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

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
path = "/content/drive/MyDrive/Dataset/netflix_titles.csv"
df = pd.read_csv(path)
df
```

	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...
...	...	...	...	...	...	...	...	...	...	...	...	...

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.sample(10)
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	descript:
5317	s5318	Movie	Berlin Syndrome	Cate Shortland	Matthias Habich, Max Riemelt, Teresa Palmer, L...	Australia	August 25, 2017	2017	R	116 min	International Movies, Thrillers	What starts a passion one-ni stand q
8479	s8480	Movie	The Redeemed and the Dominant: Fittest on Earth	Heber Cannon, Mariah Moore, Marston Sawyers	NaN	United States	July 1, 2018	2017	TV-14	120 min	Documentaries, Sports Movies	Questi ab enduran doping a overa
2057	s2058	Movie	Sky Tour: The Movie	Nguyen Thanh Tung	Son Tung M-TP	Vietnam	September 2, 2020	2020	TV-G	93 min	Documentaries, International Movies, Music & M...	From preparati to performanc tl
3357	s3358	TV Show	Sleepless Society: Bedtime Wishes	NaN	Shahkrit Yamnarm, Savika Chaiyadej, Supoj Chan...	NaN	October 31, 2019	2019	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Dramas	Durin holiday stay a hotel res a
40	s41	TV Show	He-Man and the Masters of the Universe	NaN	Yuri Lowenthal, Kimberly Brooks, Antony Del Ri...	United States	September 16, 2021	2021	TV-Y7	1 Season	Kids' TV, TV Sci-Fi & Fantasy	Mighty te Adam and heroic squ of mis
2083	s2084	Movie	The Debt Collector 2	Jesse V. Johnson	Scott Adkins, Louis Mandylor,	United States	August 31, 2020	2020	TV-MA	97 min	Action & Adventure	French a Sue have days to col

```
len(df)
```

```
8807
```

```
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
```

```
df.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

```
df.isnull().sum()
```

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

dtype: int64

```
df["type"].value_counts(normalize = True) * 100
```

type	proportion
Movie	69.615079
TV Show	30.384921

dtype: float64

```
constaraints = df["director"].apply(lambda x:str(x).split(", ")).tolist()
```

```
print(constaraints)
```

```
[['Kirsten Johnson'], ['nan'], ['Julien Leclercq'], ['nan'], ['nan'], ['Mike Flanagan'], ['Robert Cullen', 'José Luis Ucha'], ['Hai
```

```
df_new1 =pd.DataFrame(constaraints , index= df["title"])
df_new1.head()
```

title	0	1	2	3	4	5	6	7	8	9	10	11	12
Dick Johnson Is Dead	Kirsten Johnson	None	None	None	None	None	None	None	None	None	None	None	None
Blood & Water	nan	None	None	None	None	None	None	None	None	None	None	None	None
Ganglands	Julien Leclercq	None	None	None	None	None	None	None	None	None	None	None	None
Jailbirds New Orleans	nan	None	None	None	None	None	None	None	None	None	None	None	None
Kota Factory	nan	None	None	None	None	None	None	None	None	None	None	None	None

Next steps: Generate code with df_new1 | View recommended plots | New interactive sheet

```
df_new1 = df_new1.stack()
df_new1.head(50)
```

title		0
Dick Johnson Is Dead	0	Kirsten Johnson
Blood & Water	0	nan
Ganglands	0	Julien Leclercq
Jailbirds New Orleans	0	nan
Kota Factory	0	nan
Midnight Mass	0	Mike Flanagan
My Little Pony: A New Generation	0	Robert Cullen
	1	José Luis Ucha
Sankofa	0	Haile Gerima
The Great British Baking Show	0	Andy Devonshire
The Starling	0	Theodore Melfi
Vendetta: Truth, Lies and The Mafia	0	nan
Bangkok Breaking	0	Kongkiat Komesiri
Je Suis Karl	0	Christian Schwochow
Confessions of an Invisible Girl	0	Bruno Garotti
Crime Stories: India Detectives	0	nan
Dear White People	0	nan
Europe's Most Dangerous Man: Otto Skorzeny in Spain	0	Pedro de Echave García
	1	Pablo Azorín Williams
Falsa identidad	0	nan
Intrusion	0	Adam Salky
Jaguar	0	nan
Monsters Inside: The 24 Faces of Billy Milligan	0	Olivier Megaton
Resurrection: Ertugrul	0	nan
Avvai Shanmughi	0	K.S. Ravikumar
Go! Go! Cory Carson: Chrissy Takes the Wheel	0	Alex Woo
	1	Stanley Moore
Jeans	0	S. Shankar
Love on the Spectrum	0	nan
Minsara Kanavu	0	Rajiv Menon
Grown Ups	0	Dennis Dugan
Dark Skies	0	Scott Stewart
Paranoia	0	Robert Luketic
Ankahi Kahaniya	0	Ashwiny Iyer Tiwari
	1	Abhishek Chaubey
	2	Saket Chaudhary
Chicago Party Aunt	0	nan
Sex Education	0	nan
Squid Game	0	nan
Tayo and Little Wizards	0	nan
The Father Who Moves Mountains	0	Daniel Sandu
The Stronghold	0	Cédric Jimenez
Angry Birds	0	nan
Birth of the Dragon	0	George Nolfi
Chhota Bheem	0	nan
He-Man and the Masters of the Universe	0	nan
Jaws	0	Steven Spielberg
Jaws 2	0	Jeannot Szwarc

Jaws 3	**0**	Joe Alves
Jaws: The Revenge	**0**	Joseph Sargent

dtype: object

```
df_new1=pd.DataFrame(df_new1.reset_index())
df_new1.rename(columns={0:'Director'}, inplace=True)
df_new1.drop(['level_1'], axis=1, inplace=True)
df_new1.head()
```

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

Next steps: Generate code with df_new1 | View recommended plots | New interactive sheet

```
# unnesting the directors column -
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:'Director'}, inplace=True)
df_new1.drop(['level_1'], axis=1, inplace=True)
df_new1.head()
```

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

Next steps: Generate code with df_new1 | View recommended plots | New interactive sheet

```
# unnesting the cast column -
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:'Cast'}, inplace=True)
df_new2.drop(['level_1'], axis=1, inplace=True)
df_new2.head()
```

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

Next steps: Generate code with df_new2 | View recommended plots | New interactive sheet

```
# unnesting the listed_in column -
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()
```

	title	Genre
0	Dick Johnson Is Dead	Documentaries
1	Blood & Water	International TV Shows
2	Blood & Water	TV Dramas
3	Blood & Water	TV Mysteries
4	Ganglands	Crime TV Shows

Next steps: Generate code with df_new3 | View recommended plots | New interactive sheet

```
df_new3.groupby(['Genre']).agg({'title':'nunique'}).reset_index().sort_values(by = 'title', ascending= False)
```

	Genre	title
16	International Movies	2752
12	Dramas	2427
7	Comedies	1674
17	International TV Shows	1351
10	Documentaries	869
0	Action & Adventure	859
34	TV Dramas	763
15	Independent Movies	756
4	Children & Family Movies	641
24	Romantic Movies	616
33	TV Comedies	581
41	Thrillers	577
8	Crime TV Shows	470
18	Kids' TV	451
11	Docuseries	395
22	Music & Musicals	375
25	Romantic TV Shows	370
14	Horror Movies	357
30	Stand-Up Comedy	343
23	Reality TV	255
3	British TV Shows	253
26	Sci-Fi & Fantasy	243
29	Sports Movies	219
2	Anime Series	176
28	Spanish-Language TV Shows	174
32	TV Action & Adventure	168
19	Korean TV Shows	151
6	Classic Movies	116
20	LGBTQ Movies	102
36	TV Mysteries	98
27	Science & Nature TV	92
37	TV Sci-Fi & Fantasy	84
35	TV Horror	75
1	Anime Features	71
9	Cult Movies	71
40	Teen TV Shows	69
13	Faith & Spirituality	65
39	TV Thrillers	57
21	Movies	57
31	Stand-Up Comedy & Talk Shows	56
5	Classic & Cult TV	28
38	TV Shows	16

```
dfx = df_new3[df_new3['Genre'].isin(['Comedies', 'TV Comedies', 'Stand-Up Comedy', 'Stand-Up Comedy & Talk Shows'])]
len(dfx)
```

```
2654
```

```
#unnesting the country column
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

Next steps: Generate code with df_new4 | View recommended plots | New interactive sheet

```
df_new4.groupby(['Country']).agg({'title':'nunique'}).reset_index().sort_values(by = 'title', ascending= False)
```

	Country	title
119	United States	3689
47	India	1046
127	nan	831
117	United Kingdom	804
22	Canada	445
...	...	...
70	Mongolia	1
101	Somalia	1
36	Ethiopia	1
15	Botswana	1
88	Poland,	1

128 rows × 2 columns

```
df_new4["Country"].value_counts().sort_values(ascending = False)[:10]
```

Country	count
United States	3689
India	1046
nan	831
United Kingdom	804
Canada	445
France	393
Japan	318
Spain	232
South Korea	231
Germany	226

dtype: int64

```
# merging the unnested director data with unnested cast 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 cast by Unknown
df_new["Cast"].replace(['nan'], ['Unknown'], inplace = True)
df_new["Country"].replace(['nan'], [np.nan], inplace = True)
df_new["Director"].replace(['nan'], ['Unknown'], inplace = True)

df_new.head()
```

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

```
df_final = df_new.merge(df[['show_id', 'type', 'title', 'date_added',
                        'release_year', 'rating', 'duration']] , on = ["title"] , how = 'left')
df_final.head()
```

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

```
df_final.isnull().sum() * 100 / df_final.shape[0]
```

	0
title	0.000000
Cast	0.000000
Director	0.000000
Genre	0.000000
Country	5.889866
show_id	0.000000
type	0.000000
date_added	0.078221
release_year	0.000000
rating	0.033170
duration	0.001485

dtype: float64

```
df_final.dropna(subset= ['duration', 'rating', 'release_year'] , axis = 0 , inplace = True)

df_final["Country"].value_counts()
```

Country	count
United States	59346
India	22814
United Kingdom	12945
Japan	8635
France	8254
...	...
Palestine	2
Kazakhstan	1
Nicaragua	1
United States,	1
Uganda	1

127 rows × 1 columns

dtype: int64

```
df_final["Country"].value_counts().idxmax()
```

```
'United States'
```

```
df_final.fillna(df_final["Country"].value_counts().idxmax(), inplace= True)
```

```
df_final['Year_added'] = df_final["date_added"].str.split(',' , expand = True)[1]
df_final['Year_added']
```

	Year_added
0	2021
1	2021
2	2021
3	2021
4	2021
...	...
201986	2019
201987	2019
201988	2019
201989	2019
201990	2019

201921 rows × 1 columns

dtype: object

```
df_movies = df_final[df_final['type']=='Movie']
df_shows = df_final[df_final['type'] == 'TV Show']
```

Ques. 4.1 - Most of the movies available to watch are rated as?

a. G b. PG c. MA d. PG-13

ANS - MA

```
df_movies.groupby(['rating']).agg({'title' : 'nunique'}).reset_index().sort_values(by =["title"] , ascending = False)[:5]
```

	rating	title
8	TV-MA	2062
6	TV-14	1427
5	R	797
9	TV-PG	540
4	PG-13	490

Ques. 4.2 - Most of the TV shows available to watch are rated as?

a. TV-Y b. TV-PG c. TV-14 d. TV-MA

ANS - TV-MA

```
df_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values(by = ["title"], ascending=False)[:5]
```

	rating	title
4	TV-MA	1145
2	TV-14	733
5	TV-PG	323
7	TV-Y7	195
6	TV-Y	176

```
df_movies.groupby(["duration"]).agg({"title":"nunique"}).reset_index().sort_values(by = ["title"], ascending=False)[:5]
```

	duration	title
195	90 min	152
198	93 min	146
199	94 min	146
202	97 min	146
196	91 min	144

Ques. 5.1 - What is the average duration of movies present on Netflix? (Choose the nearest available option)

a. 80 mins b. 110 mins c. 135 mins d. 150 mins

ANS - 110 mins

```
df_movies["duration"].str.split(expand=True)
```

	0	1
0	90	min
159	91	min
160	91	min
161	91	min
162	91	min
...	...	...
201986	111	min
201987	111	min
201988	111	min
201989	111	min
201990	111	min

145831 rows × 2 columns

```
df_movies["duration"] = df_movies["duration"].str.split(expand=True)[0]
df_movies['duration'].astype(int).mean().round()
```

```
<ipython-input-79-f01d56288e8b>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df_movies["duration"] = df_movies["duration"].str.split(expand=True)[0]
107.0
```

Ques. 5.2 - What is the average duration of TV shows present on Netflix? (Choose the nearest available option)

a. 2 Season b. 3 Seasons c. 4 Seasons d. 5 Seasons

ANS - 2 Seasons

```
df_shows["duration"]=df_shows["duration"].str.split(expand= True)[0]
df_shows["duration"].astype(int).mean().round()
```

```
<ipython-input-80-7db1471b5c32>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df_shows["duration"]=df_shows["duration"].str.split(expand= True)[0]
2.0
```

```
df_movies['_duration']=df_movies['duration'].str.split(expand=True)[0]
df_movies['_duration']=df_movies['_duration'].astype(int)
```

```
<ipython-input-81-8542cf20ab3a>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df_movies['_duration']=df_movies['duration'].str.split(expand=True)[0]
<ipython-input-81-8542cf20ab3a>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df_movies['_duration']=df_movies['_duration'].astype(int)
```

```
bins = [1, 50 , 80 , 100, 120 , 150, 200, 315]
labels = ['1-50' , '50-80' , '80-100', '100-120','120-150', '150-200', '200-315']

df_movies['duration_bins'] = pd.cut(df_movies['_duration'] , bins = bins , labels = labels)
df_movies.head()
```

```
<ipython-input-82-493208f739ae>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  df_movies['duration_bins'] = pd.cut(df_movies['_duration'] , bins = bins , labels = labels)
```

	title	Cast	Director	Genre	Country	show_id	type	date_added	release_year	rating	duration	Year_added	_d
0	Dick Johnson Is Dead	Unknown	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90	2021	
159	My Little Pony: A New Generation	Vanessa Hudgens	Robert Cullen	Children & Family Movies	United States	s7	Movie	September 24, 2021	2021	PG	91	2021	
160	My Little Pony: A New Generation	Vanessa Hudgens	José Luis Ucha	Children & Family Movies	United States	s7	Movie	September 24, 2021	2021	PG	91	2021	

```
df_movies.groupby(['duration_bins']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],
                                                                                         ascending=False)[:5]
```

```
<ipython-input-84-c1beda0b249d>:1: FutureWarning: The default of observed=False is deprecated and will be changed to True in a futur
  df_movies.groupby(['duration_bins']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],
```

	duration_bins	title
2	80-100	2221
3	100-120	1671
4	120-150	897
1	50-80	806
0	1-50	286

Ques. 5.4 - The duration of most of the TV shows present on Netflix is..

a. 1 Season b. 3 Seasons c. 7 Seasons d. 9 Seasons

ANS - 1 Season

```
df_shows.groupby(['duration']).agg({'title':'nunique'}).reset_index().sort_values(by=['title'],ascending=False)[:5]
```

	duration	title
0	1	1791
7	2	425
8	3	199
9	4	95
10	5	65

```
df_final.head()
```

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

```
df_final1  = df_final.copy(deep = True)
```

Univariate Analysis separately for shows and movies -

> Add blockquote

```
df_shows.groupby(['Genre']).agg({'title':'nunique'}).reset_index().sort_values(by = ['title'], ascending=False)[:10]
```

	Genre	title
5	International TV Shows	1350
15	TV Dramas	763
14	TV Comedies	580
3	Crime TV Shows	470
6	Kids' TV	450
4	Docuseries	395
9	Romantic TV Shows	370
8	Reality TV	255
1	British TV Shows	253
0	Anime Series	175

```
df_movies.groupby(['Genre']).agg({'title':'nunique'}).reset_index().sort_values(by = ['title'], ascending=False)[:10]
```

	Genre	title
11	International Movies	2752
7	Dramas	2426
4	Comedies	1674
6	Documentaries	869
0	Action & Adventure	859
10	Independent Movies	756
2	Children & Family Movies	641
15	Romantic Movies	616
19	Thrillers	577
14	Music & Musicals	375

```
df_shows.groupby(['Country']).agg({'title':'nunique'}).reset_index().sort_values(by = ['title'], ascending=False)[:10]
```

	Country	title
63	United States	1329
62	United Kingdom	272
30	Japan	198
52	South Korea	170
8	Canada	126
19	France	90
25	India	84
57	Taiwan	70
2	Australia	65
53	Spain	61

```
df_movies.groupby(['Country']).agg({'title':'nunique'}).reset_index().sort_values(by = ['title'], ascending=False)[:10]
```

	Country	title
114	United States	3187
43	India	962
112	United Kingdom	532
20	Canada	319
34	France	303
36	Germany	182
100	Spain	171
51	Japan	119
23	China	114
65	Mexico	111

```
df_shows.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)[:10]
```

	rating	title
4	TV-MA	1145
2	TV-14	733
5	TV-PG	323
7	TV-Y7	195
6	TV-Y	176
3	TV-G	94
0	NR	5
1	R	2
8	TV-Y7-FV	1

```
df_movies.groupby(['rating']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)[:10]
```

	rating	title
8	TV-MA	2062
6	TV-14	1427
5	R	797
9	TV-PG	540
4	PG-13	490
3	PG	287
11	TV-Y7	139
10	TV-Y	131
7	TV-G	126
2	NR	75

```
df_shows.groupby(['duration']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'], ascending=False)[:10]
```

	duration	title
0	1	1791
7	2	425
8	3	199
9	4	95
10	5	65
11	6	33
12	7	23
13	8	17
14	9	9
1	10	7

```
df_movies.groupby(['duration']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'], ascending=False)[:10]
```

	duration	title
195	90	152
198	93	146
199	94	146
202	97	146
196	91	144
200	95	137
201	96	130
197	92	129
3	102	122
203	98	120

Ques. 8.1 - Find out which of these actors/actresses has starred in a maximum number of Netflix movies?

a. Anupam Kher b. Shah Rukh Khan c. Salman Khan d. Amitabh Bachchan

ANS - Anupam Kher

```
df_movies.head()
```

	title	Cast	Director	Genre	Country	show_id	type	date_added	release_year	rating	duration	Year_added	_d
0	Dick Johnson Is Dead	Unknown	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020	PG-13	90	2021	
159	My Little Pony: A New Generation	Vanessa Hudgens	Robert Cullen	Children & Family Movies	United States	s7	Movie	September 24, 2021	2021	PG	91	2021	
160	My Little Pony: A New Generation	Vanessa Hudgens	José Luis Ucha	Children & Family Movies	United States	s7	Movie	September 24, 2021	2021	PG	91	2021	

```
df_cast = df_movies.groupby(['Cast']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)
df_cast= df_cast[df_cast["Cast"]!= 'Unknown']
df_cast
```

	Cast	title
2102	Anupam Kher	42
21774	Shah Rukh Khan	35
17187	Naseeruddin Shah	32
18058	Om Puri	30
638	Akshay Kumar	30
...	...	...
9990	Jacob Davich	1
9989	Jacob Craner	1
9988	Jacob Buster	1
9987	Jacob Blair	1
25944	Ṣọpẹ́ Dìrísù	1

25944 rows × 2 columns

Next steps: Generate code with df_cast | View recommended plots | New interactive sheet

Ques. 8.2 - Find out which of these actors/actresses has starred in a maximum number of Netflix TV shows?

a. Grey Griffin b. Yuki Kaji c. Takahiro Sakurai d. Hiroshi Kamiya

ANS - Takahiro Sakurai

```
df_cast = df_shows.groupby(['Cast']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],ascending=False)
df_cast= df_cast[df_cast["Cast"]!= 'Unknown']
df_cast.head(10)
```

	Cast	title
13217	Takahiro Sakurai	25
14568	Yuki Kaji	19
2873	Daisuke Ono	17
6797	Junichi Suwabe	17
252	Ai Kayano	16
14552	Yuichi Nakamura	16
6754	Jun Fukuyama	15
14484	Yoshimasa Hosoya	15
3126	David Attenborough	14
5089	Hiroshi Kamiya	13

Next steps: Generate code with df_cast | View recommended plots | New interactive sheet

Ques. 9.1 - Which of the following TV show directors is most popular over the platform?

a. Abhishek Chaubey b. Oliver Stone c. Peter McDonnell d. Ken Burns

ANS - Ken Burns

```
df_director=df_shows.groupby(['Director']).agg({"title":"nunique"}).reset_index().sort_values(by=['title'],
                                                                                                ascending=False)
df_director=df_director[df_director['Director']!='Unknown']
df_director.head(10)
```

	Director	title
146	Ken Burns	3
8	Alastair Fothergill	3
259	Stan Lathan	2
128	Joe Berlinger	2
100	Hsu Fu-chun	2
84	Gautham Vasudev Menon	2
103	Iginio Straffi	2
168	Lynn Novick	2
251	Shin Won-ho	2
235	Rob Seidenglanz	2

Next steps: Generate code with df_director | View recommended plots | New interactive sheet

Ques. 9.2 - Which of the following movie directors is most popular over the platform?

a. Shannon Hartman b. Anurag Kashyap c. Rajiv Chilaka d. Marcus Raboy

ANS - Rajiv Chilaka

```
df_director = df_movies.groupby(['Director']).agg({'title':'nunique'}).reset_index().sort_values(by = ['title'],ascending= False)
df_director = df_director[df_director['Director']!= 'Unknown']
df_director.head()
```

	Director	title
3580	Rajiv Chilaka	22
1816	Jan Suter	21
3631	Raúl Campos	19
4259	Suhas Kadav	16
2737	Marcus Raboy	15

Next steps: Generate code with df_director | View recommended plots | New interactive sheet

Univariate Analysis separately for shows in Japan -

```
df_japan_shows = df_shows[df_shows['Country']=='Japan']
df_japan_shows.head()
```

	title	Cast	Director	Genre	Country	show_id	type	date_added	release_year	rating	duration	Year_added
1754	Yowamushi Pedal	Daiki Yamashita	Unknown	Anime Series	Japan	s77	TV Show	September 14, 2021	2013	TV-14	1	2021
1755	Yowamushi Pedal	Daiki Yamashita	Unknown	International TV Shows	Japan	s77	TV Show	September 14, 2021	2013	TV-14	1	2021
1756	Yowamushi Pedal	Kohsuke Toriumi	Unknown	Anime Series	Japan	s77	TV Show	September 14, 2021	2013	TV-14	1	2021

Ques. 10.1 - What kind of Genre do the Japanese usually prefer watching?

a. Anime b. Comedy c. Action d. Horror

ANS - Anime

```
df_japan_shows.groupby(['Genre']).agg({'title':'nunique'}).reset_index().sort_values(by= ['title'],ascending=False)[:10]
```

	Genre	title
4	International TV Shows	150
0	Anime Series	142
5	Kids' TV	29
7	Romantic TV Shows	21
11	TV Dramas	21
2	Crime TV Shows	16
16	Teen TV Shows	14
10	TV Comedies	10
6	Reality TV	9
15	TV Thrillers	6

Univariate Analysis separately for shows in South Korea -

Ques. 10.2 - What kind of Genre do Koreans usually prefer watching?

a. Drama b. Romantic c. Thriller d. Adventure

ANS - Romantic

```
df_korea_shows=df_shows[(df_shows['Country']=='South Korea')]
```

```
df_korea_shows.groupby(['Genre']).agg({'title':'nunique'}).reset_index().sort_values(by = ['title'], ascending = False)[:10]
```

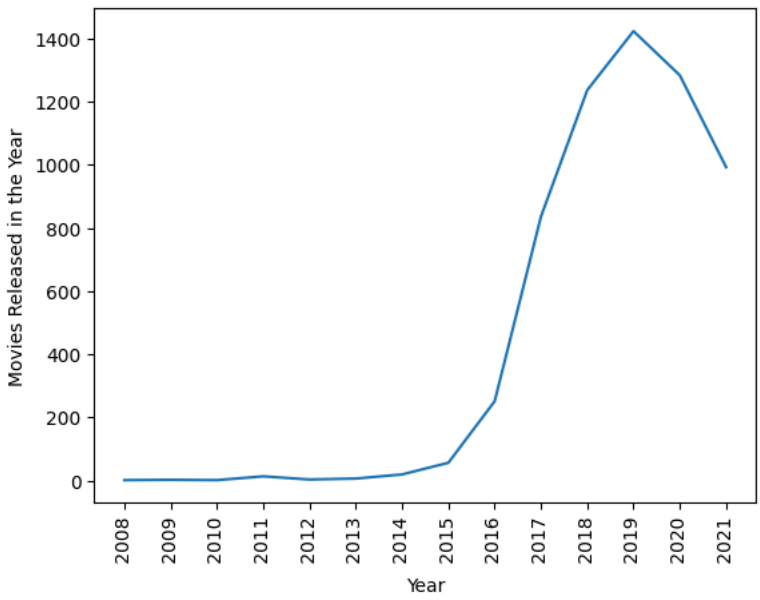
	Genre	title
1	International TV Shows	152
3	Korean TV Shows	132
5	Romantic TV Shows	77
9	TV Dramas	38
0	Crime TV Shows	24
8	TV Comedies	19
2	Kids' TV	16
7	TV Action & Adventure	9
4	Reality TV	4
6	Stand-Up Comedy & Talk Shows	4

Ques. 11.1 - From which year afterwards does the number of movies being added on Netflix start dropping drastically?

a. After 2020 b. After 2018 c. After 2021 d. After 2019

ANS - After 2019

```
df_year= df_movies.groupby(['Year_added']).agg({'title':'nunique'}).reset_index()
sns.lineplot(data = df_year , x = 'Year_added', y = 'title')
plt.xlabel("Year")
plt.ylabel("Movies Released in the Year")
plt.xticks(rotation=90)
plt.show()
```

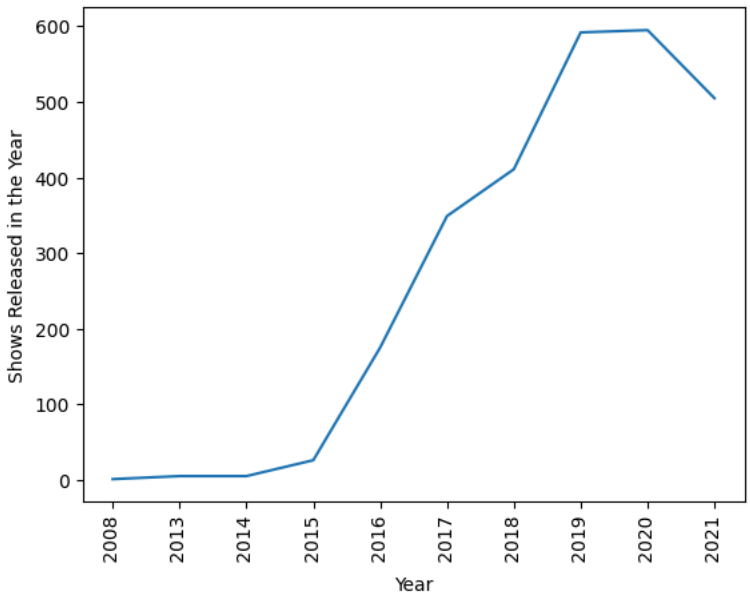


Ques. 11.2 - From which year afterwards does the number of TV shows being added on Netflix start dropping drastically?

a. After 2019 b. After 2020 c. After 2021 d. After 2018

ANS - After 2020

```
df_year= df_shows.groupby(['Year_added']).agg({'title':'nunique'}).reset_index()
sns.lineplot(data = df_year , x = 'Year_added', y = 'title')
plt.xlabel("Year")
plt.ylabel("Shows Released in the Year")
plt.xticks(rotation=90)
plt.show()
```




```
df_usa_shows = df_shows[df_shows['Country']=='United States']
df_ind_shows = df_shows[df_shows['Country']=='India']
```

```
df_usa_movies = df_movies[df_movies['Country']=='United States']
df_ind_movies = df_movies[df_movies['Country']=='India']
```

Ques. 12.1 - Which is the most popular actor-director pair for TV shows across the United States?

a. Marty Adams & Eli Roth b. Mark Sheppard & Phil Sgriccia c. Dave Chapelle & Stan Latham d. Marisol Nicoles & Rob Seidenglanz

ANS - Dave Chappelle and Stan Lathan