

```
!gdown 1AXID_Fvkt6dNsJdEPlx-Vlea1UwD15Ht -O 'Netflix.csv'

Downloading...
From: https://drive.google.com/uc?id=1AXID\_Fvkt6dNsJdEPlx-Vlea1UwD15Ht
To: /content/Netflix.csv
100% 3.40M/3.40M [00:00<00:00, 195MB/s]
```

```
import numpy as np, pandas as pd, math, matplotlib.pyplot as plt, seaborn as sns
```

```
netflix = pd.read_csv('Netflix.csv')
netflix
```

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

Saved successfully!

```
netflix.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
```

```
netflix.isna().sum()
```

```
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
```

```
netflix['type'].value_counts()
```

| Movie | 6131 |
|--------------------|-------|
| TV Show | 2676 |
| Name: type, dtype: | int64 |

▼ Converted date_added to datetime

```
netflix['date_added'] = pd.to_datetime(netflix['date_added'].str.strip(),format='%B %d, %Y')
```

```
netflix.describe()
```

| 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 |

```
netflix.describe(include = 'object' )
```

| | show_id | type | title | director | cast | country | rating | duration | l1 |
|--------|---------|-------|--------------|---|-------|----------|--------|----------|----|
| count | 8807 | 8807 | 8807 | 6173 | 7982 | 7976 | 8803 | 8804 | |
| unique | 8807 | 2 | 8807 | 4528 | 7692 | 748 | 17 | 220 | |
| top | s1 | Movie | Dick Johnson | Rajiv Gupta, Anupam Bhattacharya, David United States | TV-MA | 1 Season | Inte | | |

Saved successfully!

Split Director Column on comma, fill missing value with '#' (which will be updated later)

▼ with best possible director) , melt it, strip individual director names to remove whitespaces

```
netflix['director'] = netflix['director'].str.split(',')
# Fill missing value of director with '#' and will replace this value later with an possible appropriate director
spdirectordf = pd.DataFrame(netflix['director'].fillna('#').tolist()).add_prefix('director_')
len(max(netflix[~netflix['director'].isna()]['director'], key=len))
# Number of columns is as per max number of directors we have for a movie in the dataset.
```

13

```
spdirectordf.head()
```

| | director_0 | director_1 | director_2 | director_3 | director_4 | director_5 | director_6 |
|---|-----------------|------------|------------|------------|------------|------------|------------|
| 0 | Kirsten Johnson | None | None | None | None | None | None |
| 1 | # | None | None | None | None | None | None |
| 2 | Julien Leclercq | None | None | None | None | None | None |
| 3 | # | None | None | None | None | None | None |

```
netflixWithspDirector = pd.concat([netflix, spdirectordf],axis =1)
netflixWithspDirector = pd.melt(netflixWithspDirector,id_vars=netflix.columns.tolist())
netflixWithspDirector.drop(['director'],axis=1,inplace=True)
```

```

netflixWithspDirector.rename({'value':'director'},axis = 1,inplace=True)
# IMPORTANT : For movies where we have less number of director than max number of directors we have for a movie, None is added
# IMPORTANT : After MELTING (director will be transformed in a single column) we will REMOVE the rows having None as value for director to
netflixWithspDirector = netflixWithspDirector[~netflixWithspDirector['director'].isna()]
netflixWithspDirector = netflixWithspDirector.loc[:,['show_id','type','director']]
netflixWithspDirector.head()



| show_id | type | director                |  |
|---------|------|-------------------------|--|
| 0       | s1   | Movie Kirsten Johnson   |  |
| 1       | s2   | TV Show #               |  |
| 2       | s3   | TV Show Julien Leclercq |  |
| 3       | s4   | TV Show #               |  |
| 4       | s5   | TV Show #               |  |



# Strip out Starting and Trailing WhiteSpaces
netflixWithspDirector['director'] = netflixWithspDirector['director'].str.strip()
netflixWithspDirector['director'].unique()
# Column Director with value '#' denotes missing value in director column of original dataset
# Will replace this value later with an possible appropriate director

array(['Kirsten Johnson', '#', 'Julien Leclercq', ..., 'Hiroshi Yamazaki',
       'James Gunn', 'Mark Henn'], dtype=object)

netflixWithspDirector['show_id'].nunique()

8807

```

Split listed_in Column on comma, there are no missing values as per .info() did in upper cells, melt it, strip individual genre names to remove whitespaces

```

# Doing Same operation for listed-in

netflix['listed_in'] = netflix['listed_in'].str.split(',')

#IMPORTANT Didnot used fillna because there is no null for listed_in column as inferred by .info method in previous cell
spgenredf = pd.DataFrame(netflix['listed_in'].tolist()).add_prefix('genre_')
Saved successfully! [X] .isna()]['listed_in'], key=len))
# Number of columns is as per max number of listed_in we have for a movie in the dataset.

3

```

```
spgenredf.head()
```

| | genre_0 | genre_1 | genre_2 | |
|---|------------------------|------------------------|-----------------------|--|
| 0 | Documentaries | None | None | |
| 1 | International TV Shows | TV Dramas | TV Mysteries | |
| 2 | Crime TV Shows | International TV Shows | TV Action & Adventure | |
| 3 | Docuseries | Reality TV | None | |
| 4 | International TV Shows | Romantic TV Shows | TV Comedies | |

```

netflixWithspGenre = pd.concat([netflix,spgenredf],axis =1)
netflixWithspGenre = pd.melt(netflixWithspGenre,id_vars=netflix.columns.tolist())
netflixWithspGenre.rename({'value':'genre'},axis = 1,inplace=True)
# IMPORTANT : For movies where we have less number of listed_in than max number of listed_in we have for a movie, None is added
# IMPORTANT : After melting (multiple genre column will be transformed in a single column named genre) we will remove the rows having None
netflixWithspGenre = netflixWithspGenre[~netflixWithspGenre['genre'].isna()]
netflixWithspGenre = netflixWithspGenre.loc[:,['show_id','type','genre']]
netflixWithspGenre.head()

```

```

show_id      type      genre
0           s1   Movie   Documentaries
1           s2 TV Show International TV Shows
# Strip out Starting and Trailing Whitespaces
netflixWithspGenre['genre'] = netflixWithspGenre['genre'].str.strip()
netflixWithspGenre['genre'].unique()

array(['Documentaries', 'International TV Shows', 'Crime TV Shows',
       'Docuseries', 'TV Dramas', 'Children & Family Movies', 'Dramas',
       'British TV Shows', 'Comedies', 'TV Comedies', 'Thrillers',
       'Horror Movies', "Kids' TV", 'Action & Adventure', 'Reality TV',
       'Anime Series', 'International Movies', 'Sci-Fi & Fantasy',
       'Classic Movies', 'TV Shows', 'Stand-Up Comedy',
       'TV Action & Adventure', 'Movies', 'Stand-Up Comedy & Talk Shows',
       'Classic & Cult TV', 'Anime Features', 'Romantic TV Shows',
       'Cult Movies', 'Independent Movies', 'TV Horror',
       'Spanish-Language TV Shows', 'Music & Musicals', 'Romantic Movies',
       'LGBTQ Movies', 'TV Sci-Fi & Fantasy', 'Sports Movies',
       'Korean TV Shows', 'Science & Nature TV', 'Faith & Spirituality',
       'Teen TV Shows', 'TV Mysteries', 'TV Thrillers'], dtype=object)

netflixWithspGenre['show_id'].nunique()

8807

netflix['country'].value_counts()
#Looks Like country column also contains multiple comma seperated values

United States          2818
India                  972
United Kingdom         419
Japan                  245
South Korea             199
...
Romania, Bulgaria, Hungary    1
Uruguay, Guatemala        1
France, Senegal, Belgium     1
Mexico, United States, Spain, Colombia  1
United Arab Emirates, Jordan    1
Name: country, Length: 748, dtype: int64

```

Split Country Column on comma, fill missing value with '#' (which will be updated later)

Saved successfully! country), melt it, strip individual country names to remove whitespaces

```

# Doing Same operation for listed-in

netflix['country'] = netflix['country'].str.split(',')

spcountrydf = pd.DataFrame(netflix['country'].fillna('#').tolist()).add_prefix('country_')

len(max(netflix[~netflix['country'].isna()]['country'], key=len))
# Number of columns is as per max number of country we have for a movie in the dataset.

```

12

spcountrydf.head()

| | country_0 | country_1 | country_2 | country_3 | country_4 | country_5 | country_6 | count |
|---|---------------|-----------|-----------|-----------|-----------|-----------|-----------|-------|
| 0 | United States | None | None | None | None | None | None | 1 |
| 1 | South Africa | None | None | None | None | None | None | 1 |
| 2 | # | None | None | None | None | None | None | 1 |
| 3 | # | None | None | None | None | None | None | 1 |

```

netflixWithspCountry = pd.concat([netflix, spcountrydf], axis=1)
netflixWithspCountry = pd.melt(netflixWithspCountry, id_vars=netflix.columns.tolist())
netflixWithspCountry.drop(['country'], axis=1, inplace=True)
netflixWithspCountry.rename({'value': 'country'}, axis=1, inplace=True)

```

```
# IMPORTANT : For movies where we have less number of country than max number of country we have for a movie, None is added
# IMPORTANT : After melting (multiple country column will be transformed in a single column named country) we will remove
# the rows having None as value for country because they were added due to above step and is not needed
netflixWithspCountry = netflixWithspCountry[~netflixWithspCountry['country'].isna()]
netflixWithspCountry = netflixWithspCountry.loc[:,['show_id','type','country']]
netflixWithspCountry.head()
```

| show_id | type | country |
|---------|------|----------------------|
| 0 | s1 | Movie United States |
| 1 | s2 | TV Show South Africa |
| 2 | s3 | TV Show # |
| 3 | s4 | TV Show # |
| 4 | s5 | TV Show India |

```
# Strip out Starting and Trailing WhiteSpaces
netflixWithspCountry['country'] = netflixWithspCountry['country'].str.strip()
netflixWithspCountry['country'].unique()
# Column country with value '#' denotes missing value in country column of original dataset
# Will replace this value later with an possible appropriate country if needed

array(['United States', 'South Africa', '#', 'India', 'United Kingdom',
       'Germany', 'Mexico', 'Turkey', 'Australia', 'Finland', 'China',
       'Nigeria', 'Japan', 'Spain', 'France', 'Belgium', 'South Korea',
       'Argentina', 'Russia', 'Canada', 'Hong Kong', 'Italy', '',
       'Ireland', 'New Zealand', 'Jordan', 'Colombia', 'Switzerland',
       'Israel', 'Brazil', 'Taiwan', 'Bulgaria', 'Poland', 'Saudi Arabia',
       'Thailand', 'Indonesia', 'Egypt', 'Kuwait', 'Malaysia', 'Vietnam',
       'Sweden', 'Lebanon', 'Romania', 'Philippines', 'Iceland',
       'Denmark', 'United Arab Emirates', 'Netherlands', 'Norway',
       'Syria', 'Mauritius', 'Austria', 'Czech Republic', 'Cameroon',
       'Uruguay', 'Kenya', 'Chile', 'Luxembourg', 'Bangladesh',
       'Portugal', 'Hungary', 'Senegal', 'Singapore', 'Serbia', 'Namibia',
       'Peru', 'Mozambique', 'Belarus', 'Ghana', 'Zimbabwe',
       'Puerto Rico', 'Pakistan', 'Cyprus', 'Paraguay', 'Croatia',
       'Cambodia', 'Georgia', 'Soviet Union', 'Greece', 'West Germany',
       'Iran', 'Venezuela', 'Slovenia', 'Guatemala', 'Ukraine', 'Jamaica',
       'Somalia', 'Nepal', 'Palestine', 'Algeria', 'Malta', 'Angola',
       'Iraq', 'Malawi', 'Qatar', 'Morocco', 'Slovakia', 'Bermuda',
       'Armenia', 'Sri Lanka', 'Cuba', 'Nicaragua', 'Azerbaijan',
       'Botswana', 'Vatican City', 'Kazakhstan', 'Lithuania',
       'East Germany', 'Burkina Faso', 'Cayman Islands', 'Albania',
       'Ecuador', 'Dominican Republic', 'Samoa', 'Sudan', 'Latvia',
       'Liechtenstein', 'Panama', 'Uganda', 'Montenegro', 'Bahamas',
       '', 'Mongolia'], dtype=object)
```

Saved successfully!

```
netflixWithspCountry['show_id'].nunique()
```

8807

Split Cast Column on comma, fill missing value with '#' (which will be updated later with best possible cast) , melt it, strip individual cast names to remove whitespaces

```
spcastdf = netflix['cast'].fillna('#').str.split(',',expand = True).add_prefix('cast_')

# Number of columns is as per max number of cast we have for a movie in the dataset.
spcastdf
```

| | cast_0 | cast_1 | cast_2 | cast_3 | cast_4 | cast_5 | cast_6 |
|-----|---------------|----------------|---------------|----------------|------------------|------------------|--------------------------|
| 0 | # | None | None | None | None | None | None |
| 1 | Ama Qamata | Khosi Ngema | Gail Mabalane | Thabang Molaba | Dillon Windvogel | Natasha Thahane | Arno Greeff Tsh |
| 2 | Sami Bouajila | Tracy Gotoas | Samuel Jouy | Nabiha Akkari | Sofia Lesaffre | Salim Kechiouche | Noureddine Farihi Ge Ram |
| 3 | # | None | None | None | None | None | None |
| 4 | Mayur More | Jitendra Kumar | Ranjan Raj | Alam Khan | Ahsaas Channa | Revathi Pillai | Urvi Singh Arun |
| ... | ... | ... | ... | ... | ... | ... | ... |

```
netflixWithspCast = pd.concat([netflix,spcastdf],axis =1)
netflixWithspCast = pd.melt(netflixWithspCast,id_vars=netflix.columns.tolist())
netflixWithspCast.drop(['cast'],axis=1,inplace=True)
netflixWithspCast.rename({'value':'cast'},axis = 1,inplace=True)
# IMPORTANT : For movies where we have less number of cast than max number of cast we have for a movie, None is added
# IMPORTANT : After melting (multiple cast column will be transformed in a single column named cast) we will remove
# the rows having None as value for cast because they were added due to above step and is not needed
netflixWithspCast = netflixWithspCast[~netflixWithspCast['cast'].isna()]
netflixWithspCast = netflixWithspCast.loc[:,['show_id','type','cast']]
netflixWithspCast.head()
```

| show_id | type | cast | edit |
|---------|------|---------|---------------|
| 0 | s1 | Movie | # |
| 1 | s2 | TV Show | Ama Qamata |
| 2 | s3 | TV Show | Sami Bouajila |
| 3 | s4 | TV Show | # |
| 4 | s5 | TV Show | Mayur More |

```
# Strip out Starting and Trailing WhiteSpaces
netflixWithspCast['cast'] = netflixWithspCast['cast'].str.strip()
netflixWithspCast['cast'].value_counts()
# Column cast with value '#' denotes missing value in cast column of original dataset
# Will replace this value later with an possible appropriate cast if needed
```

```
# 825
Anupam Kher 43
```

Saved successfully!

```
...
Kim Hyun-wook 1
Élodie Bouchez 1
Anthony Quayle 1
Elva Josephson 1
Ayize Ma'at 1
Name: cast, Length: 36440, dtype: int64
```

```
netflixWithspCast['show_id'].nunique()
```

```
8807
```

```
netflixWithspDirector, netflixWithspGenre , netflixWithspCountry, netflixWithspCast
```

```
(  show_id      type      director
0      s1    Movie    Kirsten Johnson
1      s2  TV Show      #
2      s3  TV Show  Julien Leclercq
3      s4  TV Show      #
4      s5  TV Show      #
...
95585  s7516  Movie   Rusty Cundieff
102764 s5888  Movie    Mike Gabriel
103787 s6911  Movie  Hiroshi Yamazaki
104392 s7516  Movie   James Gunn
111571 s5888  Movie    Mark Henn
[9612 rows x 3 columns],
  show_id      type      genre
0      s1    Movie  Documentaries
1      s2  TV Show  International TV Shows
2      s3  TV Show    Crime TV Shows
3      s4  TV Show   Docuseries
4      s5  TV Show  International TV Shows
...
...
```

| | | | |
|-------|-------|---------|------------------|
| 26414 | s8801 | TV Show | TV Dramas |
| 26415 | s8802 | Movie | Thrillers |
| 26416 | s8803 | Movie | Thrillers |
| 26417 | s8804 | TV Show | TV Comedies |
| 26420 | s8807 | Movie | Music & Musicals |

```
[19323 rows x 3 columns],  
   show_id      type        country  
0          s1  Movie  United States  
1          s2  TV Show  South Africa  
2          s3  TV Show          #  
3          s4  TV Show          #  
4          s5  TV Show  India
```

| | | | |
|--------|-------|-------|---------------|
| 78859 | s8404 | Movie | Germany |
| 85496 | s6234 | Movie | Sweden |
| 87666 | s8404 | Movie | Netherlands |
| 94030 | s6234 | Movie | United States |
| 103110 | s6234 | Movie | Uruguay |

```
[10850 rows x 3 columns],
   show_id      type      cast
0          s1    Movie      #
1          s2  TV Show  Ama Qamata
2          s3  TV Show  Sami Bouajila
3          s4  TV Show      #
4          s5  TV Show  Mayur More
...
417703    s3775  TV Show  Jon Hamm
424590    s1855  TV Show  Ayize Ma'tat
426510    s3775  TV Show  Oona Chaplin
433397    s1855  TV Show  Lovie Simone
435317    s3775  TV Show  Rafe Spall
```

[64951 rows x 3 columns])

- ▼ Categorized Genres in below categories

1. Comedy : 'Comedies', 'TV Comedies', 'Stand-Up Comedy', 'Stand-Up Comedy & Talk Shows'
 2. Romance : 'Romantic Movies', 'Romantic TV Shows'
 3. Horror : 'Horror Movies', 'TV Horror'
 4. Thrillers : 'Thrillers', 'TV Thrillers', 'TV Mysteries', 'Crime TV Shows',
 5. Sci-Fi : 'Sci-Fi & Fantasy', 'Science & Nature TV', 'TV Sci-Fi & Fantasy'

Saved successfully!  TV Shows' Spanish-Language TV Shows,"British TV Shows"

8. Family & Kids : "Kids' TV", 'Anime Series' , 'Children & Family Movies' , 'Anime Features'
 9. Documentary : 'Documentaries' , 'Docuseries'
 10. Global : 'International TV Shows' , 'International Movies'
 11. Action & Adventure : 'Action & Adventure' , 'TV Action & Adventure' , 'Reality TV' , 'Teen TV Shows'
 12. Classic : 'Classic Movies' , 'Classic & Cult TV' , 'Cult Movies'
 13. 'LGBTQ Movies'
 14. 'Faith & Spirituality'
 15. 'Sports Movies'
 16. Other Movies : 'Movies' , 'Independent Movies'
 17. 'Music & Musicals'

```

netflixWithspGenre.loc[ ( netflixWithspGenre['genre'] == 'TV Dramas' ) | ( netflixWithspGenre['genre'] == 'Dramas' ) | ( netflixWithspGenre['genre'] == 'Documentaries' ) | ( netflixWithspGenre['genre'] == 'Comedies' ) | ( netflixWithspGenre['genre'] == 'Thrillers' ) | ( netflixWithspGenre['genre'] == 'Horror Movies' ) | ( netflixWithspGenre['genre'] == 'Romantic TV Shows' ) | ( netflixWithspGenre['genre'] == 'Sci-Fi & Fantasy' ) | ( netflixWithspGenre['genre'] == 'Romantic Movies' ) | ( netflixWithspGenre['genre'] == 'LGBTQ Movies' ) | ( netflixWithspGenre['genre'] == 'Science & Nature TV' ) | ( netflixWithspGenre['genre'] == 'Faith & Spirituality' ) | ( netflixWithspGenre['genre'] == 'Sports Movies' ) | ( netflixWithspGenre['genre'] == 'Teen TV Shows' ) | ( netflixWithspGenre['genre'] == 'TV Thrillers' ) | ( netflixWithspGenre['genre'] == 'Korean TV Shows' ) | ( netflixWithspGenre['genre'] == 'Spanish-Language TV Shows' ) | ( netflixWithspGenre['genre'] == 'International TV Shows' ) | ( netflixWithspGenre['genre'] == 'Independent Movies' ), 'simplified_genre']

netflixWithspGenre['genre'].unique()

array(['Documentaries', 'International TV Shows', 'Crime TV Shows',
       'Docuseries', 'TV Dramas', 'Children & Family Movies', 'Dramas',
       'British TV Shows', 'Comedies', 'TV Comedies', 'Thrillers',
       'Horror Movies', 'Kids' TV', 'Action & Adventure', 'Reality TV',
       'Anime Series', 'International Movies', 'Sci-Fi & Fantasy',
       'Classic Movies', 'TV Shows', 'Stand-Up Comedy',
       'TV Action & Adventure', 'Movies', 'Stand-Up Comedy & Talk Shows',
       'Classic & Cult TV', 'Anime Features', 'Romantic TV Shows',
       'Cult Movies', 'Independent Movies', 'TV Horror',
       'Spanish-Language TV Shows', 'Music & Musicals', 'Romantic Movies',
       'LGBTQ Movies', 'TV Sci-Fi & Fantasy', 'Sports Movies',
       'Korean TV Shows', 'Science & Nature TV', 'Faith & Spirituality',
       'Teen TV Shows', 'TV Mysteries', 'TV Thrillers'], dtype=object)

```

Saved successfully! ×

Replace default value for Director ('#') which are present due to missing data in original

netflix dataset , with director having most movies in that genre and show type

```
netflixWithspDirector[netflixWithspDirector['director'] == '#']
```

| show_id | type | director | edit |
|---------|-------|----------|------|
| 1 | s2 | TV Show | # |
| 3 | s4 | TV Show | # |
| 4 | s5 | TV Show | # |
| 10 | s11 | TV Show | # |
| 14 | s15 | TV Show | # |
| ... | ... | ... | ... |
| 8795 | s8796 | TV Show | # |
| 8796 | s8797 | TV Show | # |
| 8797 | s8798 | TV Show | # |
| 8800 | s8801 | TV Show | # |
| 8803 | s8804 | TV Show | # |

2634 rows × 3 columns

```
netflixWithspGenreAndDirector = netflixWithspGenre.merge(netflixWithspDirector, on = 'show_id')
netflixWithspGenreAndDirector['type'] = netflixWithspGenreAndDirector['type_x']
```

```
netflixWithspGenreAndDirector.drop(['type_x','type_y'],axis =1 , inplace=True)
netflixWithspGenreAndDirector.head()
```

| show_id | genre | simplified_genre | director | type | |
|---------|-------|------------------------|-------------|-----------------|---------|
| 0 | s1 | Documentaries | Documentary | Kirsten Johnson | Movie |
| 1 | s2 | International TV Shows | Global | # | TV Show |
| 2 | s2 | TV Dramas | Drama | # | TV Show |
| 3 | s2 | TV Mysteries | Thrillers | # | TV Show |
| 4 | s3 | Crime TV Shows | Thrillers | Julien Leclercq | TV Show |

```
directorHavingMostMoviesinAGenreAndShowType = netflixWithspGenreAndDirector[~(netflixWithspGenreAndDirector['director'] == '#')].groupby('genre').size().reset_index()
directorHavingMostMoviesinAGenreAndShowType.rename({'director': 'PossibleDirector'}, axis=1, inplace=True)
directorHavingMostMoviesinAGenreAndShowType.head()
```

| genre | type | PossibleDirector | |
|--------------------------|---------|------------------|--|
| Action & Adventure | Movie | Ömer Faruk Sorak | |
| Anime Features | Movie | Zhao Ji | |
| Anime Series | TV Show | Yasuhiro Irie | |
| British TV Shows | TV Show | Toby Haynes | |
| Children & Family Movies | Movie | Éric Warin | |

```
netflixWithspGenreAndDirector = netflixWithspGenreAndDirector.merge(directorHavingMostMoviesinAGenreAndShowType, on='genre')
netflixWithspGenreAndDirector['type'] = netflixWithspGenreAndDirector['type_x']
netflixWithspGenreAndDirector.drop(['type_x','type_y'],axis =1 , inplace=True)
netflixWithspGenreAndDirector
```

| show_id | genre | simplified_genre | director | PossibleDirector | type |
|---------|-------|-------------------|-------------|------------------------|-----------|
| 0 | s1 | Documentaries | Documentary | Kirsten Johnson | Movie |
| 1 | s17 | Documentaries | Documentary | Pedro de Echave García | Movie |
| 2 | s17 | Documentaries | Documentary | Pablo Azorín Williams | Movie |
| 4 | s69 | Documentaries | Documentary | Tyler Greco | Movie |
| | | | | Hanns-Bruno Kammercöns | |
| ... | ... | ... | ... | ... | ... |
| 20909 | s7722 | Classic & Cult TV | Classic | # | Sunny Lau |
| 20910 | s8190 | Classic & Cult TV | Classic | # | Sunny Lau |

```
netflixWithspGenreAndDirector.loc[netflixWithspGenreAndDirector['director'] == '#', 'director'] = netflixWithspGenreAndDirector.loc[netflixWithspGenreAndDirector['genre'] == 'Documentaries', 'director']
```

```
netflixWithspGenreAndDirector
```

```

show_id      genre simplified_genre   director PossibleDirector    type
Kirsten
netflixWithspGenreAndDirector = netflixWithspGenreAndDirector.loc[:,['show_id','type','genre','simplified_genre','director']]
netflixWithspGenreAndDirector
# Director ('#') which are present due to missing data in original netflix dataset , with director having most movies in that genre and s

      show_id   type      genre simplified_genre   director
0       s1     Movie Documentaries Documentary Kirsten Johnson
1      s17     Movie Documentaries Documentary Pedro de Echave García
2      s17     Movie Documentaries Documentary Pablo Azorín Williams
3      s46     Movie Documentaries Documentary Tyler Greco
4      s69     Movie Documentaries Documentary Hanns-Bruno Kammertöns
...
20909   s7722  TV Show Classic & Cult TV      Classic Sunny Lau
20910   s8190  TV Show Classic & Cult TV      Classic Sunny Lau
20911   s8542  TV Show Classic & Cult TV      Classic Sunny Lau
20912   s8600  TV Show Classic & Cult TV      Classic Michael Cumming
20913   s8645  TV Show Classic & Cult TV      Classic Sunny Lau
20914 rows x 5 columns

```

```
netflixWithspGenreAndDirector['director'].value_counts()
```

```

Ziad Doueiri          2331
Vijay Roche           541
Vijay S. Bhanushali  434
Takuya Igarashi       410
Vanessa Roth          336
...
Huang Jianming        1
Tony Leondis           1
Young Jun Lee          1
Roy Allen Smith        1
Eric Abrams            1
Name: director, Length: 4993, dtype: int64

```

Saved successfully!  or Cast ('#') which are present due to missing data in original netflix dataset, with cast having most movies in that genre and show type

```

netflixWithspGenreAndCast = netflixWithspGenre.merge(netflixWithspCast)
castHavingMostMoviesinAGenreAndShowType = netflixWithspGenreAndCast[~(netflixWithspGenreAndCast['cast'] == '#')].groupby(['genre','type'])
castHavingMostMoviesinAGenreAndShowType.rename({'cast': 'PossibleCast'}, axis =1, inplace = True)
netflixWithspGenreAndCast = netflixWithspGenreAndCast.merge(castHavingMostMoviesinAGenreAndShowType)
netflixWithspGenreAndCast.loc[netflixWithspGenreAndCast['cast'] == '#','cast'] = netflixWithspGenreAndCast.loc[netflixWithspGenreAndCast['cast'] == '#','PossibleCast']
netflixWithspGenreAndCast = netflixWithspGenreAndCast.loc[:,['show_id','type','genre','cast','simplified_genre']]
netflixWithspGenreAndCast

```

| | show_id | type | genre | cast | simplified_genre | 🔗 |
|--------|---------|---------|-------------------|--------------------|------------------|---|
| 0 | s1 | Movie | Documentaries | Ángel Mosqueda | Documentary | |
| 1 | s17 | Movie | Documentaries | Ángel Mosqueda | Documentary | |
| 2 | s46 | Movie | Documentaries | Ángel Mosqueda | Documentary | |
| 3 | s69 | Movie | Documentaries | Michael Schumacher | Documentary | |
| 4 | s89 | Movie | Documentaries | Malcolm X | Documentary | |
| ... | ... | ... | ... | ... | ... | |
| 149507 | s8645 | TV Show | Classic & Cult TV | Russ Tamblyn | Classic | |
| 149508 | s8645 | TV Show | Classic & Cult TV | Don S. Davis | Classic | |
| 149509 | s8645 | TV Show | Classic & Cult TV | Chris Mulkey | Classic | |
| 149510 | s8645 | TV Show | Classic & Cult TV | Gary Hershberger | Classic | |
| 149511 | s8645 | TV Show | Classic & Cult TV | Grace Zabriskie | Classic | |

149512 rows x 5 columns

```
netflixWithspGenreAndCast['cast'].value_counts()

Ángel Mosqueda      427
Şükrü Özyıldız     312
İlayda Akdoğan     208
Anupam Kher         120
Shah Rukh Khan     99
...
Marianne Williamson 1
Angie Harmon         1
Aaron Michael Drozin 1
Kendall Ryan Sanders 1
Sam Jones            1
Name: cast, Length: 36439, dtype: int64
```

Replace default value for Country ('#') which are present due to missing data in original

- netflix dataset, with country where most movies in that genre and show type and produced

```
netflixWithspGenreAndCountry = netflixWithspGenre.merge(netflixWithspCountry)
countryHavingMostMoviesinAGenreAndShowType = netflixWithspGenreAndCountry[~(netflixWithspGenreAndCountry['country'] == '#')].groupby(['genre'])
countryHavingMostMoviesinAGenreAndShowType.rename({'country': 'PossibleCountry'}, axis =1, inplace = True)
netflixWithspGenreAndCountry = netflixWithspGenreAndCountry.merge(countryHavingMostMoviesinAGenreAndShowType)
netflixWithspGenreAndCountry.loc[netflixWithspGenreAndCountry['country'] == '#','country'] = netflixWithspGenreAndCountry.loc[netflixWithspGenreAndCountry['genre'] == 'Documentaries','PossibleCountry']
netflixWithspGenreAndCountry = netflixWithspGenreAndCountry.loc[:,['show_id','type','genre','country','simplified_genre']]
netflixWithspGenreAndCountry
```

| | show_id | type | genre | country | simplified_genre | edit |
|---------------------|---------|---------|-------------------|----------------|------------------|------|
| 0 | s1 | Movie | Documentaries | United States | Documentary | |
| 1 | s17 | Movie | Documentaries | Zimbabwe | Documentary | |
| 2 | s46 | Movie | Documentaries | Zimbabwe | Documentary | |
| 3 | s69 | Movie | Documentaries | Zimbabwe | Documentary | |
| 4 | s89 | Movie | Documentaries | Zimbabwe | Documentary | |
| ... | ... | ... | ... | ... | ... | |
| 23759 | s7722 | TV Show | Classic & Cult TV | United States | Classic | |
| 23760 | s8190 | TV Show | Classic & Cult TV | United States | Classic | |
| Saved successfully! | | | | | | |
| 23762 | s8600 | TV Show | Classic & Cult TV | United Kingdom | Classic | |
| 23763 | s8645 | TV Show | Classic & Cult TV | United States | Classic | |

23764 rows × 5 columns

Splitting Duration with space, setting the first part as duration and converting the column to int

```
netflix['duration'].str.split(expand=True).add_prefix('duration')['duration1'].unique()
```

```
array(['min', 'Seasons', 'Season'], dtype=object)
```

```
netflix['duration']= netflix['duration'].str.split(expand=True).add_prefix('duration')['duration0']
netflix['duration'] = netflix['duration'].astype(str).astype(int)
netflix.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]
```

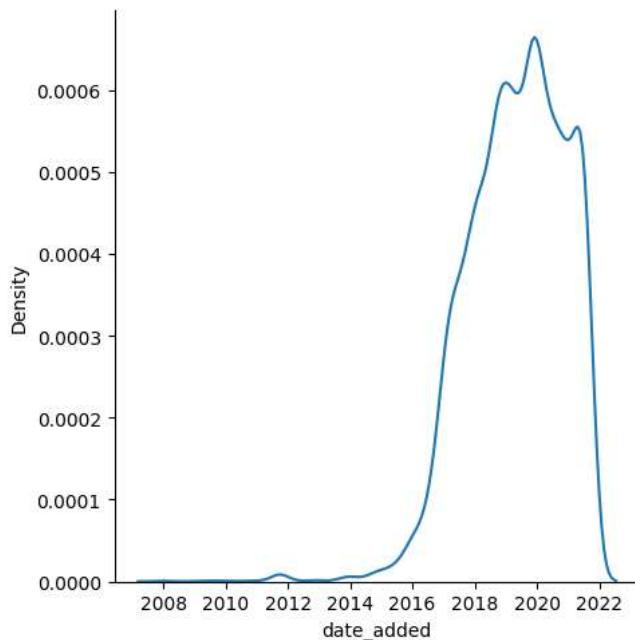
```

7   release_year 8807 non-null    int64
8   rating      8803 non-null    object
9   duration     8807 non-null    int64
10  listed_in   8807 non-null    object
11  description  8807 non-null    object
dtypes: datetime64[ns](1), int64(2), object(9)
memory usage: 825.8+ KB

```

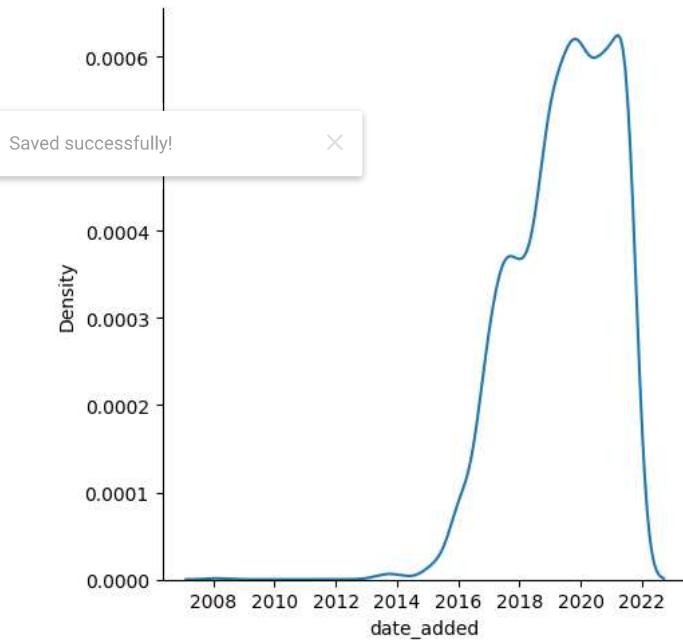
```
sns.displot(netflix[(netflix['type'] == 'Movie')]['date_added'], kind='kde')
```

```
<seaborn.axisgrid.FacetGrid at 0x7fd09819d450>
```



```
sns.displot(netflix[(netflix['type'] == 'TV Show')]['date_added'], kind='kde')
```

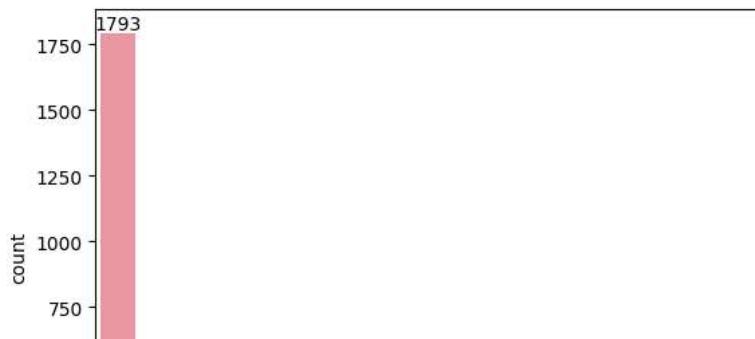
```
<seaborn.axisgrid.FacetGrid at 0x7fd08e50fa90>
```



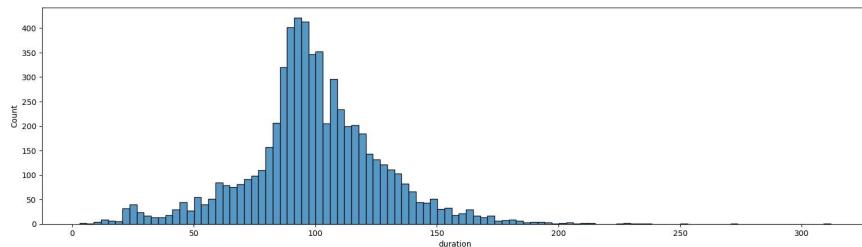
```

ax = sns.countplot(data = netflix[(netflix['type'] == 'TV Show')] , x ='duration' )
for i in ax.containers:
    ax.bar_label(i, fontsize = 10)

```

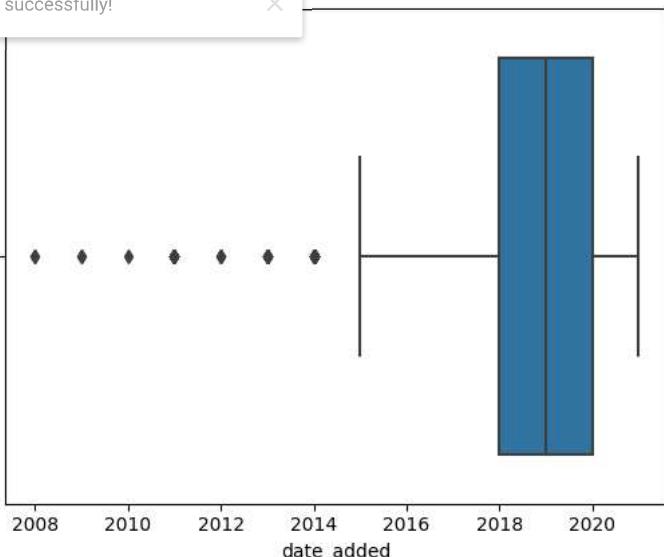


```
plt.figure(figsize=(19,5))
ax = sns.histplot(data = netflix[('type') == 'Movie'] , x ='duration' )
```



```
sns.boxplot(x= netflix["date_added"].dt.year)
plt.show()
```

Saved successfully!



```
plt.figure(figsize=(12, 6))

ax = sns.countplot(data = netflixWithspGenre,
                   x = "simplified_genre",
                   order = list(netflixWithspGenre["simplified_genre"].value_counts().index)[::-1],
                   palette= "viridis")

plt.xticks(rotation = 45, fontsize = 10)

for i in ax.containers:
```

```

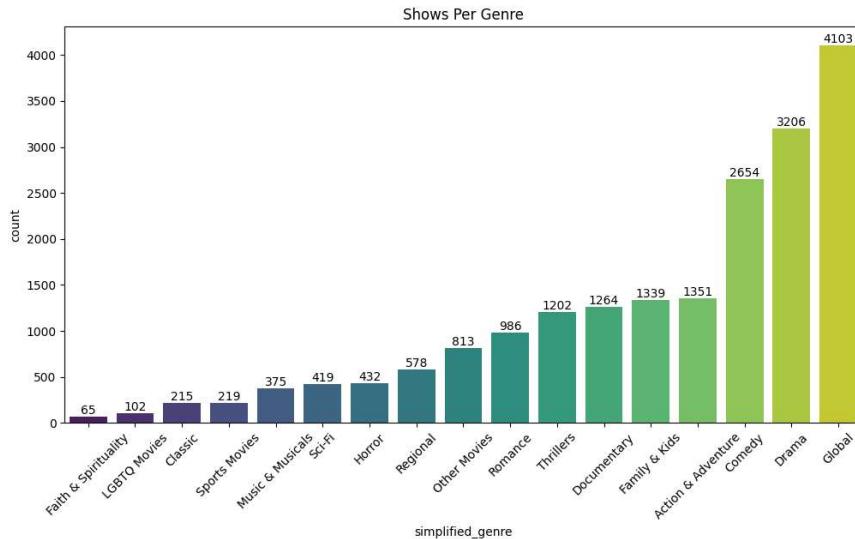
ax.bar_label(i, fontsize = 10)

plt.title("Shows Per Genre")

plt.show()

# NOTE : One show can belong to multiple genre since listed_in column
# had comma separated genre category (Which we have categorized for simplicity)

```



```

[{'date_added'].dt.year
Saved successfully!
show_id'].str.split('s',expand=True).add_prefix('show_')['show_1']
show_id_number'].astype(str).astype(int)

```

▼ 7. Business Insights

- netflixWithspGenreAndDirector
- netflixWithspGenreAndCast
- netflixWithspGenreAndCountry

are 3 variables which have been created above to drive insights from the dataset after processing comma separated values from Column director,cast,listed_in, country, stripping whitespaces, filling missing values with best possible director,cast,genre(renamed listed_in),country as per the genre and show type

```
netflixWithspGenreAndDirector.head()
```

| | show_id | type | genre | simplified_genre | director | edit |
|---|---------|-------|---------------|------------------|------------------------|------|
| 0 | s1 | Movie | Documentaries | Documentary | Kirsten Johnson | |
| 1 | s17 | Movie | Documentaries | Documentary | Pedro de Echave García | |
| 2 | s17 | Movie | Documentaries | Documentary | Pablo Azorín Williams | |
| 3 | s46 | Movie | Documentaries | Documentary | Tyler Greco | |
| 4 | s69 | Movie | Documentaries | Documentary | Hanns-Bruno Kammertöns | |

```
netflixWithspGenreAndCast.head()
```

| show_id | type | genre | cast | simplified_genre | |
|---------|------|---------------------|--------------------|------------------|--|
| 0 | s1 | Movie Documentaries | Ángel Mosqueda | Documentary | |
| 1 | s17 | Movie Documentaries | Ángel Mosqueda | Documentary | |
| 2 | s46 | Movie Documentaries | Ángel Mosqueda | Documentary | |
| 3 | s69 | Movie Documentaries | Michael Schumacher | Documentary | |

```
netflixWithspGenreAndCountry.head()
```

| show_id | type | genre | country | simplified_genre | |
|---------|------|---------------------|---------------|------------------|--|
| 0 | s1 | Movie Documentaries | United States | Documentary | |
| 1 | s17 | Movie Documentaries | Zimbabwe | Documentary | |
| 2 | s46 | Movie Documentaries | Zimbabwe | Documentary | |
| 3 | s69 | Movie Documentaries | Zimbabwe | Documentary | |
| 4 | s89 | Movie Documentaries | Zimbabwe | Documentary | |

```
# Genre in Which Movies Are Added In Last 10 Years
temp = netflix[['show_id','date_added_year']].merge(netflixWithspGenre,on = 'show_id')
temp = temp.loc[:,['show_id','date_added_year','type','simplified_genre']]
genreinWhichMostMoviesAreAddedInLast10Years = temp[(temp['type'] == 'Movie') & (temp['date_added_year'].isin(temp['date_added_year'].value_counts().index))]
genreinWhichMostMoviesAreAddedInLast10Years = genreinWhichMostMoviesAreAddedInLast10Years.groupby('simplified_genre')['show_id'].count()
Top5MovieGenresinLast10Years = genreinWhichMostMoviesAreAddedInLast10Years.head(5)['simplified_genre']
Top5MovieGenresinLast10Years
```

| | |
|---|--------------------|
| 0 | Global |
| 1 | Drama |
| 2 | Comedy |
| 3 | Documentary |
| 4 | Action & Adventure |

Name: simplified_genre, dtype: object

```
# Genre in Which TV Show Are Added In Last 10 Years
temp = netflix[['show_id','date_added_year']].merge(netflixWithspGenre,on = 'show_id')
temp = temp.loc[:,['show_id','date_added_year','type','simplified_genre']]
genreinWhichMostTVShowAreAddedInLast10Years = temp[(temp['type'] == 'TV Show') & (temp['date_added_year'].isin(temp['date_added_year'].value_counts().index))]
genreinWhichMostTVShowAreAddedInLast10Years = genreinWhichMostTVShowAreAddedInLast10Years.groupby('simplified_genre')['show_id'].count().sort_values(ascending=False).reset_index()
# Top 3 TV Show Genres in Last 10 years as per TV Show added count on netflix in last 10 years
Top5TVShowGenresinLast10Years = genreinWhichMostTVShowAreAddedInLast10Years.groupby('simplified_genre')['show_id'].count().sort_values(ascending=False).reset_index()
Top5TVShowGenresinLast10Years
```

Saved successfully!

| | |
|---|---------------|
| 2 | Comedy |
| 3 | Family & Kids |
| 4 | Thrillers |

Name: simplified_genre, dtype: object

```
mergedData = netflixWithspGenreAndDirector[['show_id','simplified_genre','director','type']].merge(netflixWithspGenreAndCast[['show_id','cast','country']],on='show_id')
mergedData.head()
```

| show_id | simplified_genre | director | type | cast | country |
|---------|------------------|-------------|------------------------|-------|---------------------------------|
| 0 | s1 | Documentary | Kirsten Johnson | Movie | Ángel Mosqueda United States |
| 1 | s17 | Documentary | Pedro de Echave García | Movie | Ángel Mosqueda Zimbabwe |
| 2 | s17 | Documentary | Pedro de Echave García | Movie | Ángel Mosqueda Zimbabwe |
| 3 | - | - | Pedro de Echave | - | - |

```
# 2.1 Who are the directors, cast worked in this genre and in which country were the movies produced
```

```
directorCastCountryAsPerTop5GenreinMovies = mergedData[(mergedData['simplified_genre'].isin(Top5MovieGenresinLast10Years)) & (mergedData['type'] == 'Movie')]
result = directorCastCountryAsPerTop5GenreinMovies.sort_values(ascending = False).reset_index().head(15)
result[['director','cast','country']]
```

→

| | director | cast | country |
|----|---------------------|------------------|-------------|
| 0 | Yılmaz Erdoğan | Yılmaz Erdoğan | Turkey |
| 1 | Thierry Donard | Wille Lindberg | France |
| 2 | Toshiya Shinohara | Kappei Yamaguchi | Japan |
| 3 | Toshiya Shinohara | Kumiko Watanabe | Japan |
| 4 | Toshiya Shinohara | Satsuki Yukino | Japan |
| 5 | Omoni Oboli | Omoni Oboli | Nigeria |
| 6 | Toshiya Shinohara | Koji Tsujitani | Japan |
| 7 | David Dhawan | Anupam Kher | India |
| 8 | Toshiya Shinohara | Houko Kuwashima | Japan |
| 9 | Cathy Garcia-Molina | Joross Gamboa | Philippines |
| 10 | Thierry Donard | Jesse Richman | France |

```
# 2.2 Who are the directors, cast worked in this genre and in which country were the TV Show produced
```

```
directorCastCountryAsPerTop5GenreinTVShow = mergedData[(mergedData['simplified_genre'].isin(Top5TVShowGenresInLast10Years)) & (mergedData['show_id'].isin(Top5ShowsInLast10Years))]
```

```
result = directorCastCountryAsPerTop5GenreinTVShow.sort_values(ascending = False).reset_index().head(20)
```

```
result[['director','cast','country']]
```

| | director | cast | country |
|----|---------------|------------------|---------------|
| 0 | Ziad Doueiri | Yuki Kaji | Japan |
| 1 | Ziad Doueiri | Takahiro Sakurai | Japan |
| 2 | Yasuhiro Irie | Takahiro Sakurai | Japan |
| 3 | Ziad Doueiri | Tay Ping Hui | Singapore |
| 4 | Ziad Doueiri | Zhang Zhenhuan | Singapore |
| 5 | Yasuhiro Irie | Yuki Kaji | Japan |
| 6 | Ziad Doueiri | Rui En | Singapore |
| 7 | Ziad Doueiri | Yuichi Nakamura | Japan |
| 8 | Ziad Doueiri | İlayda Akdoğan | United States |
| 9 | Ziad Doueiri | Jun Fukuyama | Japan |
| 10 | Ziad Doueiri | Junichi Suwabe | Japan |
| 11 | Ziad Doueiri | Mamoru Miyano | Japan |
| 12 | Yasuhiro Irie | Yuichi Nakamura | Japan |
| 13 | Ziad Doueiri | İsmail Filiz | United States |
| 14 | Yasuhiro Irie | Ai Kayano | Japan |
| 15 | Yasuhiro Irie | Jun Fukuyama | Japan |
| 16 | Yasuhiro Irie | Junichi Suwabe | Japan |
| 17 | Yasuhiro Irie | Daisuke Ono | Japan |
| 18 | Ziad Doueiri | Kenjiro Tsuda | Japan |

Saved successfully!

```
# Most Watched Movie Genres in terms of time
```

```
mostWatchedMoviesGenresInMins = netflix[netflix['type'] == 'Movie'][['show_id','duration']].merge(netflixWithspGenre, on = 'show_id')
```

```
mostWatchedMoviesGenresInMins = mostWatchedMoviesGenresInMins[['duration','simplified_genre']]
```

```
mostWatchedMoviesGenresInMins = mostWatchedMoviesGenresInMins.groupby('simplified_genre')['duration'].aggregate('sum')
```

```
Top5MostWatchedMovieGenresInLast10Years = mostWatchedMoviesGenresInMins.sort_values(ascending=False).reset_index().head(5)[['simplified_genre','duration']]
```

```
Top5MostWatchedMovieGenresInLast10Years
```

| | |
|---|--------------------|
| 0 | Global |
| 1 | Drama |
| 2 | Comedy |
| 3 | Action & Adventure |
| 4 | Other Movies |

```
Name: simplified_genre, dtype: object
```

```
# Most Watched TV Show Genres in terms of number of season (Assumption all season have equal number of episodes and each episode is of same duration)
```

```
mostWatchedTVShowGenres = netflix[netflix['type'] == 'TV Show'][['show_id','duration']].merge(netflixWithspGenre, on = 'show_id')
```

```
mostWatchedTVShowGenres = mostWatchedTVShowGenres[['duration','simplified_genre']]
```

```
mostWatchedTVShowGenres = mostWatchedTVShowGenres.groupby('simplified_genre')['duration'].aggregate('sum')
```

```
Top5MostWatchedTVShowGenreinLast10Years = mostWatchedTVShowGenres.sort_values(ascending=False).reset_index().head(5)[['simplified_genre']]
Top5MostWatchedTVShowGenreinLast10Years
```

```
0      Global
1      Drama
2      Comedy
3  Family & Kids
4      Thrillers
Name: simplified_genre, dtype: object
```

3.1 Who are the directors, cast worked in this most watched genre and in which country were the movies produced

```
directorCastCountryAsPerTop5MostWatchedMovieGenres = mergedData[(mergedData['simplified_genre'].isin(Top5MostWatchedMovieGenresinLast10Y
result = directorCastCountryAsPerTop5GenreinMovies.sort_values(ascending = False).reset_index().head(15)
result[['director','cast','country']]
```

| | director | cast | country | edit |
|----|---------------------|------------------|-------------|------|
| 0 | Yılmaz Erdoğan | Yılmaz Erdoğan | Turkey | |
| 1 | Thierry Donard | Wille Lindberg | France | |
| 2 | Toshiya Shinohara | Kappei Yamaguchi | Japan | |
| 3 | Toshiya Shinohara | Kumiko Watanabe | Japan | |
| 4 | Toshiya Shinohara | Satsuki Yukino | Japan | |
| 5 | Omoni Oboli | Omoni Oboli | Nigeria | |
| 6 | Toshiya Shinohara | Koji Tsujitani | Japan | |
| 7 | David Dhawan | Anupam Kher | India | |
| 8 | Toshiya Shinohara | Houko Kuwashima | Japan | |
| 9 | Cathy Garcia-Molina | Joross Gamboa | Philippines | |
| 10 | Thierry Donard | Jesse Richman | France | |
| 11 | Thierry Donard | Matt Annetts | France | |
| 12 | Sooraj R. Barjatya | Salman Khan | India | |
| 13 | Wilson Yip | Donnie Yen | China | |
| 14 | Wilson Yip | Donnie Yen | Hong Kong | |

3.2 Who are the directors, cast worked in this most watched genre and in which country were the movies produced

Saved successfully!

```
hedTVShowGenres = mergedData[(mergedData['simplified_genre'].isin(Top5MostWatchedTVShowGenreinLast10Y
result = directorCastCountryAsPerTop5MostWatchedTVShowGenres.sort_values(ascending = False).reset_index().head(15)
result[['director','cast','country']]
```

| | director | cast | country | edit |
|----|---------------|------------------|---------------|------|
| 0 | Ziad Doueiri | Yuki Kaji | Japan | |
| 1 | Ziad Doueiri | Takahiro Sakurai | Japan | |
| 2 | Yasuhiro Irie | Takahiro Sakurai | Japan | |
| 3 | Ziad Doueiri | Tay Ping Hui | Singapore | |
| 4 | Ziad Doueiri | Zhang Zhenhuan | Singapore | |
| 5 | Yasuhiro Irie | Yuki Kaji | Japan | |
| 6 | Ziad Doueiri | Rui En | Singapore | |
| 7 | Ziad Doueiri | Yuichi Nakamura | Japan | |
| 8 | Ziad Doueiri | İlayda Akdoğan | United States | |
| 9 | Ziad Doueiri | Jun Fukuyama | Japan | |
| 10 | Ziad Doueiri | Daisuke Ono | Japan | |
| 11 | Ziad Doueiri | Junichi Suwabe | Japan | |
| 12 | Ziad Doueiri | Mamoru Miyano | Japan | |
| 13 | Yasuhiro Irie | Yuichi Nakamura | Japan | |
| 14 | Ziad Doueiri | İsmail Filiz | United States | |

```
# Which director has made movies in Top 5 Most Watched Movie Genres in last 10 years
netflixWithspGenreAndDirector[(netflixWithspGenreAndDirector['simplified_genre'].isin(Top5MostWatchedMovieGenresinLast10Years)) & (netfli
```

| | |
|------------------------------|-----|
| Şenol Sönmez | 109 |
| Cathy Garcia-Molina | 29 |
| Yoo Byung-jae | 26 |
| Youssef Chahine | 24 |
| Wayne Orr | 24 |
| Name: director, dtype: int64 | |

```
# Which actors worked in Top 5 Most Watched Movie Genres in last 10 years
```

```
netflixWithspGenreAndCast[(netflixWithspGenreAndCast['simplified_genre'].isin(Top5MostWatchedMovieGenresinLast10Years)) & (netflixWithspC
```

| | |
|--------------------------|-----|
| Şükrü Özyıldız | 191 |
| Anupam Kher | 100 |
| Shah Rukh Khan | 86 |
| Naseeruddin Shah | 81 |
| Paresh Rawal | 68 |
| Name: cast, dtype: int64 | |

```
# In Which countries movies are produced for Top 5 Most Watched Movie Genres in last 10 years
```

```
netflixWithspGenreAndCountry[(netflixWithspGenreAndCountry['simplified_genre'].isin(Top5MostWatchedMovieGenresinLast10Years)) & (netflixW
```

| | |
|-----------------------------|------|
| United States | 2768 |
| India | 2159 |
| United Kingdom | 641 |
| France | 541 |
| Canada | 331 |
| Name: country, dtype: int64 | |

```
# Which director has made TV Shows in Top 5 Most Watched Movie Genres in last 10 years
```

```
netflixWithspGenreAndDirector[(netflixWithspGenreAndDirector['simplified_genre'].isin(Top5MostWatchedTVShowGenreinLast10Years)) & (netfli
```

| | |
|---|------|
| Ziad Doueiri | 2329 |
| Vijay Roche | 541 |
| Vijay S. Bhanushali | 434 |
| Yasuhiro Irie | 167 |
| Vasan Bala | 89 |
| ... | |
| Jared Hess | 1 |
| Joshua Zeman | 1 |
| Adrián García Bogliano | 1 |
| John Dower | 1 |
| Eric Abrams | 1 |
| Name: director, Length: 267, dtype: int64 | |

Saved successfully!

X Watched Movie Genres in last 10 years
thspGenreAndCast['simplified_genre'].isin(Top5MostWatchedTVShowGenreinLast10Years)) & (netflixWithspC

| | |
|---|-----|
| Şükrü Özyıldız | 121 |
| Ismail Filiz | 78 |
| Takahiro Sakurai | 50 |
| Zyon Allen | 44 |
| Yuki Kaji | 41 |
| ... | |
| Aldis Hodge | 1 |
| Jonny Weston | 1 |
| Joseph Cross | 1 |
| Derek Mio | 1 |
| Sam Jones | 1 |
| Name: cast, Length: 14313, dtype: int64 | |

```
# In Which country TV Shows are produced for Top 5 Most Watched Movie Genres in last 10 years
```

```
netflixWithspGenreAndCountry[(netflixWithspGenreAndCountry['simplified_genre'].isin(Top5MostWatchedTVShowGenreinLast10Years)) & (netflixW
```

| | |
|---|------|
| United States | 1234 |
| West Germany | 406 |
| Japan | 382 |
| United Kingdom | 304 |
| South Korea | 257 |
| ... | |
| Azerbaijan | 2 |
| Malta | 2 |
| Uruguay | 1 |
| Hungary | 1 |
| Greece | 1 |
| Name: country, Length: 66, dtype: int64 | |

```
# Top 3 Most Watched Movie Genre in Last 10 years
```

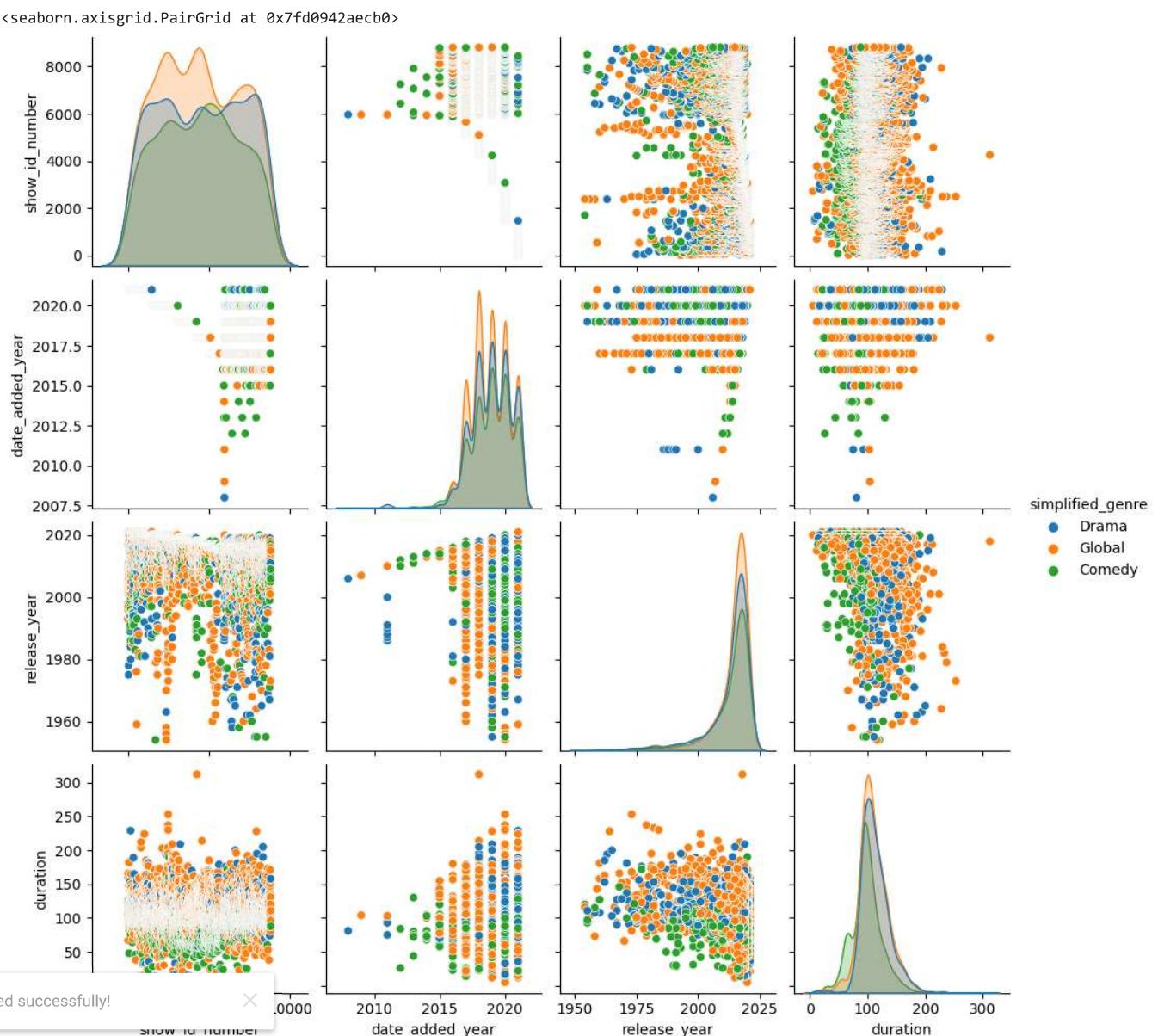
```
Top3MovieGenresinLast10Years = Top5MovieGenresinLast10Years.head(3)
```

```
# 'show_id_number' is generated from show_id after removing first char 's'
```

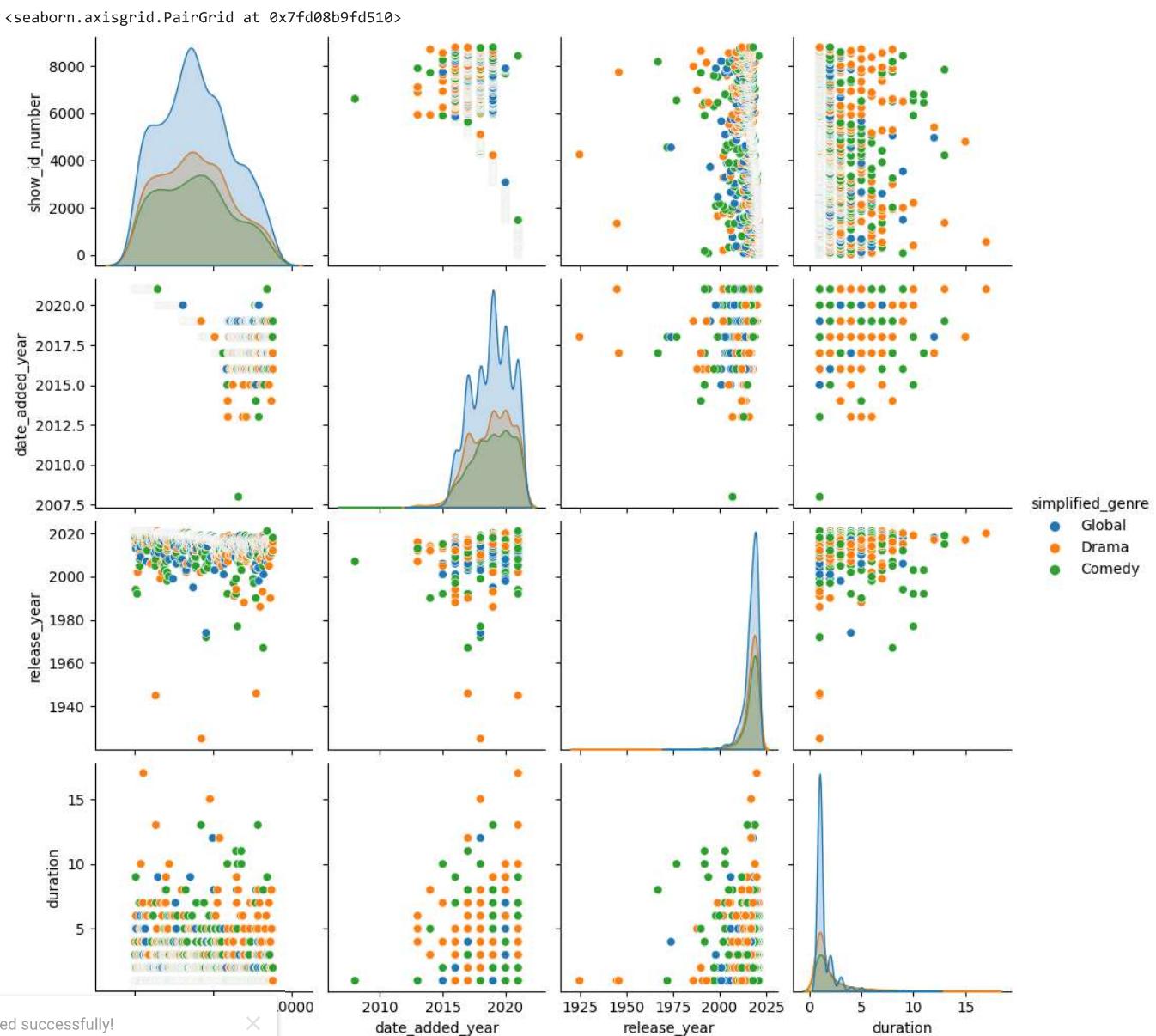
```
DSTop3MovieGenresinLast10Years = netflix[['show_id', 'show_id_number', 'date_added_year', 'release_year', 'duration']].merge(netflixWithspGe
```

```
DSTop3MovieGenresinLast10Years = DSTop3MovieGenresinLast10Years[['show_id_number', 'date_added_year', 'release_year', 'duration', 'simplified
```

```
sns.pairplot(DSTop3TVShowGenresinLast10Years[(DSTop3TVShowGenresinLast10Years['type'] == 'Movie') & (DSTop3TVShowGenresinLast10Years['simplified_genre'].notna())], x_vars=['show_id_number', 'date_added_year', 'release_year', 'duration'], y_vars=['show_id_number', 'date_added_year', 'release_year', 'duration'], hue='simplified_genre')
```



```
# Top 3 Most Watched TV Show Genre in Last 10 years
Top3MostWatchedTVShowGenreinLast10Years = Top5MostWatchedTVShowGenreinLast10Years.head(3)
# 'show_id_number' is generated from show_id after removing first char 's'
DSTop3TVShowGenresinLast10Years = netflix[['show_id', 'show_id_number', 'date_added_year', 'release_year', 'duration']].merge(netflixWithspc)
DSTop3TVShowGenresinLast10Years = DSTop3TVShowGenresinLast10Years[['show_id_number', 'date_added_year', 'release_year', 'duration', 'simplified_genre']]
sns.pairplot(DSTop3TVShowGenresinLast10Years[(DSTop3TVShowGenresinLast10Years['type'] == 'TV Show') & (DSTop3TVShowGenresinLast10Years['simplified_genre'].notna())], x_vars=['show_id_number', 'date_added_year', 'release_year', 'duration'], y_vars=['show_id_number', 'date_added_year', 'release_year', 'duration'], hue='simplified_genre')
```



✓ 11s completed at 2:37PM