Netflix Binge

By: Mary Colasanto, Terril Vallikalam, Bao Huynh, and Chadwick Nguyen

Project Outline

Analyze a 2021 dataset of Netflix movies and tv shows

General Analysis

1. Distribution of Movies and TV Shows Over Time
   a. The amount of movies vs. tv shows on Netflix
   b. Additions of movies vs. tv shows per year
   c. The most popular parental rating that was viewed

2. Country Collaboration
   a. How many countries were involved in making movies?
   b. The increase in genre popularities in 2020
   c. US production vs. the rest of the world

Data Cleaning

```python
# Extract relevant columns
new_netflix_df = netflix_df.loc[:, ['type', 'title', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in']]
new_netflix_df
```

	type	title	country	date_added	release_year	rating	duration	listed_in
0	TV Show	3%	Brazil	August 14, 2020	2020	TV-MA	4 Seasons	International TV Shows, TV Dramas, TV Sci-Fi &...
1	Movie	7:19	Mexico	December 23, 2016	2016	TV-MA	93 min	Dramas, International Movies
2	Movie	23:59	Singapore	December 20, 2018	2011	R	78 min	Horror Movies, International Movies
3	Movie	9	United States	November 16, 2017	2009	PG-13	80 min	Action & Adventure, Independent Movies, Sci-Fi...
4	Movie	21	United States	January 1, 2020	2008	PG-13	123 min	Dramas
...	...	...	...	...	...	...	...	...
7782	Movie	Zozo	Sweden, Czech Republic, United Kingdom, Denmar...	October 19, 2020	2005	TV-MA	99 min	Dramas, International Movies
7783	Movie	Zubaan	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals
7784	Movie	Zulu Man in Japan	NaN	September 25,	2019	TV-	44 min	Documentaries, International

After importing our data, we needed to create a subset of our data

```python
# Initial count in columns
new_netflix_df.count()
```

```
type            7787
title           7787
country         7280
date_added      7777
release_year    7787
rating          7780
duration        7787
listed_in       7787
dtype: int64
```

```python
# Drop columns with NaN
netflix = new_netflix_df.dropna()
```

```python
# Final count in columns
netflix.count()
```

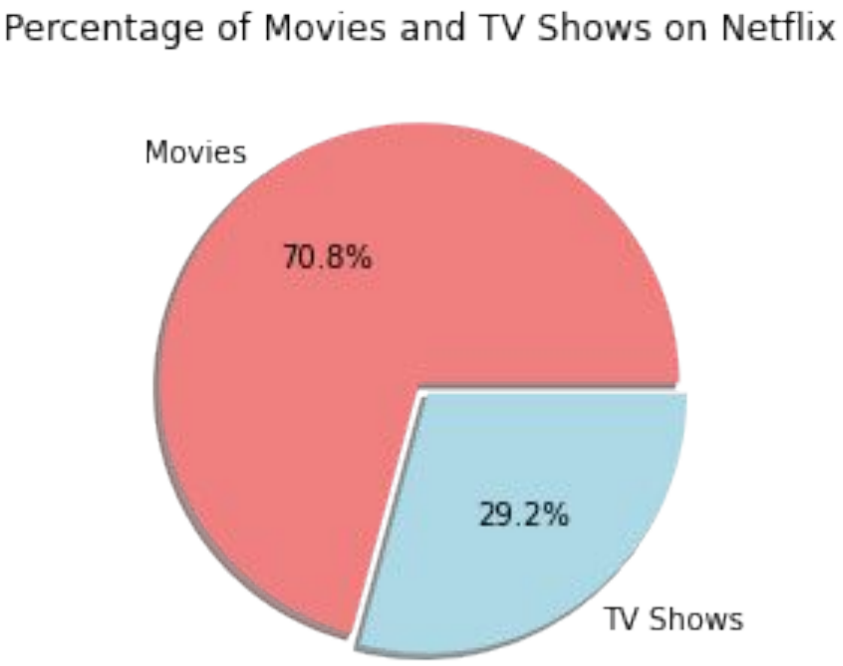
```
type            7265
title           7265
country         7265
date_added      7265
release_year    7265
rating          7265
duration        7265
listed_in       7265
dtype: int64
```

```python
# Rename columns
netflix_renamed = netflix.rename(columns={"type": "Movie/TV Show",
                                          "title":"Title",
                                          "country":"Country of Production",
                                          "date_added":"Date Added to Netflix",
                                          "release_year":"Release Year",
                                          "rating":"Parental Rating",
                                          "duration":"Duration",
                                          "listed_in":"Genre"})
netflix_renamed
```

	Movie/TV Show	Title	Country of Production	Date Added to Netflix	Release Year	Parental Rating	Duration	Genre
0	TV Show	3%	Brazil	August 14, 2020	2020	TV-MA	4 Seasons	International TV Shows, TV Dramas, TV Sci-Fi &...
1	Movie	7:19	Mexico	December 23, 2016	2016	TV-MA	93 min	Dramas, International Movies
2	Movie	23:59	Singapore	December 20, 2018	2011	R	78 min	Horror Movies, International Movies
3	Movie	9	United States	November 16, 2017	2009	PG-13	80 min	Action & Adventure, Independent Movies, Sci-Fi...
4	Movie	21	United States	January 1, 2020	2008	PG-13	123 min	Dramas
...	...	...	...	...	...	...	...	...
7781	Movie	Zoom	United States	January 11, 2020	2006	PG	88 min	Children & Family Movies, Comedies

After extraction, we needed to drop rows with NaN values to ensure we can use the correct data for future analysis

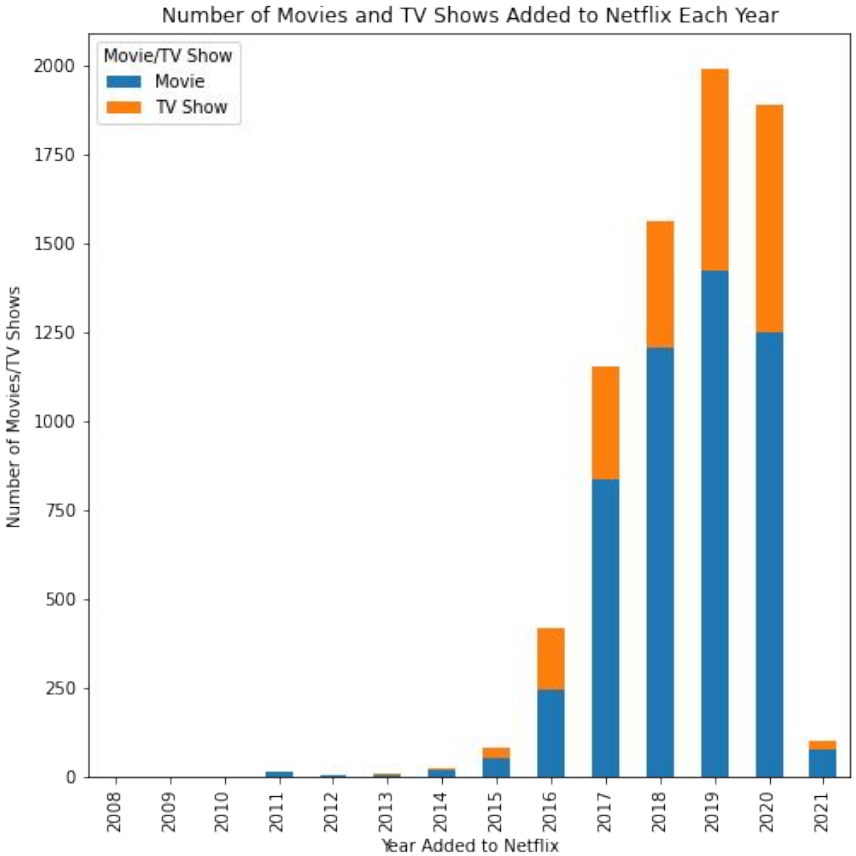
1. Distribution of Movies and TV Shows Over Time



Distribution of Movies vs. TV Shows on Netflix

```python
# Create stacked bar chart
netflix.groupby(["Year Added to Netflix", "Movie/TV Show"]).size().unstack().plot(kind='bar', stacked=True,
                legend="upper left", figsize=(8,8))
plt.title("Number of Movies and TV Shows Added to Netflix Each Year")
plt.ylabel("Number of Movies/TV Shows")
plt.savefig("Images/Number of Movies and TV Shows Added to Netflix Each Year.png")
plt.show()
```

TV Shows and Movies Added Per Year

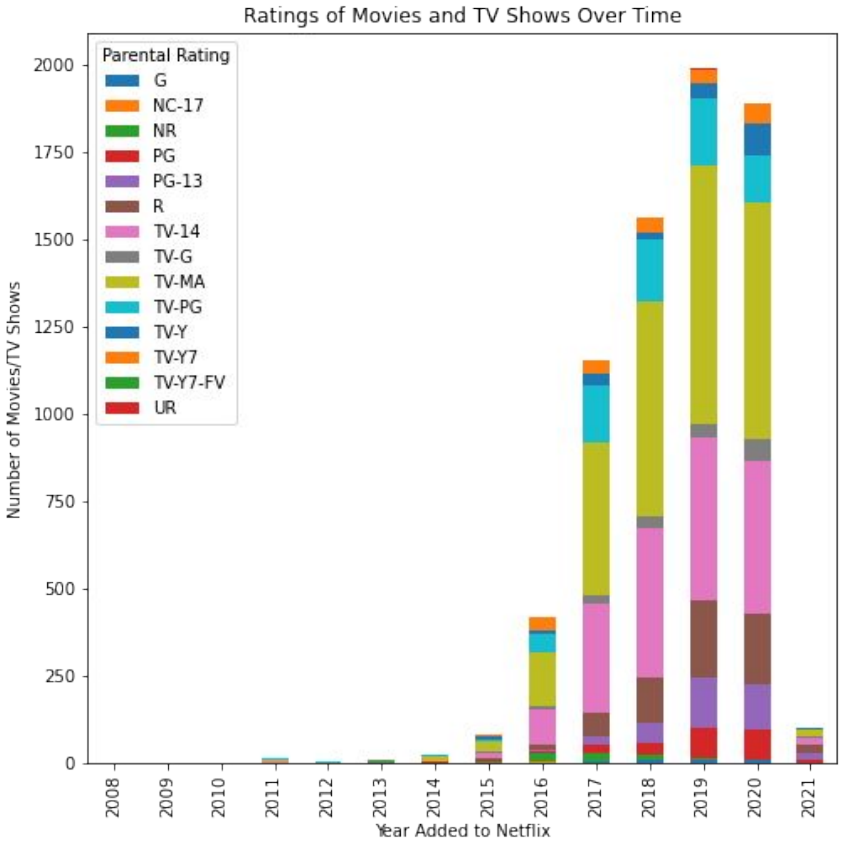


Number of Movies and TV Shows Added to Netflix Each Year

```
netflix.groupby(["Year Added to Netflix", "Parental Rating"]).size().unstack().plot(kind='bar', stacked=True,
                legend="upper left", figsize=(8,8))

plt.title("Ratings of Movies and TV Shows Over Time")
plt.ylabel("Number of Movies/TV Shows")
plt.savefig("Images/Ratings of Movies and TV Shows Over Time.png")
plt.show()
```

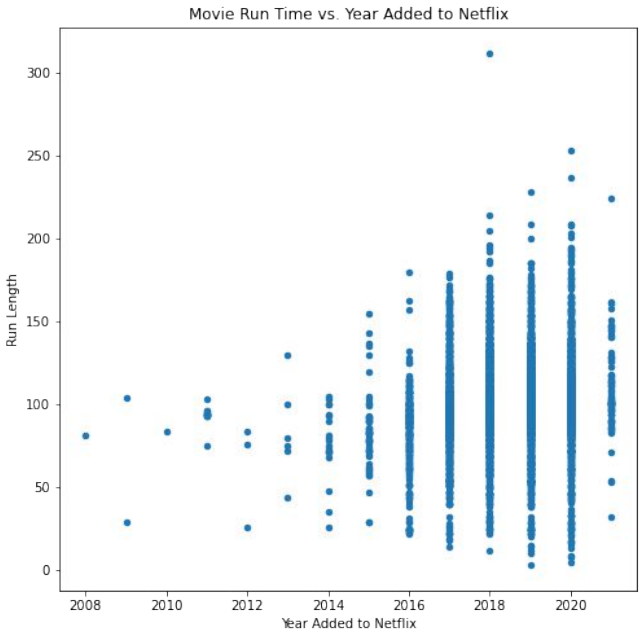
Distribution of Parental Ratings Added to Netflix Per Year

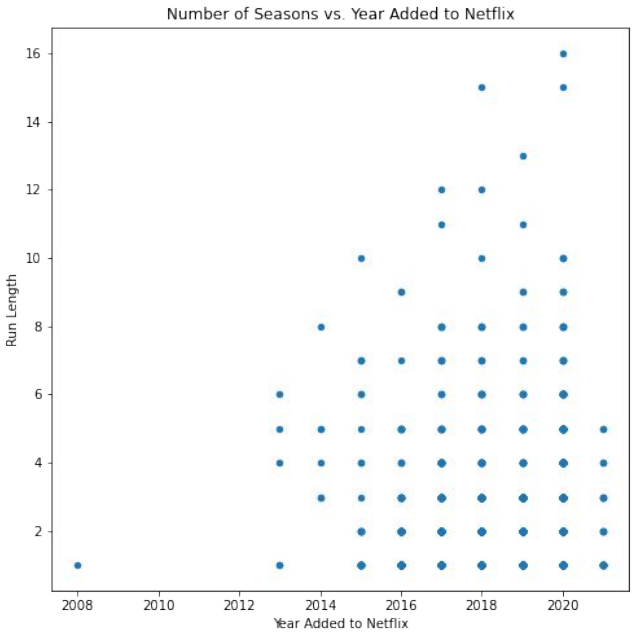


Number of Movies and TV Shows Added to Netflix Each Year

```python
# Create scatterplot
movie_duration = netflix.loc[netflix["Movie/TV Show"] == "Movie"]
movie_duration.plot(kind="scatter", x="Year Added to Netflix", y="Run Length", figsize=(8,8),
                    title="Movie Run Time vs. Year Added to Netflix")
plt.savefig("Images/Movie Run Time vs. Year Added to Netflix.png")
plt.show()
```

Length of Movies/TV Shows vs. Year Added to Netflix

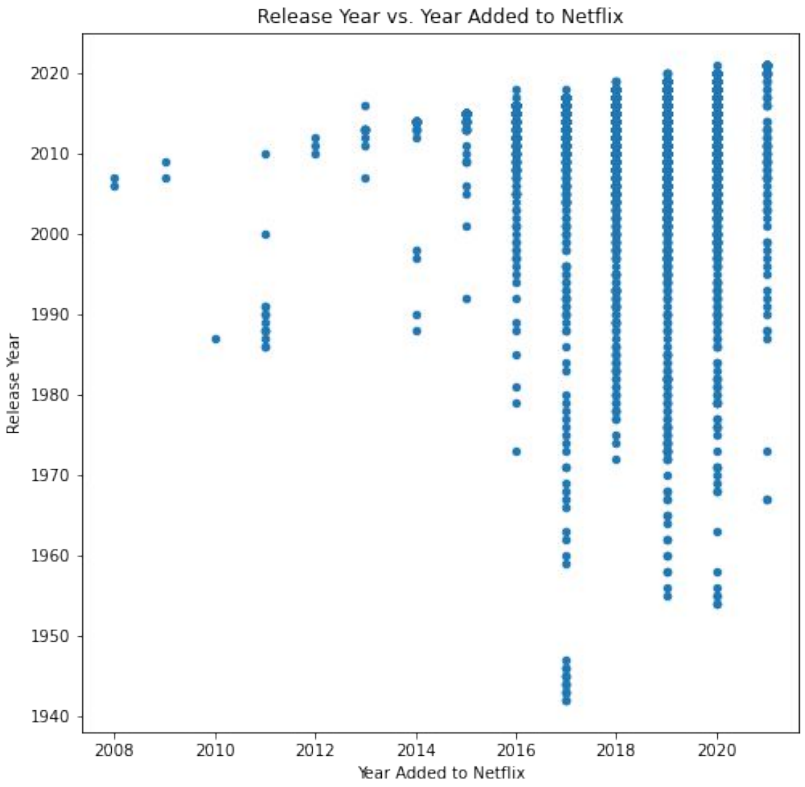




Length of Movies/TV Shows vs. Year Added to Netflix

```python
# Create scatterplot
netflix.plot(kind="scatter", x="Year Added to Netflix", y="Release Year", figsize=(8,8),
                    title="Release Year vs. Year Added to Netflix")
plt.savefig("Images/Release Year vs. Year Added to Netflix.png")
plt.show()
```

Release Year vs. Year Added to Netflix



Release Year vs. Year Added to Netflix

```python
# Determine the difference between "Year Added to Netflix" and "Release Year"
netflix_df["Time_to_Netflix"] = netflix_df["Year Added to Netflix"] - netflix_df["Release Year"]
netflix_df["Time_to_Netflix"].value_counts()
```

```
 0     2590
 1     1388
 2      595
 3      403
 4      310
       ...
-2        1
 58       1
 62       1
 70       1
 63       1
Name: Time_to_Netflix, Length: 73, dtype: int64
```

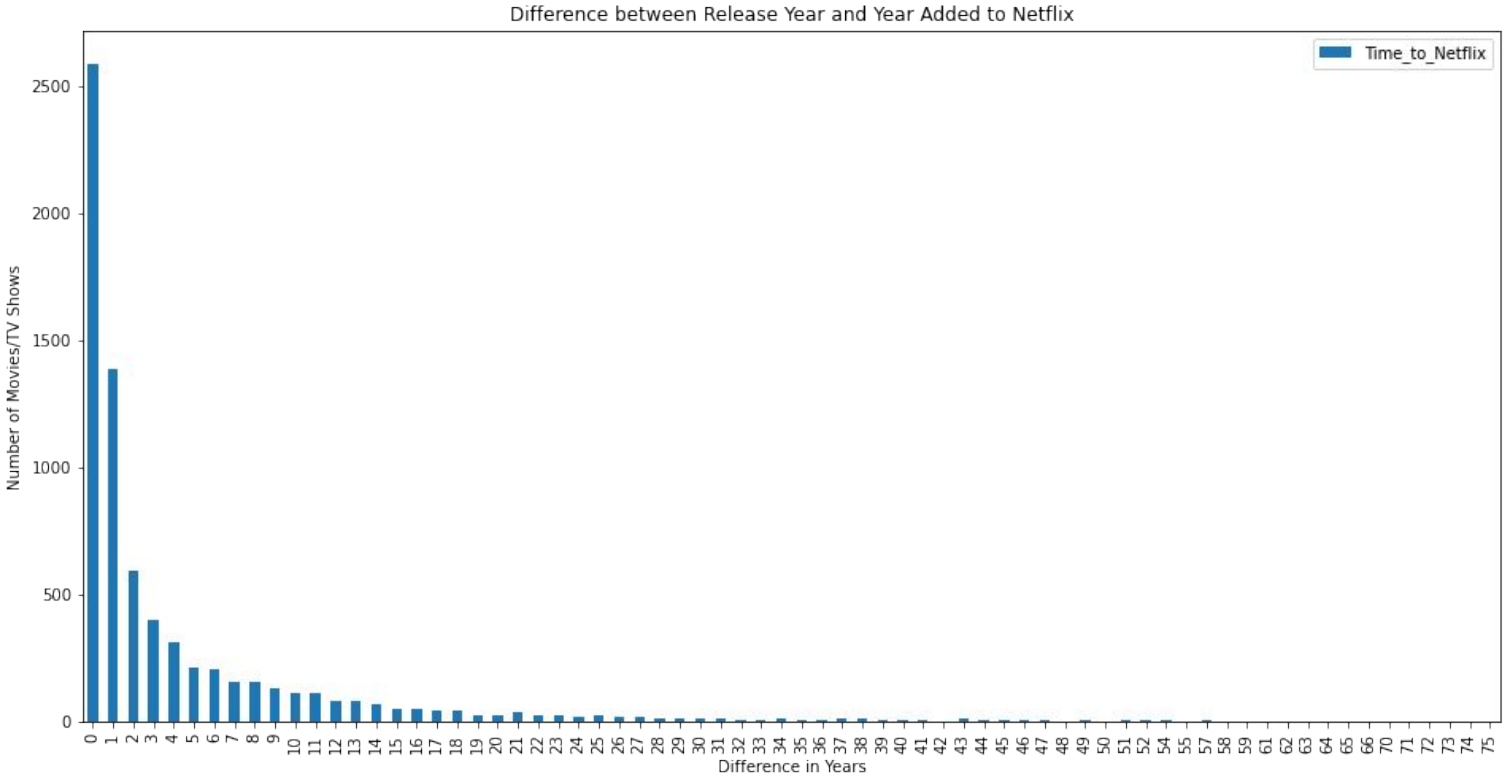
```python
# Remove negative differences and reset the index
top_73 = netflix_df.loc[netflix_df["Time_to_Netflix"] >= 0, :]
top_73 = top_73["Time_to_Netflix"].value_counts().reset_index().sort_values('index').set_index('index')
top_73.head(30)
```

```python
# Create a bar chart
top_73.plot(kind = "bar", figsize = (16,8))
plt.xlabel("Difference in Years")
plt.ylabel("Number of Movies/TV Shows")
plt.title("Difference between Release Year and Year Added to Netflix")
plt.savefig("Images/Difference between Release Year and Year Added to Netflix.png")
plt.show()
```

Difference Between Release Year and Year Added to Netflix

```
netflix_df.sort_values("Time_to_Netflix")
```

	Movie/TV Show	Title	Country of Production	Date Added to Netflix	Release Year	Parental Rating	Duration	Genre	Calendar Date Added to Netflix	Year Added to Netflix	...	Country 4	Country 5	Count
2890	TV Show	Jack Taylor	United States, Ireland	March 31, 2013	2016	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Dramas	March 31	2013	...	NaN	NaN	Na
5103	TV Show	Sense8	United States	December 23, 2016	2018	TV-MA	2 Seasons	Crime TV Shows, TV Dramas, TV Mysteries	December 23	2016	...	NaN	NaN	Na
3640	TV Show	Maradona in Mexico	Argentina, United States, Mexico	November 13, 2019	2020	TV-MA	1 Season	Docuseries, Spanish-Language TV Shows	November 13	2019	...	NaN	NaN	Na
2145	TV Show	Fuller House	United States	December 6, 2019	2020	TV-PG	5 Seasons	TV Comedies	December 6	2019	...	NaN	NaN	Na
2776	Movie	Incoming	Serbia, United States	October 26, 2018	2019	TV-MA	89 min	Action & Adventure, Sci-Fi & Fantasy	October 26	2018	...	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7107	Movie	Why We Fight: The Battle of	United States	March 31, 2017	1943	TV-PG	82 min	Documentaries	March 31	2017	...	NaN	NaN	Na



Difference Between Release Year vs. Year Added to Netflix

2. Country Collaboration

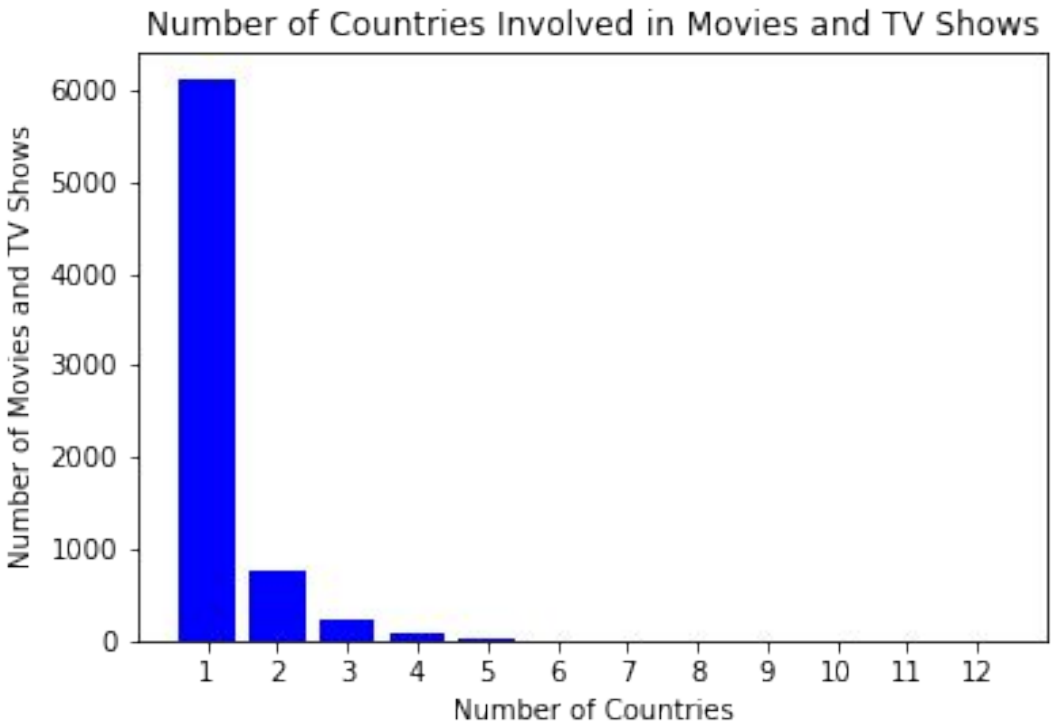
```python
# Get country count for each row
netflix_df["country_count"] = netflix_df["Country of Production"].str.split(", ").apply(lambda x: len(x))
```

```python
# Determine value counts of the number of countries involved
netflix_df["country_count"].value_counts().sort_values(ascending=False)
```

```
1     6115
2      760
3      240
4       96
5       32
6       14
7        5
8        1
12       1
10       1
Name: country_count, dtype: int64
```

```python
# Create bar graph
country_count = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
movies_shows = [6115, 760, 240, 96, 32, 14, 5, 1, 0, 1, 9, 1]
x_axis = np.arange(len(movies_shows))
plt.bar(x_axis, movies_shows, color="b", align="center")
tick_locations = [value for value in x_axis]
plt.xticks(tick_locations, country_count)
plt.title("Number of Countries Involved in Movies and TV Shows")
plt.xlabel("Number of Countries")
plt.ylabel("Number of Movies and TV Shows")
plt.savefig("Images/Number of Countries Involved in Movies and TV Shows on Netflix.png")
plt.show()
```

Number of Countries Involved in Movies and TV Shows



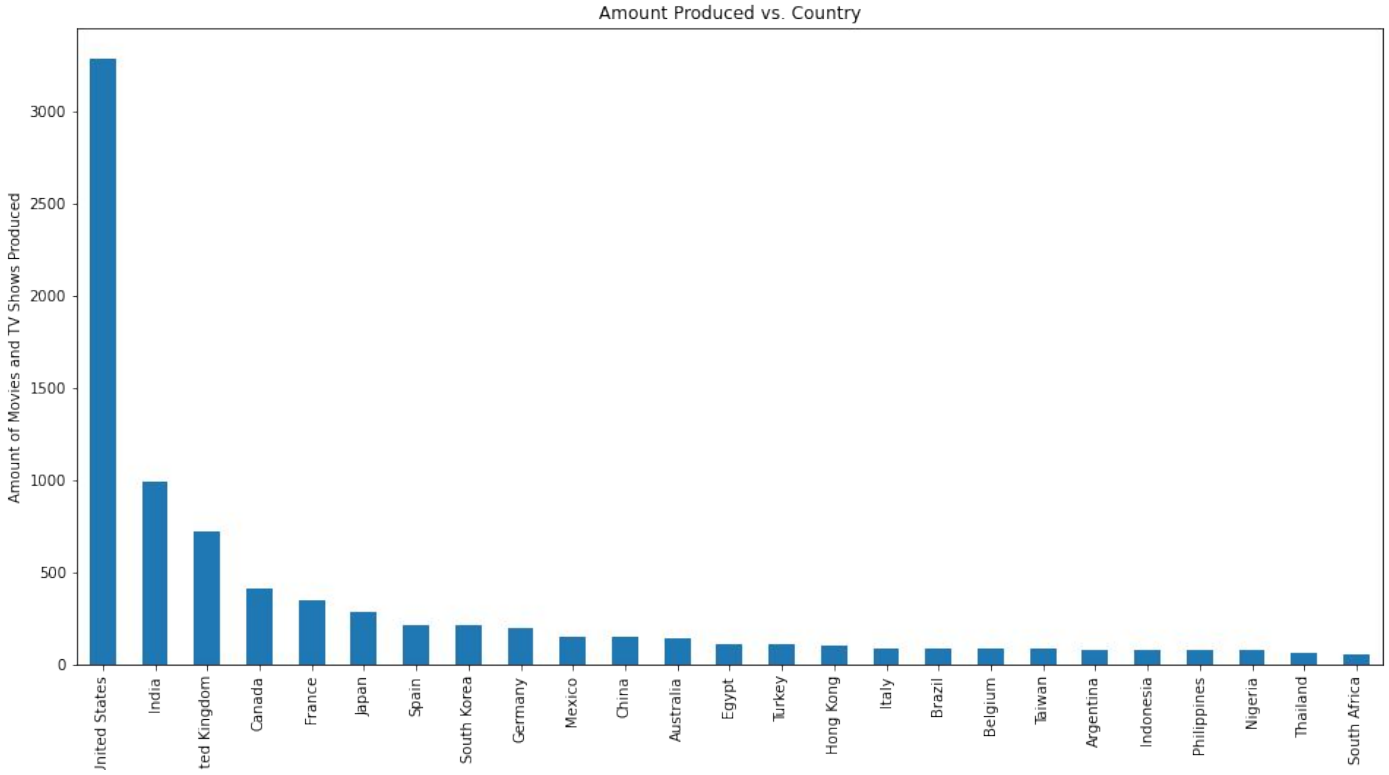
Number of Countries Involved in Movies and TV Shows

```python
# Determine the top 25 countries
countrycount = netflix_df["Country of Production"].str.get_dummies(sep=', ').sum().sort_values(ascending=False)
top_25 = countrycount.head(25)
top_25
```

```
United States     3287
India              990
United Kingdom     721
Canada             412
France             349
```

```python
# Create bar chart
top_25.plot(kind = "bar", figsize = (16,8))
plt.title("Amount Produced vs. Country")
plt.xlabel("Countries")
plt.ylabel("Amount of Movies and TV Shows Produced")
plt.savefig("Images/Amount Produced vs. Country.png")
plt.show()
```

Amount Produced vs. Country



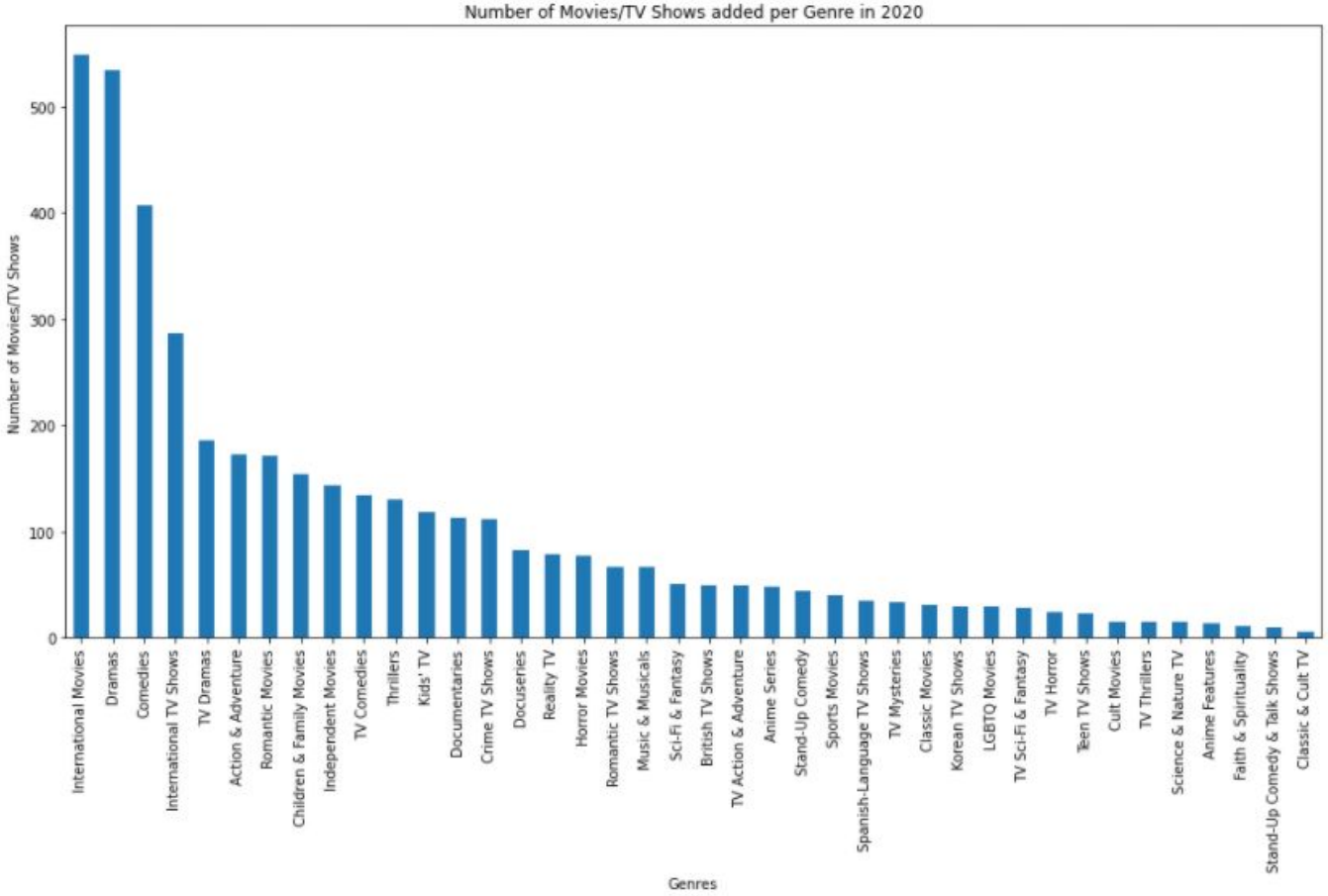
Amount Produced vs. Country

```python
# Show data frame of movies and tv shows added in 2020
netflix_2020 = netflix_df.loc[netflix_df["Year Added to Netflix"] == 2020]
```

```python
# Comma separate "Genre"
Genre = netflix_2020.Genre.str.get_dummies(sep=', ').sum().sort_values(ascending=False)
top_40 = Genre.head(40)
```

```python
# Create bar chart
top_40.plot(kind = "bar", figsize = (16,8))
plt.title("Number of Movies and TV Shows Added per Genre in 2020")
plt.xlabel("Genres")
plt.ylabel("Number of Movies/TV Shows")
plt.savefig("Images/Number of Movies and TV Shows Added per Genre in 2020.png")
plt.show()
```

Number of Movies and TV Shows Added per Genre in 2020



Number of Movies and TV Shows Added per Genre in 2020

3. Difficulties

Country of Production
Sweden, Czech Republic, United Kingdom, Denmar...
India
Australia
United Kingdom, Canada, United States

```
In [15]: # Get rid of spaces in "Country of Production"
         #netflix_final = [x.strip() for x in netflix_final["Country of Production"].split(",")]
         #netflix_final

In [16]: netflix_final["Country of Production"] = netflix_final["Country of Production"].str.replace(", ", ",")

In [17]: netflix_final["Country of Production"] = netflix_final["Country of Production"].str.strip()

In [18]: netflix_final["Country of Production"] = netflix_final["Country of Production"].str.split(",")

In [19]: #netflix_final["Country of Production"] = netflix_final["Country of Production"].str.strip()

In [20]: #netflix_final["Country of Production"] = netflix_final["Country of Production"].apply(lambda x: pd.Series(x))

In [21]: #netflix_final["Country of Production"] = netflix_final["Country of Production"].str.strip('[]').astype(str)
```

```
['United States']      2546
['India']               923
['United Kingdom']      396
 'United States']       341
['United States'        328
['Japan']               224
['South Korea']         183
['United Kingdom'       180
['Canada']              177
['Spain']               134
['France']              115
['Egypt']               101
['Mexico']              100
['Turkey']              100
 'Canada']               99
['Canada'                82
 'France']               82
['France'                81
['Australia']            81
 'United Kingdom']       80
dtype: int64
```

Difficulties

```python
# Determine the top 25 countries
countrycount = netflix_df["Country of Production"].str.get_dummies(sep=', ').sum().sort_values(ascending=False)
top_25 = countrycount.head(25)
top_25
```

```
United States     3287
India              990
United Kingdom     721
Canada             412
France             349
Japan              285
Spain              215
South Korea        212
Germany            199
Mexico             154
China              147
Australia          142
Egypt              110
Turkey             108
Hong Kong          102
Italy               89
Brazil              88
Belgium             85
Taiwan              85
Argentina           82
Indonesia           80
Philippines         78
Nigeria             76
Thailand            65
South Africa        54
dtype: int64
```

pandas.get_dummies

`pandas.get_dummies`(*data, prefix=None, prefix_sep='_', dummy_na=False, columns=None, sparse=False, drop_first=False, dtype=None*) ¶ [source]

Convert categorical variable into dummy/indicator variables.

Parameters:

data : *array-like, Series, or DataFrame*

Data of which to get dummy indicators.

prefix : *str, list of str, or dict of str, default None*

String to append DataFrame column names. Pass a list with length equal to the number of columns when calling get_dummies on a DataFrame. Alternatively, *prefix* can be a dictionary mapping column names to prefixes.

prefix_sep : *str, default '_'*

If appending prefix, separator/delimiter to use. Or pass a list or dictionary as with *prefix*.

dummy_na : *bool, default False*

Add a column to indicate NaNs, if False NaNs are ignored.

columns : *list-like, default None*

Column names in the DataFrame to be encoded. If *columns* is None then all the columns with *object* or *category* dtype will be converted.

sparse : *bool, default False*

Whether the dummy-encoded columns should be backed by a `SparseArray` (True) or a regular NumPy array (False).

drop_first : *bool, default False*

Whether to get k-1 dummies out of k categorical levels by removing the first level.

dtype : *dtype, default np.uint8*

Data type for new columns. Only a single dtype is allowed.

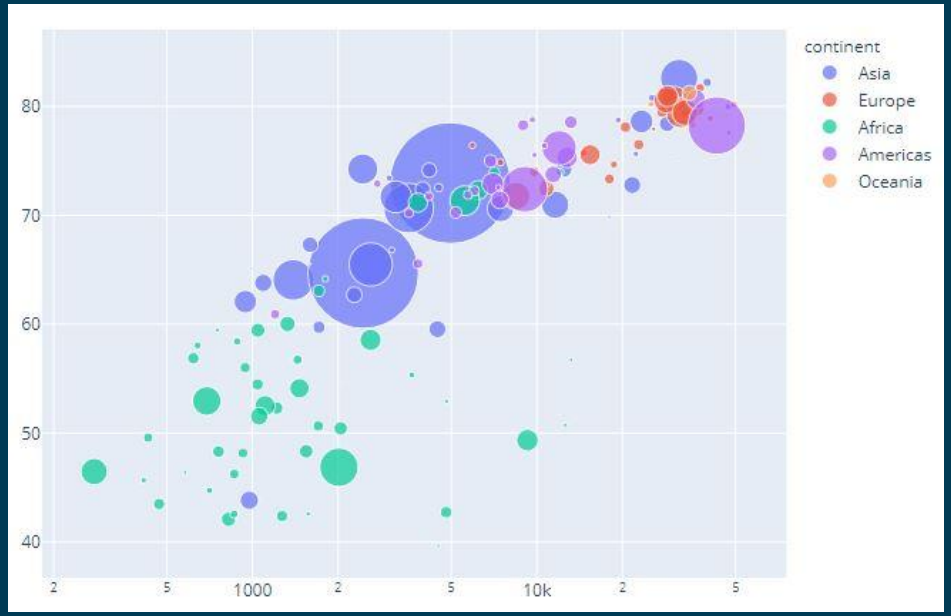
Returns:

DataFrame

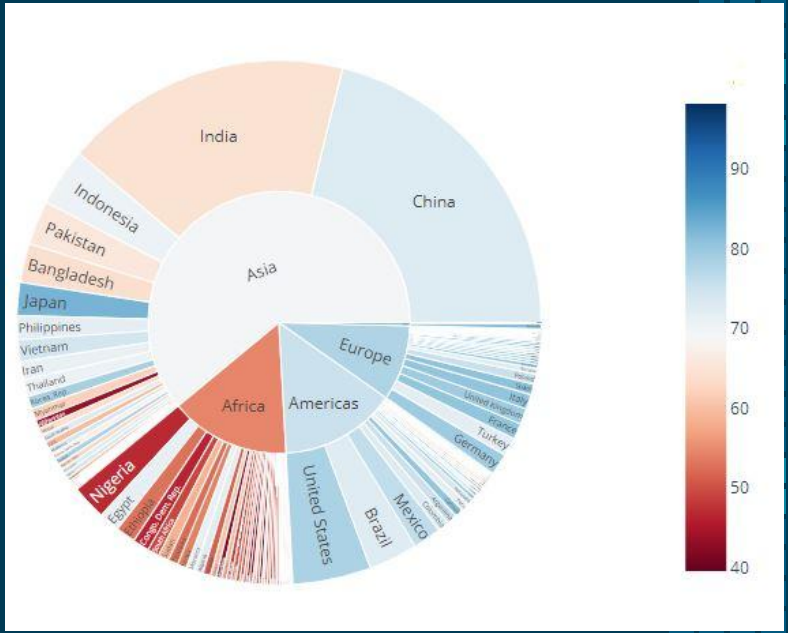
Dummy-coded data.

Difficulties

Plotly - Bubble Chart Example



Plotly - Sunburst Chart Example



https://plotly.com/python

Thank you!