KSR DATAVIZON

Key to Sustainable Results

INDIAN BOX OFFICE PROJECT

PYTHON

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt #Just to show the charts not interactive
import plotly.express as px #Interactive charts
```

In [2]:
```
df_boxoffice = pd.read_csv("D:\\Data Engineering\\Python\\BoxOffice Project\\Datasets_Final\\Boxoffice_Fact.csv")
df_director = pd.read_csv("D:\\Data Engineering\\Python\\BoxOffice Project\\Datasets_Final\\Director_dim.csv")
df_genere = pd.read_csv("D:\\Data Engineering\\Python\\BoxOffice Project\\Datasets_Final\\Genere_dim.csv")
df_language = pd.read_csv("D:\\Data Engineering\\Python\\BoxOffice Project\\Datasets_Final\\Language_dim.csv")
```

In [3]:
```
df_boxoffice.sample()
```

Out[3]:

	FilmID	Title	Release_Date	DirectorID	Lead_Actor/Actress	LanguageID	Industry	GenreID	Budget_in_Crores	First_Day_Colle
286	316	Sardar	21-Oct-22	158	Karthi	503	Kollywood	627	35	

In [4]:
```
df_boxoffice.isna().sum()
```

Out[4]:
```
FilmID                                  0
Title                                   0
Release_Date                            0
DirectorID                              0
Lead_Actor/Actress                      0
LanguageID                              0
Industry                                0
GenreID                                 0
Budget_in_Crores                        0
First_Day_Collection_Worldwide_in_Crores    0
Worldwide_Collection _in_Crores         0
Overseas_Collection_in_Crores           9
India_Gross_Collection_in_Crores        0
Verdict                                 0
IMDb_Rating                             0
Runtime_(mins)                          0
OTT_Platform                            0
dtype: int64
```

In [5]:
```
df_boxoffice['Budget_in_Crores'].str.isnumeric().sum() #to find how numeric value count
```

Out[5]:
```
587
```

In [6]:
```
df_boxoffice['Budget_in_Crores'] = df_boxoffice['Budget_in_Crores'].str.replace(":","")
```

In [7]:
```
df_boxoffice['Budget_in_Crores'] = df_boxoffice['Budget_in_Crores'].str.replace(" ","")
```

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

Out[8]:
```
FilmID                                  int64
Title                                  object
Release_Date                           object
DirectorID                              int64
Lead_Actor/Actress                     object
LanguageID                              int64
Industry                               object
GenreID                                 int64
Budget_in_Crores                       object
First_Day_Collection_Worldwide_in_Crores    object
Worldwide_Collection _in_Crores        object
Overseas_Collection_in_Crores          object
India_Gross_Collection_in_Crores       object
Verdict                                object
IMDb_Rating                           float64
Runtime_(mins)                          int64
OTT_Platform                           object
dtype: object
```

In [9]:
```
df_boxoffice['Budget_in_Crores'] = df_boxoffice['Budget_in_Crores'].astype(float)
```

In [10]:
```
df_boxoffice['Budget_in_Crores'].sum()
```

Out[10]:
```
34653.0
```

In [11]:
```
df_boxoffice['First_Day_Collection_Worldwide_in_Crores'].str.isnumeric().sum()
```

Out[11]:
```
176
```

In [12]:
```
df_boxoffice['First_Day_Collection_Worldwide_in_Crores'] = df_boxoffice['First_Day_Collection_Worldwide_in_Crores'].str.re
```

In [13]:
```
df_boxoffice['First_Day_Collection_Worldwide_in_Crores'] = df_boxoffice['First_Day_Collection_Worldwide_in_Crores'].str.re
```

In [14]:
```
df_boxoffice['First_Day_Collection_Worldwide_in_Crores'] = df_boxoffice['First_Day_Collection_Worldwide_in_Crores'].astype
```

In [15]:
```
df_boxoffice['First_Day_Collection_Worldwide_in_Crores'].sum()
```

Out[15]:
```
11070.27
```

In [16]:
```
df_boxoffice['Worldwide_Collection _in_Crores'].str.isnumeric().tail()
```

Out[16]:
```
599    False
600    False
601    False
602    False
603    False
Name: Worldwide_Collection _in_Crores, dtype: bool
```

In [17]:
```
df_boxoffice['Worldwide_Collection _in_Crores'] = df_boxoffice['Worldwide_Collection _in_Crores'].str.replace(":","")
```

In [18]:
```
df_boxoffice['Worldwide_Collection _in_Crores'] = df_boxoffice['Worldwide_Collection _in_Crores'].str.replace(" ","")
```

In [19]:
```
df_boxoffice['Worldwide_Collection _in_Crores'] = df_boxoffice['Worldwide_Collection _in_Crores'].astype(float)
```

In [20]:
```
df_boxoffice['Worldwide_Collection _in_Crores'].sum()
```

Out[20]:
```
79878.79
```

In [21]:
```
df_boxoffice['Overseas_Collection_in_Crores'] = df_boxoffice['Overseas_Collection_in_Crores'].str.replace(":","")
```

In [22]:
```
df_boxoffice['Overseas_Collection_in_Crores'] = df_boxoffice['Overseas_Collection_in_Crores'].str.replace(" ","")
```

In [23]:
```
df_boxoffice['Overseas_Collection_in_Crores'] = df_boxoffice['Overseas_Collection_in_Crores'].astype(float)
```

In [24]:
```
df_boxoffice['Overseas_Collection_in_Crores'].sum()
```

Out[24]:
```
20953.07
```

In [25]:
```
df_boxoffice['India_Gross_Collection_in_Crores'] = df_boxoffice['India_Gross_Collection_in_Crores'].str.replace(":","")
```

In [26]:
```
df_boxoffice['India_Gross_Collection_in_Crores'] = df_boxoffice['India_Gross_Collection_in_Crores'].astype(float)
```

In [27]:
```
df_boxoffice['India_Gross_Collection_in_Crores'].sum()
```

Out[27]:
```
60064.1
```

In [28]:
```
df_boxoffice.dtypes
```

Out[28]:
```
FilmID                                 int64
Title                                 object
Release_Date                          object
DirectorID                             int64
Lead_Actor/Actress                    object
LanguageID                             int64
Industry                              object
GenreID                                int64
Budget_in_Crores                     float64
First_Day_Collection_Worldwide_in_Crores    float64
Worldwide_Collection _in_Crores      float64
Overseas_Collection_in_Crores        float64
India_Gross_Collection_in_Crores     float64
Verdict                               object
IMDb_Rating                          float64
Runtime_(mins)                         int64
OTT_Platform                          object
dtype: object
```

In [29]:
```
list(df_boxoffice["Verdict"].unique())
```

Out[29]:
```
['Blockbuster',
 'All Time Blockbuster',
 'Below Average',
 ': Disaster',
 ': Blockbuster',
 ': All Time Blockbuster',
 'Super Hit',
 ': Flop',
 'Above Average',
 'Flop',
 ': Hit',
 ':  Super Hit',
 ': Super Hit',
 ': Average',
 'Disaster',
 'Average',
 'Hit',
 ': Below Average',
 ': Above Average',
 ' Below Average',
 'Super Hitt']
```

In [30]:
```
df_boxoffice["Verdict"] = df_boxoffice["Verdict"].str.replace(":","")
```

In [31]:
```
df_boxoffice["Verdict"].unique()
```

Out[31]:
```
array(['Blockbuster', 'All Time Blockbuster', 'Below Average',
       ' Disaster', ' Blockbuster', ' All Time Blockbuster', 'Super Hit',
       ' Flop', 'Above Average', 'Flop', ' Hit', '  Super Hit',
       ' Super Hit', ' Average', 'Disaster', 'Average', 'Hit',
       ' Below Average', ' Above Average', 'Super Hitt'], dtype=object)
```

In [32]:
```
df_boxoffice["Verdict"] = df_boxoffice["Verdict"].str.lstrip()  #lstrip will remove the spaces left side of string
```

In [33]:
```
df_boxoffice["Verdict"].unique()
```

Out[33]:
```
array(['Blockbuster', 'All Time Blockbuster', 'Below Average', 'Disaster',
       'Super Hit', 'Flop', 'Above Average', 'Hit', 'Average',
       'Super Hitt'], dtype=object)
```

In [34]:
```
df_boxoffice["IMDb_Rating"].sum()
```

Out[34]:
```
3892.1
```

In [35]:
```
df_boxoffice["OTT_Platform"].unique()
```

Out[35]:
```
array(['Netflix', 'ZEE5', 'Disney+ Hotstar', 'Amazon Prime Video',
       'Aha  ', 'Google Play Movies', 'Amazon Prime Video ',
       'Disney+ Hotstar ', 'Netflix ', 'Sun NXT', 'ZEE5  ', 'Sun NXT  ',
       'Amazon Prime Video  ', 'SonyLIV  ', 'Netflix  ', 'Voot  ',
       'Disney+ Hotstar  ', 'aha  ', 'ZEE5 ', 'SonyLIV', 'ZEE5      ',
       'JioCinema', 'Aha'], dtype=object)
```

In [36]:
```
df_boxoffice["OTT_Platform"] = df_boxoffice["OTT_Platform"].str.rstrip() #rstrip will remove the spaces Right side of stri
```

In [37]:
```
df_boxoffice["OTT_Platform"].unique()
```

Out[37]:
```
array(['Netflix', 'ZEE5', 'Disney+ Hotstar', 'Amazon Prime Video', 'Aha',
       'Google Play Movies', 'Sun NXT', 'SonyLIV', 'Voot', 'aha',
       'JioCinema'], dtype=object)
```

In [38]:
```
df_boxoffice["OTT_Platform"] = df_boxoffice["OTT_Platform"].str.replace("aha","Aha")
```

In [39]:
```
df_boxoffice["OTT_Platform"].unique()
```

Out[39]:
```
array(['Netflix', 'ZEE5', 'Disney+ Hotstar', 'Amazon Prime Video', 'Aha',
       'Google Play Movies', 'Sun NXT', 'SonyLIV', 'Voot', 'JioCinema'],
      dtype=object)
```

In [40]:
```
df_boxoffice["LanguageID"].unique()
```

Out[40]:
```
array([501, 503, 507, 505, 509, 511, 515, 517, 513, 519], dtype=int64)
```

In [41]:
```
df_language
```

Out[41]:

	Language	LanguageID
0	Hindi	501
1	Tamil	503
2	Telugu	505
3	Kannada	507
4	Malayalam	509

In [42]:
```
df_boxoffice.groupby(["LanguageID","Industry"])[["Budget_in_Crores"]].sum()
```

Out[42]:

		Budget_in_Crores
LanguageID	**Industry**	
501	**Bollywood**	13640.0
503	**Kollywood**	8400.0
505	**Tollywood**	7272.5
507	**Sandalwood**	866.0
509	**Mollywood**	1199.5
511	**Tollywood**	1872.0
513	**Mollywood**	143.0
515	**Bollywood**	895.0
517	**Kollywood**	282.0
519	**Sandalwood**	83.0

In [43]:
```
#511 --> 505
#513 --> 509
#515 --> 501
#517 --> 503
#519 --> 507
```

In [44]:
```
df_boxoffice["LanguageID"] = df_boxoffice["LanguageID"].replace({511:505,513:509,515:501,517:503,519:507}) #replacing the
```

In [45]:
```
df_boxoffice.groupby(["LanguageID","Industry"])[["Budget_in_Crores"]].sum()
```

Out[45]:

		Budget_in_Crores
LanguageID	**Industry**	
501	**Bollywood**	14535.0
503	**Kollywood**	8682.0
505	**Tollywood**	9144.5
507	**Sandalwood**	949.0
509	**Mollywood**	1342.5

In [46]:
```
df_boxoffice['Release_Date'] = pd.to_datetime(df_boxoffice['Release_Date']) #converting the date column data type to date.
```

```
C:\Users\supar\AppData\Local\Temp\ipykernel_18288\623155405.py:1: UserWarning: Could not infer format, so each element will
be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a forma
t.
  df_boxoffice['Release_Date'] = pd.to_datetime(df_boxoffice['Release_Date']) #converting the date column data type to dat
e.
```

In [47]:
```
df_boxoffice.dtypes
```

Out[47]:
```
FilmID                                        int64
Title                                        object
Release_Date                         datetime64[ns]
DirectorID                                    int64
Lead_Actor/Actress                           object
LanguageID                                    int64
Industry                                     object
GenreID                                       int64
Budget_in_Crores                            float64
First_Day_Collection_Worldwide_in_Crores    float64
Worldwide_Collection _in_Crores             float64
Overseas_Collection_in_Crores               float64
India_Gross_Collection_in_Crores            float64
Verdict                                      object
IMDb_Rating                                 float64
Runtime_(mins)                                int64
OTT_Platform                                 object
dtype: object
```

In [48]:
```
df_boxoffice["Year"] = df_boxoffice['Release_Date'].dt.year  #extracting the year from released date
```

In [49]:
```
df_boxoffice["Month"] = df_boxoffice["Release_Date"].dt.month #extracting the month number from release date
```

In [50]:
```
df_boxoffice["Month_Name"] = df_boxoffice['Release_Date'].dt.month_name() # extracting the month name from release date
```

In [51]:
```
df_boxoffice["Week_Name"] = df_boxoffice['Release_Date'].dt.day_name() #extracting the week name from release date
```

In [52]:
```
df_boxoffice.head(5)
```

Out[52]:

	FilmID	Title	Release_Date	DirectorID	Lead_Actor/Actress	LanguageID	Industry	GenreID	Budget_in_Crores	First_Day_Co
0	34	Sanju	2018-06-29	101	Ranbir Kapoor	501	Bollywood	623	100.0	
1	36	Simmba	2018-12-28	105	Ranveer Singh	501	Bollywood	605	125.0	
2	225	Janatha Garage	2016-09-01	140	N.T. Rama Rao Jr.	503	Kollywood	605	50.0	
3	13	Kaththi	2014-10-22	107	Joseph Vijay	503	Kollywood	606	70.0	
4	31	Maanikya	2014-05-01	130	V. Ravichandran	507	Sandalwood	606	15.0	

5 rows × 21 columns

In [53]:
```
df_boxoffice["Overseas_Collection_in_Crores"]=df_boxoffice["Overseas_Collection_in_Crores"].fillna(0) #filling null vales
```

In [54]:
```
#Key KPIs
```

In [55]:
```
#Totalfilms in this period
```

In [56]:
```
films = df_boxoffice['FilmID'].count()
print("Total films :",films)
```

```
Total films : 604
```

In [57]:
```
#Key KPIs
```

In [58]:
```
df_boxoffice[["Budget_in_Crores","Worldwide_Collection _in_Crores","First_Day_Collection_Worldwide_in_Crores",\
              "Overseas_Collection_in_Crores","India_Gross_Collection_in_Crores"]].sum()
```

Out[58]:
```
Budget_in_Crores                          34653.00
Worldwide_Collection _in_Crores           79878.79
First_Day_Collection_Worldwide_in_Crores  11070.27
Overseas_Collection_in_Crores             20953.07
India_Gross_Collection_in_Crores          60064.10
dtype: float64
```

In [59]:
```
df_boxoffice.groupby(['Title'])[["Worldwide_Collection _in_Crores"]].sum().sort_values("Worldwide_Collection _in_Crores",a
#Top  10 filmsbased on world wide collections
```

Out[59]:

Title	Worldwide_Collection _in_Crores
Dangal	2122.30
Bāhubali 2: The Conclusion	1788.00
RRR (Rise Roar Revolt)	1230.00
K.G.F: Chapter 2	1215.00
Jawan	1160.00
Pathaan	1055.00
Kalki 2898-AD	1042.25
Bajrangi Bhaijaan	922.10
Animal	915.00
Secret Superstar	912.60

In [60]:
```
df_boxoffice.groupby(["Year"])[["FilmID"]].count().sort_values("FilmID",ascending=False).rename(columns={"FilmID":"Film Co
#No of films released by year
```

Out[60]:

Year	Film Count
2022	80
2023	79
2019	74
2017	72
2018	71
2024	65
2016	41
2014	32
2015	32
2021	31
2020	27

In [61]:
```
#Top  10 filmsbased on india collections
df_boxoffice.groupby(['Title'])[["India_Gross_Collection_in_Crores"]].sum().sort_values("India_Gross_Collection_in_Crores"
```

Out[61]:

Title	India_Gross_Collection_in_Crores
Bāhubali 2: The Conclusion	1417.00
K.G.F: Chapter 2	1000.85
RRR (Rise Roar Revolt)	915.85
Kalki 2898-AD	767.25
Jawan	760.00
Stree 2: Sarkate Ka Aatank	713.07
Animal	660.00
Pathaan	657.50
Gadar 2	620.50
Dangal	587.00

In [62]:
```
#Top  10 filmsbased on overses collections
df_boxoffice.groupby(["Title"])[["Overseas_Collection_in_Crores"]].sum().sort_values("Overseas_Collection_in_Crores",ascer
```

Out[62]:

Title	Overseas_Collection_in_Crores
Dangal	1535.30
Secret Superstar	831.40
Bajrangi Bhaijaan	489.70
Jawan	400.00
Pathaan	397.50
Bāhubali 2: The Conclusion	371.00
RRR (Rise Roar Revolt)	314.15
PK	303.00
Kalki 2898-AD	275.00
Animal	255.00

In [63]:
```
#Top  10 filmsbased on firstday collections
df_boxoffice.groupby(["Title"])[["First_Day_Collection_Worldwide_in_Crores"]].sum()\
            .sort_values("First_Day_Collection_Worldwide_in_Crores",ascending=False).iloc[:10]
```

Out[63]:

Title	First_Day_Collection_Worldwide_in_Crores
RRR (Rise Roar Revolt)	223.0
Bāhubali 2: The Conclusion	217.0
Kalki 2898-AD	177.7
K.G.F: Chapter 2	159.0
Salaar: Part 1 - Ceasefire	158.1
Leo	142.7
Devara: Part 1	142.0
Jawan	129.1
Adipurush	127.5
Animal	116.0

In [64]:
```
#world wide collection by Verdict
df_boxoffice.groupby(["Verdict"])[["Worldwide_Collection _in_Crores"]].sum().sort_values("Worldwide_Collection _in_Crores"
```

Out[64]:

Verdict	Worldwide_Collection _in_Crores
Blockbuster	25556.43
Hit	12366.04
All Time Blockbuster	11126.55
Super Hit	10106.47
Below Average	4887.41
Flop	4327.42
Disaster	4071.31
Average	3578.32
Above Average	3288.90
Super Hitt	569.94

In [65]:
```
df_boxoffice["Verdict"] = df_boxoffice["Verdict"].str.replace("Super Hitt","Super Hit")
```

In [66]:
```
df_boxoffice["Verdict"].unique()
```

Out[66]:
```
array(['Blockbuster', 'All Time Blockbuster', 'Below Average', 'Disaster',
       'Super Hit', 'Flop', 'Above Average', 'Hit', 'Average'],
      dtype=object)
```

In [67]:
```
df_boxoffice.groupby(["Verdict"])[["Worldwide_Collection _in_Crores"]].sum().sort_values("Worldwide_Collection _in_Crores"
```

Out[67]:

Verdict	Worldwide_Collection _in_Crores
Blockbuster	25556.43
Hit	12366.04
All Time Blockbuster	11126.55
Super Hit	10676.41
Below Average	4887.41
Flop	4327.42
Disaster	4071.31
Average	3578.32
Above Average	3288.90

In [68]:
```
#weekday wise films released
df_boxoffice.groupby("Week_Name")[["FilmID"]].count().rename(columns={"FilmID":"Film Count"})
```

Out[68]:

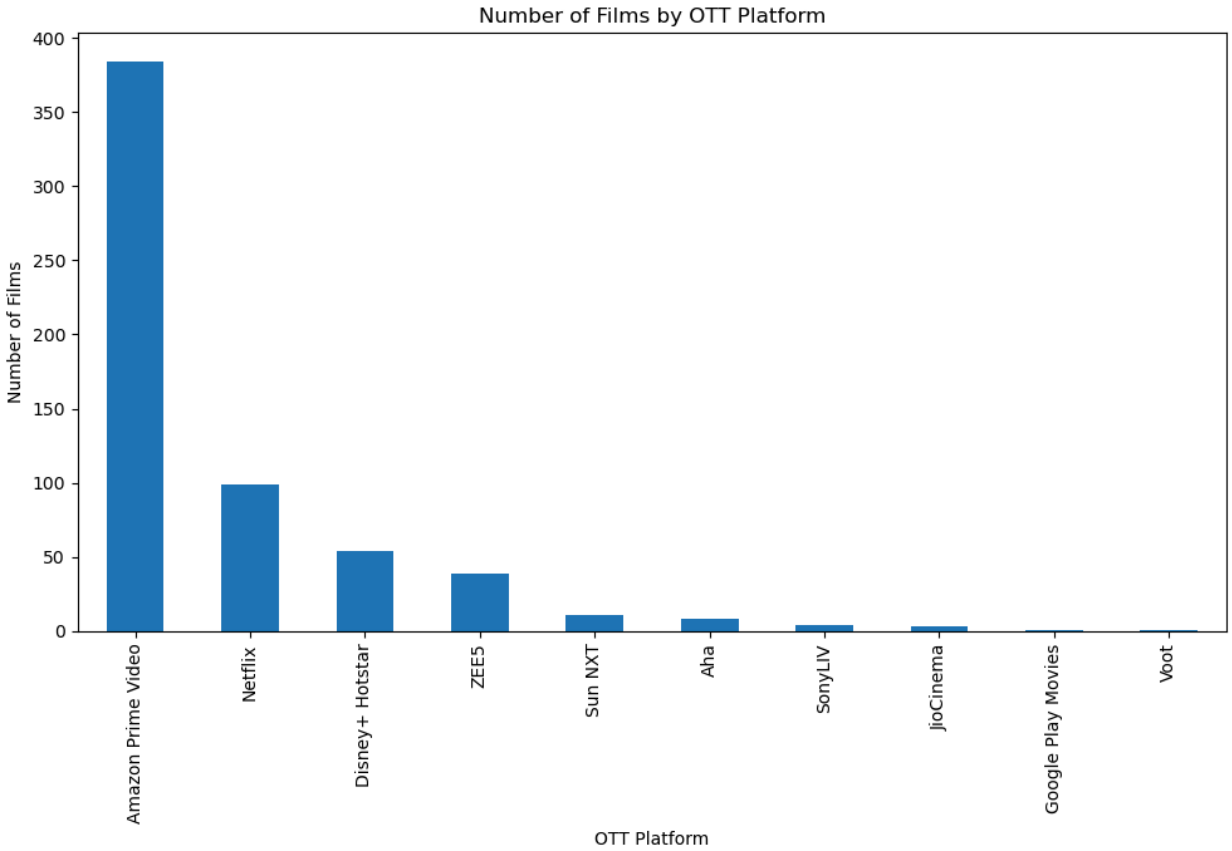
Week_Name	Film Count
Friday	423
Saturday	10
Sunday	7
Thursday	124
Tuesday	5
Wednesday	35

In [69]:
```
#which OTT platofrm has more films
df_boxoffice.groupby(["OTT_Platform"])[["FilmID"]].count().sort_values("FilmID",ascending=False).rename(columns={"FilmID":
```

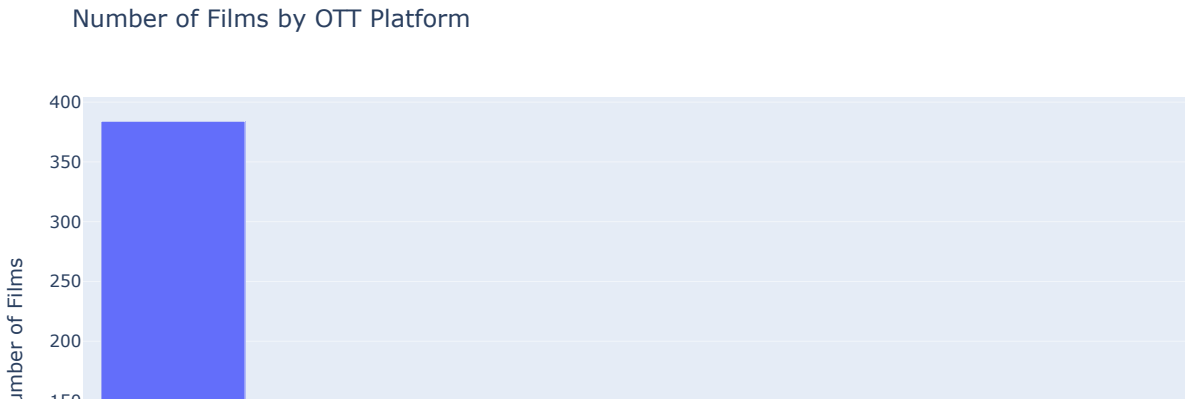
Out[69]:

OTT_Platform	Film Count
Amazon Prime Video	384
Netflix	99
Disney+ Hotstar	54
ZEE5	39
Sun NXT	11
Aha	8
SonyLIV	4
JioCinema	3
Google Play Movies	1
Voot	1

In [70]:
```
df_boxoffice.groupby(["OTT_Platform"])[["FilmID"]].count().sort_values("FilmID",ascending=False).plot(kind='bar',legend=Fa
plt.title("Number of Films by OTT Platform")
plt.xlabel("OTT Platform")
plt.ylabel("Number of Films")
#plt.xticks(rotation=45)
plt.tight_layout()              #not inetractive
```



In [71]:
```
ottwisefilms = df_boxoffice.groupby(["OTT_Platform"])[["FilmID"]].count().sort_values("FilmID",ascending=False).reset_inde
fig = px.bar(ottwisefilms, x='OTT_Platform',y='FilmID',title="Number of Films by OTT Platform",\
                           labels={"FilmID":"Number of Films","OTT_Platform":"OTT Platform"},\
                           hover_data={"OTT_Platform":True,"FilmID":True})
fig.show()
```



In [72]:
```
df_boxoffice.groupby(["OTT_Platform"])[["FilmID"]].count().sort_values("FilmID",ascending=False).rename(columns={"FilmID":
```

Out[72]:

OTT_Platform	Film Count
Amazon Prime Video	384
Netflix	99
Disney+ Hotstar	54
ZEE5	39
Sun NXT	11
Aha	8
SonyLIV	4
JioCinema	3
Google Play Movies	1
Voot	1

In [73]:
```
df_boxoffice.groupby(["OTT_Platform"])[["FilmID"]].count().sort_values("FilmID",ascending=False).rename(columns={"FilmID":
```

Out[73]:

OTT_Platform	Film Count
Amazon Prime Video	384
Netflix	99
Disney+ Hotstar	54
ZEE5	39
Sun NXT	11
Aha	8
SonyLIV	4
JioCinema	3
Google Play Movies	1
Voot	1

In [74]:
```
#Top 10 Directors by films released
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner")\
  .groupby(["Director"])[["FilmID"]].count().sort_values("FilmID",ascending=False).iloc[:10].rename(columns={"FilmID":"Fil
```

Out[74]:

Director	Film Count
Trivikram Srinivas	6
Rohit Shetty	6
A.R. Murugadoss	5
Siva	5
Anil Ravipudi	5
Boyapati Srinu	5
Koratala Siva	5
Remo D'Souza	4
Atlee	4
Siddharth Anand	4

In [75]:
```
#Top 10 directors by world wide collection
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner")\
  .groupby(["Director"])[["Worldwide_Collection _in_Crores"]].sum().sort_values("Worldwide_Collection _in_Crores",ascendir
```

Out[75]:

Director	Worldwide_Collection _in_Crores
S.S. Rajamouli	3668.00
Siddharth Anand	2229.45
Nitesh Tiwari	2122.30
Prashanth Neel	2101.75
Atlee	1865.35
Rajkumar Hirani	1834.50
Rohit Shetty	1656.60
Lokesh Kanagaraj	1350.13
Sandeep Reddy Vanga	1342.00
Kabir Khan	1220.55

In [76]:
```
#Top10 lead actors by world wide collection
df_boxoffice.groupby(["Lead_Actor/Actress"])[["Worldwide_Collection _in_Crores"]].sum()\
            .sort_values("Worldwide_Collection _in_Crores",ascending=False).iloc[:10]
```

Out[76]:

Lead_Actor/Actress	Worldwide_Collection _in_Crores
Prabhas	5091.50
Salman Khan	4515.20
Shah Rukh Khan	3800.10
Akshay Kumar	3168.22
Joseph Vijay	3076.11
Aamir Khan	2914.30
Rajinikanth	2695.55
Ajay Devgn	2571.21
N.T. Rama Rao Jr.	2242.60
Ranbir Kapoor	2111.25

In [77]:
```
#Top 10 movies by IMDb rating
df_boxoffice[["Title","IMDb_Rating"]].sort_values("IMDb_Rating",ascending=False).iloc[:10]
```

Out[77]:

	Title	IMDb_Rating
121	12th Fail	8.8
316	Rocketry: The Nambi Effect	8.7
83	777 Charlie	8.7
140	Kishkindha Kaandam	8.6
79	The Kashmir Files	8.6
437	Jersey	8.5
25	96	8.5
512	Sachin	8.5
350	Meiyazhagan	8.5
204	Kumbalangi Nights	8.5

In [78]:
```
#Bottom 10 movies by IMDb rating
df_boxoffice[["Title","IMDb_Rating"]].sort_values("IMDb_Rating",ascending=True).iloc[:10]
```

Out[78]:

	Title	IMDb_Rating
468	Race 3	1.9
491	Baaghi 3	2.2
571	Student of the Year 2	2.2
496	Heropanti 2	2.3
338	Chandramukhi 2	2.6
417	Liger	2.6
478	Gunday	2.7
213	Adipurush	2.7
565	Dabangg 3	3.0
495	A Flying Jatt	3.1

In [79]:
```
#5 longest run time movies
df_boxoffice[["Title","Runtime_(mins)"]].sort_values("Runtime_(mins)",ascending=False).iloc[:5]
```

Out[79]:

	Title	Runtime_(mins)
114	Animal	204
39	I	188
45	RRR (Rise Roar Revolt)	187
103	Avane Srimannarayana	186
261	Jilla	185

In [80]:
```
#5 shortest run time movies
df_boxoffice[["Title","Runtime_(mins)"]].sort_values('Runtime_(mins)',ascending=True).iloc[:5]
```

Out[80]:

	Title	Runtime_(mins)
599	Kill	105
497	Raksha Bandhan	108
560	Bhoot: Part One - The Haunted Ship	114
477	Hichki	116
407	The Ghazi Attack	116

In [81]:
```
#Top7 movies by world wide collection in Bollywood
df_boxoffice[df_boxoffice["Industry"] == "Bollywood"][["Title","Worldwide_Collection _in_Crores"]]\
                                                    .sort_values("Worldwide_Collection _in_Crores",ascending=False).iloc[
```

Out[81]:

	Title	Worldwide_Collection _in_Crores
524	Dangal	2122.30
572	Jawan	1160.00
573	Pathaan	1055.00
503	Bajrangi Bhaijaan	922.10
114	Animal	915.00
46	Secret Superstar	912.60
127	Stree 2: Sarkate Ka Aatank	857.07
460	PK	792.00
115	Gadar 2	686.00
27	Sultan	607.70

In [82]:
```
#Top7 movies by world wide collection in Tollywood
df_boxoffice[df_boxoffice["Industry"] == "Tollywood"][["Title","Worldwide_Collection _in_Crores"]]\
                                                    .sort_values("Worldwide_Collection _in_Crores",ascending=False).iloc[
```

Out[82]:

	Title	Worldwide_Collection _in_Crores
396	Bāhubali 2: The Conclusion	1788.00
45	RRR (Rise Roar Revolt)	1230.00
126	Kalki 2898-AD	1042.25
397	Bāhubali: The Beginning	650.00
212	Salaar: Part 1 - Ceasefire	617.75
200	Saaho	451.00
455	Devara: Part 1	421.60

In [83]:
```
#Top7 movies by world wide collection in Kollywood
df_boxoffice[df_boxoffice["Industry"] == "Kollywood"][["Title","Worldwide_Collection _in_Crores"]]\
                                    .sort_values("Worldwide_Collection _in_Crores",ascending=False).iloc[:7]
```

Out[83]:

	Title	Worldwide_Collection _in_Crores
285	2	701.00
116	Leo	605.90
117	Jailer	604.50
77	Ponniyin Selvan: Part One	488.36
343	The GOAT	451.23
74	Vikram	414.43
333	Ponniyin Selvan: Part Two	344.63

In [84]:
```
#Top7 movies by world wide collection in Sandalwood
df_boxoffice[df_boxoffice["Industry"] == "Sandalwood"][["Title","Worldwide_Collection _in_Crores"]]\
                                    .sort_values("Worldwide_Collection _in_Crores",ascending=False).iloc[:10]
```

Out[84]:

	Title	Worldwide_Collection _in_Crores
149	K.G.F: Chapter 2	1215.00
186	Kantara	407.82
148	K.G.F: Chapter 1	238.00
83	777 Charlie	102.75
187	VR (Vikrant Rona)	100.35
188	James	94.20
201	Kurukshetra	81.00
225	Kaatera	80.50
103	Avane Srimannarayana	75.30
144	Raajakumara	75.00

In [85]:
```
#Top7 movies by world wide collection in Mollywood
df_boxoffice[df_boxoffice["Industry"] == "Mollywood"][["Title","Worldwide_Collection _in_Crores"]]\
                                    .sort_values("Worldwide_Collection _in_Crores",ascending=False).iloc[:10]
```

Out[85]:

	Title	Worldwide_Collection _in_Crores
245	Manjummel Boys	241.10
217	2018	180.03
129	The Goat Life	157.35
130	Aavesham	154.79
177	Pulimurugan	150.00
132	Premalu	131.18
98	Lucifer	129.20
136	A.R.M	102.00
138	Guruvayoor Ambalanadayil	90.15
15	Bheeshmaparvam	88.20

In [86]:
```
#write query to display industry and verdict wise films count
df_boxoffice.groupby(["Industry","Verdict"])[["Verdict"]].count().rename(columns={"Verdict":"Count"})
```

Out[86]:

		Count
Industry	**Verdict**	
Bollywood	**Above Average**	7
	All Time Blockbuster	5
	Average	18
	Below Average	14
	Blockbuster	30
	Disaster	25
	Flop	22
	Hit	39
	Super Hit	22
Kollywood	**Above Average**	5
	Average	8
	Below Average	19
	Blockbuster	44
	Disaster	13
	Flop	12
	Hit	16
	Super Hit	23
Mollywood	**All Time Blockbuster**	4
	Blockbuster	46
	Disaster	5
	Flop	2
	Hit	10
	Super Hit	17
Sandalwood	**Above Average**	2
	All Time Blockbuster	5
	Average	3
	Below Average	4
	Blockbuster	7
	Disaster	1
	Hit	6
	Super Hit	4
Tollywood	**Above Average**	9
	All Time Blockbuster	2
	Average	13
	Below Average	10
	Blockbuster	39
	Disaster	24
	Flop	21
	Hit	25
	Super Hit	23

In [87]:
```
#Write query to get films based on budget in Bollowood
df_boxoffice[df_boxoffice["Industry"] == "Bollywood"][["Title","Budget_in_Crores"]]
```

Out[87]:

	Title	Budget_in_Crores
0	Sanju	100.0
1	Simmba	125.0
10	Tanu Weds Manu Returns	40.0
16	Ek Villain	40.0
26	Padmaavat	190.0
...	...	...
599	Kill	20.0
600	Madgaon Express	30.0
601	Swatantrya Veer Savarkar	25.0
602	Laapataa Ladies	8.0
603	Merry Christmas	50.0

182 rows × 2 columns

In [88]:
```
#Write query to get films based on budget in Tollywood
df_boxoffice[df_boxoffice["Industry"] == "Tollywood"][["Title","Budget_in_Crores"]]
```

Out[88]:

	Title	Budget_in_Crores
5	Srimanthudu	70.0
11	Geetha Govindam	8.5
12	Mahanubhavudu	15.0
20	Race Gurram	50.0
21	Nenu Local	20.0
...	...	...
454	Das Ka Dhamki	15.0
455	Devara: Part 1	250.0
456	Guntur Kaaram	150.0
457	Saripodhaa Sanivaaram	60.0
458	Naa Saami Ranga	30.0

166 rows × 2 columns

In [89]:
```
#Write query to get films based on budget in Kollywood
df_boxoffice[df_boxoffice["Industry"] == "Kollywood"][["Title","Budget_in_Crores"]]
```

Out[89]:

	Title	Budget_in_Crores
2	Janatha Garage	50.0
3	Kaththi	70.0
6	Kadaikutty Singam	25.0
7	Irumbu Thirai	20.0
9	Veeram	45.0
...	...	...
348	Thangalaan	80.0
349	Garudan	20.0
350	Meiyazhagan	35.0
351	Sarfira	85.0
352	Lal Salaam	60.0

140 rows × 2 columns

In [90]:
```
#Write query to get films based on budget in Sandalwood
df_boxoffice[df_boxoffice["Industry"] == "Sandalwood"][["Title","Budget_in_Crores"]]
```

Out[90]:

	Title	Budget_in_Crores
4	Maanikya	15.0
13	Ugramm	4.0
36	Tagaru	10.0
57	Hebbuli	20.0
83	777 Charlie	15.0
103	Avane Srimannarayana	25.0
108	Yajamana	20.0
144	Raajakumara	20.0
145	Yuvarathnaa	25.0
148	K.G.F: Chapter 1	50.0
149	K.G.F: Chapter 2	100.0
155	The Villain	45.0
156	Mr. And Mrs. Ramchari	6.0
169	Anjaniputra	20.0
171	Chakravarthy	20.0
181	Roberrt	30.0
182	Pogaru	25.0
183	Kotigobba 3	25.0
185	Salaga	10.0
186	Kantara	16.0
187	VR (Vikrant Rona)	80.0
188	James	40.0
201	Kurukshetra	60.0
202	Pailwaan	35.0
203	Natasaarvabhowma	20.0
207	Odeya	15.0
225	Kaatera	45.0
228	Kranti	35.0
230	Kabzaa	70.0
235	Ghost	20.0
237	Sapta Sagaradaache Ello: Side A	20.0
255	Bheema	8.0

In [91]:
```
#Write query to get films based on budget in Mollywood
df_boxoffice[df_boxoffice["Industry"] == "Mollywood"][["Title","Budget_in_Crores"]]
```

Out[91]:

	Title	Budget_in_Crores
8	Abrahaminte Santhathikal	10.0
15	Bheeshmaparvam	18.0
18	Njan Prakashan	8.0
23	Aadhi	7.0
30	Varathan	3.5
...	...	...
246	Turbo	35.0
247	Bramayugam	30.0
248	Abraham Ozler	15.0
252	Malaikottai Vaaliban	60.0
256	Nunakkuzhi	8.0

84 rows × 2 columns

In [92]:
```
#Top 5 movies by IMDb rating from Bollowood
df_boxoffice[df_boxoffice["Industry"] == "Bollywood"][["Title","IMDb_Rating"]].sort_values("IMDb_Rating",ascending=False).
```

Out[92]:

	Title	IMDb_Rating
121	12th Fail	8.8
79	The Kashmir Files	8.6
512	Sachin	8.5
602	Laapataa Ladies	8.4
524	Dangal	8.3

In [93]:
```
#Top 5 movies by IMDb rating from  Kollywood
df_boxoffice[df_boxoffice["Industry"] == "Kollywood"][["Title","IMDb_Rating"]].sort_values("IMDb_Rating",ascending=False).
```

Out[93]:

	Title	IMDb_Rating
316	Rocketry: The Nambi Effect	8.7
350	Meiyazhagan	8.5
135	Maharaja	8.5
25	96	8.5
274	Vada Chennai	8.4

In [94]:
```
#Top 5 movies by IMDb rating from Tollywood
df_boxoffice.query('Industry == "Tollywood"')[["Title","IMDb_Rating"]].sort_values("IMDb_Rating",ascending=False).iloc[:5]
```

Out[94]:

	Title	IMDb_Rating
437	Jersey	8.5
84	Sita Ramam	8.5
35	Mahanati	8.4
360	Rangasthalam	8.2
226	Hi Nanna	8.2

In [95]:
```
#Top 5 movies by IMDb rating from Sandalwood
df_boxoffice.query('Industry == "Sandalwood"')[["Title","IMDb_Rating"]].sort_values("IMDb_Rating",ascending=False).iloc[:5
```

Out[95]:

	Title	IMDb_Rating
83	777 Charlie	8.7
148	K.G.F: Chapter 1	8.2
149	K.G.F: Chapter 2	8.2
237	Sapta Sagaradaache Ello: Side A	8.2
186	Kantara	8.2

In [96]:
```
#Top 5 movies by IMDb rating from Mollywood
df_boxoffice.query('Industry == "Mollywood"')[["Title","IMDb_Rating"]].sort_values("IMDb_Rating",ascending=False).iloc[:5]
```

Out[96]:

	Title	IMDb_Rating
140	Kishkindha Kaandam	8.6
204	Kumbalangi Nights	8.5
189	Jana Gana Mana	8.3
217	2018	8.3
147	Bangalore Days	8.3

In [97]:
```
#Write a query to get language wise budget?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby(["Language"])[["Budget_in_Crores"]].sum().sort_valu
```

Out[97]:

Language	Budget_in_Crores
Hindi	14535.0
Telugu	9144.5
Tamil	8682.0
Malayalam	1342.5
Kannada	949.0

In [98]:
```
#Write a query to get language wise how many directors are there?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby(["Language"])[["DirectorID"]].nunique()
```

Out[98]:

Language	DirectorID
Hindi	121
Kannada	26
Malayalam	67
Tamil	87
Telugu	101

In [99]:
```
#Write a query to get language wise worldwide total collection ?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby(["Language"])[["Worldwide_Collection _in_Crores"]].
                                                              .sort_values("Worldwide_Collection _in_Crores",ascending=Fal
```

Out[99]:

Language	Worldwide_Collection _in_Crores
Hindi	38610.79
Telugu	18065.27
Tamil	15235.99
Malayalam	4597.32
Kannada	3369.42

In [100…
```
#Write a query to get language, lead actor/actress wise films they acted?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby(["Language","Lead_Actor/Actress"])[["FilmID"]].cour
                                                              .sort_values("FilmID",ascending=False)
```

Out[100…

Language	Lead_Actor/Actress	FilmID
Hindi	Akshay Kumar	20
	Ajay Devgn	14
Tamil	Dhanush	13
Malayalam	Mammootty	13
Telugu	Nani	12
...	...	...
Tamil	Premgi Amaren	1
	Pradeep Ranganathan	1
Kannada	Rishab Shetty	1
Tamil	N.T. Rama Rao Jr.	1
Telugu	amantha Ruth Prabhu	1

192 rows × 1 columns

In [101…

```
#Write a query to get language, year wise films released?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby(["Language","Year"])[["FilmID"]].count().rename(col
```

Out[101…

Language	Year	No. of Films Released
Hindi	**2014**	14
	2015	10
	2016	19
	2017	24
	2018	15
	2019	17
	2020	12
	2021	4
	2022	24
	2023	23
	2024	20
Kannada	**2014**	3
	2017	4
	2018	3
	2019	6
	2021	5
	2022	5
	2023	5
	2024	1
Malayalam	**2014**	1
	2015	6
	2016	5
	2017	8
	2018	8
	2019	13
	2020	6
	2021	3
	2022	12
	2023	8
	2024	14
Tamil	**2014**	8
	2015	8
	2016	9
	2017	12
	2018	22
	2019	19
	2020	3
	2021	8
	2022	17
	2023	18
	2024	16
Telugu	**2014**	6
	2015	8
	2016	8
	2017	24
	2018	23

		No. of Films Released
Language	**Year**	
	2019	19
	2020	6
	2021	11
	2022	22
	2023	25
	2024	14

In [102…
```
#Write a query to get films which was not released on overseas?
df_boxoffice.query('Overseas_Collection_in_Crores == 0')[["Title"]]
```

Out[102…

	Title
4	Maanikya
13	Ugramm
36	Tagaru
57	Hebbuli
169	Anjaniputra
171	Chakravarthy
181	Roberrt
182	Pogaru
183	Kotigobba 3
185	Salaga
225	Kaatera
228	Kranti
237	Sapta Sagaradaache Ello: Side A
255	Bheema
340	DD Returns
446	118

In [103…
```
#Write a query to get language wise 3 longest runtime moves?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby("Language").apply(lambda x: x.nlargest(3,"Runtime_(
  .reset_index(drop=True)[["Language","Title","Runtime_(mins)"]]
```

```
C:\Users\supar\AppData\Local\Temp\ipykernel_18288\4127993966.py:2: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the
grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explic
itly select the grouping columns after groupby to silence this warning.
```

Out[103…

	Language	Title	Runtime_(mins)
0	Hindi	Animal	204
1	Hindi	M.S. Dhoni: The Untold Story	184
2	Hindi	Maidaan	181
3	Kannada	Avane Srimannarayana	186
4	Kannada	Kurukshetra	185
5	Kannada	Kaatera	183
6	Malayalam	Marakkar: Lion of the Arabian Sea	181
7	Malayalam	Ayyappanum Koshiyum	177
8	Malayalam	Lucifer	175
9	Tamil	I	188
10	Tamil	Jilla	185
11	Tamil	Cobra	183
12	Telugu	RRR (Rise Roar Revolt)	187
13	Telugu	Arjun Reddy	182
14	Telugu	Kalki 2898-AD	180

In [104…

```
#Write a query to get language wise 5 shortest runtime moves?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby(["Language"]).apply(lambda x: x.nsmallest(5,"Runtim
  .reset_index(drop=True)[["Language","Title","Runtime_(mins)"]]
```

```
C:\Users\supar\AppData\Local\Temp\ipykernel_18288\142857738.py:2: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the
grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explic
itly select the grouping columns after groupby to silence this warning.
```

Out[104…

	Language	Title	Runtime_(mins)
0	Hindi	Kill	105
1	Hindi	Raksha Bandhan	108
2	Hindi	Bhoot: Part One - The Haunted Ship	114
3	Hindi	Hichki	116
4	Hindi	Bareilly Ki Barfi	116
5	Kannada	Salaga	124
6	Kannada	Tagaru	129
7	Kannada	Ugramm	132
8	Kannada	Ghost	132
9	Kannada	Kabzaa	134
10	Malayalam	Malikappuram	121
11	Malayalam	Sudani from Nigeria	123
12	Malayalam	Vaazha: Biopic of a Billion Boys	125
13	Malayalam	Uyare	125
14	Malayalam	Nunakkuzhi	125
15	Tamil	Kochadaiiyaan	118
16	Tamil	Dhilluku Dhuddu 2	119
17	Tamil	Kadaram Kondan	121
18	Tamil	Naane Varuvean	122
19	Tamil	Badhaai ho	124
20	Telugu	The Ghazi Attack	116
21	Telugu	Guru	116
22	Telugu	Raju Gari Gadhi 2	117
23	Telugu	Nela Ticket	120
24	Telugu	118	120

In [105…

```
#Write a query to get language wise top 5 films based first day collections?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby(["Language"])\
  .apply(lambda x: x.nlargest(5,"First_Day_Collection_Worldwide_in_Crores"))\
  .reset_index(drop=True)[["Language","Title","First_Day_Collection_Worldwide_in_Crores"]]
```

```
C:\Users\supar\AppData\Local\Temp\ipykernel_18288\1664424904.py:3: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the
grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explic
itly select the grouping columns after groupby to silence this warning.
```

Out[105…

	Language	Title	First_Day_Collection_Worldwide_in_Crores
0	Hindi	Jawan	129.10
1	Hindi	Animal	116.00
2	Hindi	Pathaan	104.80
3	Hindi	Tiger 3	94.00
4	Hindi	Stree 2: Sarkate Ka Aatank	80.00
5	Kannada	K.G.F: Chapter 2	159.00
6	Kannada	VR (Vikrant Rona)	25.50
7	Kannada	K.G.F: Chapter 1	25.00
8	Kannada	James	23.50
9	Kannada	Roberrt	16.90
10	Malayalam	Marakkar: Lion of the Arabian Sea	20.30
11	Malayalam	Odiyan	18.30
12	Malayalam	Kurup	18.30
13	Malayalam	The Goat Life	16.22
14	Malayalam	Turbo	15.70
15	Tamil	Leo	142.70
16	Tamil	2	105.00
17	Tamil	Maharaja	104.84
18	Tamil	The GOAT	100.75
19	Tamil	Jailer	96.60
20	Telugu	RRR (Rise Roar Revolt)	223.00
21	Telugu	Bāhubali 2: The Conclusion	217.00
22	Telugu	Kalki 2898-AD	177.70
23	Telugu	Salaar: Part 1 - Ceasefire	158.10
24	Telugu	Devara: Part 1	142.00

In [106…

```
#Write a query to get language wise top 5 films based India gross collections?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby(["Language"]).apply(lambda x: x.nlargest(5,"India_G
  .reset_index(drop=True)[["Language","Title","India_Gross_Collection_in_Crores"]]
```

```
C:\Users\supar\AppData\Local\Temp\ipykernel_18288\2480763724.py:2: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the
grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explic
itly select the grouping columns after groupby to silence this warning.
```

Out[106…

	Language	Title	India_Gross_Collection_in_Crores
0	Hindi	Jawan	760.00
1	Hindi	Stree 2: Sarkate Ka Aatank	713.07
2	Hindi	Animal	660.00
3	Hindi	Pathaan	657.50
4	Hindi	Gadar 2	620.50
5	Kannada	K.G.F: Chapter 2	1000.85
6	Kannada	Kantara	363.82
7	Kannada	K.G.F: Chapter 1	228.00
8	Kannada	777 Charlie	96.95
9	Kannada	VR (Vikrant Rona)	94.75
10	Malayalam	Manjummel Boys	167.65
11	Malayalam	2018	110.53
12	Malayalam	Pulimurugan	105.90
13	Malayalam	Aavesham	98.79
14	Malayalam	The Goat Life	97.85
15	Tamil	2	529.00
16	Tamil	Jailer	408.50
17	Tamil	Leo	401.90
18	Tamil	Ponniyin Selvan: Part One	313.36
19	Tamil	The GOAT	295.13
20	Telugu	Bāhubali 2: The Conclusion	1417.00
21	Telugu	RRR (Rise Roar Revolt)	915.85
22	Telugu	Kalki 2898-AD	767.25
23	Telugu	Bāhubali: The Beginning	516.00
24	Telugu	Salaar: Part 1 - Ceasefire	487.75

In [107…

```
#Write a query to get language, Director wise films count?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").merge(df_director,left_on="DirectorID",right_on="Director_I
  .groupby(["Language","Director"])[["FilmID"]].count().reset_index().rename(columns={"FilmID":"Films Count"})
```

Out[107…

	Language	Director	Films Count
0	Hindi	A.R. Murugadoss	1
1	Hindi	Aanand L. Rai	3
2	Hindi	Abhinay Deo	1
3	Hindi	Abhishek Pathak	1
4	Hindi	Abhishek Varman	2
...	...	...	...
397	Telugu	Vikram K. Kumar	3
398	Telugu	Vikram Sirikonda	1
399	Telugu	Vimal Krishna	1
400	Telugu	Vishwak Sen	1
401	Telugu	Vivek Athreya	2

402 rows × 3 columns

In [108…

```
#Write a query to get language wise OTT platofrm wise films count?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby(["Language","OTT_Platform"])[["FilmID"]].count()
```

Out[108…

Language	OTT_Platform	FilmID
Hindi	Aha	1
	Amazon Prime Video	78
	Disney+ Hotstar	29
	JioCinema	3
	Netflix	52
	SonyLIV	1
	ZEE5	18
Kannada	Amazon Prime Video	27
	Disney+ Hotstar	1
	SonyLIV	1
	Voot	1
	ZEE5	2
Malayalam	Aha	2
	Amazon Prime Video	60
	Disney+ Hotstar	7
	Netflix	5
	Sun NXT	7
	ZEE5	3
Tamil	Aha	1
	Amazon Prime Video	100
	Disney+ Hotstar	9
	Netflix	19
	SonyLIV	2
	Sun NXT	4
	ZEE5	5
Telugu	Aha	4
	Amazon Prime Video	119
	Disney+ Hotstar	8
	Google Play Movies	1
	Netflix	23
	ZEE5	11

In [109…

```
#What are top 10 fimns based language and first day collection?
pd.merge(df_language,df_boxoffice,on="LanguageID",how="inner").groupby(["Language"])\
  .apply(lambda x: x.nlargest(10,"First_Day_Collection_Worldwide_in_Crores")).reset_index(drop=True)[["Language","Title",'
```

```
C:\Users\supar\AppData\Local\Temp\ipykernel_18288\3736190320.py:3: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the
grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explic
itly select the grouping columns after groupby to silence this warning.
```

Out[109…

	Language	Title	First_Day_Collection_Worldwide_in_Crores
0	Hindi	Jawan	129.10
1	Hindi	Animal	116.00
2	Hindi	Pathaan	104.80
3	Hindi	Tiger 3	94.00
4	Hindi	Stree 2: Sarkate Ka Aatank	80.00
5	Hindi	Thugs of Hindostan	76.00
6	Hindi	War	74.30
7	Hindi	Sultan	71.30
8	Hindi	Brahmastra Part One: Shiva	70.00
9	Hindi	Prem Ratan Dhan Payo	66.20
10	Kannada	K.G.F: Chapter 2	159.00
11	Kannada	VR (Vikrant Rona)	25.50
12	Kannada	K.G.F: Chapter 1	25.00
13	Kannada	James	23.50
14	Kannada	Roberrt	16.90
15	Kannada	Kaatera	13.00
16	Kannada	Kabzaa	12.50
17	Kannada	The Villain	12.00
18	Kannada	Kranti	11.60
19	Kannada	Pogaru	10.25
20	Malayalam	Marakkar: Lion of the Arabian Sea	20.30
21	Malayalam	Odiyan	18.30
22	Malayalam	Kurup	18.30
23	Malayalam	The Goat Life	16.22
24	Malayalam	Turbo	15.70
25	Malayalam	King of Kotha	15.00
26	Malayalam	Lucifer	14.00
27	Malayalam	Bheeshmaparvam	12.20
28	Malayalam	Malaikottai Vaaliban	12.15
29	Malayalam	Aavesham	10.50
30	Tamil	Leo	142.70
31	Tamil	2	105.00
32	Tamil	Maharaja	104.84
33	Tamil	The GOAT	100.75
34	Tamil	Jailer	96.60
35	Tamil	Kabali	90.00
36	Tamil	Beast	82.40
37	Tamil	Ponniyin Selvan: Part One	80.70
38	Tamil	Sarkar	71.00
39	Tamil	Vettaiyan	67.00
40	Telugu	RRR (Rise Roar Revolt)	223.00
41	Telugu	Bāhubali 2: The Conclusion	217.00
42	Telugu	Kalki 2898-AD	177.70
43	Telugu	Salaar: Part 1 - Ceasefire	158.10
44	Telugu	Devara: Part 1	142.00
45	Telugu	Adipurush	127.50
46	Telugu	Sye Raa Narasimha Reddy	85.00

	Language	Title	First_Day_Collection_Worldwide_in_Crores
47	Telugu	Bāhubali: The Beginning	73.00
48	Telugu	Guntur Kaaram	68.70
49	Telugu	Sarkaru Vaari Paata	68.50

In [110…
```
#Write a query to get director wise number of fims released in from year 2017 to 2019
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID").query('Year in [2017,2018,2019]')\
  .groupby(["Director"])[["FilmID"]].count().sort_values("FilmID",ascending=False)
```

Out[110…

Director	FilmID
Maruthi Dasari	3
A.R. Murugadoss	2
H. Vinoth	2
Boyapati Srinu	2
Atlee	2
...	...
Laxman Utekar	1
Lokesh Kanagaraj	1
Luv Ranjan	1
M. Padmakumar	1
Zoya Akhtar	1

190 rows × 1 columns

In [111…
```
#Write a query to get director wise world wide collections?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID").groupby(["Director"])[["Worldwide_Collectic
  .sort_values("Worldwide_Collection _in_Crores",ascending=False)
```

Out[111…

Director	Worldwide_Collection _in_Crores
S.S. Rajamouli	3668.00
Siddharth Anand	2229.45
Nitesh Tiwari	2122.30
Prashanth Neel	2101.75
Atlee	1865.35
...	...
Krishna Chaitanya	20.25
B. Unnikrishnan	20.15
Kranthi Madhav, Gaurav Mahaur	20.00
Vikram Sirikonda	20.00
R Balki	20.00

384 rows × 1 columns

In [112…
```
#Write a query to get director wise first day world wide collections?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID").groupby("Director")[["First_Day_Collection_
  .sort_values("First_Day_Collection_Worldwide_in_Crores",ascending=False)
```

Out[112...

Director	First_Day_Collection_Worldwide_in_Crores
S.S. Rajamouli	513.00
Prashanth Neel	344.10
Koratala Siva	334.00
Trivikram Srinivas	276.30
Atlee	270.20
...	...
R.S. Vimal	0.81
Vishnu Sasi Shankar	0.52
Dinjith Ayyathan	0.50
Ram Kumar	0.50
Venu Yeldandi	0.50

384 rows × 1 columns

In [113...

```
#Write a query to get director wise India gross collections?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID").groupby(["Director"])[["India_Gross_Collect
  .sort_values("India_Gross_Collection_in_Crores",ascending=False)
```

Out[113...

Director	India_Gross_Collection_in_Crores
S.S. Rajamouli	2848.85
Prashanth Neel	1747.60
Siddharth Anand	1560.91
Atlee	1256.05
Rohit Shetty	1246.20
...	...
Manu Ashokan	15.90
Aashiq Abu	15.80
Shree Karthick	14.80
R Balki	14.00
B. Unnikrishnan	13.15

384 rows × 1 columns

In [114...

```
#Write a query to get director wise overseas collections?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID").groupby("Director")[["Overseas_Collection_i
  .sort_values("Overseas_Collection_in_Crores",ascending=False)
```

Out[114...

Director	Overseas_Collection_in_Crores
Nitesh Tiwari	1535.30
Advait Chandan	892.40
Siddharth Anand	819.37
S.S. Rajamouli	819.15
Rajkumar Hirani	635.00
...	...
Chinthan	0.00
S. Prem Anand	0.00
Duniya Vijay	0.00
Shiva Karthik	0.00
Hemanth M. Rao	0.00

384 rows × 1 columns

In [115...

```
#Write a query to get director, lead actor/actress and number of films?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID").groupby(["Director","Lead_Actor/Actress"])[
  .rename(columns={"FilmID":"Count"})
```

Out[115...

Director	Lead_Actor/Actress	Count
A.R. Murugadoss	**Akshay Kumar**	1
	Joseph Vijay	2
	Mahesh Babu	1
	Rajinikanth	1
Aanand L. Rai	**Akshay Kumar**	1
...	**...**	...
Vysakh	**Mammootty**	2
	Mohanlal	1
Zakariya	**Soubin Shahir**	1
Zoya Akhtar	**Anil Kapoor**	1
	Ranveer Singh	1

559 rows × 1 columns

In [116...

```
#Write a query to get films which is having budget on between 150 crores and 277 crores?
df_boxoffice[(df_boxoffice["Budget_in_Crores"] >= 150) & (df_boxoffice["Budget_in_Crores"] <= 277)][["Title","Budget_in_Cr
```

Out[116…

	Title	Budget_in_Crores
26	Padmaavat	190.0
47	Tiger Zinda Hai	200.0
77	Ponniyin Selvan: Part One	250.0
91	Tanhaji: The Unsung Warrior	150.0
114	Animal	150.0
116	Leo	250.0
117	Jailer	200.0
199	War	175.0
214	Varisu	200.0
260	Beast	150.0
268	Darbar	200.0
277	Annaatthe	180.0
301	Valimai	150.0
333	Ponniyin Selvan: Part Two	250.0
344	Vettaiyan	200.0
345	Indian 2	250.0
371	Pushpa: The Rise - Part 1	150.0
396	Bāhubali 2: The Conclusion	250.0
397	Bāhubali: The Beginning	180.0
455	Devara: Part 1	250.0
456	Guntur Kaaram	150.0
461	Happy New Year	150.0
463	Bang Bang	160.0
468	Race 3	150.0
469	Dilwale	165.0
473	Zero	200.0
474	Sooryavanshi	180.0
484	Thugs of Hindostan	275.0
493	Gangubai Kathiawadi	180.0
498	Laal Singh Chaddha	180.0
519	Prem Ratan Dhan Payo	180.0
541	83	200.0
543	Bellbottom	160.0
546	Vikram Vedha	150.0
547	Samrat Prithviraj	220.0
548	Bachchhan Paandey	165.0
549	Shamshera	150.0
573	Pathaan	250.0
574	Tiger 3	250.0
576	Rocky Aur Rani Kii Prem Kahaani	150.0
586	Fighter	250.0
591	Bade Miyan Chote Miyan	250.0

In [117…

```
#Write a query to get director, week name wise films released?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").groupby(["Director","Week_Name"
  .sort_values("FilmID",ascending=False).rename(columns={"FilmID":"Film Count"})
```

Out[117…

Director	Week_Name	Film Count
Rohit Shetty	Friday	6
Siva	Friday	4
Mohit Suri	Friday	4
Surender Reddy	Friday	3
Prashanth Neel	Friday	3
...	...	...
Maruthi Dasari	Saturday	1
Arun Matheswaran	Friday	1
Martin Prakkat	Thursday	1
Mari Selvaraj	Thursday	1
Anil Sharma	Friday	1

474 rows × 1 columns

In [118…
```
#Write a query to get OTT platofrm and director wise films released?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").groupby(["OTT_Platform","Direct
```

Out[118…

OTT_Platform	Director	FilmID
Aha	Gopichand Malineni	1
	Jeethu Joseph	1
	Mallik Ram	1
	Neeraj Pandey	1
	Prabhu	1
...	...	...
ZEE5	Trinadha Rao Nakkina	1
	Venky Atluri	2
	Vetrimaaran	1
	Vidyadhar Kagita	1
	Vivek Agnihotri	1

473 rows × 1 columns

In [119…
```
#Write a query to get director wise films released on Friday only?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").query('Week_Name in "Friday"')\
  .groupby(["Director","Week_Name"])[["FilmID"]].count().rename(columns={"FilmID":"Film Count"})
```

Out[119…

Director	Week_Name	Film Count
A.R. Murugadoss	Friday	2
Aanand L. Rai	Friday	2
Aashiq Abu	Friday	1
Abhinay Deo	Friday	1
Abhishek Nama	Friday	1
...	...	...
Vivek Agnihotri	Friday	1
Vivek Athreya	Friday	1
Vysakh	Friday	2
Zakariya	Friday	1
Zoya Akhtar	Friday	2

311 rows × 1 columns

In [120…
```
#Write a query to get films based on IMDb rating between 6.5 and 7.7?
df_boxoffice.query('IMDb_Rating  >= 6.5 and IMDb_Rating <= 7.7')[["Title","IMDb_Rating"]]
```

Out[120…

	Title	IMDb_Rating
0	Sanju	7.6
2	Janatha Garage	7.2
4	Maanikya	6.5
5	Srimanthudu	7.5
6	Kadaikutty Singam	6.8
...	...	...
596	Khel Khel Mein	6.7
599	Kill	7.6
600	Madgaon Express	7.0
601	Swatantrya Veer Savarkar	7.7
603	Merry Christmas	6.9

235 rows × 2 columns

In [121…
```
#Write a query to get director,films and IMDb ratings?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner")[["Director","Title","IMDb_Ratir
```

Out[121…

	Director	Title	IMDb_Rating
0	Rajkumar Hirani	Sanju	7.6
1	Rajkumar Hirani	PK	8.1
2	Rajkumar Hirani	Dunki	6.5
3	Farah Khan	Happy New Year	5.0
4	Sajid Nadiadwala	Kick	6.0
...	...	...	...
599	Sriram Raghavan	Merry Christmas	6.9
600	Elan	Star	6.5
601	Vidyadhar Kagita	Gaami	6.5
602	Vi Anand	Ooru Peru Bhairavakona	5.6
603	Krishna Chaitanya	Gangs of Godavari	5.0

604 rows × 3 columns

In [122…
```
#Write a query to get films with highest budget based flop verdict? #budget greater that 100cr
df_boxoffice[["Title","Budget_in_Crores","Verdict"]].sort_values("Budget_in_Crores",ascending=False).query('(Verdict in "F
```

Out[122…

	Title	Budget_in_Crores	Verdict
484	Thugs of Hindostan	275.0	Flop
541	83	200.0	Flop
277	Annaatthe	180.0	Flop
468	Race 3	150.0	Flop
546	Vikram Vedha	150.0	Flop
301	Valimai	150.0	Flop
280	Vivegam	120.0	Flop
366	Spyder	120.0	Flop
162	Street Dancer 3D	100.0	Flop
449	Bro	100.0	Flop
403	Godfather	100.0	Flop
491	Baaghi 3	100.0	Flop
502	Tubelight	100.0	Flop

In [123…
```
#Write a query to get total number of directors?
dire=df_director["Director_ID"].count()
print("The totla directors are present in this dataset is :",dire)
```

```
The totla directors are present in this dataset is : 384
```

In [124…
```
#Write a query to get vedridct wise total films released?
df_boxoffice.groupby(["Verdict"])[["FilmID"]].count().sort_values("FilmID",ascending=False).rename(columns={"FilmID":"Film
```

Out[124…

Verdict	Film Count
Blockbuster	166
Hit	96
Super Hit	89
Disaster	68
Flop	57
Below Average	47
Average	42
Above Average	23
All Time Blockbuster	16

In [125…
```
#Write a query to get top 10 directors based number of films?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").groupby(["Director"])[["FilmID"
  .sort_values("FilmID",ascending=False).iloc[:10].rename(columns={"FilmID":"Films Count"})
```

Out[125…

Director	Films Count
Trivikram Srinivas	6
Rohit Shetty	6
A.R. Murugadoss	5
Siva	5
Anil Ravipudi	5
Boyapati Srinu	5
Koratala Siva	5
Remo D'Souza	4
Atlee	4
Siddharth Anand	4

In [126…
```
#Write a query to get top 5 directors based on world wide collections and also industry wise?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").groupby(["Industry","Director"]
  .sum().sort_values(["Industry","Worldwide_Collection _in_Crores"],ascending=[True,False]).groupby("Industry").head(5)
```

Out[126...

Industry	Director	Worldwide_Collection _in_Crores
Bollywood	Siddharth Anand	2229.45
	Nitesh Tiwari	2122.30
	Rajkumar Hirani	1834.50
	Rohit Shetty	1656.60
	Sandeep Reddy Vanga	1292.00
Kollywood	Lokesh Kanagaraj	1350.13
	S. Shankar	1089.83
	Nelson Dilipkumar	952.88
	Mani Ratnam, Sruti Harihara Subramanian	832.99
	Siva	727.90
Mollywood	Vysakh	300.65
	Chidambaram	241.10
	Jithu Madhavan	223.79
	Jude Anthany Joseph	180.03
	Girish A.D.	165.28
Sandalwood	Prashanth Neel	1484.00
	Rishab Shetty	407.82
	Santhosh Ananddram	166.70
	Tarun Sudhir	150.00
	Kiranraj K	102.75
Tollywood	S.S. Rajamouli	3668.00
	Nag Ashwin	1125.75
	Trivikram Srinivas	877.45
	Koratala Siva	838.60
	Sukumar	709.50

In [127...

```
#Write a query to get top 3 directors based on India gross collections in Bollywood industry?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").query('Industry in "Bollywood"'
  .groupby(["Director"])[["India_Gross_Collection_in_Crores"]].sum().sort_values("India_Gross_Collection_in_Crores",ascend
```

Out[127...

Director	India_Gross_Collection_in_Crores
Siddharth Anand	1560.91
Rohit Shetty	1246.20
Rajkumar Hirani	1199.50

In [128...

```
#Write a query to get top 3 directors based on India gross collections in Tollywood industry?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").query('Industry in "Tollywood"'
  .groupby(["Director"])[["India_Gross_Collection_in_Crores"]].sum().sort_values("India_Gross_Collection_in_Crores",ascend
```

Out[128...

Director	India_Gross_Collection_in_Crores
S.S. Rajamouli	2848.85
Nag Ashwin	840.25
Trivikram Srinivas	711.65

In [129...

```
#Write a query to get top 3 directors based on India gross collections in Kollywood industry?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").query('Industry in "Kollywood"'
  .groupby(["Director"])[["India_Gross_Collection_in_Crores"]].sum().sort_values("India_Gross_Collection_in_Crores",ascend
```

Out[129…

Director	India_Gross_Collection_in_Crores
Lokesh Kanagaraj	956.88
S. Shankar	826.83
Nelson Dilipkumar	669.88

In [130…
```
#Write a query to get top 3 directors based on India gross collections in Mollywood industry?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").query('Industry in "Mollywood"'
  .groupby(["Director"])[["India_Gross_Collection_in_Crores"]].sum().sort_values("India_Gross_Collection_in_Crores",ascend
```

Out[130…

Director	India_Gross_Collection_in_Crores
Vysakh	208.55
Chidambaram	167.65
Jithu Madhavan	144.49

In [131…
```
#Write a query to get top 3 directors based on India gross collections in Sandalwood industry?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").query('Industry in "Sandalwood"'
  .groupby(["Director"])[["India_Gross_Collection_in_Crores"]].sum().sort_values("India_Gross_Collection_in_Crores",ascend
```

Out[131…

Director	India_Gross_Collection_in_Crores
Prashanth Neel	1259.85
Rishab Shetty	363.82
Santhosh Ananddram	164.40

In [132…
```
#Write to get total number of generes?
gen=df_genere["Genre"].count()
print("The total number of genre are :",gen)
```

```
The total number of genre are : 159
```

In [133…
```
#Write query to get director,language,genere films count?
pd.merge(df_director,df_boxoffice,left_on="Director_ID",right_on="DirectorID",how="inner").merge(df_language,on="LanguageI
  .merge(df_genere,on="GenreID",how="inner").groupby(["Director","Language","Genre"])[["FilmID"]].count()
```

Out[133…

Director	Language	Genre	FilmID
A.R. Murugadoss	**Hindi**	**Action, Crime, Thriller**	1
	Tamil	**Action, Crime, Thriller**	1
		Action, Drama	2
	Telugu	**Action, Crime, Thriller**	1
Aanand L. Rai	**Hindi**	**Comedy, Drama, Family**	1
...	**...**	**...**	...
Vysakh	**Malayalam**	**Action, Comedy, Drama**	1
		Action, Thriller	1
Zakariya	**Malayalam**	**Comedy, Sport**	1
Zoya Akhtar	**Hindi**	**Comedy, Drama, Romance**	1
		Drama, Music, Romance	1

566 rows × 1 columns

In [134…
```
#Write a query to genere wise budget?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").groupby(["Genre"])[["Budget_in_Crores"]].sum().sort_values("Budg
```

Out[134...

Genre	Budget_in_Crores
Action, Drama	3113.0
Action, Crime, Drama, Thriller	2937.0
Action, Crime, Thriller	1560.0
Action, Drama, Thriller	1457.5
Comedy, Drama, Romance	1308.5
...	...
Drama, Fantasy	6.0
Action, Comedy, Family, Romance	5.0
Biography, Drama, Family	4.0
Comedy, Sport	2.0
Action, Drama, History, Thriller, War	0.0

159 rows × 1 columns

In [135...
```
#Write a query to get genere wise first day worldwide collections?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").groupby(["Genre"])[["First_Day_Collection_Worldwide_in_Crores"]]
  .sort_values("First_Day_Collection_Worldwide_in_Crores",ascending=False)
```

Out[135...

Genre	First_Day_Collection_Worldwide_in_Crores
Action, Drama	1433.05
Action, Crime, Drama, Thriller	1396.49
Action, Drama, Thriller	532.86
Action, Crime, Thriller	452.20
Action, Adventure, Drama	425.32
...	...
Crime, Drama	1.95
Drama, Fantasy	1.60
Biography, Drama, Family	1.30
Action, Comedy, Family, Romance	1.20
Biography, Drama, Romance	0.81

159 rows × 1 columns

In [136...
```
#Write a query to get genere wise overseas collections?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").groupby(["Genre"])[["Overseas_Collection_in_Crores"]].sum()\
  .sort_values("Overseas_Collection_in_Crores",ascending=False)
```

Out[136...

Genre	Overseas_Collection_in_Crores
Action, Crime, Drama, Thriller	2043.13
Action, Drama	1598.18
Action, Biography, Drama, Sport	1535.30
Comedy, Drama	872.96
Drama, Music	846.40
...	...
Crime, Horror, Mystery, Thriller	2.20
Action, Drama, Sci-Fi, Thriller	1.30
Comedy, Drama, Horror	1.00
Action, Comedy, Musical	0.00
Action, Crime	0.00

159 rows × 1 columns

In [137...

```
#Write a query to get genere wise India gross collections?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").groupby(["Genre"])[["India_Gross_Collection_in_Crores"]].sum()\
  .sort_values("India_Gross_Collection_in_Crores",ascending=False)
```

Out[137...

Genre	India_Gross_Collection_in_Crores
Action, Crime, Drama, Thriller	6212.32
Action, Drama	5851.50
Comedy, Drama, Romance	2334.30
Action, Adventure, Thriller	2079.98
Action, Drama, Thriller	2044.35
...	...
Action, Crime, Drama, History	18.00
Crime, Thriller	17.50
Action, Drama, Fantasy, Thriller, Western	17.15
Drama, Sci-Fi, Thriller	14.80
Crime, Drama, Horror, Mystery, Thriller	14.00

159 rows × 1 columns

In [138...

```
#Write a query to get genere wise top 2 longest run time movies?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").groupby(["Genre"]).apply(lambda x: x.nlargest(2,"Runtime_(mins)"
  [["Genre","Title","Runtime_(mins)"]]
```

```
C:\Users\supar\AppData\Local\Temp\ipykernel_18288\3311827853.py:2: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the
grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explic
itly select the grouping columns after groupby to silence this warning.
```

Out[138…

	Genre	Title	Runtime_(mins)
0	Action	Kaatera	183
1	Action	Yajamana	164
2	Action, Adventure	Thugs of Hindostan	164
3	Action, Adventure	RDX: Robert Dony Xavier	151
4	Action, Adventure, Comedy	Total Dhamaal	130
...	...	...	...
233	Mystery	Vaazhai	134
234	Mystery	Nunakkuzhi	125
235	Mystery, Thriller	Kahaani 2	127
236	Romance	Love Story	155
237	Romance	Mr. Majnu	145

238 rows × 3 columns

In [139…

```
#Write a query to get genere wise 2 shortest runtime movies?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").groupby(["Genre"]).apply(lambda x: x.nsmallest(2,"Runtime_(mins)
[["Genre","Title","Runtime_(mins)"]]
```

```
C:\Users\supar\AppData\Local\Temp\ipykernel_18288\3485221982.py:2: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the
grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explic
itly select the grouping columns after groupby to silence this warning.
```

Out[139…

	Genre	Title	Runtime_(mins)
0	Action	Garudan	133
1	Action	Motta Shiva Ketta Shiva	151
2	Action, Adventure	RDX: Robert Dony Xavier	151
3	Action, Adventure	Thugs of Hindostan	164
4	Action, Adventure, Comedy	Dishoom	124
...	...	...	...
233	Mystery	Nunakkuzhi	125
234	Mystery	Vaazhai	134
235	Mystery, Thriller	Kahaani 2	127
236	Romance	18 Pages	135
237	Romance	Tholi Prema	137

238 rows × 3 columns

In [140…

```
#Write a query to get verdict, genere wise films released?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner")[["Verdict","Genre","Title"]]
```

Out[140…

	Verdict	Genre	Title
0	All Time Blockbuster	Comedy, Drama, Sci-Fi	PK
1	Hit	Action, Comedy, Crime, Drama, Music	Happy New Year
2	Blockbuster	Action, Comedy, Crime, Thriller	Jailer
3	Below Average	Action, Comedy, Crime, Thriller	Beast
4	Average	Action, Comedy, Crime, Thriller	Gaddalakonda Ganesh
…	…	…	…
599	Hit	Mystery	Nunakkuzhi
600	Disaster	Action, Drama, Fantasy, Thriller, Western	Malaikottai Vaaliban
601	Hit	Adventure, Drama, Fantasy, Mystery, Sci-Fi, Th…	Gaami
602	Hit	Fantasy, Thriller	Ooru Peru Bhairavakona
603	Disaster	Action, Crime, Drama, History	Gangs of Godavari

604 rows × 3 columns

In [141…

```
#Write a query to get genere, OTT platform wise films count?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").groupby(["Genre","OTT_Platform"])[["FilmID"]].count().reset_inde
```

Out[141…

	Genre	OTT_Platform	FilmID
0	Action	Amazon Prime Video	7
1	Action, Adventure	Amazon Prime Video	2
2	Action, Adventure, Comedy	Disney+ Hotstar	2
3	Action, Adventure, Comedy, Drama	Amazon Prime Video	1
4	Action, Adventure, Comedy, Drama	Disney+ Hotstar	1
…	…	…	…
259	Mystery	ZEE5	1
260	Mystery, Thriller	ZEE5	1
261	Romance	Amazon Prime Video	2
262	Romance	Netflix	1
263	Romance	ZEE5	1

264 rows × 3 columns

In [142…

```
#Write a query to get genere wise films count?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").groupby(["Genre"])[["FilmID"]].count().sort_values("FilmID",asce
  .rename(columns={"FilmID":"Film Count"})
```

Out[142…

	Film Count
Genre	
Action, Drama	44
Action, Crime, Drama, Thriller	36
Comedy, Drama, Romance	29
Action, Drama, Thriller	28
Action, Crime, Thriller	25
…	…
Biography, Drama, War	1
Biography, Drama, Romance	1
Biography, Drama, Music	1
Action, Comedy, Drama, Fantasy, Romance	1
Action, Horror, Mystery, Thriller	1

159 rows × 1 columns

In [143…

```
#Write a query to get genere wise films count  in Tollywood Industry?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").query('Industry in "Tollywood"').groupby(["Genre"])[["FilmID"]].
```

```
  .sort_values("FilmID",ascending=False).rename(columns={"FilmID":"Film Count"})
```

Out[143…

Genre	Film Count
Action, Drama	19
Action, Crime, Drama, Thriller	9
Comedy, Drama, Romance	8
Action, Comedy	8
Action, Drama, Thriller	8
...	...
Action, Horror, Thriller	1
Action, Mystery, Thriller	1
Action, Comedy, Crime, Drama, Romance	1
Action, Romance, Thriller	1
Action, Drama, Sport	1

65 rows × 1 columns

In [144…

```
#Write a query to get genere wise films count in Kollywood Industry?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").query('Industry in "Kollywood"').groupby(["Genre"])[["FilmID"]].
  .sort_values("FilmID",ascending=False).rename(columns={"FilmID":"Film Count"})
```

Out[144...

Genre	Film Count
Action, Drama	17
Action, Crime, Drama, Thriller	14
Action, Crime, Thriller	13
Action, Drama, Thriller	8
Action, Thriller	7
Action, Crime, Drama	6
Action, Comedy, Drama	5
Comedy, Horror	4
Action	3
Comedy, Drama	3
Action, Drama, Romance	3
Comedy	3
Action, Comedy, Drama, Romance	3
Comedy, Drama, Romance	3
Action, Comedy, Drama, Horror	2
Action, Comedy, Crime, Thriller	2
Action, Comedy, Crime, Drama, Thriller	2
Drama	2
Action, Crime, Drama, Mystery, Thriller	2
Action, Sci-Fi, Thriller	2
Action, Comedy	2
Action, Adventure, Sci-Fi, Thriller	2
Action, Adventure, Drama	2
Horror	2
Action, Drama, Sport	2
Drama, Romance	1
Horror, Thriller	1
Comedy, Crime, Drama, Thriller	1
Drama, Family, Romance	1
Comedy, Drama, Horror	1
Comedy, Romance	1
Drama, Family	1
Comedy, Drama, Musical, Romance	1
Comedy, Crime, Drama	1
Action, Drama, Romance, Sci-Fi, Thriller	1
Biography, Drama	1
Animation, Action, Adventure, History	1
Action, Romance, Thriller	1
Action, Horror, Thriller	1
Action, Drama, Sci-Fi, Thriller	1
Action, Comedy, Horror	1
Action, Comedy, Fantasy, Horror	1
Action, Comedy, Drama, Musical, Sci-Fi, Thriller	1
Action, Comedy, Drama, Fantasy, Thriller	1
Action, Comedy, Crime, Drama, Sci-Fi	1
Action, Comedy, Crime	1

Genre	Film Count
Action, Adventure, Sci-Fi	1
Action, Adventure, Fantasy	1
Action, Adventure, Drama, Thriller	1
Action, Adventure, Drama, History	1
Mystery	1

In [145…

```
#Write a query to get genere wise films count in Mollywood Industry?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").query('Industry in "Mollywood"').groupby(["Genre"])[["FilmID"]].
  .sort_values("FilmID",ascending=False).rename(columns={"FilmID":"Film Count"})
```

Out[145...

Genre	Film Count
Action, Drama, Thriller	6
Comedy, Drama, Romance	6
Comedy, Drama	5
Drama, Thriller	4
Comedy, Romance	4
Action, Crime, Drama, Thriller	4
Comedy	3
Drama	2
Crime, Drama, Mystery, Thriller	2
Action, Thriller	2
Crime, Mystery, Thriller	2
Action, Drama, History, War	2
Comedy, Thriller	2
Action, Crime, Thriller	2
Comedy, Crime, Drama	2
Action, Comedy, Thriller	2
Action, Comedy, Drama	2
Action, Comedy	2
Horror, Mystery, Thriller	2
Comedy, Horror	1
Comedy, Sport	1
Action, Adventure	1
Crime, Thriller	1
Crime, Drama, Thriller	1
Drama, Fantasy	1
Drama, Horror, Thriller	1
Drama, Musical, Romance	1
Drama, Mystery, Thriller	1
Comedy, Family, Romance	1
Biography, Drama, Family	1
Comedy, Drama, Family	1
Action, Drama, Fantasy, Thriller, Western	1
Action, Adventure, Drama	1
Action, Adventure, Drama, Romance	1
Action, Comedy, Drama, Thriller	1
Action, Crime, Drama, Mystery, Thriller	1
Action, Drama	1
Action, Drama, Fantasy, Thriller	1
Action, Drama, History	1
Biography, Drama, Romance	1
Action, Mystery, Thriller	1
Adventure, Drama	1
Adventure, Drama, Romance, Thriller	1
Adventure, Drama, Thriller	1
Biography, Crime, Drama, Thriller	1
Action, Adventure, Comedy, Drama	1

Genre	Film Count
Mystery	1

In [146…
```
#Write a query to get genere wise films count in Bollywood Industry?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").query('Industry in "Bollywood"').groupby(["Genre"])[["FilmID"]].
  .sort_values("FilmID",ascending=False).rename(columns={"FilmID":"Film Count"})
```

Out[146…

Genre	Film Count
Comedy, Drama	12
Comedy, Drama, Romance	12
Comedy, Romance	8
Action, Adventure, Thriller	7
Biography, Drama	6
…	…
Action, Drama, History	1
Action, Crime, Romance, Thriller	1
Action, Crime, Drama, Mystery, Thriller	1
Action, Comedy, Horror	1
Mystery, Thriller	1

93 rows × 1 columns

In [147…
```
#Write a query to get genere wise films count in Sandalwood Industry?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").query('Industry in "Sandalwood"').groupby("Genre")[["FilmID"]].c
  .sort_values("FilmID",ascending=False).rename(columns={"FilmID":"Film Count"})
```

Out[147…

Genre	Film Count
Action, Drama	5
Action	4
Action, Crime	3
Action, Crime, Drama, Thriller	3
Action, Crime, Thriller	2
Action, Drama, Romance	2
Action, Thriller	2
Action, Adventure, Drama, Thriller	1
Action, Comedy, Drama	1
Action, Comedy, Musical	1
Action, Crime, Drama, Mystery, Thriller	1
Action, Adventure, Comedy, Drama, Fantasy	1
Action, Drama, Sport	1
Action, Drama, Thriller	1
Action, Romance	1
Adventure, Comedy, Drama	1
Drama, Romance	1
History, War	1

In [148…
```
#Write a query to get lead actors/actress wise,genere and films count?
pd.merge(df_genere,df_boxoffice,on="GenreID",how="inner").groupby(["Lead_Actor/Actress"])[["Genre","FilmID"]].nunique()\
  .rename(columns={"Genre":"Genre Count","FilmID":"Film Count"})
```

Out[148…

Lead_Actor/Actress	Genre Count	Film Count
Aamir Khan	2	2
Adah Sharma	1	1
Aditya Roy Kapoor	1	1
Adivi Sesh	3	3
Ajay Devgn	13	14
...	...	...
Vishwak Sen	3	3
Yami Gautam	1	1
Yash	2	3
Zaira Wasim	1	1
amantha Ruth Prabhu	1	1

183 rows × 2 columns

In [149…
```
#Write a query to get 5th rank movie based on Worldwide total collections?
df_boxoffice[["Title","Worldwide_Collection _in_Crores"]].sort_values("Worldwide_Collection _in_Crores",ascending=False).r
```

Out[149…
```
index                            573
Title                         Pathaan
Worldwide_Collection _in_Crores    1055.0
Name: 5, dtype: object
```

In [150…
```
#Write a query to get 5th rank movie by industry wise based on First day worldwide collections?
df_boxoffice.groupby(["Industry"]).apply(lambda x: x.sort_values("First_Day_Collection_Worldwide_in_Crores",ascending=Fals
            .iloc[4] if len(x) > 4 else None).dropna().reset_index(drop=True)[["Title","Industry","First_Day_Collection_Wo
```

```
C:\Users\supar\AppData\Local\Temp\ipykernel_18288\2165254264.py:2: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the
grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explic
itly select the grouping columns after groupby to silence this warning.
```

Out[150…

	Title	Industry	First_Day_Collection_Worldwide_in_Crores
0	Stree 2: Sarkate Ka Aatank	Bollywood	80.0
1	Jailer	Kollywood	96.6
2	Turbo	Mollywood	15.7
3	Roberrt	Sandalwood	16.9
4	Devara: Part 1	Tollywood	142.0

In [151…
```
#Write a query to get 3rd rank movie by industry wise based on IMDb Ratings?
df_boxoffice.groupby(["Industry"]).apply(lambda x: x.sort_values("IMDb_Rating",ascending=False)\
            .iloc[2] if len(x) > 2 else None).dropna().reset_index(drop=True)[["Industry","Title","IMDb_Rating"]]
```

```
C:\Users\supar\AppData\Local\Temp\ipykernel_18288\704882789.py:2: DeprecationWarning:

DataFrameGroupBy.apply operated on the grouping columns. This behavior is deprecated, and in a future version of pandas the
grouping columns will be excluded from the operation. Either pass `include_groups=False` to exclude the groupings or explic
itly select the grouping columns after groupby to silence this warning.
```

Out[151…

	Industry	Title	IMDb_Rating
0	Bollywood	Sachin	8.5
1	Kollywood	Maharaja	8.5
2	Mollywood	Jana Gana Mana	8.3
3	Sandalwood	K.G.F: Chapter 2	8.2
4	Tollywood	Mahanati	8.4

In [152…
```
#Calculate YoY% Budget growth?
df_boxoffice.groupby(["Year"])[["Budget_in_Crores"]].sum().assign(YoY_Budget_Growth=lambda x: x["Budget_in_Crores"].pct_ch
```

Out[152…

Year	Budget_in_Crores	YoY_Budget_Growth
2014	1820.0	NaN
2015	1906.0	4.725275
2016	2142.0	12.381952
2017	3291.0	53.641457
2018	3679.0	11.789730
2019	2544.0	-30.850775
2020	1387.0	-45.479560
2021	1855.0	33.741889
2022	5722.0	208.463612
2023	5877.0	2.708843
2024	4430.0	-24.621405

In [153…

```
#Calculate YoY% Worldwide total collelctions growth?
df_boxoffice.groupby(["Year"])[["Worldwide_Collection _in_Crores"]].sum()\
            .assign(YoY_Collection_Growth=lambda x: x["Worldwide_Collection _in_Crores"].pct_change()*100)
```

Out[153…

Year	Worldwide_Collection _in_Crores	YoY_Collection_Growth
2014	4645.00	NaN
2015	5470.20	17.765339
2016	7088.80	29.589412
2017	9659.20	36.260016
2018	8275.30	-14.327273
2019	8242.72	-0.393702
2020	2348.75	-71.505158
2021	2711.65	15.450772
2022	9853.16	263.364003
2023	13461.72	36.623378
2024	8122.29	-39.663802

In [154…

```
#Calculate YoY% Indian Gross colelctions growth?
df_boxoffice.groupby("Year")[["India_Gross_Collection_in_Crores"]].sum()\
            .assign(YoY_Growth=lambda x: x["India_Gross_Collection_in_Crores"].pct_change()*100)
```

Out[154…

Year	India_Gross_Collection_in_Crores	YoY_Growth
2014	3711.10	NaN
2015	3935.30	6.041335
2016	4459.90	13.330623
2017	6852.90	53.655912
2018	6327.40	-7.668286
2019	6655.58	5.186649
2020	1966.35	-70.455618
2021	2238.40	13.835279
2022	7716.21	244.719889
2023	10080.17	30.636284
2024	6120.79	-39.278901

In [ ]: