```
url='https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.
```

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

```
import plotly.express as px
```

```
df = pd.read_csv(url)
```

```
#Getting some basic information about dataset
df.head(2)
```

	show_id	type	title	director	cast	country	date_added	release_year	ratin
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-1
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-M

```
# To get column names of the dataset:
df.columns
```

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

As we can see from df.head() that the data consists of columns such as

- Show_id
- Type
- Title
- Director
- Cast

- Country
- Date added
- Release_year
- Rating
- Duration
- Listed_in
- Description

```
# To get dimensions of the data
df.shape
```

```
(8807, 12)
```

```
# To get total number of elements in dataset
df.size
```

```
105684
```

```
# To get datatypes of each columns:
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
```

```
# To get information of every column we will going to use
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
```

```
# Getting statistical report of dataset
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

GETTING STARTED WITH EXPLORATORY DATA ANALYSIS

```
# Dropping unused column, as we can in our dataset show_id column works similar to index colu
df.drop('show_id',axis=1,inplace=True)
```

Objective 1: To find Duplicates record and If any, Removal of Duplicate Records

```
# df['cast'].replace(np.nan,'No Data',inplace=True)
# df['director'].replace(np.nan,'No Data',inplace=True)
# df.dropna(inplace = True)
# df.drop_duplicates(inplace = True)
```

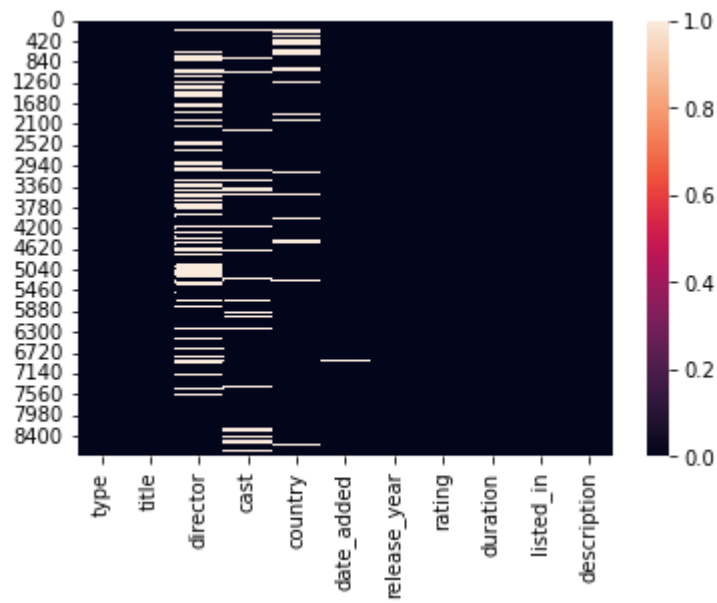
Objective 2: Finding Null Values

```
#The below code will summarize and return total no of null values in each columns
df.isnull().sum()
```

```
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
```

```
sns.heatmap(df.isnull())
#White spaces represents null values
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3052ec5350>
```



From above codes we can conclude that director column as most amount of null values and other than that cast,country column consists of larger no of null values.

Question: In which year Highest number of Content was released by Netflix?

```
df.dtypes
```

```
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
```

From the datatypes we can see that the datatype of date_added is object instead of datetime.

```
# Type casting data type of date_added
df['Released_Date'] = pd.to_datetime(df['date_added'])
```

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

```
array(['United States', 'South Africa', nan, 'India',
       'United States, Ghana, Burkina Faso, United Kingdom, Germany, Ethiopia',
       'United Kingdom', 'Germany, Czech Republic', 'Mexico', 'Turkey',
       'Australia', 'United States, India, France', 'Finland',
       'China, Canada, United States',
       'South Africa, United States, Japan', 'Nigeria', 'Japan',
       'Spain, United States', 'France', 'Belgium',
       'United Kingdom, United States', 'United States, United Kingdom',
       'France, United States', 'South Korea', 'Spain',
       'United States, Singapore', 'United Kingdom, Australia, France',
       'United Kingdom, Australia, France, United States',
       'United States, Canada', 'Germany, United States',
       'South Africa, United States', 'United States, Mexico',
       'United States, Italy, France, Japan',
       'United States, Italy, Romania, United Kingdom',
       'Australia, United States', 'Argentina, Venezuela',
       'United States, United Kingdom, Canada', 'China, Hong Kong',
       'Russia', 'Canada', 'Hong Kong', 'United States, China, Hong Kong',
       'Italy, United States', 'United States, Germany',
       'United Kingdom, Canada, United States', ', South Korea',
       'Ireland', 'India, Nepal',
       'New Zealand, Australia, France, United States', 'Italy',
       'Italy, Brazil, Greece', 'Argentina', 'Jordan', 'Colombia',
       'United States, Japan', 'Belgium, United Kingdom',
       'Switzerland, United Kingdom, Australia', 'Israel, United States',
       'Canada, United States', 'Brazil', 'Argentina, Spain', 'Taiwan',
       'United States, Nigeria', 'Bulgaria, United States',
       'Spain, United Kingdom, United States', 'United States, China',
       'United States, France',
       'Spain, France, United Kingdom, United States',
       ', France, Algeria', 'Poland', 'Germany',
       'France, Israel, Germany, United States, United Kingdom',
       'New Zealand', 'Saudi Arabia', 'Thailand', 'Indonesia',
       'Egypt, Denmark, Germany', 'United States, Switzerland',
       'Hong Kong, Canada, United States', 'Kuwait, United States',
```

```
       'France, Canada, United States, Spain',
       'France, Netherlands, Singapore', 'France, Belgium',
       'Ireland, United States, United Kingdom', 'Egypt', 'Malaysia',
       'Israel', 'Australia, New Zealand', 'United Kingdom, Germany',
       'Belgium, Netherlands', 'South Korea, Czech Republic',
       'Australia, Germany', 'Vietnam', 'United Kingdom, Belgium',
       'United Kingdom, Australia, United States',
       'France, Japan, United States',
       'United Kingdom, Germany, Spain, United States',
       'United Kingdom, United States, France, Italy',
       'United States, Germany, Canada',
       'United States, France, Italy, United Kingdom',
       'United States, United Kingdom, Germany, Hungary',
       'United States, New Zealand', 'Sweden', 'China', 'Lebanon',
       'Romania', 'Finland, Germany', 'Lebanon, Syria', 'Philippines',
       'Iceland', 'Denmark', 'United States, India',
       'Philippines, Singapore, Indonesia',
       'China, United States, Canada', 'Lebanon, United Arab Emirates',
       'Canada, United States, Denmark', 'United Arab Emirates',
       'Mexico, France, Colombia', 'Netherlands',
       'Germany, United States, France', 'United States, Bulgaria',
       'United Kingdom, France, Germany, United States',
       'Norway, Denmark', 'Syria, France, Lebanon, Qatar',
```

As we can see that the content was released in multiple countries that is leading to many junk data arriving and giving wrong analysis, that's why melting the data into individual countries

```
countries = df.copy()
countries = pd.concat([countries,df['country'].str.split(',',expand=True)],axis=1)
countries = countries.melt(id_vars=['type','title'],value_vars = range(12),value_name = 'Coun
countries = countries[countries['Country'].notna()]
```

Similarly with the casts as the casts are merged in a list and unable to identify the total casts required for a show, so seperating the casts count from the column.

```
df['cast_count'] = df['cast'].str.split(',')
df = df[df['cast_count'].notna()]
df['cast_count'] = df['cast_count'].apply(lambda x:len(x))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: 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_
  This is separate from the ipykernel package so we can avoid doing imports until
```

```
df.dtypes
```

```
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
Released_Date    datetime64[ns]
cast_count                int64
dtype: object
```

```
#As we can see the datatype is changed to datetime64 and a new column is created.
# Now drop the previous column
df.drop('date_added',axis=1,inplace=True)
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:4913: SettingWithCopyWarning
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
  errors=errors,
```

Question:

Which year Netflix was releasing highest number of content?

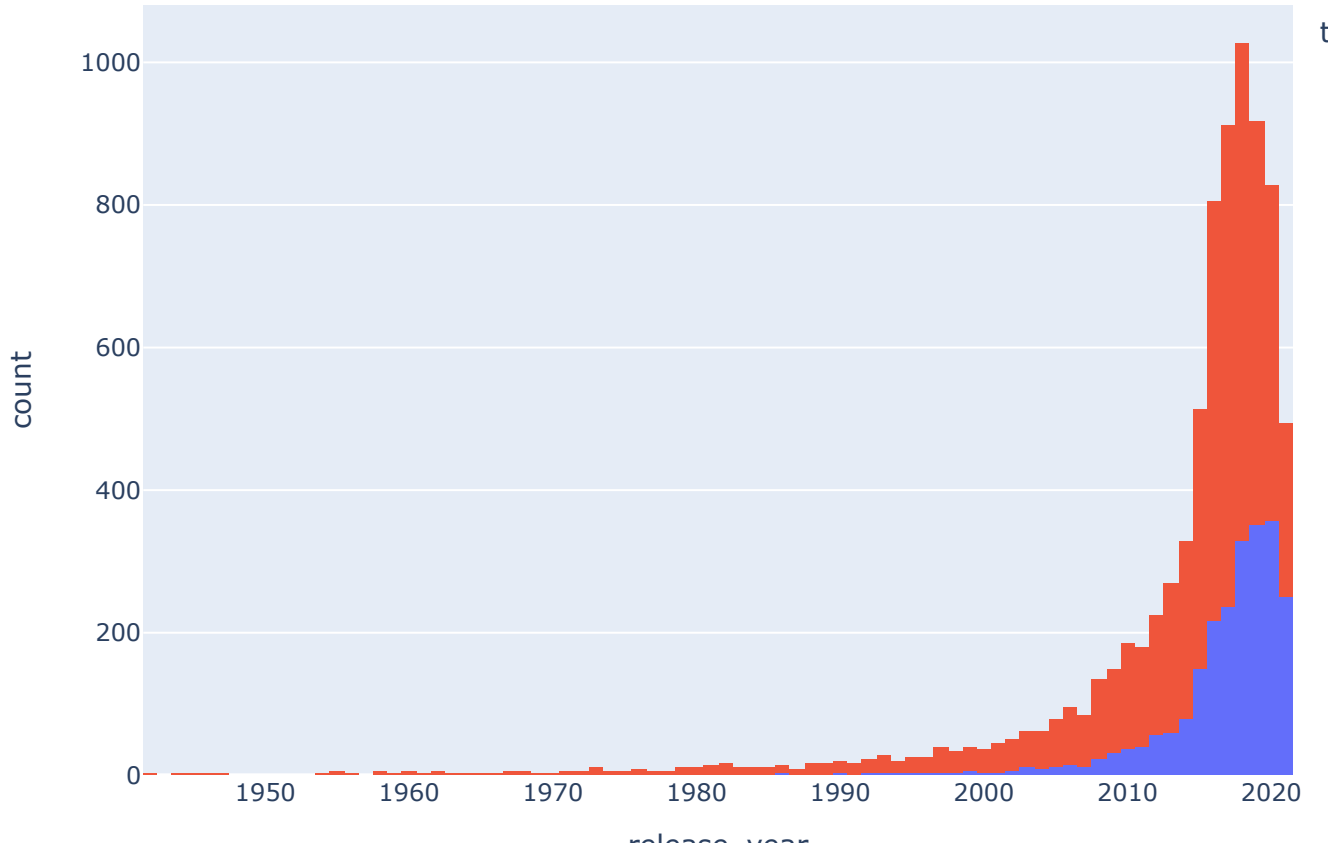
```
year_releases= df['release_year'].value_counts()
year_releases.head()
```

```
2018    1026
2019     917
2017     912
2020     827
2016     804
Name: release_year, dtype: int64
```

```
# Hence from above observation we can see that '2018' was peak year for content releasing for
```

```
# year_releases.plot(kind = 'bar')
# plt.show()

px.histogram(df,x='release_year',color='type')
```




```
# From above histogram we can conclude that 2018 was peak year with 380 and 767 Movies and TV
```

Question:

Display Total Number of Movies and Total Series till date.

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

```
array(['TV Show', 'Movie'], dtype=object)
```

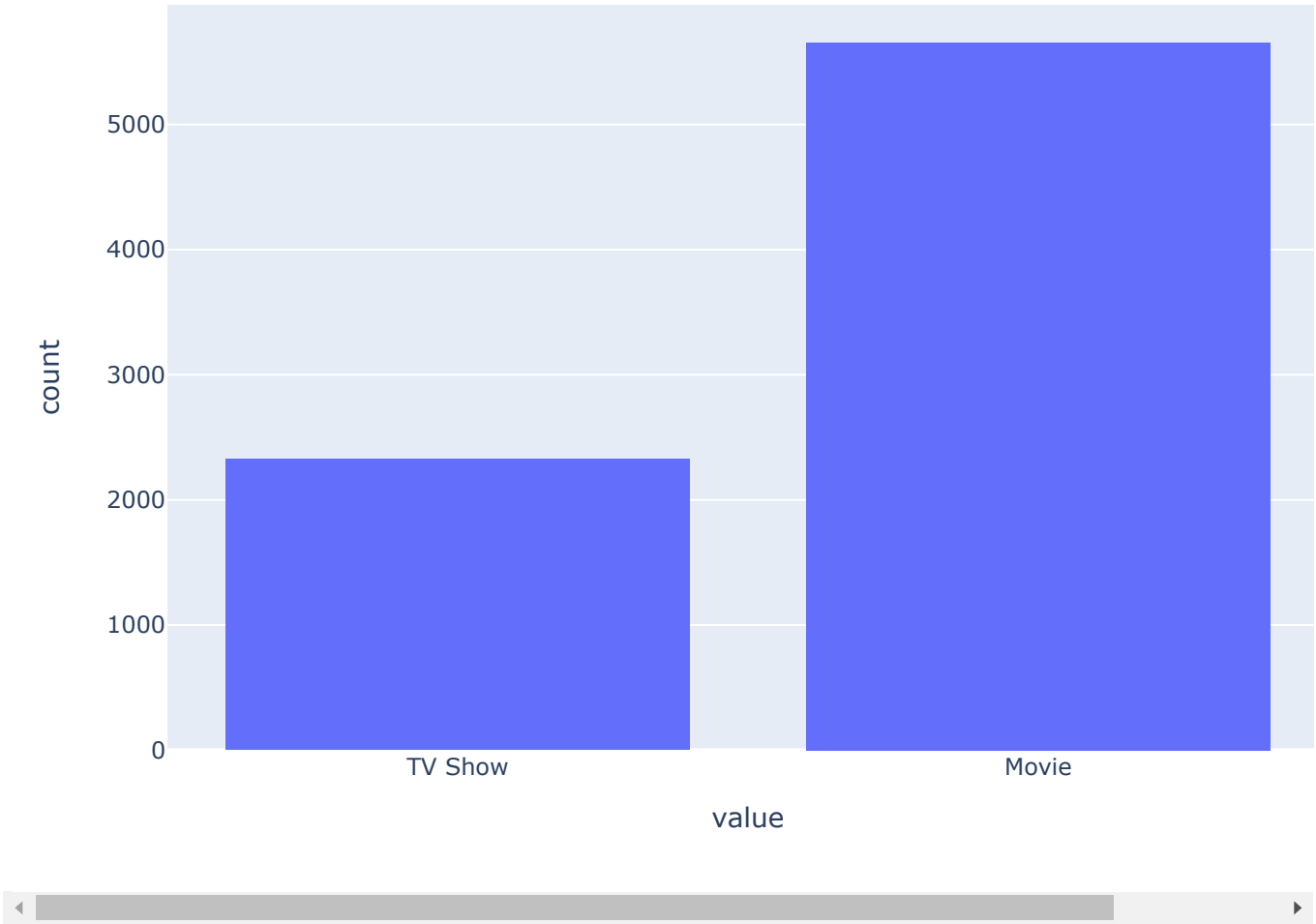
```
# Hence from above we can conclude that there is only 2 categories in which all content is ca
```

```
df.groupby('type').type.count()
```

```
type
Movie      5656
TV Show    2326
Name: type, dtype: int64
```

```
# Hence from above result we can see that the number of Total Movies releases is greater than
```

```
px.histogram(df['type'])
```




```
# From above graph, we can conclude that Netflix mainly focused on Movie release than TV Show
#across worlds but TV-Shows are mainly produced by Netflix itself. Movies culture in world is
# becoming popular. Hence the difference we can see by above graph.
```

Question To show trend of Movies and TV shows across years.

```
sns.histplot(data=df,x = 'release_year',kde=True,hue = 'type')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f304e1bcf90>
```

From above histplot we can conclude that the from beginning Netflix produces or supports Movies more rather than TV shows that's the trendline of Movies is above as compare to trendline of TV shows.

So Netflix has produced around 972 TV-Shows and Movies for India Viewers. In which 893 are Movies and 79 are shows.

Question:

Which season Netflix releases more content?

```
df['Released_Month'] = df['Released_Date'].dt.month.fillna(0)
```

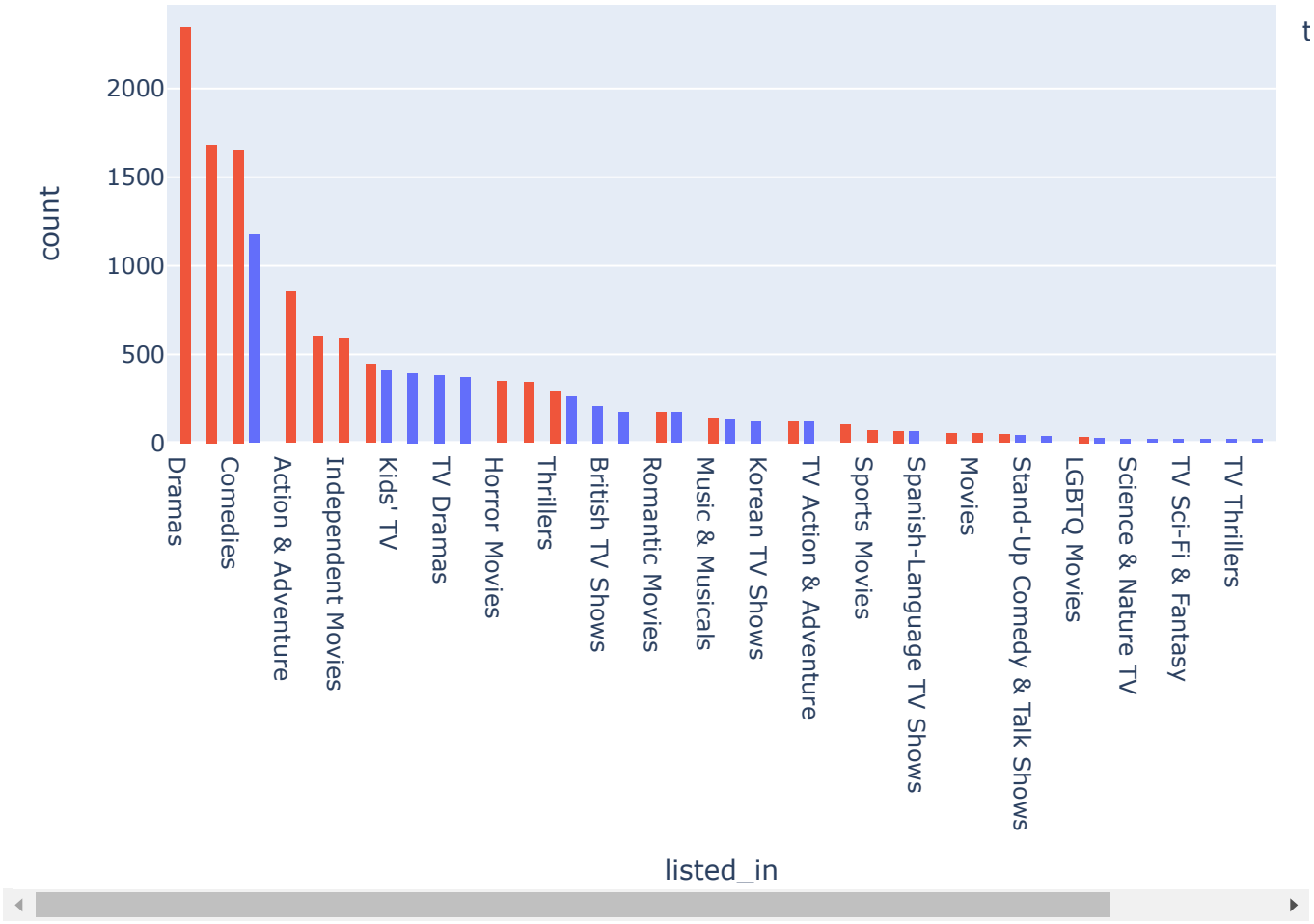
```
px.histogram(df,x='Released_Date',color='Released_Month')
```

Question:

What kind of genre is mostly made/released by Netflix in TV Shows and Movies?

```
genre = df.copy()
genre = pd.concat([genre,df['listed_in'].str.split(',',expand=True)],axis=1)
genre = genre.melt(id_vars = ['type','title'],value_vars = range(2),value_name = 'listed_in')
genre = genre[genre['listed_in'].notna()]
genre['listed_in'] = genre['listed_in'].str.strip()
```

```
px.histogram(genre,x = 'listed_in',color='type',barmode='group').update_xaxes(categoryorder =
```



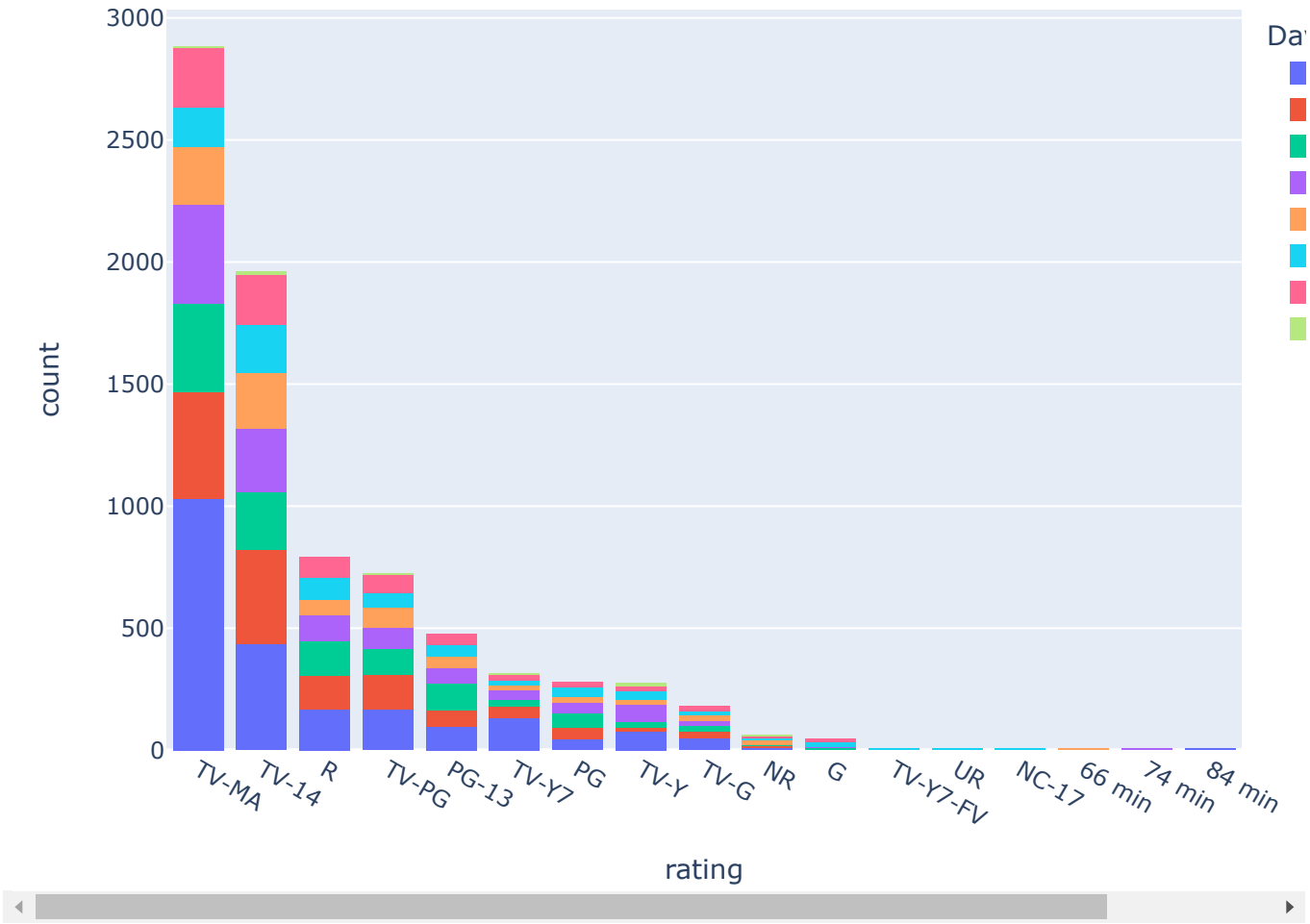
As we can see most of the content available on Netflix is regarding movies so the top 3 content's genre are related with movies only, this shows domination of Movies over TV Shows in Netflix.

As we can see Dramas, International and Comedies movie genre are most famous/popular content in Netflix.

Talking about TV Shows popular content are International, Kids TV and Crime TV Shows.

Double-click (or enter) to edit

```
df['Day'] = df['Released_Date'].dt.day_name().fillna(0)
px.histogram(df,x = 'rating',color='Day').update_xaxes(categoryorder = 'total descending')
```




```
df.head(1)
```

type	title	director	cast	country	release_year	rating	duration	listed_in
			Ama					

```
df_US_content.head(1)
```

	index	type	title	director	cast	country	release_year	rating	duration	li
0	9	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	2021	PG-13	104 min	Co

Question:

Top content absorbing countries in world.

```
px.histogram(countries,'Country',color='type' ).update_xaxes(categoryorder = 'total descendin
```

From Above graph we can see the top content absorbing countries are:

1. United States with 2752 Movies and 938 TV Shows content releasing till now
2. India with 962 Movies and 84 TV Shows.
3. United Kingdom 534 Movies and 272 TV Shows.

We can also conclude that United Kingdom is on 2nd number in case of producing TV Shows

So Let's Focus on Top 2 Countries Content more.

```
df_US_content = df.loc[df['country'] == 'United States'].reset_index()
```

Question:

How much movies and TV Shows individually produced by Netflix US?

```
df_US_content.shape
df_US_content['type'].value_counts()
```

```
Movie      1864
TV Show     624
Name: type, dtype: int64
```

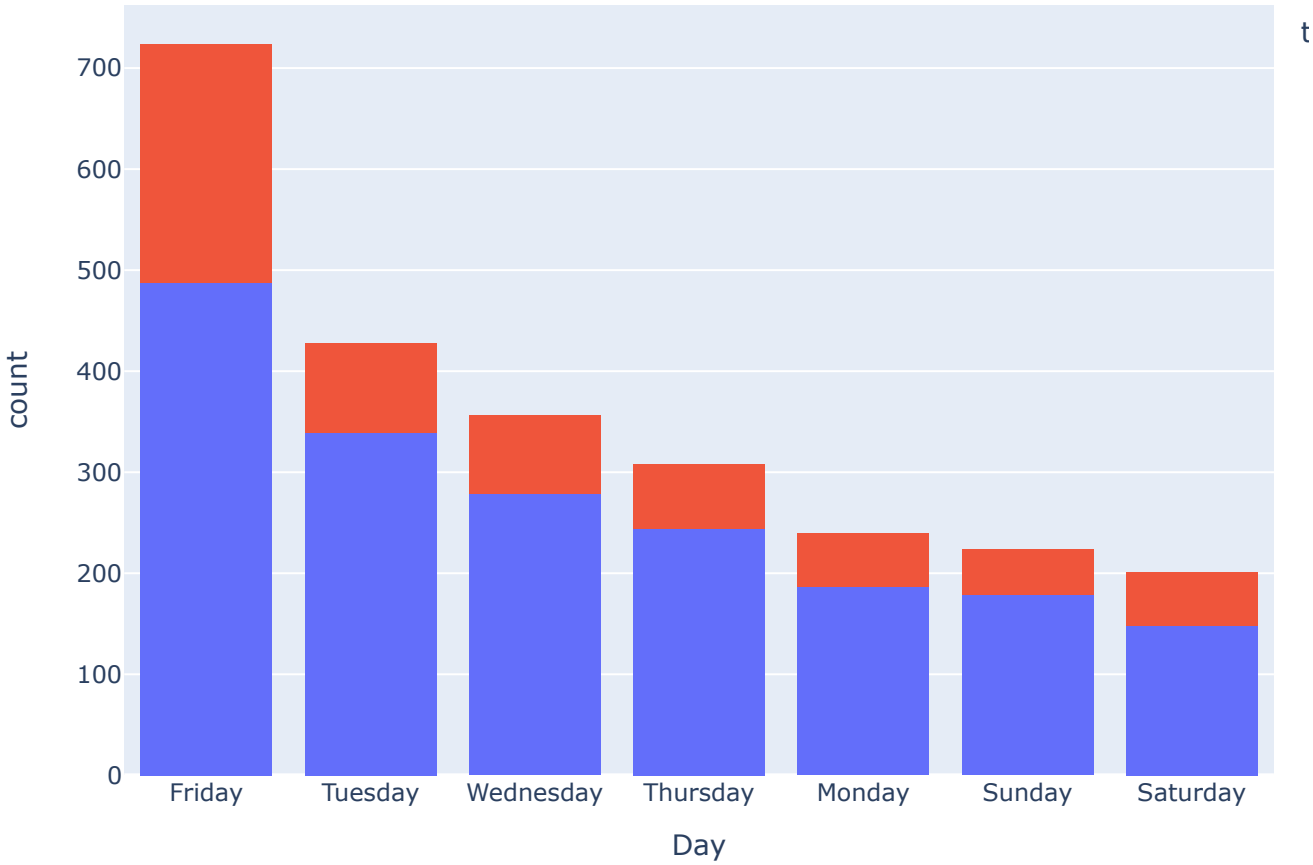
Question:

On what day in US does Netflix released most of its content

```
df_US_content['Day'].value_counts()
```

```
Friday       724
Tuesday      428
Wednesday    357
Thursday     308
Monday       240
Sunday       224
Saturday     201
Name: Day, dtype: int64
```

```
px.histogram(data_frame=df_US_content,x='Day',color = 'type').update_xaxes(categoryorder ='to
```



Most Content on Netflix is released on Friday and Tuesday, as after Friday a long weekend arrives which helps Netflix to increase viewerships and TRP.

This is the Success story for most of the Netflix Shows.

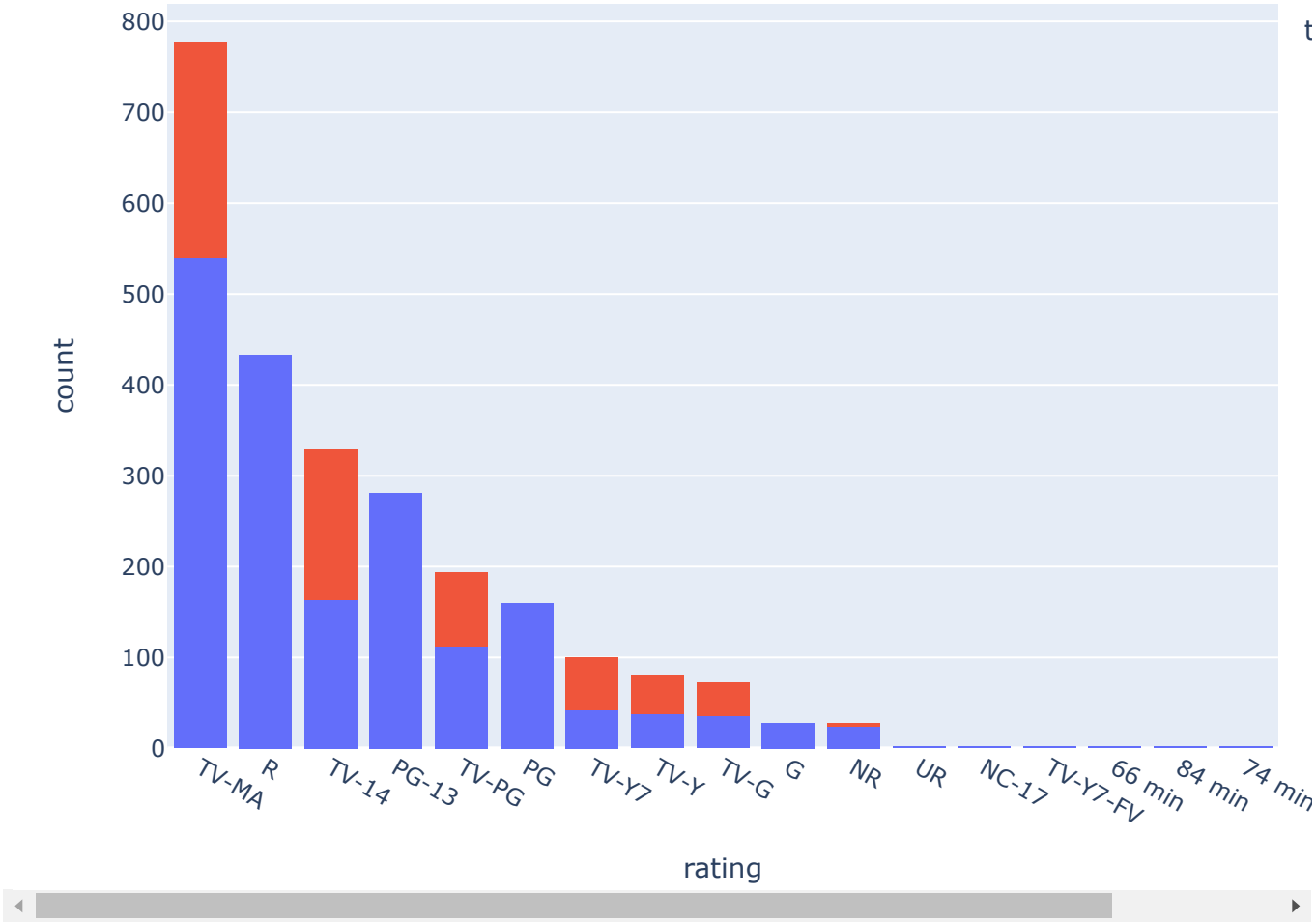
Question:

What ratings of TV Shows and Movies generally Netflix produces?

```
df_US_content['rating'].value_counts().head(10)
```

```
TV-MA    778
R        433
TV-14    329
PG-13    281
TV-PG    193
PG       160
TV-Y7    100
TV-Y      81
TV-G      73
G         28
Name: rating, dtype: int64
```

```
px.histogram(data_frame=df_US_content,x='rating',color = 'type').update_xaxes(categoryorder =
```



Netflix generally produces most of its content TV Shows of ratings TV-MA due to which we can depicts that Netflix produces TV Shows mostly for 18+ as its audience most of the viewers and targeted audiences are 18+.

Other than this Netflix movies are most of them are Rated R, which again depicts there targeted audience and viewers are mostly 18+ and matured.

Question:

What is the average number of cast is required in a content by Netflix?

```
df.head(2)
```

	type	title	director	cast	country	release_year	rating	duration	liste
1	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane,	South Africa	2021	TV-MA	2 Seasons	Internat TV Sh TV Dra Myst
	TV		Julien	Tracy					Sn

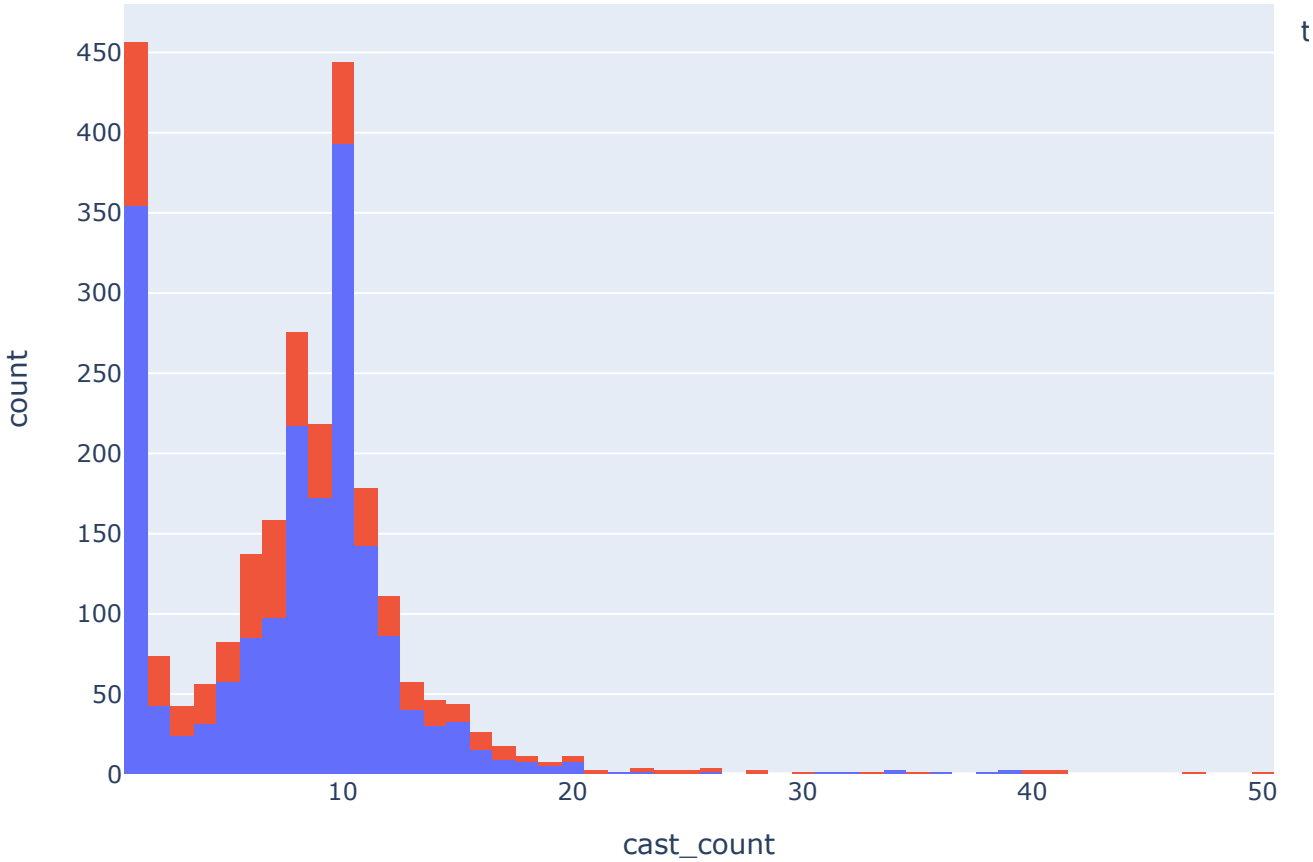
```
df_US_content['cast_count'].mean()
```

```
7.877411575562701
```

So we can observe that a Netflix US content has around 8 important cast members including the male and female actors, directors and side actors. By which we can conclude that Netflix produces a high budget films which includes a greater number of Star Power.

Let us now distinguish the Casts with Type of shows provided by Netflix to US.

```
px.histogram(data_frame=df_US_content,x='cast_count',color = 'type').update_xaxes(categoryord
```



We can see the distribution of data is very much distorted as the number of casts provided in data set is Null values, but still with given data we can conclude that common cast count is 10, and Netflix US mostly uses there various casts in Movies rather than TV-Shows.

So we can catch more number of our favourite cast in Movies rather than TV-Shows.

Question:

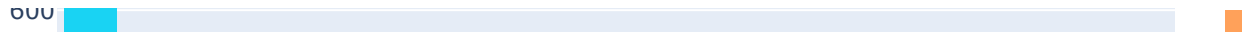
Does Netflix's rating somehow decides it's release day?

```
df_US_content['Day'].value_counts()
```

```
Friday       724
Tuesday      428
Wednesday    357
Thursday     308
Monday       240
Sunday       224
Saturday     201
0              6
Name: Day, dtype: int64
```

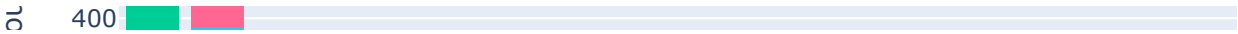
```
df_US_content['Day'] = df_US_content['Released_Date'].dt.day_name().fillna(0)
px.histogram(data_frame=df_US_content,x='rating',color = 'Day').update_xaxes(categoryorder ='
```

We can observe that there is not much variation in Days as most of the Netflix content is released on Friday only, and few about of content is released in Tuesday.



Question

Which Movie is the oldest with its description?




```
df_oldest_US_movie = df_US_content.loc[df_US_content['release_year'].min()]
df_oldest_US_movie
```

```
index                                                        7136
type                                                        Movie
title                                                  John Carter
director                                             Andrew Stanton
cast            Taylor Kitsch, Lynn Collins, Samantha Morton, ...
country                                             United States
release_year                                                 2012
rating                                                      PG-13
duration                                                  133 min
listed_in                    Action & Adventure, Sci-Fi & Fantasy
description     One minute, Civil War veteran John Carter is e...
Released_Date                                 2019-05-02 00:00:00
cast_count                                                     38
Day                                                      Thursday
Name: 1942, dtype: object
```

From above observation we can declare that the oldest movie we can watch on Netflix US is **John Carter** which is PG-13 rating.

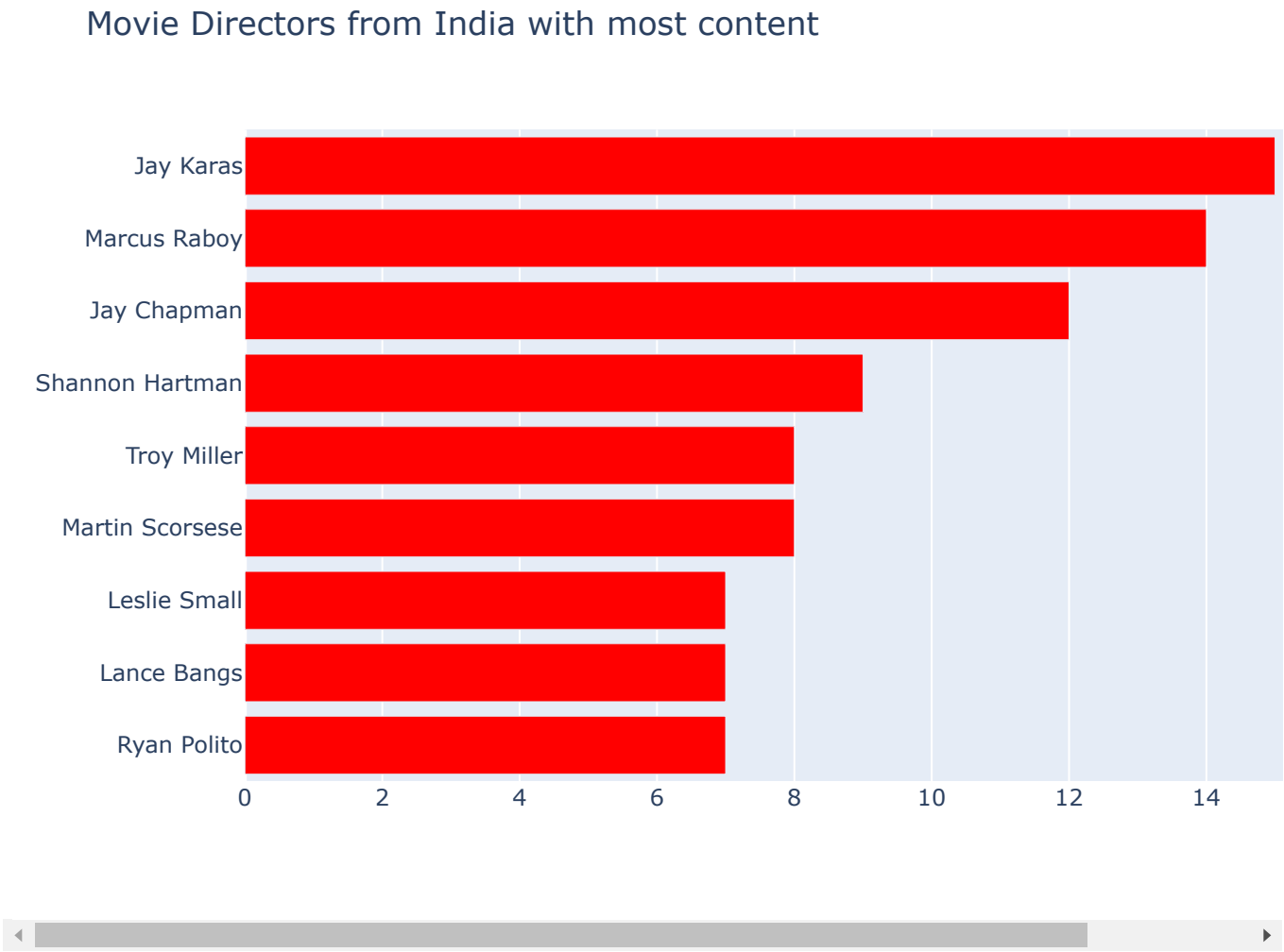
Question:

Movie Directors from United States with most content?

```
from collections import Counter
from plotly import graph_objects as go
import plotly.express as px
small = df[df["type"] == "Movie"]
small = small[small["country"] == "United States"]

col = "director"
categories = ", ".join(small[col].fillna("")).split(", ")
counter_list = Counter(categories).most_common(10)
```

```
counter_list = [_ for _ in counter_list if _[0] != ""]
labels = [_[0] for _ in counter_list][::-1]
values = [_[1] for _ in counter_list][::-1]
trace1 = go.Bar(y=labels, x=values, orientation="h", name="TV Shows", marker=dict(color="red"
data = [trace1]
layout = go.Layout(title="Movie Directors from India with most content", legend=dict(x=0.1, y
fig = go.Figure(data, layout=layout)
fig.show()
```



So we can conclude that most of the Netflix US Movies was directed by Jay Karas, Marcus Raboy.

So if Netflix US wants to make a profitable trade on a movie, so they can choose between these directors.

Now let's focus on 2nd most content consuming country INDIA.

```
df_India_content.shape
df_India_content['type'].value_counts()
```

```
Movie      878
TV Show     62
Name: type, dtype: int64
```

Netflix India has released around 900 Movies and 80 TV Shows for India.

```
df_India_content['Day'] = df_India_content['Released_Date'].dt.day_name()
```

```
df_India_content['Day'].value_counts()
```

```
Thursday     204
Friday       162
Monday       145
Sunday       134
Tuesday      120
Wednesday     94
Saturday      81
Name: Day, dtype: int64
```

As we can see India's mostly content is released in Thursday and Friday.

```
px.histogram(data_frame=df_India_content,x='Day',color = 'type').update_xaxes(categoryorder =
```

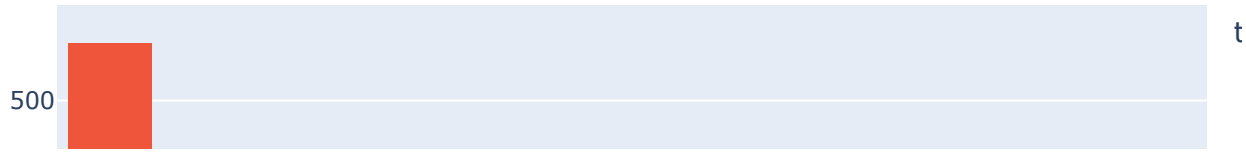
As we have seen that most of the content on Netflix India has released on Thursday and Friday but after seeing histplot we can conclude that most of the Movies on Netflix India is releasing on Thursday but TV shows are releasing on Friday.

So if Netflix India has Movie to release on Platform then it should be on Thursday but a TV show must be released on Friday.

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

```
TV-14       542
TV-MA       237
TV-PG       127
TV-Y7        12
TV-G          6
PG-13         4
TV-Y          4
NR            4
PG            2
R             1
TV-Y7-FV      1
Name: rating, dtype: int64
```

```
px.histogram(data_frame=df_India_content,x='rating',color = 'type').update_xaxes(categoryorde
```



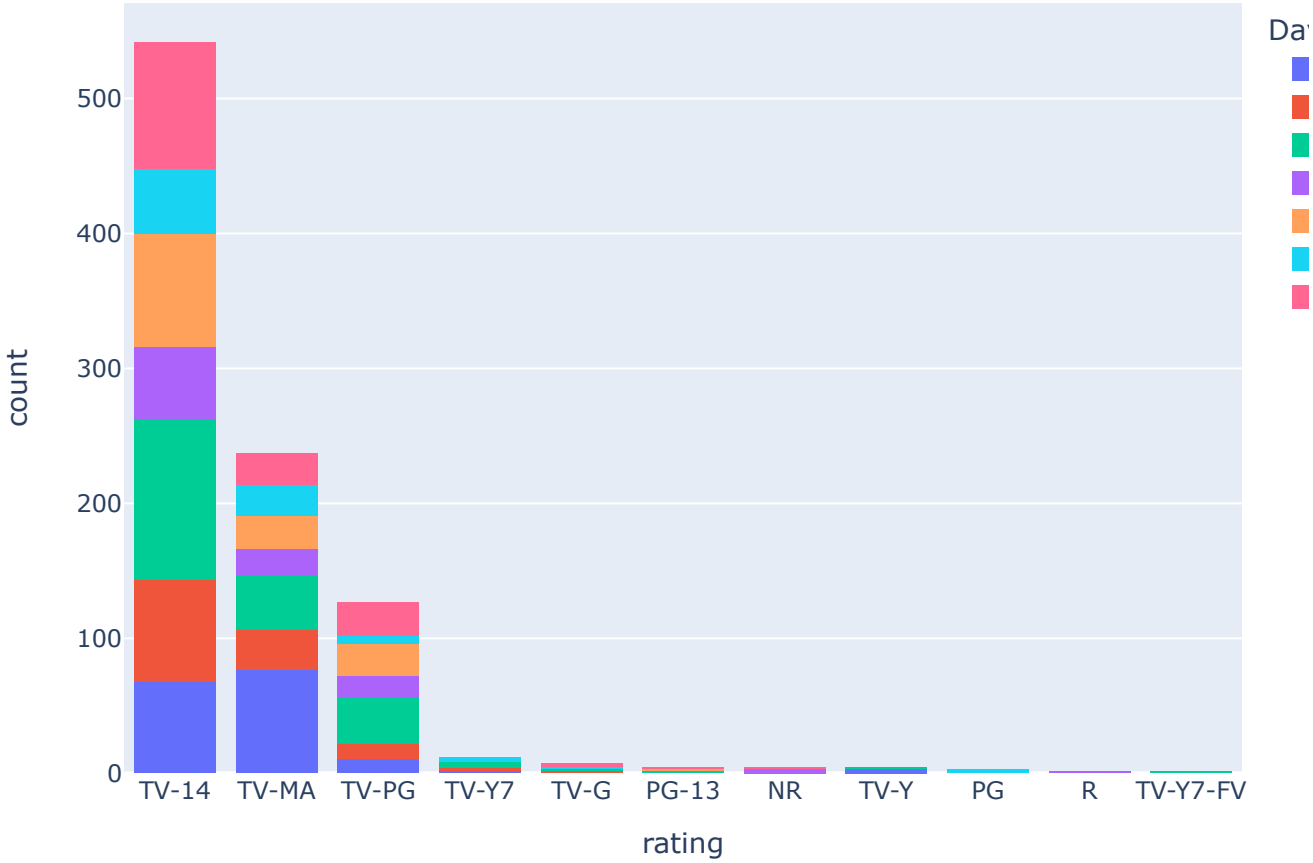
As we can see that most of the Netflix India content is in between TV-14,TV-MA,TV-PG which means that Netflix India's most viewers and targeted viewers are 14+ years.

Netflix India Movie content are for 14+ years audience but there targeted audience for TV-Shows are 21+ years. This is somehow a different strategy from Netflix India.

Question:

Does Netflix's rating somehow decides it's release day?

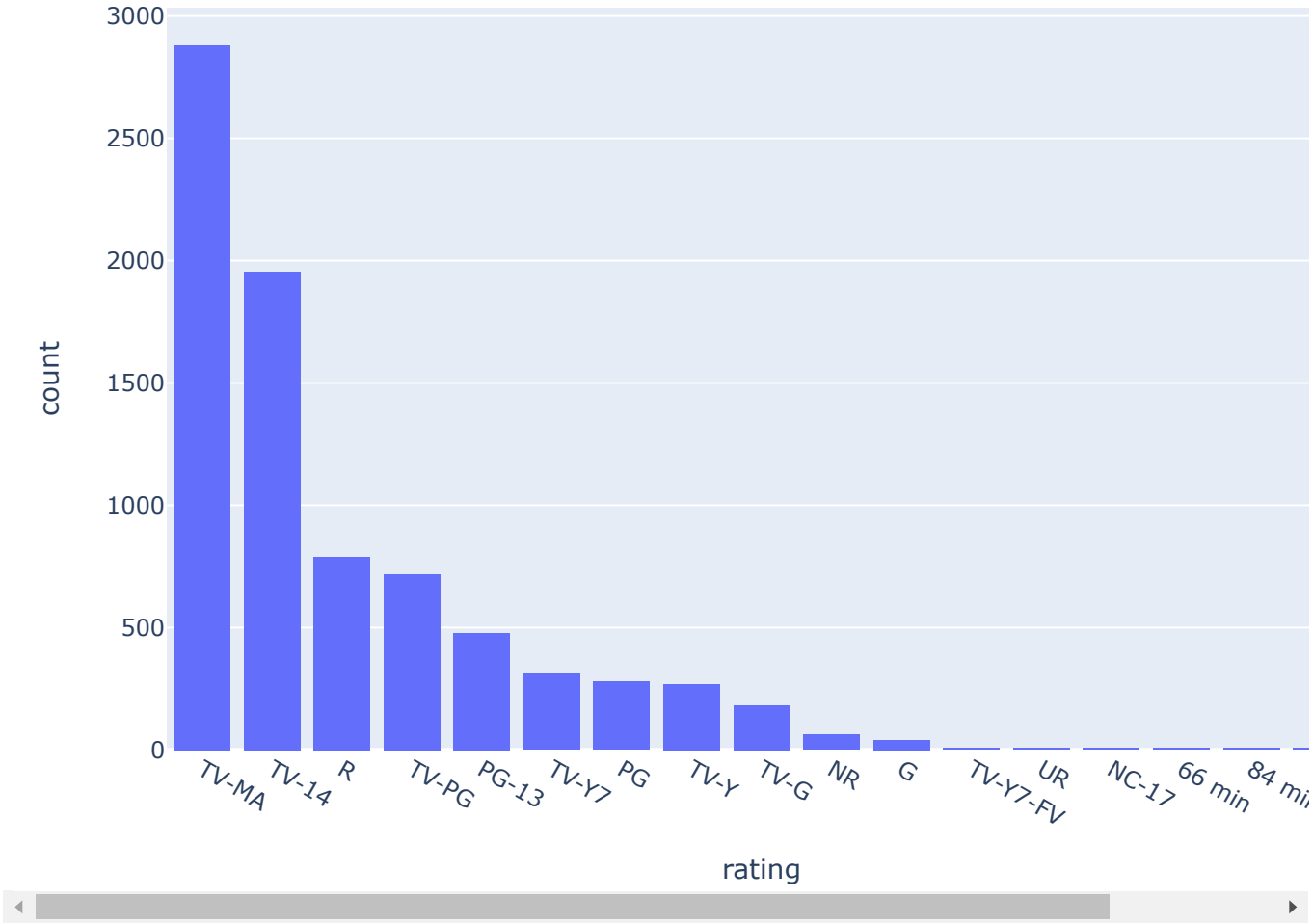
```
px.histogram(data_frame=df_India_content,x='rating',color = 'Day').update_xaxes(categoryorder
```



From above graph we can see that if the rating of that content is TV-14 it is most likely that the content will be released on Thursday or Monday.

But if the content are of other ratings then it is most likely to be released on Friday or Thusrday.

```
px.histogram(df,'rating').update_xaxes(categoryorder = 'total descending')
```




```
df['Cast_Counts'] = df['cast'].str.split(",")
df = df[df['Cast_Counts'].notna()]
df['Cast_Counts'] = df['Cast_Counts'].apply(lambda x: len(x))
```

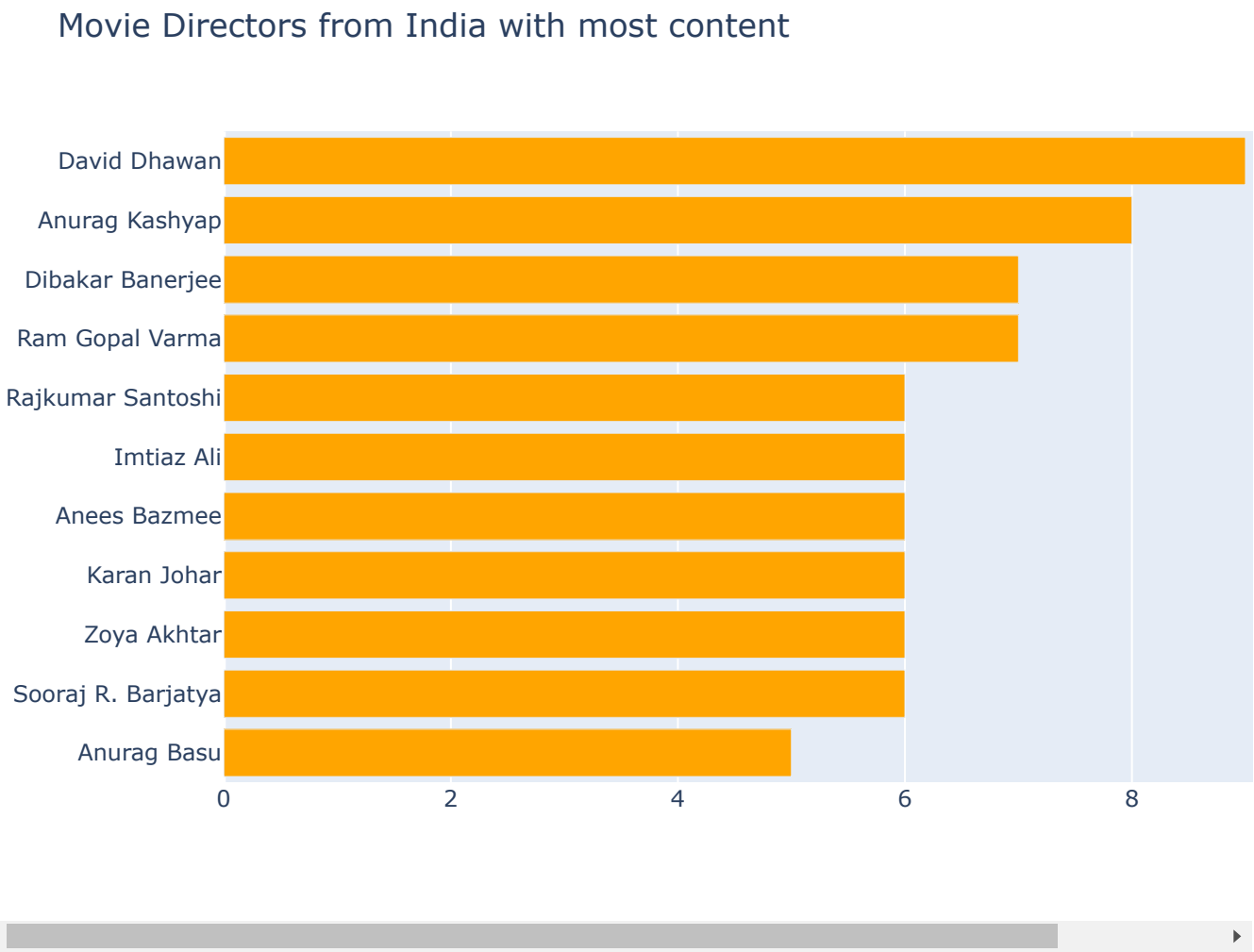
```
df.head(5)
```

	type	title	director	cast	country	release_year	rating	duration	liste
1	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021	TV-MA	2 Seasons	Interna TV Sh TV Dra Mys
2	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021	TV-MA	1 Season	Crim Sh Interna TV Sh TV
4	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021	TV-MA	2 Seasons	Interna TV Sh Rom TV Sh
5	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	NaN	2021	TV-MA	1 Season	TV Dra TV H Mys
		My Little Pony: A	Robert Cullen	Vanessa Hudgens, Kimiko					Child

```
small = df[df["type"] == "Movie"]
small = small[small["country"] == "India"]

col = "director"
categories = ", ".join(small[col].fillna("")).split(", ")
counter_list = Counter(categories).most_common(12)
counter_list = [_ for _ in counter_list if _[0] != ""]
labels = [_[0] for _ in counter_list][::-1]
values = [_[1] for _ in counter_list][::-1]
trace1 = go.Bar(y=labels, x=values, orientation="h", name="TV Shows", marker=dict(color="oran

data = [trace1]
layout = go.Layout(title="Movie Directors from India with most content", legend=dict(x=0.1, y
fig = go.Figure(data, layout=layout)
fig.show()
```



As we can observe from the bar plot that Netflix produced movies are more likely directed by David Dhawan, Anurag Kashyap, Dibakar Banerjee and Ram Gopal Verma.

As Netflix India focuses more on MA content so the movies directors are most obviously Anurag Kashyap and Ram Gopal Verma.

Question

Which is the oldest Indie movie available on Netflix Platform?

```
df_oldest_India_movie = df_India_content['release_year'].min()
df_India_content.loc[df_India_content['release_year']==df_oldest_India_movie].reset_index()
```

	level_0	index	type	title	director	cast	country	release_year	rating	durati
0	37	542	Movie	Ujala	Naresh Saigal	Mala Sinha, Shammi Kapoor, Raaj	India	1959	TV-14	143 r

From above observation we can conclude that the oldest movie we can watch on Netflix is **Ujala** which is directed by Nareh Saigal in 1959.

Conclusion:

1. From all above observations we can conclude that Netflix is providing services differently on the basis of Country's Popularity Content, New Trends, working with a huge popular cast for high reachability and productivity of Netflix.
2. Netflix also has distributed its region so the team can only focus on that region and its culture and its popular trend.
3. Netflix also try to re-lauches old movies and TV Show on its platform for more reachability among peoples, to provide Nostalgia feeling to that aged group of people.