

▼ CHANDAN KUMAR

ID: GO_STP_13267

1. Load the important libraries

```
1 import pandas as pd
2 import numpy as np
```

2. View the dataset

```
1 imdb = pd.read_csv("/content/IMDB-Movie-Data.csv")
2 imdb.head()
```

	Rank	Title	Genre	Description	Director	Actor
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, V Diesel, Bradle Cooper, Zoe S.
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapaci Logan Marsha Green, Michael Fa.
2	3	Split	Horror,Thriller	Three girls are kidnapped by a man with a dia...	M. Night Shyamalan	James McAvoy, Any Taylor-Joy, Haley L Richar.

3. Understand some basic information about the dataset and Inspect the dataframe Inspect the dataframe's columns, shapes, variable types etc.

```
1 print("Columns present:\n", imdb.columns)
2 print()
3 print("Shape of dataset:\n", imdb.shape)
4 print()
5 print("Variable types:\n", imdb.dtypes)
6 print()
7 print("Basic Statistics:\n", imdb.describe())
```

```
Columns present:
Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
      'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
      'Metascore'],
      dtype='object')
```

```
Shape of dataset:
(1000, 12)
```

```
Variable types:
```

```
Rank          int64
Title         object
Genre         object
Description    object
Director      object
Actors        object
Year          int64
Runtime (Minutes)  int64
Rating        float64
Votes         int64
Revenue (Millions) float64
Metascore     float64
dtype: object
```

```
Basic Statistics:
```

	Rank	Year	...	Revenue (Millions)	Metascore
count	1000.000000	1000.000000	...	872.000000	936.000000
mean	500.500000	2012.783000	...	82.956376	58.985043
std	288.819436	3.205962	...	103.253540	17.194757
min	1.000000	2006.000000	...	0.000000	11.000000
25%	250.750000	2010.000000	...	13.270000	47.000000
50%	500.500000	2014.000000	...	47.985000	59.500000
75%	750.250000	2016.000000	...	113.715000	72.000000
max	1000.000000	2016.000000	...	936.630000	100.000000

```
[8 rows x 7 columns]
```

4. Data Selection – Indexing and Slicing data

```
1 imdb[0:11]
```

	Rank	Title	Genre	Description	Director	Ac
0	1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe Saldana
1	2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall Green, Michael Fassbender
2	3	Split	Horror,Thriller	Three girls are kidnapped by ...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richardson

```
1 imdb.iloc[0:10, [1, 5, 8]]
```

	Title	Actors	Rating
0	Guardians of the Galaxy	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	8.1
1	Prometheus	Noomi Rapace, Logan Marshall-Green, Michael Fa...	7.0
2	Split	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	7.3
3	Sing	Matthew McConaughey,Reese Witherspoon, Seth Ma...	7.2
4	Suicide Squad	Will Smith, Jared Leto, Margot Robbie, Viola D...	6.2
5	The Great Wall	Matt Damon, Tian Jing, Willem Dafoe, Andy Lau	6.1
6	La La Land	Ryan Gosling, Emma Stone, Rosemarie DeWitt, J....	8.3
7	Mindhorn	Essie Davis, Andrea Riseborough, Julian Barrat...	6.4
8	The Lost City of Z	Charlie Hunnam, Robert Pattinson, Sienna Mille...	7.1
9	Passengers	Jennifer Lawrence, Chris Pratt, Michael Sheen,...	7.0

```
1 imdb.iloc[4]
```

Rank	5
Title	Suicide Squad
Genre	Action,Adventure,Fantasy
Description	A secret government agency recruits some of th...
Director	David Ayer
Actors	Will Smith, Jared Leto, Margot Robbie, Viola D...
Year	2016
Runtime (Minutes)	123
Rating	6.2
Votes	393727
Revenue (Millions)	325.02
Metascore	40
Name: 4, dtype: object	

```
1 imdb.loc[0:5, ["Title", "Rating", "Votes", "Metascore"]]
```

	Title	Rating	Votes	Metascore
0	Guardians of the Galaxy	8.1	757074	76.0
1	Prometheus	7.0	485820	65.0
2	Split	7.3	157606	62.0
3	Sing	7.2	60545	59.0
4	Suicide Squad	6.2	393727	40.0
5	The Great Wall	6.1	56036	42.0

5. Data Selection – Based on Conditional filtering

```
1 top_rank = imdb[imdb["Rating"] > 8.0]["Title"].count()
2 print(top_rank)
```

59

6. Groupby operations

```
1 top_movie = imdb[imdb["Rating"] > 8.0]
2 top_movie.groupby(["Title"])["Votes"].mean()
```

Title	
12 Years a Slave	486338
3 Idiots	238789
Ah-ga-ssi	33418
Bahubali: The Beginning	76193
Dangal	48969
Django Unchained	1039115
El secreto de sus ojos	144524
Gone Girl	636243
Gran Torino	595779
Guardians of the Galaxy	757074
Hachi: A Dog's Tale	177602
Hacksaw Ridge	211760
Harry Potter and the Deathly Hallows: Part 2	590595
How to Train Your Dragon	523893
Incendies	92863
Inception	1583625
Inglourious Basterds	959065
Inside Out	416689
Interstellar	1047747
Into the Wild	459304
Jagten	192263
Kimi no na wa	34110
Koe no katachi	2421
La La Land	258682
Lion	102061
Mad Max: Fury Road	632842
Mommy	33560
No Country for Old Men	660286
PK	103279

Paint It Black	61
Pan's Labyrinth	498879
Prisoners	431185
Relatos salvajes	110100
Room	224132
Rush	339722
Shutter Island	855604
Spotlight	268282
Star Wars: Episode VII - The Force Awakens	661608
Taare Zameen Par	102697
The Avengers	1045588
The Bourne Ultimatum	525700
The Dark Knight	1791916
The Dark Knight Rises	1222645
The Departed	937414
The Grand Budapest Hotel	530881
The Help	342429
The Imitation Game	532353
The Intouchables	557965
The Lives of Others	278103
The Prestige	913152
The Wolf of Wall Street	865134
There Will Be Blood	400682
Toy Story 3	586669
Twin Peaks: The Missing Pieces	1973
Up	722203
WALL·E	776897
Warrior	355722
Whiplash	477276

7. Sorting operation

```
1 most_votes = imdb.groupby(["Votes"]).mean()
2 most_votes.sort_values(by = ["Votes"], ascending = False).head()
```

	Rank	Year	Runtime (Minutes)	Rating	Revenue (Millions)	Metascore
Votes						
1791916	55.0	2008.0	152.0	9.0	533.32	82.0
1583625	81.0	2010.0	148.0	8.8	292.57	74.0
1222645	125.0	2012.0	164.0	8.5	448.13	78.0
1047747	37.0	2014.0	169.0	8.6	187.99	74.0
1045588	77.0	2012.0	143.0	8.1	623.28	69.0

8. Dealing with missing values

```
1 imdb.isnull().sum()
```

Rank	0
Title	0
Genre	0
Description	0
Director	0

```

Actors          0
Year            0
Runtime (Minutes) 0
Rating          0
Votes           0
Revenue (Millions) 128
Metascore       64
dtype: int64

```

```
1 imdb.fillna(0).isnull().sum()
```

```

Rank            0
Title           0
Genre           0
Description      0
Director        0
Actors          0
Year            0
Runtime (Minutes) 0
Rating          0
Votes           0
Revenue (Millions) 0
Metascore       0
dtype: int64

```

10. Apply() functions

```

1 rank = imdb.apply(lambda n: n*5)
2 print(rank.head())

```

```

   Rank  ... Metascore
0     5  ...      380.0
1    10  ...      325.0
2    15  ...      310.0
3    20  ...      295.0
4    25  ...      200.0

```

```
[5 rows x 12 columns]
```

✓ 0s completed at 3:56 PM

