In [2]:
```
import numpy as np
import pandas as pd
import plotly.express as px
from textblob import TextBlob
```

In [3]:
```
df=pd.read_csv("netflix_titles.csv")
```

In [4]:
```
df
```

Out[4]:

| | 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... |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **8802** | s8803 | Movie | Zodiac | David Fincher | Mark Ruffalo, Jake Gyllenhaal, Robert Downey J... | United States | November 20, 2019 | 2007 | R | 158 min | Cult Movies, Dramas, Thrillers | A political cartoonist, a crime reporter and a... |
| **8803** | s8804 | TV Show | Zombie Dumb | NaN | NaN | NaN | July 1, 2019 | 2018 | TV-Y7 | 2 Seasons | Kids' TV, Korean TV Shows, TV Comedies | While living alone in a spooky town, a young g... |
| **8804** | s8805 | Movie | Zombieland | Ruben Fleischer | Jesse Eisenberg, Woody Harrelson, Emma Stone, ... | United States | November 1, 2019 | 2009 | R | 88 min | Comedies, Horror Movies | Looking to survive in a world taken over by zo... |
| **8805** | s8806 | Movie | Zoom | Peter Hewitt | Tim Allen, Courteney Cox, Chevy Chase, Kate Ma... | United States | January 11, 2020 | 2006 | PG | 88 min | Children & Family Movies, Comedies | Dragged from civilian life, a former superhero... |
| **8806** | s8807 | Movie | Zubaan | Mozez Singh | Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan... | India | March 2, 2019 | 2015 | TV-14 | 111 min | Dramas, International Movies, Music & Musicals | A scrappy but poor boy worms his way into a ty... |

8807 rows × 12 columns

# To check how many rows and columns available in data

In [5]:
```
df.shape
```

Out[5]:
```
(8807, 12)
```

# To check columns

In [6]:
```
df.columns
```

Out[6]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

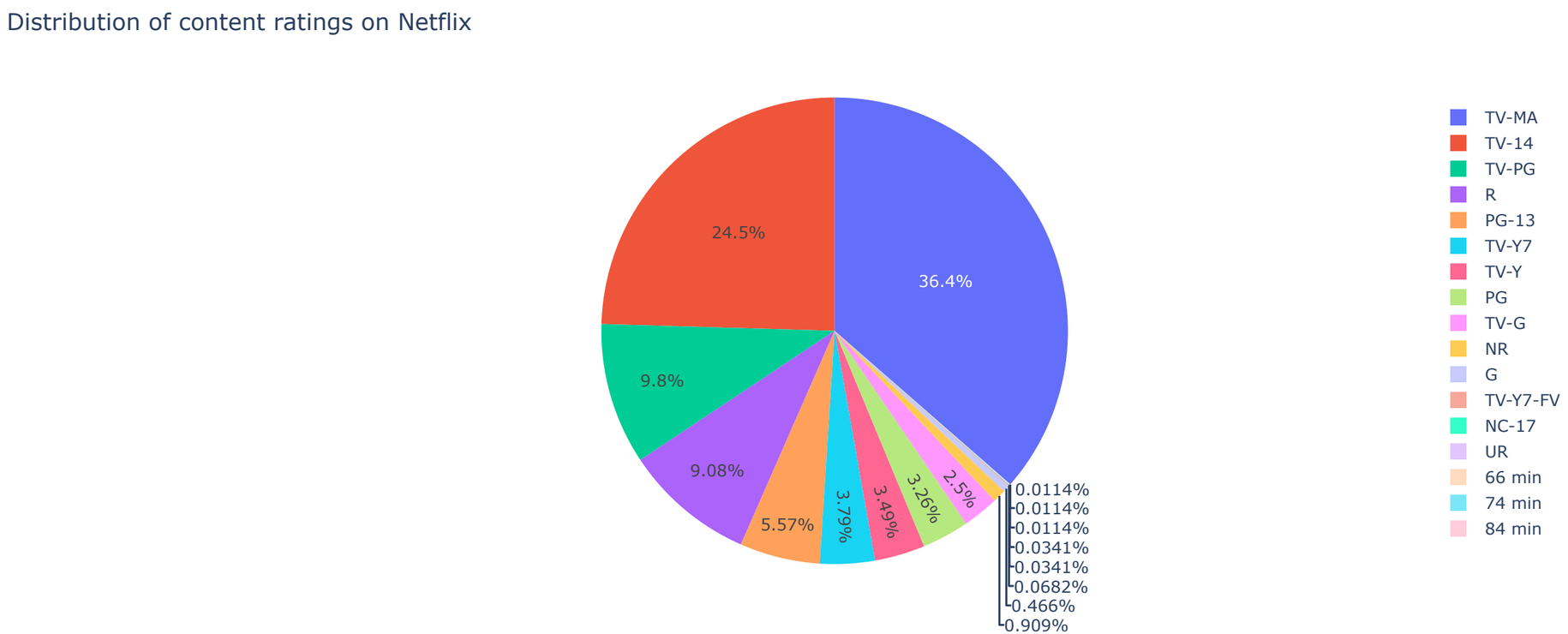
# For checking the most demanding content on netflix according to ratings

In [7]:
```
x=df.groupby(['rating']).size().reset_index(name="counts")
x
```

Out[7]:

| | rating | counts |
|---|---|---|
| **0** | 66 min | 1 |
| **1** | 74 min | 1 |
| **2** | 84 min | 1 |
| **3** | G | 41 |
| **4** | NC-17 | 3 |
| **5** | NR | 80 |
| **6** | PG | 287 |
| **7** | PG-13 | 490 |
| **8** | R | 799 |
| **9** | TV-14 | 2160 |
| **10** | TV-G | 220 |
| **11** | TV-MA | 3207 |
| **12** | TV-PG | 863 |
| **13** | TV-Y | 307 |
| **14** | TV-Y7 | 334 |
| **15** | TV-Y7-FV | 6 |
| **16** | UR | 3 |

In [8]:
```
piechart=px.pie(x,values='counts', names='rating',title='Distribution of content ratings on Netflix')
piechart.show()
#x is the count of the above dataset
```




## To fill the director name which is NoN

In [9]:
```
df.isnull()
```

Out[9]:

| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | False | False | False | False | True | False | False | False | False | False | False | False |
| **1** | False | False | False | True | False | False | False | False | False | False | False | False |
| **2** | False | False | False | False | False | True | False | False | False | False | False | False |
| **3** | False | False | False | True | True | True | False | False | False | False | False | False |
| **4** | False | False | False | True | False | False | False | False | False | False | False | False |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **8802** | False | False | False | False | False | False | False | False | False | False | False | False |
| **8803** | False | False | False | True | True | True | False | False | False | False | False | False |
| **8804** | False | False | False | False | False | False | False | False | False | False | False | False |
| **8805** | False | False | False | False | False | False | False | False | False | False | False | False |
| **8806** | False | False | False | False | False | False | False | False | False | False | False | False |

8807 rows × 12 columns

In [10]:
```
df['director']=df['director'].fillna("Director not specified")
df.head()
```

Out[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 | Director not specified | 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 | Director not specified | 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 | Director not specified | 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... |

# To collect all director name

In [11]:
```
director_list=pd.DataFrame()
print(director_list)
```

```
Empty DataFrame
Columns: []
Index: []
```

In [12]:
```
director_list=df['director'].str.split(',',expand=True).stack()
print(director_list)
```

```
0     0          Kirsten Johnson
1     0    Director not specified
2     0          Julien Leclercq
3     0    Director not specified
4     0    Director not specified
                   ...
8802  0            David Fincher
8803  0    Director not specified
8804  0          Ruben Fleischer
8805  0             Peter Hewitt
8806  0              Mozez Singh
Length: 9612, dtype: object
```

In [13]:
```
director_list=director_list.to_frame()
print(director_list)
```

```
                            0
0     0         Kirsten Johnson
1     0   Director not specified
2     0         Julien Leclercq
3     0   Director not specified
4     0   Director not specified
...                         ...
8802  0           David Fincher
8803  0   Director not specified
8804  0         Ruben Fleischer
8805  0            Peter Hewitt
8806  0             Mozez Singh

[9612 rows x 1 columns]
```

In [14]:
```
director_list.columns=['Director']
print(director_list)
```

```
                     Director
0     0         Kirsten Johnson
1     0   Director not specified
2     0         Julien Leclercq
3     0   Director not specified
4     0   Director not specified
...                         ...
8802  0           David Fincher
8803  0   Director not specified
8804  0         Ruben Fleischer
8805  0            Peter Hewitt
8806  0             Mozez Singh

[9612 rows x 1 columns]
```

# How many content the director have created

In [15]:
```
director=director_list.groupby(['Director']).size().reset_index(name='Total count')
print(director)
```

```
                     Director  Total count
0               Aaron Moorhead            2
1                  Aaron Woolf            1
2      Abbas Alibhai Burmawalla            1
3              Abdullah Al Noor            1
4           Abhinav Shiv Tiwari            1
...                        ...          ...
5116               Çagan Irmak            1
5117           Ísold Uggadóttir            1
5118       Óskar Thór Axelsson            1
5119          Ömer Faruk Sorak            2
5120              Şenol Sönmez            2

[5121 rows x 2 columns]
```

In [16]:
```
director=director[director.Director != 'Director not specified']
```

In [17]:
```
print(director)
```

```
                     Director  Total count
0               Aaron Moorhead            2
1                  Aaron Woolf            1
2      Abbas Alibhai Burmawalla            1
3              Abdullah Al Noor            1
4           Abhinav Shiv Tiwari            1
...                        ...          ...
5116               Çagan Irmak            1
5117           Ísold Uggadóttir            1
5118       Óskar Thór Axelsson            1
5119          Ömer Faruk Sorak            2
5120              Şenol Sönmez            2

[5120 rows x 2 columns]
```

# To sort the count of director created content

In [18]:
```
director=director.sort_values(by=["Total count"],ascending=False)
print(director)
```

```
             Director  Total count
4021    Rajiv Chilaka           22
4068      Raúl Campos           18
261         Jan Suter           18
4652      Suhas Kadav           16
3236     Marcus Raboy           16
...               ...          ...
2341         J. Davis            1
2342  J. Lee Thompson            1
2343  J. Michael Long            1
609    Smriti Keshari            1
2561    Joaquín Mazón            1

[5120 rows x 2 columns]
```

# To take top 5 director

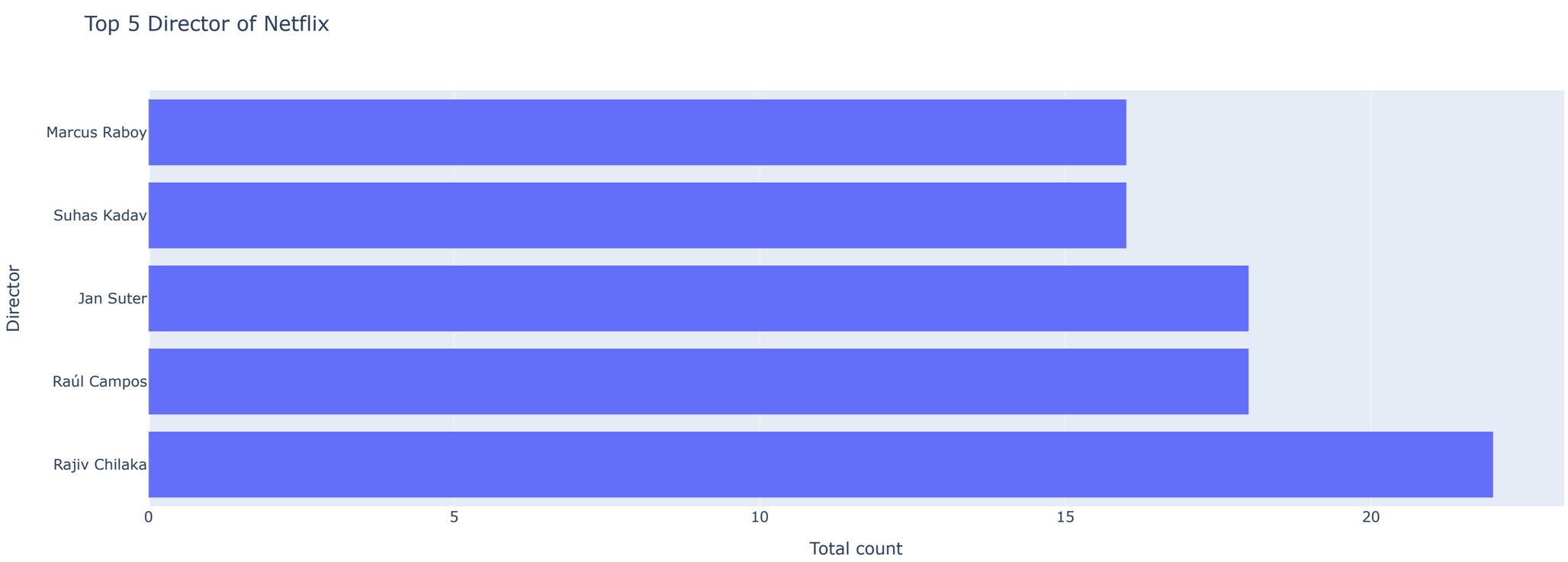
In [19]:
```
top5director=director.head(5)
```

In [20]:
```
print(top5director)
```

```
          Director  Total count
4021  Rajiv Chilaka           22
4068    Raúl Campos           18
261       Jan Suter           18
4652    Suhas Kadav           16
3236   Marcus Raboy           16
```

# To creating bar chart

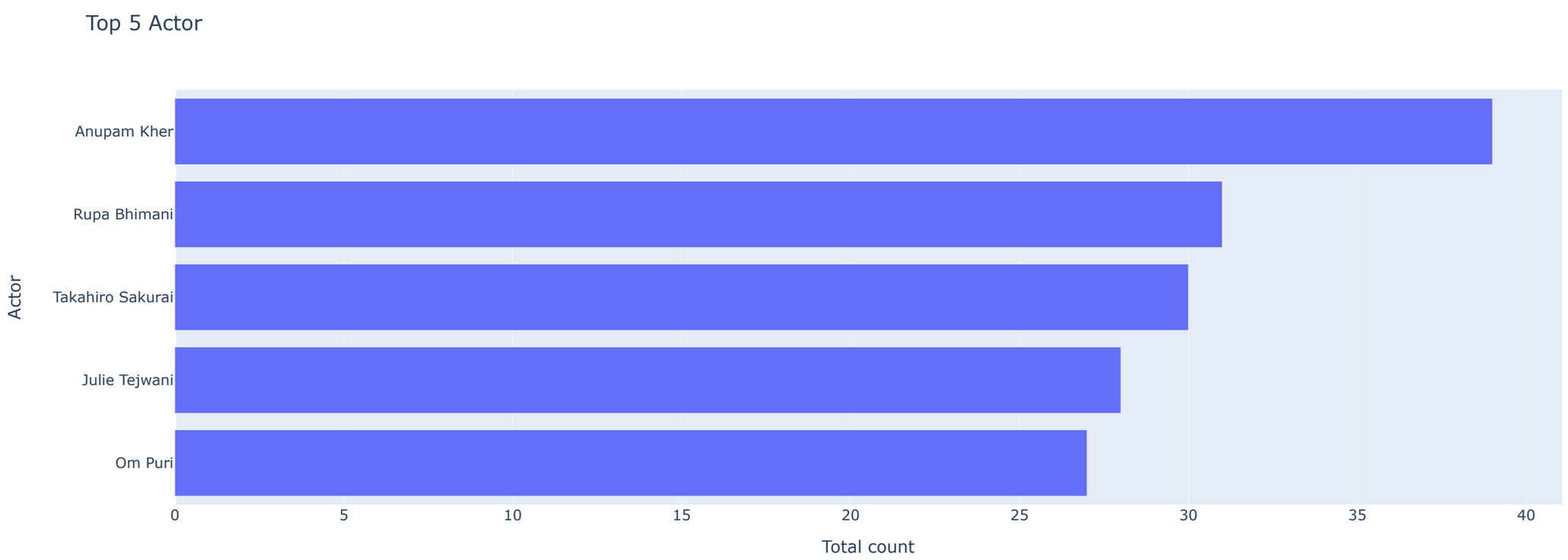
In [21]:
```
barchart=px.bar(top5director,x='Total count',y='Director',title="Top 5 Director of Netflix")
barchart.show()
```




# For Top 5 Actor

# To fill Cas NaN values

In [24]:
```
df['cast']=df['cast'].fillna("No cast specified")
cast_df=pd.DataFrame()
cast_df=df['cast'].str.split(',',expand=True).stack()
cast_df=cast_df.to_frame()
cast_df.columns=['Actor']
actors=cast_df.groupby(['Actor']).size().reset_index(name='Total count')
actors=actors[actors.Actor!='No cast specified']
actors=actors.sort_values(by=['Total count'],ascending=False)
top5Actors=actors.head()
top5Actors=top5Actors.sort_values(by=['Total count'])
barchart2=px.bar(top5Actors,x='Total count',y='Actor',title='Top 5 Actor')
barchart2.show()
```

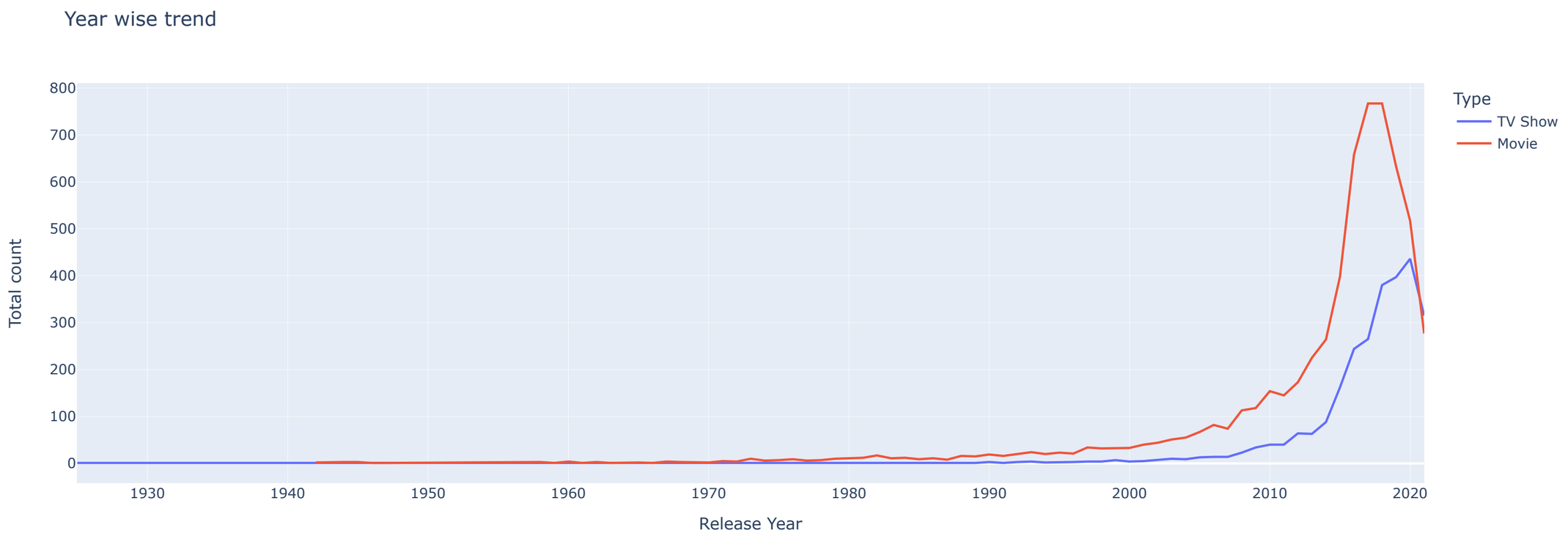



# Analyzing the content produced on netflix based on year

In [27]:
```
df1=df[['type','release_year']]
df1=df1.rename(columns={"release_year":"Release Year","type":"Type"})
df2=df1.groupby(['Release Year','Type']).size().reset_index(name='Total count')
print(df2)
```

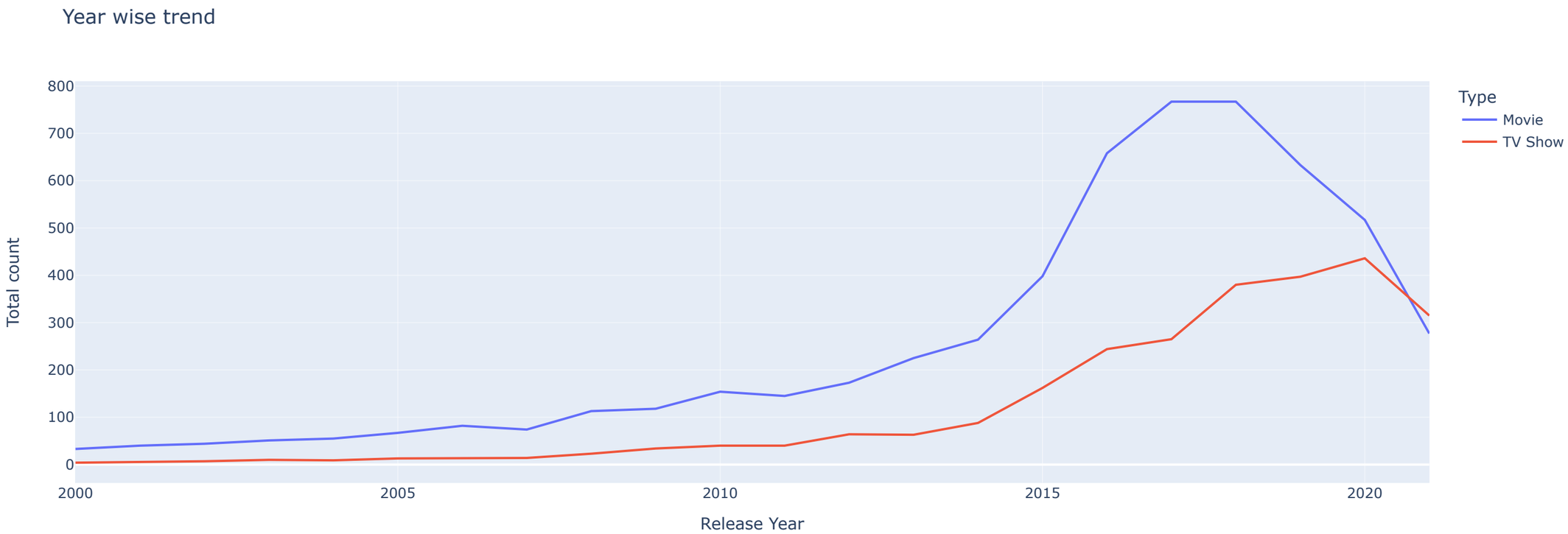
```
     Release Year     Type  Total count
0            1925  TV Show            1
1            1942    Movie            2
2            1943    Movie            3
3            1944    Movie            3
4            1945    Movie            3
..            ...      ...          ...
114          2019  TV Show          397
115          2020    Movie          517
116          2020  TV Show          436
117          2021    Movie          277
118          2021  TV Show          315

[119 rows x 3 columns]
```

In [30]:
```
graph=px.line(df2, x='Release Year', y="Total count",color="Type",title='Year wise trend')
graph.show()
```




## To taking above chart from 2000 to 2020

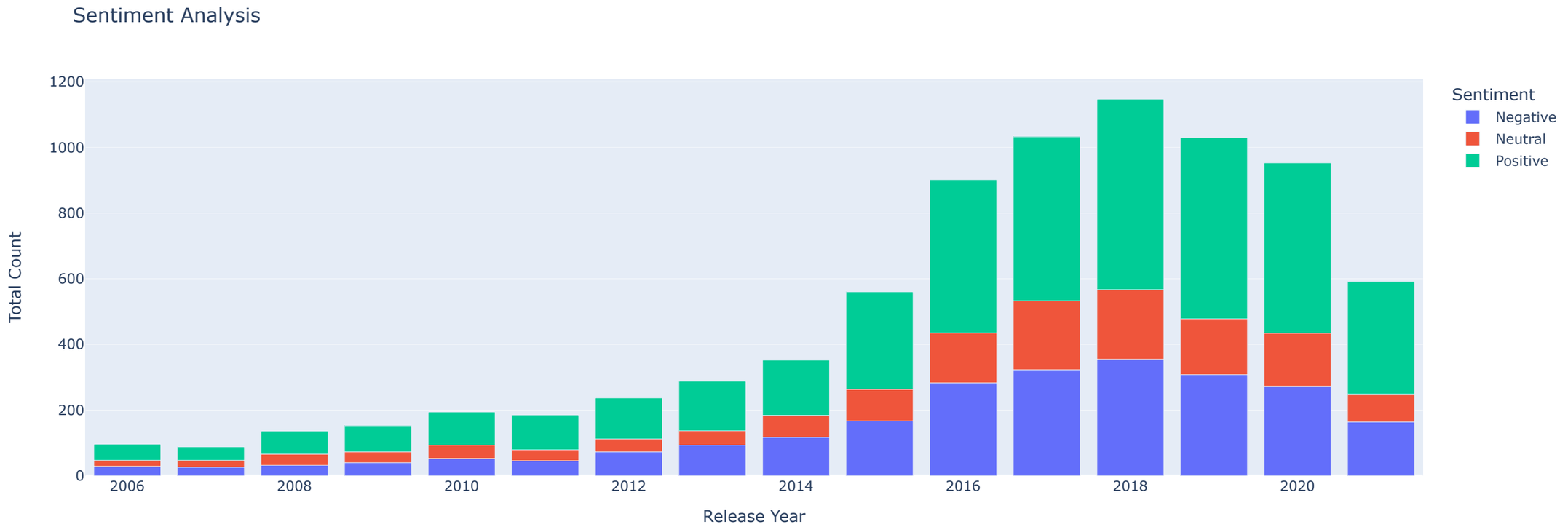
In [31]:
```
df2=df2[df2['Release Year']>=2000]
graph=px.line(df2, x='Release Year', y="Total count",color="Type",title='Year wise trend')
graph.show()
```




# To analyse the sentiment of netflix content

In [46]:
```
# Assuming df is your DataFrame containing 'release_year' and 'description' columns

df3 = df[['release_year', 'description']]
df3 = df3.rename(columns={'release_year': 'Release Year', 'description': 'Description'})

sentiments = []
for index, row in df3.iterrows():
    d = row['Description']
    testimonial = TextBlob(d)
    p = testimonial.sentiment.polarity
    if p == 0:
        sent = 'Neutral'
    elif p > 0:
        sent = 'Positive'
    else:
        sent = 'Negative'
    sentiments.append(sent)

df3['Sentiment'] = sentiments

# Group by Release Year and Sentiment, calculate counts
df3 = df3.groupby(['Release Year', 'Sentiment']).size().reset_index(name='Total Count')
```

```
df3 = df3[df3['Release Year'] > 2005]

# Plot the bar graph
barGraph = px.bar(df3, x="Release Year", y="Total Count", color="Sentiment", title="Sentiment Analysis")
barGraph.show()
```




In [ ]: