Netflix! What started in 1997 as a DVD rental service has since exploded into one of the largest entertainment and media companies.

Given the large number of movies and series available on the platform, it is a perfect opportunity to flex your exploratory data analysis skills and dive into the entertainment industry.

You work for a production company that specializes in nostalgic styles. You want to do some research on movies released in the 1990's. You'll delve into Netflix data and perform exploratory data analysis to better understand this awesome movie decade!

You have been supplied with the dataset `netflix_data.csv`, along with the following table detailing the column names and descriptions. Feel free to experiment further after submitting!

The data

netflix_data.csv

Column	Description
`show_id`	The ID of the show
`type`	Type of show
`title`	Title of the show
`director`	Director of the show
`cast`	Cast of the show
`country`	Country of origin
`date_added`	Date added to Netflix
`release_year`	Year of Netflix release
`duration`	Duration of the show in minutes
`description`	Description of the show
`genre`	Show genre

Add blockquote

```
# Importing pandas and matplotlib
import pandas as pd
import matplotlib.pyplot as plt

# Read in the Netflix CSV as a DataFrame
netflix_df = pd.read_csv("netflix.csv")

type(netflix_df)
netflix_df.head(10)
```

```
{"summary":"{\n  \"name\": \"netflix_df\",\n  \"rows\": 10,\n
\"fields\": [\n    {\n      \"column\": \"show_id\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n
\"s10\",\n          \"s3\",\n          \"s7\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"type\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"TV Show\",\n          \"Movie\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"title\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n
\"1920\",\n          \"23:59\"\n        ],\n        \"semantic_type\":
```

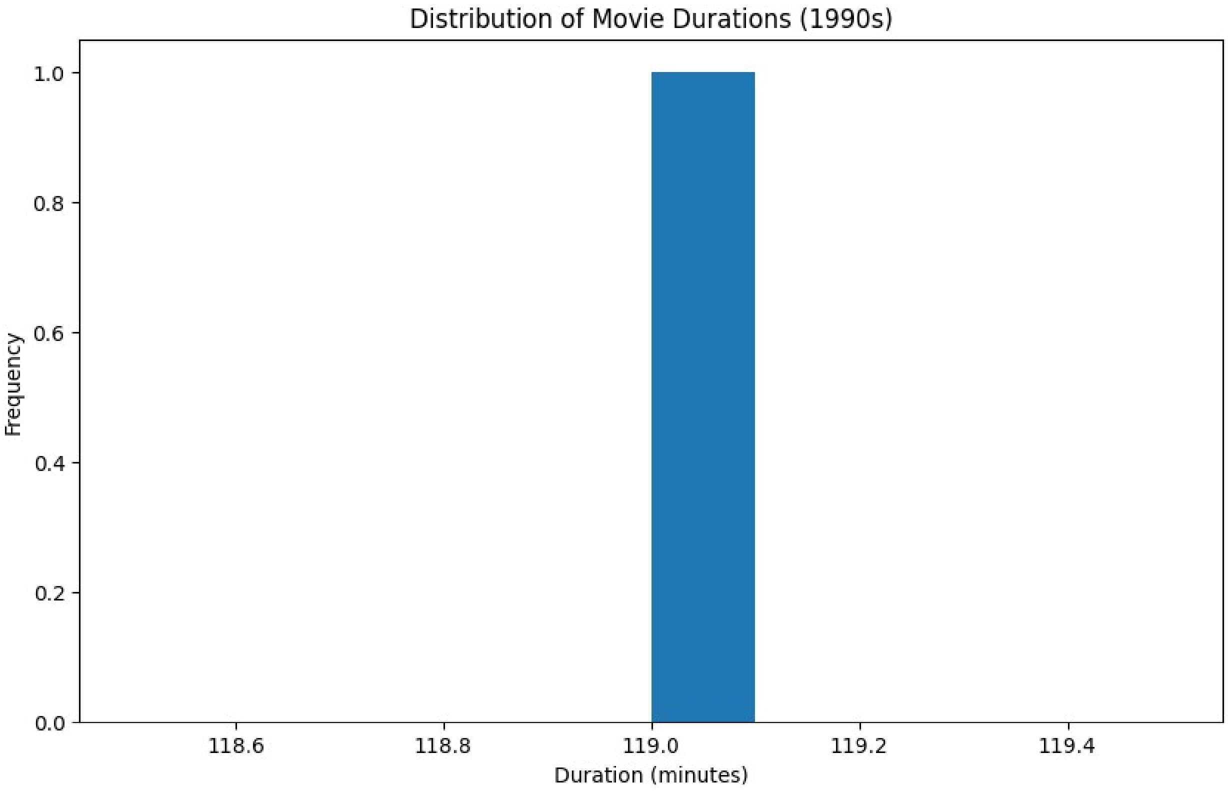
```
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"director\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 10,\n        \"samples\":
[\n          \"Vikram Bhatt\",\n          \"Gilbert Chan\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"cast\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n
\"\\\"Rajneesh Duggal, Adah Sharma, Indraneil Sengupta, Anjori Alagh,
Rajendranath Zutshi, Vipin Sharma, Amin Hajee, Shri Vallabh
Vyas\\\"\",\n          \"\\\"Tedd Chan, Stella Chung, Henley Hii,
Lawrence Koh, Tommy Kuan, Josh Lai, Mark Lee, Susan Leong, Benjamin
Lim\\\"\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"country\",\n      \"properties\": {\n        \"dtype\": \"string\",\
n        \"num_unique_values\": 6,\n        \"samples\": [\n
\"Mexico\",\n          \"Singapore\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"date_added\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n
\"\\\"December 15, 2017\\\"\",\n          \"\\\"December 20,
2018\\\"\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"release_year\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 6,\n        \"min\": 1997,\n
\"max\": 2019,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          2016,\n          2011\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"duration\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 38,\n
\"min\": 1,\n        \"max\": 143,\n        \"num_unique_values\":
10,\n        \"samples\": [\n          143,\n          78\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"description\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n          \"An
architect and his wife move into a castle that is slated to become a
luxury hotel. But something inside is determined to stop the
renovation.\",\n          \"\\\"When an army recruit is found dead,
his fellow soldiers are forced to confront a terrifying secret that's
haunting their jungle island training camp.\\\"\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"genre\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\": 4,\n
\"samples\": [\n          \"Horror Movies\",\n
\"International TV\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"netflix_df"}
```

```
df_90s = netflix_df[(netflix_df['type'] == 'Movie') &
(netflix_df['release_year'] >= 1990) & (netflix_df['release_year'] <
2000)]

print(df_90s.describe())
```

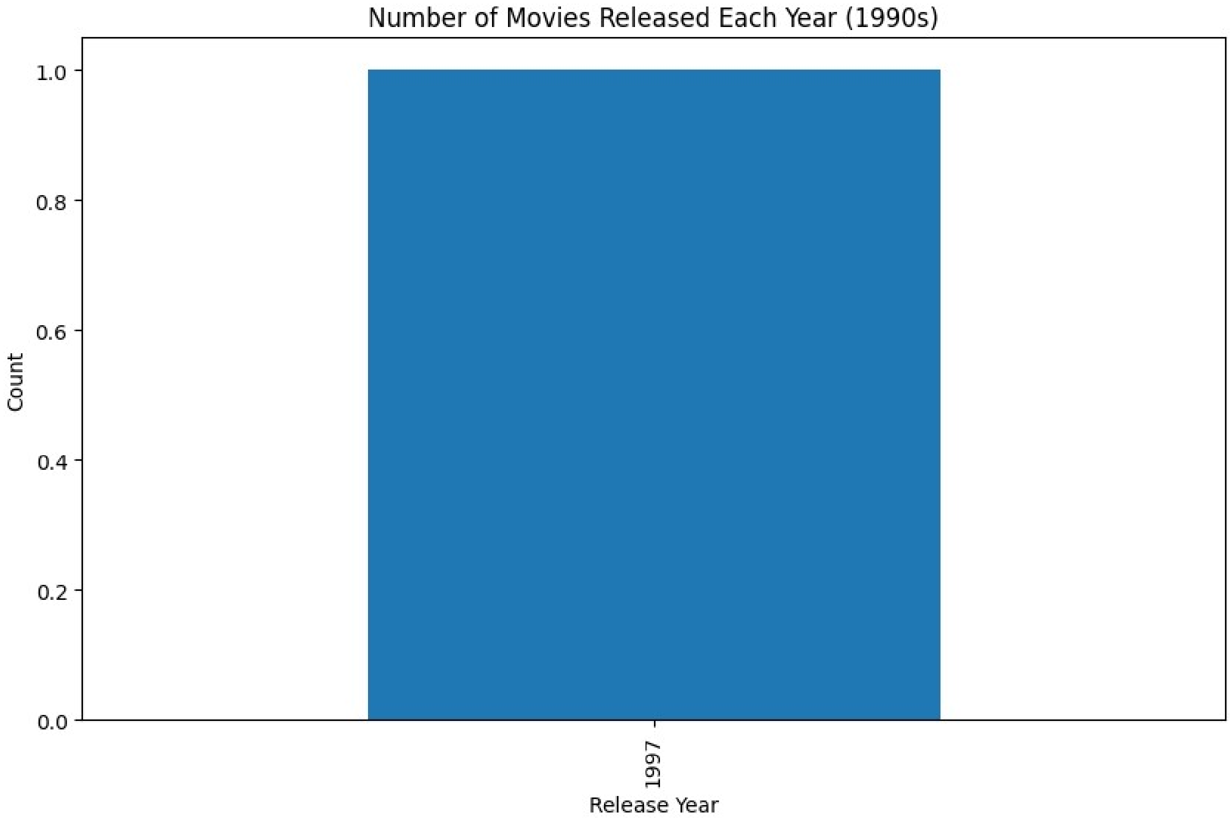
```
       release_year  duration
count           1.0       1.0
mean         1997.0     119.0
std             NaN       NaN
min          1997.0     119.0
25%          1997.0     119.0
50%          1997.0     119.0
75%          1997.0     119.0
max          1997.0     119.0
```

```
plt.figure(figsize=(10, 6))
plt.hist(df_90s['duration'])
plt.title('Distribution of Movie Durations (1990s)')
plt.xlabel('Duration (minutes)')
plt.ylabel('Frequency')
plt.show()
```



```
plt.figure(figsize=(10, 6))
df_90s['release_year'].value_counts().sort_index().plot(kind='bar')
```

```
plt.title('Number of Movies Released Each Year (1990s)')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.show()
```



```
f_90s = netflix_df[(netflix_df['type'] == 'Movie') &
(netflix_df['release_year'] >= 1990) & (netflix_df['release_year'] <
2000)]

# Find the most frequent movie duration
duration = int(df_90s['duration'].mode()[0])

# Print the most frequent movie duration
print(f"The most frequent movie duration in the 1990s is {duration}
minutes.")
```

```
The most frequent movie duration in the 1990s is 119 minutes.
```

```
df_90s = netflix_df[(netflix_df['type'] == 'Movie') &
(netflix_df['release_year'] >= 1990) & (netflix_df['release_year'] <
2000)]

# Filter for short action movies
short_action_movies = df_90s[(df_90s['duration'] < 90) &
(df_90s['genre'].str.contains('Action', case=False, na=False))]

# Count the number of short action movies
short_movie_count = short_action_movies.shape[0]

# Print the count of short action movies
print(f"The number of short action movies released in the 1990s is
{short_movie_count}.")
```

```
The number of short action movies released in the 1990s is 0.
```