Import Libraries

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving netflix.csv to netflix.csv
```

In [3]:
```
df=pd.read_csv('netflix.csv')
```

Basic Inspection

In [4]:
```
df.head()
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons

Business Objective

Netflix want to decide

1. What type of content to produce.
2. which genres perform better.
3. understand country-wise content strategy.
4. identify growth opportunities globally.

In [5]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [6]:
```
df.head()
```

Out[6]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons

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

Out[7]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

In [9]:
```
df.shape
```

Out[9]:
```
(8807, 12)
```

In [10]:
```
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
```

What observe:

- Number of rows:8807
- Number of column:12
- Datatype:int

Convert Categorical Columns

In [13]:
```
cat_cols = ['type','rating']
for col in cat_cols:
    df[col] = df[col].astype('category')
```

In [15]:
```
cat_cols
```

Out[15]: ['type', 'rating']

Check Missing Values

In [16]:
```
df.isnull().sum()
```

Out[16]:

	0
show_id	0
type	0
title	0
director	2634
cast	825
country	831
date_added	10
release_year	0
rating	4
duration	3
listed_in	0
description	0

dtype: int64

Missing value in column:

- director
- cast
- country
- date_added
- rating
- duration

Movies vs TV Shows

In [17]:
```
df['type'].value_counts()
```

Out[17]:

type	count
Movie	6131
TV Show	2676

dtype: int64

Top Countries

In [19]:
```
df['country'].value_counts().head(10)
```

Out[19]:

country	count
United States	2818
India	972
United Kingdom	419
Japan	245
South Korea	199
Canada	181
Spain	145
France	124
Mexico	110
Egypt	106

dtype: int64

Top Ratings

In [20]:
```
df['rating'].value_counts()
```

Out[20]:

rating	count
TV-MA	3207
TV-14	2160
TV-PG	863
R	799
PG-13	490
TV-Y7	334
TV-Y	307
PG	287
TV-G	220
NR	80
G	41
TV-Y7-FV	6
NC-17	3
UR	3
66 min	1
84 min	1
74 min	1

dtype: int64

It's clear:

1. Netflix has more movie the TV shows
2. USA contributes the highest content
3. TV-MA and TV-14 has highest rating

We must "unnest" columns like:

- Cast
- Director
- Country
- Listed_in (Genre)

Because they have multiple value sepreated by comma

Split Columns

In [27]:
```
df['country'] = df['country'].str.split(',')
```

```
df_country = df.explode('country')
```

In [29]:
```
df['listed_in'] = df['listed_in'].str.split(',')
df_genre = df.explode('listed_in')
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
/tmp/ipython-input-336390790.py in <cell line: 0>()
----> 1 df['listed_in'] = df['listed_in'].str.split(',')
      2 df_genre = df.explode('listed_in')

/usr/local/lib/python3.12/dist-packages/pandas/core/generic.py in __getattr__(self, name)
   6297         ):
   6298             return self[name]
-> 6299         return object.__getattribute__(self, name)
   6300
   6301     @final

/usr/local/lib/python3.12/dist-packages/pandas/core/accessor.py in __get__(self, obj, cls)
    222             # we're accessing the attribute of the class, i.e., Dataset.geo
    223             return self._accessor
--> 224         accessor_obj = self._accessor(obj)
    225         # Replace the property with the accessor object. Inspired by:
    226         # https://www.pydanny.com/cached-property.html

/usr/local/lib/python3.12/dist-packages/pandas/core/strings/accessor.py in __init__(self, data)
    189         from pandas.core.arrays.string_ import StringDtype
    190
--> 191         self._inferred_dtype = self._validate(data)
    192         self._is_categorical = isinstance(data.dtype, CategoricalDtype)
    193         self._is_string = isinstance(data.dtype, StringDtype)

/usr/local/lib/python3.12/dist-packages/pandas/core/strings/accessor.py in _validate(data)
    243
    244         if inferred_dtype not in allowed_types:
--> 245             raise AttributeError("Can only use .str accessor with string values!")
    246         return inferred_dtype
    247

AttributeError: Can only use .str accessor with string values!
```

In [30]:
```
df['cast'] = df['cast'].str.split(',')
df_cast = df.explode('cast')
```

In [31]:
```
df['director'] = df['director'].str.split(',')
df_director = df.explode('director')
```

In [32]:
```
df_genre
```

Out[32]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	durat
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	[United States]	September 25, 2021	2020	PG-13	90 r
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	[South Africa]	September 24, 2021	2021	TV-MA	Seas
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Sea
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Sea
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	[India]	September 24, 2021	2021	TV-MA	Seas
...	...	...	...	...	...	...	...	...	...	
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	[United States]	November 20, 2019	2007	R	158 r
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	Seas
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	[United States]	November 1, 2019	2009	R	88 r

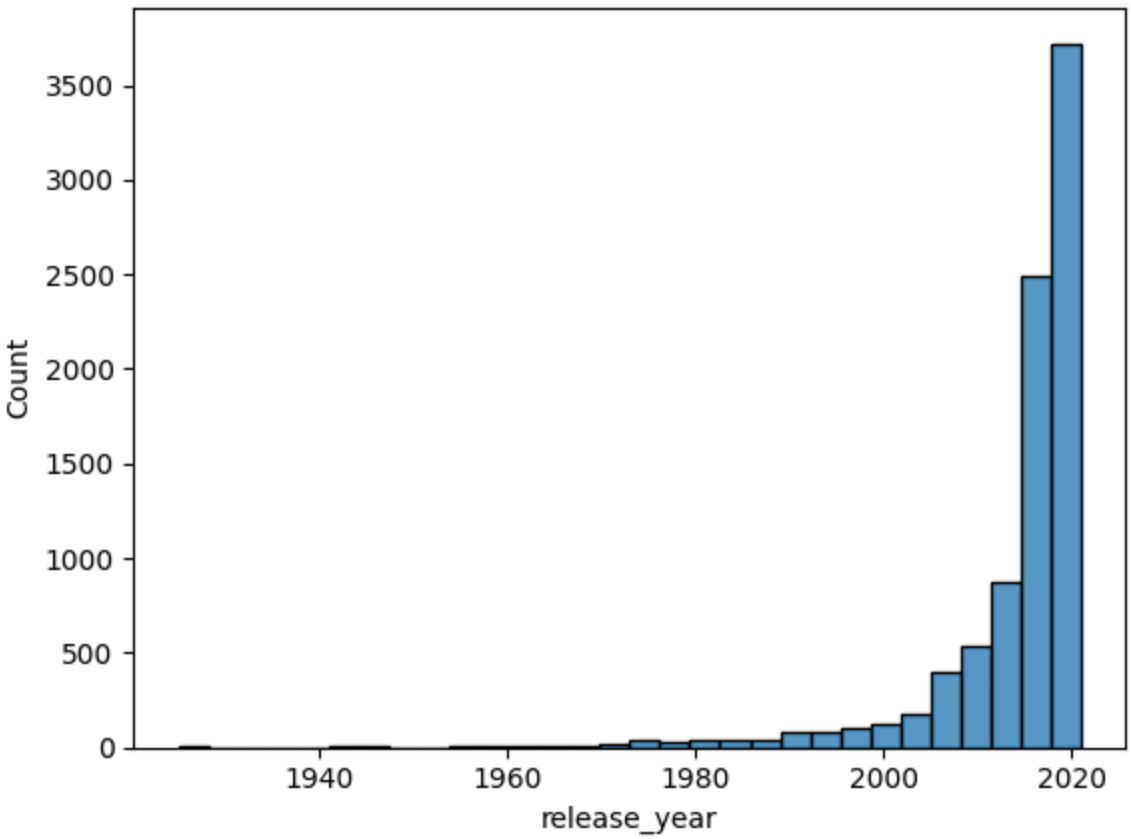
	show_id	type	title	director	cast	country	date_added	release_year	rating	durat
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	[United States]	January 11, 2020	2006	PG	88 r
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	[India]	March 2, 2019	2015	TV-14	111 r

8807 rows × 12 columns

Visual Analysis

Release Year Distribution

In [37]:
```
plt.figure()
sns.histplot(df['release_year'], bins=30)
plt.show()
```



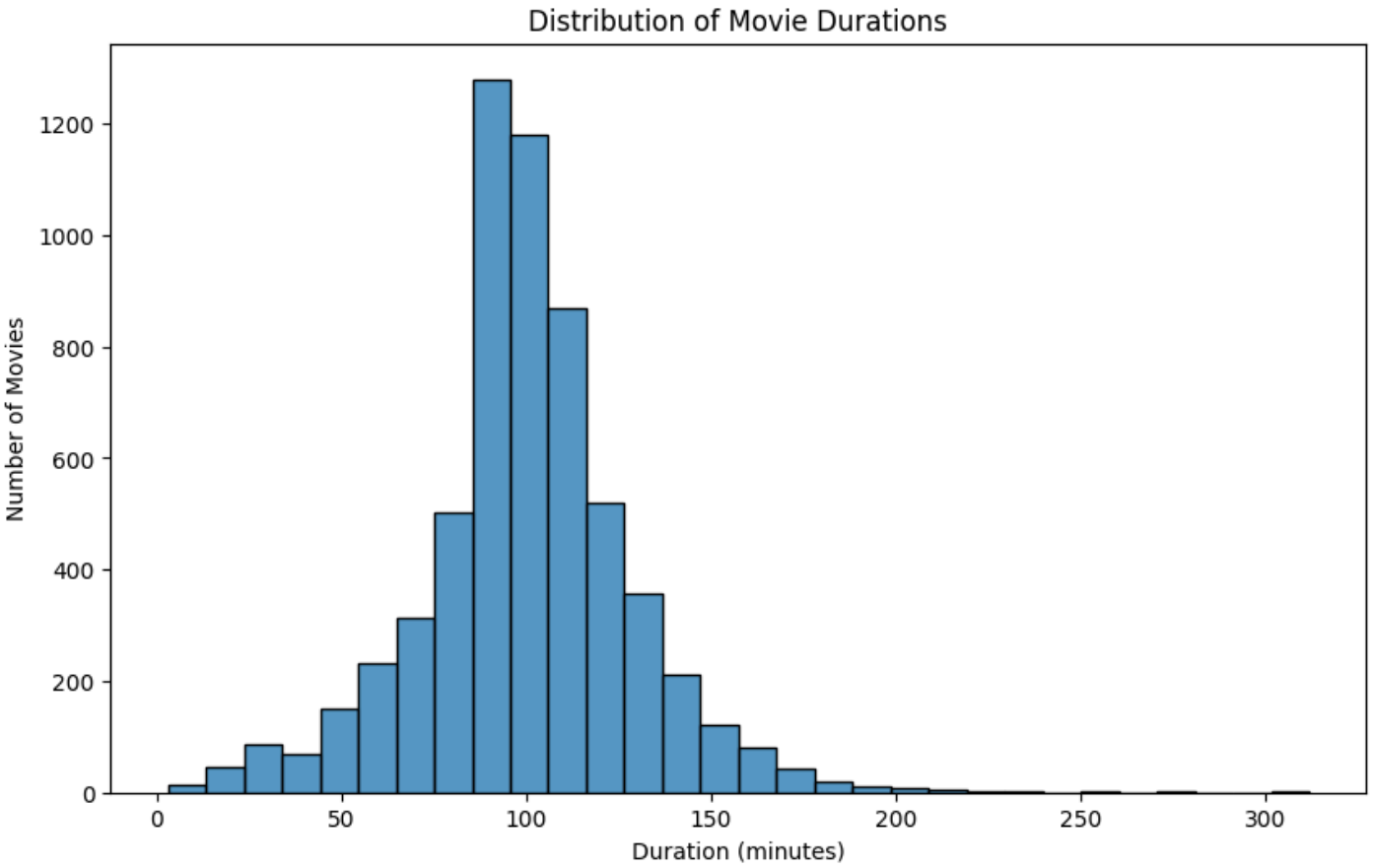
Insight:

- Massive increase in content after 2015

- Shows aggressive expansion strategy

In [39]:
```
df_movies = df[df['type']=='Movie'].copy()
df_movies['duration'] = df_movies['duration'].str.replace(' min','', regex=False)
df_movies.dropna(subset=['duration'], inplace=True)
df_movies['duration'] = df_movies['duration'].astype(int)

plt.figure(figsize=(10, 6))
sns.histplot(df_movies['duration'], bins=30)
plt.title('Distribution of Movie Durations')
plt.xlabel('Duration (minutes)')
plt.ylabel('Number of Movies')
plt.show()
```

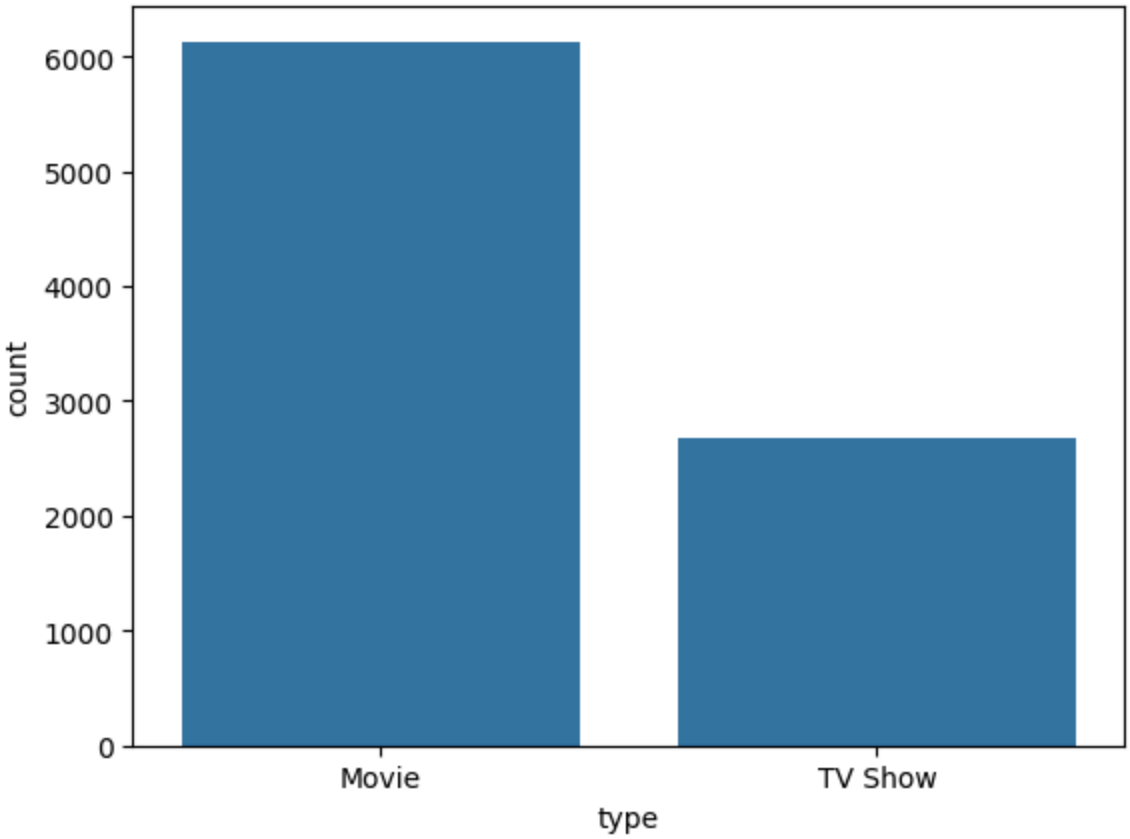


Insight:

Most movie are Durations between 80-120

Movies vs TV Shows countplot

In [40]:
```
plt.figure()
sns.countplot(x='type', data=df)
plt.show()
```

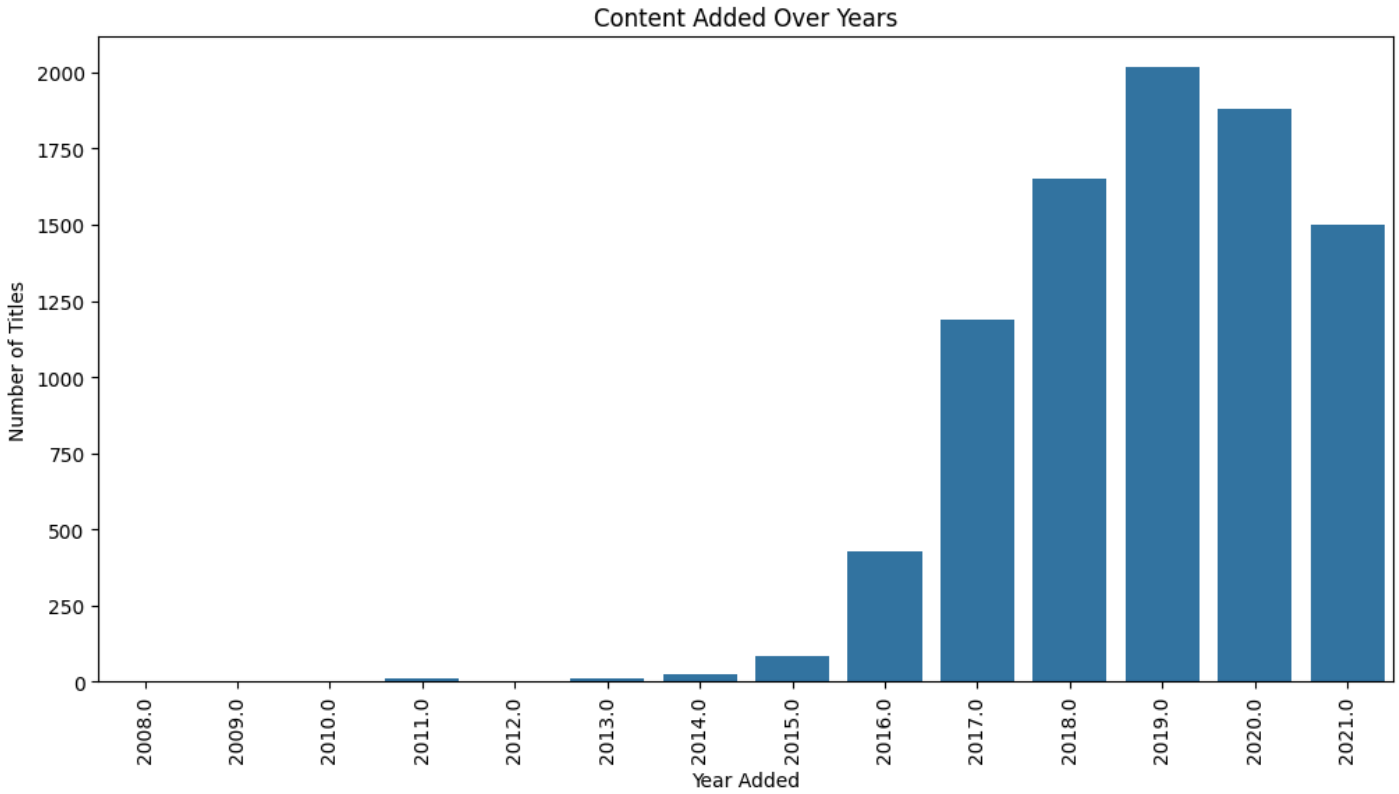


Movie dominating over TV-shows

Content Added Over Years

In [42]:
```
df['date_added'] = pd.to_datetime(df['date_added'], format='mixed')
df['year_added'] = df['date_added'].dt.year

plt.figure(figsize=(12,6))
sns.countplot(x='year_added', data=df)
plt.title('Content Added Over Years')
plt.xlabel('Year Added')
plt.ylabel('Number of Titles')
plt.xticks(rotation=90)
plt.show()
```

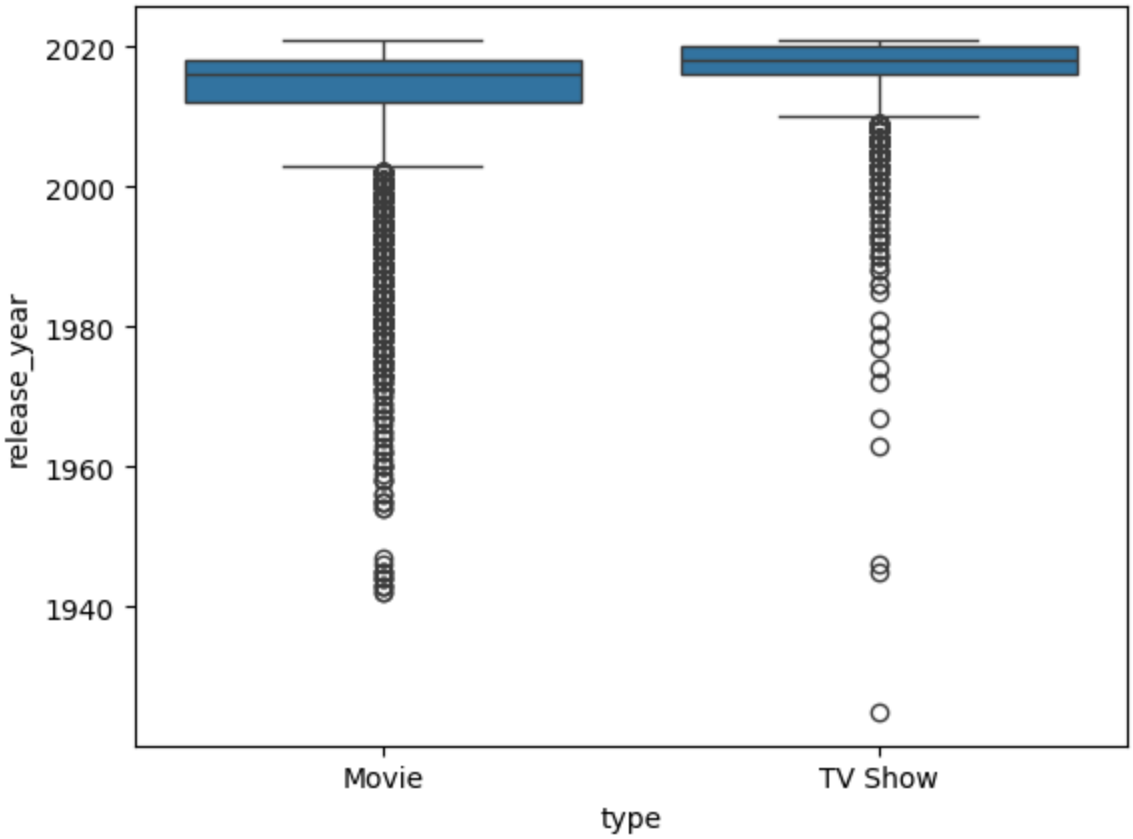


Insight :

- Rapid increase between 2016 to 2019

Type vs Release Year

In [43]:
```
plt.figure()
sns.boxplot(x='type', y='release_year', data=df)
plt.show()
```



Insight:

- In recent year TV-Show are more then movie.
- Netflix shifted focus towards TV-show.

Mising value

In [47]:
```
df['director']=df['director'].fillna('Unknown')
df['cast']=df['cast'].fillna('Unknown')
df['country']=df['country'].fillna('Unknown')
```

In [48]:
```
df['date_added'] = df['date_added'].astype(object).fillna('Unknown')
df['rating'] = df['rating'].cat.add_categories('Unknown')
df['rating'] = df['rating'].fillna('Unknown')
df['duration'] = df['duration'].fillna('Unknown')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/tmp/ipython-input-1739804598.py in <cell line: 0>()
      1 df['date_added'] = df['date_added'].astype(object).fillna('Unknown')
----> 2 df['rating'] = df['rating'].cat.add_categories('Unknown')
      3 df['rating'] = df['rating'].fillna('Unknown')
      4 df['duration'] = df['duration'].fillna('Unknown')

/usr/local/lib/python3.12/dist-packages/pandas/core/accessor.py in f(self, *args, **kwargs)
    110         def _create_delegator_method(name: str):
    111             def f(self, *args, **kwargs):
--> 112                 return self._delegate_method(name, *args, **kwargs)
    113
    114             f.__name__ = name

/usr/local/lib/python3.12/dist-packages/pandas/core/arrays/categorical.py in _delegate_method(sel
f, name, *args, **kwargs)
   2939
   2940         method = getattr(self._parent, name)
-> 2941         res = method(*args, **kwargs)
   2942         if res is not None:
   2943             return Series(res, index=self._index, name=self._name)

/usr/local/lib/python3.12/dist-packages/pandas/core/arrays/categorical.py in add_categories(self,
new_categories)
   1328         already_included = set(new_categories) & set(self.dtype.categories)
   1329         if len(already_included) != 0:
-> 1330             raise ValueError(
   1331                 f"new categories must not include old categories: {already_included}"
   1332             )

ValueError: new categories must not include old categories: {'Unknown'}
```

In [49]:
```
df.isnull().sum()
```

Out[49]:

	0
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	8807
description	0
year_added	10

dtype: int64

In [50]:
```
df['listed_in']=df['listed_in'].fillna('Unknown')
```

In [51]:
```
df.isnull().sum()
```

Out[51]:

	0
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
year_added	10

dtype: int64

In [52]:
```
df['year_added']=df['year_added'].fillna('Unknown')
```

In [53]:
```
df.isnull().sum()
```

Out[53]:

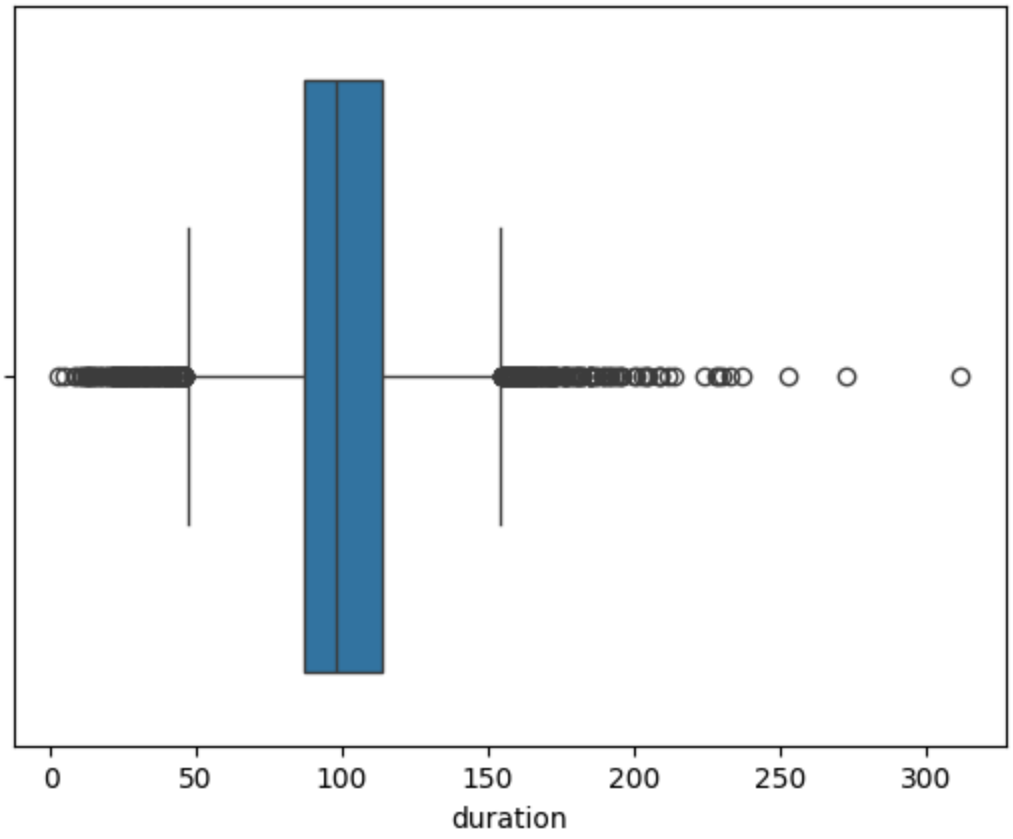
	0
show_id	0
type	0
title	0
director	0
cast	0
country	0
date_added	0
release_year	0
rating	0
duration	0
listed_in	0
description	0
year_added	0

dtype: int64

Missing categorical values replaced with "Unknown".

In [54]:
```
sns.boxplot(x=df_movies['duration'])
```

Out[54]:
```
<Axes: xlabel='duration'>
```



Few are long duration movie

Observation :

1. Netflix has more Movies than TV Shows overall.
2. However, in recent years, TV Shows are increasing rapidly.
3. Majority content comes from USA.
4. Content production increased drastically after 2015.
5. Most movies are around 90-120 minutes.
6. TV-MA rating dominates

patterns

- Netflix expanded aggressively after 2015.
- Strong dominance in US market.
- Recent shift toward TV Shows.
- Adult content drives engagement.

Content Strategy

- Increase production of TV Shows since trend is rising.
- Focus on 8-10 episode series.
- Continue producing 90-120 min movies.
- Invest in India, South Korea, and European content.

- Produce region-specific original shows.
- Collaborate with local directors & actors.

Release Strategy

- Launch major TV Shows during Q3-Q4 (Holiday seasons).

In [ ]: