```
from google.colab import files
```

```
uploaded = files.upload()
```

Browse... No files selected. Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving netflix.csv to netflix.csv
```

```
import pandas as pd
import io

COLUMNS = ["show_id","type","title","director","cast","country","date_added","release_year'
df = pd.read_csv(io.BytesIO(uploaded['netflix.csv']), names=COLUMNS, skipinitialspace=True,
print(df)
```

```
     show_id     type                  title         director  \
0         s1    Movie   Dick Johnson Is Dead  Kirsten Johnson
1         s2  TV Show          Blood & Water              NaN
2         s3  TV Show              Ganglands  Julien Leclercq
3         s4  TV Show  Jailbirds New Orleans              NaN
4         s5  TV Show           Kota Factory              NaN
...      ...      ...                    ...              ...
8802   s8803    Movie                 Zodiac    David Fincher
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
```

✓ 0s completed at 11:03 AM

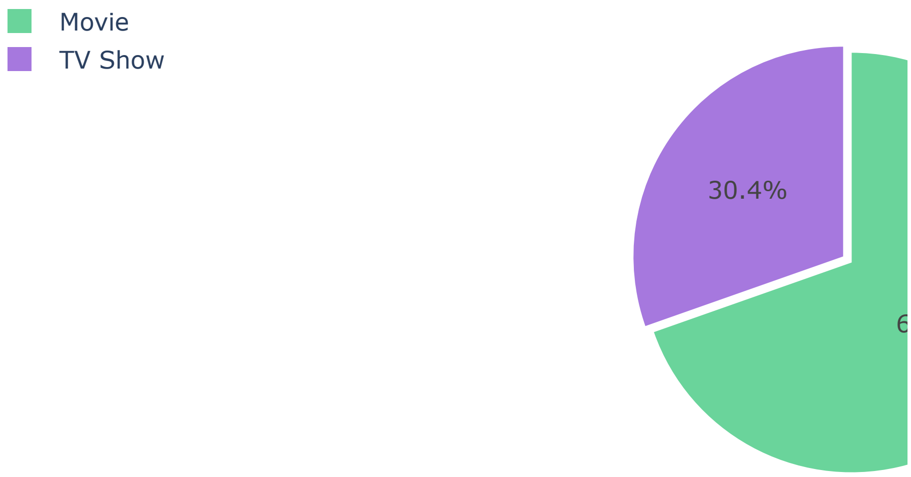
```
8805    January 11, 2020          2006     PG     88 min
8806       March 2, 2019          2015  TV-14    111 min

                                           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  year_added  \
0     As her father nears the end of his life, filmm...         NaN
1     After crossing paths at a party, a Cape Town t...         NaN
2     To protect his family from a powerful drug lor...         NaN
3     Feuds, flirtations and toilet talk go down amo...         NaN
4     In a city of coaching centers known to train I...         NaN
```

```
import plotly.graph_objects as go
from plotly.offline import init_notebook_mode, iplot
import pandas as pd

## add new features in the dataset
df["date_added"] = pd.to_datetime(df["date_added"], errors='coerce', format='%m%d%Y')
df['year_added'] = df['date_added'].dt.year
df['month_added'] = df['date_added'].dt.month

#df['season_count'] = df.apply(lambda x : x['duration'].split(" ")[0] if "Season" in x['dur
#df['duration'] = df.apply(lambda x : x['duration'].split(" ")[0] if "Season" not in x['dur
df.head()
```

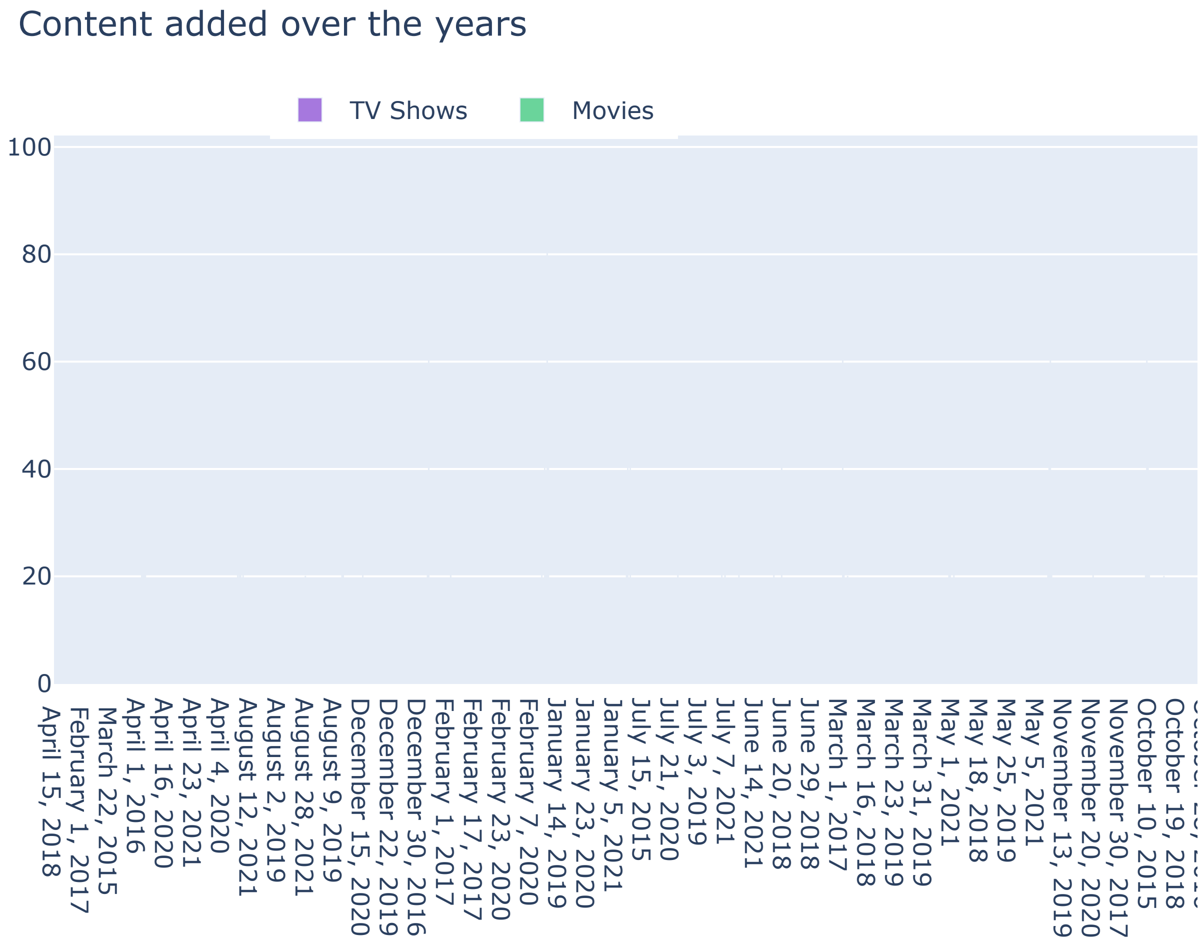
	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	NaT	2020
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	NaT	2021
					Sami			

2	s3	TV Show	Ganglands	Julien Leclercq	Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	NaT	2021
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	NaT	2021
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	NaT	2021

```
col = "type"
grouped = df[col].value_counts().reset_index()
grouped = grouped.rename(columns = {col : "count", "index" : col})

## plot
trace = go.Pie(labels=grouped[col], values=grouped['count'], pull=[0.05, 0], marker=dict(c
layout = go.Layout(title="", height=400, legend=dict(x=0.1, y=1.1))
fig = go.Figure(data = [trace], layout = layout)
iplot(fig)
```



```
col = "release_year"

vc1 = d1[col].value_counts().reset_index()
vc1 = vc1.rename(columns = {col : "count", "index" : col})
vc1['percent'] = vc1['count'].apply(lambda x : 100*x/sum(vc1['count']))
vc1 = vc1.sort_values(col)

vc2 = d2[col].value_counts().reset_index()
vc2 = vc2.rename(columns = {col : "count", "index" : col})
vc2['percent'] = vc2['count'].apply(lambda x : 100*x/sum(vc2['count']))
vc2 = vc2.sort_values(col)

trace1 = go.Bar(x=vc1[col], y=vc1["count"], name="TV Shows", marker=dict(color="#a678de"))
trace2 = go.Bar(x=vc2[col], y=vc2["count"], name="Movies", marker=dict(color="#6ad49b"))
data = [trace1, trace2]
layout = go.Layout(title="Content added over the years", legend=dict(x=0.1, y=1.1, orientat
fig = go.Figure(data, layout=layout)
fig.show()
```



```
small = df.sort_values("release_year", ascending = True)
small = small[small['duration'] != ""]
small[['title', "release_year"]][:15]
```

	title	release_year
4250	Pioneers: First Women Filmmakers*	1925
7790	Prelude to War	1942
8205	The Battle of Midway	1942
8660	Undercover: How to Operate Behind Enemy Lines	1943
8739	Why We Fight: The Battle of Russia	1943
8763	WWII: Report from the Aleutians	1943
8640	Tunisian Victory	1944
8436	The Negro Soldier	1944
8419	The Memphis Belle: A Story of a\nFlying Fortress	1944
7930	San Pietro	1945
1331	Five Came Back: The Reference Films	1945
7219	Know Your Enemy - Japan	1945
7575	Nazi Concentration Camps	1945
7743	Pioneers of African-American Cinema	1946
7294	Let There Be Light	1946

```
print('Some of the oldest TV Shows on Netflix')
small = df.sort_values("release_year", ascending = True)
small = small[small['season_count'] != ""]
small[['title', "release_year"]][:15]
```

```
Some of the oldest TV Shows on Netflix
```

	title	release_year
4250	Pioneers: First Women Filmmakers*	1925

7790	Prelude to War	1942
8205	The Battle of Midway	1942
8660	Undercover: How to Operate Behind Enemy Lines	1943
8739	Why We Fight: The Battle of Russia	1943
8763	WWII: Report from the Aleutians	1943
8640	Tunisian Victory	1944
8436	The Negro Soldier	1944
8419	The Memphis Belle: A Story of a\nFlying Fortress	1944
7930	San Pietro	1945
1331	Five Came Back: The Reference Films	1945
7219	Know Your Enemy - Japan	1945
7575	Nazi Concentration Camps	1945
7743	Pioneers of African-American Cinema	1946
7294	Let There Be Light	1946

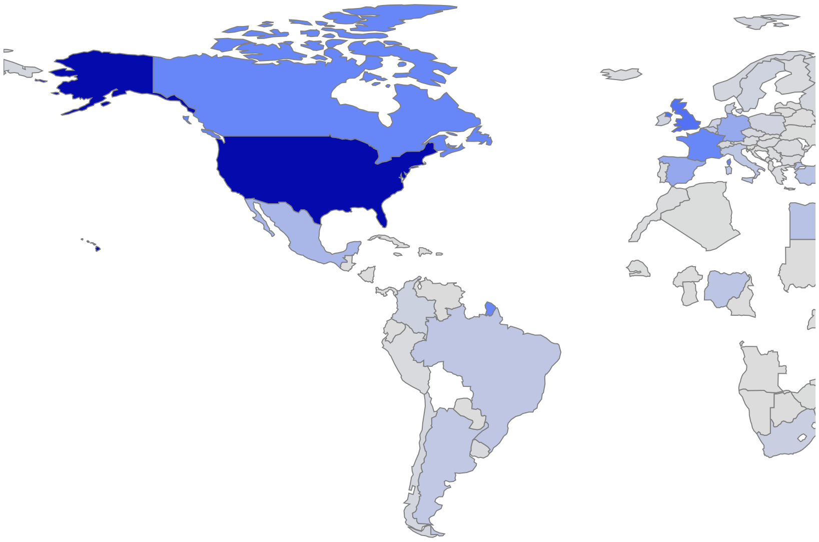
```
country_codes = {'afghanistan': 'AFG',
 'albania': 'ALB',
 'algeria': 'DZA',
 'american samoa': 'ASM',
 'andorra': 'AND',
 'angola': 'AGO',
 'anguilla': 'AIA',
 'antigua and barbuda': 'ATG',
 'argentina': 'ARG',
 'armenia': 'ARM',
 'aruba': 'ABW',
 'australia': 'AUS',
 'austria': 'AUT',
 'azerbaijan': 'AZE',
 'bahamas': 'BHM',
 'bahrain': 'BHR',
 'bangladesh': 'BGD',
 'barbados': 'BRB',
 'belarus': 'BLR',
 'belgium': 'BEL',
 'belize': 'BLZ',
 'benin': 'BEN',
 'bermuda': 'BMU',
 'bhutan': 'BTN',
 'bolivia': 'BOL',
 'bosnia and herzegovina': 'BIH',
 'botswana': 'BWA',
 'brazil': 'BRA',
 'british virgin islands': 'VGB',
 'brunei': 'BRN',
 'bulgaria': 'BGR',
 'burkina faso': 'BFA',
 'burma': 'MMR',
 'burundi': 'BDI',
 'cabo verde': 'CPV',
 'cambodia': 'KHM',
 'cameroon': 'CMR',
 'canada': 'CAN',
 'cayman islands': 'CYM',
 'central african republic': 'CAF',
 'chad': 'TCD',
 'chile': 'CHL',
 'china': 'CHN',
 'colombia': 'COL',
 'comoros': 'COM',
 'congo democratic': 'COD',
 'Congo republic': 'COG',
 'cook islands': 'COK',
 'costa rica': 'CRI',
 "cote d'ivoire": 'CIV',
 'croatia': 'HRV',
 'cuba': 'CUB',
 'curacao': 'CUW',
 'cyprus': 'CYP',
 'czech republic': 'CZE',
 'denmark': 'DNK',
 'djibouti': 'DJI',
 'dominica': 'DMA',
 'dominican republic': 'DOM',
 'ecuador': 'ECU',
 'egypt': 'EGY',
 'el salvador': 'SLV',
 'equatorial guinea': 'GNQ',
 'eritrea': 'ERI',
 'estonia': 'EST',
 'ethiopia': 'ETH',
 'falkland islands': 'FLK',
 'faroe islands': 'FRO',
 'fiji': 'FJI',
 'finland': 'FIN',
 'france': 'FRA',
 'french polynesia': 'PYF',
 'gabon': 'GAB',
 'gambia, the': 'GMB',
 'georgia': 'GEO',
 'germany': 'DEU',
 'ghana': 'GHA'
```

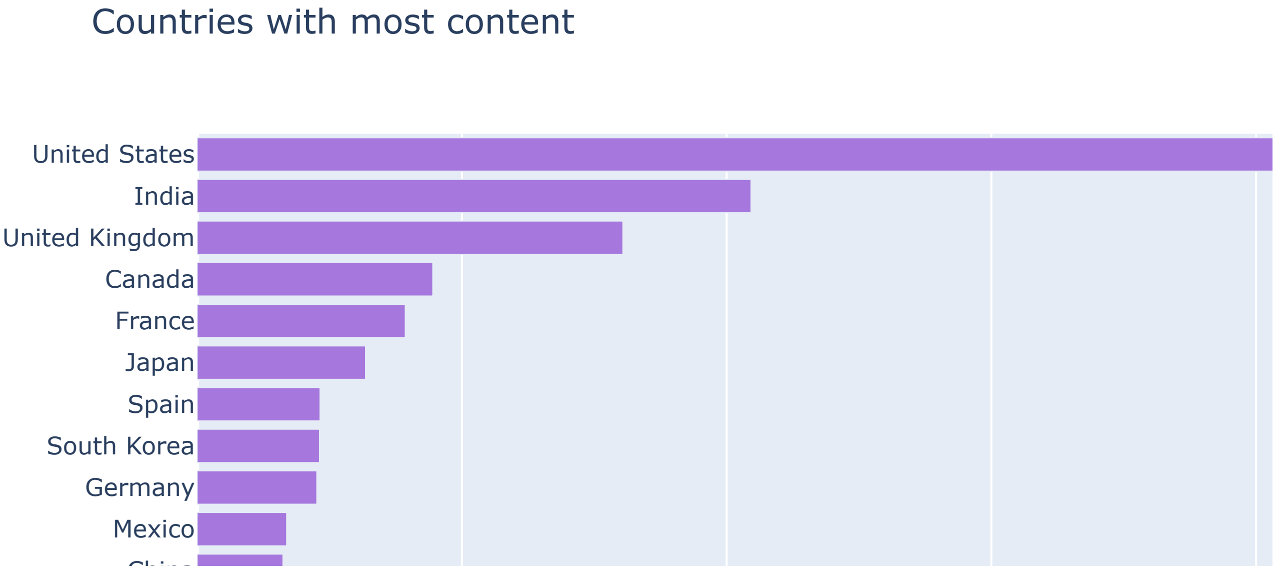
```
    'ghana': 'GHA',
    'gibraltar': 'GIB',
    'greece': 'GRC',
    'greenland': 'GRL',
    'grenada': 'GRD',
    'guam': 'GUM',
    'guatemala': 'GTM',
    'guernsey': 'GGY',
    'guinea-bissau': 'GNB',
    'guinea': 'GIN',
    'guyana': 'GUY',
    'haiti': 'HTI',
    'honduras': 'HND',
    'hong kong': 'HKG',
    'hungary': 'HUN',
    'iceland': 'ISL',
    'india': 'IND',
    'indonesia': 'IDN',
    'iran': 'IRN',
    'iraq': 'IRQ',
    'ireland': 'IRL',
    'isle of man': 'IMN',
    'israel': 'ISR',
    'italy': 'ITA',
    'jamaica': 'JAM',
    'japan': 'JPN',
    'jersey': 'JEY',
    'jordan': 'JOR',
    'kazakhstan': 'KAZ',
    'kenya': 'KEN',
    'kiribati': 'KIR',
    'north korea': 'PRK',
    'south korea': 'KOR',
    'kosovo': 'KSV',
    'kuwait': 'KWT',
    'kyrgyzstan': 'KGZ',
    'laos': 'LAO',
    'latvia': 'LVA',
    'lebanon': 'LBN',
    'lesotho': 'LSO',
    'liberia': 'LBR',
    'libya': 'LBY',
    'liechtenstein': 'LIE',
    'lithuania': 'LTU',
    'luxembourg': 'LUX',
    'macau': 'MAC',
    'macedonia': 'MKD',
    'madagascar': 'MDG',
    'malawi': 'MWI',
    'malaysia': 'MYS',
    'maldives': 'MDV',
    'mali': 'MLI',
    'malta': 'MLT',
    'marshall islands': 'MHL',
    'mauritania': 'MRT',
    'mauritius': 'MUS',
    'mexico': 'MEX',
    'micronesia': 'FSM',
    'moldova': 'MDA',
    'monaco': 'MCO',
    'mongolia': 'MNG',
    'montenegro': 'MNE',
    'morocco': 'MAR',
    'mozambique': 'MOZ',
    'namibia': 'NAM',
    'nepal': 'NPL',
    'netherlands': 'NLD',
    'new caledonia': 'NCL',
    'new zealand': 'NZL',
    'nicaragua': 'NIC',
    'nigeria': 'NGA',
    'niger': 'NER',
    'niue': 'NIU',
    'northern mariana islands': 'MNP',
    'norway': 'NOR',
    'oman': 'OMN',
    'pakistan': 'PAK',
    'palau': 'PLW',
    'panama': 'PAN',
    'papua new guinea': 'PNG',
    'paraguay': 'PRY',
    'peru': 'PER',
    'philippines': 'PHL',
    'poland': 'POL',
    'portugal': 'PRT',
    'puerto rico': 'PRI',
    'qatar': 'QAT',
    'romania': 'ROU',
    'russia': 'RUS',
    'rwanda': 'RWA',
    'saint kitts and nevis': 'KNA',
    'saint lucia': 'LCA',
    'saint martin': 'MAF',
    'saint pierre and miquelon': 'SPM',
    'saint vincent and the grenadines': 'VCT',
    'samoa': 'WSM',
    'san marino': 'SMR',
    'sao tome and principe': 'STP',
    'saudi arabia': 'SAU',
    'senegal': 'SEN',
    'serbia': 'SRB',
    'seychelles': 'SYC'
```

```
 'seychelles': 'SYC',
 'sierra leone': 'SLE',
 'singapore': 'SGP',
 'sint maarten': 'SXM',
 'slovakia': 'SVK',
 'slovenia': 'SVN',
 'solomon islands': 'SLB',
 'somalia': 'SOM',
 'south africa': 'ZAF',
 'south sudan': 'SSD',
 'spain': 'ESP',
 'sri lanka': 'LKA',
 'sudan': 'SDN',
 'suriname': 'SUR',
 'swaziland': 'SWZ',
 'sweden': 'SWE',
 'switzerland': 'CHE',
 'syria': 'SYR',
 'taiwan': 'TWN',
 'tajikistan': 'TJK',
 'tanzania': 'TZA',
 'thailand': 'THA',
 'timor-leste': 'TLS',
 'togo': 'TGO',
 'tonga': 'TON',
 'trinidad and tobago': 'TTO',
 'tunisia': 'TUN',
 'turkey': 'TUR',
 'turkmenistan': 'TKM',
 'tuvalu': 'TUV',
 'uganda': 'UGA',
 'ukraine': 'UKR',
 'united arab emirates': 'ARE',
 'united kingdom': 'GBR',
 'united states': 'USA',
 'uruguay': 'URY',
 'uzbekistan': 'UZB',
 'vanuatu': 'VUT',
 'venezuela': 'VEN',
 'vietnam': 'VNM',
 'virgin islands': 'VGB',
 'west bank': 'WBG',
 'yemen': 'YEM',
 'zambia': 'ZMB',
 'zimbabwe': 'ZWE'}
```

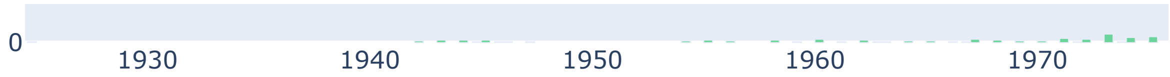
```
## countries
from collections import Counter
colorscale = ["#f7fbff", "#ebf3fb", "#deebf7", "#d2e3f3", "#c6dbef", "#b3d2e9", "#9ecae1",
    "#85bcdb", "#6baed6", "#57a0ce", "#4292c6", "#3082be", "#2171b5", "#1361a9",
    "#08519c", "#0b4083", "#08306b"
```

```
]

def geoplot(ddf):
    country_with_code, country = {}, {}
    shows_countries = ", ".join(ddf['country'].dropna()).split(", ")
    for c,v in dict(Counter(shows_countries)).items():
        code = ""
        if c.lower() in country_codes:
            code = country_codes[c.lower()]
        country_with_code[code] = v
        country[c] = v

    data = [dict(
            type = 'choropleth',
            locations = list(country_with_code.keys()),
            z = list(country_with_code.values()),
            colorscale = [[0,"rgb(5, 10, 172)"],[0.65,"rgb(40, 60, 190)"],[0.75,"rgb(70, 1
                        [0.80,"rgb(90, 120, 245)"],[0.9,"rgb(106, 137, 247)"],[1,"rgb(220,
            autocolorscale = False,
            reversescale = True,
            marker = dict(
                line = dict (
                    color = 'gray',
                    width = 0.5
                ) ),
            colorbar = dict(
                autotick = False,
                title = ''),
          ) ]

    layout = dict(
        title = '',
        geo = dict(
            showframe = False,
            showcoastlines = False,
            projection = dict(
                type = 'Mercator'
            )
        )
    )

    fig = dict( data=data, layout=layout )
    iplot( fig, validate=False, filename='d3-world-map' )
    return country

country_vals = geoplot(df)
tabs = Counter(country_vals).most_common(25)

labels = [_[0] for _ in tabs][::-1]
values = [_[1] for _ in tabs][::-1]
trace1 = go.Bar(y=labels, x=values, orientation="h", name="", marker=dict(color="#a678de")
trace1 = go.Bar(y=labels, x=values, orientation="h", name="", marker=dict(color="#a678de"),

data = [trace1]
layout = go.Layout(title="Countries with most content", height=700, legend=dict(x=0.1, y=1.
fig = go.Figure(data, layout=layout)
fig.show()
```



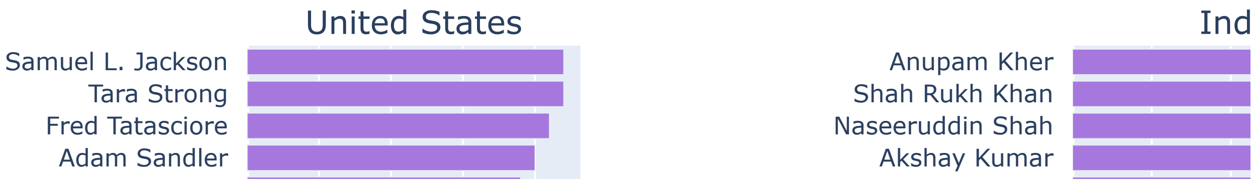


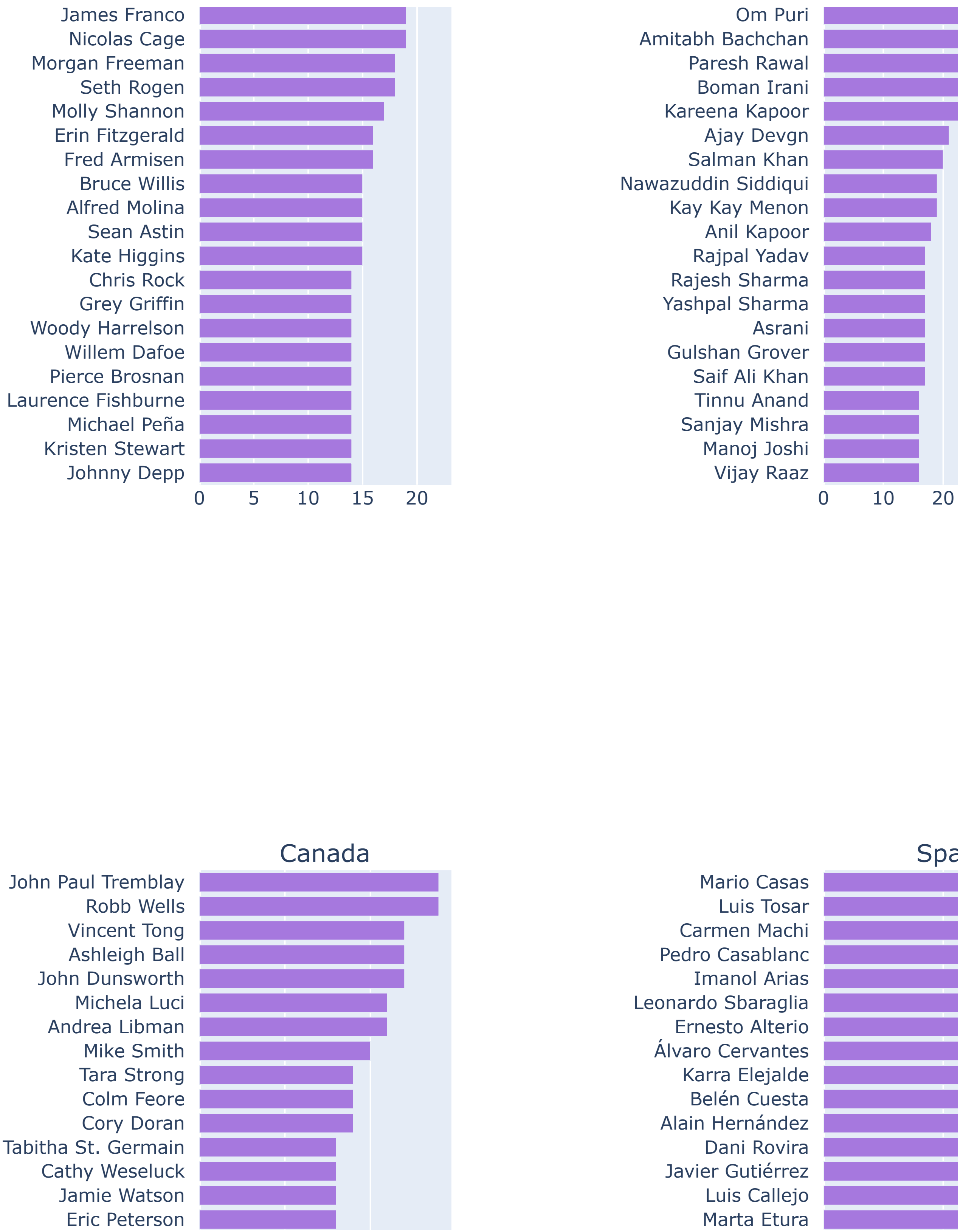
```
col = "rating"

vc1 = d1[col].value_counts().reset_index()
vc1 = vc1.rename(columns = {col : "count", "index" : col})
vc1['percent'] = vc1['count'].apply(lambda x : 100*x/sum(vc1['count']))
vc1 = vc1.sort_values(col)

vc2 = d2[col].value_counts().reset_index()
vc2 = vc2.rename(columns = {col : "count", "index" : col})
vc2['percent'] = vc2['count'].apply(lambda x : 100*x/sum(vc2['count']))
vc2 = vc2.sort_values(col)

trace1 = go.Bar(x=vc1[col], y=vc1["count"], name="TV Shows", marker=dict(color="#a678de"))
trace2 = go.Bar(x=vc2[col], y=vc2["count"], name="Movies", marker=dict(color="#6ad49b"))
data = [trace1, trace2]
layout = go.Layout(title="Content added over the years", legend=dict(x=0.1, y=1.1, orientat
fig = go.Figure(data, layout=layout)
fig.show()
```



```
print('Top Actors on Netflix with Most Movies')
def country_trace(country, flag = "movie"):
    df["from_us"] = df['country'].fillna("").apply(lambda x : 1 if country.lower() in x.low
    small = df[df["from_us"] == 1]
    if flag == "movie":
        small = small[small["duration"] != ""]
    else:
        small = small[small["season_count"] != ""]
    cast = ", ".join(small['cast'].fillna("")).split(", ")
    tags = Counter(cast).most_common(25)
    tags = [_ for _ in tags if "" != _[0]]

    labels, values = [_[0]+"  " for _ in tags], [_[1] for _ in tags]
    trace = go.Bar(y=labels[::-1], x=values[::-1], orientation="h", name="", marker=dict(co
    return trace

from plotly.subplots import make_subplots
traces = []
titles = ["United States", "","India","", "United Kingdom", "Canada","", "Spain","", "Japa
for title in titles:
    if title != "":
        traces.append(country_trace(title))

fig = make_subplots(rows=2, cols=5, subplot_titles=titles)
fig.add_trace(traces[0], 1,1)
fig.add_trace(traces[1], 1,3)
fig.add_trace(traces[2], 1,5)
fig.add_trace(traces[3], 2,1)
fig.add_trace(traces[4], 2,3)
fig.add_trace(traces[5], 2,5)

fig.update_layout(height=1200, showlegend=False)
fig.show()
```

```
Top Actors on Netflix with Most Movies
```

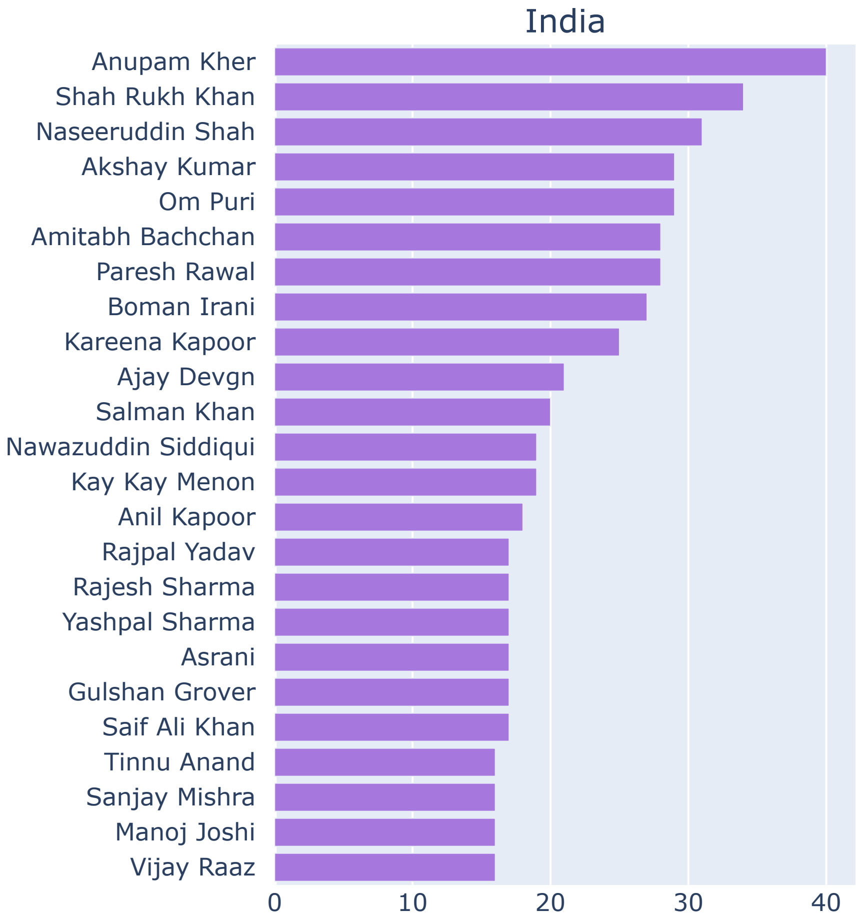




```
print('Top Actors on Netflix with Most TV Shows')
traces = []
titles = ["India","", "United Kingdom"]
for title in titles:
    if title != "":
        traces.append(country_trace(title, flag="tv_shows"))
```

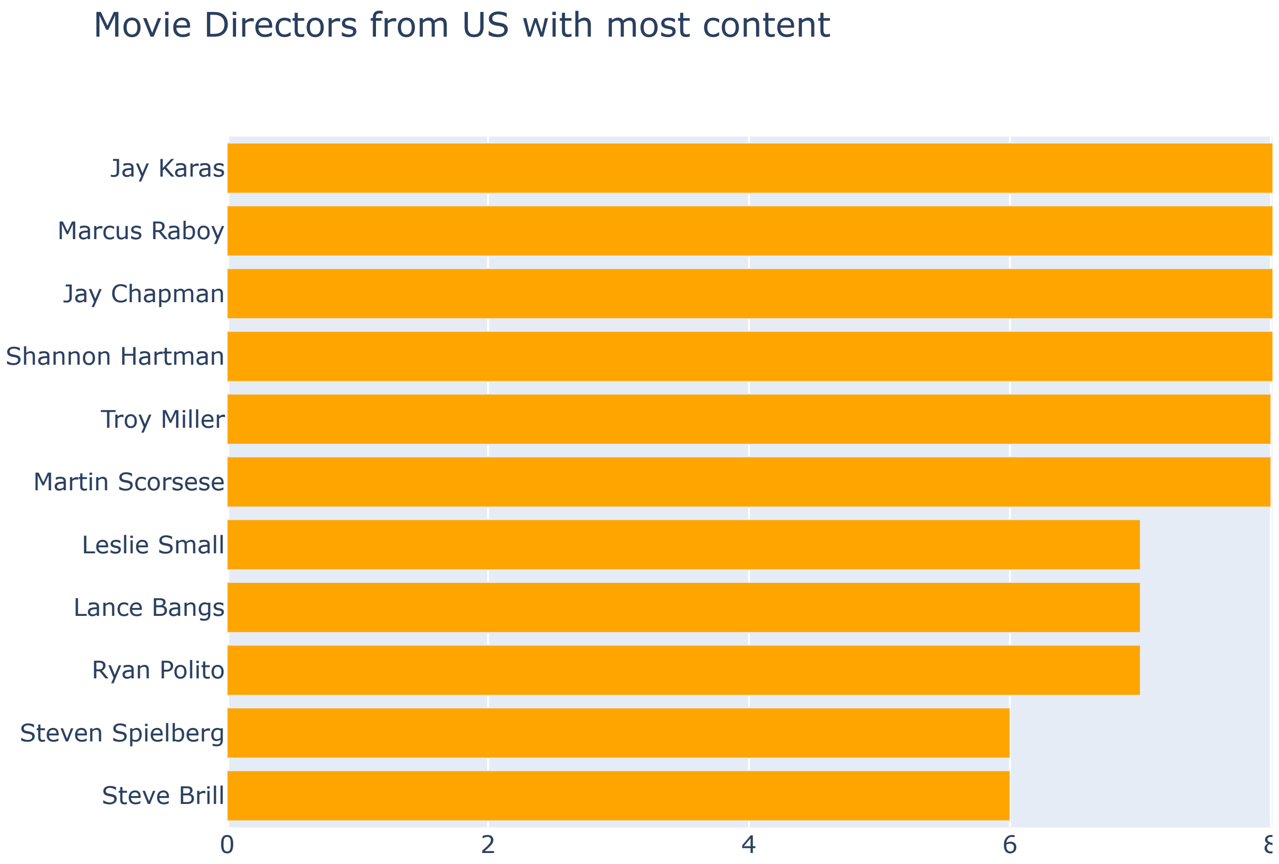
```
fig = make_subplots(rows=1, cols=3, subplot_titles=titles)
fig.add_trace(traces[0], 1,1)
fig.add_trace(traces[1], 1,3)

fig.update_layout(height=600, showlegend=False)
fig.show()
```

```
     Top Actors on Netflix with Most TV Shows
```



```
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(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="or

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



```
print('Standup Comedies by Jay Karas')
tag = "jay karas"
df["relevant"] = df['director'].fillna("").apply(lambda x : 1 if tag in x.lower() else 0)
small = df[df["relevant"] == 1]
small[['title', 'release_year', 'listed_in']]
```

```
Standup Comedies by Jay Karas
```

	title	release_year	listed_in
2695	The Main Event	2020	Children & Family Movies, Comedies, Sports Movies

3646	Demetri Martin: The Overthinker	2018	Stand-Up Comedy
3733	Adam Devine: Best Time of Our Lives	2019	Stand-Up Comedy
4803	Bill Burr: You People Are All the Same	2012	Stand-Up Comedy
4863	Ali Wong: Hard Knock Wife	2018	Stand-Up Comedy
5086	Tom Segura: Disgraceful	2018	Stand-Up Comedy
5230	Christina P: Mother Inferior	2017	Stand-Up Comedy
5622	Bill Burr: Walk Your Way Out	2017	Stand-Up Comedy
5808	Jeff Foxworthy and Larry the Cable Guy: We've ...	2016	Stand-Up Comedy
5817	Jim Gaffigan: Mr. Universe	2012	Stand-Up Comedy
5847	Ali Wong: Baby Cobra	2016	Stand-Up Comedy
5875	Tom Segura: Mostly Stories	2016	Stand-Up Comedy
5894	Anjelah Johnson: Not Fancy	2015	Stand-Up Comedy

```
print('Standup Comedies')
tag = "Stand-Up Comedy"
df["relevant"] = df['listed_in'].fillna("").apply(lambda x : 1 if tag.lower() in x.lower()
small = df[df["relevant"] == 1]
small[small["country"] == "United States"][["title", "country","release_year"]].head(10)
```

```
Standup Comedies by Jay Karas
```

	title	country	release_year
359	The Original Kings of Comedy	United States	2000
511	Chelsea	United States	2017
826	Bo Burnham: Inside	United States	2021
1189	Nate Bargatze: The Greatest Average American	United States	2021
1191	The Fluffy Movie	United States	2014
1278	Brian Regan: On the Rocks	United States	2021
1352	Tiffany Haddish Presents: They Ready	United States	2021
1450	Eddie Murphy: Raw	United States	1987
1502	London Hughes: To Catch a D*ck	United States	2020
1530	Schulz Saves America	United States	2020

```
tag = "Stand-Up Comedy"
df["relevant"] = df['listed_in'].fillna("").apply(lambda x : 1 if tag.lower() in x.lower()
small = df[df["relevant"] == 1]
small[small["country"] == "India"][["title", "country","release_year"]].head(10)
```

	title	country	release_year
1542	Vir Das: Outside In - The Lockdown Special	India	2020
2458	Kenny Sebastian: The Most Interesting Person i...	India	2020
2644	Yours Sincerely, Kanan Gill	India	2020
2765	Ladies Up	India	2019
2869	Amit Tandon: Family Tandoncies	India	2019
2987	Vir Das: For India	India	2020
5371	Aditi Mittal: Things They Wouldn't Let Me Say	India	2017
6825	Gangs of Hassepur	India	2014
7453	Midnight Misadventures With Mallika Dua	India	2018