


Project_1

Analyzing IMDB Movie Dataset

The Objective of this project is to analyze the IMDb Movie Dataset to uncover trends in ratings, genres and Revenues and Analyzing Based on the given Questions ..

```
import pandas as pn
df = pn.read_csv("IMDBMovie.csv")
df
```

	ID int64 1 - 1000	Title object The Host 0.2% Guardians of... .. 0.1% 997 others 99.7%	Genre object Action 29.3% Drama 19.5% 11 others 51.2%	Director object Ridley Scott 0.8% M. Night Shy... .. 0.6% 642 others 98.6%	Year int64 2006 - 2016	Runtime_minutes i... 66 - 191	Rating float64 1.9 - 9.0	V
0	1	Guardians of the ...	Action	James Gunn	2014	121	8.1	
1	2	Prometheus	Adventure	Ridley Scott	2012	124	7	
2	3	Split	Horror	M. Night Shyamal...	2016	117	7.3	
3	4	Sing	Animation	Christophe Lour...	2016	108	7.2	
4	5	Suicide Squad	Action	David Ayer	2016	123	6.2	
5	6	The Great Wall	Action	Yimou Zhang	2016	103	6.1	
6	7	La La Land	Comedy	Damien Chazelle	2016	128	8.3	
7	8	Mindhorn	Comedy	Sean Foley	2016	89	6.4	
8	9	The Lost City of Z	Action	James Gray	2016	141	7.1	
9	10	Passengers	Adventure	Morten Tyldum	2016	116	7	

1000 rows, showing 10 per page << < Page 1 of 100 > >> 

TASK-1

```
#How many rows are there in the IMDB dataset?
df.shape[0]
```


1000

Code Commentary : df.shape[0] retrieves the number of rows present in the DataFrame named as df.

TASK-2

```
# What is the 75th percentile of rating in the IMDB dataset?
df.describe()
```

	ID float64	Year float64	Runtime_minutes f...	Rating float64	Votes float64	Revenue_millions f...	
cou...	1000	1000	1000	1000	1000	872	
me...	500.5	2012.783	113.172	6.7232	169808.255	82.95637615	
std	288.8194361	3.205961508	18.81090817	0.9454287893	188762.6475	103.2535405	
min	1	2006	66	1.9	61	0	
25%	250.75	2010	100	6.2	36309	13.27	
50%	500.5	2014	111	6.8	110799	47.985	
75%	750.25	2016	123	7.4	239909.75	113.715	
max	1000	2016	191	9	1791916	936.63	

8 rows, showing 10 per page << < Page 1 of 1 > >> 

Code Commentary : The function df.describe() provides summary statistics for numerical columns in the DataFrame df, including count, mean, standard deviation, minimum, quartiles, and maximum values.

TASK-3

```
# How many NA values are there in the field 'Revenue'?  
df['Revenue_millions'].isnull().sum()
```

128

Code Commentary : Using sum function calculates the total number of missing values in the 'Revenue_millions' column of the DataFrame df.

TASK-4

```
#How many movies have revenue higher than 75 million?  
(df['Revenue_millions'] > 75).sum()
```

318

Code Commentary : Count the Sum of occurrences where the 'Revenue_millions' column in DataFrame df has a value greater than 75.

TASK-5

```
# How many movies have revenue greater than 50 million but rating less than 7?  
df[(df['Revenue_millions'] > 50) & (df['Rating'] < 7)][['Title']].count()
```

211

Code Commentary : Counts the number of movies in DataFrame df Using count function that have a revenue greater than 50 million and a rating less than 7.

TASK-6

```
# What is the total revenue generated by movies in the year 2015?  
df[df['Year'] == 2015]['Revenue_millions'].sum()
```

8854.119999999999

Code Commentary : Calculates the total revenue generated by movies released in the year 2015 in the DataFrame df.

TASK-7

```
# What is the average rating for the genre adventure in the year 2015?  
df[(df['Genre'] == "Adventure") & (df['Year'] == 2015)][['Rating']].mean()
```

6.8

Code Commentary : Computes the average rating of adventure genre movies released in the year 2015 in the DataFrame df.

TASK-8

```
# What is the average duration of movies in rows 75 to 150? Please note that the rows in python start from 0.
```

```
df.iloc[75:151]["Runtime_minutes"].mean()  
#OR  
cond = df.iloc[75:151]  
cond ["Runtime_minutes"].mean()
```

```
127.47368421052632
```

Code Commentary : Using mean function Calculates the average runtime in minutes for the movies indexed from 75 to 150 in the DataFrame df.

TASK-9

```
# Which year generated the highest revenue?
```

```
df.groupby(by = "Year")["Revenue_millions"].sum().sort_values(ascending = False)
```

```
Year  
2016    11211.65  
2015     8854.12  
2014     7997.40  
2013     7666.72  
2012     6910.29  
2010     5989.65  
2011     5431.96  
2009     5292.26  
2008     5053.22  
2007     4306.23  
2006     3624.46  
Name: Revenue_millions, dtype: float64
```

Code Commentary : Here Using groupby function groups the DataFrame df by the "Year" column, calculates the sum of "Revenue_millions" for each year, and show the results in descending order based on the total revenue.

TASK-10

```
# What is the maximum revenue out of (10,20,30,40,50) rows?
```

```
df.iloc[10:60:10]["Revenue_millions"].max()
```

```
936.63
```

Code Commentary : By using iloc function it selects every 10th row from index 10 to index 59 in the DataFrame df, and then calculates the maximum value of the "Revenue_millions" column within this subset.

TASK-11

```
# How many movies with the genres 'Adventure', 'Action', 'Horror', and 'Crime' exist in the IMDB dataset?
```

```
df[df["Genre"].isin(["Adventure", "Action", "Horror", "Crime"])]["Title"].count()
```

```
485
```

Code Commentary : Using Count Function counts the number of movies in the DataFrame df that belong to the genres Adventure, Action, Horror, or Crime.

TASK-12

```
# Create a genre-level report with metrics average rating, the average number of votes, and average revenue.
# What is the average rating of the 'Horror' genre?
df[df["Genre"] == "Horror"]["Rating"].mean()
```

```
5.867391304347826
```

Code Commentary : Calculates the mean (average) rating of movies belonging to the Horror genre in the DataFrame df.

TASK-13

```
# How many movies has Billy Ray directed and find the year of release of those movies
cond = df[df["Director"] == "Billy Ray"]["Title"].count()
print("The no of movies has Billy Ray directed is",cond)
year = df[df["Director"] == "Billy Ray"]["Year"]
print("The year of movie that billy ray directed is",year)
```

```
The no of movies has Billy Ray directed is 1
The year of movie that billy ray directed is 995    2015
Name: Year, dtype: int64
```

Code Commentary : This code first counts the number of movies directed by "Billy Ray" and stores it in the variable cond. It then prints this count. Next, it retrieves the years of the movies directed by "Billy Ray" and prints them.

TASK-14

```
# How many movies were released in the year 2012 - 14. What type of genre were released the most
x = df[df["Year"].isin([2012,2013,2014])]["Title"].count()
print("The no of movies were released in the year 2012 - 14 is",x)
y= df[df["Year"].isin([2012,2013,2014])]["Genre"].min()
print("Type of Genre were released the most is",y)
```

```
The no of movies were released in the year 2012 - 14 is 253
Type of Genre were released the most is Action
```

Code Commentary : In 1st Line of code counts the number of movies released between 2012 and 2014 and i stored it in variable x, then prints it out. Subsequently, it determines the genre that appears first alphabetically among the movies released in that time frame, storing it in variable y, and prints out the most released genre.

TASK-15

```
# Which Movie had the highest vote and what genre it belongs to.
print(df[df["Votes"] == df["Votes"].max()]["Title"])
df[df["Votes"] == df["Votes"].max()]["Genre"]
```

```
54    The Dark Knight
Name: Title, dtype: object
```

```
54    Action
Name: Genre, dtype: object
```

Code Commentary : This code prints the titles of the movie(s) with the maximum number of votes and retrieves their corresponding genres.

TASK-16

```
# Find the Director whose movie grossed the highest. Also find the total revenue generated by that director
x = df[df["Revenue_millions"] == df["Revenue_millions"].max()][["Director"]]
print("The Director whose movie grossed the highest is",x)
y = df[df["Director"] == 'J.J. Abrams']['Revenue_millions'].sum()
print("The total revenue generated by director J.J. Abrams is",y)
```

```
The Director whose movie grossed the highest is 50    J.J. Abrams
Name: Director, dtype: object
The total revenue generated by director J.J. Abrams is 1683.4499999999998
```

Code Commentary : First I identifies the directors whose movies grossed the highest revenue and stores the director names in variable x, then prints it out. I got the answer as "j.j.Abrams" And I calculates the total revenue generated by the director "J.J. Abrams" and stores it in variable y, and prints out the total revenue.

TASK-17

```
#Create a report to showcase the revenue of each movie, as % revenue concerning the total revenue of the respective genre
#For example if a movie 'ABC' has genre 'Action' and released in 2015, then % revenue will be
#(Revenue of the movie 'ABC' *100)/ (Total revenue of the genre 'Action' in 2015)
```

```
#What is the % revenue of the movie 'Split' in its respective genre and year?
```

```
df['_%Revenue'] = (df['Revenue_millions'] / df.groupby(['Genre', 'Year'])['Revenue_millions'].transform('sum')) * 100
split_revenue = df[df['Title'] == 'Split']['_%Revenue'].values[0]
print(f'The % revenue of the movie 'Split' in its respective genre and year is : {split_revenue:}')
```

```
The % revenue of the movie 'Split' in its respective genre and year is : 29.42041024985622
```

Code Commentary : This code snippet calculates the percentage of revenue generated by each movie within its respective genre and year, assigns it to a new column named '%_Revenue', and then extracts the percentage revenue of the movie 'Split' using its respective genre and year. Finally, it prints out the percentage revenue of the movie 'Split'.