```
import pandas as pd

import matplotlib.pyplot as plt

movie = pd.read_csv(r"/content/movie_metadata.csv");

print("Head of the dataset:")
movie.head()
```

Head of the dataset:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	a
0	Color	James Cameron	723.0	178.0	0.0	
1	Color	Gore Verbinski	302.0	169.0	563.0	
2	Color	Sam Mendes	602.0	148.0	0.0	
3	Color	Christopher Nolan	813.0	164.0	22000.0	
4	NaN	Doug Walker	NaN	NaN	131.0	

5 rows × 28 columns

```
print("\nTail of the dataset:")
print(movie.tail())
```

```
Tail of the dataset:
      color     director_name  num_critic_for_reviews  duration  \
5038  Color       Scott Smith                     1.0      87.0
5039  Color               NaN                    43.0      43.0
5040  Color  Benjamin Roberds                    13.0      76.0
5041  Color       Daniel Hsia                    14.0     100.0
5042  Color          Jon Gunn                    43.0      90.0

      director_facebook_likes  actor_3_facebook_likes      actor_2_name  \
5038                      2.0                   318.0     Daphne Zuniga
5039                      NaN                   319.0     Valorie Curry
5040                      0.0                     0.0     Maxwell Moody
5041                      0.0                   489.0     Daniel Henney
5042                     16.0                    16.0  Brian Herzlinger

      actor_1_facebook_likes    gross                      genres  ...  \
5038                   637.0      NaN                Comedy|Drama  ...
5039                   841.0      NaN  Crime|Drama|Mystery|Thriller  ...
5040                     0.0      NaN         Drama|Horror|Thriller  ...
5041                   946.0  10443.0          Comedy|Drama|Romance  ...
5042                    86.0  85222.0                   Documentary  ...

     num_user_for_reviews language  country  content_rating  budget  \
5038                  6.0  English   Canada             NaN     NaN
5039                359.0  English      USA           TV-14     NaN
5040                  3.0  English      USA             NaN  1400.0
5041                  9.0  English      USA           PG-13     NaN
5042                 84.0  English      USA              PG  1100.0

      title_year actor_2_facebook_likes imdb_score  aspect_ratio  \
5038      2013.0                  470.0        7.7           NaN
5039         NaN                  593.0        7.5         16.00
5040      2013.0                    0.0        6.3           NaN
5041      2012.0                  719.0        6.3          2.35
5042      2004.0                   23.0        6.6          1.85

     movie_facebook_likes
5038                   84
5039                32000
5040                   16
5041                  660
5042                  456

[5 rows x 28 columns]
```

```
for i in movie.columns:
    print(i)
```

```
color
director_name
num_critic_for_reviews
```

```
duration
director_facebook_likes
actor_3_facebook_likes
actor_2_name
actor_1_facebook_likes
gross
genres
actor_1_name
movie_title
num_voted_users
cast_total_facebook_likes
actor_3_name
facenumber_in_poster
plot_keywords
movie_imdb_link
num_user_for_reviews
language
country
content_rating
budget
title_year
actor_2_facebook_likes
imdb_score
aspect_ratio
movie_facebook_likes
```

```
print("\nInformation about the dataset:")
movie.info()
```

```
Information about the dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5043 entries, 0 to 5042
Data columns (total 28 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   color                      5024 non-null   object 
 1   director_name              4939 non-null   object 
 2   num_critic_for_reviews     4993 non-null   float64
 3   duration                   5028 non-null   float64
 4   director_facebook_likes    4939 non-null   float64
 5   actor_3_facebook_likes     5020 non-null   float64
 6   actor_2_name               5030 non-null   object 
 7   actor_1_facebook_likes     5036 non-null   float64
 8   gross                      4159 non-null   float64
 9   genres                     5043 non-null   object 
 10  actor_1_name               5036 non-null   object 
 11  movie_title                5043 non-null   object 
 12  num_voted_users            5043 non-null   int64  
 13  cast_total_facebook_likes  5043 non-null   int64  
 14  actor_3_name               5020 non-null   object 
 15  facenumber_in_poster       5030 non-null   float64
 16  plot_keywords              4890 non-null   object 
 17  movie_imdb_link            5043 non-null   object 
 18  num_user_for_reviews       5022 non-null   float64
 19  language                   5031 non-null   object 
 20  country                    5038 non-null   object 
 21  content_rating             4740 non-null   object 
 22  budget                     4551 non-null   float64
 23  title_year                 4935 non-null   float64
 24  actor_2_facebook_likes     5030 non-null   float64
 25  imdb_score                 5043 non-null   float64
 26  aspect_ratio               4714 non-null   float64
 27  movie_facebook_likes       5043 non-null   int64  
dtypes: float64(13), int64(3), object(12)
memory usage: 1.1+ MB
```

```
print("\nShape of the dataset:")
movie.shape
```

```
Shape of the dataset:
(5043, 28)
```

```
movie.describe()
```

	num_critic_for_reviews	duration	director_facebook_likes	actor_3_faceboo
count	4993.000000	5028.000000	4939.000000	5020
mean	140.194272	107.201074	686.509212	645
std	121.601675	25.197441	2813.328607	1665
min	1.000000	7.000000	0.000000	0
25%	50.000000	93.000000	7.000000	133
50%	110.000000	103.000000	49.000000	371
75%	195.000000	118.000000	194.500000	636
max	813.000000	511.000000	23000.000000	23000

```
print("\nDescriptive statistics of the dataset:")
print(movie.describe())
```

```
Descriptive statistics of the dataset:
       num_critic_for_reviews     duration  director_facebook_likes  \
count             4993.000000  5028.000000              4939.000000
mean               140.194272   107.201074               686.509212
std                121.601675    25.197441              2813.328607
min                  1.000000     7.000000                 0.000000
25%                 50.000000    93.000000                 7.000000
50%                110.000000   103.000000                49.000000
75%                195.000000   118.000000               194.500000
max                813.000000   511.000000             23000.000000

       actor_3_facebook_likes  actor_1_facebook_likes         gross  \
count             5020.000000             5036.000000  4.159000e+03
mean               645.009761             6560.047061  4.846841e+07
std               1665.041728            15020.759120  6.845299e+07
min                  0.000000                0.000000  1.620000e+02
25%                133.000000              614.000000  5.340988e+06
50%                371.500000              988.000000  2.551750e+07
75%                636.000000            11000.000000  6.230944e+07
max              23000.000000           640000.000000  7.605058e+08

       num_voted_users  cast_total_facebook_likes  facenumber_in_poster  \
count     5.043000e+03                5043.000000           5030.000000
mean      8.366816e+04                9699.063851              1.371173
std       1.384853e+05               18163.799124              2.013576
min       5.000000e+00                   0.000000              0.000000
25%       8.593500e+03                1411.000000              0.000000
50%       3.435900e+04                3090.000000              1.000000
75%       9.630900e+04               13756.500000              2.000000
max       1.689764e+06              656730.000000             43.000000

       num_user_for_reviews        budget   title_year  \
count           5022.000000  4.551000e+03  4935.000000
mean             272.770808  3.975262e+07  2002.470517
std              377.982886  2.061149e+08    12.474599
min                1.000000  2.180000e+02  1916.000000
25%               65.000000  6.000000e+06  1999.000000
50%              156.000000  2.000000e+07  2005.000000
75%              326.000000  4.500000e+07  2011.000000
max             5060.000000  1.221550e+10  2016.000000

       actor_2_facebook_likes   imdb_score  aspect_ratio  movie_facebook_likes
count             5030.000000  5043.000000   4714.000000           5043.000000
mean              1651.754473     6.442138      2.220403           7525.964505
std               4042.438863     1.125116      1.385113          19320.445110
min                  0.000000     1.600000      1.180000              0.000000
25%                281.000000     5.800000      1.850000              0.000000
50%                595.000000     6.600000      2.350000            166.000000
75%                918.000000     7.200000      2.350000           3000.000000
max             137000.000000     9.500000     16.000000         349000.000000
```

```
print("\nMissing values in the dataset:")
print(movie.isnull().sum())
```

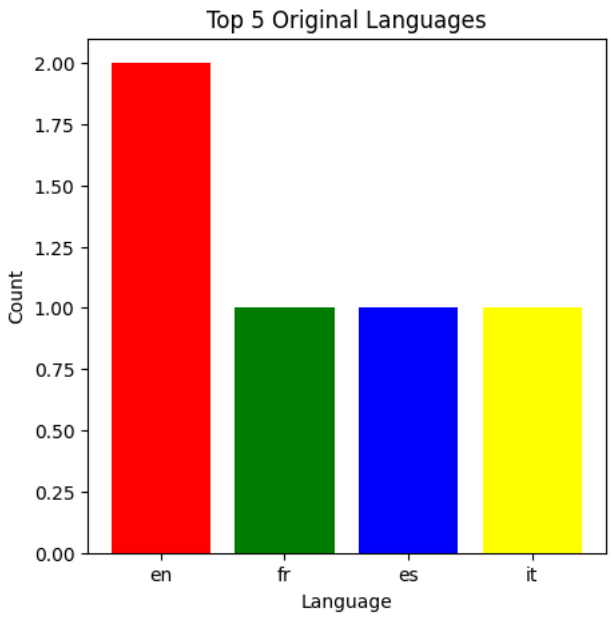
```
Missing values in the dataset:
color                       19
director_name              104
num_critic_for_reviews      50
duration                    15
director_facebook_likes    104
actor_3_facebook_likes      23
actor_2_name                13
actor_1_facebook_likes       7
gross                      884
```

```
genres                      0
actor_1_name                7
movie_title                 0
num_voted_users             0
cast_total_facebook_likes   0
actor_3_name               23
facenumber_in_poster       13
plot_keywords             153
movie_imdb_link             0
num_user_for_reviews       21
language                   12
country                     5
content_rating            303
budget                    492
title_year                108
actor_2_facebook_likes     13
imdb_score                  0
aspect_ratio              329
movie_facebook_likes        0
dtype: int64
```

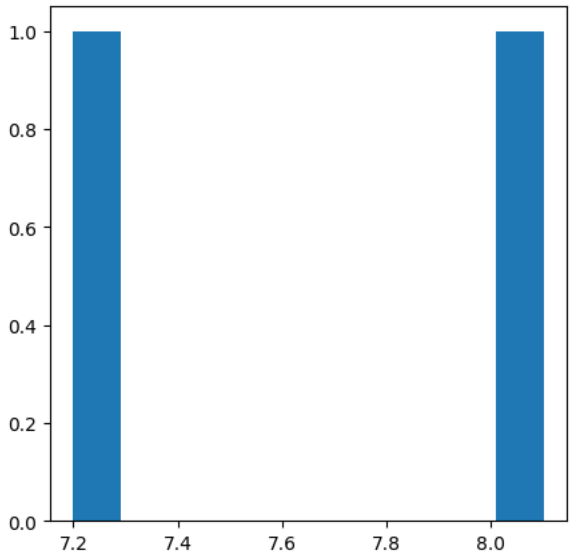
```
movie = pd.DataFrame({"original_language": ["en", "fr", "en", "es", "it"]})


print("\nValue counts for 'original_language':")
movie["original_language"].value_counts()
```

```
Value counts for 'original_language':
en    2
fr    1
es    1
it    1
Name: original_language, dtype: int64
```

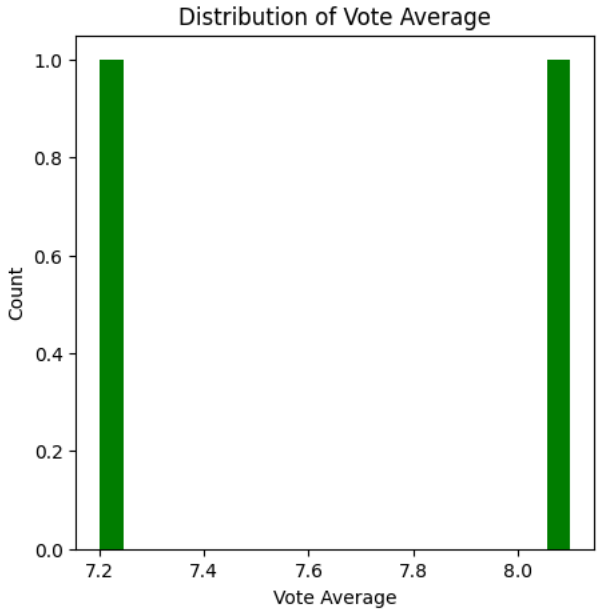
```
plt.figure(figsize=(5, 5))
plt.bar(list(movie["original_language"].value_counts().head().keys()),
        list(movie["original_language"].value_counts().head()),
        color=["red", "green", "blue", "yellow", "orange"])
plt.title("Top 5 Original Languages")
plt.xlabel("Language")
plt.ylabel("Count")
plt.show()
```




```
plt.figure(figsize = (5,5))
plt.hist(movie["vote_average"])
plt.show()
```




```
plt.figure(figsize=(5, 5))
plt.hist(movie["vote_average"], color="green", bins=20)
plt.title("Distribution of Vote Average")
plt.xlabel("Vote Average")
plt.ylabel("Count")
plt.show()
```




```
high_rated_movies = movie[movie["vote_average"] > 8]
```

```
high_rated_movies.head()
```

	original_language	vote_average
1	fr	8.1

```
high_rated_movies.shape
```

```
(1, 2)
```

```
top5_high = high_rated_movies.sort_values(by = "vote_average", ascending =False).head()
```

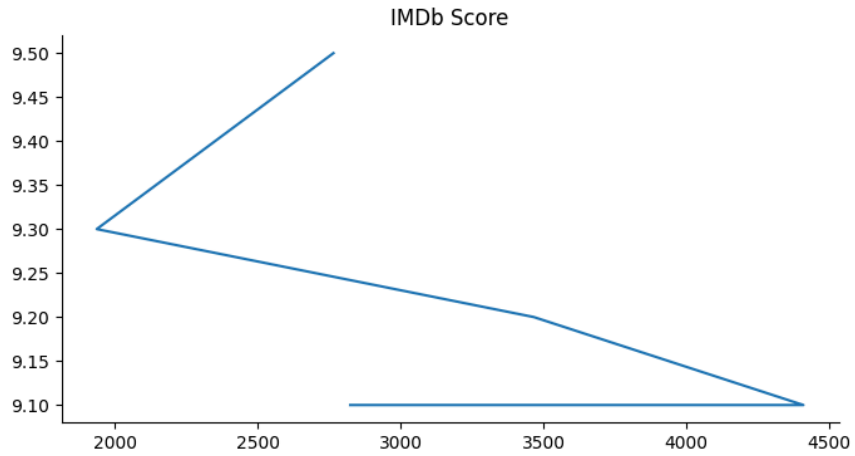
```
top5_high
```

	original_language	vote_average
1	fr	8.1

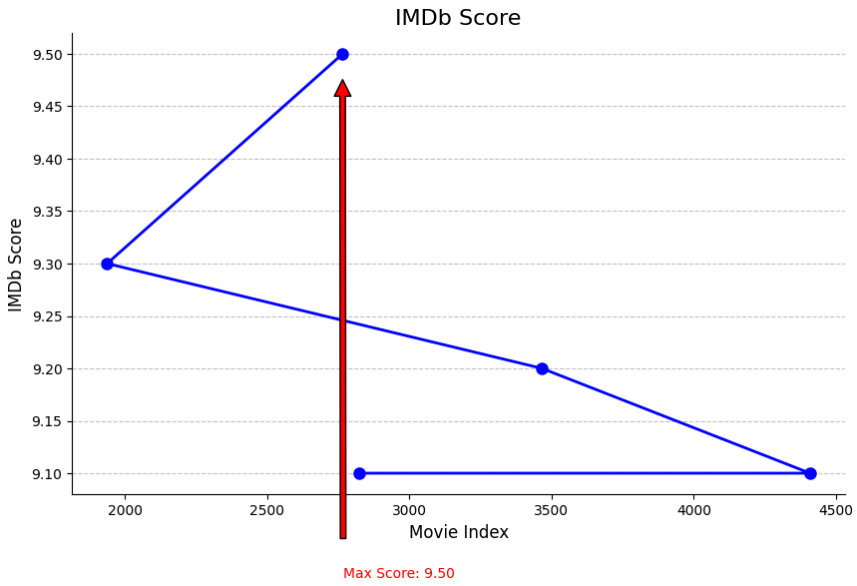
```
top5_revenue = movie.sort_values(by="imdb_score", ascending=False).head()
print(top5_revenue)
```

```
      color         director_name  num_critic_for_reviews  duration  \
2765  Color        John Blanchard                     NaN      65.0
1937  Color        Frank Darabont                   199.0     142.0
3466  Color  Francis Ford Coppola                   208.0     175.0
4409    NaN        John Stockwell                     2.0      90.0
2824  Color                   NaN                    53.0      55.0

      director_facebook_likes  actor_3_facebook_likes     actor_2_name  \
2765                      0.0                   176.0    Andrea Martin
1937                      0.0                   461.0   Jeffrey DeMunn
3466                      0.0                  3000.0    Marlon Brando
4409                    134.0                   354.0       T.J. Storm
2824                      NaN                     2.0  Olaf Lubaszenko

      actor_1_facebook_likes        gross       genres  ...  \
2765                   770.0          NaN       Comedy  ...
1937                 11000.0   28341469.0  Crime|Drama  ...
3466                 14000.0  134821952.0  Crime|Drama  ...
4409                260000.0          NaN       Action  ...
2824                    20.0     447093.0        Drama  ...

     num_user_for_reviews language  country  content_rating      budget  \
2765                  NaN  English   Canada             NaN         NaN
1937               4144.0  English      USA               R  25000000.0
3466               2238.0  English      USA               R   6000000.0
4409                  1.0      NaN      USA             NaN  17000000.0
2824                 37.0   Polish   Poland           TV-MA         NaN

      title_year actor_2_facebook_likes imdb_score  aspect_ratio  \
2765         NaN                 179.0        9.5          1.33
1937      1994.0                 745.0        9.3          1.85
3466      1972.0               10000.0        9.2          1.85
4409      2016.0                 454.0        9.1           NaN
2824         NaN                   3.0        9.1          1.33

     movie_facebook_likes
2765                    0
1937               108000
3466                43000
4409                    0
2824                    0

[5 rows x 28 columns]
```

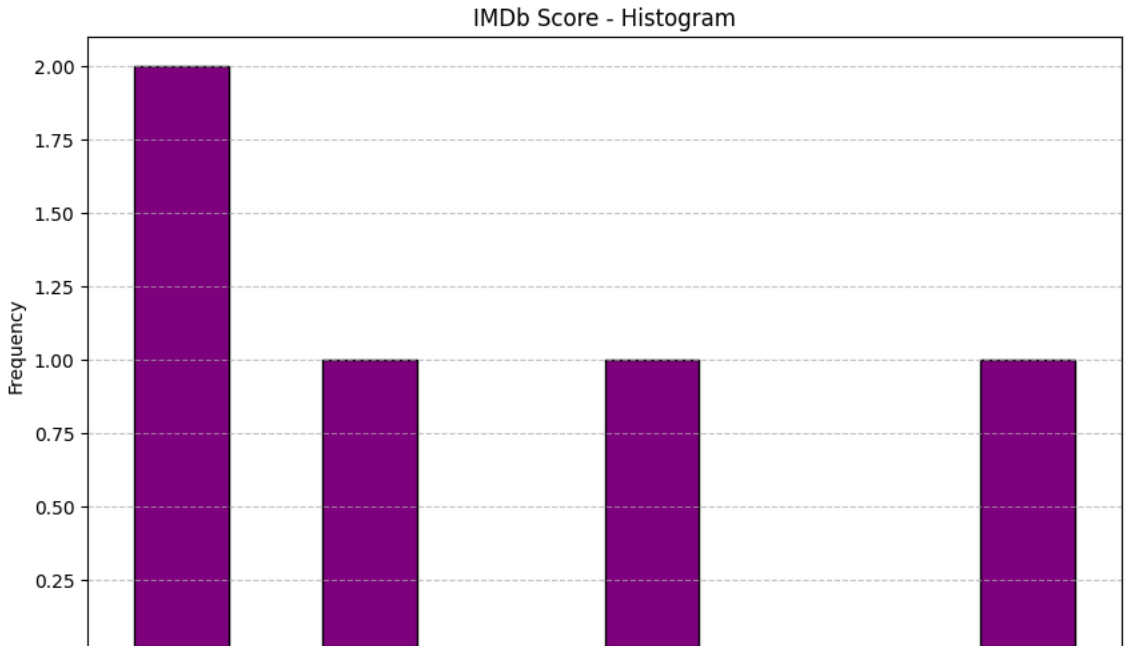
```
from matplotlib import pyplot as plt

top5_revenue['imdb_score'].plot(kind='line', figsize=(8, 4), title='IMDb Score')
plt.gca().spines[['top', 'right']].set_visible(False)
plt.show()
```



```
line_color = 'b'

ax = top5_revenue['imdb_score'].plot(kind='line', figsize=(10, 6), color=line_color, marker='o', linestyle='-', linewidth=2, markersize

ax.set_title('IMDb Score', fontsize=16)
ax.set_xlabel('Movie Index', fontsize=12)
ax.set_ylabel('IMDb Score', fontsize=12)

ax.tick_params(axis='both', which='both', labelsize=10)

ax.spines[['top', 'right']].set_visible(False)
ax.grid(axis='y', linestyle='--', alpha=0.7)

max_score_index = top5_revenue['imdb_score'].idxmax()
max_score = top5_revenue['imdb_score'].max()
ax.annotate(f'Max Score: {max_score:.2f}', xy=(max_score_index, max_score),
            xytext=(max_score_index + 2, max_score - 0.5),
            arrowprops=dict(facecolor='red', shrink=0.05),
            fontsize=10, color='red')

plt.show()
```




```
plt.figure(figsize=(10, 6))
plt.hist(top5_revenue['imdb_score'], bins=10, color='purple', edgecolor='black')
plt.title('IMDb Score - Histogram')
plt.xlabel('IMDb Score')
plt.ylabel('Frequency')
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```


IMDb Score - Histogram
Frequency
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25