Exploring and Visualizing a Dataset

Author - Kanishk Karam

===

STEP1: Import required Libraries

In [1]:

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

STEP2: Import File/ Dataset

In [2]:

```
df = pd. read_csv('movies_metadata.csv')
df
```

```
C:\Users\kanis\AppData\Local\Temp\ipykernel_17096\744748106.py:1: DtypeWarning: Columns (10) have mixed types. S
pecify dtype option on import or set low_memory=False.
  df = pd. read_csv('movies_metadata.csv')
```

Out[2]:

	adult	belongs_to_collection	budget	genres	homepage	id	imdb_id	original_language
0	False	{'id': 10194, 'name': 'Toy Story Collection', ...	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...	http://toystory.disney.com/toy-story	862	tt0114709	en
1	False	NaN	65000000	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...	NaN	8844	tt0113497	en
2	False	{'id': 119050, 'name': 'Grumpy Old Men Collect...	0	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...	NaN	15602	tt0113228	en
3	False	NaN	16000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	NaN	31357	tt0114885	en
4	False	{'id': 96871, 'name': 'Father of the Bride Col...	0	[{'id': 35, 'name': 'Comedy'}]	NaN	11862	tt0113041	en
...	...	...	...	...	...	...	...	...
45461	False	NaN	0	[{'id': 18, 'name': 'Drama'}, {'id': 10751, 'n...	http://www.imdb.com/title/tt6209470/	439050	tt6209470	fa
45462	False	NaN	0	[{'id': 18, 'name': 'Drama'}]	NaN	111109	tt2028550	tl
45463	False	NaN	0	[{'id': 28, 'name': 'Action'}, {'id': 18, 'nam...	NaN	67758	tt0303758	en
45464	False	NaN	0	[]	NaN	227506	tt0008536	en
45465	False	NaN	0	[]	NaN	461257	tt6980792	en

45466 rows × 24 columns

2.1 Checking few top columns using head()

In [5]:

```
df.head()
```

Out[5]:

	adult	belongs_to_collection	budget	genres	homepage	id	imdb_id	original_language	original_titl
0	False	{'id': 10194, 'name': 'Toy Story Collection', ...	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...	http://toystory.disney.com/toy-story	862	tt0114709	en	Toy Stor
1	False	NaN	65000000	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...	NaN	8844	tt0113497	en	Jumar
2	False	{'id': 119050, 'name': 'Grumpy Old Men Collect...	0	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...	NaN	15602	tt0113228	en	Grumpie Old Me
3	False	NaN	16000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	NaN	31357	tt0114885	en	Waiting t Exhal
4	False	{'id': 96871, 'name': 'Father of the Bride Col...	0	[{'id': 35, 'name': 'Comedy'}]	NaN	11862	tt0113041	en	Father of th Bride Part

5 rows × 24 columns

STEP3: Understanding Data Structure

3.1 : summary of a DataFrame

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45466 entries, 0 to 45465
Data columns (total 24 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   adult                  45466 non-null  object 
 1   belongs_to_collection  4494 non-null   object 
 2   budget                 45466 non-null  object 
 3   genres                 45466 non-null  object 
 4   homepage               7782 non-null   object 
 5   id                     45466 non-null  object 
 6   imdb_id                45449 non-null  object 
 7   original_language      45455 non-null  object 
 8   original_title         45466 non-null  object 
 9   overview               44512 non-null  object 
 10  popularity             45461 non-null  object 
 11  poster_path            45080 non-null  object 
 12  production_companies   45463 non-null  object 
 13  production_countries   45463 non-null  object 
 14  release_date           45379 non-null  object 
 15  revenue                45460 non-null  float64
 16  runtime                45203 non-null  float64
 17  spoken_languages       45460 non-null  object 
 18  status                 45379 non-null  object 
 19  tagline                20412 non-null  object 
 20  title                  45460 non-null  object 
 21  video                  45460 non-null  object 
 22  vote_average           45460 non-null  float64
 23  vote_count             45460 non-null  float64
dtypes: float64(4), object(20)
memory usage: 8.3+ MB
```

3.2: gaining insights / summary

In [4]:

```
df.describe()
```

Out[4]:

	revenue	runtime	vote_average	vote_count
count	4.546000e+04	45203.000000	45460.000000	45460.000000
mean	1.120935e+07	94.128199	5.618207	109.897338
std	6.433225e+07	38.407810	1.924216	491.310374
min	0.000000e+00	0.000000	0.000000	0.000000
25%	0.000000e+00	85.000000	5.000000	3.000000
50%	0.000000e+00	95.000000	6.000000	10.000000
75%	0.000000e+00	107.000000	6.800000	34.000000
max	2.787965e+09	1256.000000	10.000000	14075.000000

STEP4: Cleaning The Data

In [5]:

```
#  Missing values handling
df.dropna(inplace=True)
```

In [6]:

```
# Cleaning duplicates
df.drop_duplicates(inplace=True)
```

STEP5: Visualizing The Data

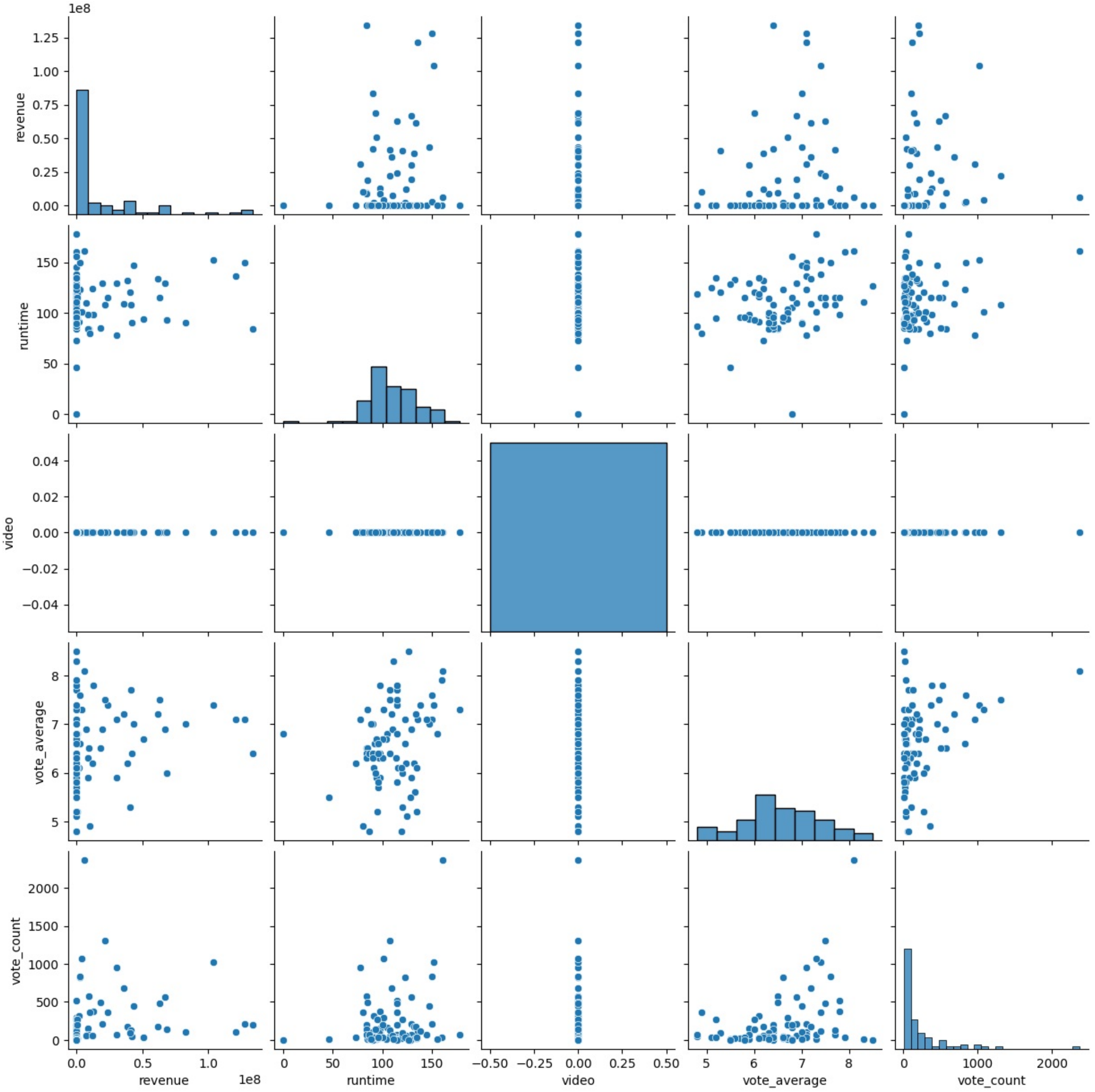
5.1: Pair Plots

In [7]:

```
sns.pairplot(df[df.original_language!='en'])
```

Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x215420c69c0>
```



In [9]:

```
df.revenue
```

Out[9]:

```
1237     8.332000e+07
8407     0.000000e+00
1909     2.674472e+08
993      2.635914e+08
4197     0.000000e+00
             ...
33356    8.205804e+08
44009    1.020063e+09
43294    3.501701e+08
44842    6.049421e+08
44274    3.699080e+08
Name: revenue, Length: 693, dtype: float64
```

In [11]:

```
df.revenue.describe()
```

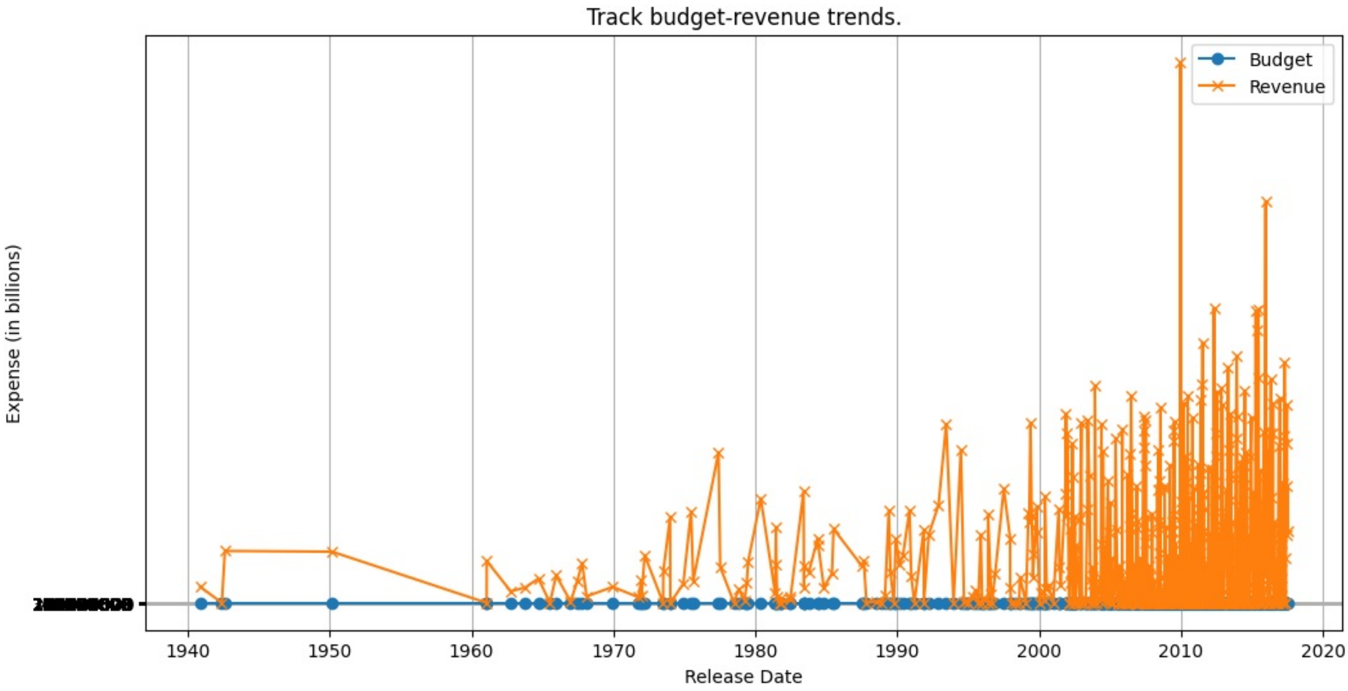
Out[11]:

```
count    6.930000e+02
mean     2.348037e+08
std      3.299089e+08
min      0.000000e+00
25%      0.000000e+00
50%      8.332000e+07
75%      3.613666e+08
max      2.787965e+09
Name: revenue, dtype: float64
```

5.2: Line PLots

In [8]:

```
# 'release_date' column to datetime convert it
df['release_date'] = pd.to_datetime(df['release_date'])

# Sort the DataFrame by release date
df.sort_values(by='release_date', inplace=True)

plt.figure(figsize=(12, 6))
plt.plot(df['release_date'], df['budget'], label='Budget', marker='o')
plt.plot(df['release_date'], df['revenue'], label='Revenue', marker='x')
plt.title('Track budget-revenue trends.')
plt.xlabel('Release Date')
plt.ylabel('Expense (in billions)')
plt.legend()
plt.grid(True)
plt.show()
```



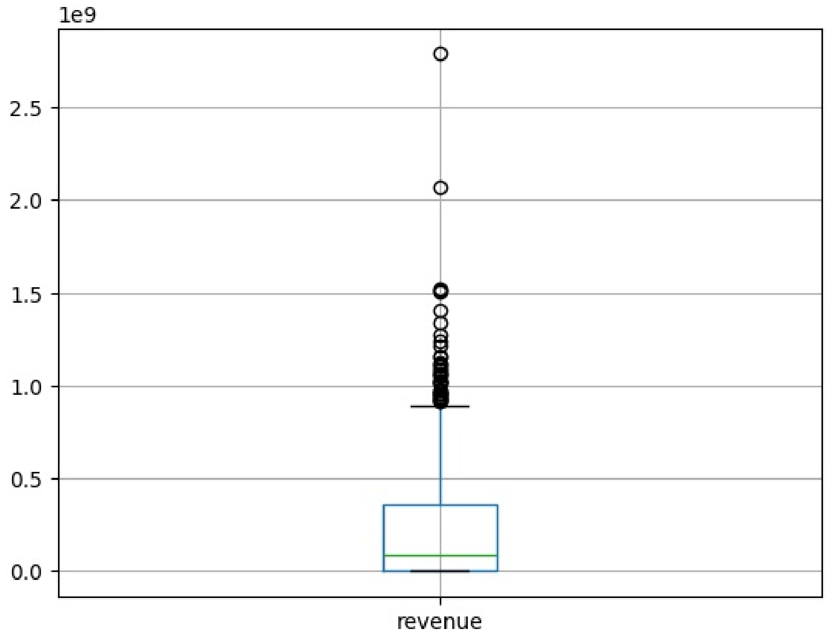
5.3 Box plot of Revenue

In [12]:

```
df.boxplot(column='revenue')
```

Out[12]:

```
<Axes: >
```

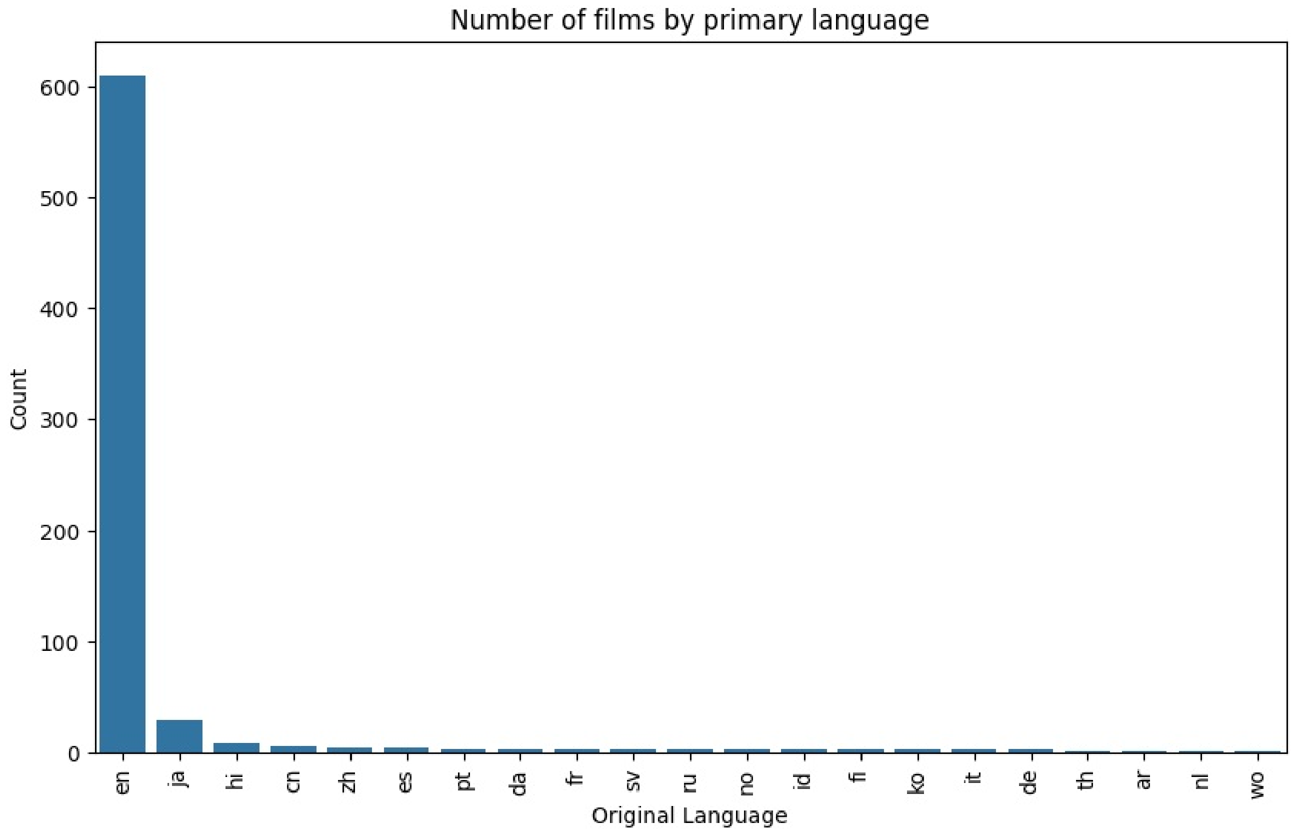


5.4: Bar Plot

In [13]:

```
plt.figure(figsize=(10, 6))
sns.countplot(x='original_language', data=df, order=df['original_language'].value_counts().index)
```

```
plt.title('Number of films by primary language')
plt.xlabel('Original Language')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



5.5: Distplot / Distribution plots of Revenue

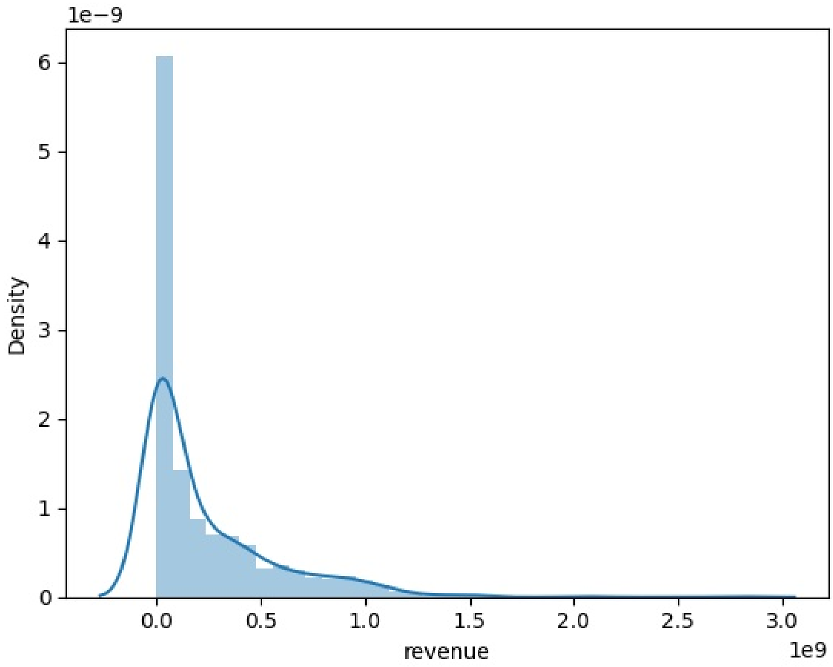
In [14]:

```
sns.distplot(df['revenue'])
```

```
C:\Users\kanis\AppData\Local\Temp\ipykernel_17096\2222233393.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['revenue'])
```

Out[14]:

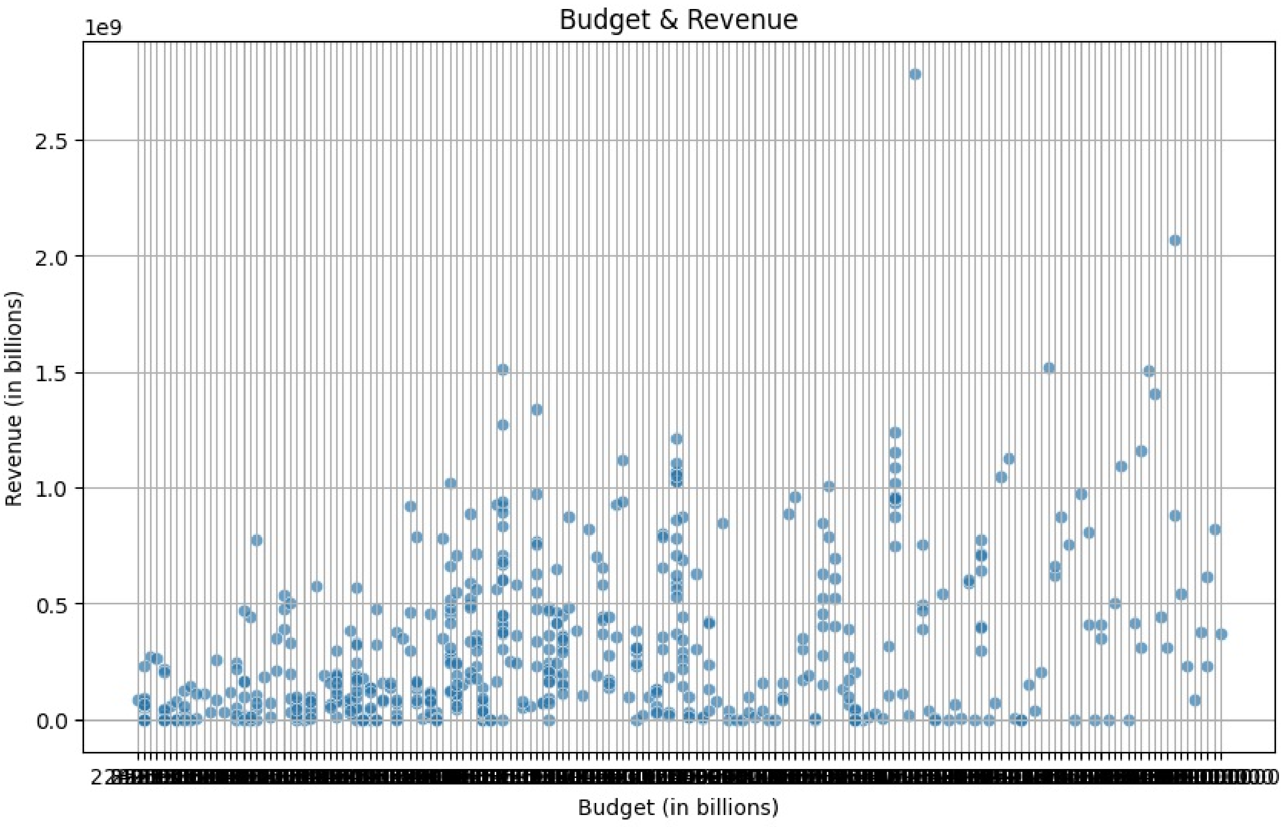
```
<Axes: xlabel='revenue', ylabel='Density'>
```



5.6: Scatter Plot

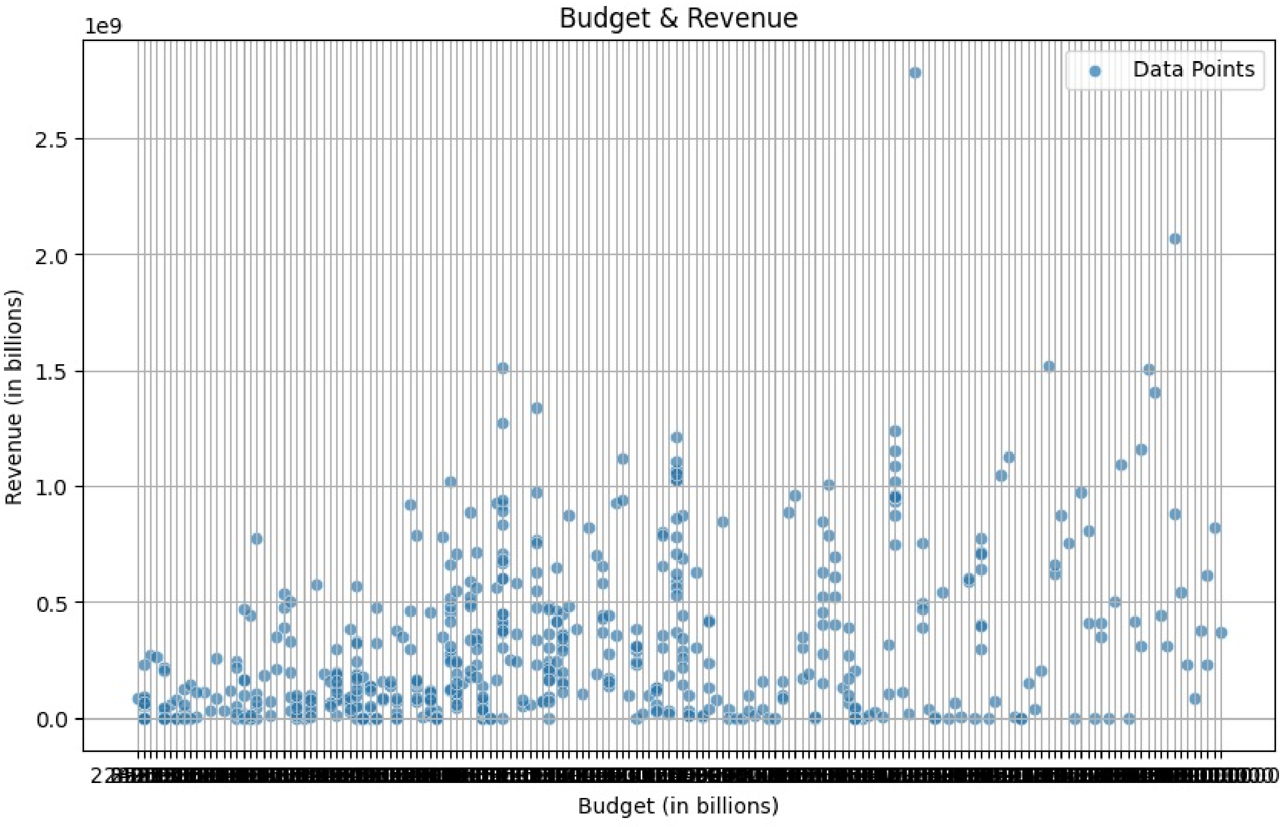
In [15]:

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='budget', y='revenue', data=df, alpha=0.7)
plt.title('Budget & Revenue')
plt.xlabel('Budget (in billions)')
plt.ylabel('Revenue (in billions)')
plt.grid(True)
plt.show()
```



5.7: Customizing Visualizations

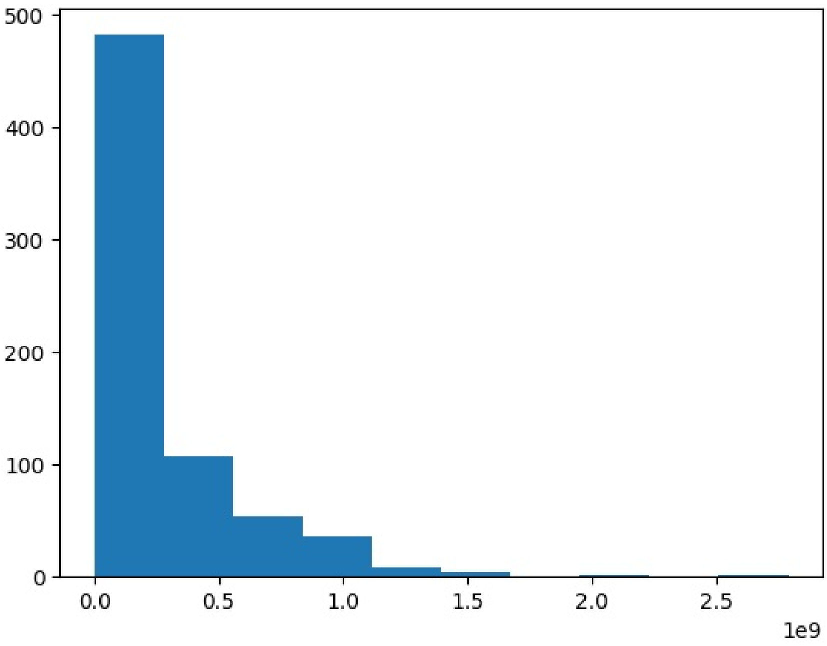
```
In [16]: plt.figure(figsize=(10, 6))
         scatter_plot = sns.scatterplot(x='budget', y='revenue', data=df, alpha=0.7)
         plt.title('Budget & Revenue')
         plt.xlabel('Budget (in billions)')
         plt.ylabel('Revenue (in billions)')
         plt.grid(True)

         # Customize legend
         scatter_plot.legend(['Data Points'], loc='upper right')

         plt.show()
```



5.8: Histogram

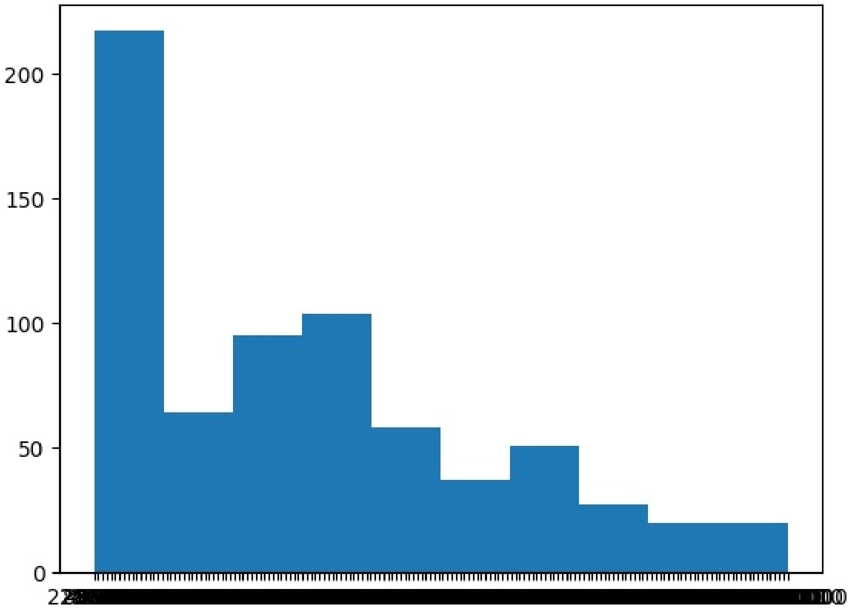
In [17]:

```
plt.hist(df.revenue)
plt.show()
```



In [18]:

```
plt.hist(df.budget)
plt.show()
```

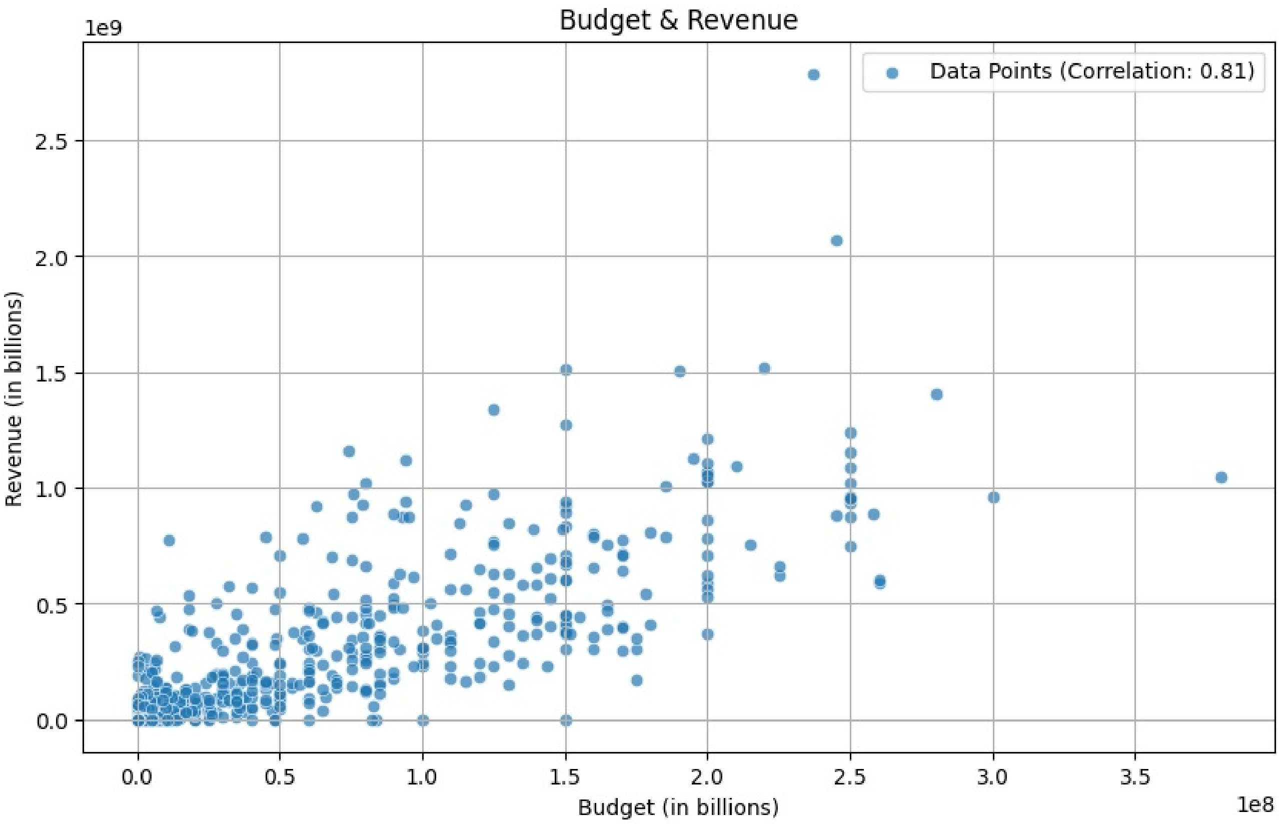


STEP6: Analysing Data and Insights

In [20]:

```
# Converting 'budget' and 'revenue' columns to numeric
df['budget'] = pd.to_numeric(df['budget'], errors='coerce')
df['revenue'] = pd.to_numeric(df['revenue'], errors='coerce')

# Calculate the correlation coefficient
correlation_coefficient = df['budget'].corr(df['revenue'])

plt.figure(figsize=(10, 6))
scatter_plot = sns.scatterplot(x='budget', y='revenue', data=df, alpha=0.7)
plt.title('Budget & Revenue')
plt.xlabel('Budget (in billions)')
plt.ylabel('Revenue (in billions)')
plt.grid(True)

# Customize legend
scatter_plot.legend([f'Data Points (Correlation: {correlation_coefficient:.2f})'], loc='upper right')

plt.show()
```

```
# Print the correlation coefficient
print(f"Correlation coefficient between 'budget' & 'revenue': {correlation_coefficient:.2f}")
```



```
Correlation coefficient between 'budget' & 'revenue': 0.81
```

6.1: Kurtosis

In [21]:
```
df.revenue.kurtosis()
```

Out[21]:
```
7.4693571169582365
```

6.2: Original Languages

In [22]:
```
df.original_language
```

Out[22]:
```
1237     en
8407     en
1909     en
993      en
4197     it
         ..
33356    en
44009    en
43294    en
44842    en
44274    en
Name: original_language, Length: 693, dtype: object
```

In [23]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 693 entries, 1237 to 44274
Data columns (total 24 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   adult                  693 non-null    object
 1   belongs_to_collection  693 non-null    object
 2   budget                 693 non-null    int64
 3   genres                 693 non-null    object
 4   homepage               693 non-null    object
 5   id                     693 non-null    object
 6   imdb_id                693 non-null    object
 7   original_language      693 non-null    object
 8   original_title         693 non-null    object
 9   overview               693 non-null    object
 10  popularity             693 non-null    object
 11  poster_path            693 non-null    object
 12  production_companies   693 non-null    object
 13  production_countries   693 non-null    object
 14  release_date           693 non-null    datetime64[ns]
 15  revenue                693 non-null    float64
 16  runtime                693 non-null    float64
 17  spoken_languages       693 non-null    object
 18  status                 693 non-null    object
 19  tagline                693 non-null    object
 20  title                  693 non-null    object
 21  video                  693 non-null    object
 22  vote_average           693 non-null    float64
 23  vote_count             693 non-null    float64
dtypes: datetime64[ns](1), float64(4), int64(1), object(18)
memory usage: 135.4+ KB
```

In [ ]:

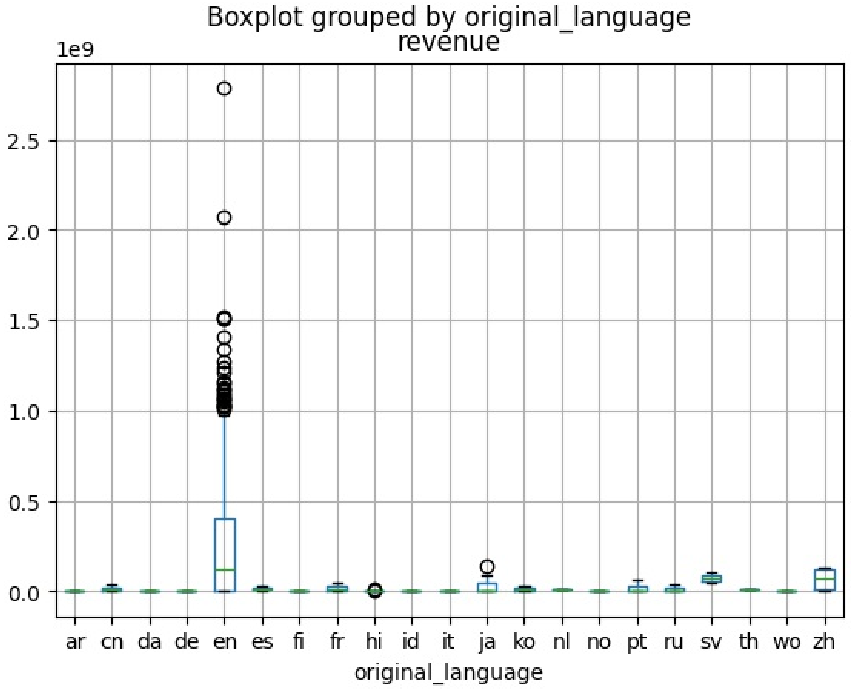
STEP7: Analysis of Budget & Revenue

In [27]:

```
df.boxplot(column='revenue', by='original_language')
```

Out[27]:

```
<Axes: title={'center': 'revenue'}, xlabel='original_language'>
```

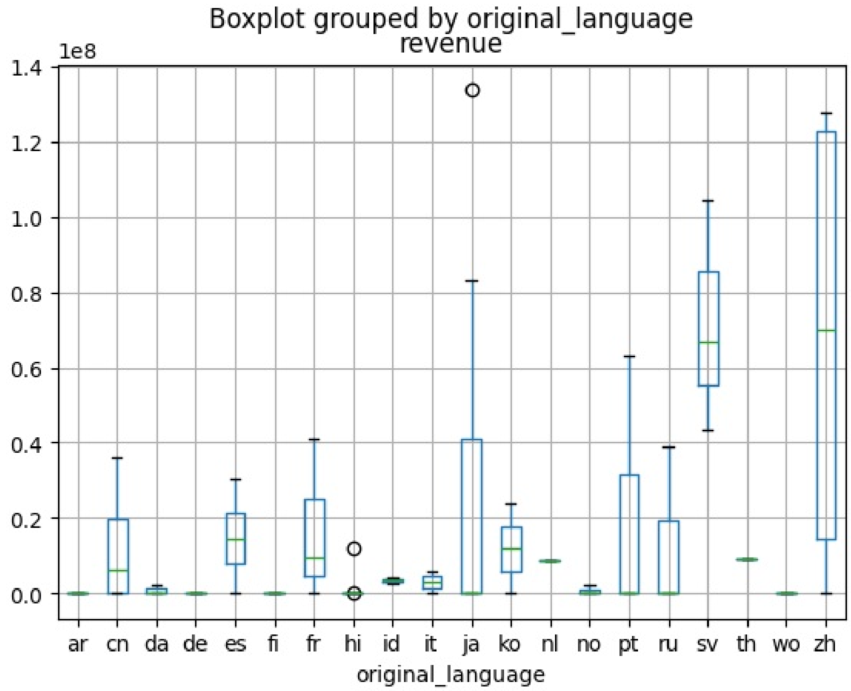


In [28]:

```
df[df.original_language!='en'].boxplot(column='revenue',by='original_language')
```

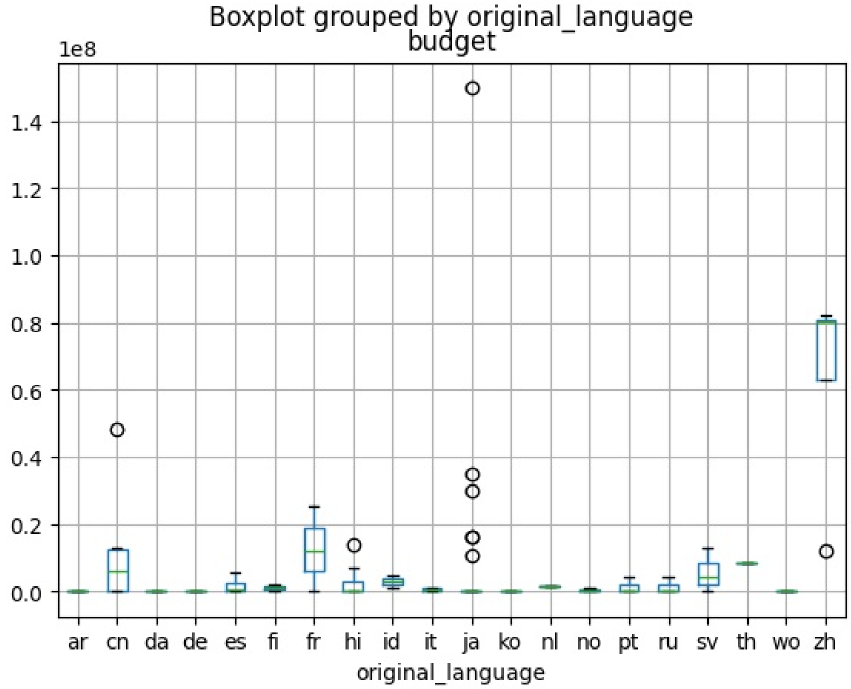
Out[28]:

```
<Axes: title={'center': 'revenue'}, xlabel='original_language'>
```



```
In [29]: df[df.original_language!='en'].boxplot(column='budget',by='original_language')
```

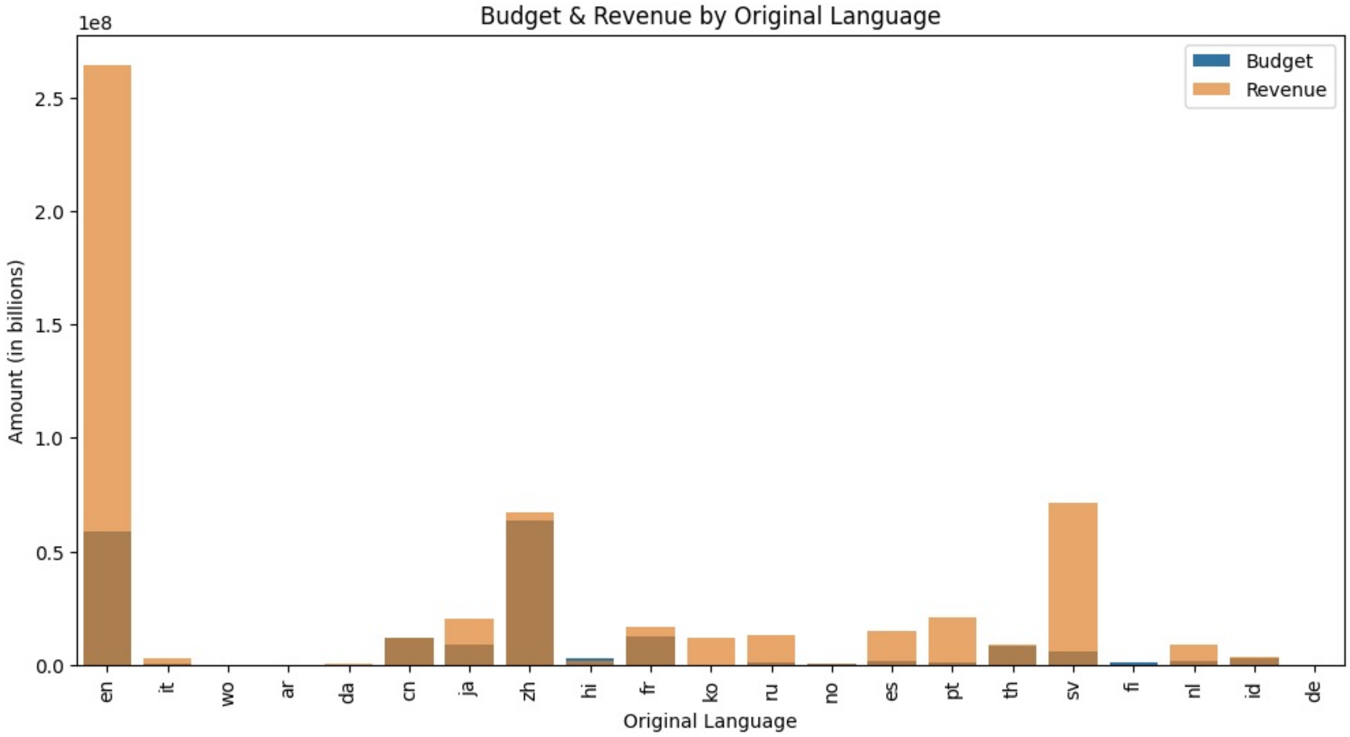
```
Out[29]: <Axes: title={'center': 'budget'}, xlabel='original_language'>
```



STEP8: Grouped Bar Plots

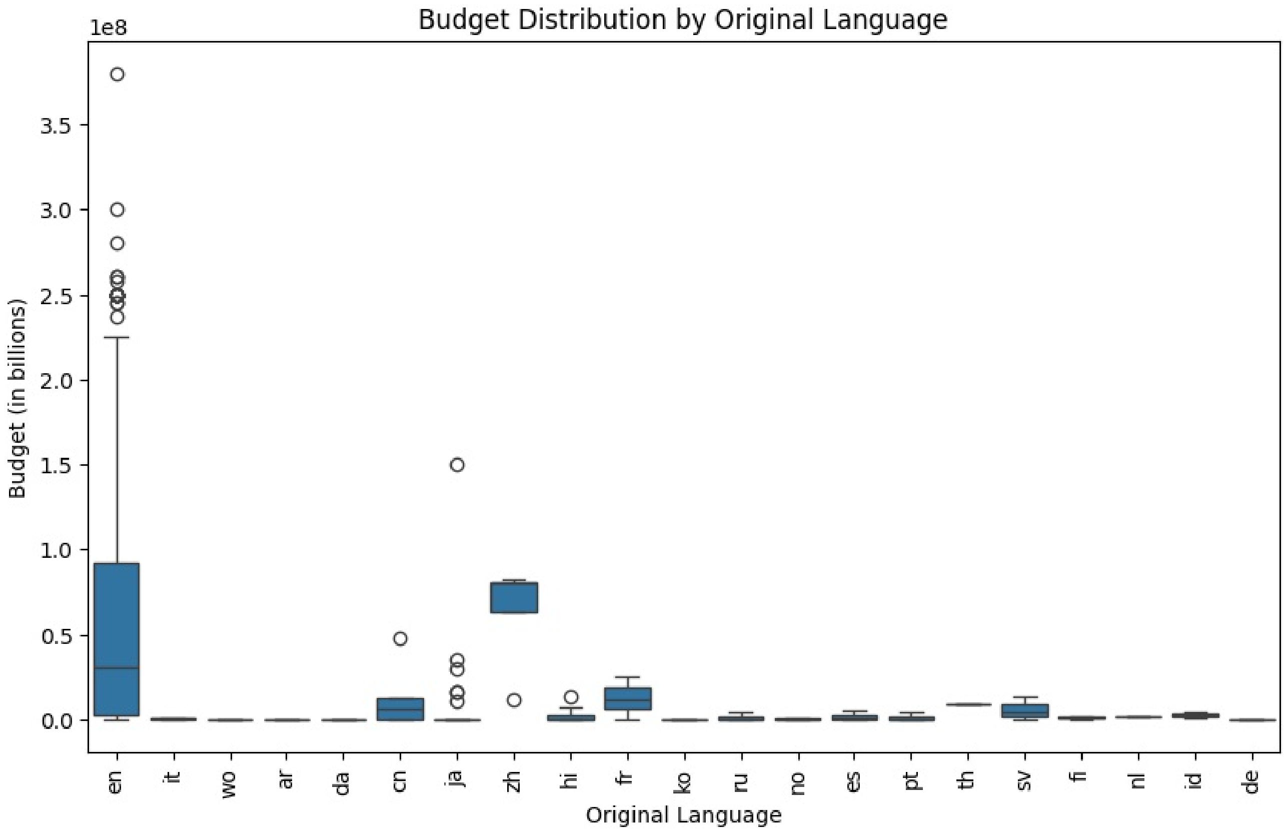
```
In [37]: plt.figure(figsize=(12, 6))
         sns.barplot(x='original_language', y='budget', data=df, ci=None, label='Budget')
         sns.barplot(x='original_language', y='revenue', data=df, ci=None, label='Revenue', alpha=0.7)
         plt.title('Budget & Revenue by Original Language')
         plt.xlabel('Original Language')
         plt.ylabel('Amount (in billions)')
         plt.xticks(rotation=90)
         plt.legend()
         plt.show()
```

```
C:\Users\kanis\AppData\Local\Temp\ipykernel_17096\2463698461.py:2: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='original_language', y='budget', data=df, ci=None, label='Budget')
C:\Users\kanis\AppData\Local\Temp\ipykernel_17096\2463698461.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='original_language', y='revenue', data=df, ci=None, label='Revenue', alpha=0.7)
```



STEP9: Box Plots

```
In [38]: plt.figure(figsize=(10, 6))
         sns.boxplot(x='original_language', y='budget', data=df)
         plt.title('Budget Distribution by Original Language')
         plt.xlabel('Original Language')
         plt.ylabel('Budget (in billions)')
         plt.xticks(rotation=90)
         plt.show()
```



In [39]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 693 entries, 1237 to 44274
Data columns (total 24 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   adult                  693 non-null    object
 1   belongs_to_collection  693 non-null    object
 2   budget                 693 non-null    int64
 3   genres                 693 non-null    object
 4   homepage               693 non-null    object
 5   id                     693 non-null    object
 6   imdb_id                693 non-null    object
 7   original_language      693 non-null    object
 8   original_title         693 non-null    object
 9   overview               693 non-null    object
 10  popularity             693 non-null    object
 11  poster_path            693 non-null    object
 12  production_companies   693 non-null    object
 13  production_countries   693 non-null    object
 14  release_date           693 non-null    datetime64[ns]
 15  revenue                693 non-null    float64
 16  runtime                693 non-null    float64
 17  spoken_languages       693 non-null    object
 18  status                 693 non-null    object
 19  tagline                693 non-null    object
 20  title                  693 non-null    object
 21  video                  693 non-null    object
 22  vote_average           693 non-null    float64
 23  vote_count             693 non-null    float64
dtypes: datetime64[ns](1), float64(4), int64(1), object(18)
memory usage: 135.4+ KB
```

In [40]:
```
df.fillna(0, inplace=True)
```

In [41]:
```
df
```

Out[41]:

	adult	belongs_to_collection	budget	genres	homepage	id	imdb_id	original
1237	False	{'id': 55427, 'name': 'Fantasia Collection', '...	2280000	[{'id': 16, 'name': 'Animation'}, {'id': 10751...	http://movies.disney.com/fantasia	756	tt0032455	
8407	False	{'id': 158365, 'name': 'Why We Fight', 'poster...	0	[{'id': 99, 'name': 'Documentary'}, {'id': 36,...	http://www.archive.org/details/PreludeToWar	23336	tt0035209	
1909	False	{'id': 87250, 'name': 'Bambi Collection', 'pos...	858000	[{'id': 16, 'name': 'Animation'}, {'id': 18, '...	http://movies.disney.com/bambi	3170	tt0034492	
993	False	{'id': 55419, 'name': 'Cinderella Collection',...	2900000	[{'id': 10751, 'name': 'Family'}, {'id': 14, '...	http://movies.disney.com/cinderella-1950	11224	tt0042332	
4197	False	{'id': 441439, 'name': 'Alienation Trilogy', '...	0	[{'id': 18, 'name': 'Drama'}]	http://www.imdb.com/title/tt0054130/	41050	tt0054130	
...	...	...	...	...	...	...	...	
33356	False	{'id': 468552, 'name': 'Wonder Woman Collectio...	149000000	[{'id': 28, 'name': 'Action'}, {'id': 12, 'nam...	http://www.warnerbros.com/wonder-woman	297762	tt0451279	
44009	False	{'id': 86066, 'name': 'Despicable Me Collectio...	80000000	[{'id': 28, 'name': 'Action'}, {'id': 16, 'nam...	http://www.despicable.me	324852	tt3469046	
43294	False	{'id': 87118, 'name': 'Cars Collection', 'post...	175000000	[{'id': 10751, 'name': 'Family'}, {'id': 35, '...	http://cars.disney.com	260514	tt3606752	
44842	False	{'id': 8650, 'name': 'Transformers Collection'...	260000000	[{'id': 28, 'name': 'Action'}, {'id': 878, 'na...	http://www.transformersmovie.com/	335988	tt3371366	
44274	False	{'id': 173710, 'name': 'Planet of the Apes (Re...	152000000	[{'id': 18, 'name': 'Drama'}, {'id': 878, 'nam...	http://www.foxmovies.com/movies/war-for-the-pl...	281338	tt3450958	

693 rows × 24 columns

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js