EDA on Netflix Data

1. Introduction

Founded in 1997, Netflix is American subscription video over-the-top streaming service and production company with over 192M subscribers worldwide! As of mid-2021, Netflix has over 8000 movies and TV shows available from a wide variety of genres on their platform.

So, for content producers to promote their content more effectively, it is crucial to understand the trends of the content that Netflix adds to their catalog. What has been the trend in Netflix streaming in recent years? What type of content is popular on Netflix? This project aims to conduct an exploratory data analysis on Netflix's content library that would help the media company make informed business decisions.

Business Task:

Conduct data analysis on the trends of the content that Netflix adds to their catalog.

Business Problem:

Investigate what content should be bought or produced for 'Youflix', a hypothetical streaming service.

2. Data Import and Check

The dataset for this analysis was downloaded from Netflix Movies and TV Shows Data uploaded Kaggle via Public Domain. This dataset was made available through Shivam Bansal on Kaggle.

Libraries needed

In [1]:

```
# Libraries needed
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objs as go
import plotly.offline as py
import plotly.express as px
import re
%matplotlib inline
```

In [2]:

```
import warnings
warnings.filterwarnings('ignore')
from IPython.display import set_matplotlib_formats
set_matplotlib_formats("retina")
```

Data load and check

In [3]:

```
netflix = pd.read_csv(r"C:₩Users₩eunbi₩Desktop₩DS₩EDA on Netflix₩netflix_titles.csv")
netflix.shape
```

Out[3]:

```
(8807, 12)
```

In [4]:

```
netflix.nunique()
```

Out[4]:

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

I loaded the CSV file using Pandas Library and named the imported dataset as netflix. The dataset contains 8807 different movie/TVshow data: 6126 Movie data and 2664 TV Show data. The columns consist of 12 typical movie/TVshows descriptions, such as type, title, director, cast, country, date added and release year.

Let's look at the first five data.

First 5 rows

In [5]:

```
netflix.head()
```

Out[5]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries
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
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...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV
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 ...

We can see that there are NaN values in some columns.

3. Data Pre-processing

Missing data

In [6]:

```
# total missing values
netflix.isnull().sum().sum()
```

Out[6]:

```
4307
```

In [7]:

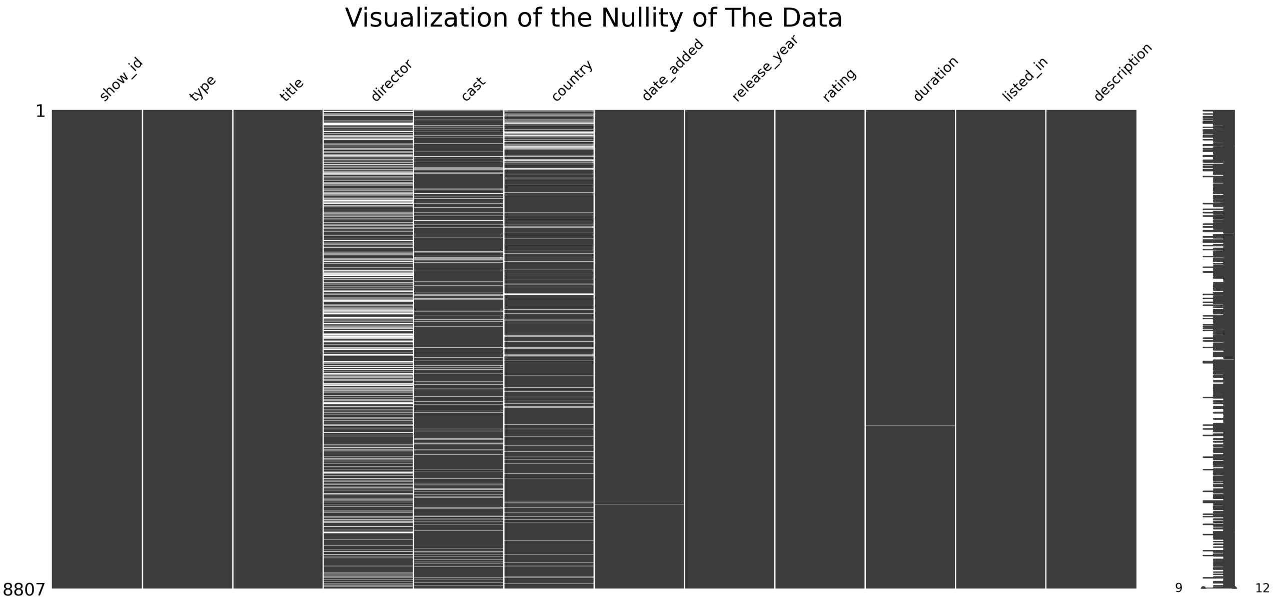
```
# missing values
netflix.isnull().sum()
```

Out[7]:

```
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
```

In [8]:

```python
# visualization of the nullity
import missingno as msno

msno.matrix(netflix)
plt.title("Visualization of the Nullity of The Data", fontsize=30)
plt.show()
```



There is a total of 4307 missing values in the data. The visualization above shows that there are a lot of missing values in "director", "cast" and "country" columns. "data_added", "rating" and "duration" also have a few missing values. The easiest way to get rid of them would be to delete the rows with the missing data. However, this is not the best option for this EDA. Since the number of missing values is large, dropping them would result in a significant loss of information. For the columns containing the majority of null values, such as "director", "cast", "country" and "rating", I choose to fill each missing value with 'unavailable' using fillna() function. "date_added", "duration" and "rating" can be dropped from the dataset because they contain an insignificant portion of the data.

In [9]:

```python
# replace missing values with 'unavailable'
netflix["director"] = netflix["director"].fillna("Unavailable")
netflix["cast"] = netflix["cast"].fillna("Unavailable")
netflix["country"] = netflix["country"].fillna("Unavailable")
netflix.dropna(subset=["date_added", "duration", "rating"], inplace=True)
```

In [10]:

```python
# missing values after cleaning
netflix.isnull().sum()
```

Out[10]:

```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

In [11]:

```
# data shape
netflix.shape
```

Out[11]:

```
(8790, 12)
```

Finally, we can see that there are no more missing values in the data. The data size is reduced to 8790.

Dates data

The datatype of "date_added" is object type. Transforming this variable to datetime data will be helpful. I will extract year and month from "date added" and add new columns "year_added" and "month_added".

In [12]:

```
# convert datatype to datetime
netflix["date_added"] = pd.to_datetime(netflix["date_added"])
# new column 'year_added'
netflix["year_added"] = netflix["date_added"].dt.year
# new column 'month_added'
netflix["month_added"] = netflix["date_added"].dt.month
netflix.head(1)
```

Out[12]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Unavailable	United States	2021-09-25	2020	PG-13	90 min	Documentaries

Unnesting columns

Some columns, such as "cast" and "country", contain a list of string values. Unnesting these columns will be possible. First, I want to create separate lines for each cast member in a movie.

In [13]:

```
# unnesting "cast" columns
constraint=netflix["cast"].apply(lambda x: str(x).split(', ')).tolist()
actors_un = pd.DataFrame(constraint,index=netflix["title"])
actors_un = actors_un.stack()
actors_un = pd.DataFrame(actors_un.reset_index())
actors_un.rename(columns={0:'Actors'},inplace=True)
actors_un.drop(['level_1'],axis=1,inplace=True)
actors_un.head()
```

Out[13]:

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

A lot of contents are listed in multiple genres. I want to create separate lines for each genre as well.

In [14]:

```
# unnesting "listed_in" columns
constraint2 = netflix["listed_in"].apply(lambda x: str(x).split(', ')).tolist()
genres_un = pd.DataFrame(constraint2,index=netflix["title"])
genres_un = genres_un.stack()
genres_un = pd.DataFrame(genres_un.reset_index())
genres_un.rename(columns={0:"Genre"},inplace=True)
genres_un.drop(["level_1"],axis=1,inplace=True)
genres_un.head()
```

Out[14]:

	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

Lastly, I'm going to create separate lines for each country in a movie.

In [15]:

```
# unnesting "country" columns
constraint3 = netflix["country"].apply(lambda x: str(x).split(', ')).tolist()
country_un = pd.DataFrame(constraint3, index=netflix["title"])
country_un = country_un.stack()
country_un = pd.DataFrame(country_un.reset_index())
country_un.rename(columns={0:"Country"}, inplace=True)
country_un.drop(["level_1"], axis=1, inplace=True)
country_un.head()
```

Out[15]:

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

In [16]:

```
# merge unnested actors data with unnested genres data
netflix_un = actors_un.merge(genres_un, on=["title"], how="inner")
# merge the above data with unnested  countries data
netflix_un = netflix_un.merge(country_un, on=["title"], how="inner")
# merge the anove data with the original data
cols = ["show_id", "director", "type", "title", "date_added", "year_added", "month_added", "release_yea
netflix_new = netflix_un.merge(netflix[cols], on=["title"], how="left")
netflix_new.head()
```

Out[16]:

	title	Actors	Genre	Country	show_id	director	type	date_added	year_added	month_added
0	Dick Johnson Is Dead	Unavailable	Documentaries	United States	s1	Kirsten Johnson	Movie	2021-09-25	2021	9
1	Blood & Water	Ama Qamata	International TV Shows	South Africa	s2	Unavailable	TV Show	2021-09-24	2021	9
2	Blood & Water	Ama Qamata	TV Dramas	South Africa	s2	Unavailable	TV Show	2021-09-24	2021	9
3	Blood & Water	Ama Qamata	TV Mysteries	South Africa	s2	Unavailable	TV Show	2021-09-24	2021	9
4	Blood & Water	Khosi Ngema	International TV Shows	South Africa	s2	Unavailable	TV Show	2021-09-24	2021	9

Duration

I will convert 'duration' to integer type. I will remove 'min' and 'Seasons' and convert the data type.

In [17]:

```
netflix_new.loc[netflix_new['type'] == 'Movie', 'duration'].unique()
```

Out[17]:

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

In [18]:

```
netflix_new.loc[netflix_new['type'] == 'Movie', 'duration'] = netflix_new.loc[netflix_new['type'] == '
```

In [19]:

```
netflix_new.loc[netflix_new['type'] == 'TV Show', 'duration'].unique()
```

Out[19]:

```
array(['2 Seasons', '1 Season', '9 Seasons', '4 Seasons', '5 Seasons',
       '3 Seasons', '6 Seasons', '7 Seasons', '10 Seasons', '8 Seasons',
       '17 Seasons', '13 Seasons', '15 Seasons', '12 Seasons',
       '11 Seasons'], dtype=object)
```

In [20]:

```
netflix_new.loc[netflix_new['type'] == 'TV Show',
                'duration'] = netflix_new.loc[netflix_new['type'] == 'TV Show', 'duration'].str[:-7]
netflix_new.loc[netflix_new['type'] == 'TV Show',
                'duration'] = netflix_new.loc[netflix_new['type'] == 'TV Show', 'duration'].str.rstrip
```

In [21]:

```
netflix_new.loc[netflix_new['type'] == 'TV Show', 'duration'].unique()
```

Out[21]:

```
array(['2', '1', '9', '4', '5', '3', '6', '7', '10', '8', '17', '13',
       '15', '12', '11'], dtype=object)
```

In [22]:

```
netflix_new['duration'] = netflix_new['duration'].astype('int')
```

New column

I want to add a new column about targeting age of contents based on ratings.

In [23]:

```
# new column 'target_age'
netflix_new['target_age'] = ''

netflix_new.loc[netflix_new['rating']=='TV-Y', 'target_age'] = 'Kids'
netflix_new.loc[netflix_new['rating']=='G', 'target_age'] = 'Kids'
netflix_new.loc[netflix_new['rating']=='TV-G', 'target_age'] = 'Kids'

netflix_new.loc[netflix_new['rating']=='TV-Y7', 'target_age'] = 'Older Kids'
netflix_new.loc[netflix_new['rating']=='TV-Y7-FV', 'target_age'] = 'Older Kids'
netflix_new.loc[netflix_new['rating']=='TV-PG', 'target_age'] = 'Older Kids'
netflix_new.loc[netflix_new['rating']=='PG', 'target_age'] = 'Older Kids'

netflix_new.loc[netflix_new['rating']=='PG-13', 'target_age'] = 'Teens'
netflix_new.loc[netflix_new['rating']=='PG-14', 'target_age'] = 'Teens'
netflix_new.loc[netflix_new['rating']=='TV-14', 'target_age'] = 'Teens'
netflix_new.loc[netflix_new['rating']=='NC-17', 'target_age'] = 'Teens'

netflix_new.loc[netflix_new['rating']=='R', 'target_age'] = 'Adults'
netflix_new.loc[netflix_new['rating']=='TV-MA', 'target_age'] = 'Adults'

netflix_new.loc[netflix_new['rating']=='NR', 'target_age'] = 'Not rated'
netflix_new.loc[netflix_new['rating']=='UR', 'target_age'] = 'Not rated'
```

I also want to categorize 'release_year' into decade groups.

In [24]:

```
# new column 'release_decade'
netflix_new['release_decade'] = ''
netflix_new.loc[netflix_new['release_year'] < 1970, 'release_decade'] = '~1970s'
netflix_new.loc[(netflix_new['release_year'] >= 1970) & (netflix_new['release_year'] < 1980), 'release
netflix_new.loc[(netflix_new['release_year'] >= 1980) & (netflix_new['release_year'] < 1990), 'release
netflix_new.loc[(netflix_new['release_year'] >= 1990) & (netflix_new['release_year'] < 2000), 'release
netflix_new.loc[(netflix_new['release_year'] >= 2000) & (netflix_new['release_year'] < 2010), 'release
netflix_new.loc[(netflix_new['release_year'] >= 2010) & (netflix_new['release_year'] < 2020), 'release
netflix_new.loc[(netflix_new['release_year'] >= 2020), 'release_decade'] = '2020s'
```

Lastly, I will add 'content_age', the difference between the year added and release year.

In [25]:

```
netflix_new['content_age'] = netflix_new['year_added'] - netflix_new['release_year']
```

In [26]:

```
netflix_new.Country.nunique()
```

Out[26]:

```
128
```

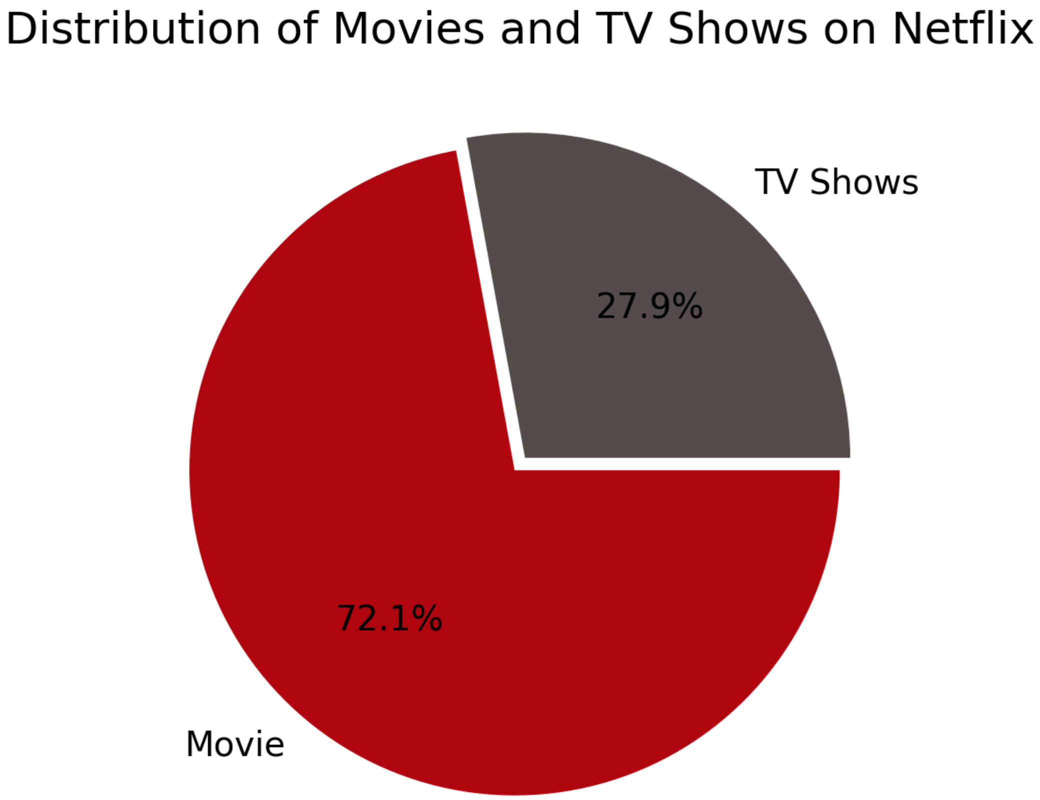
Now, it's finally time to explore the data and create visualizations.

4. Visualizations

Content Preferences on Netflix: Movies vs. TV Shows

The dataset consists of 6126 movie data and 2664 TV show data. I want to compare them with ratio. Pie chart would be a good choice for the proportions of categorical data.

In [27]:
```
distinct = netflix_new.drop_duplicates(subset = ['year_added', 'title', 'Country'], keep = 'last').res
# Calculate the ratio of Movies and TV Shows
label = ["Movie", "TV Shows"]
mtratio = distinct["type"].value_counts(normalize=True)
# Draw a pie chart indicating the percentage of Moves and TV Shows
plt.rcParams["figure.figsize"] = (13,6)
plt.pie(mtratio, labels=label, colors=['#B20710', '#564d4d'], counterclock=False,
        explode=(0.025,0.025), textprops={'fontsize': 14},
        autopct='%.1f%%')
plt.title("Distribution of Movies and TV Shows on Netflix", fontsize=18)
plt.show()
```



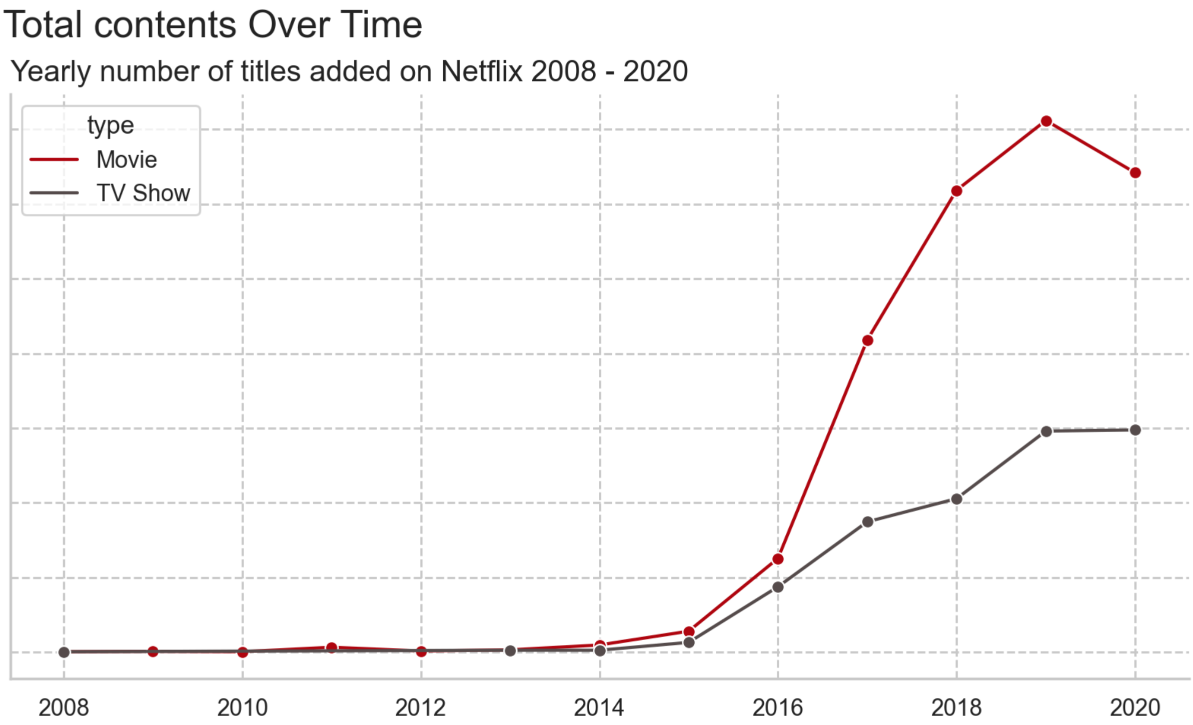
- There are more than 6000 movies and almost 3,000 TV shows on Netflix, with movies being the majority.
- There are far more movie titles (69.7%) than TV shows titles (30.3%).

Movies and TV Shows added over time

Now, I will explore the amount of content Netflix has added over the previous years. I will extract year and month from "date added" and add new columns "year_added" and "month_added".

In [28]:

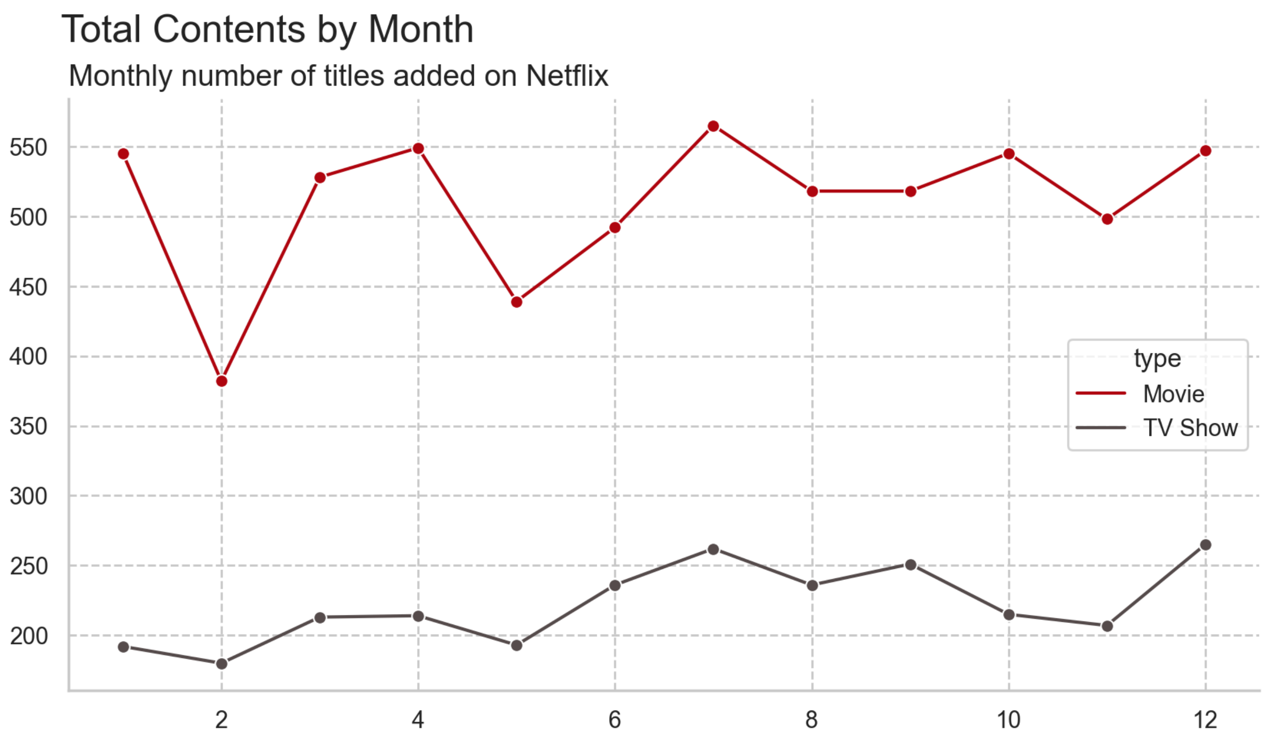
```
by_type = netflix.groupby(['year_added', 'type'])['title'].count().reset_index()
by_type = by_type[by_type['year_added'] != 2021]
sns.set(rc = {'figure.figsize':(10, 5)})
sns.set_style("whitegrid", {'grid.linestyle': '--'})
ax = sns.lineplot(by_type, x = 'year_added', y = 'title', hue = 'type',
             color = 'type', palette=['#B20710', '#564d4d'], marker='o')
plt.title('Yearly number of titles added on Netflix 2008 - 2020', fontsize = 14, loc = 'left')
plt.suptitle('Total contents Over Time', fontsize = 18, x = 0.12, y=.99, horizontalalignment='left')
plt.xlabel('')
plt.ylabel('')
sns.despine()
plt.show()
```



- Based on the plot above, we can say that Netflix started growing its business as a streaming platform from 2013. Since 2015, the amount of content added has been rapidly increasing.
- Besides, it clearly appears that Netflix has increasingly focused on TV shows rather than movies in recent years. While the number of TV show releases kept increasing, the amount of increase in the number of movie releases has reduced since 2017 and the number even dropped in 2020.

In [29]:

```
month = netflix.groupby(['month_added', 'type'])['title'].count().reset_index()
sns.lineplot(month, x = 'month_added', y = 'title', hue = 'type',
             color = 'type', palette=['#B20710', '#564d4d'], marker='o')
plt.title('Monthly number of titles added on Netflix', fontsize = 14, loc='left')
plt.suptitle('Total Contents by Month', fontsize = 18, x = 0.12, y=.99, horizontalalignment='left')
plt.xlabel('')
plt.ylabel('')
#plt.fill_between(month.loc[month['type']=='Movie', 'month_added'].values, month.loc[month['type']=='Mo
#                                                                                                     'title'].valu
#plt.fill_between(month.loc[month['type']=='TV Show', 'month_added'].values, month.loc[month['type']=='
#        'title'].values, color = '#564d4d')
sns.despine()
plt.show()
```


Total Contents by Month

Monthly number of titles added on Netflix

- Holiday seasons — December, January, and July - seem to be the best time for the release of new content on Netflix. Netflix acknowledges its customers' time-spending habits, that they tend to have more time off during those periods of the year.
- Also we can notice that Netflix adds more new Movies during the second half of the year, from June to December.

Which country produces the most content on Netflix?

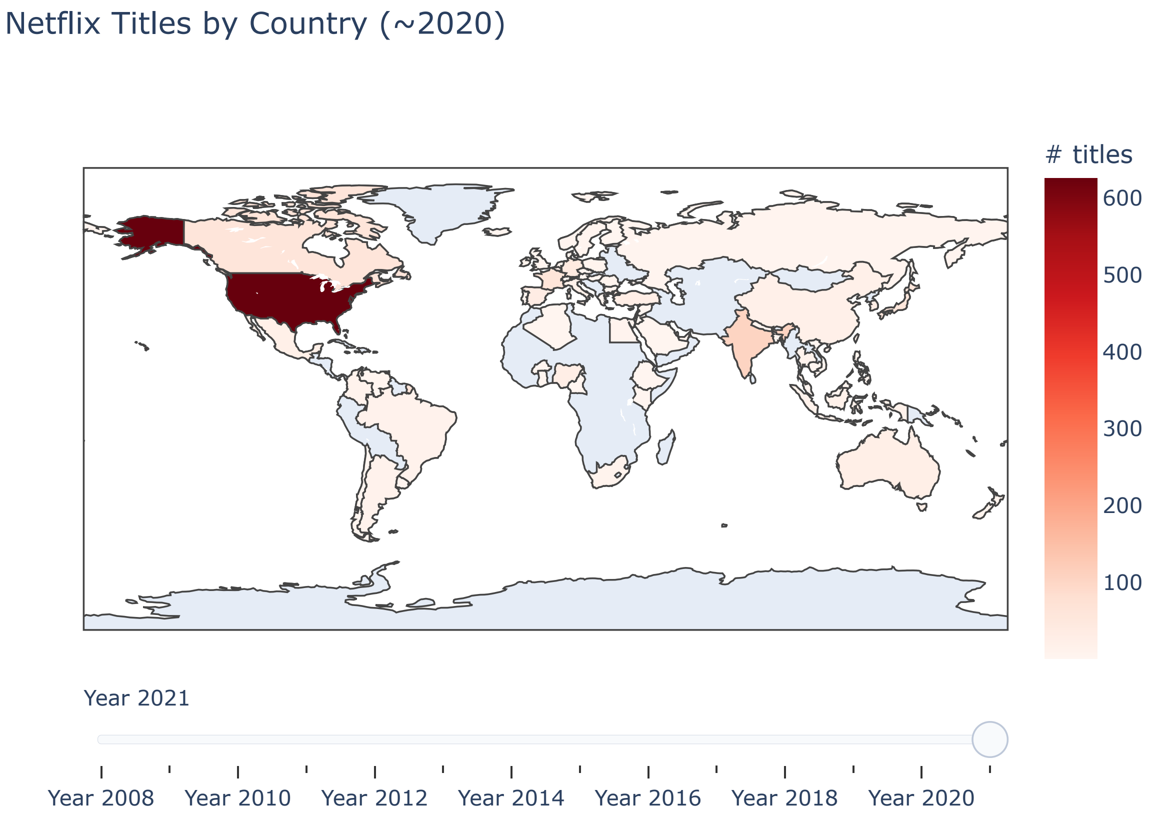
In [30]:

```
# path to geojson file
geo_path = r"C:₩Users₩eunbi₩Desktop₩DS₩--geojson₩countries.geo.json"
import json
geo_json = json.load(open(geo_path, encoding="utf-8"))
```

In [31]:

```
distinct['year_added'] = distinct['year_added'].astype('object')
count = distinct.groupby(['year_added','Country'])['title'].count().reset_index()
count.loc[count['Country'] == 'United States', 'Country'] = 'United States of America'

# minimum_year
year = 2008
count = count.copy()
data_slider = []
for year in count['year_added'].unique():
    df_segmented =  count.loc[(count['year_added']== year)]

    for col in df_segmented.columns:
        df_segmented[col] = df_segmented[col].astype(str)

    data_each_yr = dict(
                        type='choropleth',
                        locations = df_segmented['Country'],
                        z=df_segmented['title'].astype(float),
                        locationmode='country names',
                        colorscale = 'reds',
                        colorbar= {'title':'# titles'})

    data_slider.append(data_each_yr)
```

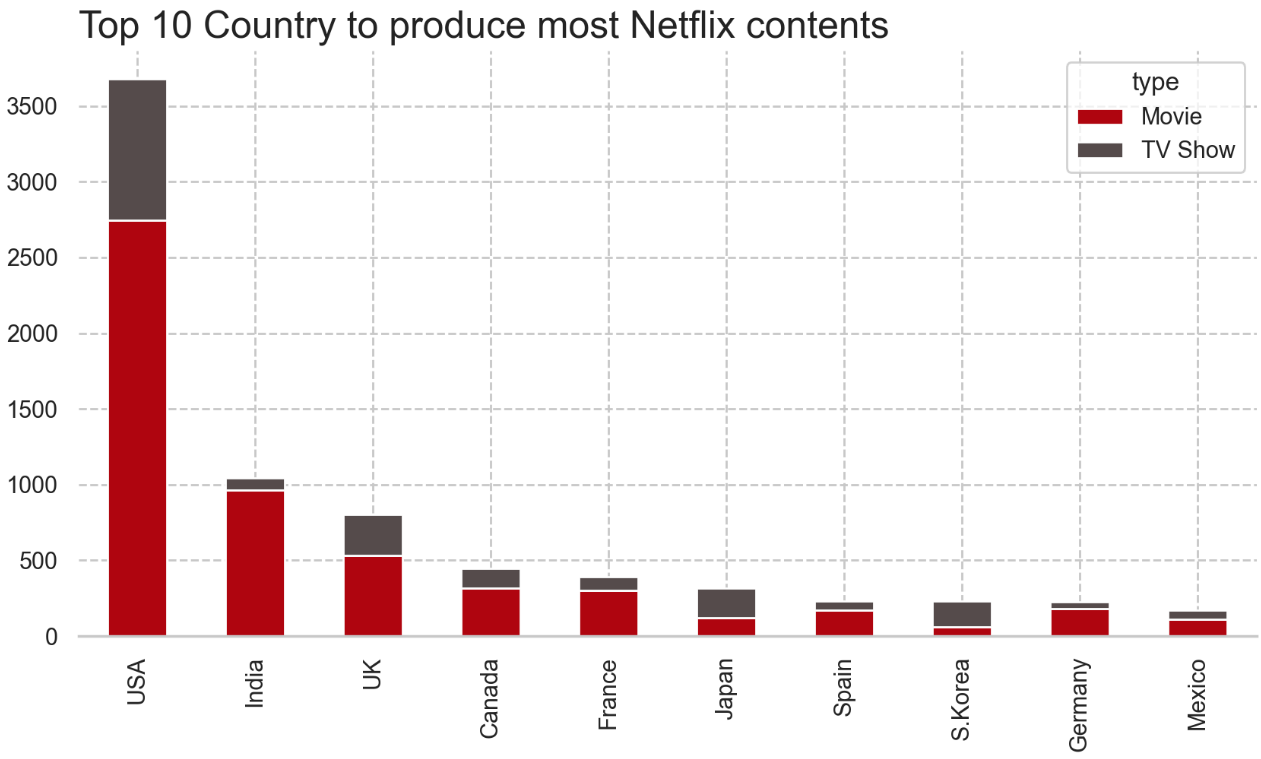
```
steps = []
for i in range(len(data_slider)):
    step = dict(method='restyle',
                args=['visible', [False] * len(data_slider)],
                label='Year {}'.format(i + 2008))
    step['args'][1][i] = True
    steps.append(step)

sliders = [dict(active=0, pad={"t": 1}, steps=steps)]

layout = dict(title ='Netflix Titles by Country (~2020)', geo=dict(scope='world'),
              sliders=sliders)

fig = go.Figure(dict(data=data_slider, layout=layout))
fig.update_layout(width=700,height=500)
py.offline.iplot(fig)
```



- It was 2010 when Netflix began expanding their markets internationally to Canada, which is geographically close and shares many cultural similarities with the United States.
- In 2016, Netflix began its worldwide expansion.

In [32]:
```
distinct.loc[distinct['Country'] == 'United States', 'Country'] = 'USA'
distinct.loc[distinct['Country'] == 'United Kingdom', 'Country'] = 'UK'
distinct.loc[distinct['Country'] == 'South Korea', 'Country'] = 'S.Korea'
country_filtered = distinct.loc[distinct["Country"]!= "Unavailable",]
countries = country_filtered.groupby(["Country", 'type'])["title"].nunique().reset_index()
countries = countries.sort_values(by="title", ascending=False).reset_index()
countries = countries.drop("index", axis=1)
countries = countries.pivot(index = 'Country', columns='type', values='title')
countries['total'] = countries['Movie'] + countries['TV Show']
countries = countries.sort_values('total', ascending = False)
countries.drop(columns = 'total', inplace = True)
countries = countries.reset_index()
sns.set_style("whitegrid", {'grid.linestyle': '--'})
```

```
plt = countries.head(10).set_index('Country').plot(kind = 'bar', stacked = True, color = ['#B20710',
sns.despine()
plt.set_ylabel('')
plt.set_xlabel('')
plt.set_title('Top 10 Country to produce most Netflix contents', fontsize = 18, loc = 'left')
plt.spines['top'].set_visible(False)
plt.spines['right'].set_visible(False)
plt.spines['left'].set_visible(False)
```



- Netflix offers internationally diverse content. The majority of Netflix contents obviously come from United States. Although India and the U.K. rank second and third among Netflix content producers, they have a significant distance behind the U.S.
- Bollywood and Hollywood, 2 big film industries in the world, have the number of Movies on Netflix outweighs the number of TV Shows.
- The gap between Movies and TV Shows from Mexico is smaller, which means the preference between Movie content and TV Show content from Mexico is more balanced.
- On the other hand, Netflix seems to invest more in TV series from East Asian countries — Japan and South Korea, while it invests more in Movies from European countries — France, Germany, the U.k. and Spain.

Ratings and Target audience

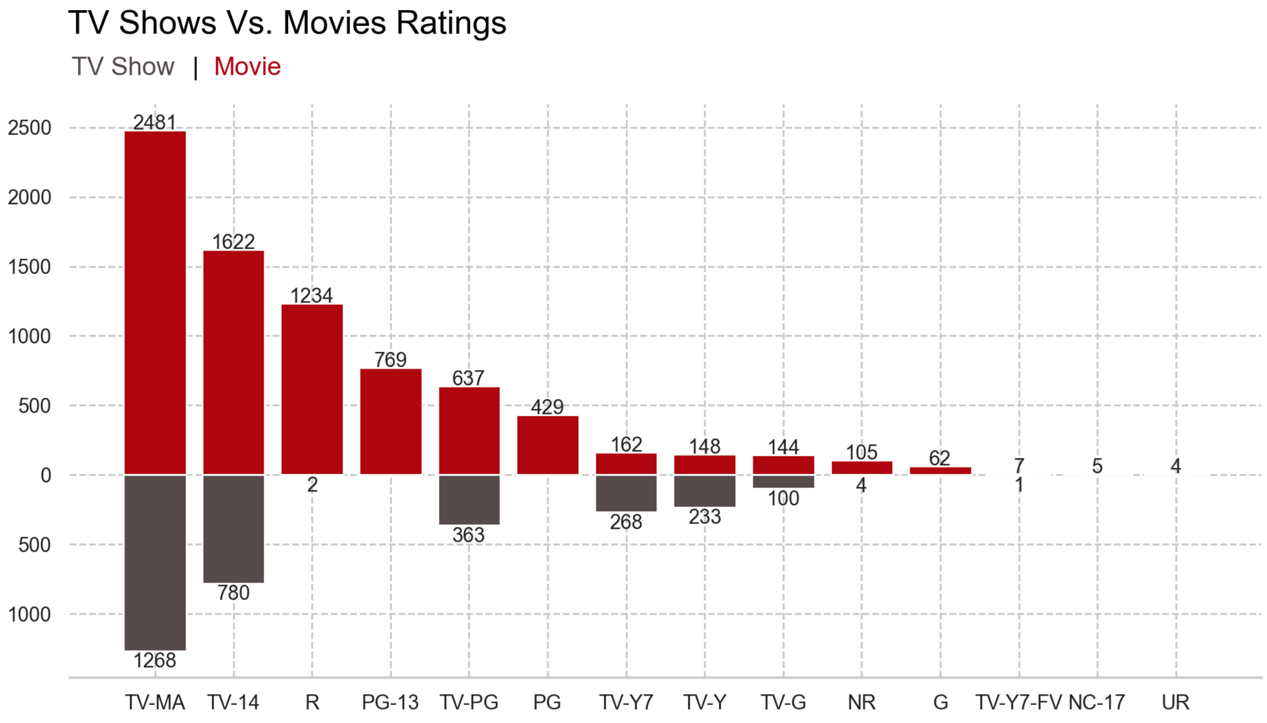
In [33]:

```
import matplotlib
import matplotlib.pyplot as plt
```

In [34]:

```
ratings_m = distinct[distinct['type']=='Movie'].groupby(['rating'])['title'].count().reset_index()
ratings_m = ratings_m.sort_values('title', ascending = False)
ratings_t = distinct[distinct['type']=='TV Show'].groupby(['rating'])['title'].count().reset_index()
ratings_t['title'] = ratings_t['title'] * (-1)

sns.set_style("whitegrid", {'grid.linestyle': '--'})
fig, ax = plt.subplots(figsize = (10, 5))
ax.bar(ratings_m['rating'], ratings_m['title'], color = '#B20710')
ax.bar(ratings_t['rating'], ratings_t['title'], color = '#564d4d')

# Formatting x labels
```

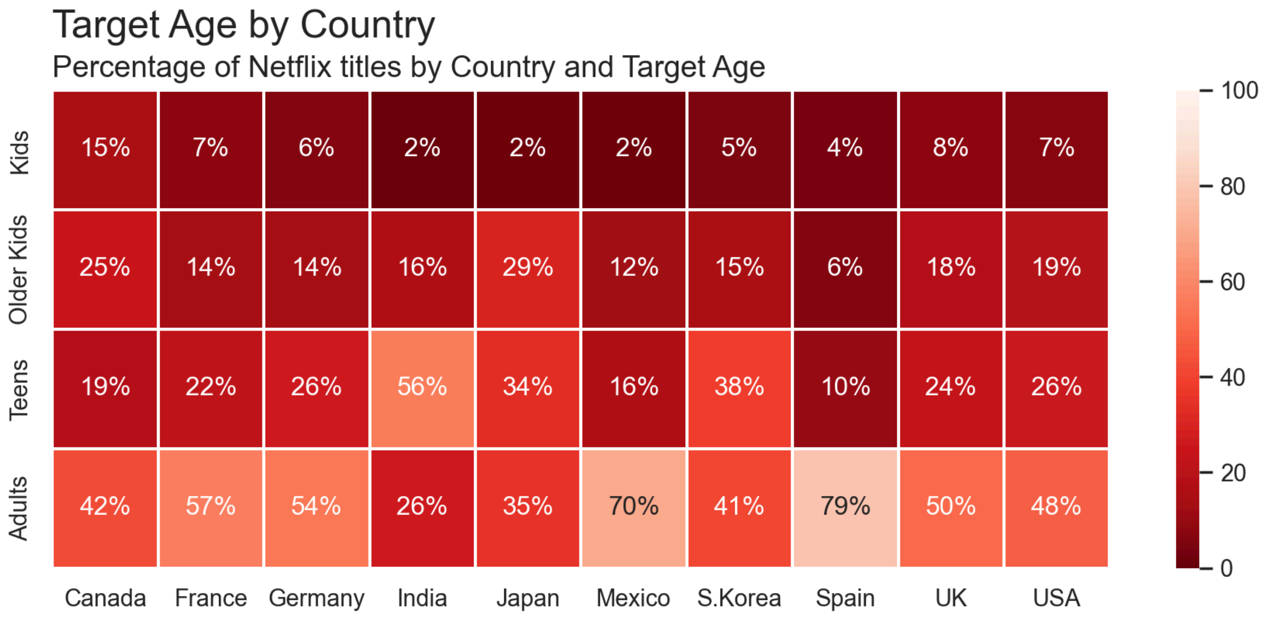
```
plt.tight_layout()
# Use absolute value for y-ticks
ticks =  ax.get_yticks()
ax.set_yticklabels([int(abs(tick)) for tick in ticks])

for bar in ax.patches:
    ax.annotate(format(bar.get_height() if bar.get_height() > 0 else  bar.get_height()*(-1), '.0f'),
                (bar.get_x() + bar.get_width() / 2, bar.get_height()-60 if bar.get_height() >= 0 else
                ha='center', va='center',
                   size=11, xytext=(0, 8),
                   textcoords='offset points')
fig.text(.24, 1.06, "TV Shows Vs. Movies Ratings", ha="center", va="bottom", size = 18, color = "black
fig.text(.115, 1, "TV Show", ha="center", va="bottom", size=14, color = "#564d4d")
fig.text(.17, 1, "|", ha="center", va="bottom", size=14, color = "black")
fig.text(.21, 1, "Movie", ha="center", va="bottom", size=14, color = "#B20710")
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['left'].set_visible(False)
```



- Some ratings, such as PG-13, PG, NC-17 and UR, are only applicable to Movies.
- The most common rating for both Movie and TV Show is TV-MA (for adult groups) and TV-14 (for teen groups).

In [35]:

```
top_countries = ['USA', 'India', 'UK', 'Canada', 'France', 'Japan', 'Spain', 'S.Korea', 'Germany', 'Mex
filtered = country_filtered[country_filtered['Country'].isin(top_countries)]
age_agg = filtered[filtered['target_age']!='Not rated'].groupby(['Country', 'target_age']).agg({'title
age_agg = age_agg.groupby(level=0).apply(lambda x:
                                                  100 * x / float(x.sum()))
age_agg.reset_index(inplace = True)
age_pivot = age_agg.pivot(index='target_age', columns = 'Country', values = 'title' )
age_agg.target_age = age_agg.target_age.astype('category')
age_pivot.index = pd.CategoricalIndex(age_pivot.index, categories= ['Kids', 'Older Kids', 'Teens', 'Adu
age_pivot.sort_index(level=0, inplace=True)
sns.set(rc = {'figure.figsize':(11, 4)})
hm = sns.heatmap(age_pivot,annot=True, fmt = '.0f', vmin = 0, vmax=100, cmap = 'Reds_r', linewidths=.
hm.set(ylabel=None, xlabel=None)
hm.set_title("Percentage of Netflix titles by Country and Target Age", fontsize = 14, loc='left')
plt.suptitle("Target Age by Country", fontsize = 18, x = 0.125, y=1.01, horizontalalignment='left')
for t in hm.texts:
    t.set_text(t.get_text() + "%")
```



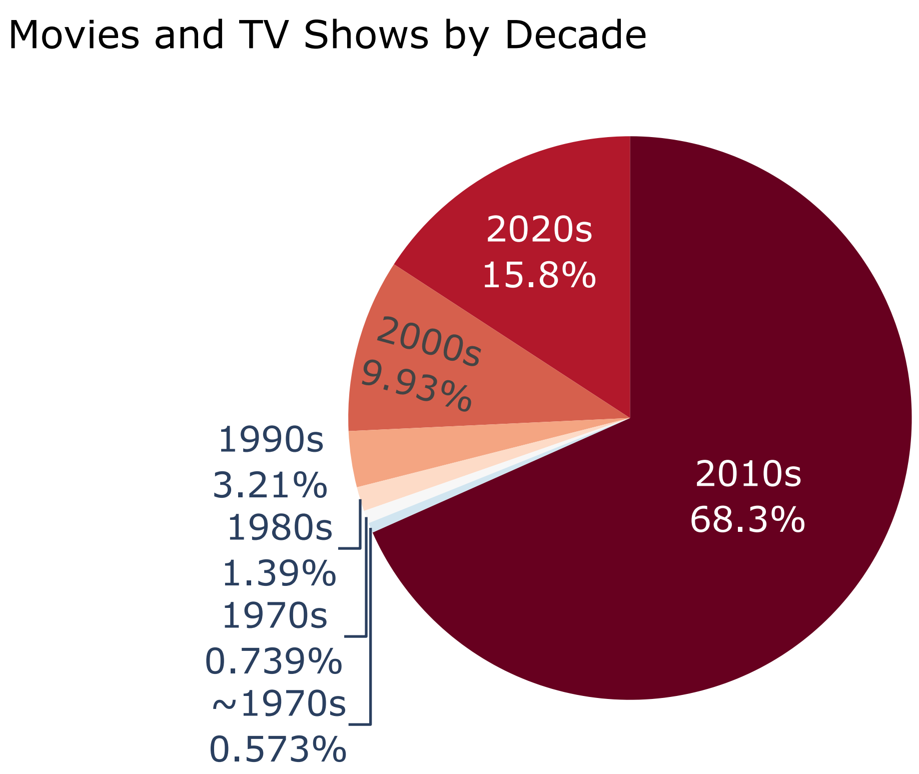
	Canada	France	Germany	India	Japan	Mexico	S.Korea	Spain	UK	USA
Kids	15%	7%	6%	2%	2%	2%	5%	4%	8%	7%
Older Kids	25%	14%	14%	16%	29%	12%	15%	6%	18%	19%
Teens	19%	22%	26%	56%	34%	16%	38%	10%	24%	26%
Adults	42%	57%	54%	26%	35%	70%	41%	79%	50%	48%

- Majority of Netflix content from European countries are made for adult viewers.
- Movies and shows from the USA and the UK target more on adult, while those produced in India target more on teenager group of audience.
- The largest groups of target audience are adults and teenagers.
- On the other hand, east asian countries, such as Japan and South Korea target on both teenager and adult groups in balance.
- The main libraries on Netflix for older kids are Canada and Japan.

How old are the content on Netflix?

In [36]:

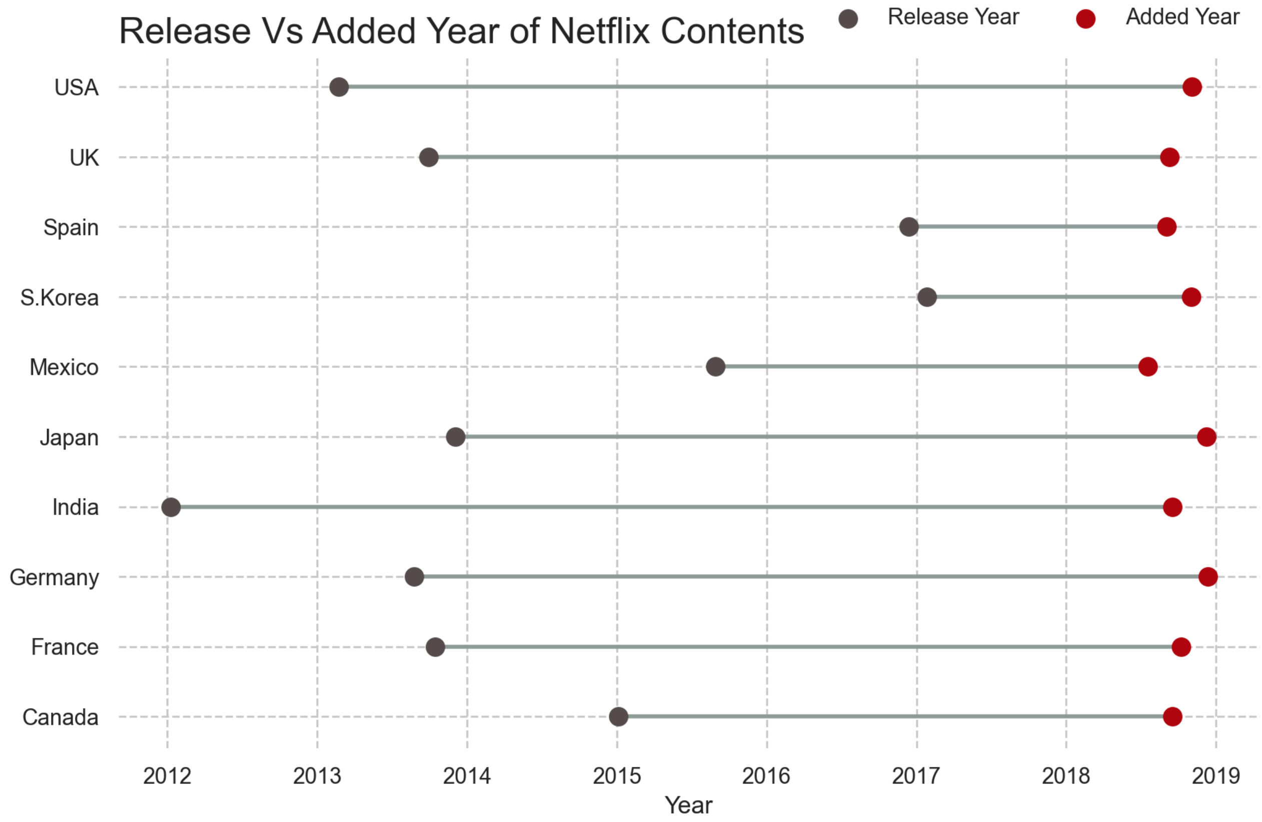
```
decade_ratio = distinct["release_decade"].value_counts(normalize=True)
decades = decade_ratio.index.to_list()
decade_ratio = decade_ratio.reset_index().rename(columns = {'index':'decade', 'release_decade': 'Percen
decade_ratio['Percentage of titles'] = decade_ratio['Percentage of titles']*100
fig = px.pie(decade_ratio, values="Percentage of titles", names="decade",
                        title="<span style='font-size:22px;color:black;'>Movies and TV Shows by Decade<
             color_discrete_sequence=px.colors.sequential.RdBu)
fig = fig.update_traces(textposition='auto', textinfo='percent+label', showlegend=False)
fig = fig.update_layout(uniformtext_minsize=20, uniformtext_mode="show")
fig.update_layout(width=700,height=500)
fig
```



- This pie chart shows that almost 70% of Netflix contents are released in 2010s.
- The relatively latest contents (2020s) are only 16 per cent.
- The less recent the year of release, the fewer the number of movies are.

In [37]:

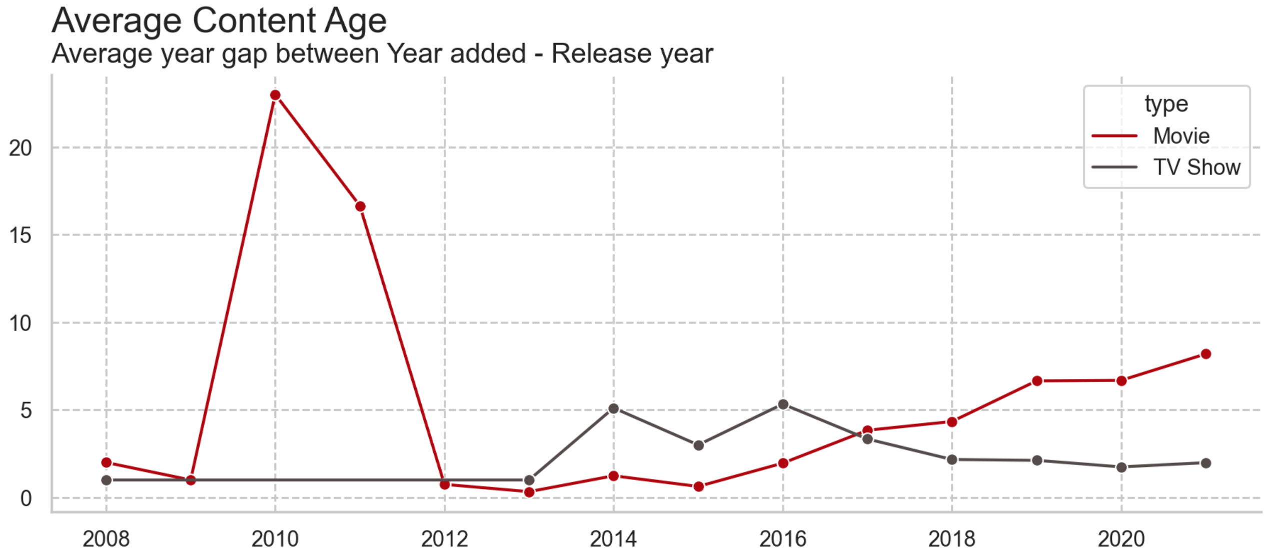
```
release = filtered[['release_year', 'Country']].groupby(['Country'])['release_year'].mean().reset_inde
added = filtered[['year_added', 'Country']].groupby(['Country'])['year_added'].mean().reset_index()
added = added.reset_index()
added['index'] = added['index'] +1
plt.figure(figsize = (9, 6))
sns.set_style('whitegrid', {'grid.linestyle':'--'})

ax = plt.axes(frameon=False)
plt.hlines(y = added['index'], xmin = release['release_year'], xmax=added['year_added'], color='#8f9c9
plt.scatter(release['release_year'], added['index'], color="#564d4d", s=75, label='Release Year', zorde
plt.scatter(added['year_added'], added['index'], color="#B20710", s=75, label='Added Year', zorder=3)

plt.legend(ncol = 2, bbox_to_anchor=(1., 1.01), loc = "lower right", frameon=False)
plt.yticks(added['index'], release['Country'])
plt.title("Release Vs Added Year of Netflix Contents", loc='left', fontsize=18)
plt.xlabel('Year')
plt.tight_layout()

plt.show()
```



- As for contents from India, USA and UK, Netflix prefers old classic ones.
- On the other hand, as for contents from Spain and South Korea, Netflix invests more in relatively newly released contents.

In [38]:

```
sns.set_style('whitegrid', {'grid.linestyle': '--'})
p = distinct.groupby(['year_added', 'type'])['content_age'].mean().reset_index()
sns.lineplot(p, x = 'year_added', y = 'content_age', hue = 'type', palette=['#B20710', '#564d4d'], mar
plt.title("Average year gap between Year added - Release year", fontsize = 14, loc = 'left')
plt.suptitle("Average Content Age", fontsize = 18, x = 0.125, y=1, horizontalalignment='left')
plt.ylabel('')
plt.xlabel('')
sns.despine()
```

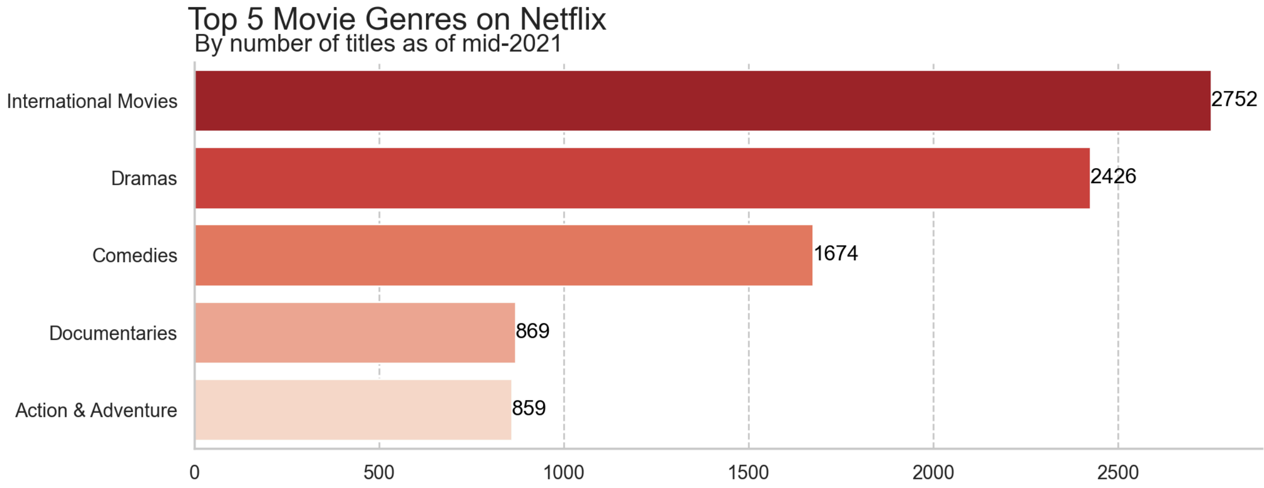


- This line plot shows that the average age of Netflix movies started increasing since 2015, while that of TV shows started decreasing.
- It is interesting that in 2010 Netflix used to invest in old movies released before the 00s.

What are the most common genres on Netflix?

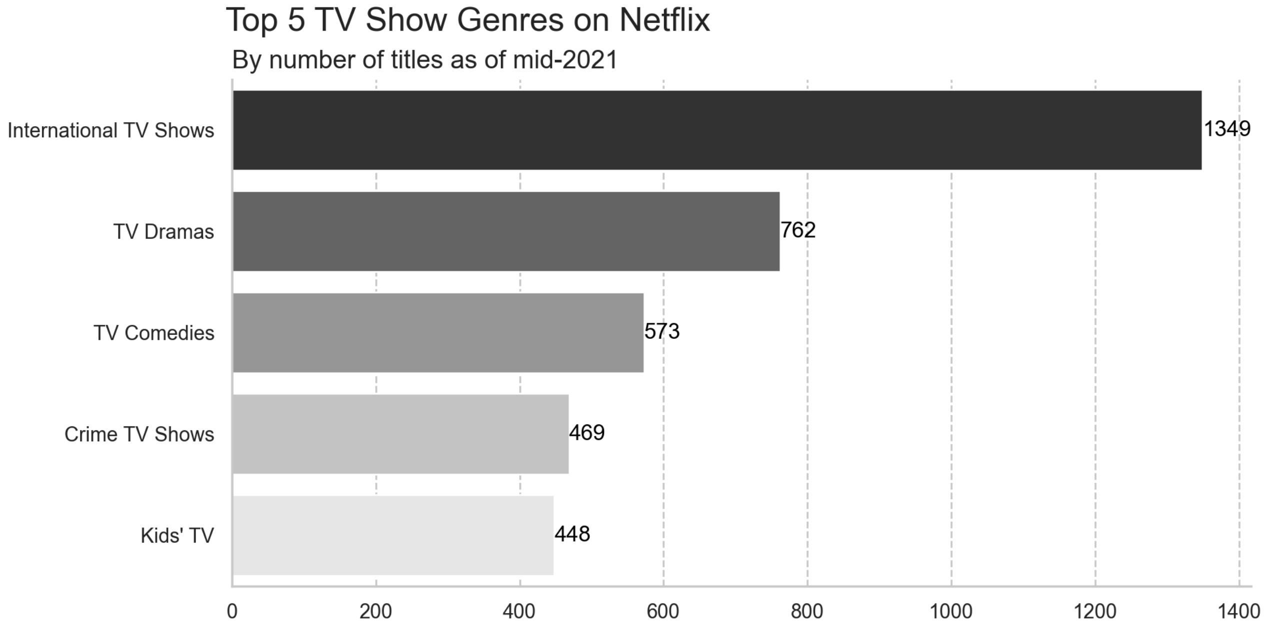
In [39]:

```
# Top 10 genres
genres = netflix_new[netflix_new['type']=='Movie'].groupby(["Genre"])["title"].nunique().reset_index()
genres = genres.sort_values(by="title", ascending=False).reset_index()
genres = genres.drop("index", axis=1)
# Draw a barplot
sns.set_style("whitegrid", {'grid.linestyle': '--'})
ax = sns.barplot(genres.head(5), y="Genre", x="title", palette="Reds_r")
for bars in ax.containers:
    heights = [b.get_width() for b in bars]
    labels = [f'{h:.0f}' if h > 0.001 else '' for h in heights]
    ax.bar_label(bars, labels=labels, label_type='edge', color = 'black')
plt.title("By number of titles as of mid-2021", fontsize = 14, loc='left')
plt.suptitle("Top 5 Movie Genres on Netflix", fontsize = 18, x = 0.12, y=0.99, horizontalalignment='le
plt.ylabel("")
plt.xlabel("")
sns.set(rc = {'figure.figsize':(10, 5)})
sns.despine()
```



In [40]:

```
# Top 10 genres
genres = netflix_new[netflix_new['type']=='TV Show'].groupby(["Genre"])["title"].nunique().reset_index
genres = genres.sort_values(by="title", ascending=False).reset_index()
genres = genres.drop("index", axis=1)
# Draw a barplot
sns.set_style("whitegrid", {'grid.linestyle': '--'})
ax = sns.barplot(genres.head(5), y="Genre", x="title", palette="Greys_r")
for bars in ax.containers:
    heights = [b.get_width() for b in bars]
    labels = [f'{h:.0f}' if h > 0.001 else '' for h in heights]
    ax.bar_label(bars, labels=labels, label_type='edge', color = 'black')
plt.title("By number of titles as of mid-2021", fontsize = 14, loc='left')
plt.suptitle("Top 5 TV Show Genres on Netflix", fontsize = 18, x = 0.12, y=0.99, horizontalalignment='
plt.ylabel("")
plt.xlabel("")
sns.set(rc = {'figure.figsize':(10, 5)})
sns.despine()
```

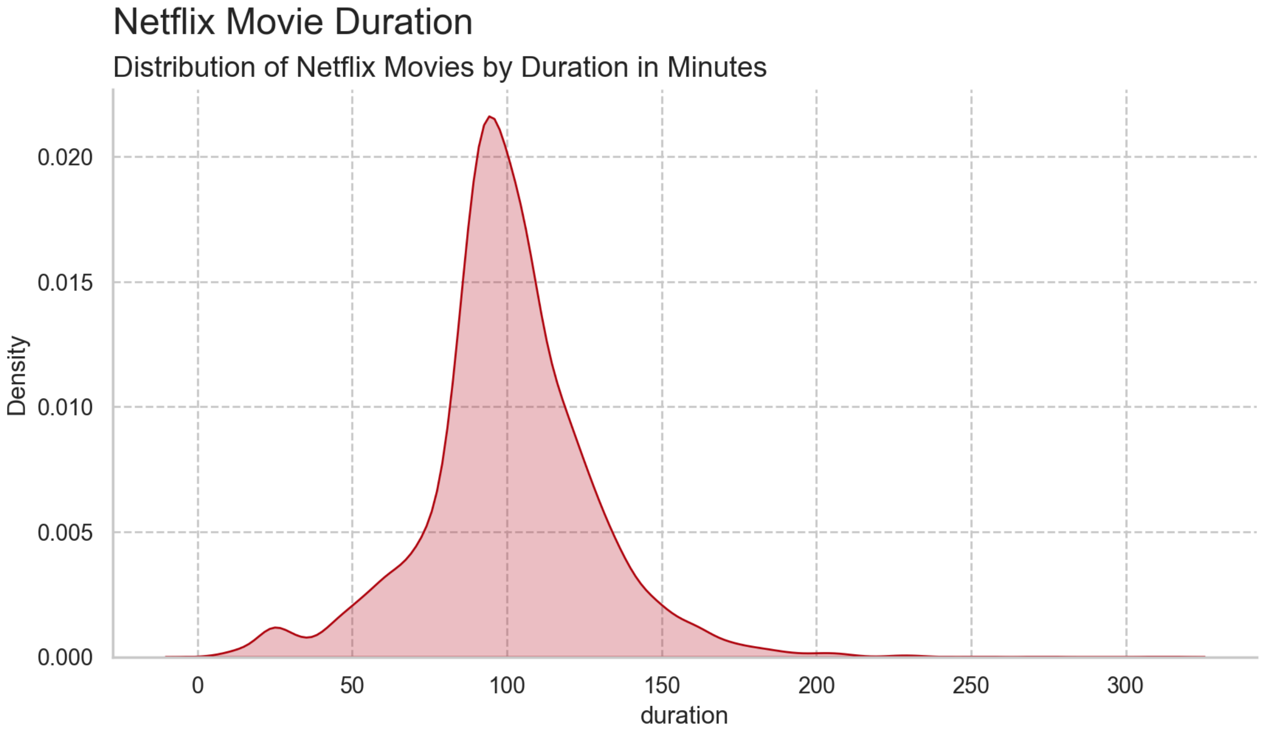


- It's interesting that the top 3 most common genres on Netflix are the same for Movie and TV Show - International, Dramas and Comedies.

Movie and TV Show duration

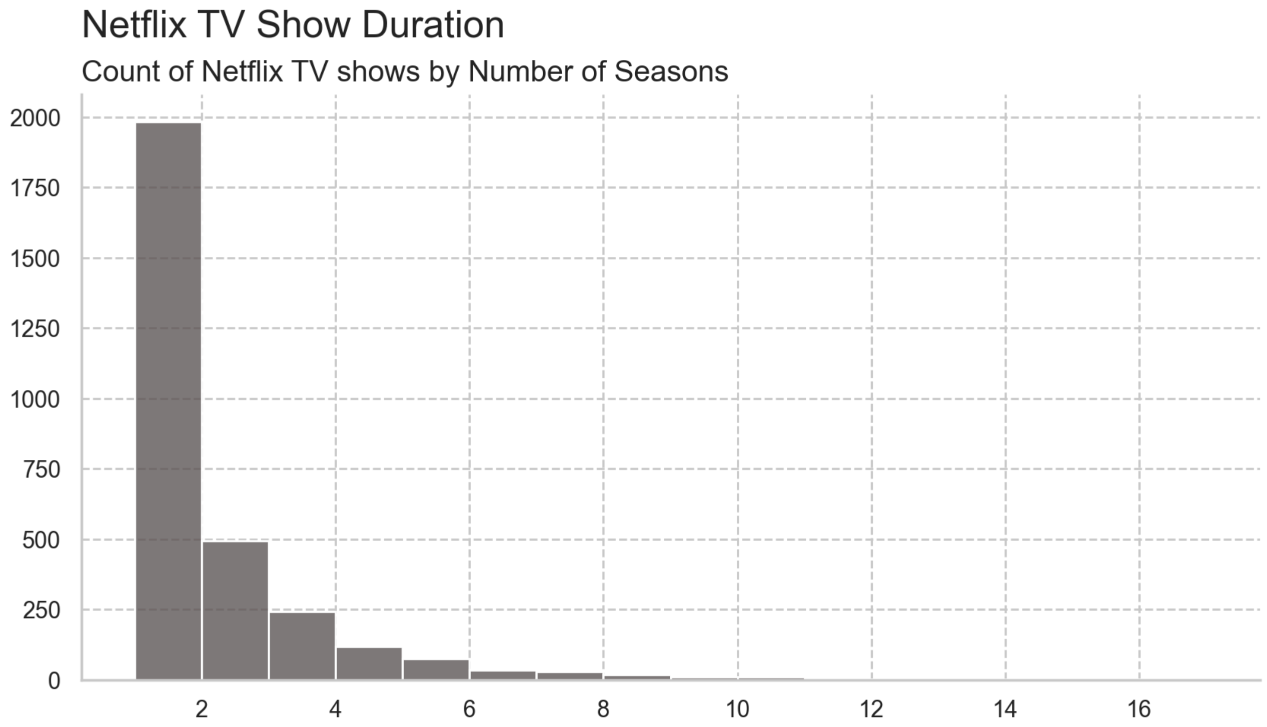
In [41]:

```
movie = distinct[distinct['type']=='Movie']
sns.set_style('whitegrid', {'grid.linestyle':'--'})
plt.ylabel('')
plt.xlabel('')
sns.kdeplot(movie['duration'], color = '#B20710', shade = True)
plt.title("Distribution of Netflix Movies by Duration in Minutes", loc='left', fontsize = 14)
plt.suptitle("Netflix Movie Duration", fontsize = 18, x = 0.125, y=.99, horizontalalignment='left')
sns.despine()
```



In [42]:

```
tv = distinct[distinct['type']=='TV Show']
sns.set_style('whitegrid', {'grid.linestyle':'--'})
sns.histplot(tv['duration'], color = '#564d4d', bins=16)
plt.ylabel('')
plt.xlabel('')
plt.title("Count of Netflix TV shows by Number of Seasons", loc='left', fontsize = 14)
plt.suptitle("Netflix TV Show Duration", fontsize = 18, x = 0.125, y=.99, horizontalalignment='left')
sns.despine()
```



- Many movies on Netflix have a duration of 75–120 minutes, which is reasonable given that a significant portion of the audience may find it challenging to watch a 3-hour movie in one sitting.
- Furthermore, Netflix is increasingly adding more single-season TV shows compared to series with two or more seasons.

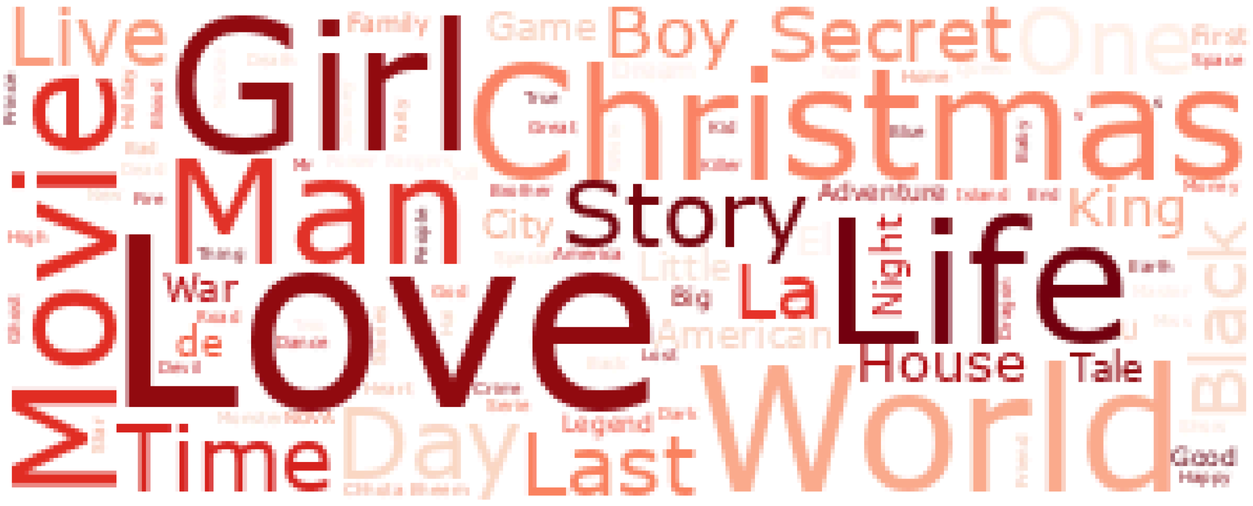
Most frequent word in Netflix titles and content descriptions

In [43]:

```
from wordcloud import WordCloud
from wordcloud import STOPWORDS
def wordcloud(data, width=1200, height=500):
    word_draw = WordCloud(
        font_path=r"C:₩Windows₩Fonts₩Verdana.ttf",
        stopwords=STOPWORDS,
        width=width, height=height,
        background_color="white",
        colormap = "Reds_r",
        random_state=42
    )
    word_draw.generate(data)

    plt.figure(figsize=(16, 8))
    plt.imshow(word_draw)
    plt.axis("off")
    plt.show()
```

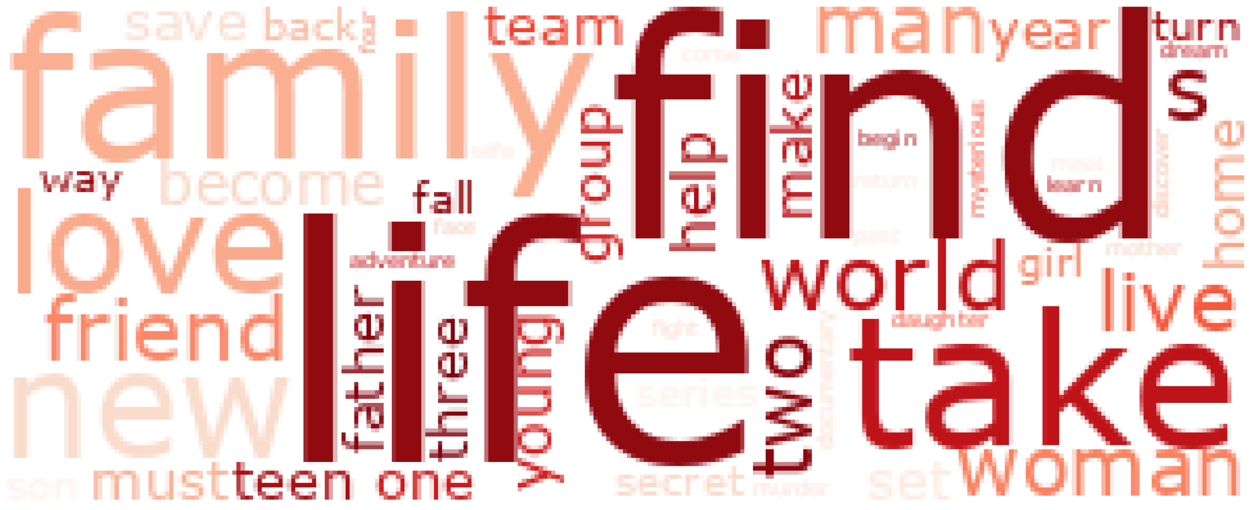
In [44]:

```
# Word cloud for content descriptions
title = netflix["title"].tolist()
title =str(title)
title = re.sub("[₩"₩-₩[₩]₩'₩.₩?]", "", title)
title = title.replace(" ", ",")
title = re.sub(",{2,}", ",", title)
wordcloud(title, width=300, height=120)
```



- This is the word cloud of titles of Netflix content. It is interesting to note that many films and series share the same keywords in their titles.
- "Love", "Life" and "Girl" are the most frequent words appearing on Netflix titles. "Story", "World", "Man", "Black" and "Time" also appear more frequently than other words in titles.

In [46]:
```
# Word cloud for content descriptions
description = netflix["description"].tolist()
description =str(description)
description = re.sub("[₩"₩-₩[₩]₩'₩.₩?]", "", description)
description = description.replace(" ", ",")
description = re.sub(",{2,}", ",", description)
wordcloud(description, width=300, height=120)
```



Looking at the wordcloud of Netflix content descriptions above, the words "find", "life" and "family" seem to be the most frequent words in the content descriptions. The words "world", "new", "take", "love" and "live" also frequently appear.

5. Key Findings

Here's the key findings from this EDA.

1. Majority of Netflix contents are movie. However, Netflix has increasingly focused on TV shows in recent years.
2. Netflix releases higher number of contents during Holiday seasons — December, January, and July when customers tend to have more time off during those periods of the year.
3. Netflix offers internationally diverse contents, targeting a global audience in more than 120 countries. The three largest content producing countries on Netflix are the U.S., India and the U.K.
4. As for contents from the U.S. and India, movies are far more popular than TV shows, while as for those from East Asian countries – South Korea and Japan, TV shows are more popular.
5. Netflix's largest target audience group is young adults. While contents from the U.S. and European countries focus more on adult viewers, those from the India target more on teens.
6. Netflix tends to promote various movies from old classic ones to newly released ones, while as for TV shows, focus more on the contents made in recent years.
7. The most popular genres on Netflix are International, Dramas and Comedies.
8. Netflix do not prefer contents with too long duration. Netflix is focusing on movies with a duration of 75–120 minutes and single-season TV shows.
9. Many films and series share the same keywords in their titles and content descriptions. "Love", "Life" and "Girl" are the most frequent words appearing on Netflix titles. “Find", “Life" and “Family" are the most frequent words appearing on content descriptions

6. Recommendations

In response to the business task and problem, the following are the concluding recommendations for the hypothetical streaming company 'Youflix' to incorporate into their content library and marketing strategy:

1. To cater to evolving viewer preferences, Youflix should maintain a balanced mix of movies and TV shows, with a slight emphasis on TV shows. The content library should offer diverse international content, focusing on movies from the U.S., India, and the U.K., and TV shows from South Korea and Japan.
2. Major content launches should be strategically planned during holiday seasons, specifically in December, January, and July.
3. Additionally, prioritizing popular genres such as International, Dramas, and Comedies will help attract a broad audience.
4. For audience targeting, Youflix should create or acquire more teen-focused content to appeal to the younger demographic in India.
5. The company should focus on movies with a duration of 75–120 minutes and single-season TV shows to cater to audiences with shorter attention spans.
6. Targeting young adults with contemporary dramas, comedies, and relatable themes is essential, along with developing targeted campaigns using social media platforms popular with this demographic.

7. Conclusions

Netflix has made streaming available to a wide audience through investments in global content. Netflix is becoming more and more popular among various age groups and social classes, and it is important to acknowledge the impact of this streaming platform on the global entertainment market and the economy in general.

By following data-driven recommendations, 'Youflix' can create a compelling content library and marketing strategy that resonates with a wide audience and provide its customers with the best content based on the customers' preferences.