```
import pandas as pd

movies_df = pd.read_csv('imdb_movies.csv')
movies_df.head()
```

```
                         names       date_x  score  \
0                    Creed III  03/02/2023    73.0
1     Avatar: The Way of Water  12/15/2022    78.0
2  The Super Mario Bros. Movie  04/05/2023    76.0
3                      Mummies  01/05/2023    70.0
4                    Supercell  03/17/2023    61.0

                                          genre  \
0                                 Drama, Action
1            Science Fiction, Adventure, Action
2  Animation, Adventure, Family, Fantasy, Comedy
3  Animation, Comedy, Family, Adventure, Fantasy
4                                        Action

                                            overview  \
0  After dominating the boxing world, Adonis Cree...
1  Set more than a decade after the events of the...
2  While working underground to fix a water main,...
3  Through a series of unfortunate events, three ...
4  Good-hearted teenager William always lived in ...

                                                crew  \
0  Michael B. Jordan, Adonis Creed, Tessa Thompso...
1  Sam Worthington, Jake Sully, Zoe Saldaña, Neyt...
2  Chris Pratt, Mario (voice), Anya Taylor-Joy, P...
3  Óscar Barberán, Thut (voice), Ana Esther Albor...
4  Skeet Ulrich, Roy Cameron, Anne Heche, Dr Quin...

                    orig_title     status            orig_lang
budget_x  \
0                    Creed III   Released              English
75000000.0
1     Avatar: The Way of Water   Released              English
460000000.0
2  The Super Mario Bros. Movie   Released              English
100000000.0
3                       Momias   Released   Spanish, Castilian
12300000.0
4                    Supercell   Released              English
77000000.0

        revenue country
0  2.716167e+08      AU
1  2.316795e+09      AU
2  7.244590e+08      AU
```

```
3  3.420000e+07      AU
4  3.409420e+08      US

# Check columns and data types
print(movies_df.info())

# Check for missing values
print(movies_df.isnull().sum())

# Basic statistics for numerical columns
print(movies_df.describe())


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10178 entries, 0 to 10177
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   names       10178 non-null  object
 1   date_x      10178 non-null  object
 2   score       10178 non-null  float64
 3   genre       10093 non-null  object
 4   overview    10178 non-null  object
 5   crew        10122 non-null  object
 6   orig_title  10178 non-null  object
 7   status      10178 non-null  object
 8   orig_lang   10178 non-null  object
 9   budget_x    10178 non-null  float64
 10  revenue     10178 non-null  float64
 11  country     10178 non-null  object
dtypes: float64(3), object(9)
memory usage: 954.3+ KB
None
names          0
date_x         0
score          0
genre         85
overview       0
crew          56
orig_title     0
status         0
orig_lang      0
budget_x       0
revenue        0
country        0
dtype: int64
              score      budget_x       revenue
count  10178.000000  1.017800e+04  1.017800e+04
mean      63.497052  6.488238e+07  2.531401e+08
std       13.537012  5.707565e+07  2.777880e+08
```

```
min         0.000000  1.000000e+00  0.000000e+00
25%        59.000000  1.500000e+07  2.858898e+07
50%        65.000000  5.000000e+07  1.529349e+08
75%        71.000000  1.050000e+08  4.178021e+08
max       100.000000  4.600000e+08  2.923706e+09
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

print(movies_df.columns)
```

```
Index(['names', 'date_x', 'score', 'genre', 'overview', 'crew',
'orig_title',
       'status', 'orig_lang', 'budget_x', 'revenue', 'country'],
      dtype='object')
```

```
df = movies_df.copy()

df = df[(df['budget_x'] > 0) & (df['revenue'] > 0)]

X = df[['budget_x', 'score']].fillna(0)  # Fill any missing scores
with 0
y = df['revenue']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression()
```

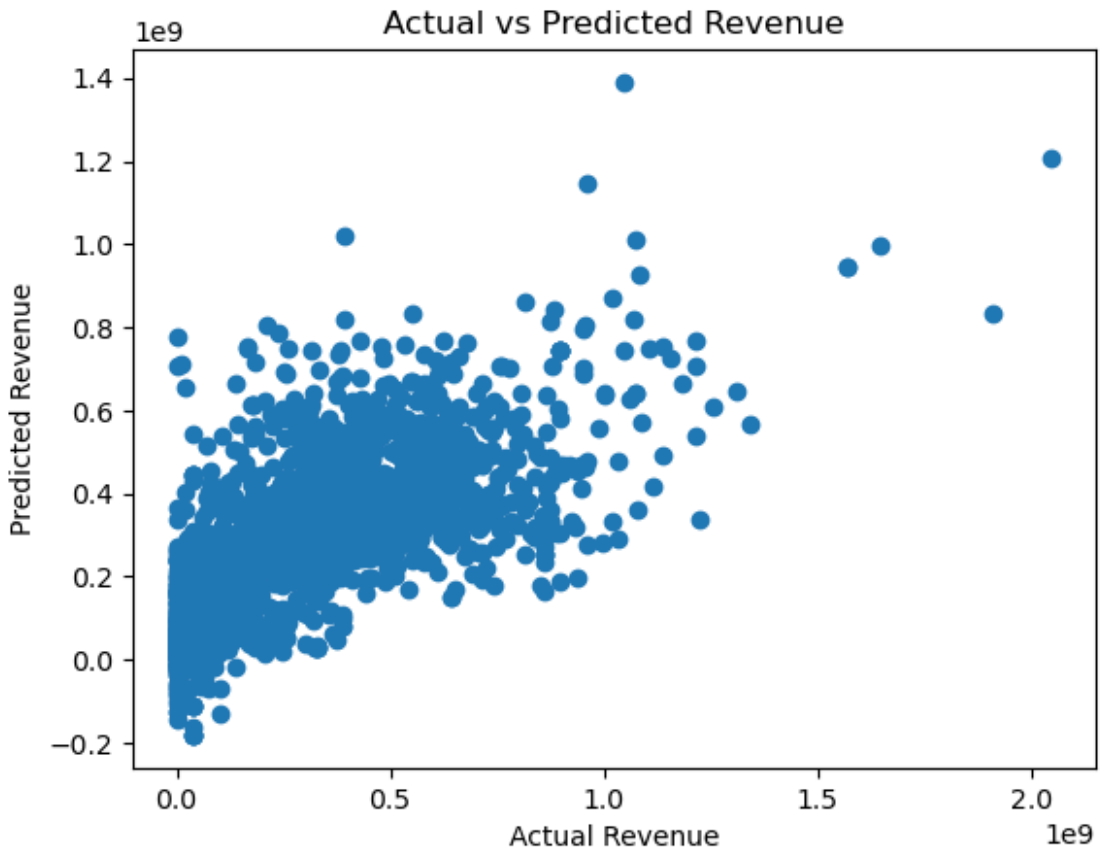
```
y_pred = model.predict(X_test)

print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("R-squared Score:", r2_score(y_test, y_pred))
```

```
Mean Squared Error: 3.841292289395672e+16
R-squared Score: 0.5173867000276696
```

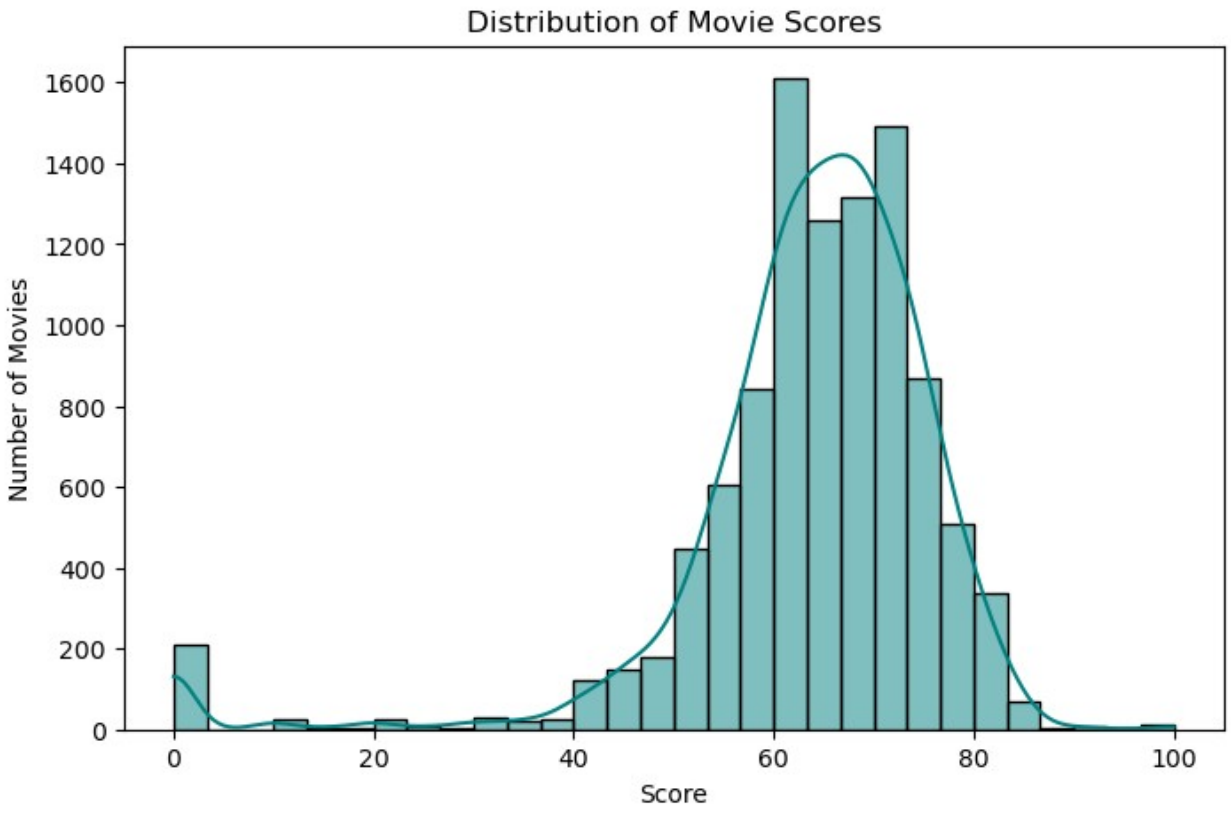
```
import matplotlib.pyplot as plt

plt.scatter(y_test, y_pred)
plt.xlabel("Actual Revenue")
plt.ylabel("Predicted Revenue")
plt.title("Actual vs Predicted Revenue")
plt.show()
```




```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8,5))
sns.histplot(movies_df['score'].dropna(), bins=30, kde=True,
color='teal')
plt.title('Distribution of Movie Scores')
plt.xlabel('Score')
plt.ylabel('Number of Movies')
plt.show()
```



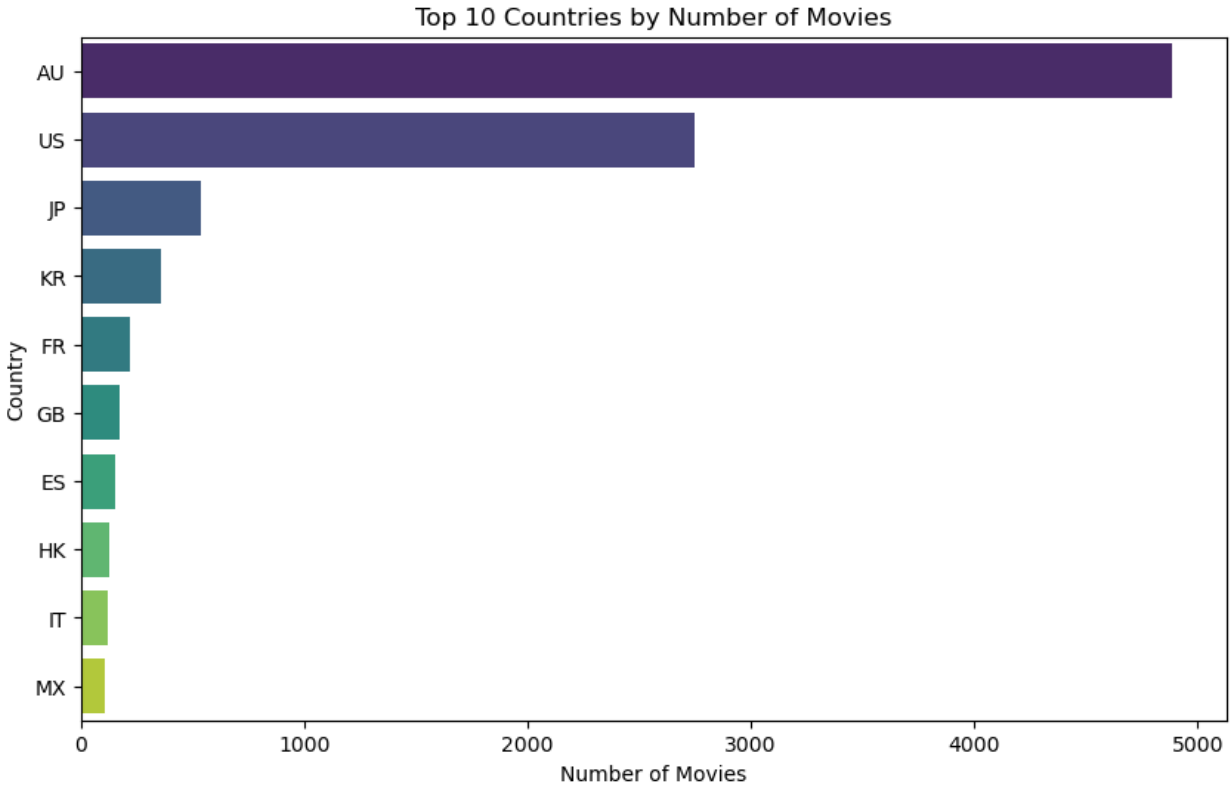

```
top_countries = movies_df['country'].value_counts().head(10)

plt.figure(figsize=(10,6))
sns.barplot(x=top_countries.values, y=top_countries.index,
palette='viridis')
plt.title('Top 10 Countries by Number of Movies')
plt.xlabel('Number of Movies')
plt.ylabel('Country')
plt.show()
```

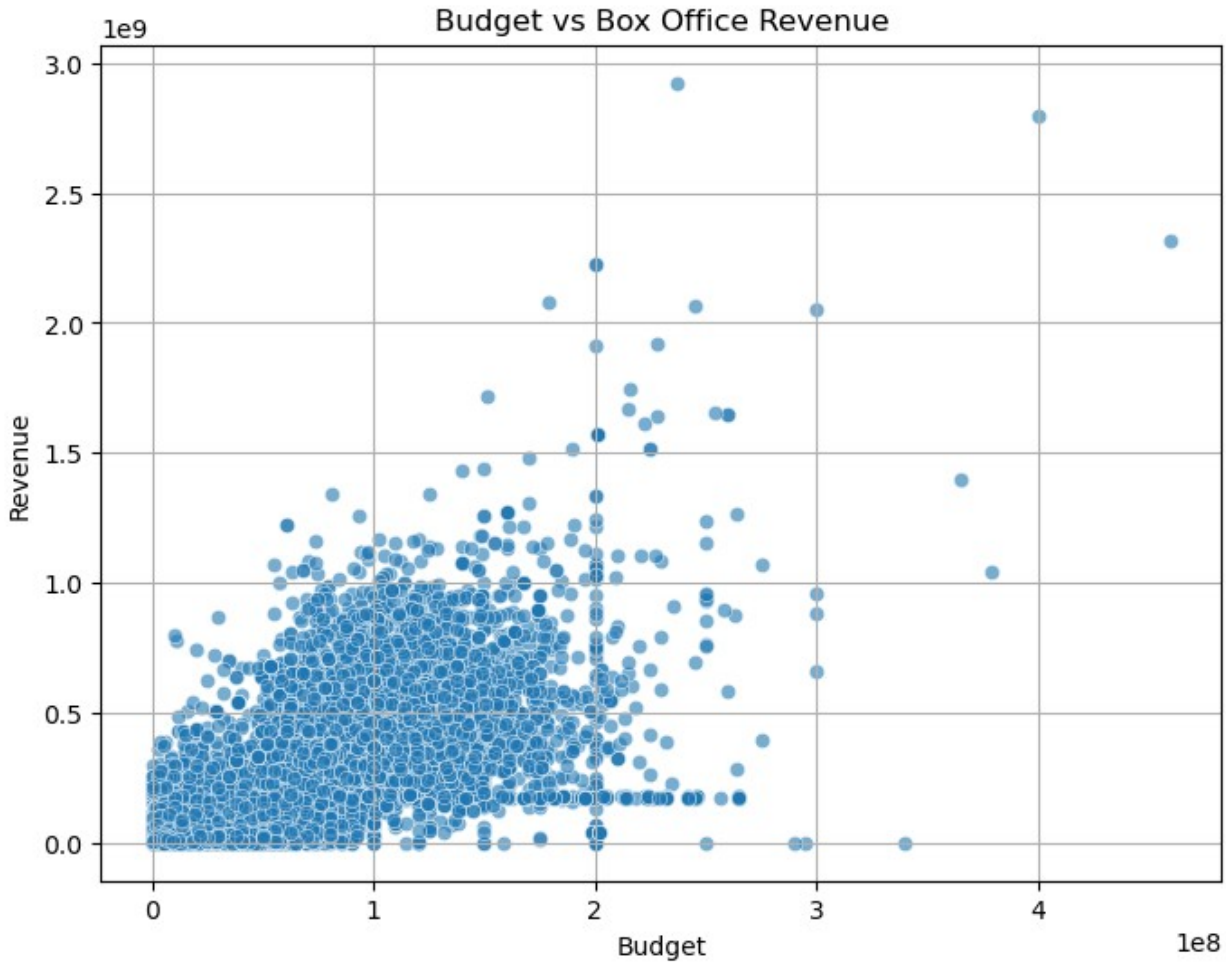
```
C:\Users\suraj\AppData\Local\Temp\ipykernel_12488\1868391935.py:4:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x=top_countries.values, y=top_countries.index,
palette='viridis')
```




```
plt.figure(figsize=(8,6))
sns.scatterplot(x='budget_x', y='revenue', data=movies_df, alpha=0.6)
plt.title('Budget vs Box Office Revenue')
plt.xlabel('Budget')
plt.ylabel('Revenue')
plt.grid(True)
plt.show()
```



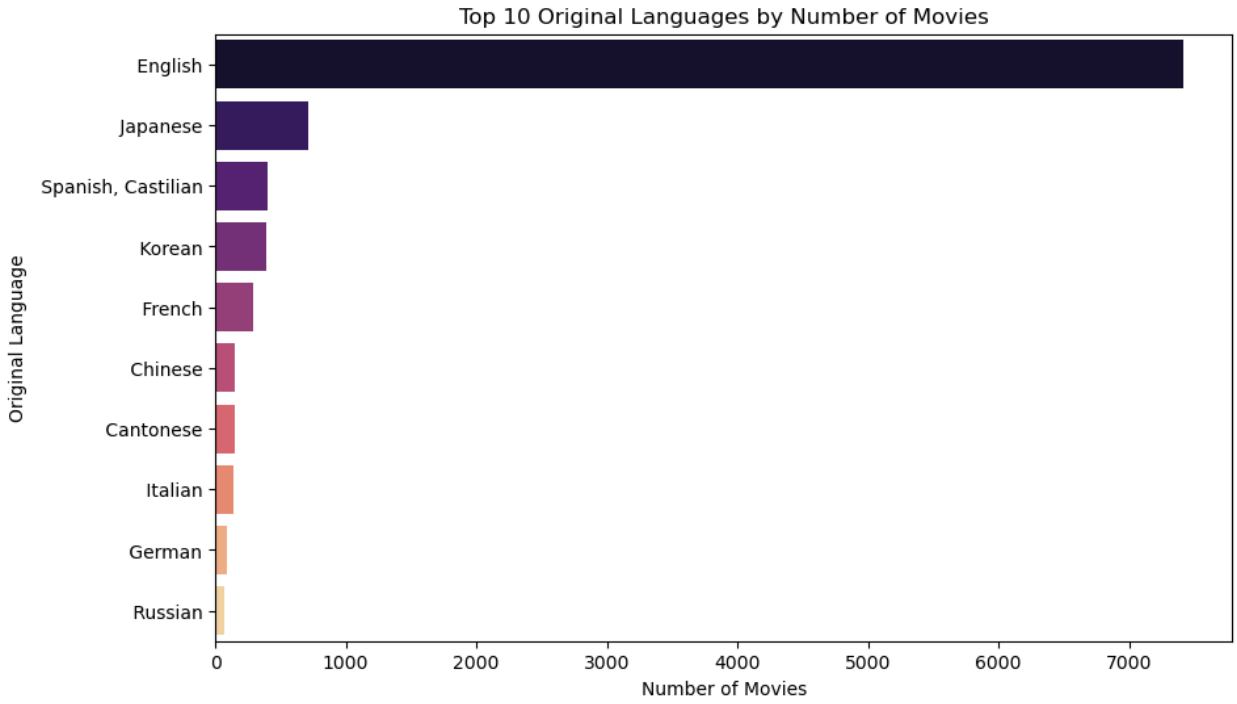

```
lang_counts = movies_df['orig_lang'].value_counts().head(10)

plt.figure(figsize=(10,6))
sns.barplot(x=lang_counts.values, y=lang_counts.index,
palette='magma')
plt.title('Top 10 Original Languages by Number of Movies')
plt.xlabel('Number of Movies')
plt.ylabel('Original Language')
plt.show()
```

```
C:\Users\suraj\AppData\Local\Temp\ipykernel_12488\3096629819.py:4:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x=lang_counts.values, y=lang_counts.index,
palette='magma')
```


Top 10 Original Languages by Number of Movies
English
Japanese
Spanish, Castilian
Korean
French
Chinese
Cantonese
Italian
German
Russian
Original Language
0
1000
2000
3000
4000
5000
6000
7000
Number of Movies