```
#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#create a dataframe
df=pd.read_csv("/content/drive/MyDrive/Classroom/tmdb_5000_movies.csv")
```

```
df.head()
```

	budget	genres	homepage	id	keywords	origi
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...	

```
df.shape
```

```
(4803, 20)
```

```
#to get information about dataset
df.info()
```

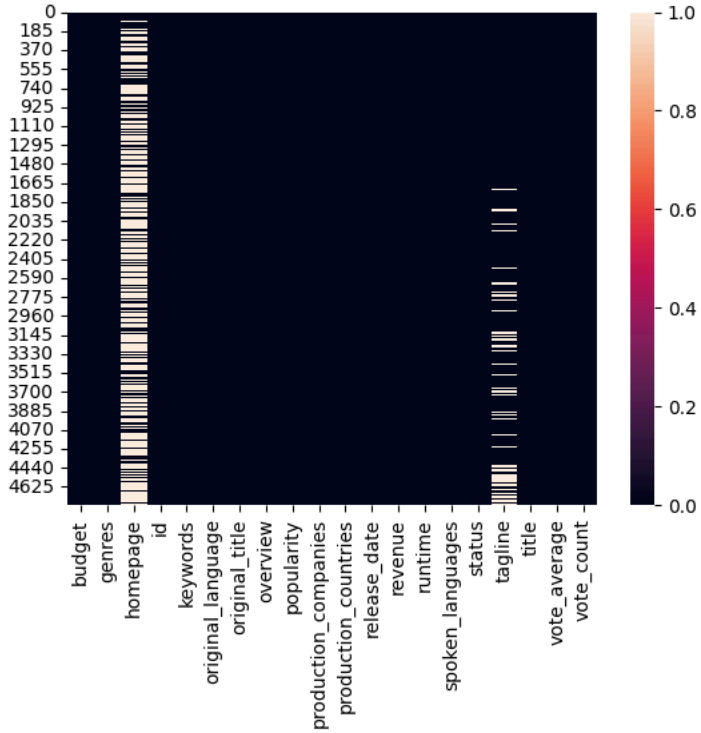
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   budget                4803 non-null   int64
 1   genres                4803 non-null   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
```

```
#check missing value
df.isnull().sum()
```

```
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
```

```
sns.heatmap(df.isnull())
plt.show()
```




```
#to check the missing value percentages
percentage_missing=df.isnull().sum()*100/len(df)
print(percentage_missing)
```

```
budget                  0.000000
genres                  0.000000
homepage               64.355611
id                      0.000000
```

```
keywords                0.000000
original_language       0.000000
original_title          0.000000
overview                0.062461
popularity              0.000000
production_companies    0.000000
production_countries    0.000000
release_date            0.020820
revenue                 0.000000
runtime                 0.041641
spoken_languages        0.000000
status                  0.000000
tagline                17.572351
title                   0.000000
vote_average            0.000000
vote_count              0.000000
dtype: float64
```

```
#drop all missing values
df.dropna(axis=0)
```

	budget	genres	homepage	id	keywords
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on novel"}, {"id":...
...	...	...	...	...	...
4773	27000	[{"id": 35, "name": "Comedy"}]	http://www.miramax.com/movie/clerks/	2292	[{"id": 1361, "name": "salesclerk"}, {"id": 30...
4781	22000	[{"id": 35, "name": "Comedy"}, {"id": 10749, "...	https://www.facebook.com/DrySpellMovie	255266	[{"id": 13043, "name": "dating"}, {"id": 15160...
4791	13	[{"id": 27, "name": "Horror"}]	http://tincanmanthemovie.com/	157185	[{"id": 14903, "name": "home invasion"}]
4796	7000	[{"id": 878, "name": "Science Fiction"}, {"id"...	http://www.primermovie.com	14337	[{"id": 1448, "name": "distrust"}, {"id": 2101...
4801	0	[]	http://shanghaicalling.com/	126186	[]

1493 rows × 20 columns

```
#check for duplicate data
duplicate_data=df.duplicated().any
print(duplicate_data)
```

```
<bound method NDFrame._add_numeric_operations.<locals>.any of 0       False
1       False
2       False
3       False
4       False
        ...
4798    False
4799    False
4800    False
4801    False
4802    False
Length: 4803, dtype: bool>
```

```
#to get stattistics about the dataframe
df.describe(include='all')
```

	budget	genres	homepage	id	keywords	or
count	4.803000e+03	4803	1712	4803.000000	4803	
unique	NaN	1175	1691	NaN	4222	
top	NaN	[{"id": 18, "name": "Drama"}]	http://www.missionimpossible.com/	NaN	[]	
freq	NaN	370	4	NaN	412	
mean	2.904504e+07	NaN	NaN	57165.484281	NaN	
std	4.072239e+07	NaN	NaN	88694.614033	NaN	
min	0.000000e+00	NaN	NaN	5.000000	NaN	
25%	7.900000e+05	NaN	NaN	9014.500000	NaN	
50%	1.500000e+07	NaN	NaN	14629.000000	NaN	
75%	4.000000e+07	NaN	NaN	58610.500000	NaN	
max	3.800000e+08	NaN	NaN	459488.000000	NaN	

```
#display the movie name having runtime >=180minitues
df.columns
```

```
Index(['budget', 'genres', 'homepage', 'id', 'keywords', 'original_language',
       'original_title', 'overview', 'popularity', 'production_companies',
       'production_countries', 'release_date', 'revenue', 'runtime',
       'spoken_languages', 'status', 'tagline', 'title', 'vote_average',
       'vote_count'],
      dtype='object')
```

```
df[df['runtime']>=180]['title']
```

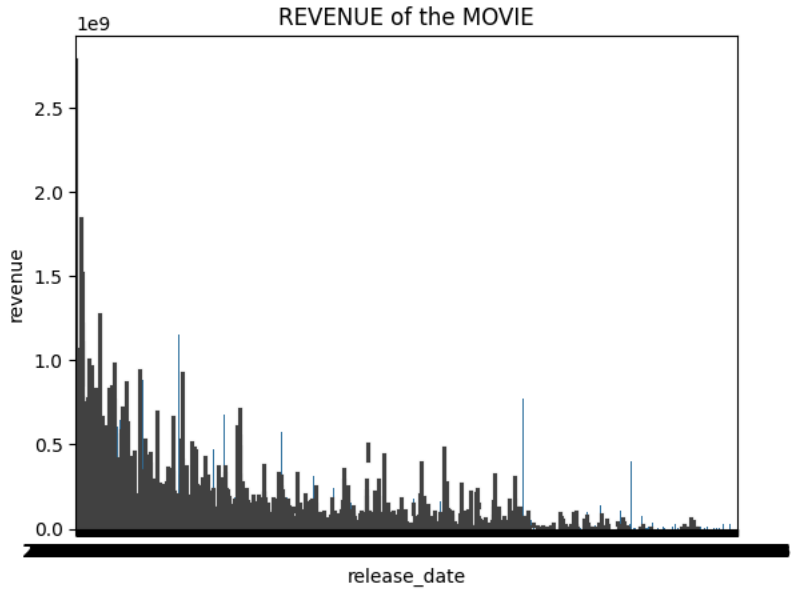
```
24                                      King Kong
25                                        Titanic
110                                   Pearl Harbor
298                          The Wolf of Wall Street
329     The Lord of the Rings: The Return of the King
676                                     Wyatt Earp
690                                The The Green Mile
855                             Gods and Generals
880                                    Grindhouse
1091                                        Nixon
1109                                Heaven's Gate
1125                                    Cleopatra
1181                                          JFK
1333                                     Magnolia
1387                                    Malcolm X
1456                                Bound by Honor
1477                                         Reds
1663                  Once Upon a Time in America
1759                               The Right Stuff
1818                             Schindler's List
1922                                   Gettysburg
2024                                       Gandhi
2192                  The Greatest Story Ever Told
2278                            Dances with Wolves
2300                  The Fall of the Roman Empire
2373                                       Hamlet
2384                                       Carlos
2536                                The Deer Hunter
2550                             Lawrence of Arabia
2631                                    The Company
2731                          The Godfather: Part II
2914                                  Doctor Zhivago
2936                                    Barry Lyndon
2962                          Kabhi Alvida Naa Kehna
3161                             Fiddler on the Roof
3191                        The Legend of Suriyothai
3285                                        Restless
```

```
3374                                    Veer-Zaara
3510                                         Emma
3723                           Anne of Green Gables
3813                             Gone with the Wind
3852                                     The Secret
3914                           Judgment at Nuremberg
4389                Chocolate: Deep Dark Secrets
4497                                     Woodstock
4535                                  Seven Samurai
4592                                    Intolerance
Name: title, dtype: object
```

```
#find the highest revenue movie
df.groupby('release_date')['revenue'].mean().sort_values(ascending=False)
```

```
release_date
2012-04-25    1.519558e+09
2015-04-01    1.506249e+09
2009-12-10    1.455100e+09
2015-04-22    1.405404e+09
2015-06-09    1.185570e+09
                  ...
2012-03-10    0.000000e+00
2000-09-06    0.000000e+00
2000-08-31    0.000000e+00
2012-03-16    0.000000e+00
2017-02-03    0.000000e+00
Name: revenue, Length: 3280, dtype: float64
```

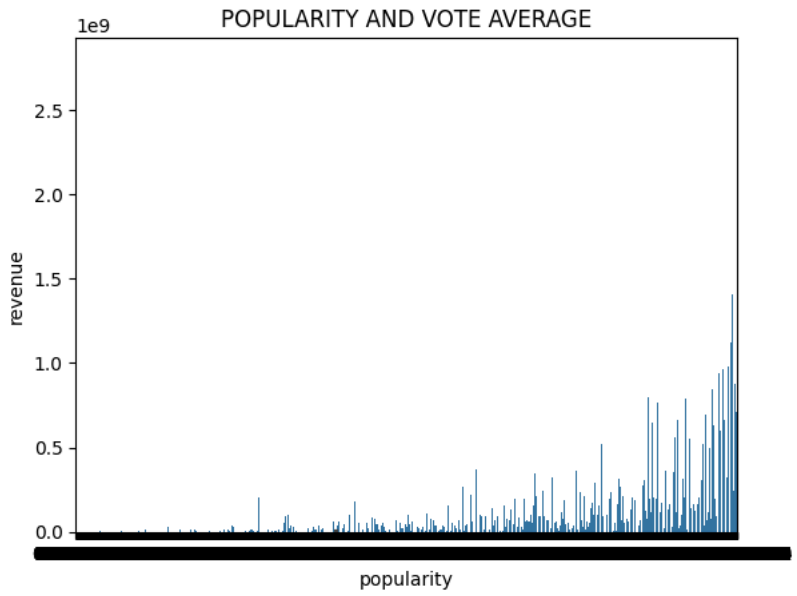
```
sns.barplot(x='release_date',y='revenue',data=df)
plt.title("REVENUE of the MOVIE")
plt.show()
```




```
df.groupby('popularity')['revenue'].mean().sort_values(ascending=False)
```

```
popularity
150.437577    2.787965e+09
100.025899    1.845034e+09
144.448633    1.519558e+09
418.708552    1.513529e+09
102.322217    1.506249e+09
                  ...
5.870319      0.000000e+00
5.871930      0.000000e+00
5.900535      0.000000e+00
5.902590      0.000000e+00
12.928269     0.000000e+00
Name: revenue, Length: 4802, dtype: float64
```

```
sns.barplot(x='popularity',y='revenue',data=df)
plt.title("POPULARITY AND VOTE AVERAGE")
plt.show()
```



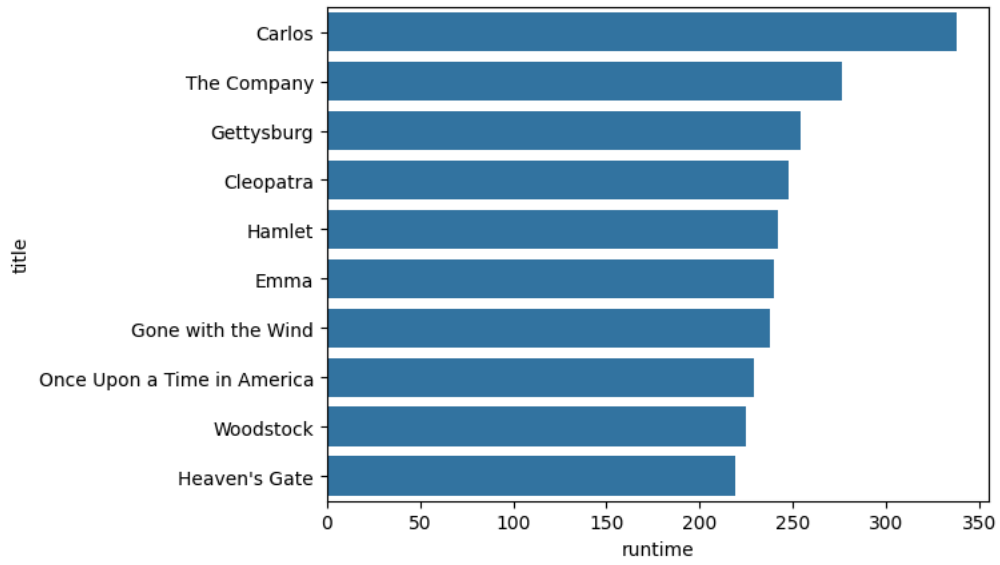

```
#TO didplay top 10 movies title in runtime
df.columns
```

```
Index(['budget', 'genres', 'homepage', 'id', 'keywords', 'original_language',
       'original_title', 'overview', 'popularity', 'production_companies',
       'production_countries', 'release_date', 'revenue', 'runtime',
       'spoken_languages', 'status', 'tagline', 'title', 'vote_average',
       'vote_count'],
      dtype='object')
```

```
top10=df.nlargest(10,'runtime')[['title','runtime']]\
.set_index('title')
print(top10)
```

```
                               runtime
title
Carlos                           338.0
The Company                      276.0
Gettysburg                       254.0
Cleopatra                        248.0
Hamlet                           242.0
Emma                             240.0
Gone with the Wind               238.0
Once Upon a Time in America      229.0
Woodstock                        225.0
Heaven's Gate                    219.0
```

```
sns.barplot(x='runtime',y=top10.index,data=top10)
plt.show()
```



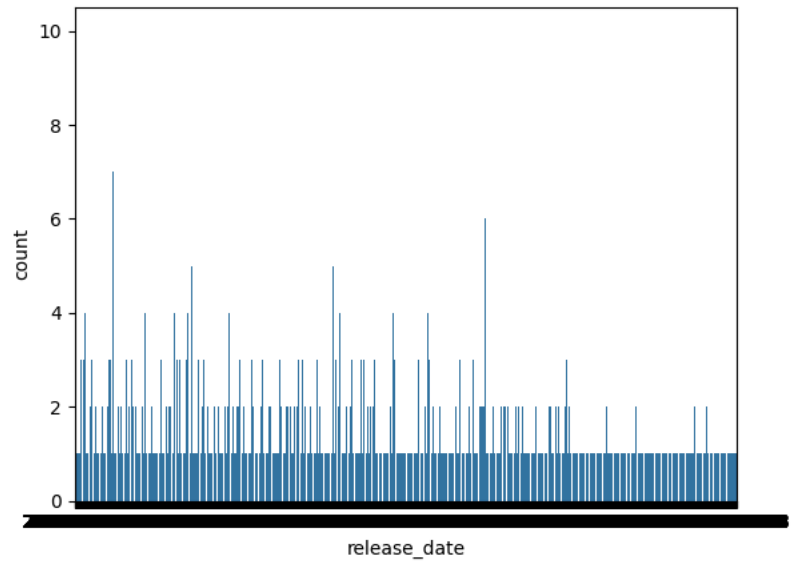

```
#number of movies in year
df.columns
```

```
Index(['budget', 'genres', 'homepage', 'id', 'keywords', 'original_language',
       'original_title', 'overview', 'popularity', 'production_companies',
       'production_countries', 'release_date', 'revenue', 'runtime',
       'spoken_languages', 'status', 'tagline', 'title', 'vote_average',
       'vote_count'],
      dtype='object')
```

```
df['release_date'].value_counts()
```

```
2006-01-01    10
2002-01-01     8
2004-09-03     7
1999-10-22     7
2013-07-18     7
              ..
2002-12-30     1
2002-08-20     1
1987-11-05     1
2004-11-11     1
2012-05-03     1
Name: release_date, Length: 3280, dtype: int64
```

```
sns.countplot(x='release_date',data=df)
plt.show()
```



```
#find highest popularity movie title
df.columns
```