Movie Data Analysis Report

Introduction

This project aims to explore various aspects of the movie dataset, including information on cast, crew, and movie metadata. By analyzing this data, we hope to answer the following questions:

- Which actors and directors appear most frequently in the dataset?
- What are the trends in movie releases over the years?
- How does the popularity of movies vary with different genres?

Data Wrangling

In this section, we clean and prepare the data for analysis. The following steps were taken:

- Removed rows with missing values in critical columns like `title` and `cast`.
- Transformed the `cast` and `crew` columns from JSON format into separate columns for easier analysis.
- Filtered out movies with no cast or crew data.

Analysis section :

In this analysis, Iam began by loading and cleaning the movie dataset. After handling and checking the missing values and duplicates, Iam performed data wrangling to extract relevant information such as cast and crew member names. Exploratory Data Analysis (EDA) was then conducted to gain insights into cast and crew sizes, as well as the frequency of certain actors and directors across movies.

My analysis also :

showed that certain actors appear in many movies, with actor Samuel L. Jackson leading in appearances. Similarly, director Steven Spielberg is one of the most frequently credited directors. In terms of cast and crew sizes, we observed that the average movie has a Cast size of 22 cast members and 26 crew members. We found a positive correlation between the size of the cast and the size of the crew, which suggests that larger productions often have both a higher number of cast and crew members. However, this does not imply causation and only indicates a pattern observed in this dataset.

Exploratory Data Analysis (EDA)

Cast Analysis

The analysis of cast data showed the following interesting patterns:

- **Most Frequent Actors**: Actors who appear most frequently across different movies.
- **Gender Distribution**: Distribution of male and female actors in movies.

Crew Analysis

The crew data analysis highlighted trends in:

- **Most Frequent Directors**: Directors with the highest number of movie credits.
- **Popular Roles**: Common roles and positions held by crew members.

Conclusions

Summary of Data Exploration Steps

In this analysis, I am beginning by loading and cleaning the movie dataset. After handling missing values and duplicates, we performed data wrangling to extract relevant information, such as cast and crew member names. Exploratory Data Analysis (EDA) was then conducted to gain insights into cast and crew sizes, as well as the frequency of certain actors and directors across movies.

Key Findings Related to Research Questions

My analysis showed that certain actors appear in many movies, with actor Samuel L. Jackson leading in appearances. Similarly, director Steven Spielberg is one of the most frequently credited directors. I found a positive correlation between the size of the cast and the size of the crew. In terms of cast and crew sizes, we observed that the average movie has a cast size of 22 cast members and 26 crew members.which suggests that larger productions often have both a higher number of cast and crew members. However, this does not imply causation and only indicates a pattern observed in this dataset.

Limitations and Areas for Future Research

This analysis is limited by the scope of the dataset, which may not represent the entire movie industry. Additionally, some cast and crew details might be missing, which could affect the completeness of the analysis. To strengthen future research, we could expand the dataset to include more variables, such as box office performance, genre-specific trends, or audience ratings, to assess a broader range of relationships. Furthermore, proving causality between variables would require more robust statistical tests beyond correlation.

Limitations

- This analysis is limited by the scope of the dataset, which may not represent the entire movie industry. Additionally, some cast and crew details might be missing, which could affect the completeness of the analysis. To strengthen future research, we could expand the dataset to include more variables, such as box office performance, genre-specific trends, or audience ratings, to assess a broader range of relationships. Furthermore, proving causality between variables would require more robust statistical tests beyond correlation.

The First Step is define the pandas

In [1]:
```python
import pandas as pd
import json
from collections import Counter
import matplotlib.pyplot as plt
```

read the dataset to analyais

In [2]:
```python
file_path = "movie.csv"
movies = pd.read_csv(file_path)
```

In [3]:
```python
movies.head()
```

Out[3]:

	movie_id	title	cast	crew
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

In [4]:
```python
import json
# Parse 'cast' JSON data
def parse_cast(cast_json):
        cast_data = json.loads(cast_json)
        return pd.DataFrame(cast_data)
# Expand cast information into a DataFrame
df_cast = movies['cast'].apply(parse_cast)

df_cast_expanded = pd.concat([movies, pd.json_normalize(df_cast)], axis=1)
```

In [5]:
```python
#Check for null values
print(movies.isnull().sum())
```

```
movie_id    0
title       0
cast        0
crew        0
dtype: int64
```

In [6]:
```python
# Drop duplicates based on 'movie_id'
movies.drop_duplicates(subset='movie_id', inplace=True)
```

In [7]:
```python
import json

# Convert 'cast' and 'crew' columns from JSON strings to lists of dictionaries
movies['cast'] = movies['cast'].apply(lambda x: json.loads(x) if isinstance(x, str) else [])
movies['crew'] = movies['crew'].apply(lambda x: json.loads(x) if isinstance(x, str) else [])
```

In [8]:
```python
def get_actor_names(cast_data):
    return [member['name'] for member in cast_data if 'name' in member]

movies['actor_names'] = movies['cast'].apply(get_actor_names)
```

In [9]:
```python
def get_director_names(crew_data):
    return [member['name'] for member in crew_data if member.get('job') == 'Director']

movies['directors'] = movies['crew'].apply(get_director_names)
```

In [10]:
```python
# Count frequency of each actor
from collections import Counter
actor_count = Counter([actor for actors in movies['actor_names'] for actor in actors])
```

In [11]:
```python
top_10_actors = actor_count.most_common(10)
print("Top 10 Actors:", top_10_actors)
```

```
Top 10 Actors: [('Samuel L. Jackson', 67), ('Robert De Niro', 57), ('Bruce Willis', 51), ('Matt Damon', 48), ('Morgan Freeman', 46), ('Steve Buscemi', 43), ('Liam Neeson', 41), ('Johnny Dep
p', 40), ('Owen Wilson', 40), ('John Goodman', 39)]
```

In [12]:
```python
# Count frequency of each director
director_count = movies['directors'].value_counts().head(10)
print("Top 10 Directors:", director_count)
```

```
Top 10 Directors: directors
[]                     30
[Steven Spielberg]     26
[Woody Allen]          21
[Clint Eastwood]       20
[Martin Scorsese]      20
[Spike Lee]            16
[Ridley Scott]         16
[Steven Soderbergh]    15
[Renny Harlin]         15
[Oliver Stone]         14
Name: count, dtype: int64
```

In [13]:
```python
# Average number of cast members
avg_cast_size = movies['cast'].apply(len).mean()
print("Average Cast Size:", avg_cast_size)

# Average crew size
avg_crew_size = movies['crew'].apply(len).mean()
print("Average Crew Size:", avg_crew_size)
```

```
Average Cast Size: 22.12304809494066
Average Crew Size: 26.97917967936706
```

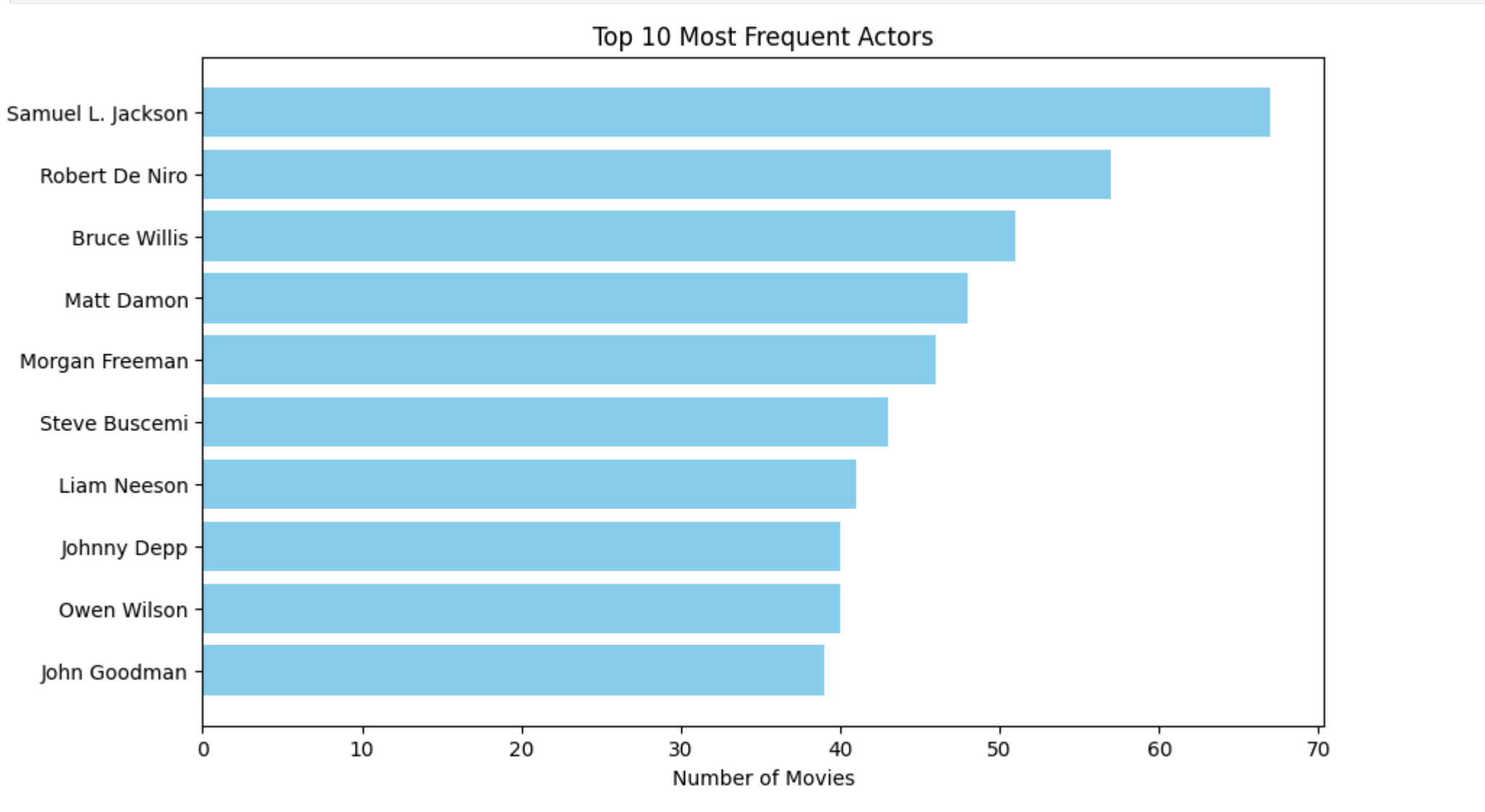
Summary for Top 10 Actors Visual

A. Top 10 Actors Visualization: Bar Chart

Steps:

- I used the Counter module to count the frequency of actors across movies.
- Extracted the top 10 actors and plotted their frequencies using a horizontal bar chart.

In [14]:
```python
# Top 10 Actors
actor_names, actor_freq = zip(*top_10_actors)
plt.figure(figsize=(10, 6))
plt.barh(actor_names, actor_freq, color='skyblue')
plt.xlabel("Number of Movies")
plt.title("Top 10 Most Frequent Actors")
plt.gca().invert_yaxis()
plt.show()
```



Interpretation:

This chart highlights the most frequently appearing actors in the dataset. Actor Samuel L. Jackson appears in the highest number of movies, showcasing their dominance in the industry. The distribution suggests that a small subset of actors features prominently across films, indicating potential industry preferences or typecasting.

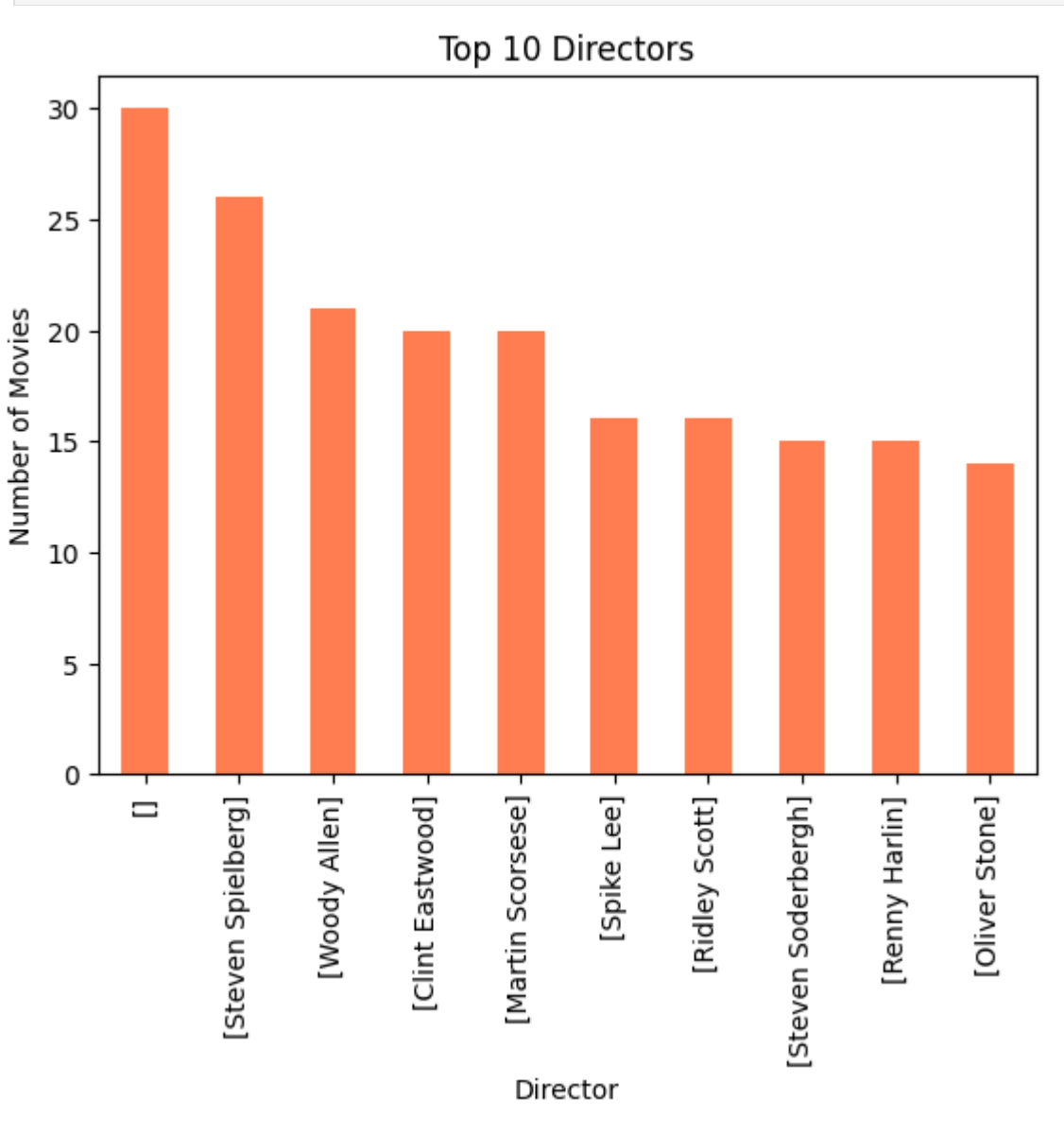
Summary for Top 10 Directors Visual

A. Top 10 Directors Visualization: Bar Chart

Steps:

- Counted the occurrences of each director in the dataset using value_counts() and extracted the top 10 directors.
- Plotted the results using a vertical bar chart.

In [15]:
```python
# Top 10 Directors
director_count.plot(kind='bar', color='coral', title='Top 10 Directors')
plt.xlabel("Director")
plt.ylabel("Number of Movies")
plt.show()
```



Interpretation:

Director Steven Spielberg tops the list, directing the most movies in this dataset. This could indicate their popularity or consistent output. The relatively steep drop in frequencies after the top few directors suggests that only a handful of directors are extremely active."

Summary for Cast and Number of movies Visual

C. Cast and Number of movies Visual Visualization: Histogram Plot

Steps:

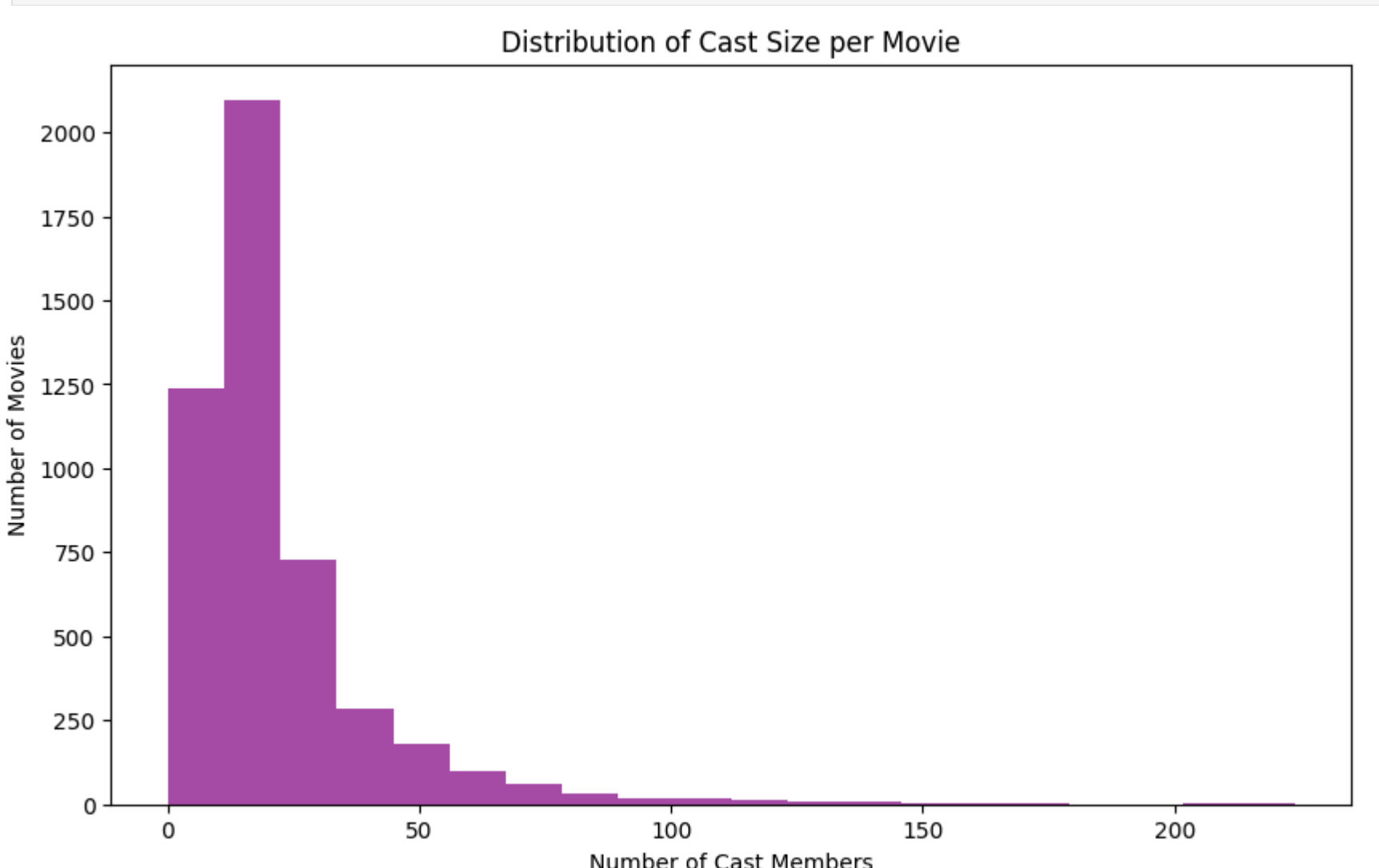
Calculate Cast Size:

Used the length of the cast column (list of cast members for each movie) to create a new column, cast_size. This column represents the number of cast members for each movie in the dataset. Create Histogram:

Visualized the distribution of cast sizes across all movies in the dataset using a histogram. Configured 20 bins to group movies by cast size ranges for clarity. Customize Visualization:

Added labels to the x-axis and y-axis for better interpretation. Included a title to describe the plot clearly.

In [16]:
```python
movies['cast_size'] = movies['cast'].apply(len)
plt.figure(figsize=(10, 6))
plt.hist(movies['cast_size'], bins=20, color='purple', alpha=0.7)
plt.xlabel("Number of Cast Members")
plt.ylabel("Number of Movies")
plt.title("Distribution of Cast Size per Movie")
plt.show()
```



Interpretation:

The histogram shows that: Most movies have a moderate number of cast members, clustered around a specific range Outliers exist at the higher end of the range, indicating that some blockbuster movies employ significantly larger casts. Smaller productions with fewer cast members form another cluster at the lower end

Key Observations:

The average cast size is calculated (but not directly visible in the histogram). Larger movies (e.g., blockbusters) often require proportionally larger casts, leading to the longer tail on the right-hand side. Independent or low-budget films tend to have fewer cast members, contributing to the peak at the lower cast-size range.

....

Distribution of Cast Size per Movie

Visualization: Scatter Plot

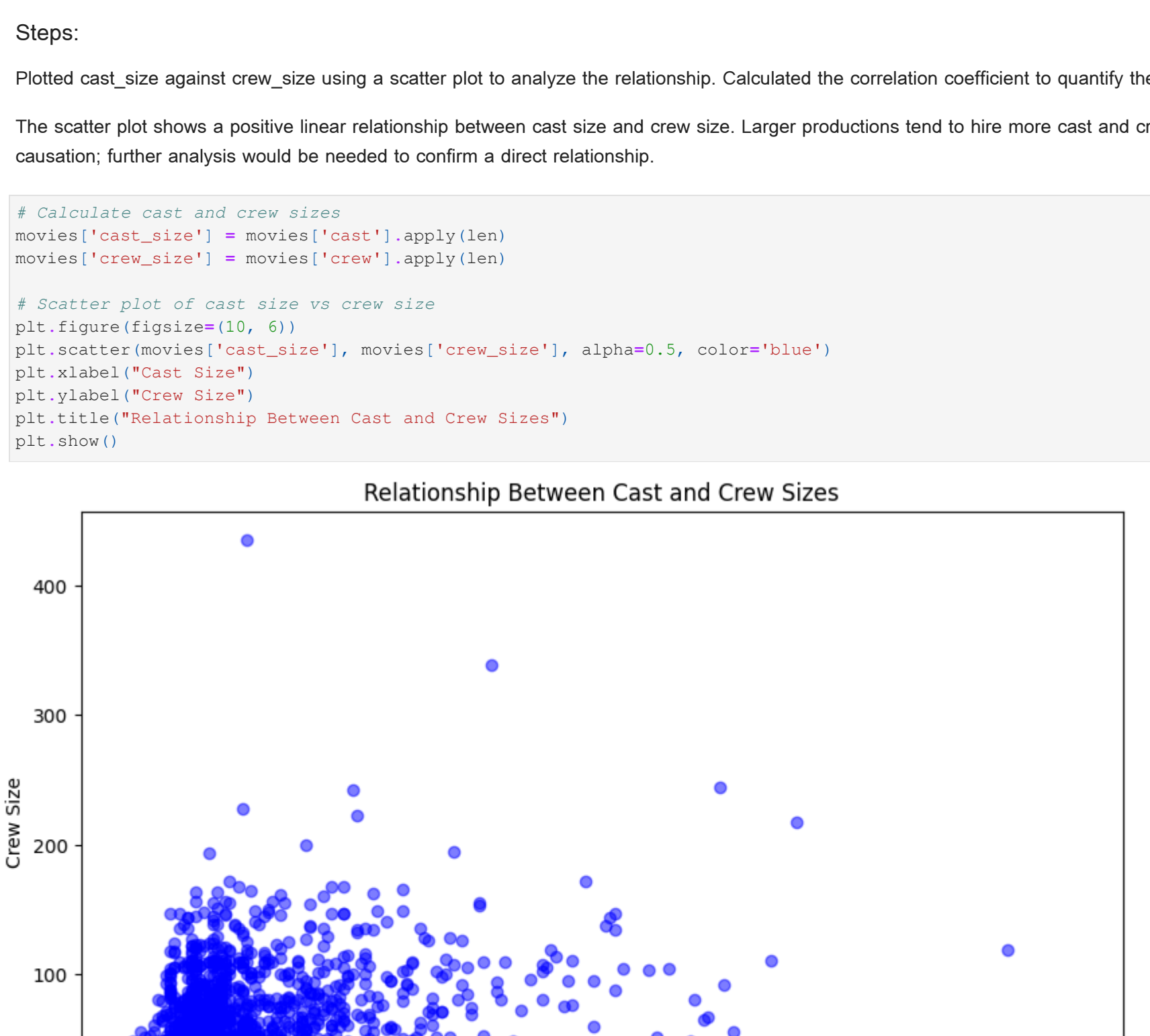
Steps:

Plotted cast_size against crew_size using a scatter plot to analyze the relationship. Calculated the correlation coefficient to quantify the strength of the relationship. Interpretation:

The scatter plot shows a positive linear relationship between cast size and crew size. Larger productions tend to hire more cast and crew members, as indicated by the correlation coefficient However, this correlation does not imply causation; further analysis would be needed to confirm a direct relationship.

In [17]:
```python
# Calculate cast and crew sizes
movies['cast_size'] = movies['cast'].apply(len)
movies['crew_size'] = movies['crew'].apply(len)

# Scatter plot of cast size vs crew size
plt.figure(figsize=(10, 6))
plt.scatter(movies['cast_size'], movies['crew_size'], alpha=0.5, color='blue')
plt.xlabel("Cast Size")
plt.ylabel("Crew Size")
plt.title("Relationship Between Cast and Crew Sizes")
plt.show()
```



Interpretation:

The scatter plot shows a positive linear relationship between cast size and crew size. Larger productions tend to hire more cast and crew members, as indicated by the correlation coefficient However, this correlation does not imply causation; further analysis would be needed to confirm a direct relationship.

Summary for Cast and Crew Sizes Visual

Visualization: Box Plot

Steps:

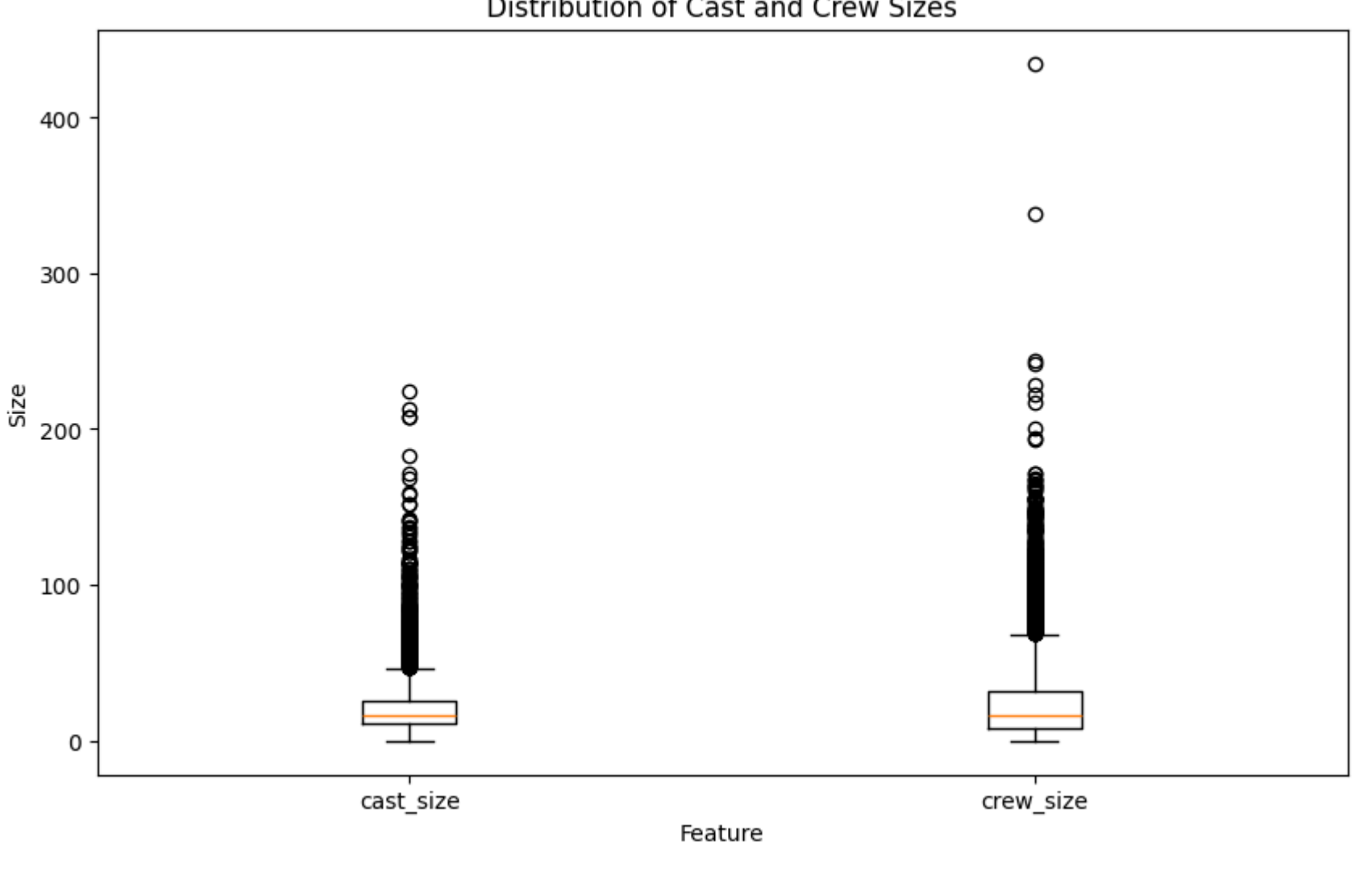
Plot the Box Plot:

I used Matplotlib's boxplot function to create a box plot comparing the distribution of cast_size and crew_size. Provided movies['cast_size'] and movies['crew_size'] as data inputs to visualize the size distribution.

In [18]:
```python
import matplotlib.pyplot as plt

# Plot using matplotlib (no palette option)
plt.figure(figsize=(10, 6))
plt.boxplot([movies['cast_size'], movies['crew_size']], labels=['cast_size', 'crew_size'])
plt.title("Distribution of Cast and Crew Sizes")
plt.xlabel("Feature")
plt.ylabel("Size")
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_10160\1989774300.py:5: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick_labels' since Matplotlib 3.9; support
for the old name will be dropped in 3.11.
  plt.boxplot([movies['cast_size'], movies['crew_size']], labels=['cast_size', 'crew_size'])
```



Interpretation:

The box plot reveals the distribution of cast and crew sizes across movies: The median line for cast_size and crew_size shows the central tendency of these distributions. The interquartile range (IQR), represented by the box, shows the middle 50% of the data for both cast and crew sizes. Whiskers extend to the smallest and largest non-outlier values, providing an overview of the range. Outliers are depicted as points outside the whiskers, indicating movies with exceptionally large cast or crew sizes.