Average Rating of Shows per Year')
plt.xlabel('Release Year')
plt.ylabel('Average Rating')
plt.grid(True)
plt.show()
```



[ ]: