In [1]:
```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
%matplotlib inline
```

In [2]:
```
netflix_df = pd.read_csv("netflix_titles.csv")
```

In [3]:
```
netflix_df['date_added'] = pd.to_datetime(netflix_df['date_added'])
netflix_df['year_added'] = netflix_df['date_added'].dt.year
netflix_df['month_added'] = netflix_df['date_added'].dt.month
```

In [4]:
```
netflix_df
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	dur
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	9
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	Se
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 S
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	1 S
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	Se
...	...	...	...	...	...	...	...	...	...	

	show_id	type	title	director	cast	country	date_added	release_year	rating	dur
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	2019-11-20	2007	R	15
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	2019-07-01	2018	TV-Y7	Se
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	2019-11-01	2009	R	8
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	2020-01-11	2006	PG	8
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	2019-03-02	2015	TV-14	11

8807 rows × 14 columns

In [ ]:

Data Information

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

Out[5]:
```
(8807, 14)
```

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

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

In [7]:

```
netflix_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 14 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   datetime64[ns]
 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        
 12  year_added    8797 non-null   float64       
 13  month_added   8797 non-null   float64       
dtypes: datetime64[ns](1), float64(2), int64(1), object(10)
memory usage: 963.4+ KB
```

In [8]:
```
netflix_df.isnull().sum()
```

Out[8]:
```
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
year_added        10
month_added       10
dtype: int64
```

Data Describe

In [9]:
```
netflix_df.describe()
```

Out[9]:

	release_year	year_added	month_added
count	8807.000000	8797.000000	8797.000000
mean	2014.180198	2018.871888	6.654996
std	8.819312	1.574243	3.436554
min	1925.000000	2008.000000	1.000000
25%	2013.000000	2018.000000	4.000000

	release_year	year_added	month_added
50%	2017.000000	2019.000000	7.000000
75%	2019.000000	2020.000000	10.000000
max	2021.000000	2021.000000	12.000000

In [10]:
```
from pandas_profiling import ProfileReport
```

In [11]:
```
netflix_profile = ProfileReport(netflix_df)
```

Data cleaning

Deleting director and cast columns because its has lot missing values

In [12]:
```
netflix_df.drop(['director', 'cast'], axis=1, inplace=True)
```

In [13]:
```
netflix_df.columns
```

Out[13]:
```
Index(['show_id', 'type', 'title', 'country', 'date_added', 'release_year',
       'rating', 'duration', 'listed_in', 'description', 'year_added',
       'month_added'],
      dtype='object')
```

In [14]:
```
#netflix_df.isnull().sum()
```

In [15]:
```
# date_df = netflix_df.dropna(subset=['date_added'])
#country_df = netflix_df.dropna(subset=['country'])
#rating_df = netflix_df.dropna(subset=['rating'])
#duration = netflix_df.dropna(subset =['duration'])
```

Missing values of country, rating, duration has replaced by the mode value.

In [16]:
```
netflix_df['country'].fillna(netflix_df['country'].mode()[0], inplace=True)
netflix_df['rating'].fillna(netflix_df['rating'].mode()[0], inplace=True)
netflix_df['duration'].fillna(netflix_df['duration'].mode()[0], inplace=True)
```

In [17]:
```
netflix_df.isnull().sum()
```

Out[17]:
```
show_id          0
type             0
title            0
country          0
date_added      10
release_year     0
rating           0
duration         0
listed_in        0
```

```
description      0
year_added      10
month_added     10
dtype: int64
```

In [18]:

```
temp = netflix_df[netflix_df['date_added'].isnull()].index
netflix_df.loc[temp]
```

Out[18]:

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in	
6066	s6067	TV Show	A Young Doctor's Notebook and Other Stories	United Kingdom	NaT	2013	TV-MA	2 Seasons	British TV Shows, TV Comedies, TV Dramas	S
6174	s6175	TV Show	Anthony Bourdain: Parts Unknown	United States	NaT	2018	TV-PG	5 Seasons	Docuseries	
6795	s6796	TV Show	Frasier	United States	NaT	2003	TV-PG	11 Seasons	Classic & Cult TV, TV Comedies	Fr lc
6806	s6807	TV Show	Friends	United States	NaT	2003	TV-14	10 Seasons	Classic & Cult TV, TV Comedies	T m
6901	s6902	TV Show	Gunslinger Girl	Japan	NaT	2008	TV-14	2 Seasons	Anime Series, Crime TV Shows	O A
7196	s7197	TV Show	Kikoriki	United States	NaT	2010	TV-Y	2 Seasons	Kids' TV	A
7254	s7255	TV Show	La Familia P. Luche	United States	NaT	2012	TV-14	3 Seasons	International TV Shows, Spanish-Language TV Sh...	T s
7406	s7407	TV Show	Maron	United States	NaT	2016	TV-MA	4 Seasons	TV Comedies	
7847	s7848	TV Show	Red vs. Blue	United States	NaT	2015	NR	13 Seasons	TV Action & Adventure, TV Comedies, TV Sci-Fi ...	T

	show_id	type	title	country	date_added	release_year	rating	duration	listed_in
8182	s8183	TV Show	The Adventures of Figaro Pho	Australia	NaT	2015	TV-Y7	2 Seasons	Kids' TV, TV Comedies

In [19]:
```
for row in temp:
    if netflix_df.loc[row, 'release_year'] < netflix_df['year_added'].min():
        netflix_df.loc[row, 'year_added'] = netflix_df['year_added'].min()
        netflix_df.loc[row, 'month_added'] = 12
    else:
        netflix_df.loc[row, 'year_added'] = netflix_df.loc[row, 'release_year']
        netflix_df.loc[row, 'month_added'] = 12
```

In [20]:
```
#netflix_df.drop('date_added', axis=1, inplace=True)

netflix_df['year_added'] = netflix_df['year_added'].astype('int')
netflix_df['month_added'] = netflix_df['month_added'].astype('int')

netflix_df = netflix_df.sort_values(by=['year_added', 'month_added'])
```

post profiing

In [21]:
```
netflix_profile = ProfileReport(netflix_df)
```

In [22]:
```
#netflix_profile
```

Exploratory Data Analysis

In [23]:
```
netflix_df.type.unique()
```

Out[23]:
```
array(['Movie', 'TV Show'], dtype=object)
```

Differentiate between Movies and TV shows

In [24]:
```
tv = netflix_df[netflix_df['type']=='TV Show'].sort_values(by=['year_added', 'month_add
mv = netflix_df[netflix_df['type']=='Movie'].sort_values(by=['year_added', 'month_added

tv['seasons'] = tv['duration'].apply(lambda x: int(x.split()[0]))
mv['minutes'] = mv['duration'].apply(lambda x: int(x.split()[0]))
```

In [25]:
```
movie_added = mv.groupby('year_added').count()['title']
tv_added = tv.groupby('year_added').count()['title']
```

```
yearly_add = pd.concat([movie_added, tv_added], axis=1)
yearly_add.columns = ['Movies', 'TV Shows']
yearly_add.fillna(0, inplace=True)
yearly_add = yearly_add.astype('int')

yearly_add
```

Out[25]:

year_added	Movies	TV Shows
2008	1	4
2009	2	0
2010	1	1
2011	13	0
2012	3	1
2013	6	6
2014	19	5
2015	56	28
2016	253	177
2017	839	349
2018	1237	413
2019	1424	592
2020	1284	595
2021	993	505

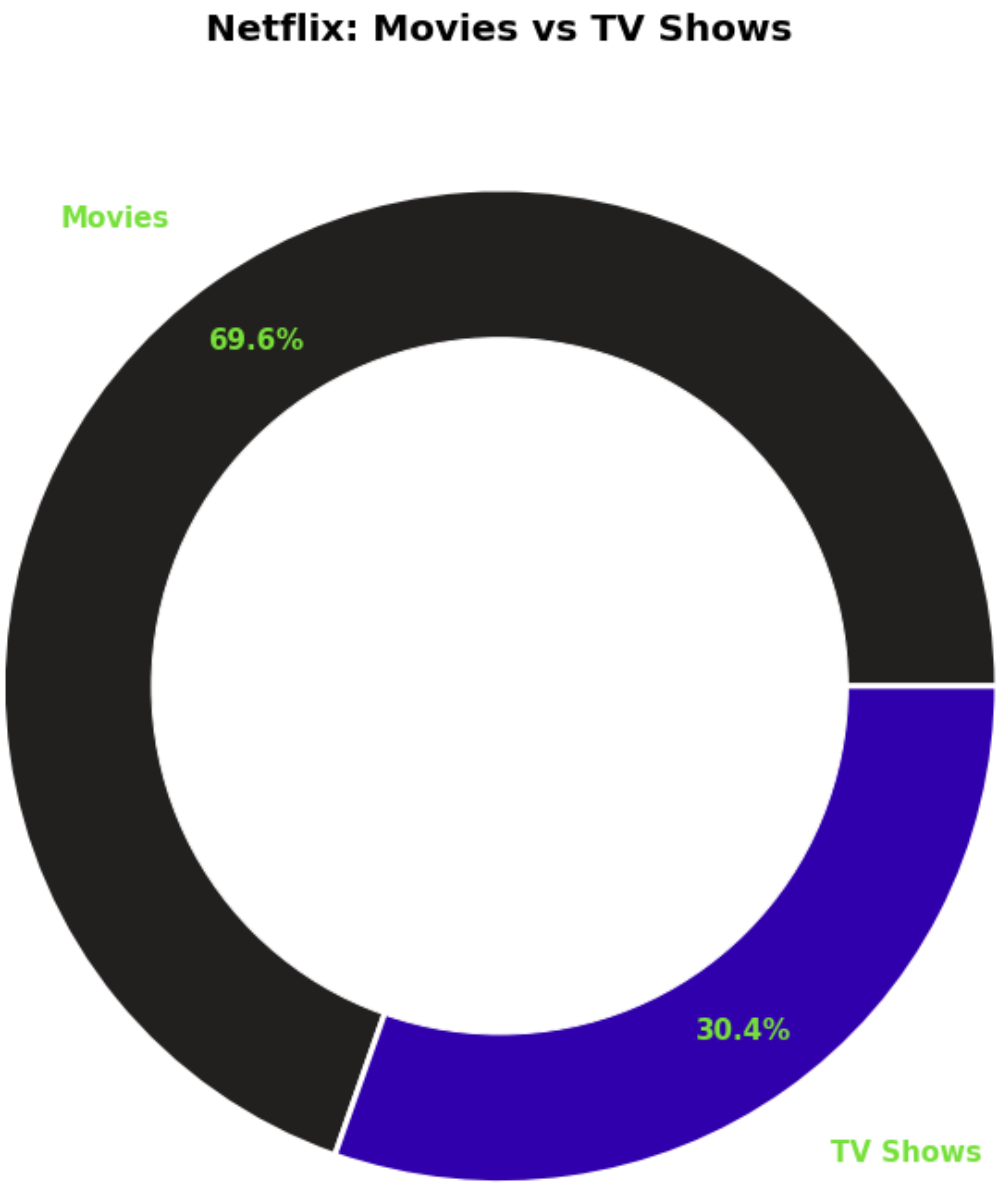
In [26]:

```
%matplotlib inline
```

In [27]:

```
total = yearly_add.sum()
names = list(total.index)
size_of_groups = list(total.values)

plt.figure(figsize=(10,10))
plt.pie(size_of_groups,
        labels=names,
        labeldistance=1.15,
        wedgeprops={'linewidth':3, 'edgecolor':'white'},
        autopct='%1.1f%%',
        pctdistance=0.85,
        textprops={'fontsize': 15, 'color':'#77e03a', 'weight':'bold'},
        colors=['#221f1f', '#3000ad']);

my_circle = plt.Circle((0,0), 0.7, color='white')
p = plt.gcf()
p.gca().add_artist(my_circle)
```

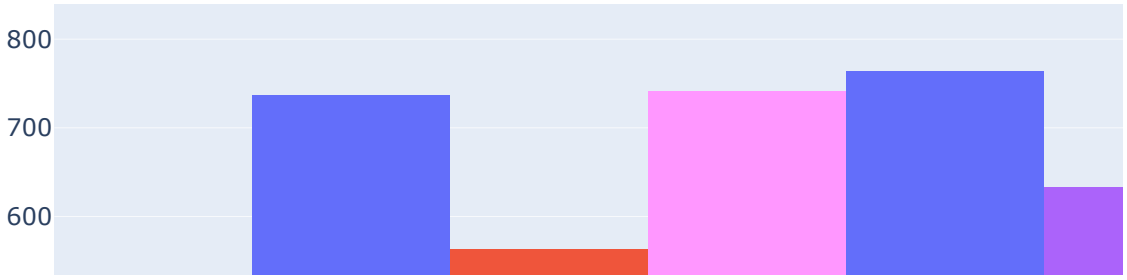
```
plt.title('Netflix: Movies vs TV Shows', fontsize=20, y=1.01, fontweight='bold', color=
plt.tight_layout()
plt.show();
```



which Months has netflix uploads most contents?

In [28]:

```
netflix_df["Month_date_added"] = netflix_df["date_added"].dt.month.fillna(0)
px.histogram(netflix_df, x = "Month_date_added", color = "Month_date_added")
```

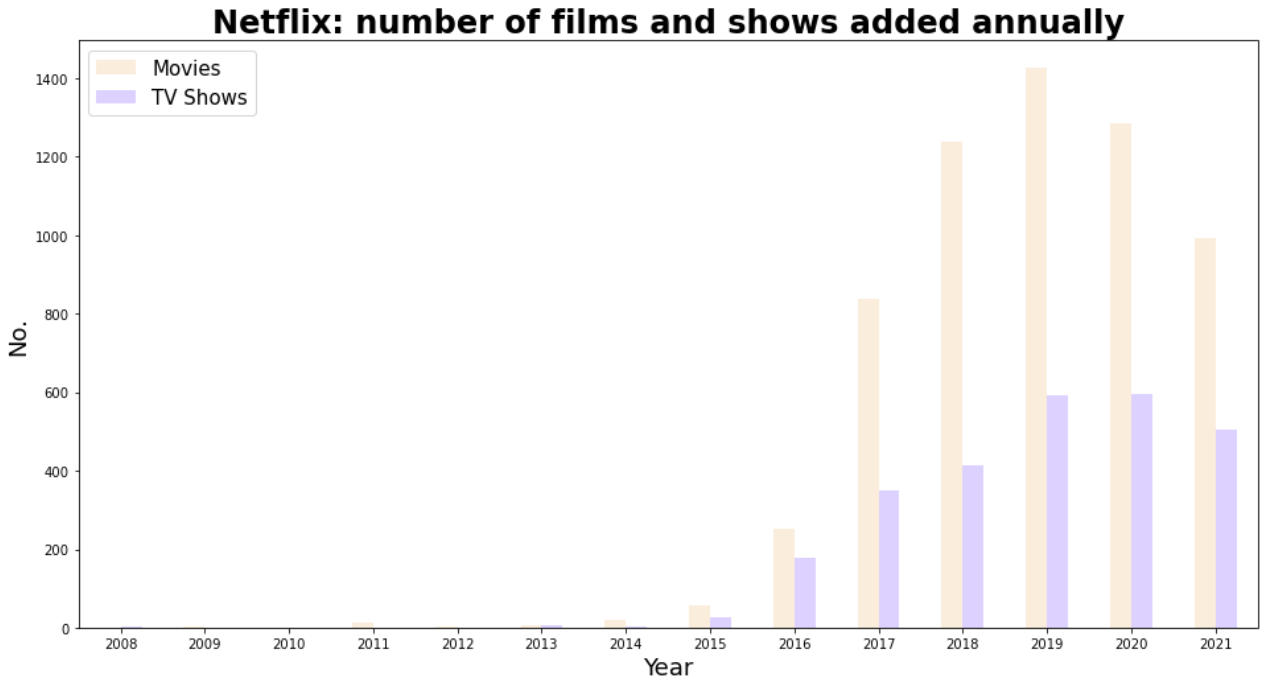


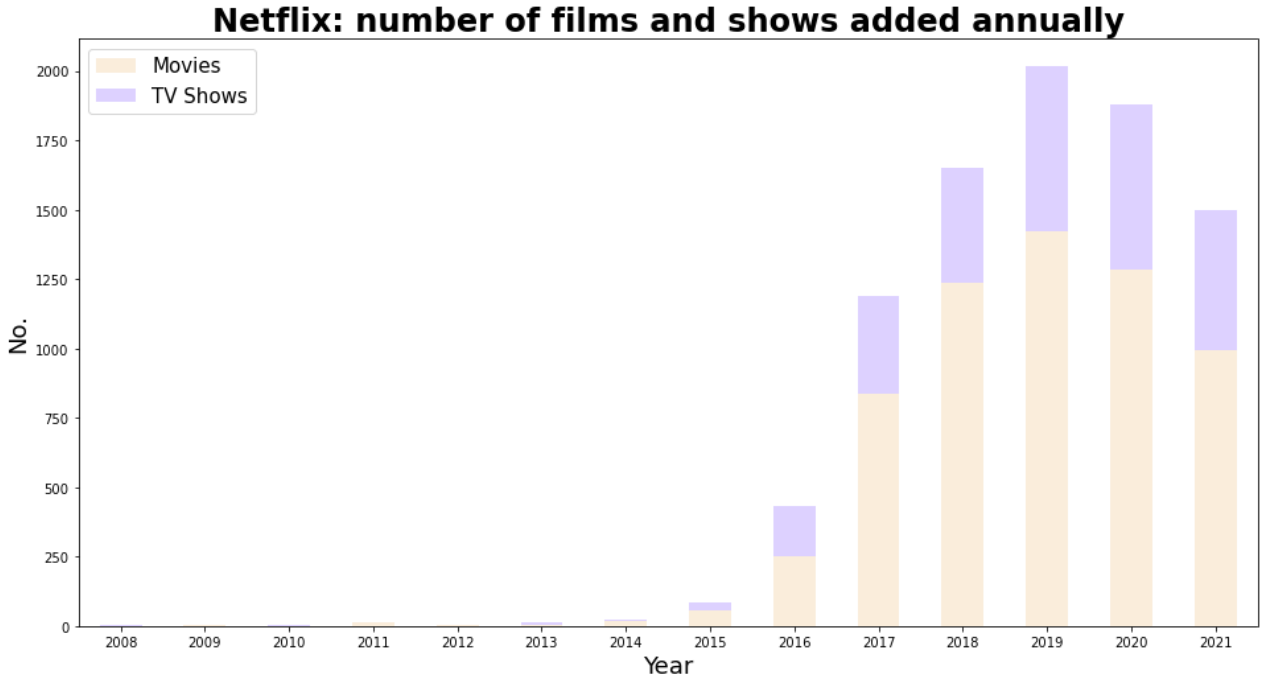
According to this data July has the most content uploads, but its nor enough for me to conclude that Netflix uploads the most content on July every year, it just seems random to me, looking at the years from 2008 to 2020 on the Date_added graph.

Number of films and shows added annually

In [29]:

```
fig, ax = plt.subplots(2, 1, figsize=(16,18))

yearly_add.plot(kind='bar', color=['#FAEBD7', '#daccff'], alpha=0.9, ax=ax[0])
ax[0].set_xlabel('Year', fontsize=18)
ax[0].set_ylabel('No.', fontsize=18)
ax[0].set_title('Netflix: number of films and shows added annually', fontsize=24, fontw
ax[0].tick_params(labelrotation=0)
ax[0].legend(loc=2, fontsize=15)

yearly_add.plot(kind='bar', stacked=True, color=['#FAEBD7', '#daccff'], alpha=0.9, ax=a
ax[1].set_xlabel('Year', fontsize=18)
ax[1].set_ylabel('No.', fontsize=18)
ax[1].set_title('Netflix: number of films and shows added annually', fontsize=24, fontw
ax[1].tick_params(labelrotation=0)
ax[1].legend(loc=2, fontsize=15)

plt.show()
```





Netflix: Movies and TV Shows released from 2008 to 2021

In [30]:
```
yearly_cum = yearly_add.cumsum()
yearly_cum
```

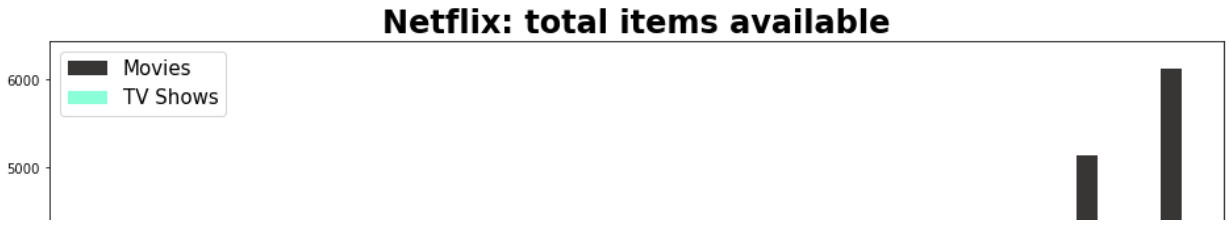
Out[30]:

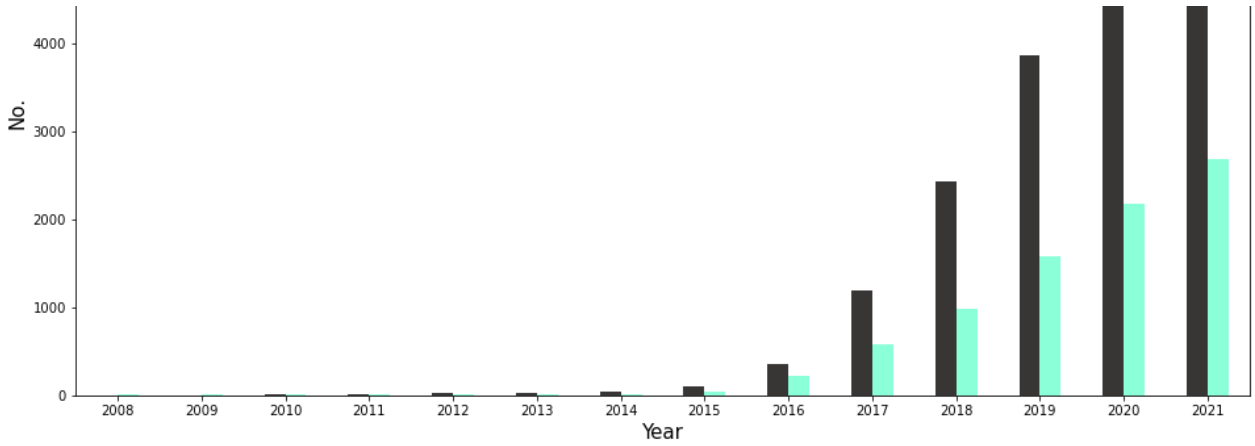
year_added	Movies	TV Shows
2008	1	4
2009	3	4

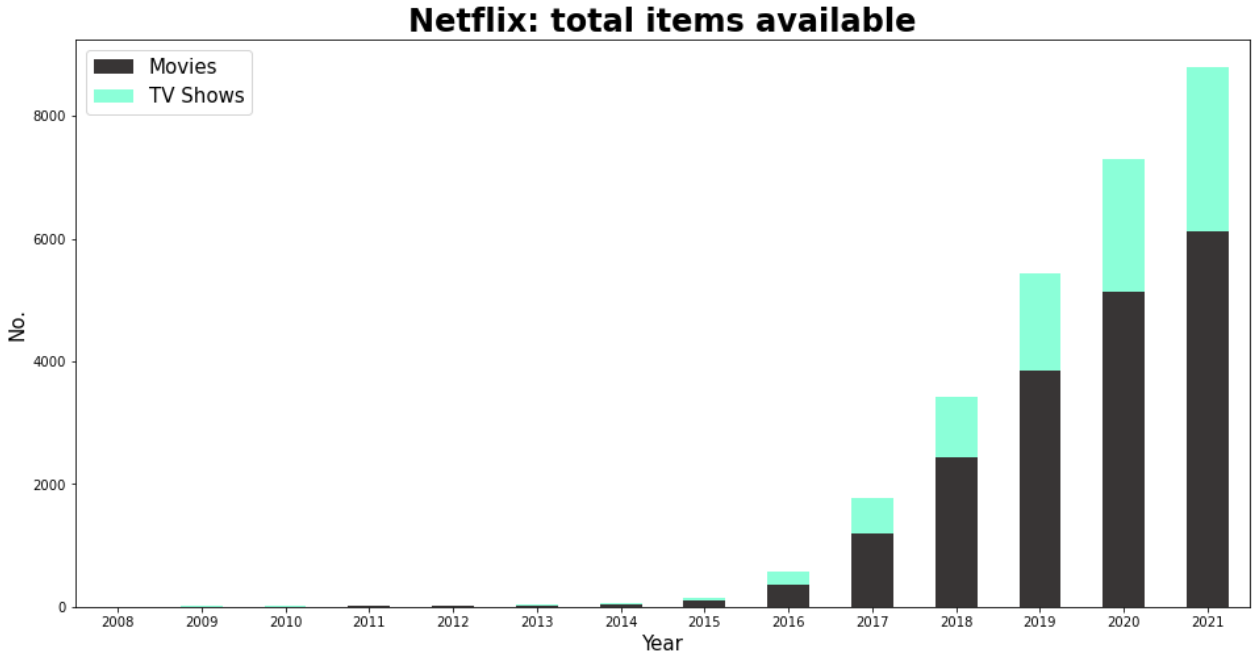
year_added	Movies	TV Shows
2010	4	5
2011	17	5
2012	20	6
2013	26	12
2014	45	17
2015	101	45
2016	354	222
2017	1193	571
2018	2430	984
2019	3854	1576
2020	5138	2171
2021	6131	2676

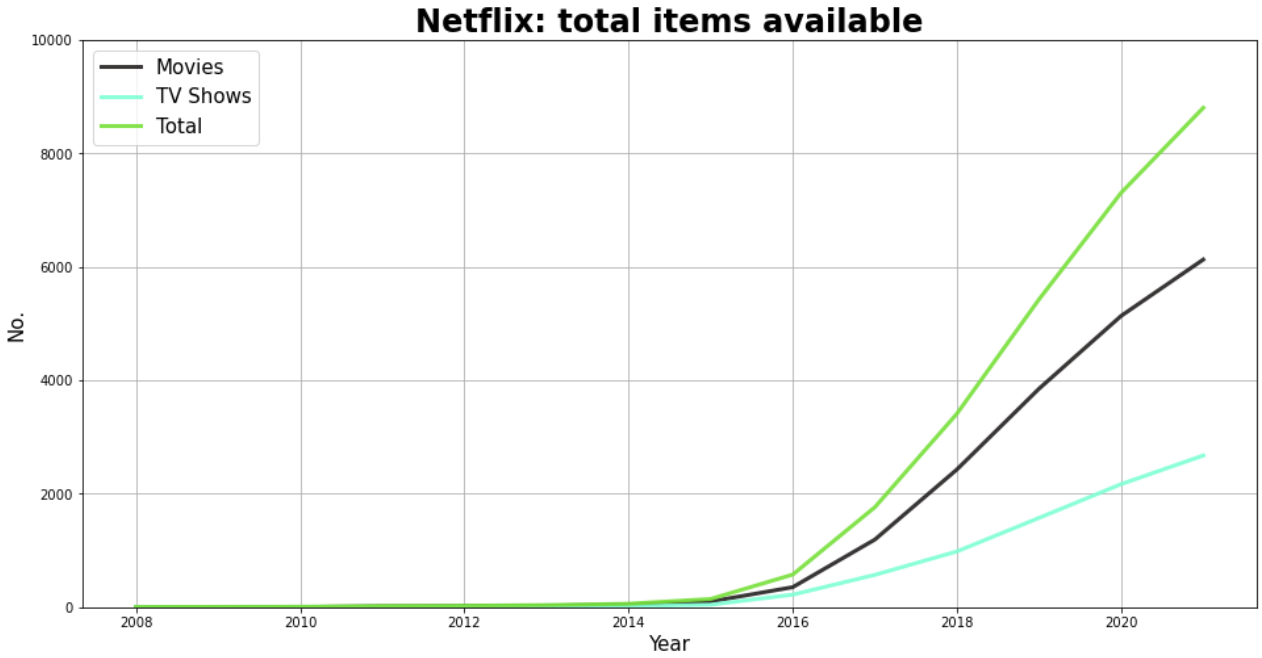
In [31]:

```
fig, ax = plt.subplots(3, 1, figsize=(16,27))

yearly_cum.plot(kind='bar', color=['#221f1f', '#7FFFD4'], alpha=0.9, ax=ax[0])
ax[0].set_xlabel('Year', fontsize=15)
ax[0].set_ylabel('No.', fontsize=15)
ax[0].set_title('Netflix: total items available', fontsize=24, fontweight='bold')
ax[0].tick_params(labelrotation=0)
ax[0].legend(loc=2, fontsize=15)

yearly_cum.plot(kind='bar', stacked=True, color=['#221f1f', '#7FFFD4'], alpha=0.9, ax=a
ax[1].set_xlabel('Year', fontsize=15)
ax[1].set_ylabel('No.', fontsize=15)
ax[1].set_title('Netflix: total items available', fontsize=24, fontweight='bold')
ax[1].tick_params(labelrotation=0)
ax[1].legend(loc=2, fontsize=15)

yearly_cum.plot(color=['#221f1f', '#7FFFD4'], alpha=0.9, linewidth=3, ax=ax[2])
ax[2].plot(yearly_cum.sum(axis=1), color='#77e03a', alpha=0.9, linewidth=3, label='Tota
ax[2].set_xlabel('Year', fontsize=15)
ax[2].set_ylabel('No.', fontsize=15)
ax[2].set_ylim([0,10000])
ax[2].set_title('Netflix: total items available', fontsize=24, fontweight='bold')
ax[2].legend(loc=2, fontsize=15)
ax[2].grid()

plt.show()
```









Which 20 Countries with the most streaming

content?

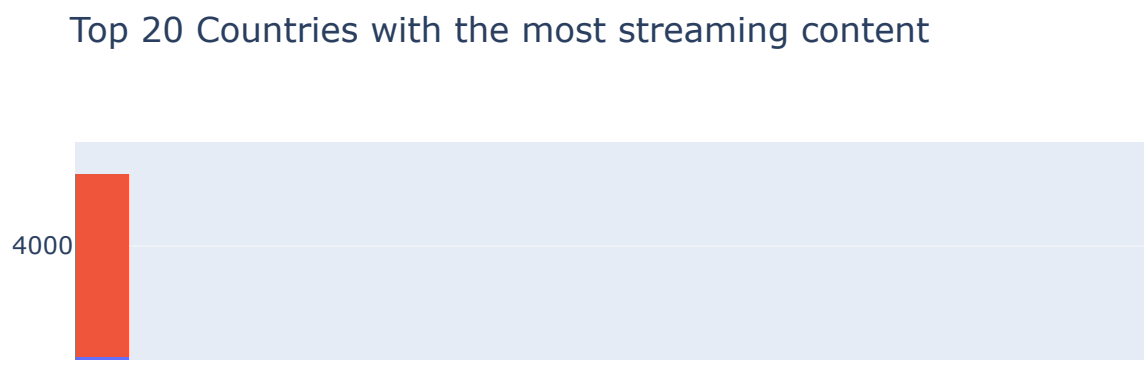
In [32]:
```
count_country = netflix_df.copy()

count_country =  pd.concat([count_country, netflix_df["country"].str.split(",", expand=
count_country = count_country.melt(id_vars=["type", "title"], value_vars=range(12), val
count_country = count_country[count_country["Country"].notna()]
count_country["Country"] = count_country["Country"].str.strip()
count_country
```

Out[32]:

	type	title	variable	Country
0	Movie	To and From New York	0	United States
1	TV Show	Dinner for Five	0	United States
2	TV Show	Frasier	0	United States
3	TV Show	Friends	0	United States
4	TV Show	Gunslinger Girl	0	Japan
...	...	...	...	...
71725	Movie	Barbecue	8	South Africa
79465	Movie	The Look of Silence	9	Netherlands
80532	Movie	Barbecue	9	Sweden
89339	Movie	Barbecue	10	United States
98146	Movie	Barbecue	11	Uruguay

10850 rows × 4 columns

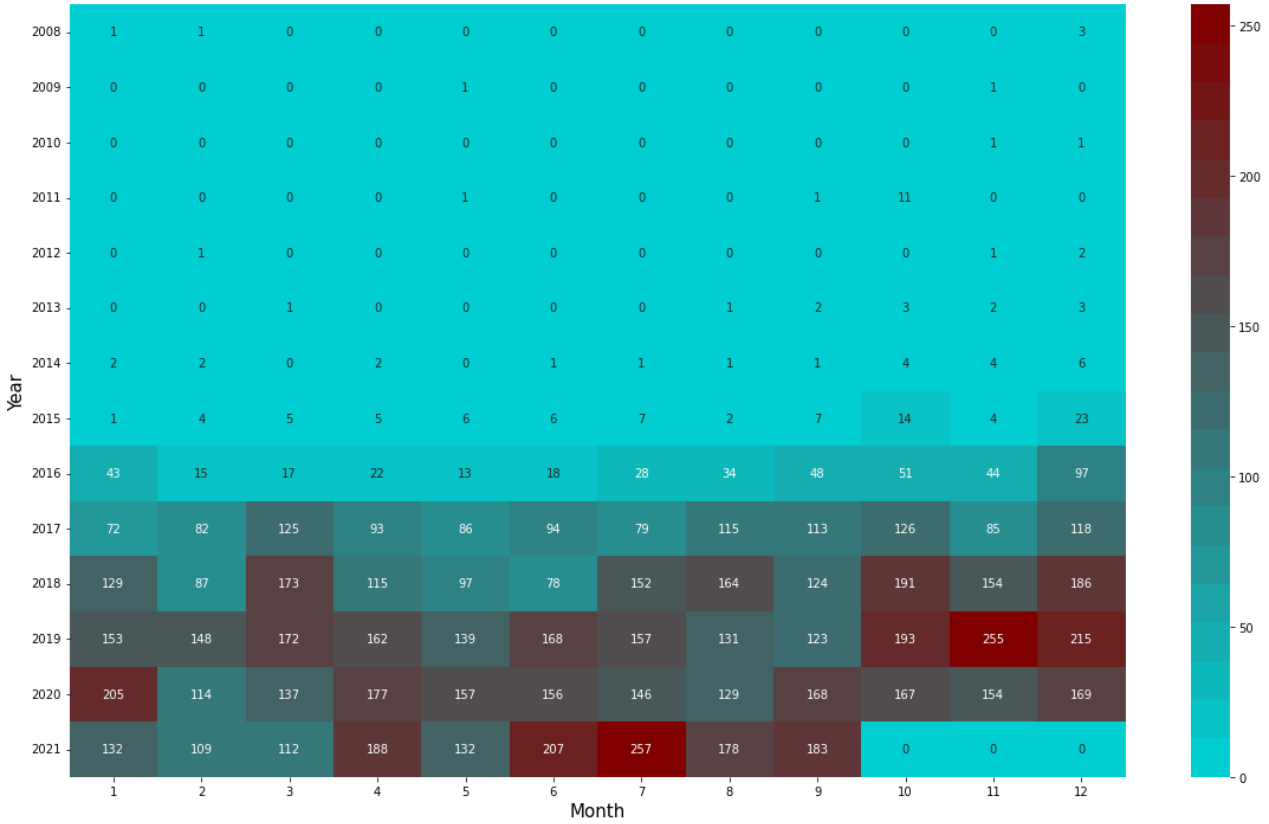
In [33]:
```
px.histogram(count_country, x = "Country", color = "type",
title="Top 20 Countries with the most streaming content").update_xaxes(categoryorder =
```



Netflix: Movies and TV Shows released monthly from 2008 to 2021

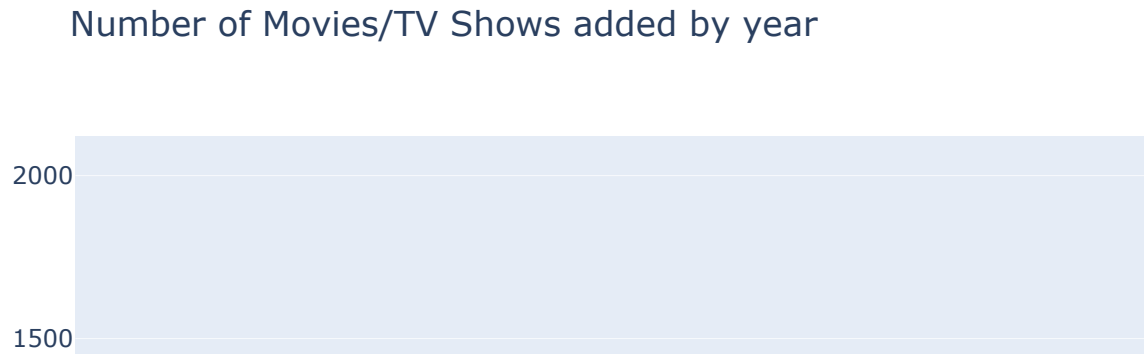
In [34]:
```
cross = pd.crosstab(netflix_df.year_added, netflix_df.month_added)
```

In [35]:
```
from matplotlib.colors import LinearSegmentedColormap
colors = ['#00CED1', '#800000']
cm = LinearSegmentedColormap.from_list("Custom", colors, N=20)

plt.figure(figsize=(20,12))
sns.heatmap(cross, cmap=cm, annot=True, fmt='g')

plt.xlabel('Month', fontsize=15)
plt.ylabel('Year', fontsize=15)
plt.yticks(rotation=0)
plt.title('', fontsize=24, fontweight='bold', y=1.01)
plt.show()
```



Number of Movies/TV Shows added by year

In [36]:
```
px.histogram(netflix_df, x= netflix_df['date_added'].dt.year, color= netflix_df['type']
             title="Number of Movies/TV Shows added by year", labels=dict(x="Year", col
```



Netflix: Average minutes of Movies / Average seasons of TV Shows

In [37]:

```
avg_movie_duration = mv.groupby('release_year').mean()['minutes']
avg_tv_seasons = tv.groupby('release_year').mean()['seasons']

plt.figure(figsize=(16,9))

ax1 = sns.lineplot(x=avg_movie_duration.index, y=avg_movie_duration.values, color='#800
ax1.legend(loc=1, fontsize=15)

ax2 = ax1.twinx()
ax2 = sns.lineplot(x=avg_tv_seasons.index, y=avg_tv_seasons.values, color='#000000', al
ax2.legend(loc=4, fontsize=15)

ax1.set_xlabel('Release Year', fontsize=15)
ax1.set_ylabel('Average mimutes', fontsize=15)
ax2.set_ylabel('Average seasons', fontsize=15)

ax2.spines['right'].set_color('#e50914')
ax2.yaxis.label.set_color('#e50914')
ax2.tick_params(axis='y', colors='#e50914')

plt.title('Netflix: Average minutes of Movies / Average seasons of TV Shows', fontsize=
plt.show()
```