In [1]:
```
# This Python 3 environment comes with many helpful analytics libraries instal
led
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/d
ocker-python
# For example, here's several helpful packages to load
#pip install dash
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import plotly.express as px

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will lis
t all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that ge
ts preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved
outside of the current session
```

In [3]:
```
df = pd.read_csv('netflix_titles.csv')
df.head(3)
```

Out[3]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	du
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	9
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	Se
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	S

In [4]:
```
df.isna().sum()
```

Out[4]:
```
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 [5]:
```
df.dropna(inplace = True)
```

In [6]:
```
# Handle missing values
df.dropna(subset=["type", "release_year", "rating"], inplace=True)
```

In [7]:
```
df.isna().sum()
```

Out[7]:
```
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 [8]:
```
df.shape
```

Out[8]:
```
(5332, 12)
```

In [9]:
```
df.drop_duplicates(inplace=True)
```

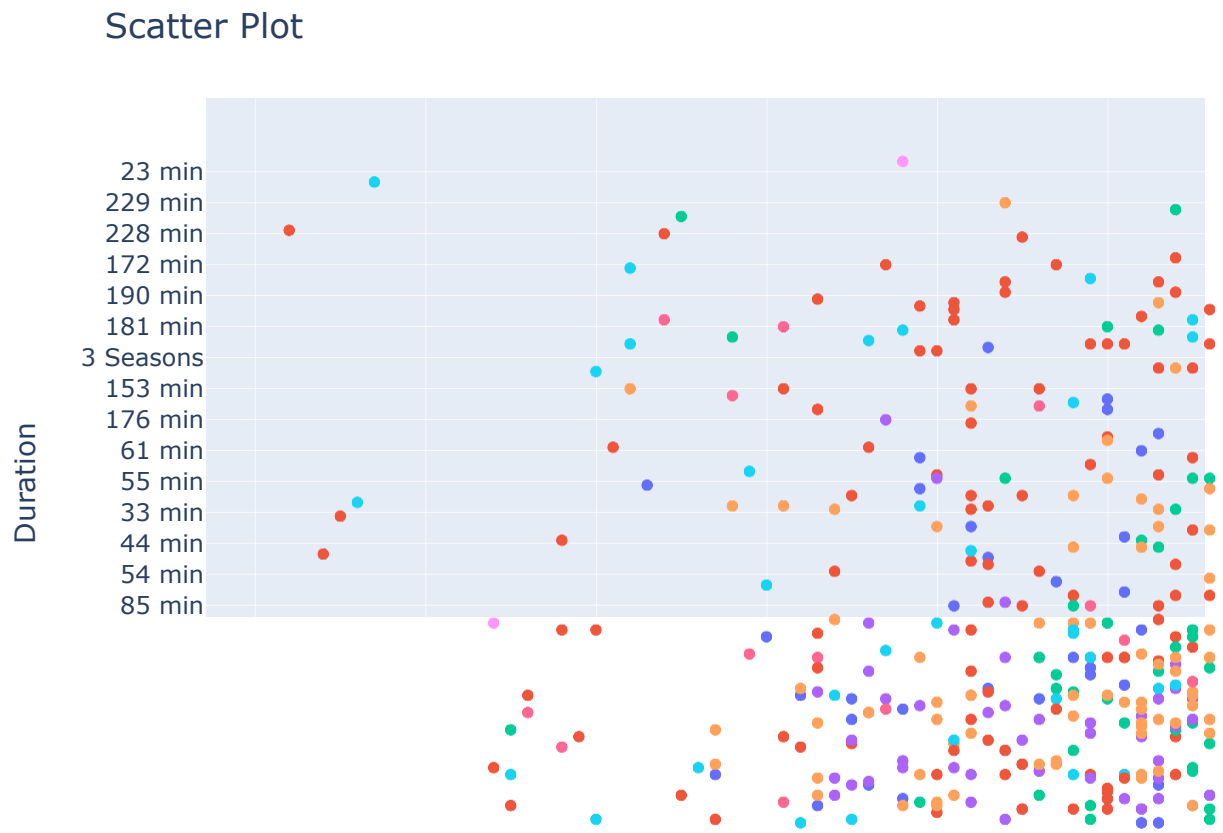
In [10]:
```
df.columns
```

Out[10]:
```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_adde
d',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

In [11]:
```
fig1 = px.histogram(df, x='type', color='type', title='Count of Types of Movie
s and TV Shows')
fig1.update_layout(
    template='plotly_dark',
    plot_bgcolor='rgba(0,0,0,0)',
    paper_bgcolor='rgba(0,0,0,0)',
    font_color="#FFFFEE"
).update_yaxes(showline=True, gridcolor='#FFFFEE')
```

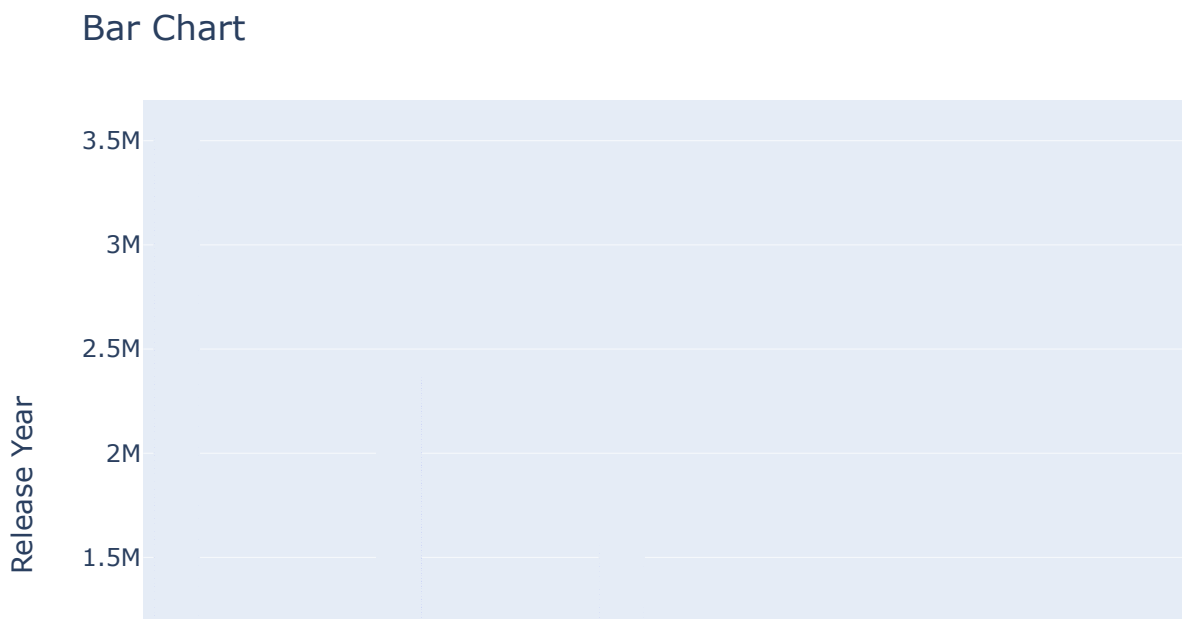
In [12]:
```
import plotly.express as px

# Scatter plot
fig_scatter = px.scatter(df, x='release_year', y='duration', color='rating')
fig_scatter.update_layout(
    title='Scatter Plot',
    xaxis_title='Release Year',
    yaxis_title='Duration',  # Add the missing closing parenthesis here
)
fig_scatter.show()
```



In [13]:
```
import plotly.express as px

# Bar chart
fig_bar = px.bar(df, x='rating', y='release_year', color='type', barmode='grou
p')
fig_bar.update_layout(
    title='Bar Chart',
    xaxis_title='Rating',
    yaxis_title='Release Year',  # Add the missing closing parenthesis here
)
fig_bar.show()
```



In [14]:
```
# Clean and preprocess the "rating" column
df["rating"] = pd.to_numeric(df["rating"], errors="coerce")  # Convert non-num
eric values to NaN

# Scatter plot: Release Year vs Rating
scatter_plot = px.scatter(
    df, x="release_year", y="rating", color="type",
    title="Netflix Movie and TV Show Ratings by Release Year and Type"
)

# Bar chart: Average Rating by Type
average_rating_by_type = df.groupby("type", as_index=False)["rating"].mean()
bar_chart = px.bar(
    average_rating_by_type, x="type", y="rating",
    title="Average Ratings of Netflix Content by Type"
)
```

In [ ]:
```
import dash
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output

# Initialize the Dash app
app = dash.Dash(__name__)

# Define the layout of the dashboard
app.layout = html.Div([
    html.H1("Interactive Netflix Dashboard"),

    dcc.Graph(id="scatter-plot", figure=scatter_plot),

    dcc.Graph(id="bar-chart", figure=bar_chart),

    # Additional interactive components can be added here
])

if __name__ == '__main__':
    app.run_server(debug=True)
```

In [ ]:
```

```