Highest Grossing Indian Movies Data Analysis

About this dataset:

This dataset provides a comprehensive look into the financial performance of the highest-grossing Indian films from 2000 to 2023. It aims to highlight the economic aspects of Indian cinema, including production budgets, worldwide gross revenues in both INR and USD, and specific gross revenues within India.This allows for an in-depth analysis of trends and patterns in the financial success of Indian films. There are a total of 105 uniqye movies/rows.

Context:

Indian cinema is one of the largest film industries in the world, producing over 2,000 films annually. While Bollywood (Hindi cinema) is perhaps the most globally recognized, the industry also includes other major regional cinemas like Tollywood (Telugu and Bengali cinema), Kollywood (Tamil cinema), and more.

Despite the industry's extensive output, there's a scarcity of datasets offering a comprehensive financial breakdown of Indian films. This dataset was created to address this gap and provide valuable insights for film analysts, researchers, and enthusiasts.

Inspiration:

The dataset was inspired by the desire to understand the economics of Indian cinema better and the factors contributing to a film's financial success. It encourages exploratory data analysis to unveil patterns and trends within the Indian film industry.

Potential research questions this dataset could answer:

-How has the profitability of Indian films changed over the years? -Are films in certain languages more financially successful than others? -Which directors or studios have the highest-grossing films?

In [1]:

```
#Importing Required Libraries
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import re
```

We will now use the Movies dataset and read it

In [4]:

```
#Reading Dataset
data_set = pd.read_csv("C://Users//Administrator//Downloads//archive (5).zip")
data_set
```

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

Out[4]:

	Film	Year	Director	Studio(s)	Budget (est.)	World wide gross (INR)	World wide gross (USD)	Gross in India (INR crore)	Primary Language
0	Dangal	2016	Nitesh Tiwari	Aamir Khan Productions\nUTV Motion Pictures\nW...	₹70 crore	₹2,024 crore	317.00	538.03	Hindi
1	Baahubali 2: The Conclusion	2017	S. S. Rajamouli	Arka Media Works	₹250 crore	₹1,810.60 crore	217.27	1416.9	Telugu\nTamil
2	RRR *	2022	S. S. Rajamouli	DVV Entertainments	₹550 crore	₹1,316 crore	157.92	944	Telugu
3	K.G.F: Chapter 2	2022	Prashanth Neel	Hombale Films	₹100 crore	₹1,225	147.00	1,008	Kannada
4	Pathaan	2023	Siddharth Anand	Yash Raj Films	₹250 crore	₹1,050.3 crore	130.00	654.28	Hindi
...	...	...	...	...	...	...	...	...	...
100	Race 2	2013	Abbas–Mustan	UTV Motion Pictures	NaN	₹173.36	20.80	139.51	Hindi
101	Bala	2019	Amar Kaushik	AA films	NaN	₹171.49	20.58	139.06	Hindi
102	Bhaag Milkha Bhaag	2013	Rakeysh Omprakash Mehra	Viacom 18 Motion Pictures	₹41 crore	₹169.96	20.40	151.29	Hindi
103	Ek Villain	2014	Mohit Suri	AA films	₹39 crore	₹169.62	20.35	146.69	Hindi
104	Golmaal 3	2010	Rohit Shetty	Eros international	₹40 crore	₹169.09	20.29	147.69	Hindi

105 rows × 9 columns

Exploring the Data

In [5]:

```
#Displaying the first 10 Rows about Data
data_set.head(10)
```

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

Out[5]:

	Film	Year	Director	Studio(s)	Budget (est.)	World wide gross (INR)	World wide gross (USD)	Gross in India (INR crore)	Primary Language
0	Dangal	2016	Nitesh Tiwari	Aamir Khan Productions\nUTV Motion Pictures\nW...	₹70 crore	₹2,024 crore	317.00	538.03	Hindi
1	Baahubali 2: The Conclusion	2017	S. S. Rajamouli	Arka Media Works	₹250 crore	₹1,810.60 crore	217.27	1416.9	Telugu\nTamil
2	RRR *	2022	S. S. Rajamouli	DVV Entertainments	₹550 crore	₹1,316 crore	157.92	944	Telugu
3	K.G.F: Chapter 2	2022	Prashanth Neel	Hombale Films	₹100 crore	₹1,225	147.00	1,008	Kannada
4	Pathaan	2023	Siddharth Anand	Yash Raj Films	₹250 crore	₹1,050.3 crore	130.00	654.28	Hindi
5	Bajrangi Bhaijaan	2015	Kabir Khan	Salman Khan Films\nEros International	₹ 10 crore	₹969 crore	151.05	444.92	Hindi
6	Secret Superstar	2017	Advait Chandan	Aamir Khan Productions	₹15 crore	₹966.86 crore	154.00	81.28	Hindi
7	PK	2014	Rajkumar Hirani	Vinod Chopra Films\nRajkumar Hirani Films	₹122 crore	₹769.89 crore	126.15	473.33	Hindi
8	Sultan	2016	Ali Abbas Zafar	Yash Raj Films	₹90 crore	₹623.33 crore	75.70	417.29	Hindi
9	2 . 0	2018	S. Shankar	Lyca Productions	₹400 crore–₹600 crore	₹620 crore	75.30	243.01	Tamil

In [6]:

```
#displaying top 5 rows from the data
data_set.head(5)
```

Out[6]:

	Film	Year	Director	Studio(s)	Budget (est.)	World wide gross (INR)	World wide gross (USD)	Gross in India (INR crore)	Primary Language
0	Dangal	2016	Nitesh Tiwari	Aamir Khan Productions\nUTV Motion Pictures\nW...	₹70 crore	₹2,024 crore	317.00	538.03	Hindi
1	Baahubali 2: The Conclusion	2017	S. S. Rajamouli	Arka Media Works	₹250 crore	₹1,810.60 crore	217.27	1416.9	Telugu\nTamil
2	RRR *	2022	S. S. Rajamouli	DVV Entertainments	₹550 crore	₹1,316 crore	157.92	944	Telugu
3	K.G.F: Chapter 2	2022	Prashanth Neel	Hombale Films	₹100 crore	₹1,225	147.00	1,008	Kannada
4	Pathaan	2023	Siddharth Anand	Yash Raj Films	₹250 crore	₹1,050.3 crore	130.00	654.28	Hindi

In [7]:

```
data_set.shape    #it will show no.of columns and rows present in dataset.
```

Out[7]:

```
(105, 9)
```

In [8]:

```
#Information about the Data set
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 105 entries, 0 to 104
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Film                      105 non-null    object 
 1   Year                      105 non-null    int64  
 2   Director                  105 non-null    object 
 3   Studio(s)                 105 non-null    object 
 4   Budget (est.)             56 non-null     object 
 5   World wide gross (INR)    105 non-null    object 
 6   World wide gross (USD)    105 non-null    float64
 7   Gross in India (INR crore)  103 non-null    object 
 8   Primary Language          103 non-null    object 
dtypes: float64(1), int64(1), object(7)
memory usage: 7.5+ KB
```

Cleaning the data

Now we will start cleaning the data

Removing or filling null values

Removing duplicate records/rows

Correcting data in wrong format or cells having wrong data

In [9]:

```
#lets analyze the columns present in  dataset
data_set.columns
```

Out[9]:

```
Index(['Film', 'Year', 'Director', 'Studio(s)', 'Budget (est.)',
       'World wide gross (INR)', 'World wide gross (USD)',
       'Gross in India (INR crore)', 'Primary Language'],
      dtype='object')
```

In [10]:

```
#From the column names it looks like the columns having spaces in between words
#removing spaces from dataset column names
data_set.columns = data_set.columns.str.replace(' ', '') # used string replace function
data_set.columns
```

Out[10]:

```
Index(['Film', 'Year', 'Director', 'Studio(s)', 'Budget(est.)',
       'Worldwidegross(INR)', 'Worldwidegross(USD)', 'GrossinIndia(INRcrore)',
       'PrimaryLanguage'],
      dtype='object')
```

In [11]:

```
data_set.head(2) # pring data, just for how it looks
```

Out[11]:

	Film	Year	Director	Studio(s)	Budget(est.)	Worldwidegross(INR)	Worldwidegross(USD)	Gross
0	Dangal	2016	Nitesh Tiwari	Aamir Khan Productions\nUTV Motion Pictures\nW...	₹70 crore	₹2,024 crore	317.00	
1	Baahubali 2: The Conclusion	2017	S. S. Rajamouli	Arka Media Works	₹250 crore	₹1,810.60 crore	217.27	

In [12]:

```
#Deriving new column for filling the Budget column from the existing column
data_set['BudgetInNum'] = data_set["Budget(est.)"].str.extract('(\d+)')
data_set.info()
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 105 entries, 0 to 104
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Film                    105 non-null    object 
 1   Year                    105 non-null    int64  
 2   Director                105 non-null    object 
 3   Studio(s)               105 non-null    object 
 4   Budget(est.)            56 non-null     object 
 5   Worldwidegross(INR)     105 non-null    object 
 6   Worldwidegross(USD)     105 non-null    float64
 7   GrossinIndia(INRcrore)  103 non-null    object 
 8   PrimaryLanguage         103 non-null    object 
 9   BudgetInNum             56 non-null     object 
dtypes: float64(1), int64(1), object(8)
memory usage: 8.3+ KB
```

In [13]:

```
#casting the datatype
data_set = data_set.astype({'BudgetInNum':'float'})
print(data_set.dtypes)
```

```
Film                       object
Year                        int64
Director                   object
Studio(s)                  object
Budget(est.)               object
Worldwidegross(INR)        object
Worldwidegross(USD)       float64
GrossinIndia(INRcrore)     object
PrimaryLanguage            object
BudgetInNum               float64
dtype: object
```

Filling Null or missing values

In data analysis, filling null values with the mean, median, or mode is a common technique for handling missing data. This approach helps maintain the overall distribution and relationships within the dataset, thereby reducing the impact of missing values on the analysis.

In [15]:

```
#Filling the Null or missing values with Mean, Median Or Mode
x = round(data_set["BudgetInNum"].mean())
data_set["BudgetInNum"].fillna(x, inplace = True)
data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 105 entries, 0 to 104
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Film                    105 non-null    object 
 1   Year                    105 non-null    int64  
 2   Director                105 non-null    object 
 3   Studio(s)               105 non-null    object 
 4   Budget(est.)            56 non-null     object 
 5   Worldwidegross(INR)     105 non-null    object 
 6   Worldwidegross(USD)     105 non-null    float64
 7   GrossinIndia(INRcrore)  103 non-null    object 
 8   PrimaryLanguage         103 non-null    object 
 9   BudgetInNum             105 non-null    float64
dtypes: float64(2), int64(1), object(7)
memory usage: 8.3+ KB
```

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

In [16]:
```
isNul = data_set['Budget(est.)'].isna()
indx = data_set[isNul].index
indx
```

Out[16]:
```
Int64Index([ 47,  48,  49,  50,  51,  52,  53,  54,  55,  56,  57,  58,  60,
             61,  62,  63,  64,  65,  66,  67,  69,  70,  71,  72,  73,  74,
             75,  76,  78,  79,  80,  81,  82,  83,  84,  86,  87,  89,  90,
             92,  93,  94,  95,  96,  97,  98,  99, 100, 101],
           dtype='int64')
```

In [17]:
```
for i in data_set['Budget(est.)'].index:
    if i in indx:
        data_set['Budget(est.)'].fillna(value=('$'+str.replace(str(data_set['BudgetI
data_set
```

Out[17]:

	Film	Year	Director	Studio(s)	Budget(est.)	Worldwidegross(INR)	Worldwidegross(USD)	G
0	Dangal	2016	Nitesh Tiwari	Aamir Khan Productions\nUTV Motion Pictures\nW...	₹70 crore	₹2,024 crore	317.00	
1	Baahubali 2: The Conclusion	2017	S. S. Rajamouli	Arka Media Works	₹250 crore	₹1,810.60 crore	217.27	
2	RRR *	2022	S. S. Rajamouli	DVV Entertainments	₹550 crore	₹1,316 crore	157.92	
3	K.G.F: Chapter 2	2022	Prashanth Neel	Hombale Films	₹100 crore	₹1,225	147.00	
4	Pathaan	2023	Siddharth Anand	Yash Raj Films	₹250 crore	₹1,050.3 crore	130.00	
...	...	...	...	...	...	...	...	
100	Race 2	2013	Abbas–Mustan	UTV Motion Pictures	$130 crore	₹173.36	20.80	
101	Bala	2019	Amar Kaushik	AA films	$130 crore	₹171.49	20.58	
102	Bhaag Milkha Bhaag	2013	Rakeysh Omprakash Mehra	Viacom 18 Motion Pictures	₹41 crore	₹169.96	20.40	
103	Ek Villain	2014	Mohit Suri	AA films	₹39 crore	₹169.62	20.35	
104	Golmaal 3	2010	Rohit Shetty	Eros international	₹40 crore	₹169.09	20.29	

105 rows × 10 columns

In [19]:
```
data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 105 entries, 0 to 104
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Film                    105 non-null    object 
 1   Year                    105 non-null    int64  
 2   Director                105 non-null    object 
 3   Studio(s)               105 non-null    object 
 4   Budget(est.)            105 non-null    object 
 5   Worldwidegross(INR)     105 non-null    object 
 6   Worldwidegross(USD)     105 non-null    float64
 7   GrossinIndia(INRcrore)  103 non-null    object 
 8   PrimaryLanguage         103 non-null    object 
 9   BudgetInNum             105 non-null    float64
dtypes: float64(2), int64(1), object(7)
memory usage: 8.3+ KB
```

In [20]:

```python
#data_set.astype({'GrossinIndia(INRcrore)':'string'}).inplace = True
data_set['GrossinIndia(INRcrore)'] = data_set['GrossinIndia(INRcrore)'].str.replace(',',
```

In [21]:

```python
#calculating the avg (mean) from the data and filling in the missing values here used lo
x = 0
c = 0
for i in data_set['GrossinIndia(INRcrore)']:
    if str(i) != 'nan':
        x = x + float(i)
    c = c+1
avg = round(x/c,2)
```

In [22]:

```python
data_set['GrossinIndia(INRcrore)'].fillna(value = avg,inplace = True)
```

In [23]:

```python
data_set['PrimaryLanguage'].unique()
```

Out[23]:

```
array(['Hindi', 'Telugu\nTamil', 'Telugu', 'Kannada', 'Tamil',
       'Telugu Hindi', nan], dtype=object)
```

In [24]:

```python
data_set['PrimaryLanguage'].fillna(value = data_set['PrimaryLanguage'].value_counts().in
data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 105 entries, 0 to 104
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Film                    105 non-null    object 
 1   Year                    105 non-null    int64  
 2   Director                105 non-null    object 
 3   Studio(s)               105 non-null    object 
 4   Budget(est.)            105 non-null    object 
 5   Worldwidegross(INR)     105 non-null    object 
 6   Worldwidegross(USD)     105 non-null    float64
 7   GrossinIndia(INRcrore)  105 non-null    object 
 8   PrimaryLanguage         105 non-null    object 
 9   BudgetInNum             105 non-null    float64
dtypes: float64(2), int64(1), object(7)
memory usage: 8.3+ KB
```

In [25]:

```python
#removing wrong data
for x in data_set.index:
    if data_set.loc[x, "Year"] > 2023:
        data_set = data_set.drop(x)
#'''Here used 2023 because the no movies should have the relesed date as future date.
```

```
        # if it is planned for relsease next year its okk but in collections it is telli
        # its released and collection the spent budget , Assuing this as wrong entry and
```

In [26]:

```
data_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 104 entries, 0 to 104
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Film                    104 non-null    object 
 1   Year                    104 non-null    int64  
 2   Director                104 non-null    object 
 3   Studio(s)               104 non-null    object 
 4   Budget(est.)            104 non-null    object 
 5   Worldwidegross(INR)     104 non-null    object 
 6   Worldwidegross(USD)     104 non-null    float64
 7   GrossinIndia(INRcrore)  104 non-null    object 
 8   PrimaryLanguage         104 non-null    object 
 9   BudgetInNum             104 non-null    float64
dtypes: float64(2), int64(1), object(7)
memory usage: 13.0+ KB
```

In [27]:

```
#Removing Duplicated Data
data_set.drop_duplicates(inplace = True)
```

Now our data set is clean

Clean data in data analysis refers to data that is accurate, complete, consistent, and free from errors or anomalies. It’s crucial for reliable analysis and decision-making. proceeding with futher steps

Data visualization

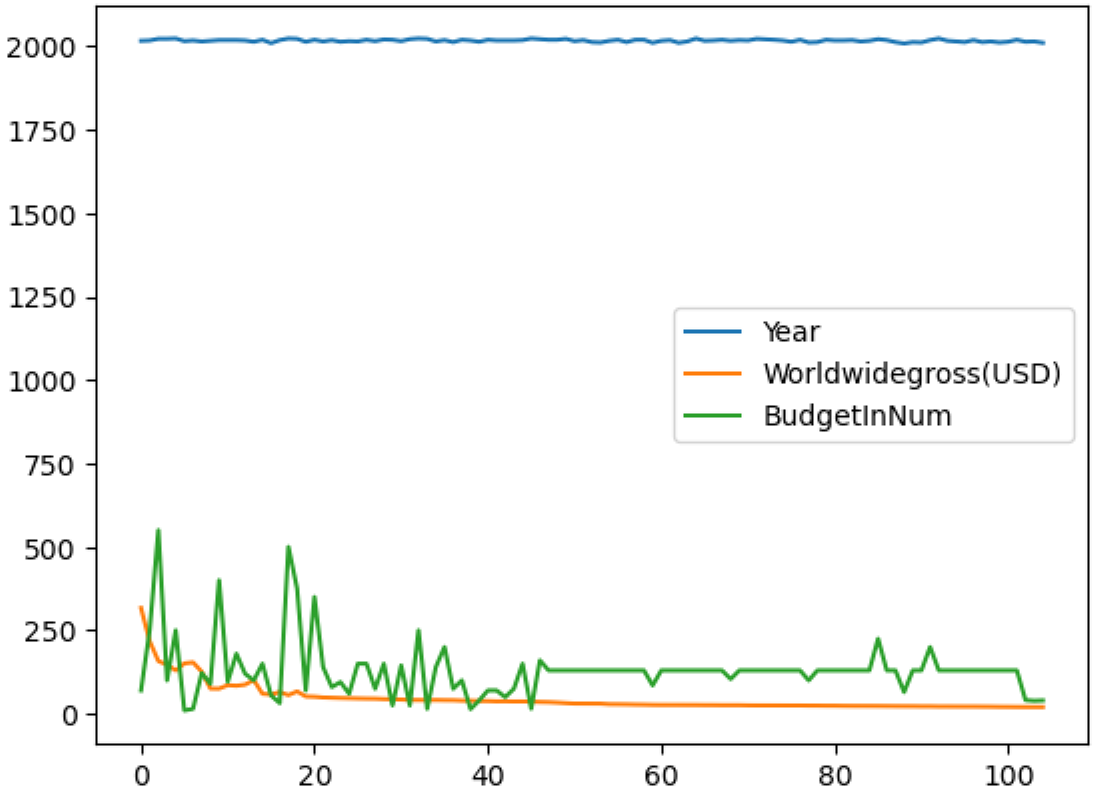
Data visualization is the use of graphical elements to represent data, making complex data more understandable, accessible, and usable.

In [28]:

```
#simple plot #  no visual info from it is clear
data_set.plot()
```

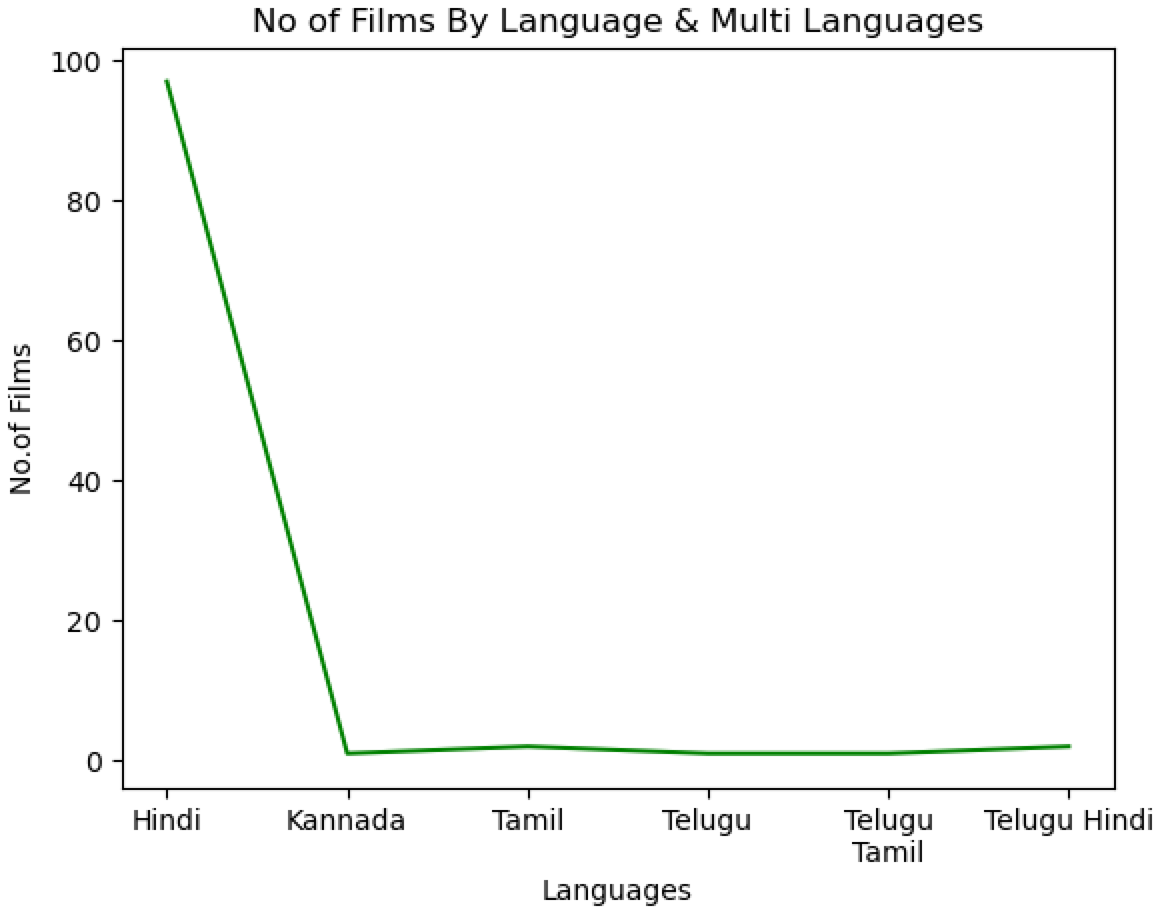
Out[28]:

```
<Axes: >
```

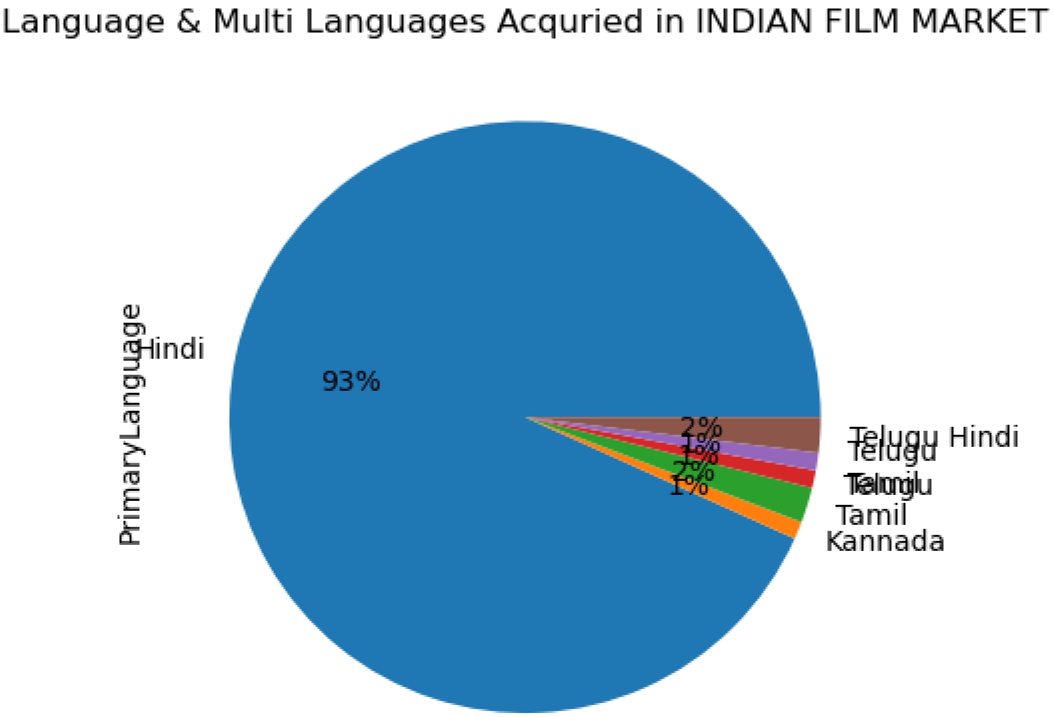


In [29]:
```
df1 = data_set.groupby(['PrimaryLanguage'])['PrimaryLanguage'].count()
df1.plot(xlabel = 'Languages', ylabel = 'No.of Films',title = "No of Films By Language &
```

Out[29]:
```
<Axes: title={'center': 'No of Films By Language & Multi Languages'}, xlabel='Language
s', ylabel='No.of Films'>
```



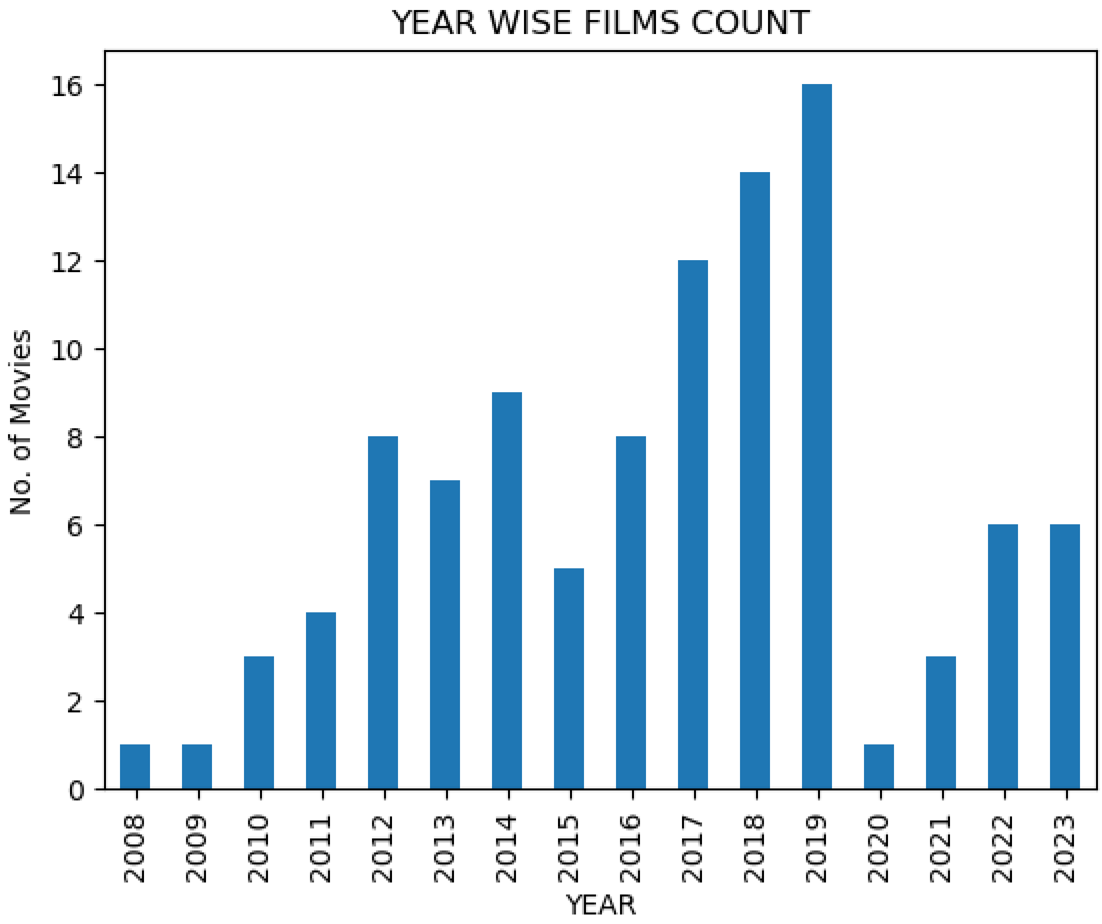
In [30]:
```
df1.plot(kind = 'pie',title = "Language & Multi Languages Acquried in INDIAN FILM MARKET
Out[30]: <Axes: title={'center': 'Language & Multi Languages Acquried in INDIAN FILM MARKET'}, yl
abel='PrimaryLanguage'>
```



Language & Multi Languages Acquried in INDIAN FILM MARKET

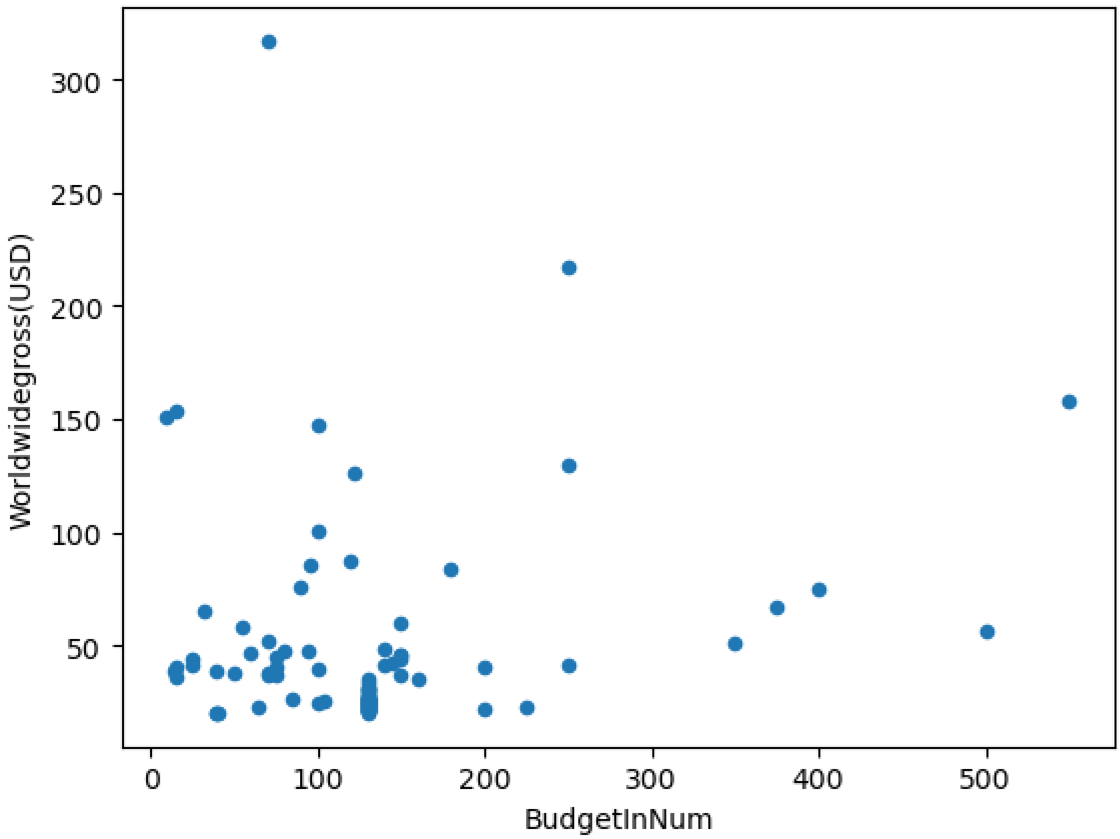
```
In [31]: df2 = data_set.groupby(['Year'])['Film'].count()
         df2
         df2.plot(kind = 'bar',xlabel = 'YEAR', ylabel = 'No. of Movies', title = 'YEAR WISE FILM
```

```
Out[31]: <Axes: title={'center': 'YEAR WISE FILMS COUNT'}, xlabel='YEAR', ylabel='No. of Movies'>
```



YEAR WISE FILMS COUNT

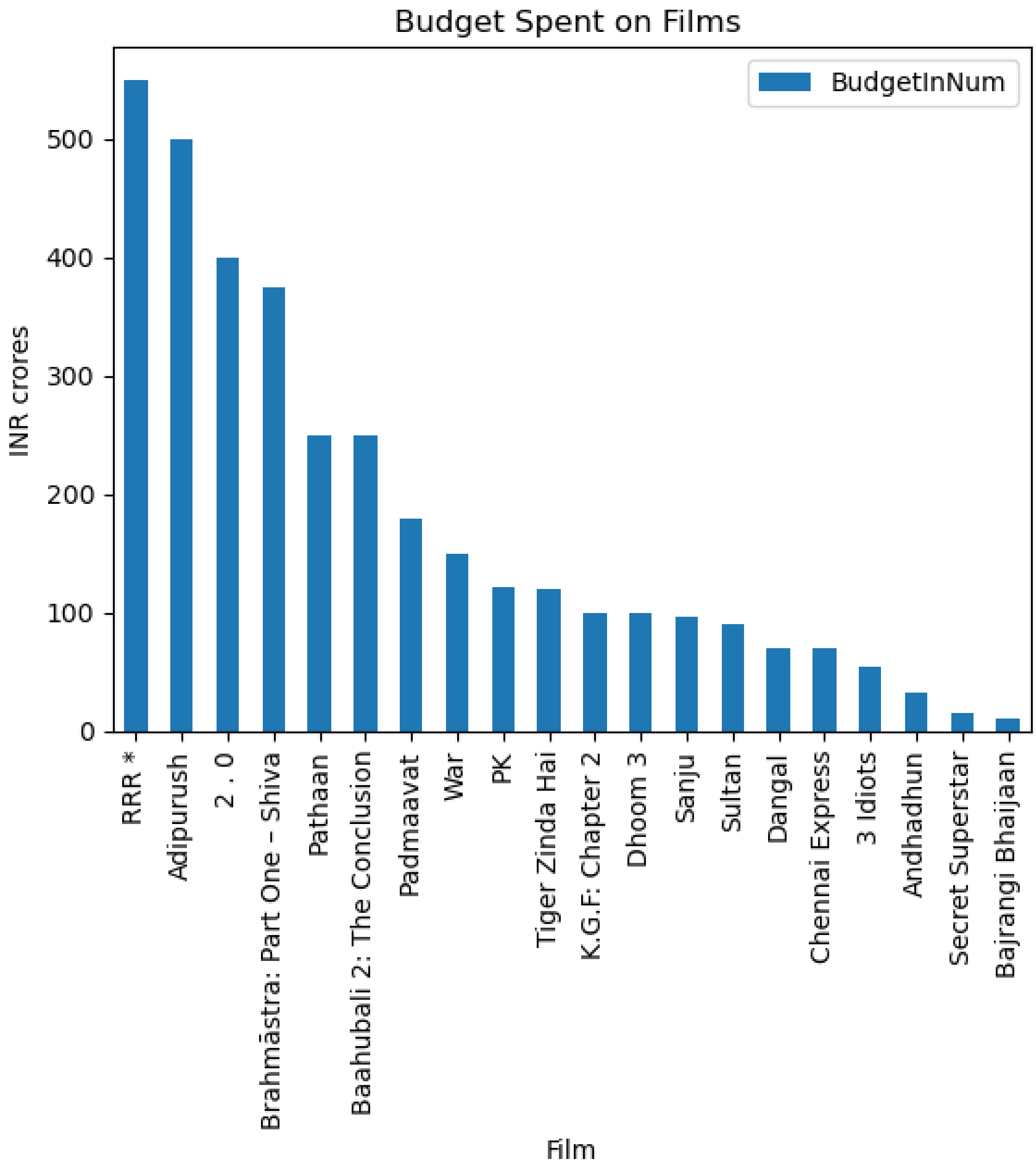
In [32]:
```
data_set.plot(kind='scatter', x='BudgetInNum', y='Worldwidegross(USD)')
plt.show()
```



In [33]:
```
data_set.head(20).sort_values(by = ['BudgetInNum'],ascending =[False]).plot(kind = 'bar'
```

Out[33]:
```
<Axes: title={'center': 'Budget Spent on Films '}, xlabel='Film', ylabel='INR crores'>
```

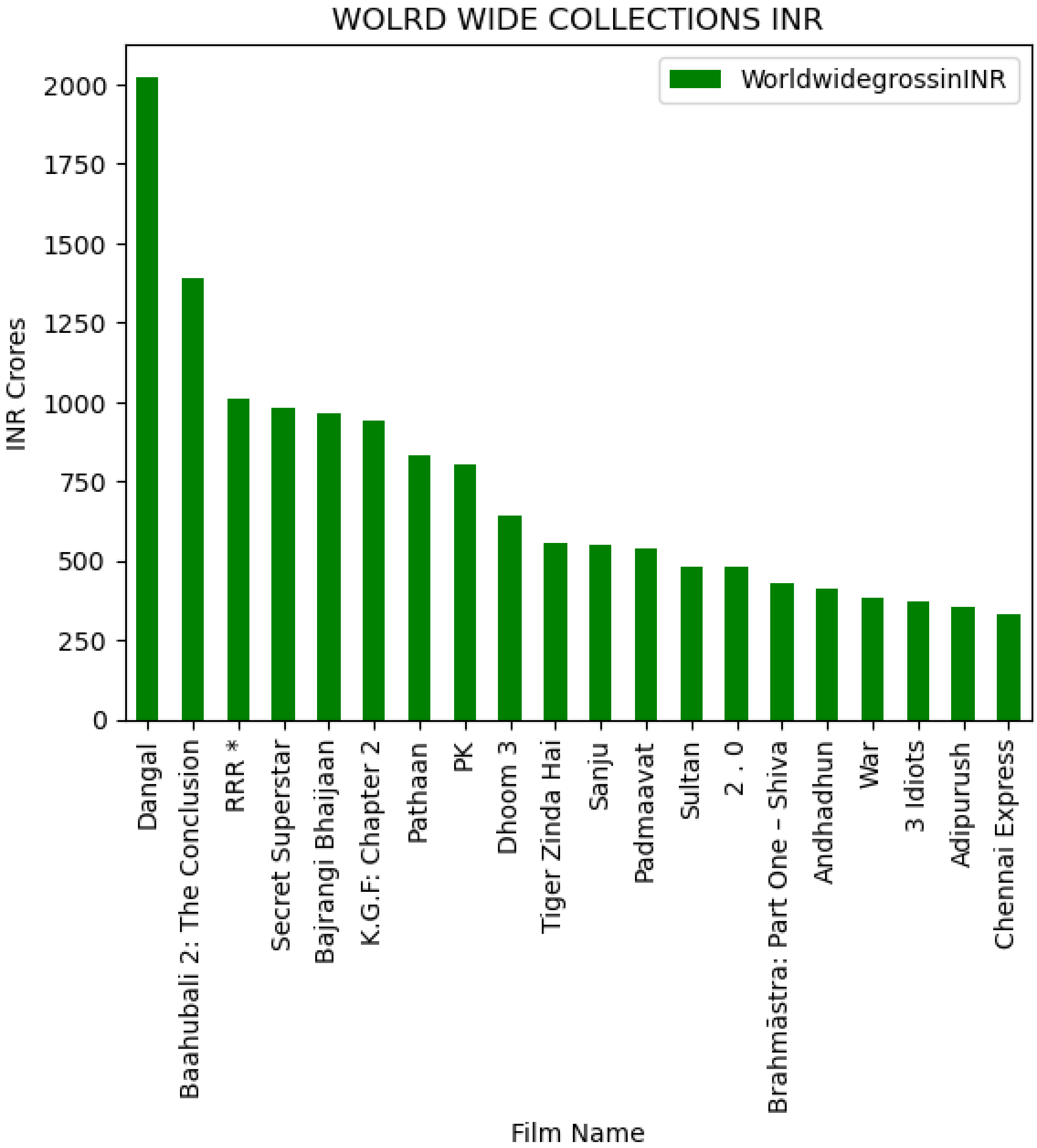
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



In [34]:
```
data_set['WorldwidegrossinINR'] = round(data_set['Worldwidegross(USD)']*6.385,2) #Not ex
data_set.head(20).sort_values(by = ['WorldwidegrossinINR'],ascending =[False]).plot(kind
```

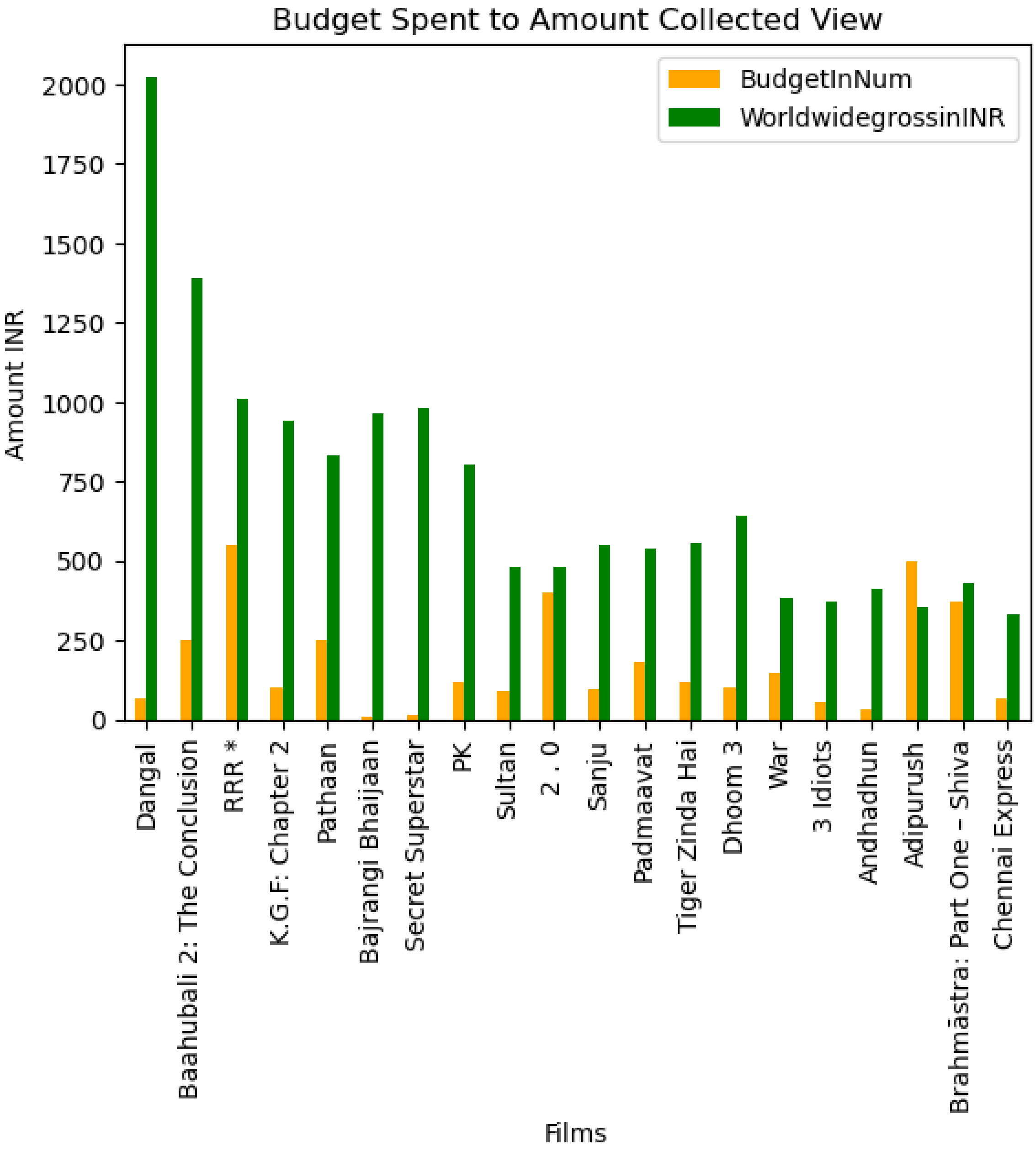
Out[34]:
```
<Axes: title={'center': 'WOLRD WIDE COLLECTIONS INR'}, xlabel='Film Name', ylabel='INR C
rores'>
```

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



```
In [35]: data_set.head(20).plot(kind= 'bar',x = 'Film', y = ['BudgetInNum','WorldwidegrossinINR']
```

```
Out[35]: <Axes: title={'center': 'Budget Spent to Amount Collected View'}, xlabel='Films', ylabel
         ='Amount INR'>
```



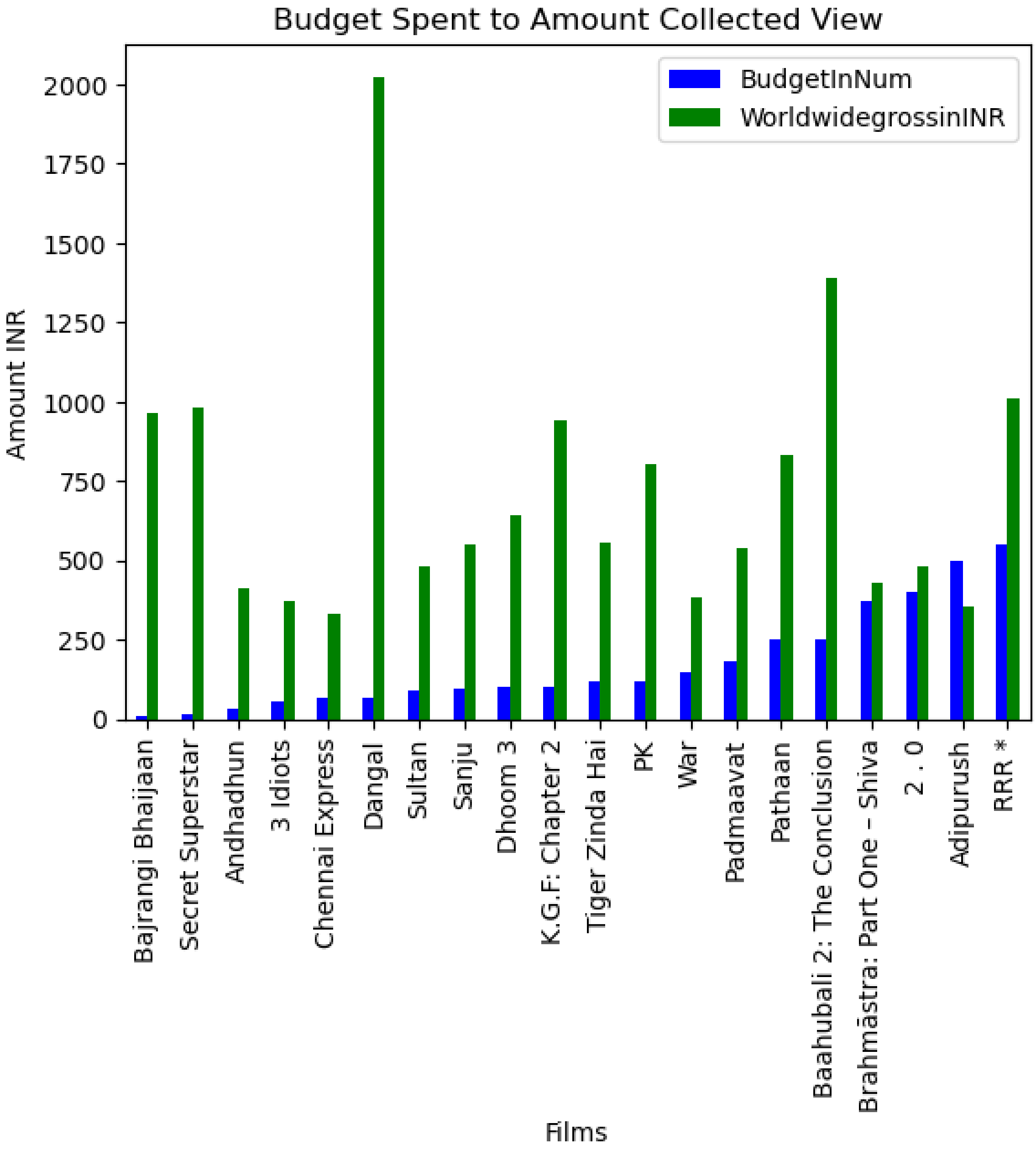
In [36]:

```
data_set.head(20).sort_values(['BudgetInNum','WorldwidegrossinINR'],ascending = [True,Tr
```

Out[36]:

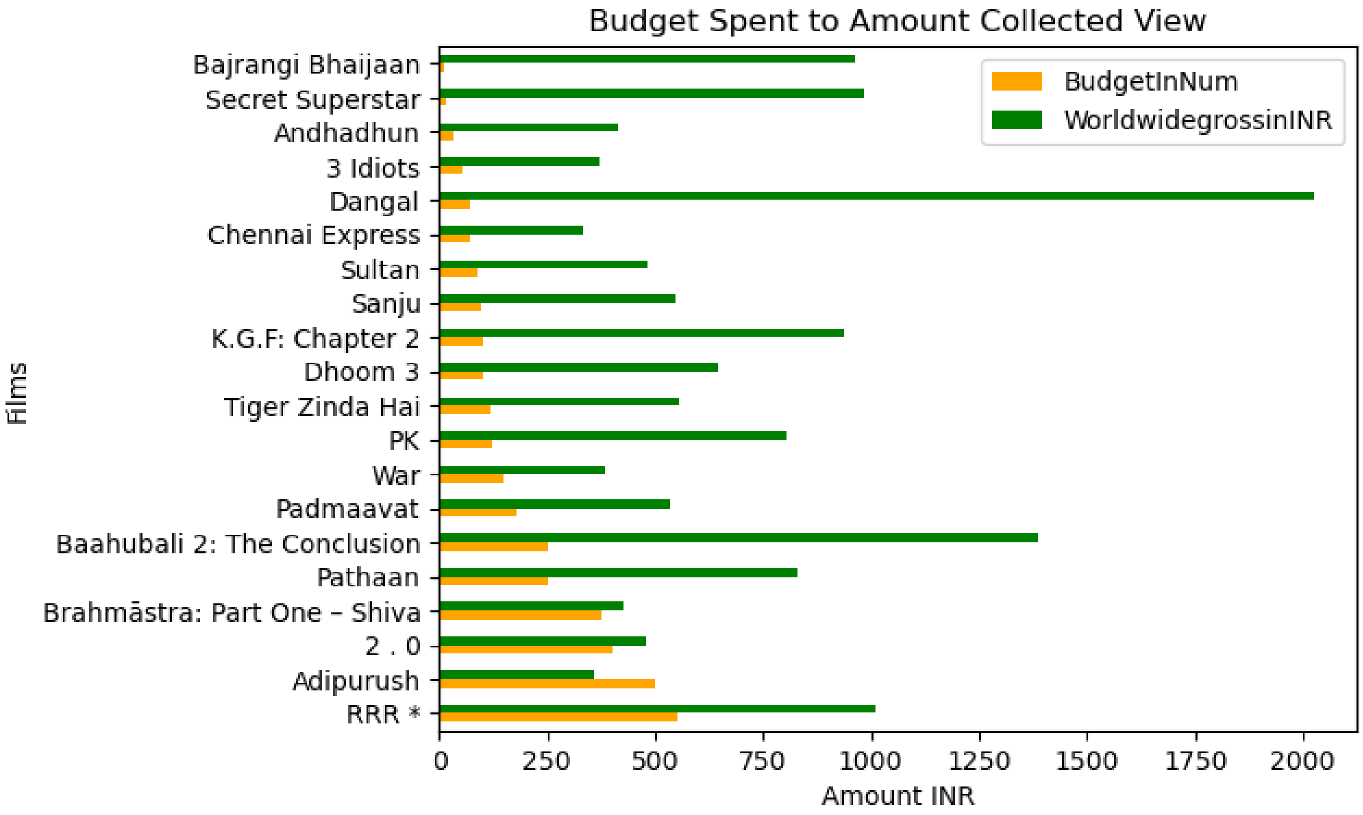
```
<Axes: title={'center': 'Budget Spent to Amount Collected View'}, xlabel='Films', ylabel
='Amount INR'>
```

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

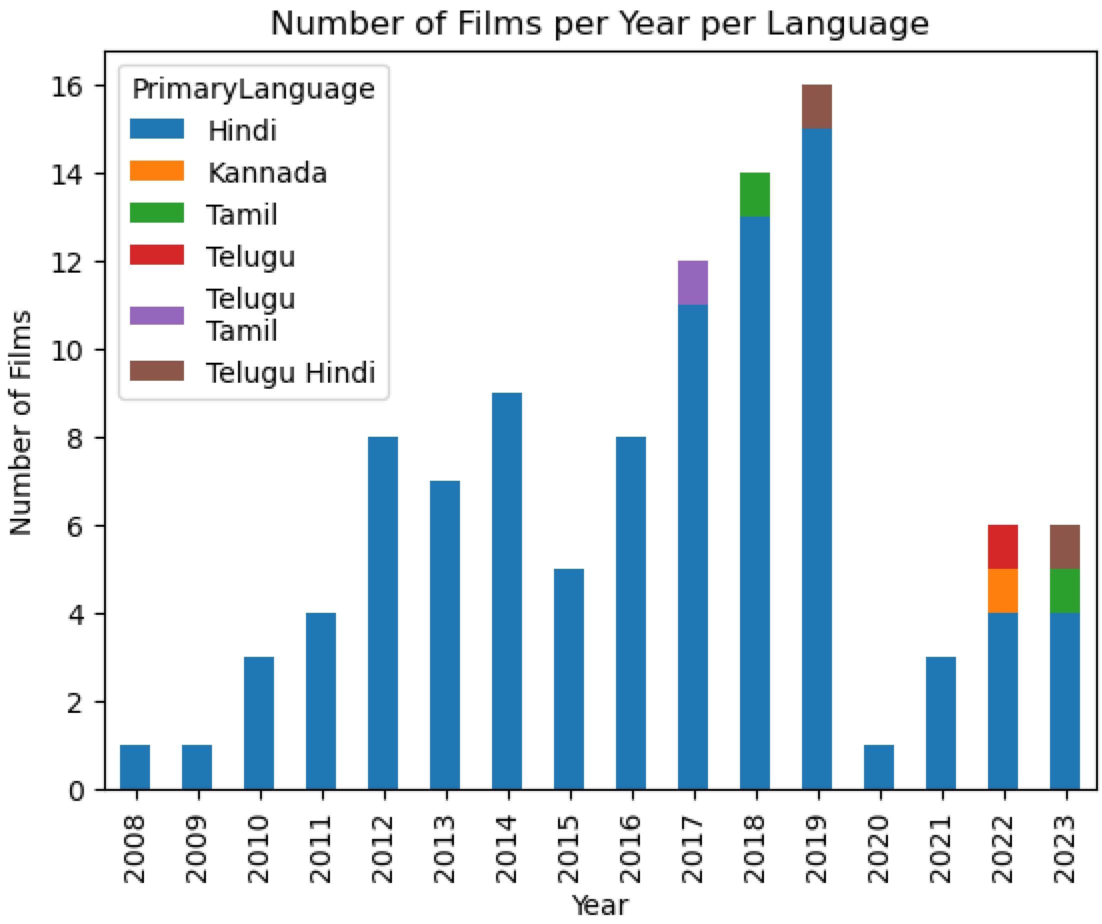


In [37]:
```
data_set.head(20).sort_values(['BudgetInNum','WorldwidegrossinINR'],ascending = [False,T
```

Out[37]:
```
<Axes: title={'center': 'Budget Spent to Amount Collected View'}, xlabel='Amount INR', y
label='Films'>
```



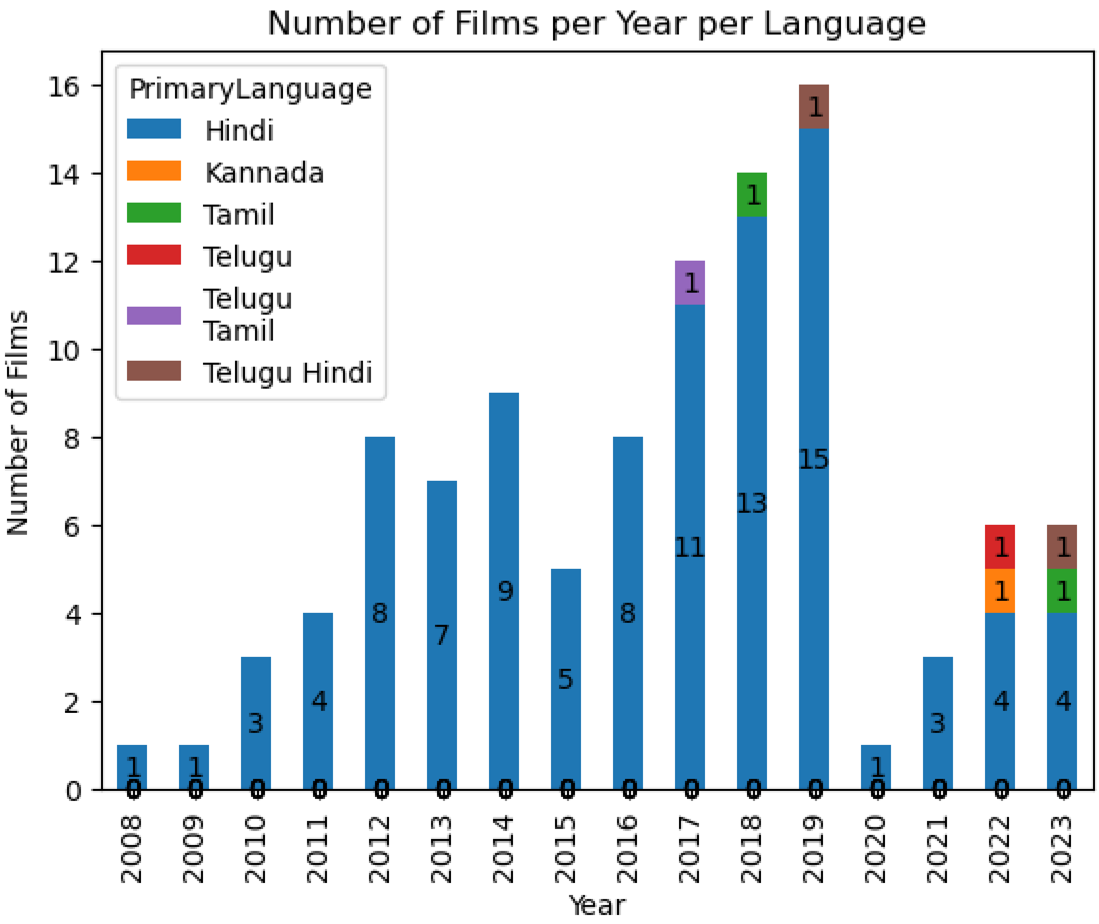
In [38]:
```
#using pyplot
df_grouped = data_set.groupby(['Year', 'PrimaryLanguage'])['Film'].count().reset_index()
# Pivoting the data for better visualization
df_pivot = df_grouped.pivot(index='Year', columns='PrimaryLanguage', values='Film')
#print(df_pivot)
df_pivot.plot(kind='bar', stacked=True)
plt.title('Number of Films per Year per Language')
plt.xlabel('Year')
plt.ylabel('Number of Films')
plt.show()
```



In [39]:

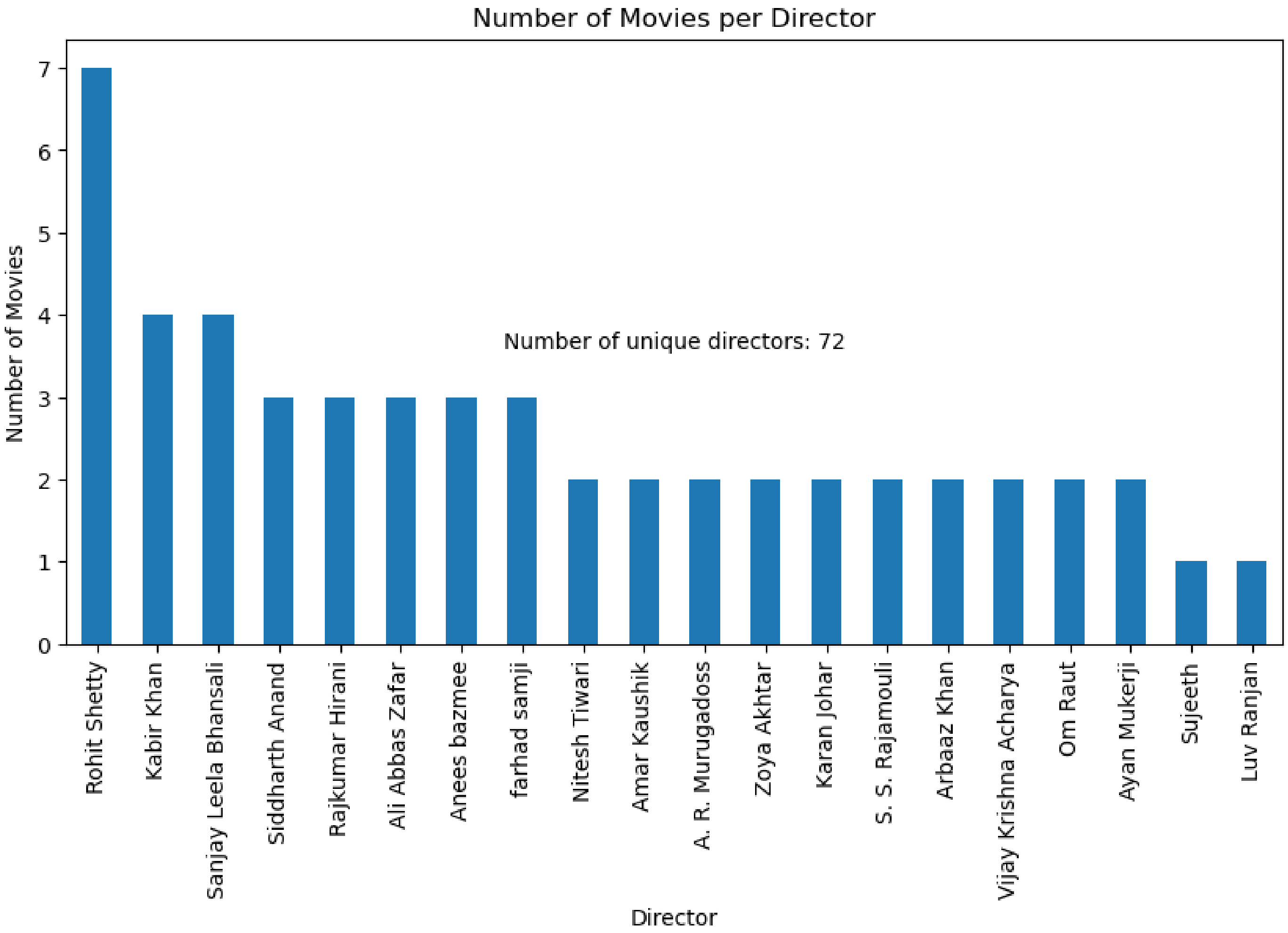
```
# using pyplot
df_grouped = data_set.groupby(['Year', 'PrimaryLanguage'])['Film'].count().reset_index()
# Pivoting the data for better visualization
df_pivot = df_grouped.pivot(index='Year', columns='PrimaryLanguage', values='Film')

ax = df_pivot.plot(kind='bar', stacked=True)

for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.text(x+width/2,
            y+height/2,
            '{:.0f}'.format(height),
            horizontalalignment='center',
            verticalalignment='center')

plt.title('Number of Films per Year per Language')
plt.xlabel('Year')
plt.ylabel('Number of Films')
plt.show()
```

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



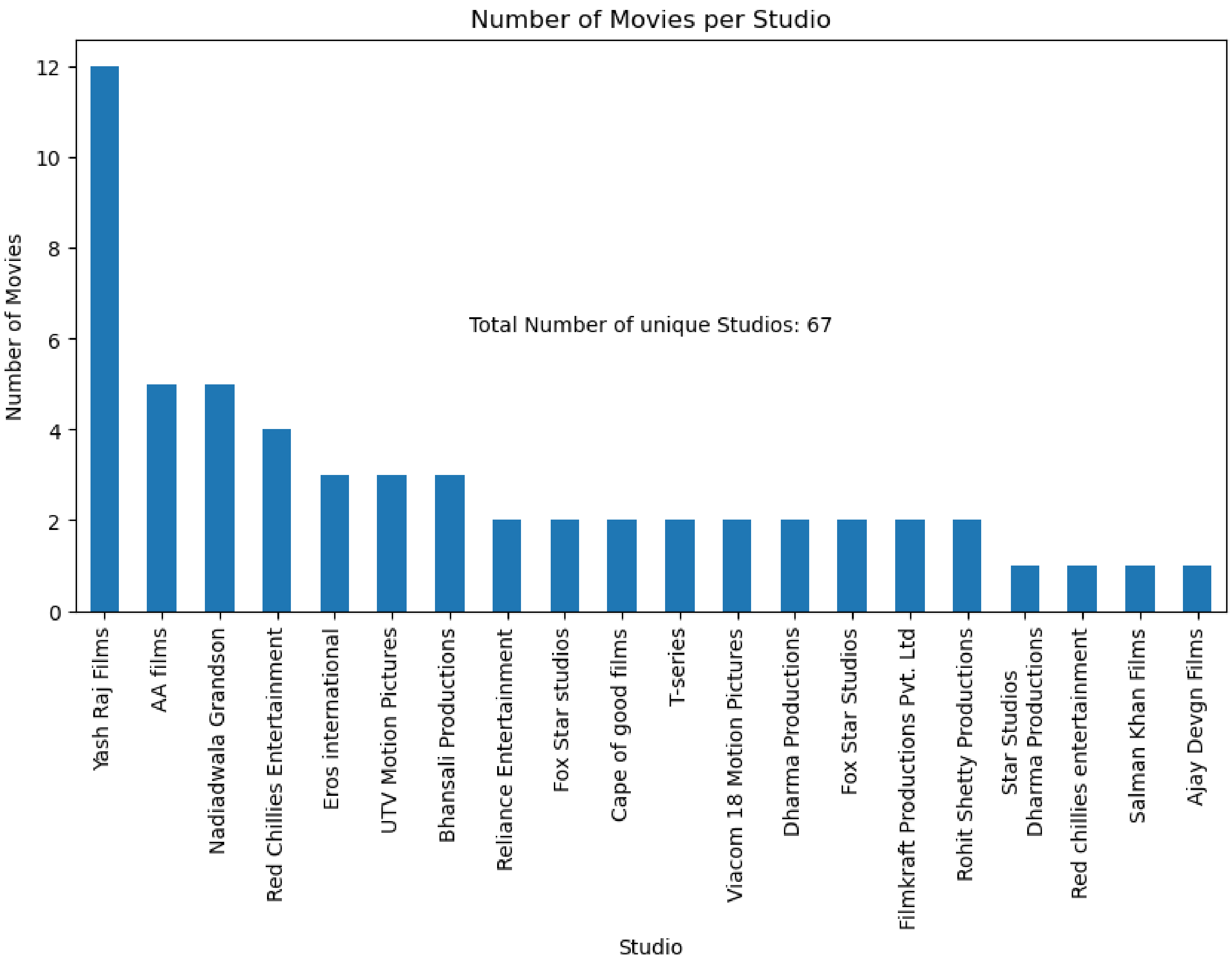
In [40]:
```
num_unique_directors = data_set['Director'].nunique()
director_counts = data_set['Director'].value_counts().head(20)
# Plot for directors
plt.figure(figsize=(10,5))
director_counts.plot(kind='bar')
plt.title('Number of Movies per Director')
plt.xlabel('Director')
plt.ylabel('Number of Movies')
# Display the number of unique directors
plt.text(0.5, 0.5, f'Number of unique directors: {num_unique_directors}', horizontalalig
plt.show()
```

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



In [41]:

```
num_unique_studios = data_set['Studio(s)'].nunique()
studio_counts = data_set['Studio(s)'].value_counts().head(20)
# Plot for studios
plt.figure(figsize=(10,5))
studio_counts.plot(kind='bar')
plt.title('Number of Movies per Studio')
plt.xlabel('Studio')
plt.ylabel('Number of Movies')
plt.text(0.5, 0.5, f'Total Number of unique Studios: {num_unique_studios}', horizontalal
plt.show()
```



Data Analysis Summary

Based on the analysis, here are some points help you understand the data better:

Hindi language films have been the most popular in the Indian film industry over the past 15 years, with the highest number of releases.

Hindi films account for approximately 95% of the Indian film industry

The year 2019 saw the highest number of film releases, while 2008, 2009, and 2020 saw the lowest .

The highest budget spent on a film was for a Telugu film named RRR .

The highest worldwide collection made by an Indian film was for Dangal, which is a Hindi-language film.

Bajrangi Bhaijaan is another Hindi-language film that made a high collection on a low budget .

Rohit Shetty has directed 7 Films, the highest number of films across the data available .

Yash Raj Films is the most commonly used studio for making films .

In [ ]:

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