"INSIGHTS ILLUMINATED : DATA DRIVEN RECOMMENDATIONS FOR MICROSOFT'S MOVIE MAKING ENDEAVORS"

1. Overview Of The Bussiness Problem.

As Microsoft ventures in to movie making,they face the challenge of identifing the best genres to make in a highly competitive industry.Microsoft requires insights derived from an analysis of movie datasets.

Therefore, this analysis is to inform the managers and decision makers on the type of movies that are likely to captivate audiences.

The tools that were used for the data analysis were python,a programming language for exploratory data analysis together with Matplotlib for data manipulation.Jupyter notebook for data analysis workflows and Seaborn was used to generate insightful plots and charts to communicate findings effectively.

STEP 1: IMPORT THE NECCESARY LIBRARIES

In [2]:
```
import csv
import pandas as pd
import matplotlib as plt
import seaborn as sns
import numpy as np
```

STEP 2: IMPORTING AND JOINING THE DATASETS

In [3]:
```
df1 = pd.read_csv(r"c:\Users\Ruth\Downloads\dsc-phase-1-project-v2-4-master\dsc-phase-1-
df2 = pd.read_csv(r"c:\Users\Ruth\Downloads\dsc-phase-1-project-v2-4-master\dsc-phase-1-
```

Joining the datasets

In [4]:
```
df3 = df1.merge(df2)
```

STEP 3 : DATA UNDERSTANDING

```
.Dataframe shape
.Columns
.dtypes
.Description
.Info
```

In [5]:
```
df3.shape
```

Out[5]:
```
(2703, 14)
```

In [6]:
```
df3.columns
```

Out[6]:
```
Index(['Unnamed: 0', 'genre_ids', 'id', 'original_language', 'original_title',
       'popularity', 'release_date', 'title', 'vote_average', 'vote_count',
       'studio', 'domestic_gross', 'foreign_gross', 'year'],
      dtype='object')
```

In [7]:
```
df3.head(10)
```

Out[7]:

| Unnamed: 0 | genre_ids | id | original_language | original_title | popularity | release_date | title | vote_av |
|---|---|---|---|---|---|---|---|---|

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | [14, 12, 16, 10751] | 10191 | en | How to Train Your Dragon | 28.734 | 2010-03-26 | How to Train Your Dragon |
| **1** | 2 | [12, 28, 878] | 10138 | en | Iron Man 2 | 28.515 | 2010-05-07 | Iron Man 2 |
| **2** | 4 | [28, 878, 12] | 27205 | en | Inception | 27.920 | 2010-07-16 | Inception |
| **3** | 7 | [16, 10751, 35] | 10193 | en | Toy Story 3 | 24.445 | 2010-06-17 | Toy Story 3 |
| **4** | 8 | [16, 10751, 35] | 20352 | en | Despicable Me | 23.673 | 2010-07-09 | Despicable Me |
| **5** | 9 | [16, 28, 35, 10751, 878] | 38055 | en | Megamind | 22.855 | 2010-11-04 | Megamind |
| **6** | 12 | [53, 12, 28] | 27578 | en | The Expendables | 21.517 | 2010-08-03 | The Expendables |
| **7** | 13 | [16, 10751] | 38757 | en | Tangled | 21.511 | 2010-11-24 | Tangled |
| **8** | 15 | [12, 14, 18, 10749] | 24021 | en | The Twilight Saga: Eclipse | 20.340 | 2010-06-23 | The Twilight Saga: Eclipse |
| **9** | 16 | [28, 53, 878] | 20504 | en | The Book of Eli | 18.985 | 2010-01-11 | The Book of Eli |

In [8]:
```
df3.dtypes
```

Out[8]:
```
Unnamed: 0             int64
genre_ids             object
id                     int64
original_language     object
original_title        object
popularity           float64
release_date          object
title                 object
vote_average         float64
vote_count             int64
studio                object
domestic_gross       float64
foreign_gross         object
year                   int64
dtype: object
```

In [9]:
```
df3.tail(5)
```

Out[9]:

| | Unnamed: 0 | genre_ids | id | original_language | original_title | popularity | release_date | title | vote_ |
|---|---|---|---|---|---|---|---|---|---|
| **2698** | 25090 | [16, 10751, 12] | 455842 | en | Elliot: The Littlest Reindeer | 2.903 | 2018-11-30 | Elliot: The Littlest Reindeer | |
| **2699** | 25148 | [28, 12, 16] | 332718 | en | Bilal: A New Breed of Hero | 2.707 | 2018-02-02 | Bilal: A New Breed of Hero | |
| **2700** | 25189 | [35] | 498919 | es | La Boda de Valentina | 2.550 | 2018-02-09 | La Boda de Valentina | |
| **2701** | 25307 | [18] | 470641 | hi | □□□□□□□□□ | 2.276 | 2018-01-12 | Mukkabaaz | |
| **2702** | 26409 | [10749, 18] | 551634 | zh | 你好，之华 | 0.600 | 2018-11-09 | Last Letter | |

In [10]:
```python
df3.describe()
```

Out[10]:

| | Unnamed: 0 | id | popularity | vote_average | vote_count | domestic_gross | year |
|---|---|---|---|---|---|---|---|
| **count** | 2703.000000 | 2703.000000 | 2703.000000 | 2703.000000 | 2703.000000 | 2.682000e+03 | 2703.000000 |
| **mean** | 11686.778024 | 213291.491306 | 10.002752 | 6.418572 | 1358.194599 | 3.629150e+07 | 2014.044395 |
| **std** | 7459.175381 | 139706.978070 | 7.294182 | 0.916424 | 2408.885097 | 7.734897e+07 | 2.440458 |
| **min** | 1.000000 | 1771.000000 | 0.600000 | 0.000000 | 1.000000 | 1.000000e+02 | 2010.000000 |
| **25%** | 5289.000000 | 76493.500000 | 5.881000 | 5.900000 | 78.000000 | 2.000000e+05 | 2012.000000 |
| **50%** | 11319.000000 | 209249.000000 | 8.627000 | 6.500000 | 393.000000 | 3.800000e+06 | 2014.000000 |
| **75%** | 17675.000000 | 334521.500000 | 12.698500 | 7.000000 | 1440.000000 | 3.882500e+07 | 2016.000000 |
| **max** | 26506.000000 | 574534.000000 | 80.773000 | 10.000000 | 22186.000000 | 9.367000e+08 | 2018.000000 |

In [11]:
```python
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2703 entries, 0 to 2702
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Unnamed: 0         2703 non-null   int64
 1   genre_ids          2703 non-null   object
 2   id                 2703 non-null   int64
 3   original_language  2703 non-null   object
 4   original_title     2703 non-null   object
 5   popularity         2703 non-null   float64
 6   release_date       2703 non-null   object
 7   title              2703 non-null   object
 8   vote_average       2703 non-null   float64
 9   vote_count         2703 non-null   int64
 10  studio             2702 non-null   object
 11  domestic_gross     2682 non-null   float64
 12  foreign_gross      1723 non-null   object
 13  year               2703 non-null   int64
dtypes: float64(3), int64(4), object(7)
memory usage: 295.8+ KB
```

STEP 4 :DATA PREPARATION

```
        .checking missing values in dataset
        .dropping the missing values
        .visualizing and percentage of missing values
        .Identifying duplicated data
```

In [12]:
```python
print('Any missing value?', df3.isnull().values.any())
```

```
Any missing value? True
```

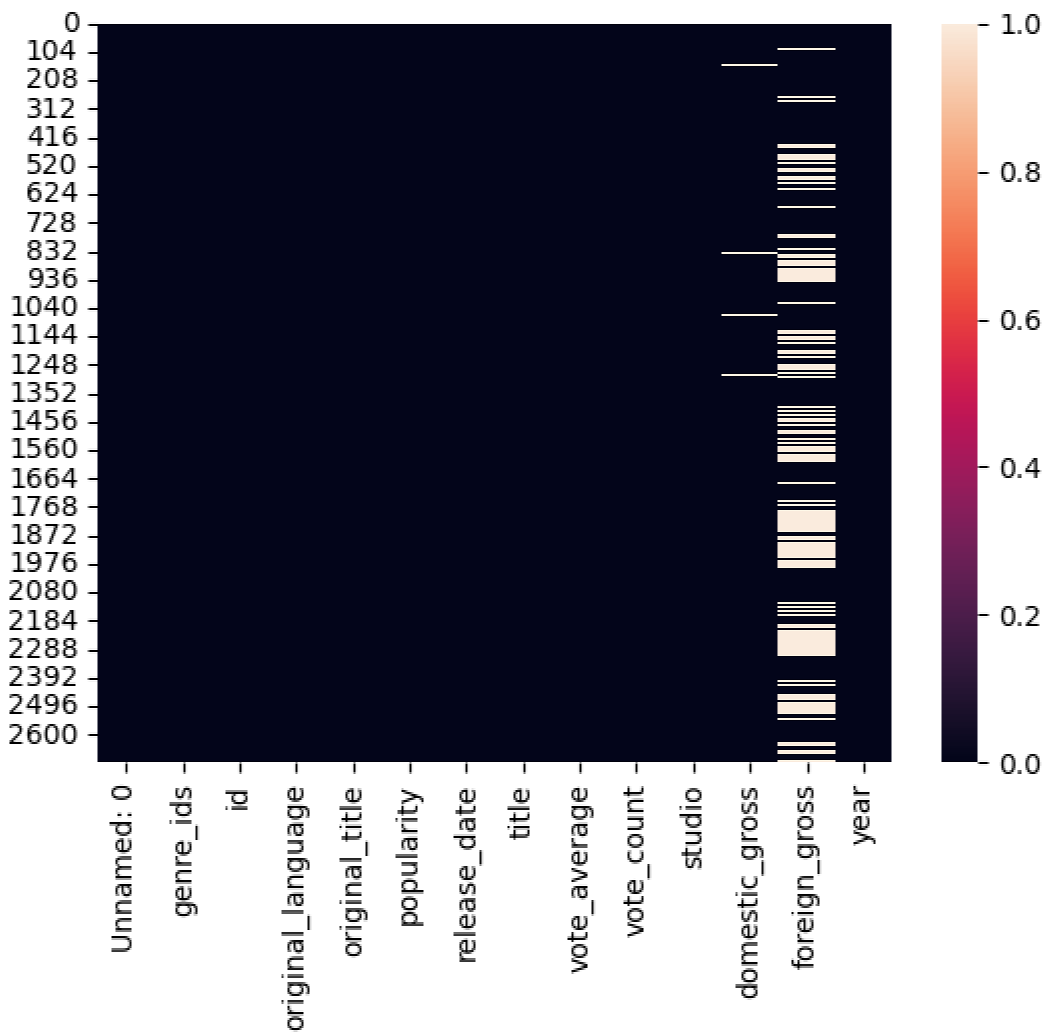
In [13]:
```python
df3.isnull().sum()
```

Out[13]:
```
Unnamed: 0            0
genre_ids             0
id                    0
original_language     0
original_title        0
popularity            0
release_date          0
title                 0
vote_average          0
vote_count            0
```

```
studio                  1
domestic_gross         21
foreign_gross         980
year                    0
dtype: int64
```

In [14]:
```python
sns.heatmap(df3.isnull())
```

Out[14]:
```
<Axes: >
```




In [15]:
```python
df_missing = df3.isnull().sum() *100/len(df3)
print(df_missing)
```

```
Unnamed: 0            0.000000
genre_ids             0.000000
id                    0.000000
original_language     0.000000
original_title        0.000000
popularity            0.000000
release_date          0.000000
title                 0.000000
vote_average          0.000000
vote_count            0.000000
studio                0.036996
domestic_gross        0.776915
foreign_gross        36.256012
year                  0.000000
dtype: float64
```

In [16]:
```python
df=df3.dropna(axis=0,)
print(df)
```

```
      Unnamed: 0                genre_ids      id original_language  \
0              1     [14, 12, 16, 10751]   10191                en   
1              2           [12, 28, 878]   10138                en   
2              4           [28, 878, 12]   27205                en   
3              7       [16, 10751, 35]   10193                en   
4              8       [16, 10751, 35]   20352                en   
...          ...                     ...     ...               ...   
2682       24335   [12, 80, 10751, 35]  425148                en   
2686       24465  [35, 10749, 18, 9648]  522921                zh   
2689       24494           [12, 35, 14]  497984                zh   
2693       24646            [18, 10749]  446132                fr   
2699       25148           [28, 12, 16]  332718                en   

                  original_title  popularity release_date  \
0        How to Train Your Dragon      28.734   2010-03-26   
1                     Iron Man 2      28.515   2010-05-07   
2                      Inception      27.920   2010-07-16   
3                    Toy Story 3      24.445   2010-06-17   
4                   Despicable Me      23.673   2010-07-09   
...                          ...         ...          ...   
2682                   Show Dogs       7.904   2018-05-18   
2686                       超时空同居       6.840   2018-05-17   
2689                        捉妖记2       6.637   2018-02-14   
2693   Gauguin: Voyage de Tahiti       5.553   2018-07-11   
2699  Bilal: A New Breed of Hero       2.707   2018-02-02   

                          title  vote_average  vote_count       studio  \
0        How to Train Your Dragon           7.7        7610         P/DW   
1                     Iron Man 2           6.8       12368         Par.   
2                      Inception           8.3       22186           WB   
3                    Toy Story 3           7.7        8340           BV   
4                   Despicable Me           7.2       10057         Uni.   
...                          ...           ...         ...          ...   
2682                   Show Dogs           5.9          92  Global Road   
2686        How Long Will I Love U           7.4          11        WGUSA   
2689              Monster Hunt 2           6.3          14          LGF   
2693   Gauguin: Voyage to Tahiti           5.6          48        Cohen   
2699  Bilal: A New Breed of Hero           6.8          54           VE   

      domestic_gross foreign_gross  year  
0        217600000.0     277300000  2010  
1        312400000.0     311500000  2010  
2        292600000.0     535700000  2010  
3        415000000.0     652000000  2010  
4        251500000.0     291600000  2010  
...              ...           ...   ...  
2682      17900000.0      21300000  2018  
2686        747000.0      82100000  2018  
2689        706000.0     361000000  2018  
2693        200000.0       3100000  2018  
2699        491000.0       1700000  2018  

[1701 rows x 14 columns]
```

In [17]:
```
df.isna().sum()
```

Out[17]:
```
Unnamed: 0           0
genre_ids            0
id                   0
original_language    0
original_title       0
popularity           0
release_date         0
title                0
vote_average         0
vote_count           0
```

```
studio               0
domestic_gross       0
foreign_gross        0
year                 0
dtype: int64
```

In [18]:
```
df.duplicated()
```

Out[18]:
```
0       False
1       False
2       False
3       False
4       False
        ...
2682    False
2686    False
2689    False
2693    False
2699    False
Length: 1701, dtype: bool
```

In [19]:
```
df.isna().sum()
```

Out[19]:
```
Unnamed: 0           0
genre_ids            0
id                   0
original_language    0
original_title       0
popularity           0
release_date         0
title                0
vote_average         0
vote_count           0
studio               0
domestic_gross       0
foreign_gross        0
year                 0
dtype: int64
```

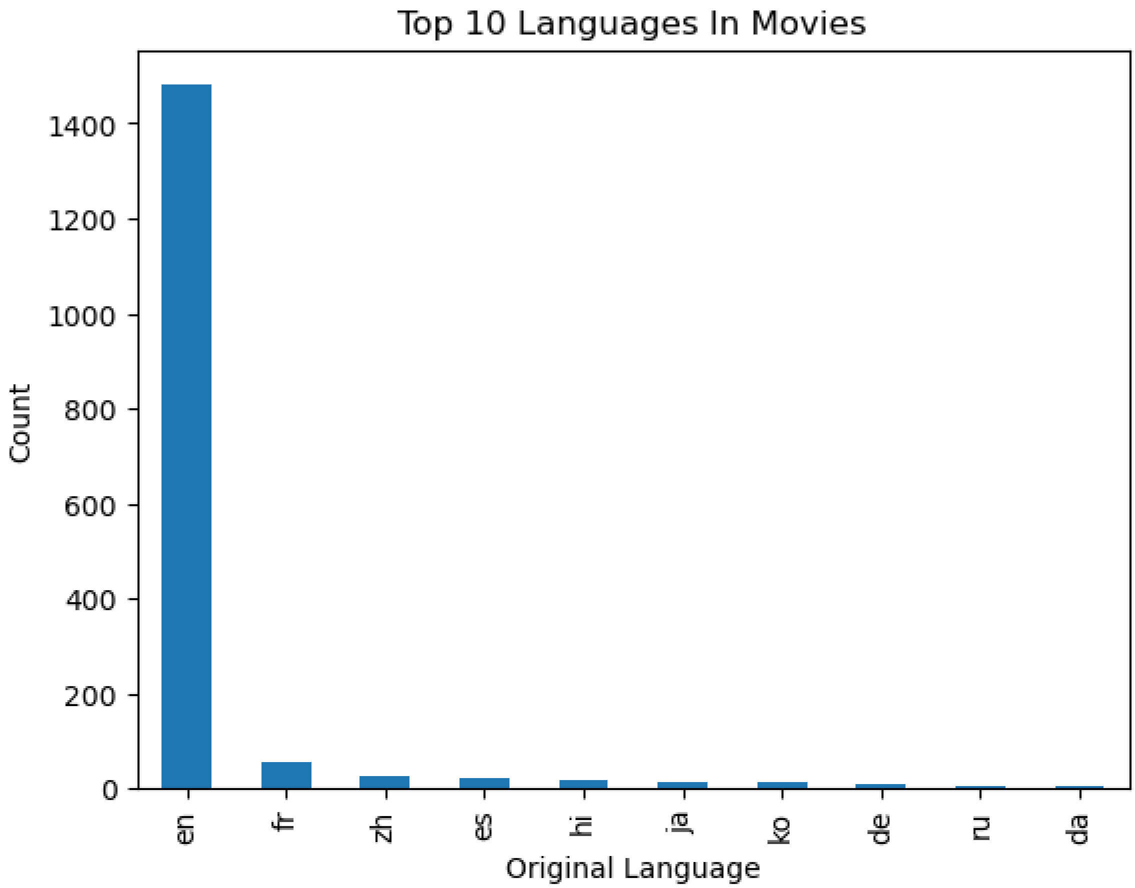
STEP 5:FEATURE UNDERSTANDING (Univariate Analysis)

```
        .plotting feature distributions.
         . Top 10 Languages using Bar Plot
         . Top Vote Count  Using Histogram
```

Language analysis

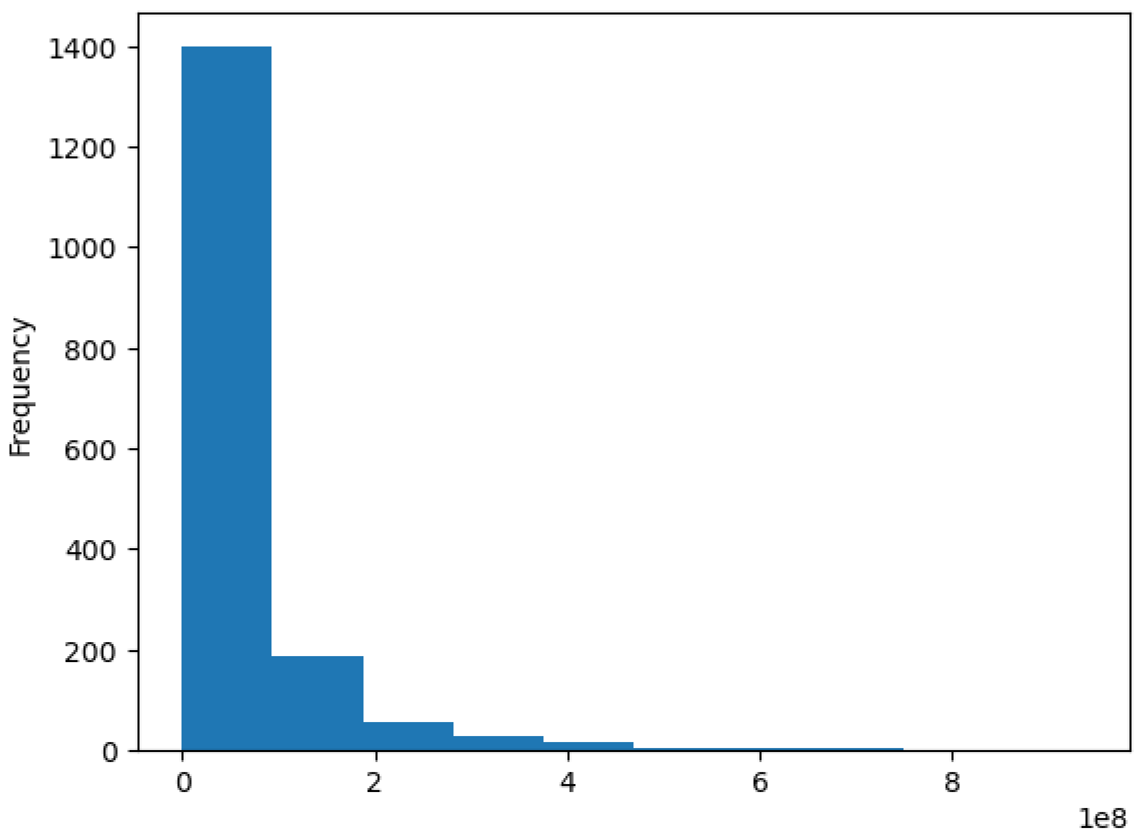
In [20]:
```
ax=df['original_language'].value_counts().head(10).plot(kind="bar",title="Top 10 Languag
ax.set_xlabel('Original Language')
ax.set_ylabel('Count')
```

Out[20]:
```
Text(0, 0.5, 'Count')
```




In [21]:
```
df["domestic_gross"].plot(kind='hist')
```

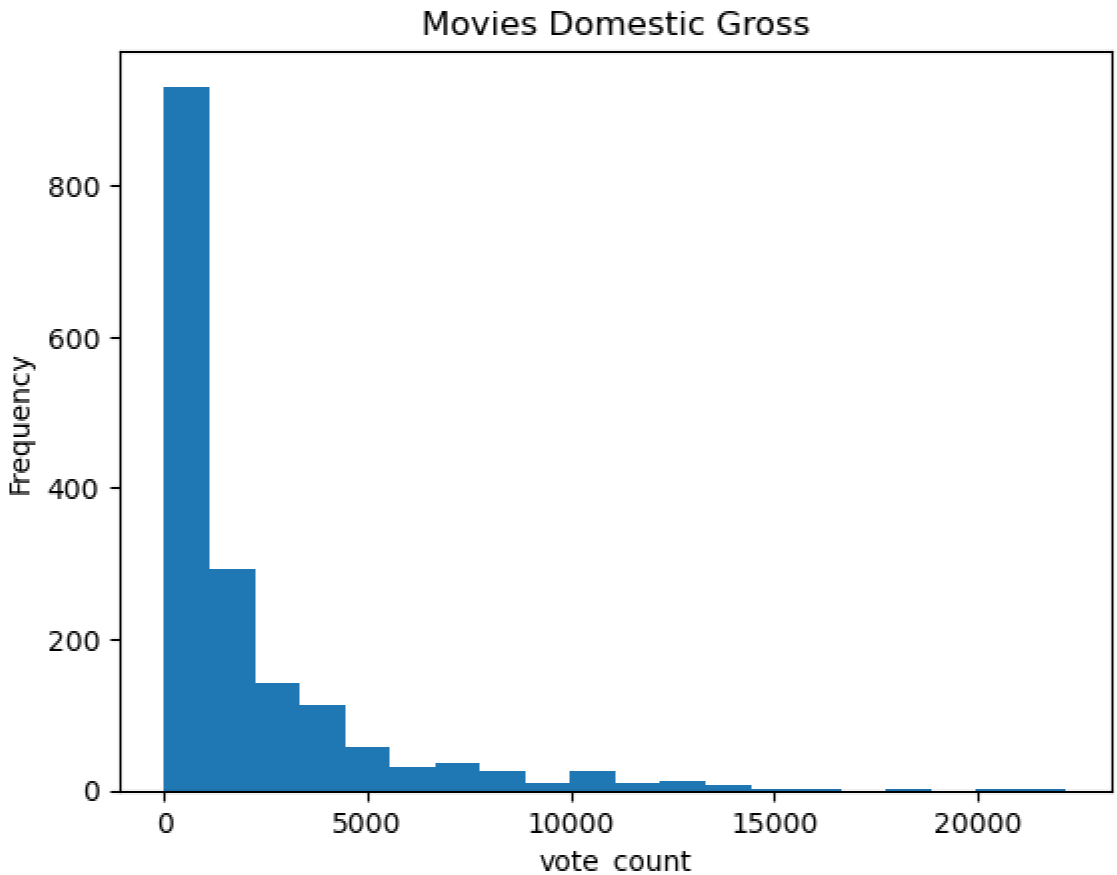
Out[21]:
```
<Axes: ylabel='Frequency'>
```




From the above distribution of the top languages used, English has dominated the global box office.By producing movies in english, Microsoft can align themselves with established industry norms.

In [22]:
```
ax = df["vote_count"].plot(kind = 'hist',bins = 20,title = 'Movies Domestic Gross')
ax.set_xlabel('vote_count')
```

Out[22]:
```
Text(0.5, 0, 'vote_count')
```




Gross revenue is an indicator of a movie's financial success.Here, we can identify patterns and trends in revenue generation helping to predict the viability of their movie projects. This information can guide to assess the performance of their movies compared to competitors.

STEP 5 :FEATURE RELATIONSHIPS

(1) Top 10 Highest Gross Movie Titles

(2) Top 10 Movies With Highest Vote Count

(3) Comparing popularity vs domestic gross

(1) Top 10 Highest Gross Movie Titles

In [23]:
```
df.nlargest(10,"domestic_gross")['title']
```

Out[23]:
```
1624    Star Wars: The Force Awakens
1625    Star Wars: The Force Awakens
608                    Black Panther
609                    Black Panther
2550          Avengers: Infinity War
2319         Star Wars: The Last Jedi
1639                  Jurassic World
2320         Star Wars: The Last Jedi
2559                   Incredibles 2
2009    Rogue One: A Star Wars Story
Name: title, dtype: object
```

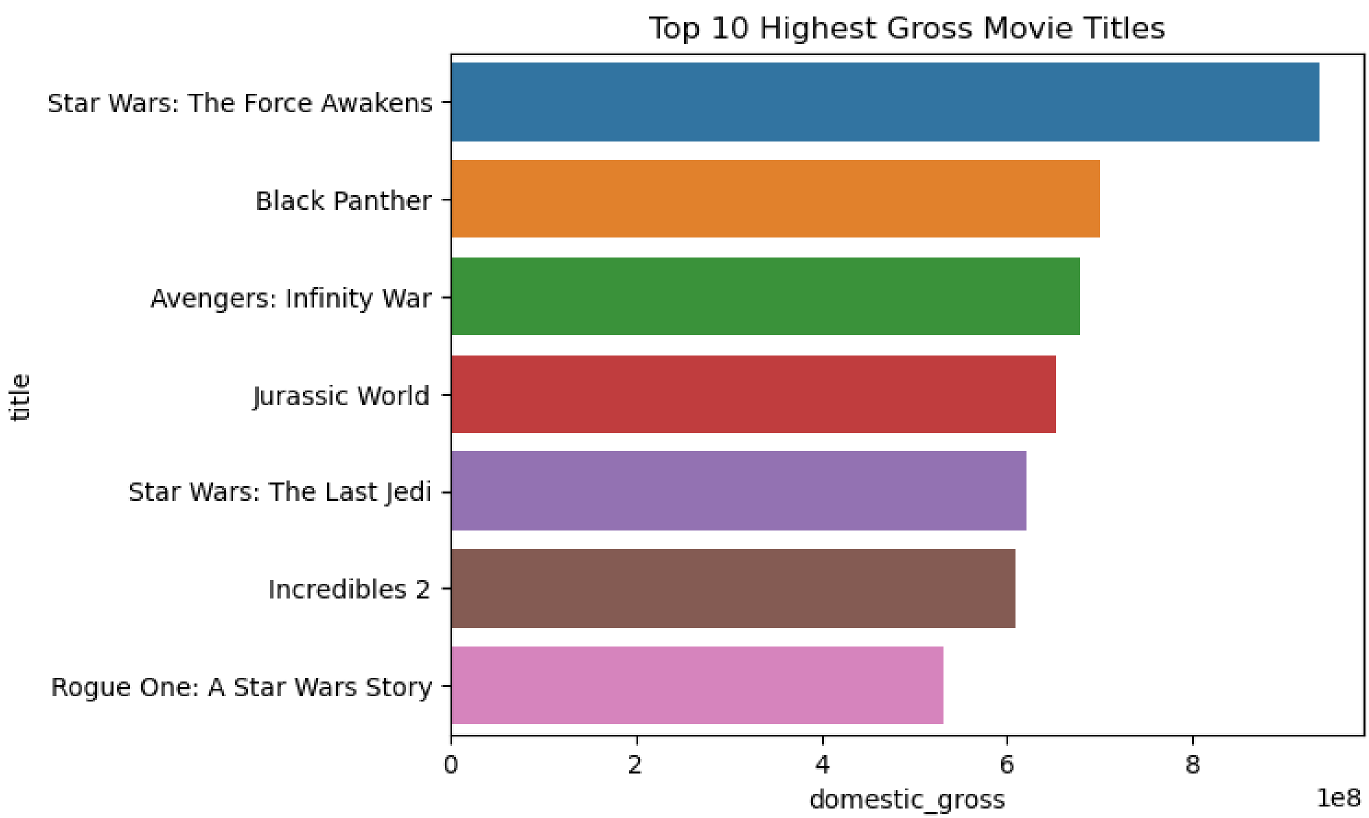
In [24]:
```
top_ten = df.nlargest(10,"domestic_gross")[['title',"domestic_gross"]].set_index('title'
```

```
print(top_ten)
```

```
                              domestic_gross
title
Star Wars: The Force Awakens     936700000.0
Star Wars: The Force Awakens     936700000.0
Black Panther                    700100000.0
Black Panther                    700100000.0
Avengers: Infinity War           678800000.0
Jurassic World                   652300000.0
Star Wars: The Last Jedi         620200000.0
Star Wars: The Last Jedi         620200000.0
Incredibles 2                    608600000.0
Rogue One: A Star Wars Story     532200000.0
```

In [27]:
```
ax = sns.barplot(x="domestic_gross",y=top_ten.index,data=top_ten)
ax.set_title('Top 10 Highest Gross Movie Titles')
```

Out[27]:
```
Text(0.5, 1.0, 'Top 10 Highest Gross Movie Titles')
```



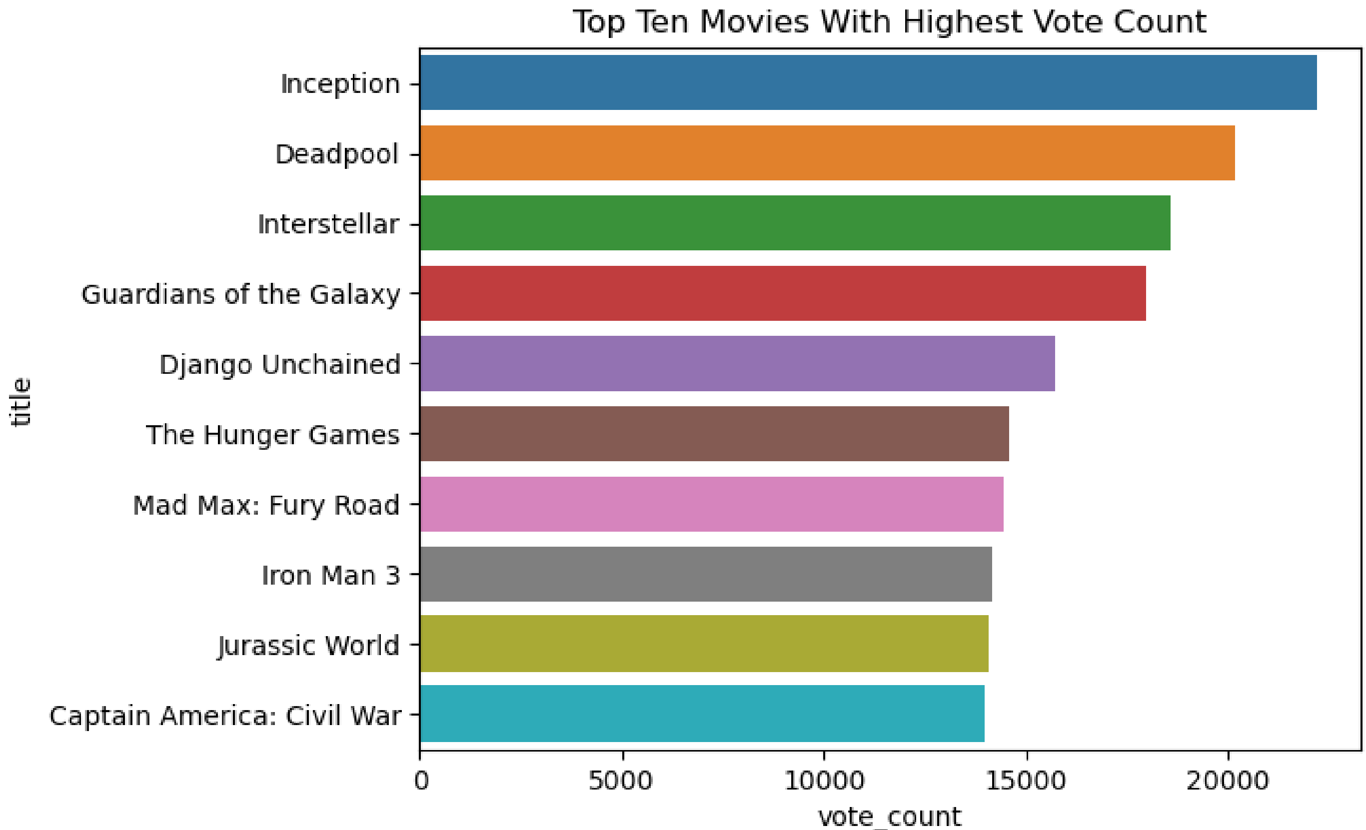

1. I recommend that Microsoft should focus on the genres and themes that have performed well based on the above dataset which was the movies with the highest revenue.Essentially, action,adventure and fantasy have consistently performed well.This will help to appeal to a wider demographic

(2)Movies With Highest Votes Count

In [ ]:
```
top10 = df.nlargest(10,"vote_count")[['title',"vote_count"]].set_index('title')
```

In [ ]:
```
ax = sns.barplot(x="vote_count",y=top10.index,data=top10)
ax.set_title('Top Ten Movies With Highest Vote Count')
```

Out[ ]:
```
Text(0.5, 1.0, 'Top Ten Movies With Highest Vote Count')
```

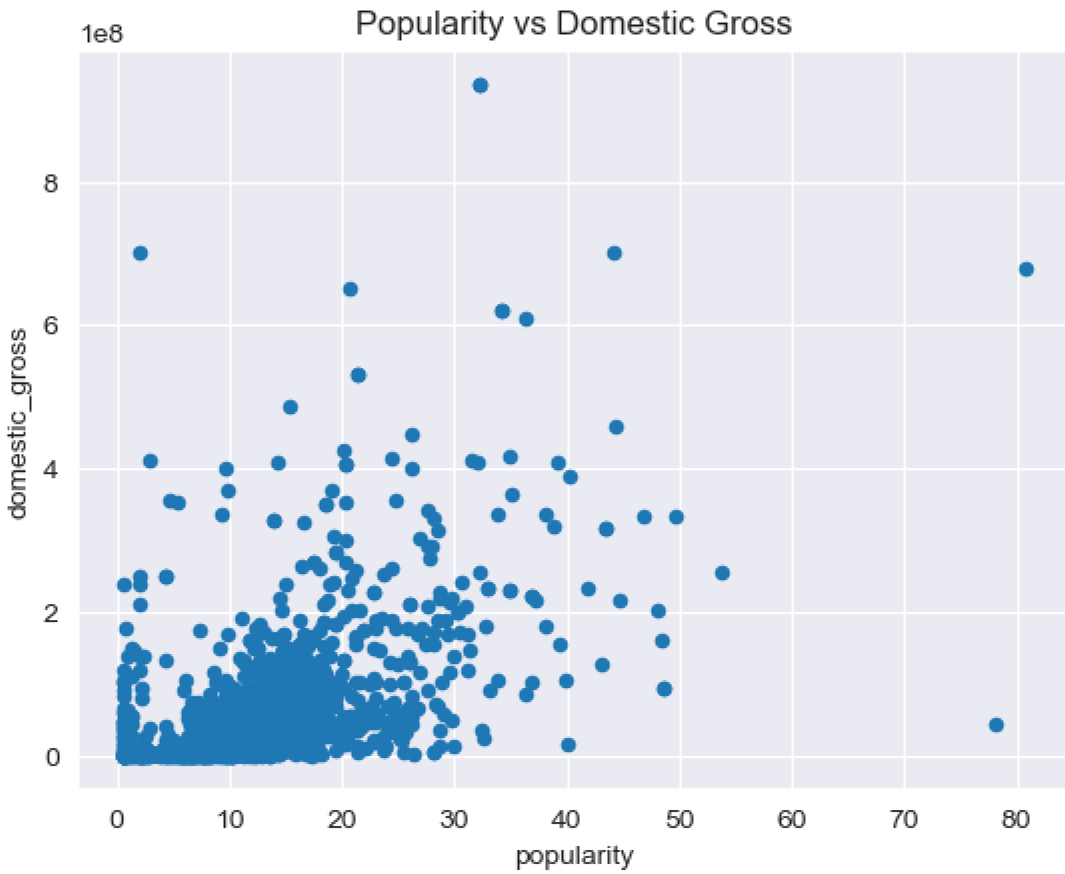



1. I recommend that In order to resonate with the audience, Microsoft should deliver movies that are visually stunning,intellectualy stimulating and emotionally resonant. Invest in top tier talent,production resources that earn votes

(3) Comparing popularity vs domestic gross

In [ ]:
```
df.plot(kind = 'scatter', x= 'popularity', y = "domestic_gross",title = "Popularity vs D
```

Out[ ]:
```
<Axes: title={'center': 'Popularity vs Domestic Gross'}, xlabel='popularity', ylabel='do
mestic_gross'>
```




1. Microsoft should prioritize projects that demonstrate potential for both popularity and revenue . From the above scatterplot,high popular movies may attract audiences but may not always translate to substantial box office gross.

In a nutshell, based on the analysis of the movie dataset,i recommend that Microsoft prioritizes genres and themes that have demonstrated the highest revenue,popularity and vote count. By understanding the audience Microsoft can tailor the movies to engage the target audience

I would like to extend my sincere gratitude to Microsoft for the opportunity to work on this project. It has been a great experience to delve in to the movie datasets and provide insights to support Microsoft's venture in to movie making. Should you have any further questions,please reach out.Thank you once again for the opportunity to contribute to this exciting project.