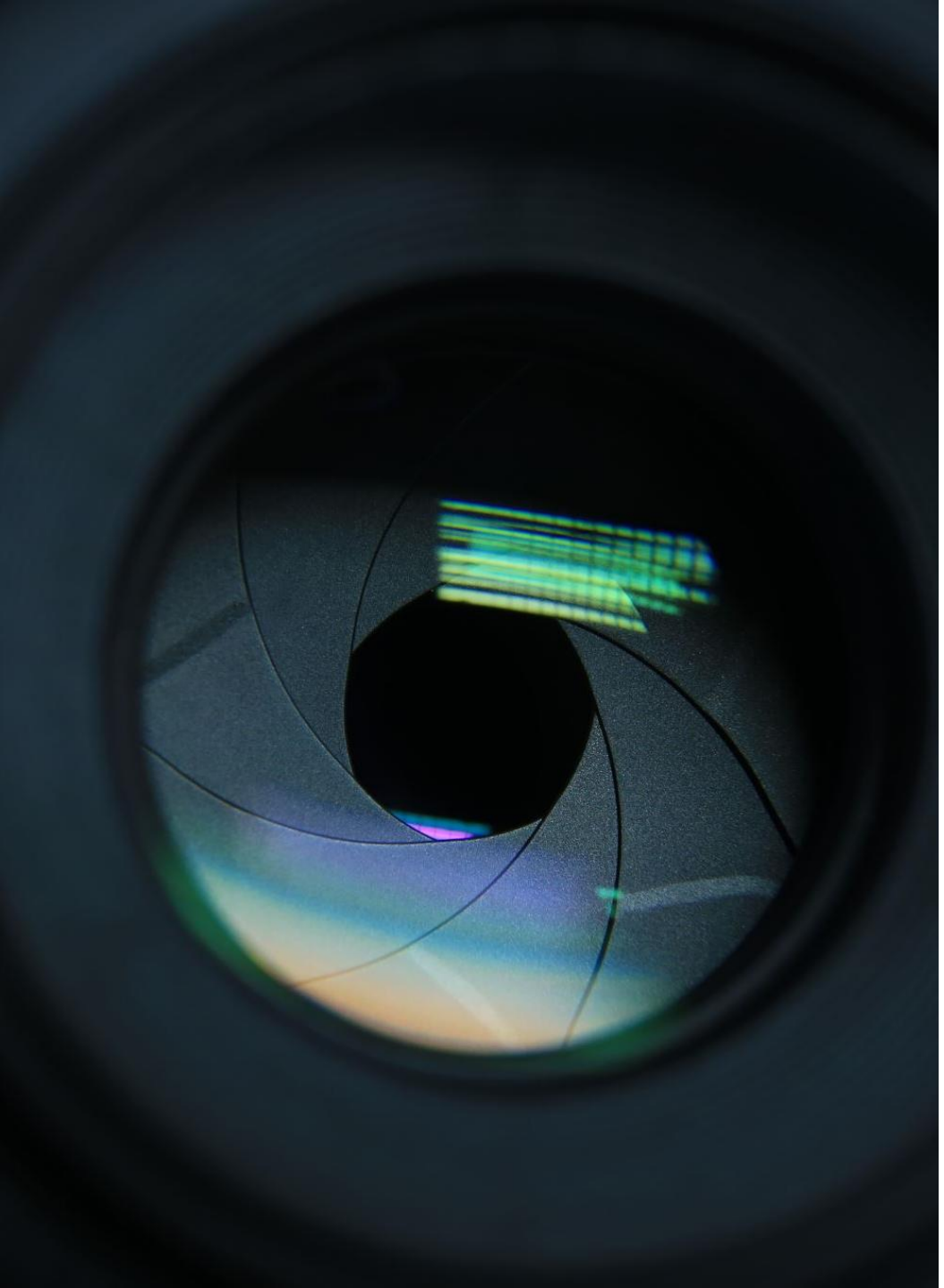
By

- Sachin B C

TMDB MOVIE DATA ANALYSIS

CAPSTONE PROJECT

CONTENTS



Project Objective

The objective of the project is to use Python programming to analyze a movie data to perform exploratory data analysis by answering the questions in the upcoming slides.

CHECKING DATATYPES

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   budget                4803 non-null   int64  
 1   genres                4803 non-null   object 
 2   homepage              1712 non-null   object 
 3   id                    4803 non-null   int64  
 4   keywords              4803 non-null   object 
 5   original_language     4803 non-null   object 
 6   original_title        4803 non-null   object 
 7   overview              4800 non-null   object 
 8   popularity            4803 non-null   float64
 9   production_companies  4803 non-null   object 
 10  production_countries  4803 non-null   object 
 11  release_date          4802 non-null   object 
 12  revenue               4803 non-null   int64  
 13  runtime               4801 non-null   float64
 14  spoken_languages      4803 non-null   object 
 15  status                4803 non-null   object 
 16  tagline               3959 non-null   object 
 17  title                 4803 non-null   object 
 18  vote_average          4803 non-null   float64
 19  vote_count            4803 non-null   int64  
dtypes: float64(3), int64(4), object(13)
memory usage: 750.6+ KB
```

CHECKING MISSING VALUES

```
In [5]: missing_values = df.isnull().sum()

        print("\nCount of Missing Values for Each Column:")
        print(missing_values)
```

```
Count of Missing Values for Each Column:
budget                     0
genres                     0
homepage                3091
id                         0
keywords                   0
original_language          0
original_title             0
overview                   3
popularity                 0
production_companies       0
production_countries       0
release_date               1
revenue                    0
runtime                    2
spoken_languages           0
status                     0
tagline                  844
title                      0
vote_average               0
vote_count                 0
dtype: int64
```

After checking missing vales of 'homepage' & 'tagline' these 2 columns will not affect the outcomes and it depends on to link

CHECKING DUPLICATES

In [8]:
```
df1 = df.drop(columns=['id'])

df_dup = df1[df1.duplicated(keep='first')]

print("\nDuplicate Rows based on all columns except 'id':")
df_dup
```

```
Duplicate Rows based on all columns except 'id':
```

Out[8]:

budget	genres	homepage	keywords	original_language	original_title	overview	popularity	production_companies	production_countries	release_date	reve

In [9]:
```
df1 = df.drop(columns=['id','genres','keywords','production_companies','homepage','overview','spoken_languages'])

df_dup = df1[df1.duplicated(keep='first')]

print("\nDuplicate Rows based on all columns except ,'genres','keywords','production_companies','homepage','overview
df_dup
```

```
Duplicate Rows based on all columns except ,'genres','keywords','production_companies','homepage','overview','spoke
n_languages':
```

Out[9]:

budget	original_language	original_title	popularity	production_countries	release_date	revenue	runtime	status	tagline	title	vote_average	vote_count

```
No dupicates are found
```

CLEANING DATA AS PER REQURED

Taking only genres names as required

In [10]:
```
def convert_genres(genres):
    if isinstance(genres, str):
        return eval(genres)
    return genres

def extract_genre_names(genres_list):
    return [genre['name'] for genre in genres_list] if isinstance(genres_list, list) else []


df['genres'] = df['genres'].apply(convert_genres)


df['genres'] = df['genres'].apply(extract_genre_names)


df['genres_name'] = df['genres'].apply(', '.join)
```

Taking only keywords name as required

In [11]:
```
def convert_keywords(keywords):
    if isinstance(keywords, str):
        return eval(keywords)
    return keywords

def extract_keyword_names(keywords_list):
    return [keyword['name'] for keyword in keywords_list] if isinstance(keywords_list, list) else []

df['keywords'] = df['keywords'].apply(convert_keywords)

df['keywords_name'] = df['keywords'].apply(extract_keyword_names)

df['keywords_name'] = df['keywords_name'].apply(', '.join)
```

DROPING UNWANTED COLUMNS

Droping unuanted columns

In [14]:
```
columns_to_drop = ['genres', 'keywords', 'production_companies', 'spoken_languages']
df.drop(columns=columns_to_drop, inplace=True)
```

In [15]:
```
df.columns
```

Out[15]:
```
Index(['budget', 'homepage', 'id', 'original_language', 'original_title',
       'overview', 'popularity', 'production_countries', 'release_date',
       'revenue', 'runtime', 'status', 'tagline', 'title', 'vote_average',
       'vote_count', 'genres_name', 'keywords_name',
       'production_companies_name', 'spoken_languages_name'],
      dtype='object')
```

Task 1

- Display the number of rows and columns in the dataset

  Number of rows: **4803**

  Number of columns: **20**

- Display titles and genres of the first 50 movies

```
Titles and Genres of the first 50 movies:
```

Out[17]:

	title	genres_name
0	Avatar	Action, Adventure, Fantasy, Science Fiction
1	Pirates of the Caribbean: At World's End	Adventure, Fantasy, Action
2	Spectre	Action, Adventure, Crime
3	The Dark Knight Rises	Action, Crime, Drama, Thriller
4	John Carter	Action, Adventure, Science Fiction
5	Spider-Man 3	Fantasy, Action, Adventure
6	Tangled	Animation, Family
7	Avengers: Age of Ultron	Action, Adventure, Science Fiction
8	Harry Potter and the Half-Blood Prince	Adventure, Fantasy, Family
9	Batman v Superman: Dawn of Justice	Action, Adventure, Fantasy
10	Superman Returns	Adventure, Fantasy, Action, Science Fiction
11	Quantum of Solace	Adventure, Action, Thriller, Crime
12	Pirates of the Caribbean: Dead Man's Chest	Adventure, Fantasy, Action
13	The Lone Ranger	Action, Adventure, Western
14	Man of Steel	Action, Adventure, Fantasy, Science Fiction
15	The Chronicles of Narnia: Prince Caspian	Adventure, Family, Fantasy
16	The Avengers	Science Fiction, Action, Adventure
17	Pirates of the Caribbean: On Stranger Tides	Adventure, Action, Fantasy
18	Men in Black 3	Action, Comedy, Science Fiction
19	The Hobbit: The Battle of the Five Armies	Action, Adventure, Fantasy

Task 2

- Identify the columns that have null values and perform the null value treatment

```
In [18]: print("\nColumns with null values:")
         df.isnull().sum()
```

```
Columns with null values:

Out[18]: budget                        0
         homepage                   3091
         id                            0
         original_language             0
         original_title                0
         overview                      3
         popularity                    0
         production_countries          0
         release_date                  1
         revenue                       0
         runtime                       2
         status                        0
         tagline                     844
         title                         0
         vote_average                  0
         vote_count                    0
         genres_name                   0
         keywords_name                 0
         production_companies_name     0
         spoken_languages_name         0
         dtype: int64
```

Task 3

- Display the movie categories that have a budget greater than $220,000

```
In [19]: print("\nMovie categories with budget greater than $220,000:")
mov_220000=df[df['budget'] > 220000]['genres_name']
mov_220000
```

```
Movie categories with budget greater than $220,000:

Out[19]: 0       Action, Adventure, Fantasy, Science Fiction
1                         Adventure, Fantasy, Action
2                           Action, Adventure, Crime
3                    Action, Crime, Drama, Thriller
4                 Action, Adventure, Science Fiction
                              ...
4680                         Crime, Horror, Thriller
4682                                          Horror
4720                                           Drama
4758                        Thriller, Science Fiction
4770                                    Drama, Comedy
Name: genres_name, Length: 3684, dtype: object
```

Task 4

- Display the movie categories where the revenue is greater than $961,000,000.

In [20]:
```
print("\nMovie categories with revenue greater than $961,000,000:")
grt_961=df[df['revenue'] > 961000000]['genres_name']
grt_961
```

```
Movie categories with revenue greater than $961,000,000:

0       Action, Adventure, Fantasy, Science Fiction
3                     Action, Crime, Drama, Thriller
7                 Action, Adventure, Science Fiction
12                        Adventure, Fantasy, Action
16                Science Fiction, Action, Adventure
17                        Adventure, Action, Fantasy
25                          Drama, Romance, Thriller
26                Adventure, Action, Science Fiction
28     Action, Adventure, Science Fiction, Thriller
29                       Action, Adventure, Thriller
31                Action, Adventure, Science Fiction
32                        Family, Fantasy, Adventure
36                Science Fiction, Action, Adventure
42                         Animation, Family, Comedy
44                                             Action
52                Action, Science Fiction, Adventure
65                    Drama, Action, Crime, Thriller
78                 Family, Adventure, Drama, Fantasy
98                        Adventure, Fantasy, Action
124                     Animation, Adventure, Family
197                       Adventure, Fantasy, Family
329                       Adventure, Fantasy, Action
506                        Animation, Comedy, Family
546             Family, Animation, Adventure, Comedy
Name: genres_name, dtype: object
```

Task 5

- Remove the rows with value 0 from both the budget and revenue columns.

```
In [21]: rev_val_0= df[(df['budget'] != 0) & (df['revenue'] != 0)]
         rev_val_0
```

Out[21]:

	budget	homepage	id	original_language	original_title	overview	popularity	production_countries	release
0	237000000	http://www.avatarmovie.com/	19995	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"iso_3166_1": "US", "name": "United States o...	10/
1	300000000	http://disney.go.com/disneypictures/pirates/	285	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[{"iso_3166_1": "US", "name": "United States o...	19/
2	245000000	http://www.sonypictures.com/movies/spectre/	206647	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	[{"iso_3166_1": "GB", "name": "United Kingdom"...	26/
3	250000000	http://www.thedarkknightrises.com/	49026	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	[{"iso_3166_1": "US", "name": "United States o...	16/
4	260000000	http://movies.disney.com/john-carter	49529	en	John Carter	John Carter is a war-weary, former military ca...	43.926995	[{"iso_3166_1": "US", "name": "United States o...	07/
...	...	...	...	...	...	...	...	...	
4773	27000	http://www.miramax.com/movie/clerks/	2292	en	Clerks	Convenience and video store clerks Dante and R...	19.748658	[{"iso_3166_1": "US", "name": "United States o...	13/

Task 6

- List the top 10 movies with the highest revenues and the top 10 movies with the least budget.

In [22]:
```
top_10_revenues = df.nlargest(10, 'revenue')[['title', 'revenue']]
print("Top 10 movies with the highest revenues:")
top_10_revenues
```

```
Top 10 movies with the highest revenues:
```

Out[22]:

	title	revenue
0	Avatar	2787965087
25	Titanic	1845034188
16	The Avengers	1519557910
28	Jurassic World	1513528810
44	Furious 7	1506249360
7	Avengers: Age of Ultron	1405403694
124	Frozen	1274219009
31	Iron Man 3	1215439994
546	Minions	1156730962
26	Captain America: Civil War	1153304495

In [23]:
```
top_10_least_budget = df.nsmallest(10, 'budget')[['title', 'budget']]
print("Top 10 movies with the least budget:")
top_10_least_budget
```

```
Top 10 movies with the least budget:
```

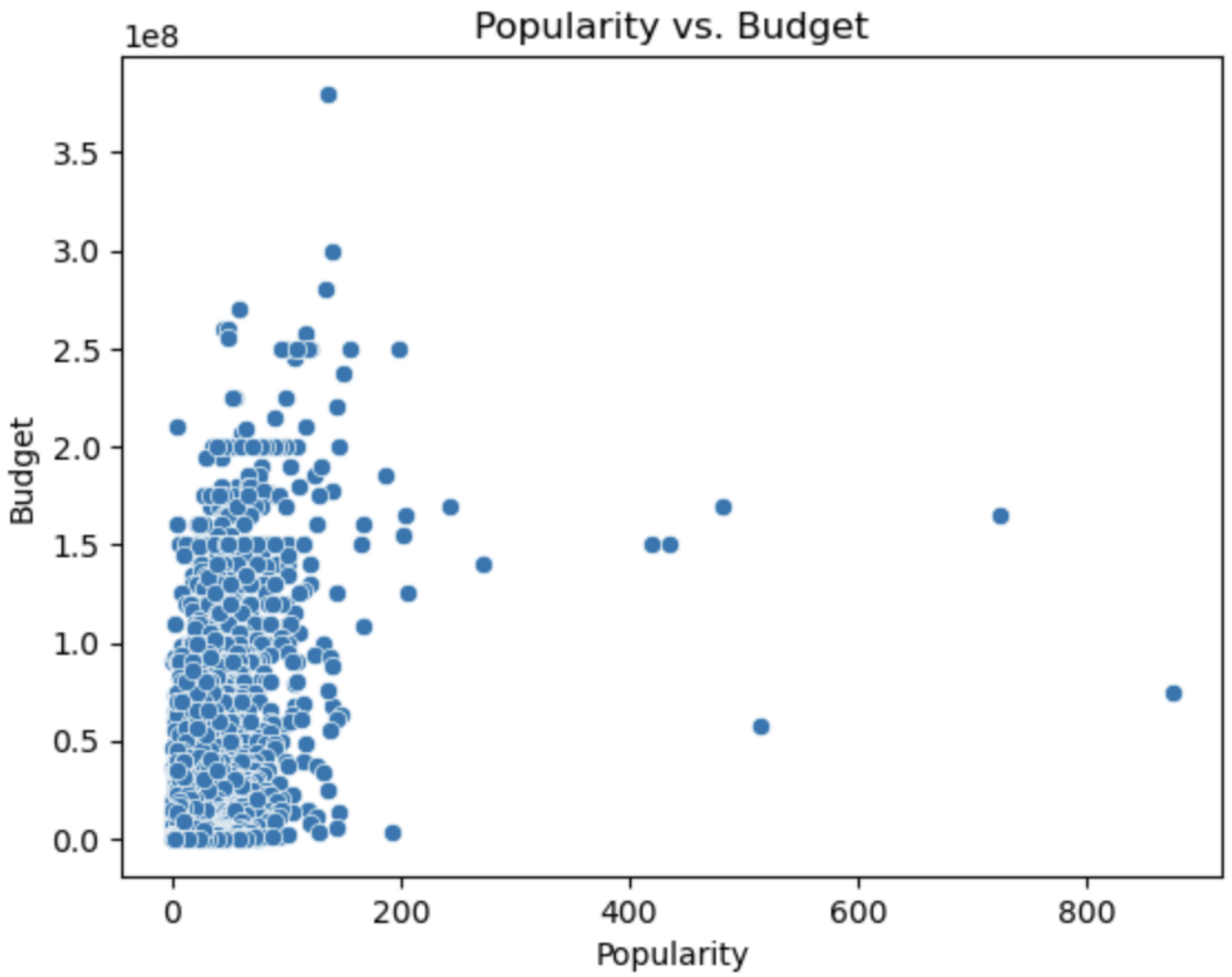
Out[23]:

	title	budget
265	The Cat in the Hat	0
321	The Campaign	0
359	Alvin and the Chipmunks: The Road Chip	0
406	Arthur Christmas	0
409	All That Jazz	0
453	The Pink Panther	0
463	Déjà Vu	0
474	Evolution	0
475	The Edge	0
489	Oceans	0

Task 7

- Are they correlated or totally uncorrelated with each other? Write the interpretation of your analysis.

- Correlation between popularity and budget: 0.5054139990665322



Task 8

- Identify and display the names of all production companies along with the number of times they appear in the dataset.

```
: production_companies_count = df['production_companies_name'].value_counts().sort_values(ascending=False)

print("\nProduction Companies and Their Frequency:")
print(production_companies_count)
```

```
Production Companies and Their Frequency:

51
Paramount Pictures
58
Universal Pictures
45
New Line Cinema
38
Columbia Pictures
37

...
Libido Cine, Aquafilms, Filmanova
1
Battleplan Productions, TF1 International, Moonstone Entertainment
1
Steeltown Entertainment, Point Park University, Shaderville
1
Euforia Film, FilmCamp, Miho Film, Barentsfilm, Yellow Bastard Production, News On Request (NOR), Zwart Arbeid
1
rusty bear entertainment, lucky crow films
1
Name: production_companies_name, Length: 3697, dtype: int64
```

Task 9

- Display the names of the top 25 production companies based on the number of movies they have produced in descending order of the number of movies produced.

```
: top_production_companies = df['production_companies_name'].value_counts().nlargest(25)

print("\nTop 25 Production Companies based on the number of movies produced:")
print(top_production_companies)
```

```
Top 25 Production Companies based on the number of movies produced:
                                                  351
Paramount Pictures                                 58
Universal Pictures                                 45
New Line Cinema                                    38
Columbia Pictures                                  37
Metro-Goldwyn-Mayer (MGM)                          32
Twentieth Century Fox Film Corporation             31
Warner Bros.                                       27
Walt Disney Pictures                               27
Touchstone Pictures                                26
Dimension Films                                    17
Columbia Pictures Corporation                      16
Miramax Films                                      16
DreamWorks Animation                               12
United Artists                                     12
Walt Disney Pictures, Pixar Animation Studios      11
Fox 2000 Pictures                                  10
Fox Searchlight Pictures                            9
Imagine Entertainment, Universal Pictures           9
Walt Disney Pictures, Walt Disney Feature Animation  9
Marvel Studios                                      8
Blue Sky Studios, Twentieth Century Fox Animation   8
Lions Gate Films                                    8
Hollywood Pictures, Cinergi Pictures Entertainment  7
United Artists, Eon Productions, Danjaq             7
Name: production_companies_name, dtype: int64
```

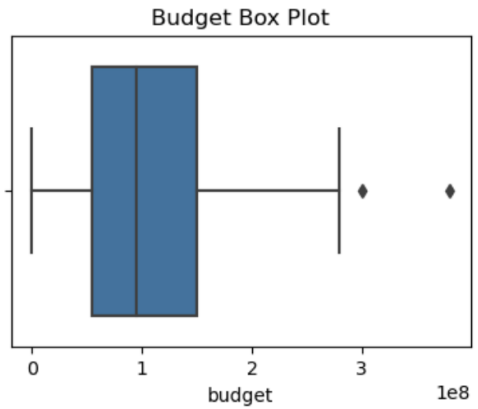
Task 10

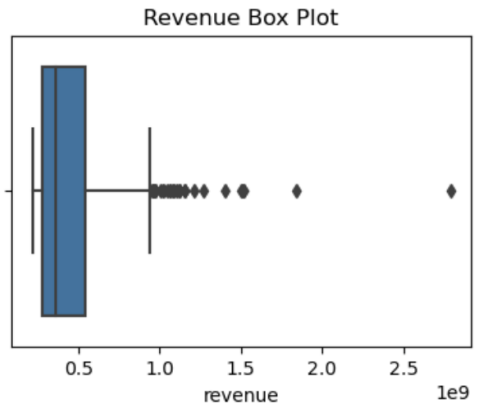
- Find the measures of central tendency for the following columns using the filtered data:
- 1. budget
- 2. revenue
- 3. runtime
- Perform outlier analysis for the above three columns using box plots

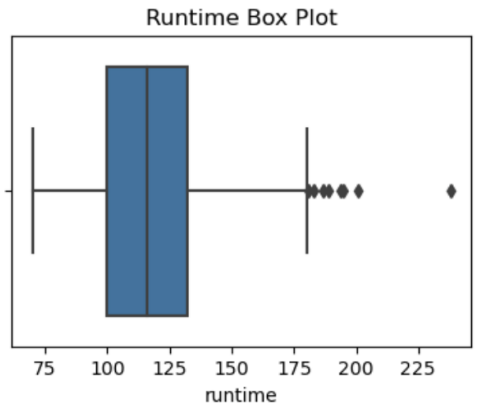
Measures of Central Tendency for Budget, Revenue, and Runtime:

:

	budget	revenue	runtime
count	5.000000e+02	5.000000e+02	500.00000
mean	1.028037e+08	4.587221e+08	118.62600
std	6.268914e+07	2.684133e+08	23.28378
min	0.000000e+00	2.190765e+08	70.00000
25%	5.500000e+07	2.814473e+08	100.00000
50%	9.500000e+07	3.630016e+08	116.00000
75%	1.500000e+08	5.471332e+08	132.00000
max	3.800000e+08	2.787965e+09	238.00000







Task 11

- Identify and display the name of the movies along with their run time for those movies that have above average runtime, using the data from the previous task

```
Movies with Above-Average Runtime:
```

	title	runtime
0	Avatar	162.0
25	Titanic	194.0
16	The Avengers	143.0
28	Jurassic World	124.0
44	Furious 7	137.0
...	...	...
1161	The Social Network	120.0
912	Interview with the Vampire	123.0
521	The Terminal	128.0
397	It's Complicated	121.0
1744	Knocked Up	129.0

232 rows × 2 columns

Summary

Dataset Overview:

- Explored a movie dataset with details on budget, genres, revenue, and more.

Data Cleaning:

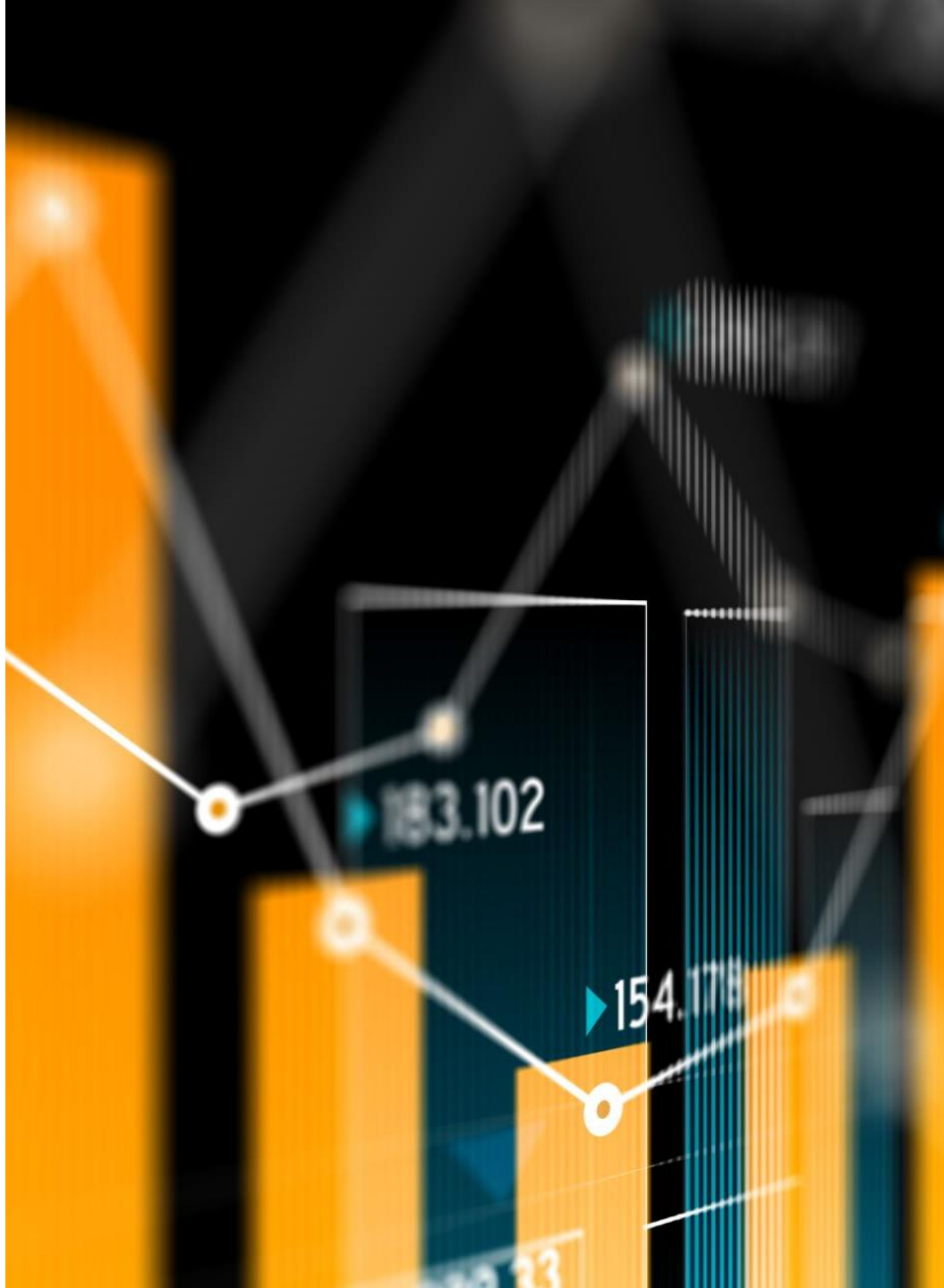
- Handled null values and imputed them using appropriate methods.
- Removed rows with budget and revenue values of 0.

Top Movies Analysis:

- Determined the top 10 movies based on revenue and the top 10 with the least budget.
- Explored central tendencies for budget, revenue, and runtime in the top 500 movies.

Correlation Analysis:

- Investigated the correlation between popularity and budget.
- Presented findings through scatter plots.



Summary

Production Companies Analysis:

- Identified and displayed production company frequencies.

- Listed the top 25 production companies based on the number of movies produced.

Above Average Runtime Movies:

- Calculated and identified movies with above-average runtimes.

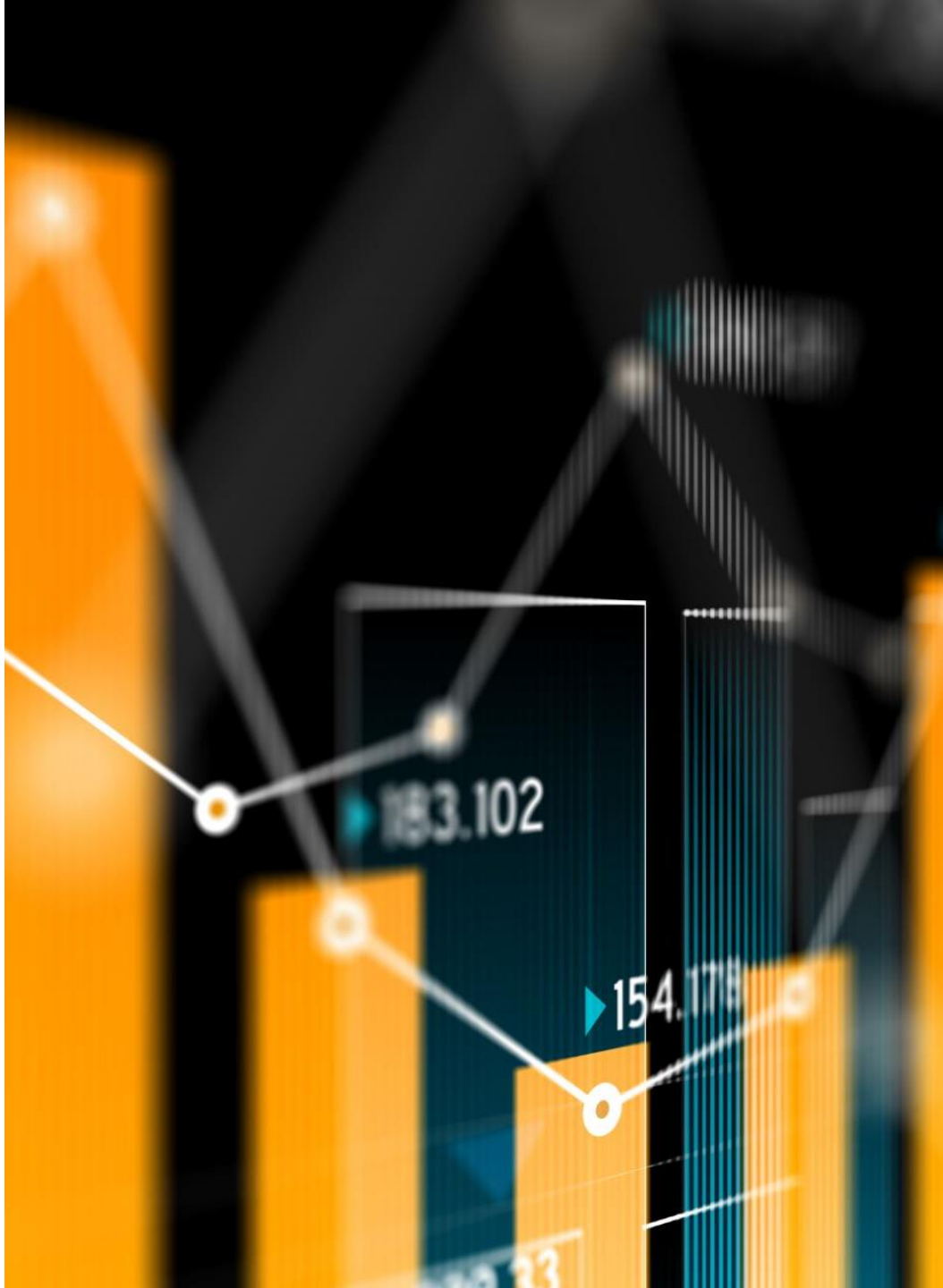
- Displayed names and runtimes of these movies.

Conclusion:

- Discovered insights into factors influencing movie success.

- Provided actionable recommendations for the production company.

- Enhanced understanding of audience preferences and industry dynamics for strategic decision-making.



Thank You