```
#importing libraries for our purpose
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv -O /netflix.csv
```

```
--2023-08-07 13:50:44--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 13.224.9.129, 13.224.9.24, 13.224.9.103, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|13.224.9.129|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3399671 (3.2M) [text/plain]
Saving to: ‘/netflix.csv’

/netflix.csv        100%[===================>]   3.24M  --.-KB/s    in 0.06s

2023-08-07 13:50:44 (56.2 MB/s) - ‘/netflix.csv’ saved [3399671/3399671]
```

```
df=pd.read_csv("/netflix.csv")
```

Analysing basic metrics 2.Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary

```
df.head(10)
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	NaN	September 24, 2021	2021	TV-MA	1 Season	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...
6	s7	Movie	My Little Pony: A New	Robert Cullen, José Luis	Vanessa Hudgens, Kimiko Glenn,	NaN	September 24, 2021	2021	PG	91 min	Children & Family Movies	Equestria's divided. But a bright-eyed

```
df.shape
```

```
(8807, 12)
```

```
df.columns
```

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
```

```
            dtype='object')
```

```
len(df)
```

```
8807
```

```
df.dtypes
```

```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

```
df.info
```

```
8803   s8804  TV Show            Zombie Dumb              NaN
8804   s8805    Movie             Zombieland  Ruben Fleischer
8805   s8806    Movie                   Zoom     Peter Hewitt
8806   s8807    Movie                 Zubaan      Mozez Singh

                                                   cast        country  \
0                                                   NaN  United States
1     Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...   South Africa
2     Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...            NaN
3                                                   NaN            NaN
4     Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...          India
...                                                 ...            ...
8802  Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...  United States
8803                                                NaN            NaN
8804  Jesse Eisenberg, Woody Harrelson, Emma Stone, ...  United States
8805  Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...  United States
8806  Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...          India

              date_added  release_year rating   duration  \
0     September 25, 2021          2020  PG-13     90 min
1     September 24, 2021          2021  TV-MA  2 Seasons
2     September 24, 2021          2021  TV-MA   1 Season
3     September 24, 2021          2021  TV-MA   1 Season
4     September 24, 2021          2021  TV-MA  2 Seasons
...                  ...           ...    ...        ...
8802   November 20, 2019          2007      R    158 min
8803        July 1, 2019          2018  TV-Y7  2 Seasons
8804    November 1, 2019          2009      R     88 min
8805    January 11, 2020          2006     PG     88 min
8806       March 2, 2019          2015  TV-14    111 min

                                          listed_in  \
0                                    Documentaries
1       International TV Shows, TV Dramas, TV Mysteries
2     Crime TV Shows, International TV Shows, TV Act...
3                               Docuseries, Reality TV
4     International TV Shows, Romantic TV Shows, TV ...
...                                               ...
8802                   Cult Movies, Dramas, Thrillers
8803            Kids' TV, Korean TV Shows, TV Comedies
8804                           Comedies, Horror Movies
8805               Children & Family Movies, Comedies
8806    Dramas, International Movies, Music & Musicals

                                        description
0     As her father nears the end of his life, filmm...
1     After crossing paths at a party, a Cape Town t...
2     To protect his family from a powerful drug lor...
3     Feuds, flirtations and toilet talk go down amo...
```

```
[8807 rows x 12 columns]>
```

Double-click (or enter) to edit

```
df.nunique()
```

```
show_id         8807
type               2
title           8807
director        4528
cast            7692
country          748
date_added      1767
release_year      74
rating            17
duration         220
listed_in        514
description     8775
dtype: int64
```

```
df.isnull().sum()/len(df)*100
```

```
show_id          0.000000
type             0.000000
title            0.000000
director        29.908028
cast             9.367549
country          9.435676
date_added       0.113546
release_year     0.000000
rating           0.045418
duration         0.034064
listed_in        0.000000
description      0.000000
dtype: float64
```

```
df.duplicated().sum()
```

```
0
```

```
df1 = df.copy()
```

```
df1.shape
```

```
(8807, 12)
```

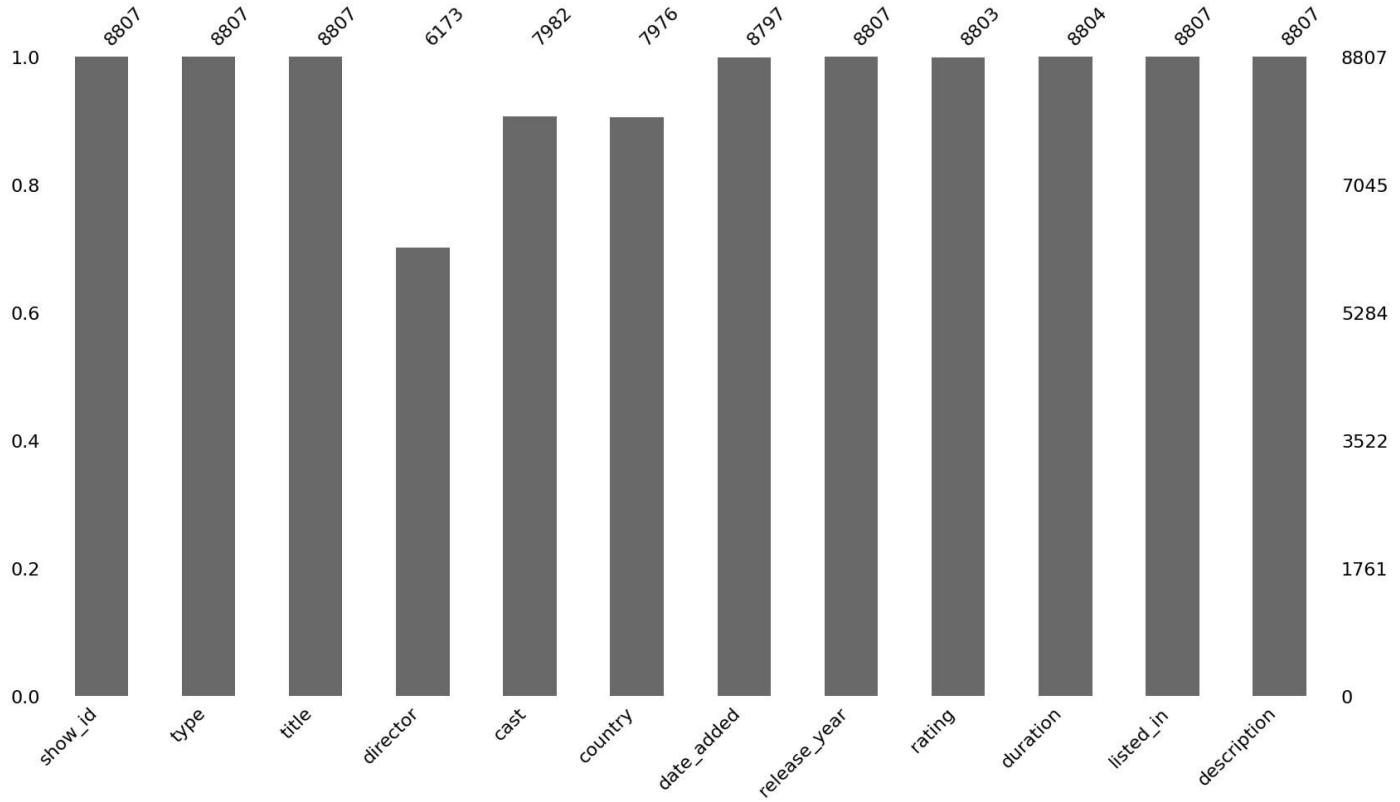
```
df1=df1.dropna()
df1.shape
```

```
(5332, 12)
```

```
df1.head(10)
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA	125 min	Dramas, Independent Movies, International Movies	On a photo shoot in Ghana, an American model s...
8	s9	TV Show	The Great British Baking Show	Andy Devonshire	Mel Giedroyc, Sue Perkins, Mary Berry, Paul Ho...	United Kingdom	September 24, 2021	2021	TV-14	9 Seasons	British TV Shows, Reality TV	A talented batch of amateur bakers face off in...
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline	United States	September 24, 2021	2021	PG-13	104 min	Comedies, Dramas	A woman adjusting to life after a loss contend

```
import missingno as msno
msno.bar(df, figsize=(20,10))
plt.show()
```




```
pip install missingno
```

```
Requirement already satisfied: missingno in /usr/local/lib/python3.10/dist-packages (0.5.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from missingno) (1.22.4)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from missingno) (3.7.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from missingno) (1.10.1)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (from missingno) (0.12.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->missingno) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->missingno) (0.11.0)
```

```
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->missingno) (4.41.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->missingno) (1.4.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->missingno) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->missingno) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->missingno) (3.1.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->missingno) (2.8.2)
Requirement already satisfied: pandas>=0.25 in /usr/local/lib/python3.10/dist-packages (from seaborn->missingno) (1.5.3)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.25->seaborn->missingno) (2022.7.1
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->missingno) (1
```

```
df.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
```

```
df.describe(include=[np.object])
```

```
<ipython-input-23-554b7518cb2b>:1: DeprecationWarning: `np.object` is a deprecated alias for the builtin `object`. To silence this warni
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  df.describe(include=[np.object])
```

	show_id	type	title	director	cast	country	date_added	rating	duration	listed_in	description
count	8807	8807	8807	6173	7982	7976	8797	8803	8804	8807	8807
unique	8807	2	8807	4528	7692	748	1767	17	220	514	8775
top	s1	Movie	Dick Johnson Is Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	TV-MA	1 Season	Dramas, International Movies	Paranormal activity at a lush, abandoned prope...
freq	1	6131	1	19	19	2818	109	3207	1793	362	4

```
df.describe(include=[np.number])
```

	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

```
import pandas as pd

# Assuming 'df' is your DataFrame

df["date_added"] = pd.to_datetime(df['date_added'])
df['day_added'] = df['date_added'].dt.day
df['year_added'] = df['date_added'].dt.year
df['month_added'] = df['date_added'].dt.month

df['year_added'] = df['year_added'].astype(int)
df['day_added'] = df['day_added'].astype(int)

# Now 'df' contains the modified columns


df['date_added'] = pd.to_datetime(df['date_added'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5332 entries, 7 to 8806
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype         
---  ------        --------------  -----         
 0   show_id       5332 non-null   object        
 1   type          5332 non-null   object        
 2   title         5332 non-null   object        
 3   director      5332 non-null   object        
 4   cast          5332 non-null   object        
 5   country       5332 non-null   object        
 6   date_added    5332 non-null   datetime64[ns]
 7   release_year  5332 non-null   int64         
 8   rating        5332 non-null   object        
 9   duration      5332 non-null   object        
 10  listed_in     5332 non-null   object        
 11  description   5332 non-null   object        
 12  day_added     5332 non-null   int64         
 13  year_added    5332 non-null   int64         
 14  month_added   5332 non-null   int64         
dtypes: datetime64[ns](1), int64(4), object(10)
memory usage: 666.5+ KB
```

3. Non-Graphical Analysis: Value counts and unique attributes:

```
for i in df.columns:
 print("The Unique values and nuniques of ",str(i),"are",df[i].unique(),df[i].nunique())
 print("..................")
```

```
 'Comedies, Horror Movies, Sci-Fi & Fantasy'
 'Action & Adventure, Comedies, Horror Movies'
 'Classic & Cult TV, Crime TV Shows, TV Dramas'
 'Action & Adventure, Documentaries, Sports Movies'
 'International Movies, LGBTQ Movies, Romantic Movies'
 'Cult Movies, Dramas, Thrillers'] 514
..................
The Unique values and nuniques of  description are ['As her father nears the end of his life, filmmaker Kirsten Johnson stages his de
 'After crossing paths at a party, a Cape Town teen sets out to prove whether a private-school swimming star is her sister who was ab
 'To protect his family from a powerful drug lord, skilled thief Mehdi and his expert team of robbers are pulled into a violent and d
 ...
 'Looking to survive in a world taken over by zombies, a dorky college student teams with an urban roughneck and a pair of grifter si
 'Dragged from civilian life, a former superhero must train a new crop of youthful saviors when the military preps for an attack by a
 "A scrappy but poor boy worms his way into a tycoon's dysfunctional family, while facing his fear of music and the truth about his p
..................
The Unique values and nuniques of  day_added are [25. 24. 23. 22. 21. 20. 19. 17. 16. 15. 14. 11. 10.  9.  8.  7.  6.  5.
  4.  3.  2.  1. 31. 29. 28. 27. 26. 18. 13. 12. 30. nan] 31
..................
The Unique values and nuniques of  year_added are [2021. 2020. 2019. 2018. 2017. 2016. 2015. 2014. 2013. 2012. 2011. 2009.
 2008.   nan 2010.] 14
..................
The Unique values and nuniques of  month_added are [ 9.  8.  7.  6.  5.  4.  3.  2.  1. 12. 11. 10. nan] 12
..................
```

```
df.columns
```

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description',
       'day_added', 'year_added', 'month_added'],
      dtype='object')
```

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

```
Movie      6131
TV Show    2676
Name: type, dtype: int64
```

```
df['show_id'].value_counts()
```

```
s1       1
s5875    1
s5869    1
s5870    1
s5871    1
        ..
s2931    1
s2930    1
s2929    1
s2928    1
s8807    1
Name: show_id, Length: 8807, dtype: int64
```

```
df['title'].value_counts()
```

```
Dick Johnson Is Dead                     1
Ip Man 2                                 1
Hannibal Buress: Comedy Camisado         1
Turbo FAST                               1
Masha's Tales                            1
                                       ..
Love for Sale 2                          1
ROAD TO ROMA                             1
Good Time                                1
Captain Underpants Epic Choice-o-Rama    1
Zubaan                                   1
Name: title, Length: 8807, dtype: int64
```

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

```
Rajiv Chilaka                    19
Raúl Campos, Jan Suter           18
Marcus Raboy                     16
Suhas Kadav                      16
Jay Karas                        14
                                 ..
Raymie Muzquiz, Stu Livingston    1
Joe Menendez                      1
Eric Bross                        1
Will Eisenberg                    1
```

```
Mozez Singh                         1
Name: director, Length: 4528, dtype: int64
```

```
df['country'].value_counts()
```

```
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
```

```
df['date_added'].value_counts()
```

```
2020-01-01    110
2019-11-01     91
2018-03-01     75
2019-12-31     74
2018-10-01     71
              ...
2017-02-21      1
2017-02-07      1
2017-01-29      1
2017-01-25      1
2020-01-11      1
Name: date_added, Length: 1714, dtype: int64
```

```
df['release_year'].value_counts()
```

```
2018    1147
2017    1032
2019    1030
2020     953
2016     902
        ...
1959       1
1925       1
1961       1
1947       1
1966       1
Name: release_year, Length: 74, dtype: int64
```

```
df['rating'].value_counts()
```

```
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
74 min         1
84 min         1
66 min         1
Name: rating, dtype: int64
```

```
df['duration'].value_counts()
```

```
1 Season     1793
2 Seasons     425
3 Seasons     199
90 min        152
94 min        146
              ...
16 min          1
186 min         1
```

```
193 min         1
189 min         1
191 min         1
Name: duration, Length: 220, dtype: int64
```

```
df['listed_in'].value_counts()
```

```
Dramas, International Movies                          362
Documentaries                                         359
Stand-Up Comedy                                       334
Comedies, Dramas, International Movies                274
Dramas, Independent Movies, International Movies      252
                                                     ...
Kids' TV, TV Action & Adventure, TV Dramas              1
TV Comedies, TV Dramas, TV Horror                       1
Children & Family Movies, Comedies, LGBTQ Movies        1
Kids' TV, Spanish-Language TV Shows, Teen TV Shows      1
Cult Movies, Dramas, Thrillers                          1
Name: listed_in, Length: 514, dtype: int64
```

```
df['description'].value_counts()
```

```
Paranormal activity at a lush, abandoned property alarms a group eager to redevelop the site, but the eerie events may not be as
unearthly as they think.    4
Challenged to compose 100 songs before he can marry the girl he loves, a tortured but passionate singer-songwriter embarks on a
poignant musical journey.    3
A surly septuagenarian gets another chance at her 20s after having her photo snapped at a studio that magically takes 50 years off her
life.                 3
Multiple women report their husbands as missing but when it appears they are looking for the same man, a police officer traces their
cryptic connection.     3
Secrets bubble to the surface after a sensual encounter and an unforeseen crime entangle two friends and a woman caught between them.
2

..
Sent away to evade an arranged marriage, a 14-year-old begins a harrowing journey of sex work and poverty in the slums of Accra.
1
When his partner in crime goes missing, a small-time crook’s life is transformed as he dedicates himself to raising the daughter his
friend left behind.     1
During 1962’s Cuban missile crisis, a troubled math genius finds himself drafted to play in a U.S.-Soviet chess match – and a deadly
game of espionage.      1
A teen's discovery of a vintage Polaroid camera develops into a darker tale when she finds that whoever takes their photo with it dies
soon afterward.       1
A scrappy but poor boy worms his way into a tycoon's dysfunctional family, while facing his fear of music and the truth about his past.
1
Name: description, Length: 8775, dtype: int64
```

4. Visual Analysis - Univariate, Bivariate after pre-processing of the data Unnesting data

**Unnesting data **

```
df['type'].value_counts().reset_index()
```

	index	type
0	Movie	6131
1	TV Show	2676

```
df['type'].value_counts().reset_index()
```

	index	type
0	Movie	6131
1	TV Show	2676

```
df['show_id'].value_counts().reset_index()
```

	index	show_id
0	s1	1
1	s5875	1
2	s5869	1
3	s5870	1
4	s5871	1
...	...	...
8802	s2931	1
8803	s2930	1
8804	s2929	1
8805	s2928	1

```
df['title'].value_counts().reset_index()
```

	index	title
0	Dick Johnson Is Dead	1
1	Ip Man 2	1
2	Hannibal Buress: Comedy Camisado	1
3	Turbo FAST	1
4	Masha's Tales	1
...	...	...
8802	Love for Sale 2	1
8803	ROAD TO ROMA	1
8804	Good Time	1
8805	Captain Underpants Epic Choice-o-Rama	1
8806	Zubaan	1

8807 rows × 2 columns

```
df['director'].value_counts().reset_index()
```

	index	director
0	Rajiv Chilaka	19
1	Raúl Campos, Jan Suter	18
2	Marcus Raboy	16
3	Suhas Kadav	16
4	Jay Karas	14
...	...	...
4523	Raymie Muzquiz, Stu Livingston	1
4524	Joe Menendez	1
4525	Eric Bross	1
4526	Will Eisenberg	1
4527	Mozez Singh	1

4528 rows × 2 columns

```
df['country'].value_counts().reset_index()
```

	index	country
0	United States	2818
1	India	972
2	United Kingdom	419
3	Japan	245
4	South Korea	199
...	...	...
743	Romania, Bulgaria, Hungary	1
744	Uruguay, Guatemala	1

```
df['date_added'].value_counts().reset_index()
```

	index	date_added
0	2020-01-01	110
1	2019-11-01	91
2	2018-03-01	75
3	2019-12-31	74
4	2018-10-01	71
...	...	...
1709	2017-02-21	1
1710	2017-02-07	1
1711	2017-01-29	1
1712	2017-01-25	1
1713	2020-01-11	1

1714 rows × 2 columns

```
df['release_year'].value_counts().reset_index()
```

	index	release_year
0	2018	1147
1	2017	1032
2	2019	1030
3	2020	953
4	2016	902
...	...	...
69	1959	1
70	1925	1
71	1961	1
72	1947	1
73	1966	1

74 rows × 2 columns

```
df['rating'].value_counts().reset_index()
```

	index	rating
0	TV-MA	3207
1	TV-14	2160
2	TV-PG	863
3	R	799
4	PG-13	490
5	TV-Y7	334
6	TV-Y	307
7	PG	287
8	TV-G	220
9	NR	80
10	G	41
11	TV-Y7-FV	6

```
df['duration'].value_counts().reset_index()
```

	index	duration
0	1 Season	1793
1	2 Seasons	425
2	3 Seasons	199
3	90 min	152
4	94 min	146
...	...	...
215	16 min	1
216	186 min	1
217	193 min	1
218	189 min	1
219	191 min	1

220 rows × 2 columns

```
df['listed_in'].value_counts().reset_index()
```

	index	listed_in
0	Dramas, International Movies	362
1	Documentaries	359
2	Stand-Up Comedy	334
3	Comedies, Dramas, International Movies	274
4	Dramas, Independent Movies, International Movies	252
...	...	...
509	Kids' TV, TV Action & Adventure, TV Dramas	1
510	TV Comedies, TV Dramas, TV Horror	1
511	Children & Family Movies, Comedies, LGBTQ Movies	1
512	Kids' TV, Spanish-Language TV Shows, Teen TV S...	1
513	Cult Movies, Dramas, Thrillers	1

514 rows × 2 columns

```
df['description'].value_counts().reset_index()
```

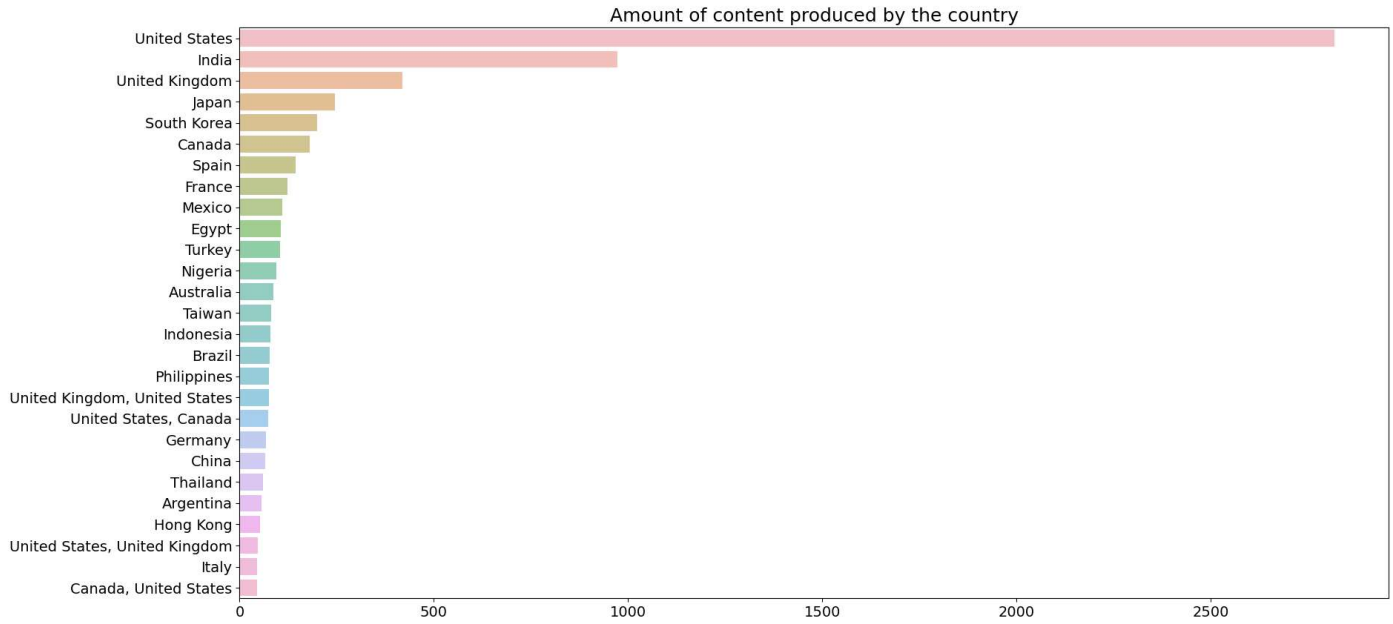
	index	description
0	Paranormal activity at a lush, abandoned prope...	4
1	Challenged to compose 100 songs before he can ...	3
2	A surly septuagenarian gets another chance at ...	3
3	Multiple women report their husbands as missin...	3
4	Secrets bubble to the surface after a sensual ...	2
...	...	...
8770	Sent away to evade an arranged marriage, a 14-...	1
8771	When his partner in crime goes missing, a smal...	1
8772	During 1962's Cuban missile crisis, a troubled...	1
8773	A teen's discovery of a vintage Polaroid camer...	1

```
countries = df['country'].value_counts()[df['country'].value_counts(normalize=True)> 0.005]
list_countries = list(countries.index)
```

```
countries = df['country'].value_counts()[df['country'].value_counts(normalize=True)> 0.005]
list_countries = list(countries.index)
```

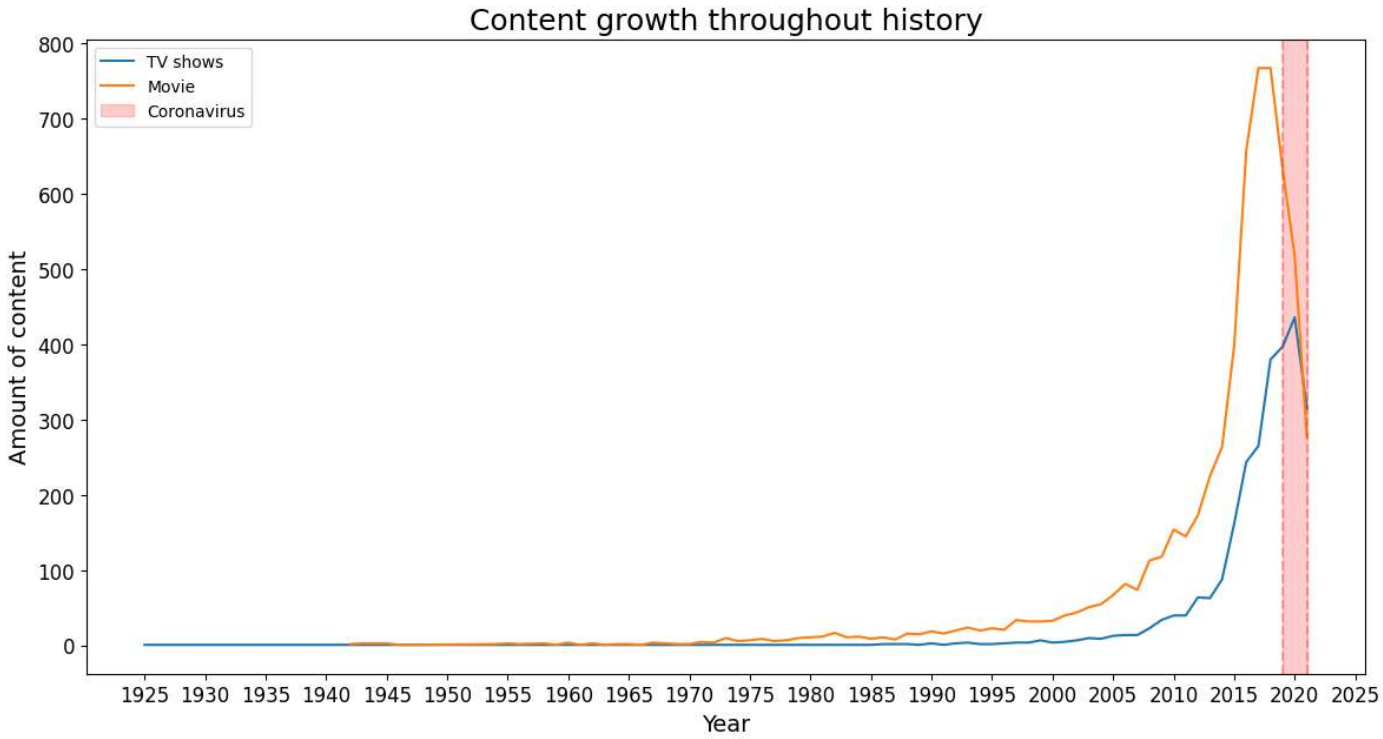
barplotting the number of content per each country:

```
plt.figure(figsize=(20,10))
plt.title('Amount of content produced by the country', fontsize=18)
plt.tick_params(labelsize=14)
sns.barplot(y=countries.index, x=countries.values, alpha=0.6)
plt.show()
```




```
TVshows = df[df['type'] == 'TV Show']
Movie = df[df['type'] == 'Movie']
```

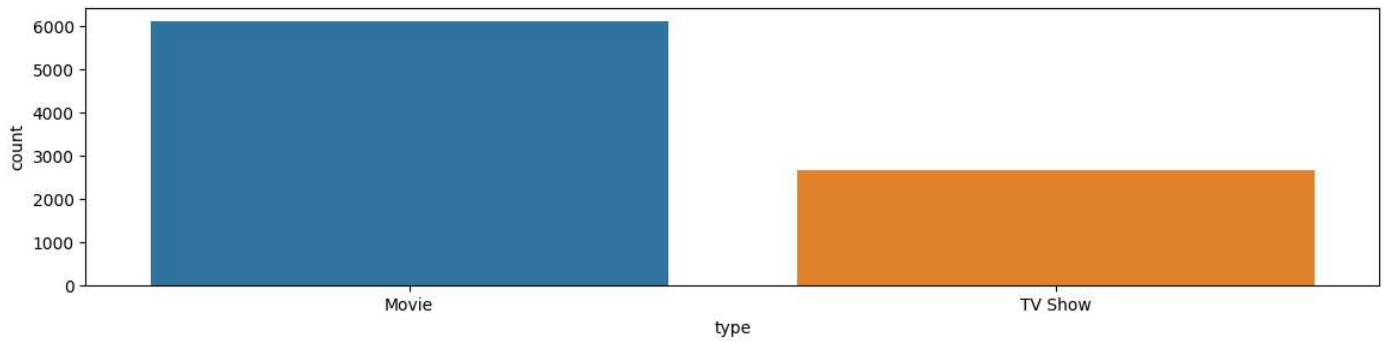
```
TVshows_progress = TVshows['release_year'].value_counts().sort_index()
Movie_progress = Movie['release_year'].value_counts().sort_index()
plt.figure(figsize=(14, 7))
plt.plot(TVshows_progress.index, TVshows_progress.values, label='TV shows')
plt.plot(Movie_progress.index, Movie_progress.values, label='Movie')
plt.axvline(2019, alpha=0.3, linestyle='--', color='r')
plt.axvline(2021, alpha=0.3, linestyle='--', color='r')
plt.axvspan(2019, 2021, alpha=0.2, color='r', label='Coronavirus')
plt.xticks(list(range(1925, 2026, 5)), fontsize=12)
plt.title('Content growth throughout history', fontsize=18)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Amount of content', fontsize=14)
plt.yticks(fontsize=12)
plt.legend()
plt.show()
```



Countplot:

```
plt.figure(figsize=(14, 3))
sns.countplot(x='type',data = df)
```

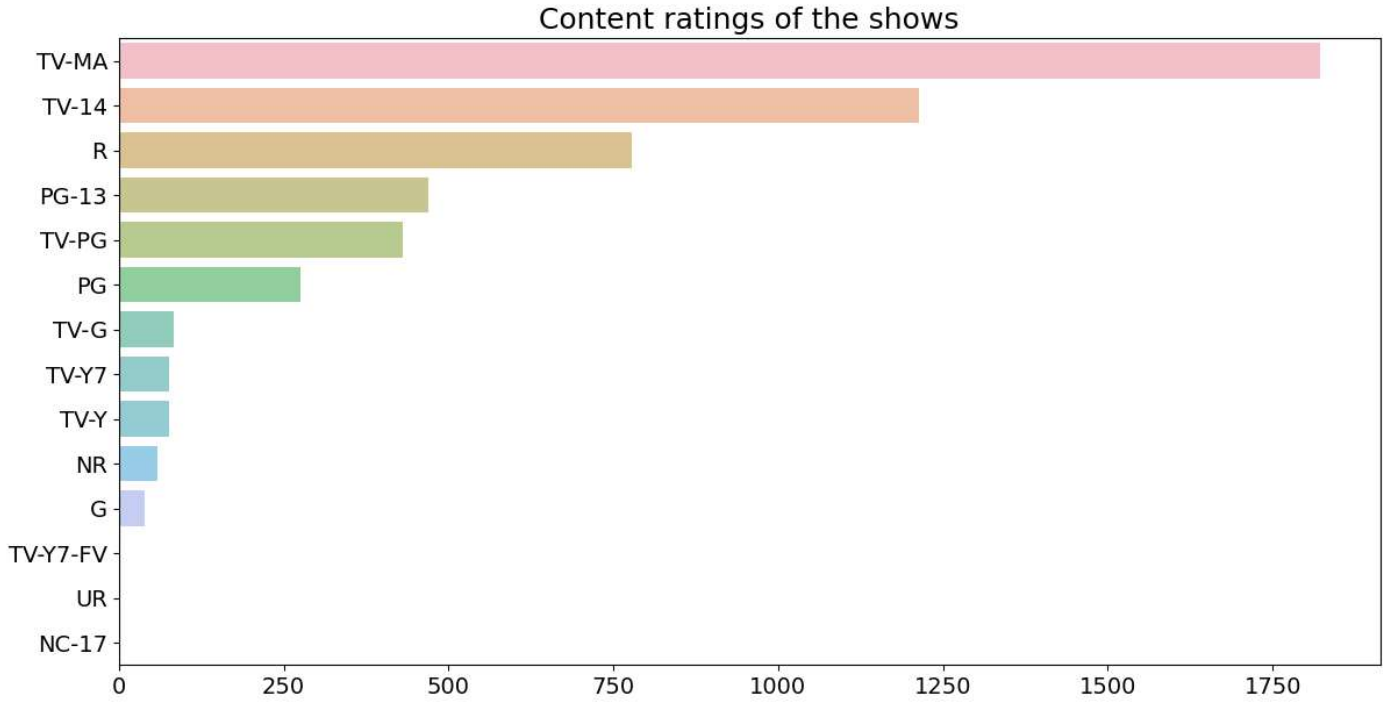
```
<Axes: xlabel='type', ylabel='count'>
```



Barplot:

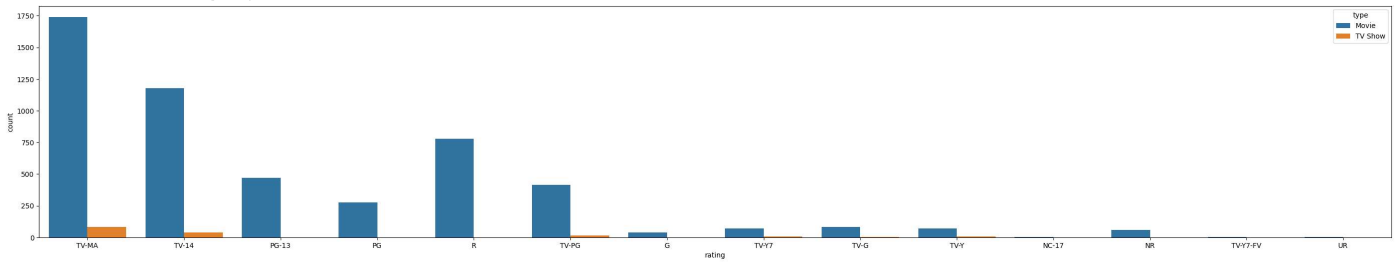
```
df.dropna(inplace=True)
rating = df['rating'].value_counts()
plt.figure(figsize=(14,7))
plt.title('Content ratings of the shows', fontsize=18)
plt.tick_params(labelsize=14)
sns.barplot(y=rating.index, x=rating.values, alpha=0.6)
```

```
<Axes: title={'center': 'Content ratings of the shows'}>
```

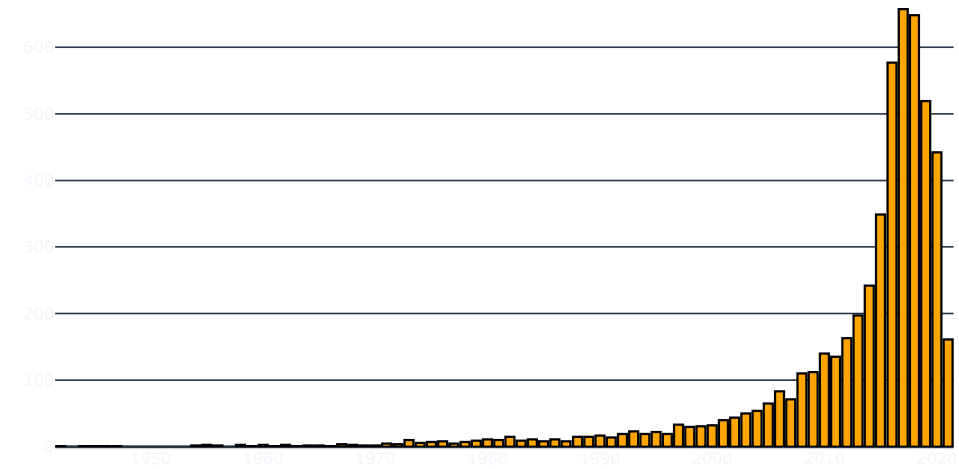



```
plt.figure(figsize = (35,6))
sns.countplot(x='rating',data = df,hue='type')
```

```
<Axes: xlabel='rating', ylabel='count'>
```




```
temp_df1 = df['release_year'].value_counts().reset_index()
import plotly.graph_objects as go
trace1 = go.Bar(
 x = temp_df1['index'],
 y = temp_df1['release_year'],
 marker = dict(color = 'rgb(255,165,0)',
 line=dict(color='rgb(0,0,0)',width=1.5)))
layout = go.Layout(template= "plotly_dark",title = 'CONTENT RELEASE OVER THE YEAR')
fig = go.Figure(data = [trace1], layout = layout)
fig.show()
```

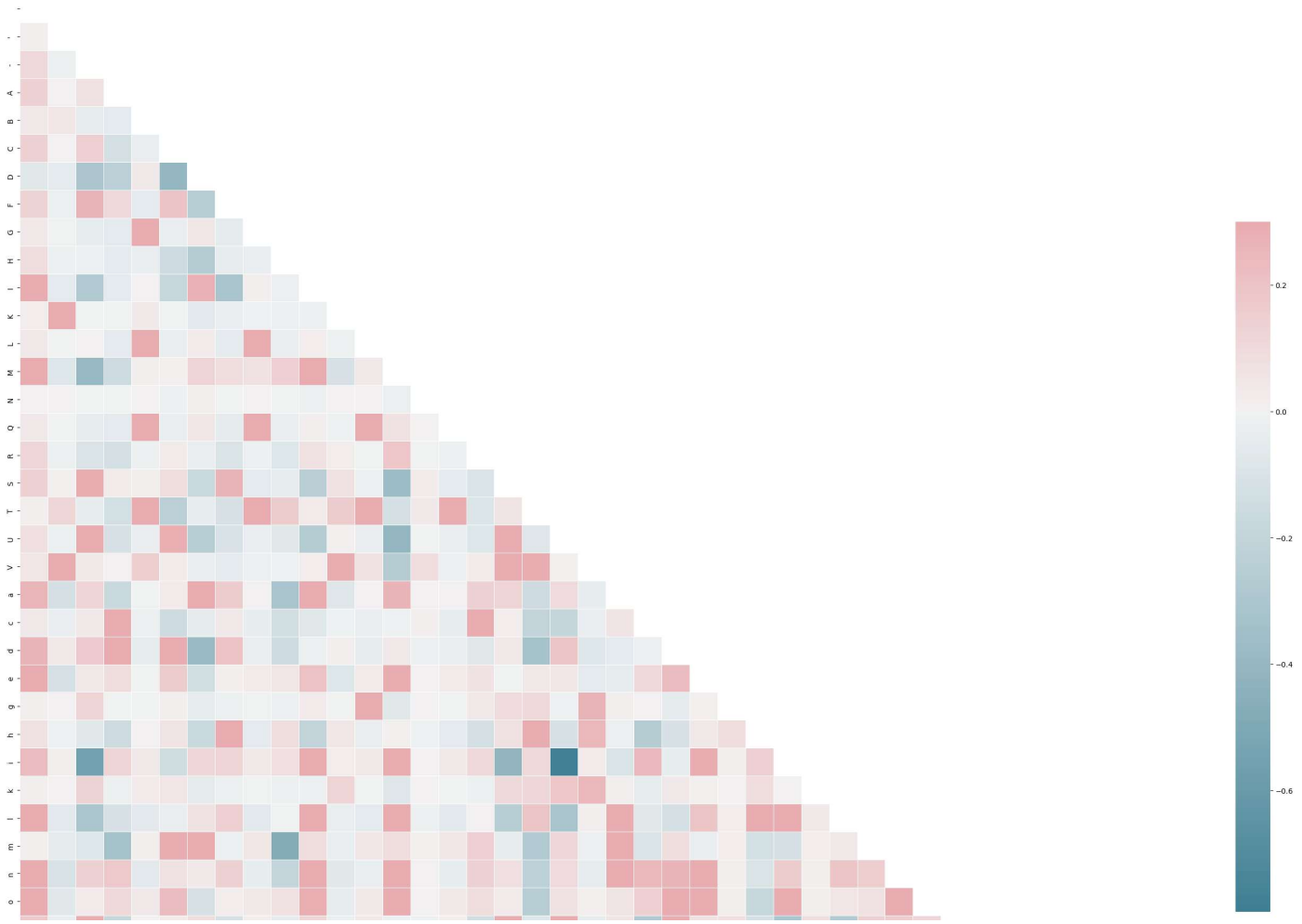


bold('**HEATMAP(Correlation)**') **bold text**

```
# bold('**HEATMAP(Correlation)**')
from sklearn.preprocessing import MultiLabelBinarizer # Similar to One-Hot Encoding
data= df['listed_in'].astype(str).apply(lambda s : s.replace('&',' ').replace(',', ' '))
test = data
mlb = MultiLabelBinarizer()
res = pd.DataFrame(mlb.fit_transform(test), columns=mlb.classes_)
corr = res.corr()
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(35, 34))
cmap = sns.diverging_palette(220, 10, as_cmap=True)
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,square=True, linewidths=.5, cbar_kws={"shrink": .5})
plt.show()
```

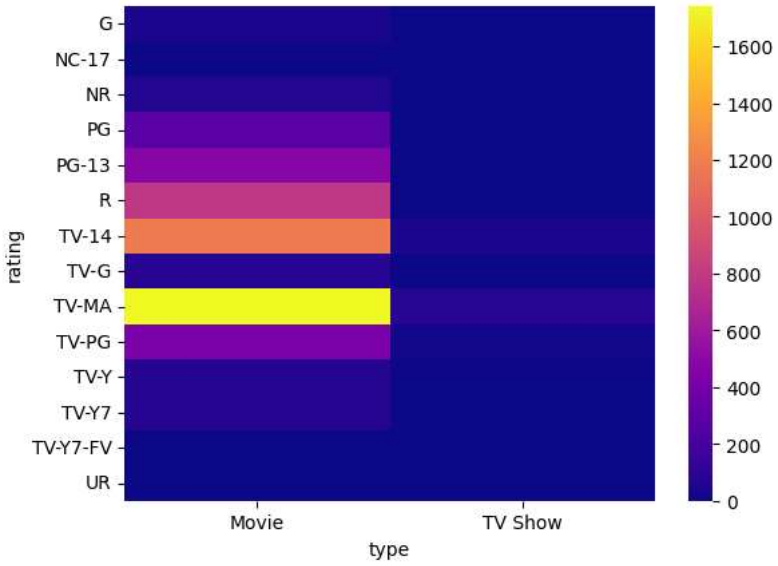
```
<ipython-input-62-b989fecd7265>:8: DeprecationWarning:

`np.bool` is a deprecated alias for the builtin `bool`. To silence this warning, use `bool` by itself. Doing this will not modify any be
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
```



```
colormap = plt.cm.plasma
sns.heatmap(pd.crosstab(df["rating"], df["type"]), cmap = colormap)
```

```
<Axes: xlabel='type', ylabel='rating'>
```




```
mf = df
mf =mf.drop(['show_id', 'title', 'director', 'cast', 'rating', 'duration', 'description'])
mf.head()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-64-37de5c59fcbf> in <cell line: 2>()
      1 mf = df
----> 2 mf =mf.drop(['show_id', 'title', 'director', 'cast', 'rating', 'duration', 'description'])
      3 mf.head()

                                 5 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in drop(self, labels, errors)
   6932         if mask.any():
   6933             if errors != "ignore":
-> 6934                 raise KeyError(f"{list(labels[mask])} not found in axis")
   6935             indexer = indexer[~mask]
   6936         return self.delete(indexer)

KeyError: "['show_id', 'title', 'director', 'cast', 'rating', 'duration', 'description'] not found in axis"

SEARCH STACK OVERFLOW
```

```
df['cast'].fillna(df['cast'].mode(), inplace = True)
```

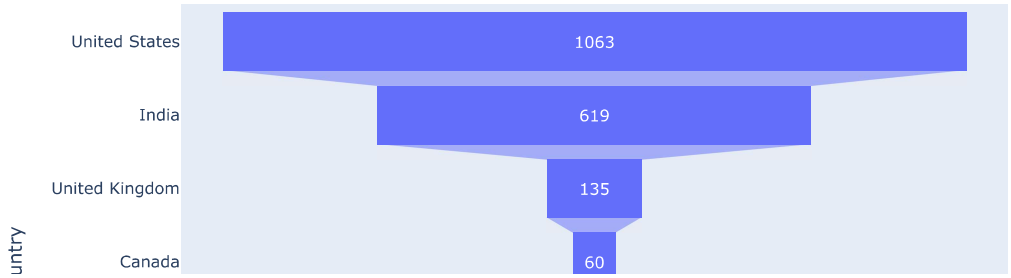
```
df['cast']
```

```
7       Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...
8       Mel Giedroyc, Sue Perkins, Mary Berry, Paul Ho...
9       Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...
12      Luna Wedler, Jannis Niewöhner, Milan Peschel, ...
24      Prashanth, Aishwarya Rai Bachchan, Sri Lakshmi...
                              ...
8801    Ali Suliman, Saleh Bakri, Yasa, Ali Al-Jabri, ...
8802    Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...
8804    Jesse Eisenberg, Woody Harrelson, Emma Stone, ...
8805    Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...
8806    Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...
Name: cast, Length: 5332, dtype: object
```

```
df.isna().sum()
```

```
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
day_added       0
year_added      0
month_added     0
dtype: int64
```

```
# plt.figure(figsize = (35,6))
df = pd.read_csv('/netflix.csv')
import plotly.express as px
data = dict(number=[1063,619,135,60,44,41,40],
 country=["United States", "India", "United Kingdom", "Canada", "Spain",'Turkey','Philippines'])
fig = px.funnel(data, x='number', y='country')
fig.show()
```



df['duration'].value_counts().reset_index()

pair plot of type and released year:


```
plt.figure(figsize = (35,6))
sns.pairplot(mf,hue='type')
```

```
<seaborn.axisgrid.PairGrid at 0x7b09f071d6c0>
```

5. Missing Value & Outlier check (Treatment optional):

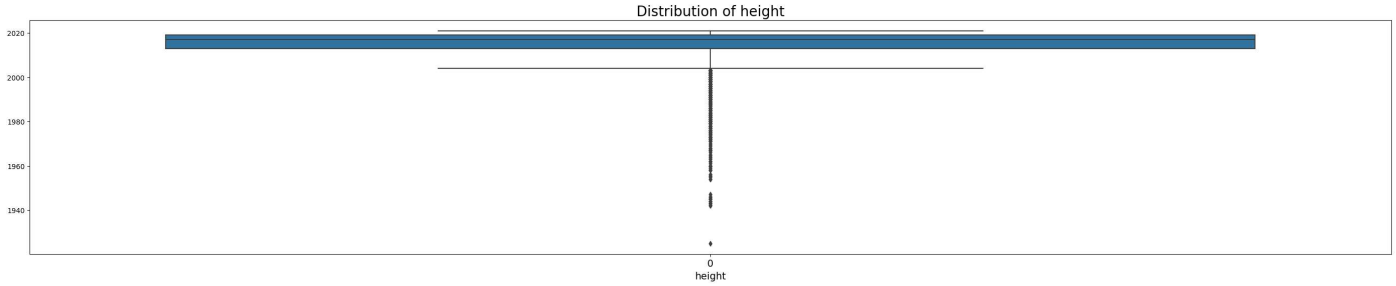
Double-click (or enter) to edit

```
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
```

```
plt.figure(figsize = (35,6))
# box plot of the release year
ax = sns.boxplot(df['release_year'])
# notation indicating an outlier
ax.annotate('Outlier', xy=(190,0), xytext=(186,-0.05), fontsize=14,
 arrowprops=dict(arrowstyle='->', ec='grey', lw=2), bbox = dict(boxstyle="round"))
# xtick, label, and title
plt.xticks(fontsize=14)
plt.xlabel('height', fontsize=14)
plt.title('Distribution of height', fontsize=20)
```

```
Text(0.5, 1.0, 'Distribution of height')
```




```
df['cast'].fillna(df['cast'].mode(), inplace = True)
```

```
df['cast']
```

```
0                                    David Attenborough
1       Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...
2       Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...
3                                                     NaN
4       Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...
                              ...
8802    Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...
8803                                                  NaN
8804    Jesse Eisenberg, Woody Harrelson, Emma Stone, ...
8805    Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...
8806    Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...
Name: cast, Length: 8807, dtype: object
```

```
df.isna().sum()
```

```
show_id            0
type               0
title              0
director        2634
cast             824
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

treating null values

```
#unnesting the directors column, i.e- creating separate lines for each director in a movie
constraint1=df['director'].apply(lambda x: str(x).split(', ')).tolist()
df_new1=pd.DataFrame(constraint1,index=df['title'])
df_new1=df_new1.stack()
df_new1=pd.DataFrame(df_new1.reset_index())
df_new1.rename(columns={0:'Directors'},inplace=True)
df_new1.drop(['level_1'],axis=1,inplace=True)
df_new1.head()
```

	title	Directors
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	nan
4	Kota Factory	nan

```
#unnesting the cast column, i.e- creating separate lines for each cast member in a movie
constraint2=df['cast'].apply(lambda x: str(x).split(', ')).tolist()
df_new2=pd.DataFrame(constraint2,index=df['title'])
df_new2=df_new2.stack()
df_new2=pd.DataFrame(df_new2.reset_index())
df_new2.rename(columns={0:'Actors'},inplace=True)
df_new2.drop(['level_1'],axis=1,inplace=True)
df_new2.head()
```

	title	Actors
0	Dick Johnson Is Dead	David Attenborough
1	Blood & Water	Ama Qamata
2	Blood & Water	Khosi Ngema
3	Blood & Water	Gail Mabalane
4	Blood & Water	Thabang Molaba

```
#unnesting the listed_in column, i.e- creating separate lines for each genre in a movie
constraint3=df['listed_in'].apply(lambda x: str(x).split(', ')).tolist()
df_new3=pd.DataFrame(constraint3,index=df['title'])
df_new3=df_new3.stack()
df_new3=pd.DataFrame(df_new3.reset_index())
df_new3.rename(columns={0:'Genre'},inplace=True)
df_new3.drop(['level_1'],axis=1,inplace=True)
df_new3.head()
```

```
#unnesting the country column, i.e- creating separate lines for each country in a movie
constraint4=df['country'].apply(lambda x: str(x).split(', ')).tolist()
df_new4=pd.DataFrame(constraint4,index=df['title'])
df_new4=df_new4.stack()
df_new4=pd.DataFrame(df_new4.reset_index())
df_new4.rename(columns={0:'country'},inplace=True)
df_new4.drop(['level_1'],axis=1,inplace=True)
df_new4.head()
```

	title	country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	nan
3	Jailbirds New Orleans	nan
4	Kota Factory	India

```
#merging the unnested director data with unnested actors data
df_new5=df_new2.merge(df_new1,on=['title'],how='inner')
#merging the above merged data with unnested genre data
df_new6=df_new5.merge(df_new3,on=['title'],how='inner')
#merging the above merged data with unnested country data
df_new=df_new6.merge(df_new4,on=['title'],how='inner')
```

```
#replacing nan values of director and actor by Unknown Actor and Director
df_new['Actors'].replace(['nan'],['Unknown Actor'],inplace=True)
df_new['Directors'].replace(['nan'],['Unknown Director'],inplace=True)
df_new['country'].replace(['nan'],[np.nan],inplace=True)
df_new.head()
```

	title	Actors	Directors	Genre	country
0	Dick Johnson Is Dead	David Attenborough	Kirsten Johnson	Documentaries	United States
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa
3	Blood & Water	Ama Qamata	Unknown Director	TV Mysteries	South Africa
4	Blood & Water	Khosi Ngema	Unknown Director	International TV Shows	South Africa

```
#merging our unnested data with the original data
df_final=df_new.merge(df[['show_id', 'type', 'title', 'date_added',
       'release_year', 'rating', 'duration']],on=['title'],how='left')
df_final.head()
```

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year	rating	duration
0	Dick Johnson Is Dead	David Attenborough	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90 min
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons
			Unknown		South		TV	September 24,			2

```
#now checking nulls
df_final.isnull().sum()
```

```
title              0
Actors             0
Directors          0
Genre              0
country        11897
show_id            0
type               0
date_added       158
release_year       0
rating            67
```

```
duration           3
dtype: int64
```

in duration column, it was observed that the nulls had values which were written in corresponding ratings column, i.e- you can't expect ratings to be in min. So the duration column nulls are replaced by corresponding values in ratings column

```
df_final.loc[df_final['duration'].isnull(),'duration']=df_final.loc[df_final['duration'].isnull(),'duration'].fillna(df_final['rating'])
```

```
df_final.loc[df_final['rating'].str.contains('min', na=False),'rating']='NR'
```

```
df_final.isnull().sum()
```

```
title              0
Actors             0
Directors          0
Genre              0
country        11897
show_id            0
type               0
date_added       158
release_year       0
rating            67
duration           0
dtype: int64
```

```
#Ratings can't be in min, so it has been made NR(i.e- Non Rated)
df_final.loc[df_final['rating'].str.contains('min', na=False),'rating']='NR'
df_final['rating'].fillna('NR',inplace=True)
pd.set_option('display.max_rows',None)
```

```
#just an attempt to observe nulls in date_added column
df_final[df_final['date_added'].isnull()].head()
```

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year	rating	duration
136893	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	British TV Shows	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons
136894	A Young Doctor's Notebook and Other Stories	Daniel Radcliffe	Unknown Director	TV Comedies	United Kingdom	s6067	TV Show	NaN	2013	TV-MA	2 Seasons
136895	A Young Doctor's Notebook and Other	Daniel	Unknown	TV Dramas	United	s6067	TV	NaN	2013	TV-MA	2

```
#date added column is imputed on the basis of release year,i.e- suppose there's a null for date_added
#when release year was 2013.So below piece of code just checks the mode of date added for release year=2013
# and imputes in place of nulls the corresponding mode

for i in df_final[df_final['date_added'].isnull()]['release_year'].unique():
  imp=df_final[df_final['release_year']==i]['date_added'].mode().values[0]
  df_final.loc[df_final['release_year']==i,'date_added']=df_final.loc[df_final['release_year']==i,'date_added'].fillna(imp)
```

```
#country column is imputed on the basis of director,i.e- suppose there's a null for country
#when we have a director whose other movies have a country given.So below piece of code just checks the mode of
#country for the director
# and imputes in place of nulls the corresponding mode

for i in df_final[df_final['country'].isnull()]['Directors'].unique():
  if i in df_final[~df_final['country'].isnull()]['Directors'].unique():
    imp=df_final[df_final['Directors']==i]['country'].mode().values[0]
    df_final.loc[df_final['Directors']==i,'country']=df_final.loc[df_final['Directors']==i,'country'].fillna(imp)
```

So we imputed the country column on the basis of directors whose other movie titles had countries given. But there might be directors who have only one occurence in our data. In that scenario, I have used Actors as a basis. i.e- for this Actor majorly acts in movies of which country? Imputation has been done on this basis. For remaining rows, country has been filled as Unknown Country

```
for i in df_final[df_final['country'].isnull()]['Actors'].unique():
  if i in df_final[~df_final['country'].isnull()]['Actors'].unique():
    imp=df_final[df_final['Actors']==i]['country'].mode().values[0]
    df_final.loc[df_final['Actors']==i,'country']=df_final.loc[df_final['Actors']==i,'country'].fillna(imp)
#If there are still nulls, I just replace it by Unknown Country
df_final['country'].fillna('Unknown Country',inplace=True)
df_final.isnull().sum()
```

```
title           0
Actors          0
Directors       0
Genre           0
country         0
show_id         0
type            0
date_added      0
release_year    0
rating          0
duration        0
dtype: int64
```

```
df_final.head()
```

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year	rating	duration
0	Dick Johnson Is Dead	David Attenborough	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90 min
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons
			Unknown		South		TV	September 24,			2

```
df_final['duration'].value_counts()
```

```
34 min          6
17 min          5
39 min          5
10 min          4
16 min          4
196 min         4
20 min          4
18 min          4
3 min           4
5 min           3
11 min          2
8 min           2
9 min           2
Name: duration, dtype: int64
```

```
#removing mins from data
df_final['duration']=df_final['duration'].str.replace(" min","")
df_final.head()
```

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year	rating	duration
0	Dick Johnson Is Dead	David Attenborough	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons
			Unknown		South		TV	September 24,			2

```
df_final['duration'].unique()
```

```
array(['90', '2 Seasons', '1 Season', '91', '125', '9 Seasons', '104',
       '127', '4 Seasons', '67', '94', '5 Seasons', '161', '61', '166',
       '147', '103', '97', '106', '111', '3 Seasons', '110', '105', '96',
       '124', '116', '98', '23', '115', '122', '99', '88', '100',
       '6 Seasons', '102', '93', '95', '85', '83', '113', '13', '182',
       '48', '145', '87', '92', '80', '117', '128', '119', '143', '114',
       '118', '108', '63', '121', '142', '154', '120', '82', '109', '101',
       '86', '229', '76', '89', '156', '112', '107', '129', '135', '136',
       '165', '150', '133', '70', '84', '140', '78', '7 Seasons', '64',
       '59', '139', '69', '148', '189', '141', '130', '138', '81', '132',
       '10 Seasons', '123', '65', '68', '66', '62', '74', '131', '39',
       '46', '38', '8 Seasons', '17 Seasons', '126', '155', '159', '137',
       '12', '273', '36', '34', '77', '60', '49', '58', '72', '204',
       '212', '25', '73', '29', '47', '32', '35', '71', '149', '33', '15',
       '54', '224', '162', '37', '75', '79', '55', '158', '164', '173',
       '181', '185', '21', '24', '51', '151', '42', '22', '134', '177',
       '13 Seasons', '52', '14', '53', '8', '57', '28', '50', '9', '26',
       '45', '171', '27', '44', '146', '20', '157', '17', '203', '41',
       '30', '194', '15 Seasons', '233', '237', '230', '195', '253',
       '152', '190', '160', '208', '180', '144', '5', '174', '170', '192',
       '209', '187', '172', '16', '186', '11', '193', '176', '56', '169',
       '40', '10', '3', '168', '312', '153', '214', '31', '163', '19',
       '12 Seasons', '179', '11 Seasons', '43', '200', '196', '167',
       '178', '228', '18', '205', '201', '191'], dtype=object)
```

```
df_final['duration_copy']=df_final['duration'].copy()
df_final1=df_final.copy()
```

```
df_final1.loc[df_final1['duration_copy'].str.contains('Season'),'duration_copy']=0
df_final1['duration_copy']=df_final1['duration_copy'].astype('int')
df_final1.head()
```

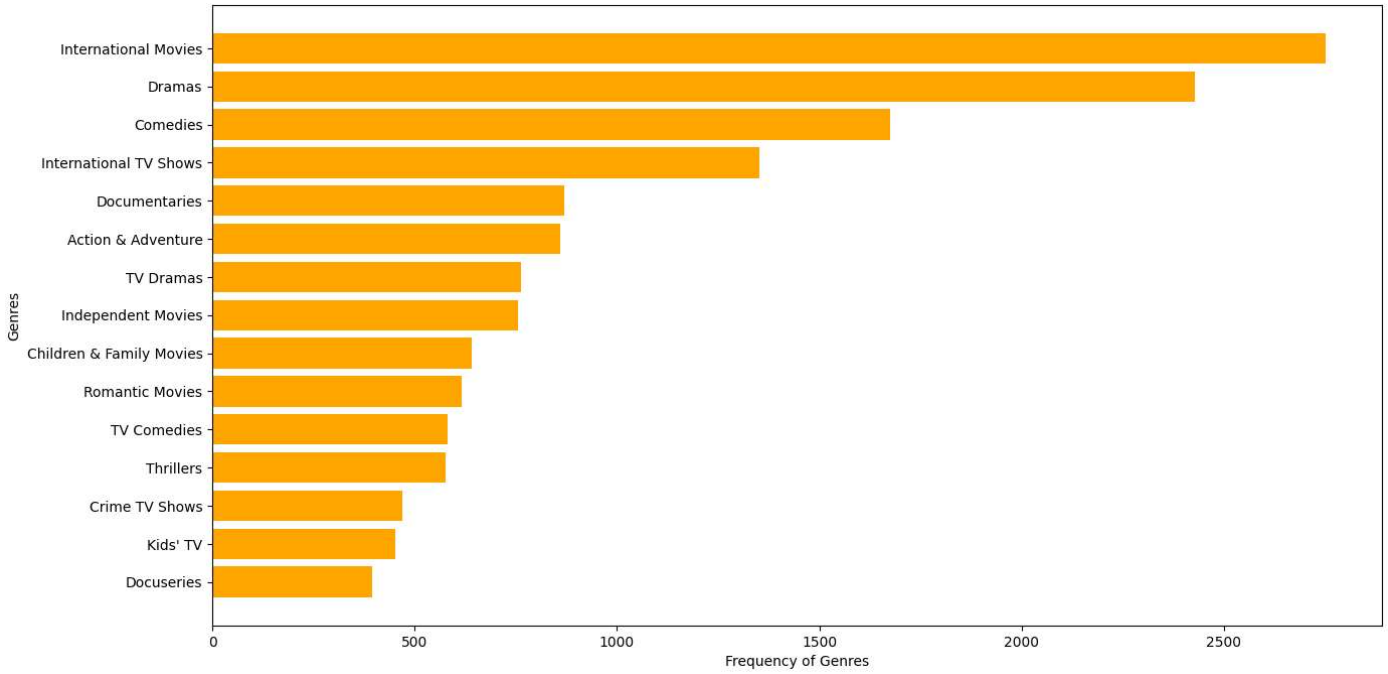
	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year	rating	duration	duration_copy
0	Dick Johnson Is Dead	David Attenborough	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90	90
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	0
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	0
3	Blood & Water	Ama Qamata	Unknown Director	TV Mysteries	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	0

```
df_final1['duration_copy'].describe()
```

```
count    201991.000000
mean         77.152789
std          52.269154
min           0.000000
25%           0.000000
50%          95.000000
75%         112.000000
max         312.000000
Name: duration_copy, dtype: float64
```

6. Insights based on Non-Graphical and Visual Analysis:

```
df_genre=df_final1.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)[:15]
plt.figure(figsize=(15,8))
plt.barh(df_genre[::-1]['Genre'], df_genre[::-1]['title'],color=['orange'])
plt.xlabel('Frequency of Genres')
plt.ylabel('Genres')
plt.show()
```

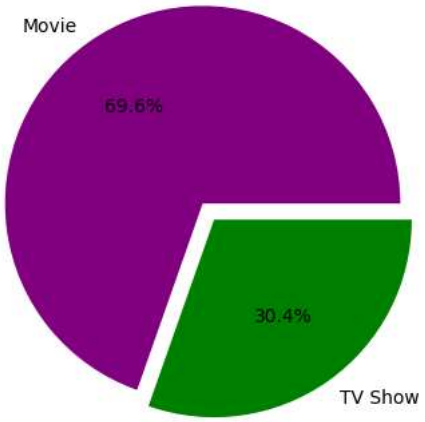


International Movies, Dramas and Comedies are the most popular .

```
#number of distinct titles on the basis of type
df_final1.groupby(['type']).agg({"title":"nunique"})
```

type	title
Movie	6131
TV Show	2676

```
df_type=df_final1.groupby(['type']).agg({"title":"nunique"}).reset_index()
plt.pie(df_type['title'],explode=(0.05,0.05), labels=df_type['type'],colors=['purple','green'],autopct='%.1f%%')
plt.show()
```



We have 70:30 ratio of Movies and TV Shows in our data

```
#number of distinct titles on the basis of country
df_final1.groupby(['country']).agg({"title":"nunique"})
```

country	title
	3
Afghanistan	1
Albania	1
Algeria	3
Angola	2
Argentina	94
Armenia	1
Australia	162
Austria	12
Azerbaijan	1
Bahamas	1
Bangladesh	4
Belarus	1
Belgium	94
Bermuda	1
Botswana	1
Brazil	103
Bulgaria	10
Burkina Faso	1
Cambodia	5
Cambodia,	1
Cameroon	2
Canada	460
Cayman Islands	2
Chile	30
China	166
Colombia	54
Croatia	4
Cuba	2
Cyprus	1
Czech Republic	23
Denmark	50
Dominican Republic	1
East Germany	1
Ecuador	1
Egypt	134
Ethiopia	1
Finland	12
France	409
Georgia	2
Germany	231
Ghana	8
Greece	11
Guatemala	2
Hong Kong	110

Hungary	11
Iceland	11
India	1138
Indonesia	97
Iran	4
Iraq	2
Ireland	46
Israel	30
Italy	102
Jamaica	1
Japan	338
Jordan	10
Kazakhstan	1
Kenya	6
Kuwait	9
Latvia	1
Lebanon	33
Liechtenstein	1
Lithuania	1
Luxembourg	12
Malawi	1
Malaysia	26
Malta	3
Mauritius	3
Mexico	175
Mongolia	1
Montenegro	1
Morocco	6
Mozambique	1
Namibia	2
Nepal	2
Netherlands	50
New Zealand	33
Nicaragua	1
Nigeria	140
Norway	30
Pakistan	24
Palestine	1
Panama	1
Paraguay	1
Peru	11
Philippines	90
Poland	41
Poland,	1
Portugal	6
Puerto Rico	1
Qatar	10

Romania	14
Russia	27
Samoa	1
Saudi Arabia	14
Senegal	3
Serbia	7
Singapore	41
Slovakia	1
Slovenia	3

The above dataframe shows a flaw in which we are seeing countries, such as Cambodia and Cambodia, or United States and United States, are shown as different countries.They should have been same

South Korea	235

```
df_final1['country'] = df_final1['country'].str.replace(',', '')
df_final1.head()
```

	title	Actors	Directors	Genre	country	show_id	type	date_added	release_year	rating	duration	duration_copy
0	Dick Johnson Is Dead	David Attenborough	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90	90
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	0
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	0
3	Blood & Water	Ama Qamata	Unknown Director	TV Mysteries	South Africa	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	0

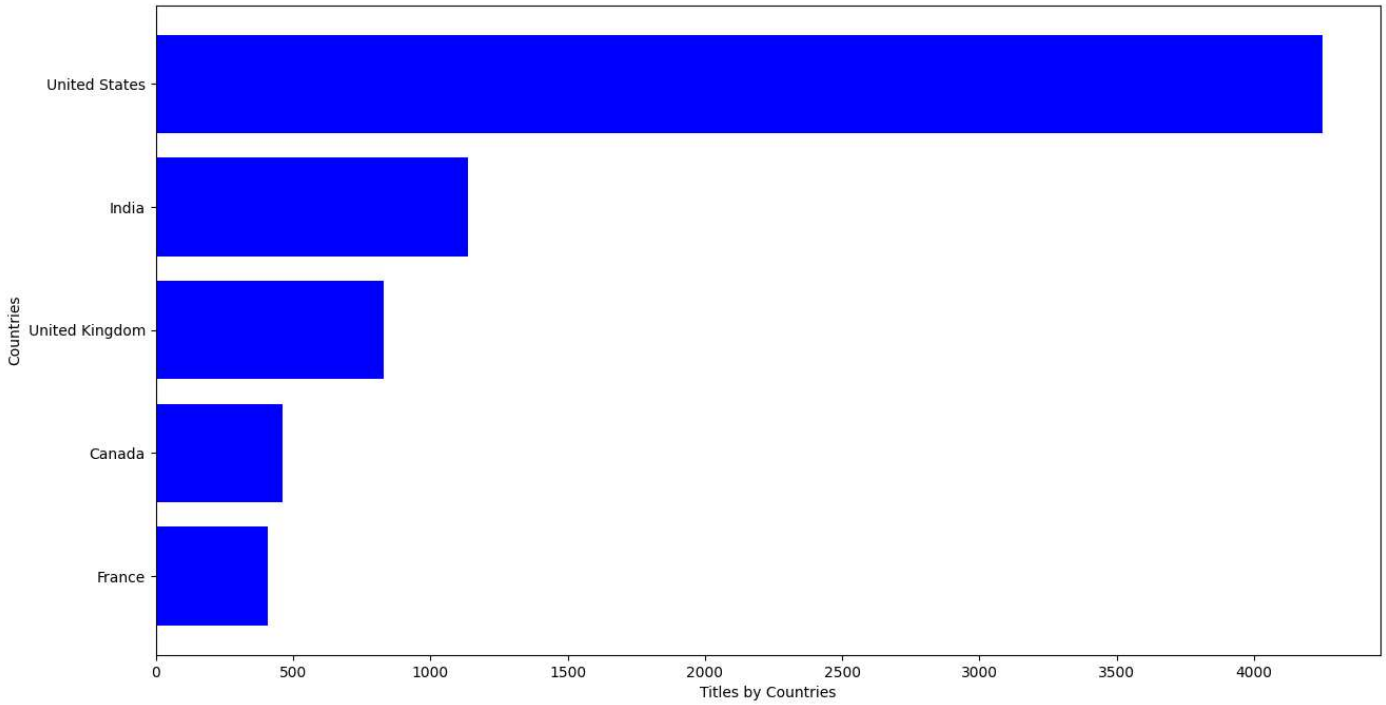
```
#number of distinct titles on the basis of country
df_final1.groupby(['country']).agg({"title":"nunique"})
```

country	title
	3
Afghanistan	1
Albania	1
Algeria	3
Angola	2
Argentina	94
Armenia	1
Australia	162
Austria	12
Azerbaijan	1
Bahamas	1
Bangladesh	4
Belarus	1
Belgium	94
Bermuda	1
Botswana	1
Brazil	103
Bulgaria	10
Burkina Faso	1
Cambodia	6
Cameroon	2
Canada	460
Cayman Islands	2
Chile	30
China	166
Colombia	54
Croatia	4
Cuba	2
Cyprus	1
Czech Republic	23
Denmark	50
Dominican Republic	1
East Germany	1
Ecuador	1
Egypt	134
Ethiopia	1
Finland	12
France	409
Georgia	2
Germany	231
Ghana	8
Greece	11
Guatemala	2
Hong Kong	110
Hungary	11

Iceland	11
India	1138
Indonesia	97
Iran	4
Iraq	2
Ireland	46
Israel	30
Italy	102
Jamaica	1
Japan	338
Jordan	10
Kazakhstan	1
Kenya	6
Kuwait	9
Latvia	1
Lebanon	33
Liechtenstein	1
Lithuania	1
Luxembourg	12
Malawi	1
Malaysia	26
Malta	3
Mauritius	3
Mexico	175
Mongolia	1
Montenegro	1
Morocco	6
Mozambique	1
Namibia	2
Nepal	2
Netherlands	50
New Zealand	33
Nicaragua	1
Nigeria	140
Norway	30
Pakistan	24
Palestine	1
Panama	1
Paraguay	1
Peru	11
Philippines	90
Poland	42
Portugal	6
Puerto Rico	1
Qatar	10
Romania	14
Russia	27

Russia	27
Samoa	1
Saudi Arabia	14
Senegal	3
Serbia	7
Singapore	41
Slovakia	1
Slovenia	3
Somalia	1
South Africa	65
South Korea	235

```
df_country=df_final1.groupby(['country']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)[:5]
plt.figure(figsize=(15,8))
plt.barh(df_country[::-1]['country'], df_country[::-1]['title'],color=['blue'])
plt.xlabel('Titles by Countries')
plt.ylabel('Countries')
plt.show()
```

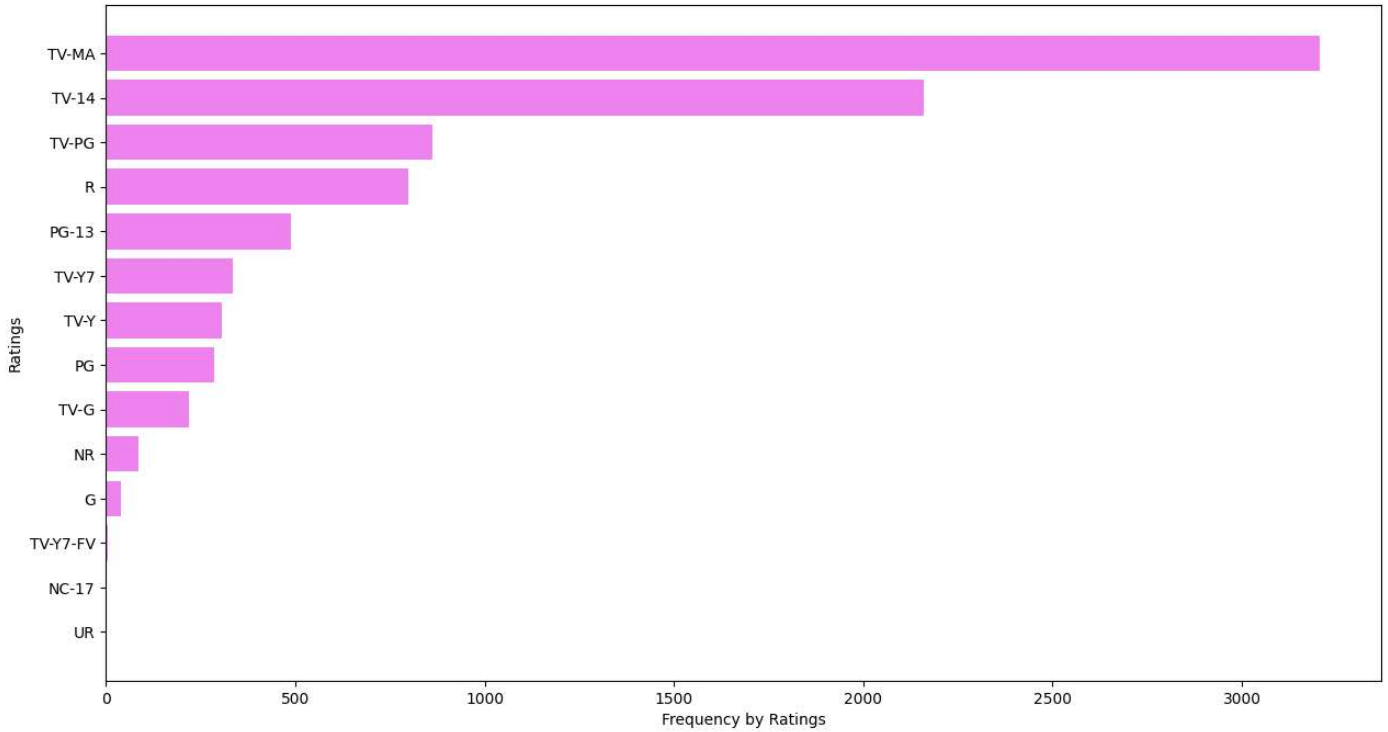


US,India,UK,Canada and France are leading countries in Content Creation on Netflix

```
#number of distinct titles on the basis of rating
df_final1.groupby(['rating']).agg({"title":"nunique"})
```

rating	title
G	41
NC-17	3
NR	87
PG	287
PG-13	490
R	799
TV-14	2160
TV-G	220
TV-MA	3207

```
df_rating=df_final1.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)[:15]
plt.figure(figsize=(15,8))
plt.barh(df_rating[::-1]['rating'], df_rating[::-1]['title'],color=['violet'])
plt.xlabel('Frequency by Ratings')
plt.ylabel('Ratings')
plt.show()
```



Most of the highly rated content on Netflix is intended for Mature Audiences, R Rated, content not intended for audience under 14 and those which require Parental Guidance

```
#number of distinct titles on the basis of duration
df_final1.groupby(['duration']).agg({"title":"nunique"})
```

duration	title
1 Season	1793
10	1
10 Seasons	7
100	108
101	116
102	122
103	114
104	104
105	101
106	111
107	98
108	87
109	69
11	2
11 Seasons	2
110	97
111	68
112	74
113	69
114	56
115	61
116	80
117	61
118	65
119	63
12	3
12 Seasons	2
120	56
121	54
122	45
123	44
124	52
125	36
126	44
127	48
128	41
129	32
13	3
13 Seasons	3
130	40
131	34
132	37
133	42
134	22
135	39

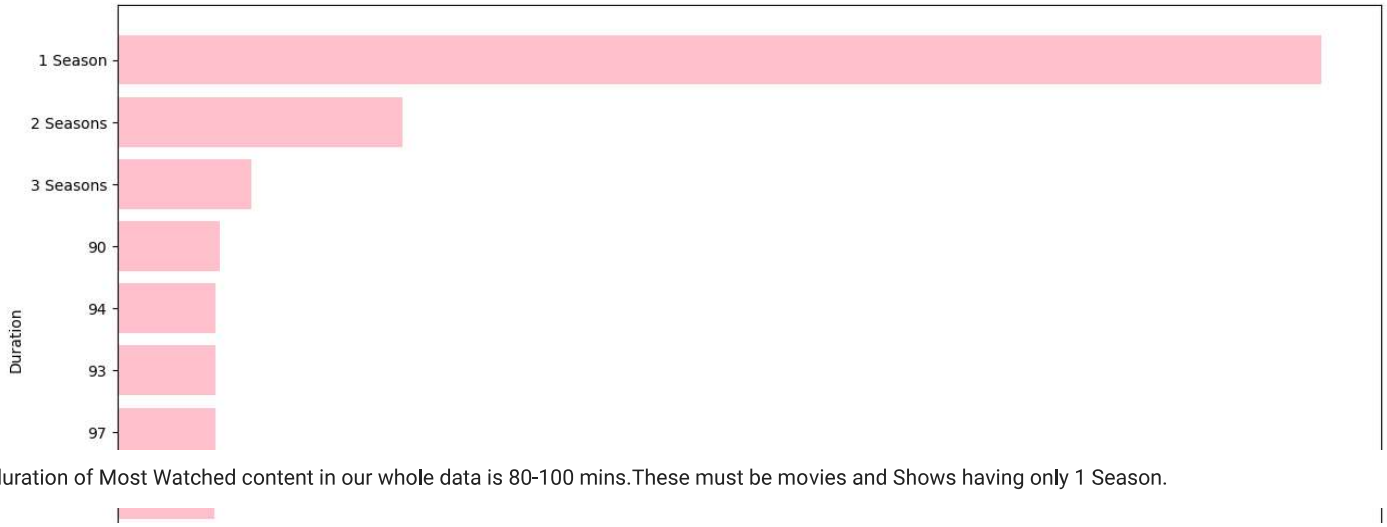
136	23
137	38
138	21
139	22
14	3
140	25
141	19
142	13
143	23
144	9
145	18
146	13
147	12
148	19
149	15
15	3
15 Seasons	2
150	17
151	15
152	5
153	11
154	13
155	10
156	10
157	6
158	12
159	6
16	1
160	6
161	10
162	14
163	11
164	4
165	8
166	8
167	1
168	7
169	2
17	3
17 Seasons	1
170	5
171	7
172	4
173	6
174	2
176	5
177	5

178	1
179	2
18	1
180	2
181	4
182	3
185	6
186	1
187	2
189	1
19	2
190	2
191	1
192	2
193	1
194	1
195	2
196	1
2 Seasons	425
20	2
200	1
201	1
203	1
204	2
205	1
208	1
209	2
21	3
212	1
214	1
22	16
224	1
228	1
229	1
23	13
230	1
233	1
237	1
24	23
25	11
253	1
26	6
27	3
273	1
28	10
29	11

3	1
3 Seasons	199
30	6
31	2
312	1
32	9
33	6
34	3
35	5
36	5
37	3
38	5
39	2
4 Seasons	95
40	13
41	3
42	9
43	1
44	19
45	10
46	24
47	11
48	8
49	9
5	1
5 Seasons	65
50	10
51	11
52	20
53	24
54	24
55	16
56	12
57	14
58	25
59	25
6 Seasons	33
60	29
61	31
62	24
63	32
64	23
65	25
66	29
67	21
68	25
69	28

7 Seasons	23
70	28
71	28
72	33
73	30
74	32
75	35
76	31
77	30
78	45
79	35
8	1
8 Seasons	17
80	43
81	62
82	52
83	65
84	68
85	73
86	103
87	101
88	116
89	106
9	1
9 Seasons	9
90	152
91	144
92	129
93	146
94	146
95	137

```
df_duration=df_final1.groupby(['duration']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)[:10]
plt.figure(figsize=(15,8))
plt.barh(df_duration[::-1]['duration'], df_duration[::-1]['title'],color=['pink'])
plt.xlabel('Frequency by Duration')
plt.ylabel('Duration')
plt.show()
```



The duration of Most Watched content in our whole data is 80-100 mins.These must be movies and Shows having only 1 Season.

```
#number of distinct titles on the basis of Actors
df_final1.groupby(['Actors']).agg({"title":"nunique"})
```

A. Murat Özgen	1
A.C. Peterson	1
A.D. Miles	3
A.J. Cook	2
A.J. Johnson	1
A.J. LoCascio	3
A.K. Hangal	4
A.R. Rahman	1
A.S. Sasi Kumar	1
AC Lim	1
AFRA	1
AJ Bowen	1
AJ Michalka	1
AJ Rivera	1
ARAH	2
Aabhas Yadav	1
Aachal Munjal	1
Aadarsh Balakrishna	2
Aadhi	1
Aadhya Anand	1
Aadil Khan	1
Aaditi Pohankar	2
Aaditya Pratap Singh	1
Aadukalam Naren	1
Aadya Bedi	1
Aahana Kumra	2
Aakarshan Singh	1
Aakash Dabhade	4
Aakash Dahiya	1
Aakash Pandey	1
Aakshath Das	1
Aamina Sheikh	1
Aamir Ahmed	1
Aamir Bashir	5
Aamir Khan	16
Aamir Qureshi	1
Aanand Kale	1
Aanchal Munjal	1
Aarav Khanna	1
Aarif Rahman	1
Aarjav Trivedi	1
Aarna Sharma	1
Aarohi Patel	1
Aaron Abrams	4
Aaron Altaras	1
Aaron Ashmore	1

Aaron Blakely	1
Aaron Burns	2
Aaron Carpenter	1
Aaron Chen	1
Aaron Chow	1
Aaron D. Spears	1
Aaron Dismuke	1
Aaron Douglas	3
Aaron Eckhart	7
Aaron Eisenberg	1
Aaron Farb	1
Aaron Glenane	1
Aaron Guy	1
Aaron Hale	1
Aaron Hernandez	1
Aaron Hilmer	1
Aaron Himelstein	2
Aaron Jakubenko	3
Aaron Jeffery	3
Aaron Keogh	1
Aaron Kwok	1
Aaron L. McGrath	2
Aaron Marsden	2
Aaron Marshall	1
Aaron McCusker	2
Aaron Merke	1
Aaron Michael Drozin	1
Aaron Moorhead	1
Aaron Moten	1
Aaron Munoz	1
Aaron Paul	8
Aaron Pearl	1
Aaron Pedersen	1
Aaron Pruner	1
Aaron Stanford	2
Aaron Staton	1
Aaron Taylor-Johnson	4
Aaron Tveit	1
Aaron Washington	1
Aaron Wolff	1
Aaron Yan	4
Aaron Yoo	3
Aarti Chhabria	1
Aarti Mann	1
Aarti Patel	1
Aarubala	1
Aarushi Sharma	1
Aasa D	1

Aarya Dave	1
Aarya DharmChand Kumar	1
Aaryan Menon	1
Aaryansh Malviya	1
Aarón Díaz	1
Aasha Pawar	1
Aashay Kulkarni	1
Aashi Rawal	1
Aashif Sheikh	1
Aashish Chaudhary	3
Aashish Kulkarni	1
Aasif Mandvi	6
Aayam Mehta	2
Aayan Boradia	1
Abayomi Alvin	1
Abba Ali Zaky	1
Abbas	1
Abbey Lee	1
Abbi Jacobson	3
Abbie Cornish	4
Abboudy Mallah	1
Abby Bergman	1
Abby Donnelly	3
Abby Miller	1
Abby Quinn	1
Abby Rakic-Platt	1
Abby Ryder Fortson	1
Abby Trott	6
Abdalah Mishrif	4
Abdalla Mahmoud	2
Abdel Aziz El Mountassir	1
Abdel Aziz Khalil	1
Abdel Ghani Benizza	1
Abdel Ghani Nagdi	1
Abdel Imam Abdullah	2
Abdel Moneim Amayri	1
Abdel Moneim Madbouly	1
Abdel Nasser Maraqbi	1
Abdel-Wareth Asar	1
Abdelaziz N'Mila	3
Abdelghani Kitab	1
Abdelghany Kamar	1
Abdelilah Wahbi	4
Abdellah Bensaid	1
Abdellah Didane	1
Abdi Sidow Farah	1
Abdillah Assoumani	1

Abdo Chahine	1
Abdramane Diakité	1
Abdul Khoza	1
Abdul Mohsen Alnimer	1
Abdul Qadir	1
Abdulateif Saud	1
Abdulaziz Al-mesallam	1
Abdulaziz Almuzaini	1
Abdulaziz Alshehri	2
Abdulaziz El Nassar	1
Abdulhussain Abdulredah	2
Abdulla Al-khudr	1
Abdulla Bu Shehri	1
Abdullah Al Hamiri	2
Abdullah Al Yousef	1
Abdullah Al-Ramsi	2
Abdullah Al-Turkumani	2
Abdullah Al-bloshi	2
Abdullah Bahman	1
Abdullah Bin Heider	1
Abdullah Ibrahim	1
Abdullah Jan	1
Abdullah Moshref	1
Abdullrahman Al Gohani	1
Abdulmajeed Al-Ruhaidi	1
Abdulmohsen Al-Qaffas	2
Abdulmohsen Alnemr	1
Abdul' Usman Zada	1
Abdur Arsyad	1
Abdur Rehman	1
Abdurrahman Arif	4
Abe Clifford-Barr	1
Abe Goldfarb	1
Abe Vigoda	1
Abeer Abrar	1
Abeer Ahmad	1
Abeer Alotaibi	1
Abeer Mansour	1
Abeer Mohammed	1
Abel Ayala	2
Abel Ferrara	1
Abel Folk	2
Abel Franco	1
Abel McSurely Bradshaw	1
Abel Tesfaye	1
Abella Bala	1
Abella Wyss	1