



Data Science Project

- **Aya Tarek**

42010581

- **Yasmeen Mamdouh**

42010457

- **Salma Elbadry**

42010548

- **Mustafa Mahmoud**

42010544

- **Ahmed Mohamed**

42010237

- **Moaz Adel**

42010458

Dr/ Sherif Elatrby

Eng/ Ahmed Nousir

Introduction

- We have a system that talks about a large group of movies consisting of 10,000 rows and 10 columns, for sure the data set contains missing values and duplicated and so on.
- The first thing we will do is prepossessing the data through a set of commands that will be clarified now in the next lines.
- The name of columns (`id`, `title`, `release_date`, `genres`, `original_languag`, `vote_count`, `popularity`, `new_budget`, `production_companies`, `vote_average`)

Preprocessing

Read the data

We will start with the first thing to do “import” for the libraries and start reading the data

```
In [237]: #Import the library of pandas  
import pandas as pd  
import numpy as np
```

```
In [238]: #Read the dataset of movies  
ds = pd.read_csv("popular_10000_movies_tmdb.csv")
```

```
In [239]: #Show the dataset  
ds
```

Out[239]:

	id	title	release_date	genres	original_language	vote_average	vote_count	popularity	budget	production_companies
0	758323	The Pope's Exorcist	4/5/2023	['Horror', 'Mystery', 'Thriller']	English	7.4	619	5089.969	18000000	['Screen Gems', '2.0 Entertainment', 'Jesus & ...']
1	640146	Ant-Man and the Wasp: Quantumania	2/15/2023	['Action', 'Adventure', 'Science Fiction']	English	6.6	2294	4665.438	200000000	['Marvel Studios', 'Kevin Feige Productions']
2	502356	The Super Mario Bros. Movie	4/6/2023	['Animation', 'Adventure', 'Family', 'Fantasy']	English	7.5	1881	3935.550	100000000	['Universal Pictures', 'Illumination', 'Ninten...']
3	868759	Ghosted	4/18/2023	['Action', 'Comedy', 'Romance']	English	7.2	652	2791.532	0	['Skydance Media', 'Apple Studios']
4	594767	Shazam! Fury of the Gods	3/15/2023	['Action', 'Comedy', 'Fantasy', 'Adventure']	English	6.8	1510	2702.593	125000000	['New Line Cinema', 'The Safran Company', 'DC ...']
...
9995	374473	I, Daniel Blake	10/21/2016	['Drama']	English	7.7	1220	10.774	0	['Why Not Productions', 'Wild Bunch', 'Sixteen...']
9996	16774	Hellboy: Animated: Sword of Storms	10/28/2006	['TV Movie', 'Fantasy', 'Animation', 'Action']	English	6.3	99	12.739	0	['IDT Entertainment', 'Film Roman']
9997	13584	Return to House on Haunted Hill	10/3/2007	['Horror', 'Thriller']	English	5.6	263	12.769	0	['Dark Castle Entertainment', 'Warner Premiere']
9998	482204	My Sister-in-Law's Job	8/31/2017	['Drama', 'Romance']	Korean	5.0	5	10.425	0	[]
9999	444539	The Bookshop	11/10/2017	['Drama']	English	6.5	382	12.625	5400000	['Zephyr Films', 'A. Contracoriente Films', 'D...']

10000 rows × 10 columns

- In the line 240 we will use "info" for the data
- In line 241 we will convert the data type of date from “object” to “date time”

```
In [240]: #Show the info about dataset
ds.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id               10000 non-null   int64  
 1   title             10000 non-null   object  
 2   release_date      9979 non-null   object  
 3   genres            10000 non-null   object  
 4   original_language 10000 non-null   object  
 5   vote_average      10000 non-null   float64 
 6   vote_count         10000 non-null   int64  
 7   popularity         10000 non-null   float64 
 8   budget             10000 non-null   int64  
 9   production_companies 10000 non-null   object  
dtypes: float64(2), int64(3), object(5)
memory usage: 781.4+ KB
```

```
In [241]: ds['release_date'] = pd.to_datetime(ds['release_date'])
ds
```

Out[241]:

	id	title	release_date	genres	original_language	vote_average	vote_count	popularity	budget	production_companies
0	758323	The Pope's Exorcist	2023-04-05	['Horror', 'Mystery', 'Thriller']	English	7.4	619	5089.069	18000000	['Screen Gems', '2.0 Entertainment', 'Jesus & ...']
1	640146	Ant-Man and the Wasp: Quantumania	2023-02-15	['Action', 'Adventure', 'Science Fiction']	English	6.6	2294	4665.438	200000000	['Marvel Studios', 'Kevin Feige Productions']
2	502356	The Super Mario Bros. Movie	2023-04-05	['Animation', 'Adventure', 'Family', 'Fantasy']	English	7.5	1861	3935.550	100000000	['Universal Pictures', 'Illumination', 'Ninten...']
3	868759	Ghosted	2023-04-18	['Action', 'Comedy', 'Romance']	English	7.2	652	2791.632	0	['Skydance Media', 'Apple Studios']
4	594767	Shazam! Fury of the Gods	2023-03-15	['Action', 'Comedy', 'Fantasy', 'Adventure']	English	6.8	1510	2702.593	125000000	['New Line Cinema', 'The Safran Company', 'DC ...']
...	
9995	374473	I, Daniel Blake	2016-10-21	['Drama']	English	7.7	1220	10.774	0	['Why Not Productions', 'Wild Bunch', 'Sixteen...']
9996	16774	Hellboy: Sword of Storms	2006-10-28	['TV Movie', 'Fantasy', 'Animation', 'Action']	English	6.3	99	12.739	0	['IDT Entertainment', 'Film Roman']
9997	13564	Return to House on Haunted Hill	2007-10-03	['Horror', 'Thriller']	English	5.6	263	12.769	0	['Dark Castle Entertainment', 'Warner Premiere']
9998	482204	My Sister-in-Law's Job	2017-08-31	['Drama', 'Romance']	Korean	5.0	5	10.425	0	[]
9999	444539	The Bookshop	2017-11-10	['Drama']	English	6.5	382	12.525	5400000	['Zephyr Films', 'A Contracorriente Films', 'D...']

10000 rows × 10 columns

- In the line 242 we will use "info" for ensure the data type become "date time"
- In line 243, we used the “shape” command to show how many rows and how many columns we have, as shown in the pictures
- In line 244, we will use the command "isnull().sum()" to check if we have null value and calculate the sum of them
- In the line 245, we will drop the null values

```
10000 rows × 10 columns

In [242]: ds.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id               10000 non-null   int64  
 1   title             10000 non-null   object  
 2   release_date      9979 non-null   datetime64[ns]
 3   genres            10000 non-null   object  
 4   original_language 10000 non-null   object  
 5   vote_average      10000 non-null   float64 
 6   vote_count         10000 non-null   int64  
 7   popularity         10000 non-null   float64 
 8   budget             10000 non-null   int64  
 9   production_companies 10000 non-null   object  
dtypes: datetime64[ns](1), float64(2), int64(3), object(4)
memory usage: 781.4+ KB

In [243]: #Shape of dataset
ds.shape

Out[243]: (10000, 10)

In [244]: #Check the null values
ds.isnull().sum()

Out[244]: id          0
title        0
release_date 21
genres        0
original_language 0
vote_average 0
vote_count    0
popularity    0
budget        0
production_companies 0
dtype: int64

In [245]: #Drop the null values
ds.dropna(inplace=True)
```

- In line 246, we will use the command "isna().any()" to check if we have null value
- In line 247, we will use the command "isnull().sum()" to check if we have null value and calculate the sum of them.
- On line 248, we used the “shape” command to show how many rows and how many columns we have, as shown in the pictures.

```
In [246]: #Check is there any null values
```

```
ds.isna().any()
```

```
Out[246]: id      False
          title    False
          release_date False
          genres   False
          original_language False
          vote_average  False
          vote_count    False
          popularity    False
          budget     False
          production_companies False
          dtype: bool
```

```
In [247]: #Check the null values
```

```
ds.isnull().sum()
```

```
Out[247]: id      0
          title    0
          release_date 0
          genres   0
          original_language 0
          vote_average 0
          vote_count    0
          popularity    0
          budget     0
          production_companies 0
          dtype: int64
```

```
In [248]: #Show the shape of dataset
```

```
ds.shape
```

```
Out[248]: (9979, 10)
```

- In line 249, we used "head (10)" to show first 10 rows

Out[249]:										
	id	title	release_date	genres	original_language	vote_average	vote_count	popularity	budget	production_companies
0	758323	The Pope's Exorcist	2023-04-05	['Horror', 'Mystery', 'Thriller']	English	7.4	619	5089.969	18000000	['Screen Gems', '2.0 Entertainment', 'Jesus & ...']
1	640146	Ant-Man and the Wasp: Quantumania	2023-02-15	['Action', 'Adventure', 'Science Fiction']	English	6.6	2294	4665.438	200000000	['Marvel Studios', 'Kevin Feige Productions']
2	502356	The Super Mario Bros. Movie	2023-04-05	['Animation', 'Adventure', 'Family', 'Fantasy']	English	7.5	1861	3935.550	100000000	['Universal Pictures', 'Illumination', 'Ninten...']
3	888759	Ghosted	2023-04-18	['Action', 'Comedy', 'Romance']	English	7.2	652	2791.532	0	['Skydance Media', 'Apple Studios']
4	594767	Shazam! Fury of the Gods	2023-03-15	['Action', 'Comedy', 'Fantasy', 'Adventure']	English	6.8	1510	2702.593	125000000	['New Line Cinema', 'The Safran Company', 'DC ...']
5	76600	Avatar: The Way of Water	2022-12-14	['Science Fiction', 'Adventure', 'Action']	English	7.7	7853	2280.912	460000000	['20th Century Studios', 'Lightstorm Entertain...']
6	447365	Guardians of the Galaxy Volume 3	2023-05-03	['Science Fiction', 'Adventure', 'Action']	English	8.3	683	2520.308	250000000	['Marvel Studios', 'Kevin Feige Productions']
7	934433	Scream VI	2023-03-08	['Horror', 'Mystery', 'Thriller']	English	7.3	1029	1862.283	35000000	['Radio Silence', 'Project X Entertainment', '...']
8	677179	Creed III	2023-03-01	['Drama', 'Action']	English	7.3	1298	1894.044	7500000	['Metro-Goldwyn-Mayer', 'Proximity Media', 'Ba...']
9	493529	Dungeons & Dragons: Honor Among Thieves	2023-03-23	['Adventure', 'Fantasy', 'Comedy']	English	7.5	964	1655.052	151000000	['Entertainment One', 'Paramount', 'Allspark P...']

- on line 250, we used "tail (10)" to show the last 10 row
- On line 251, we used "dtypes" To check the data type

In [250]: #Show the Last 10 row

```
ds.tail(10)
```

Out[250]:

	id	title	release_date	genres	original_language	vote_average	vote_count	popularity	budget	production_companies
9990	10748	St. Trinian's	2007-12-21	['Comedy', 'Family', 'Science Fiction']	English	5.9	418	7.770	0	['Ealing Studios', 'Fragile Films', 'Entertain...']
9991	917269	The Witcher Season One Recap: From the Beginning	2021-12-17	[]	English	5.6	8	9.045	0	[]
9992	914216	The Lost King	2022-10-07	['Drama', 'Comedy']	English	6.5	51	11.138	0	['Baby Cow Productions', 'BBC Film', 'Ingeniou...']
9993	823461	Good on Paper	2021-06-23	['Comedy', 'Romance']	English	5.6	265	8.544	0	['Burn Later Productions', 'Meridian Content', ...]
9994	94503	Tsubasa RESERVOIR CHRONICLE: Tokyo Revelations	2007-11-16	['Fantasy', 'Animation', 'Action', 'Adventure', ...]	Japanese	7.2	6	7.949	0	['Kodansha']
9995	374473	I, Daniel Blake	2016-10-21	['Drama']	English	7.7	1220	10.774	0	['Why Not Productions', 'Wild Bunch', 'Sixteen...']
9996	16774	Hellboy Animated: Sword of Storms	2006-10-28	['TV Movie', 'Fantasy', 'Animation', 'Action', ...]	English	6.3	99	12.739	0	['IDT Entertainment', 'Film Roman']
9997	13564	Return to House on Haunted Hill	2007-10-03	['Horror', 'Thriller']	English	5.6	263	12.769	0	['Dark Castle Entertainment', 'Warner Premiere']
9998	482204	My Sister-in-Law's Job	2017-08-31	['Drama', 'Romance']	Korean	5.0	5	10.425	0	[]
9999	444530	The Bookshop	2017-11-10	['Drama']	English	6.5	382	12.525	5400000	['Zephyr Films', 'A Contracorriente Films', 'D...']

In [251]: #Check the datatype

```
ds.dtypes
```

Out[251]:

id	int64
title	object
release_date	datetime64[ns]
genres	object
original_language	object
vote_average	float64
vote_count	int64
popularity	float64
budget	int64
production_companies	object
dtype:	object

- In line 252, show the name of all columns
- in the line 253, we used "describe" to description the dataset
- In the line 254we will use "info" for the data to show the information about the Dataset

```
In [252]: #Show the name of all column
ds.columns

Out[252]: Index(['id', 'title', 'release_date', 'genres', 'original_language',
       'vote_average', 'vote_count', 'popularity', 'budget',
       'production_companies'],
      dtype='object')

In [253]: #Show the description of dataset
ds.describe()

Out[253]:
          id  vote_average  vote_count  popularity    budget
count  9.979000e+03   9979.000000   9979.000000  9979.000000  9.979000e+03
mean   3.143436e+05    6.338822   1530.054915   31.488106   1.954562e+07
std    3.331747e+05    1.388166   2870.875959  111.528200   3.890194e+07
min    5.000000e+00    0.000000    0.000000    7.219000   0.000000e+00
25%   1.220300e+04    5.900000   123.000000   13.531500   0.000000e+00
50%   1.925770e+05    6.500000   483.000000   17.816000   3.700000e+05
75%   5.804590e+05    7.100000  1561.000000   27.175500   2.200000e+07
max   1.119921e+06   10.000000  33633.000000  5089.969000   5.793304e+08

In [254]: # In[90]:
ds.info

Out[254]: <bound method DataFrame.info of
          id              title release_date \
0      758323        The Pope's Exorcist  2023-04-05
1      640146  Ant-Man and the Wasp: Quantumania 2023-02-15
2      502356        The Super Mario Bros. Movie 2023-04-05
3      868759                  Ghosted 2023-04-18
4      594767        Shazam! Fury of the Gods 2023-03-15
...
9995  374473            I, Daniel Blake 2016-10-21
9996  16774  Hellboy Animated: Sword of Storms 2006-10-28
9997  13564     Return to House on Haunted Hill 2007-10-03
9998  482204        My Sister-in-Law's Job 2017-08-31
9999  444539           The Bookshop 2017-11-10

                                     genres original_language \
0           ['Horror', 'Mystery', 'Thriller']      English
1           ['Action', 'Adventure', 'Science Fiction']  English
2           ['Animation', 'Adventure', 'Family', 'Fantasy']...
3           ['Action', 'Comedy', 'Romance']      English
4           ['Action', 'Comedy', 'Fantasy', 'Adventure']  English
...
9995          ['Drama']      English
```

- on line 255, we will use the command "isnull().sum()" to check if we have null value and calculate the sum of them

```

1      ['Action', 'Adventure', 'Science Fiction']    English
2      ['Animation', 'Adventure', 'Family', 'Fantasy'] English
3      ['Action', 'Comedy', 'Romance']                English
4      ['Action', 'Comedy', 'Fantasy', 'Adventure']   English
...
9995     ...                                ...
9996     ['Drama']                           English
9997     ['TV Movie', 'Fantasy', 'Animation', 'Action', ...] English
9998     ['Horror', 'Thriller']                 English
9999     ['Drama', 'Romance']                  Korean
9999     ['Drama']                           English

  vote_average  vote_count  popularity  budget \
0        7.4       619  5089.969  18000000
1        6.6      2294  4665.438  20000000
2        7.5      1861  3935.550  10000000
3        7.2       652  2791.532      0
4        6.8      1510  2702.593  12500000
...
9995     ...      ...
9996     ...      ...
9997     ...      ...
9998     ...      ...
9999     ...      ...

  production_companies
0      ['Screen Gems', '2.0 Entertainment', 'Jesus & ...'] ...
1      ['Marvel Studios', 'Kevin Feige Productions'] ...
2      ['Universal Pictures', 'Illumination', 'Ninten...'] ...
3      ['Skydance Media', 'Apple Studios'] ...
4      ['New Line Cinema', 'The Safran Company', 'DC ...'] ...
...
9995     ['Why Not Productions', 'Wild Bunch', 'Sixteen...'] ...
9996     ['IDT Entertainment', 'Film Roman'] ...
9997     ['Dark Castle Entertainment', 'Warner Premiere'] ...
9998     []
9999     ['Zephyr Films', 'A Contracorriente Films', 'D...'] ...

[9999 rows x 10 columns]

```

In [255]: #check the null values
`ds.isnull().sum()`

```

Out[255]: id          0
title        0
release_date 0
genres        0
original_language 0
vote_average 0
vote_count    0
popularity    0
budget        0
production_companies 0
dtype: int64

```

- In the line 256, drop the duplicated rows.
- In the line 257, To calculate the mean of vote average used ".mean ()"
- In the line 258, To calculate the standard deviation of vote average used ".Std ()"
- In the line 259 To calculate the variance of vote average used ".Var ()"

```
In [256]: ds.drop_duplicates()

Out[256]:
   id      title release_date    genres original_language  vote_average  vote_count popularity  budget  production_companies
0  758323  The Pope's Exorcist  2023-04-05  [Horror, Mystery, Thriller]  English       7.4        619  5089.969  18000000  [Screen Gems', '2.0 Entertainment', 'Jesus & ...
1  640146  Ant-Man and the Wasp: Quantumania  2023-02-15  [Action, Adventure, Science Fiction]  English       6.6       2294  4665.438  200000000  ['Marvel Studios', 'Kevin Feige Productions']
2  502356  The Super Mario Bros. Movie  2023-04-05  [Animation, Adventure, Family, Fantasy...]  English       7.5       1861  3935.550  100000000  [Universal Pictures', 'Illumination', 'Ninten...
3  888759     Ghosted  2023-04-18  [Action, Comedy, Romance]  English       7.2       652  2791.532       0  ['Skydance Media', 'Apple Studios]
4  504767  Shazam! Fury of the Gods  2023-03-15  [Action, Comedy, Fantasy, Adventure]  English       6.8       1510  2702.593  125000000  ['New Line Cinema', 'The Safran Company', 'DC ...
...
9995  374473  I, Daniel Blake  2016-10-21  [Drama]  English       7.7       1220  10.774       0  ['Why Not Productions', 'Wild Bunch', 'Sixteen...
9996  16774  Hellboy: Sword of Storms  2006-10-28  [TV Movie, Fantasy, Animation, Action,...]  English       6.3        99  12.739       0  ['IDT Entertainment', 'Film Roman']
9997  13564  Return to House on Haunted Hill  2007-10-03  [Horror, Thriller]  English       5.6       263  12.769       0  [Dark Castle Entertainment', 'Warner Premiers]
9998  482204  My Sister-in-Law's Job  2017-08-31  [Drama, Romance]  Korean        5.0        5  10.425       0  []
9999  444539     The Bookshop  2017-11-10  [Drama]  English       6.5       382  12.525  5400000  [Zephyr Films', 'A Contracorriente Films', 'D...

9979 rows × 10 columns
```

```
In [257]: mean_vote_average = ds['vote_average'].mean()
print (mean_vote_average)

6.338821525202926
```

```
In [258]: standard_deviation = ds['vote_average'].std()
standard_deviation
```

```
Out[258]: 1.3881662242465918
```

```
In [259]: var = ds['vote_average'].var()
var
```

```
Out[259]: 1.927005466139039
```

- On line 260, To calculate the correlation between vote average, popularity used ".Corr()
- On line 261, To show the row between 1:99 used ".iloc(1:100)"
- On line 262, we define a function for the vote average. If the value is greater than 6, it displays “good,” and if it is smaller, it displays “bad.

```
In [260]: correlation = ds['vote_average'].corr(ds['popularity'])
print (correlation)
```

0.041080957530214506

```
In [261]: id = ds.iloc[1:100]
id
```

Out[261]:

		id	title	release_date	genres	original_language	vote_average	vote_count	popularity	budget	production_companies
1	640146	Ant-Man and the Wasp: Quantumania	2023-02-15	['Action', 'Adventure', 'Science Fiction']	English	6.6	2294	4865.438	200000000	['Marvel Studios', 'Kevin Feige Productions']	
2	502356	The Super Mario Bros. Movie	2023-04-05	['Animation', 'Adventure', 'Family', 'Fantasy']	English	7.5	1861	3935.550	100000000	['Universal Pictures', 'Illumination', 'Ninten...']	
3	868759	Ghosted	2023-04-18	['Action', 'Comedy', 'Romance']	English	7.2	662	2791.532	0	['Skydance Media', 'Apple Studios']	
4	594787	Shazam! Fury of the Gods	2023-03-15	['Action', 'Comedy', 'Fantasy', 'Adventure']	English	6.8	1510	2702.593	125000000	['New Line Cinema', 'The Safran Company', 'DC ...']	
5	78600	Avatar: The Way of Water	2022-12-14	['Science Fiction', 'Adventure', 'Action']	English	7.7	7853	2280.912	480000000	['20th Century Studios', 'Lightstorm Entertain...']	
...	
96	1013880	R.I.P.D. 2: Rise of the Damned	2022-11-15	['Action', 'Comedy', 'Fantasy', 'Crime']	English	6.6	462	206.501	0	['Universal 1440 Entertainment']	
97	829410	Sick	2022-09-11	['Horror', 'Thriller']	English	6.3	132	206.397	0	['Miramax', 'Outerbanks Entertainment', 'Blumh...']	
98	1058732	The Simpsons Meet the Bocelis in Feliz Navidad	2022-12-15	['Animation', 'Family', 'Comedy']	English	5.2	36	194.059	0	['Gracie Films', '20th Television', '20th Tele...']	
99	1101799	Queens on the Run	2023-04-14	['Comedy', 'Action']	Spanish	7.0	83	204.485	0	['Perro Azul', 'Selva Pictures']	
100	873256	Kiss, Kiss!	2023-04-26	['Romance', 'Comedy']	Polish	6.9	15	361.900	0	['Lightcraft']	

99 rows × 10 columns

```
In [262]: def vote_average(i):
    if i > 6:
        return ("good")
    else:
        return ("bad")
```

- On line 263, we apply this function with ".apply()".
- On line 264, we print the dataset

```
In [263]: ds['vote_category'] = ds['vote_average'].apply(vote_average)

In [264]: print(ds)

      id          title release_date \
0    758233  The Pope's Exorcist  2023-04-05
1   640146  Ant-Man and the Wasp: Quantumania 2023-02-15
2   502356  The Super Mario Bros. Movie 2023-04-05
3   868759           Ghosted 2023-04-18
4   594767       Shazam! Fury of the Gods 2023-03-15
...
9995  374473        I, Daniel Blake 2016-10-21
9996  16774 Hellboy Animated: Sword of Storms 2006-10-28
9997  13564  Return to House on Haunted Hill 2007-10-03
9998  482204     My Sister-in-law's Job 2017-08-31
9999  444539      The Bookshop 2017-11-10

      genres original_language \
0      ['Horror', 'Mystery', 'Thriller']      English
1      ['Action', 'Adventure', 'Science Fiction']      English
2      ['Animation', 'Adventure', 'Family', 'Fantasy']...
3      ['Action', 'Comedy', 'Romance']      English
4      ['Action', 'Comedy', 'Fantasy', 'Adventure']      English
...
9995      ['Drama']      English
9996  ['TV Movie', 'Fantasy', 'Animation', 'Action',...
9997      ['Horror', 'Thriller']      English
9998      ['Drama', 'Romance']      Korean
9999      ['Drama']      English

  vote_average  vote_count  popularity      budget \
0         7.4        619  5089.969  18000000
1         6.6       2294  4665.438  20000000
2         7.5       1861  3935.550  10000000
3         7.2        652  2791.532        0
4         6.8       1510  2702.593  12500000
...
9995         7.7       1220  10.774        0
9996         6.3        99  12.739        0
9997         5.6       263  12.769        0
9998         5.0        5  10.425        0
9999         6.5       382  12.525  5400000

  production_companies vote_category
0  ['Screen Gems', '2.0 Entertainment', 'Jesus & ...']      good
1  ['Marvel Studios', 'Kevin Feige Productions']      good
2  ['Universal Pictures', 'Illumination', 'Ninten...']      good
3  ['Skydance Media', 'Apple Studios']      good
4  ['New Line Cinema', 'The Safran Company', 'DC ...']      good
...
9995  ['Why Not Productions', 'Wild Bunch', 'Sixteen...']      good
9996  ['IDT Entertainment', 'Film Roman']      good
9997  ['Dark Castle Entertainment', 'Warner Premiere']      bad
9998  []      bad
9999  ['Teeku Film']      good
```

- On line 265, we used the “shape” command to show how many rows and how many columns we have
- On line 266, To calculate the median that command is “.median ()”
- On line 267, that column name is "budget"=0 that replace 0 and put median value.

```
[9979 rows x 11 columns]

In [265]: ds.shape
Out[265]: (9979, 11)

In [266]: budget_median = ds['budget'].median()

In [267]: ds['budget'] = np.where(ds['budget'] == 0, budget_median, ds['budget'])
ds

Out[267]:
   id      title release_date    genres original_language  vote_average  vote_count popularity    budget production_companies  vote_category
0  758323  The Pope's Exorcist  2023-04-05  ['Horror', 'Mystery', 'Thriller']  English          7.4        619  5089.969  18000000.0  ['Screen Gems', '2.0 Entertainment', 'Jesus & ...  good
1  640146  Ant-Man and the Wasp: Quantumania  2023-02-15  ['Action', 'Adventure', 'Science Fiction']  English          6.6       2294  4685.438  200000000.0  ['Marvel Studios', 'Kevin Feige Productions']  good
2  502356  The Super Mario Bros. Movie  2023-04-05  ['Animation', 'Adventure', 'Family', 'Fantasy', 'Sci-Fi']  English          7.5       1861  3935.550  100000000.0  ['Universal Pictures', 'Illumination', 'Ninten...  good
3  868759    Ghosted  2023-04-18  ['Action', 'Comedy', 'Romance']  English          7.2        652  2791.532  370000.0  ['Skydance Media', 'Apple Studios']  good
4  594767  Shazam! Fury of the Gods  2023-03-15  ['Action', 'Comedy', 'Fantasy', 'Adventure']  English          6.8       1510  2702.593  125000000.0  ['New Line Cinema', 'The Safran Company', 'DC ...  good
...
995 374473     I, Daniel Blake  2016-10-21  ['Drama']  English          7.7       1220  10.774  370000.0  ['Why Not Productions', 'Wild Bunch', 'Sixteen...  good
996 16774  Hellboy Animated: Sword of Storms  2006-10-28  ['TV Movie', 'Fantasy', 'Animation', 'Action', 'Sci-Fi']  English          6.3        99  12.739  370000.0  ['IDT Entertainment', 'Film Roman']  good
997 13564  Return to House on Haunted Hill  2007-10-03  ['Horror', 'Thriller']  English          5.6        263  12.769  370000.0  ['Dark Castle Entertainment', 'Warner Premiere']  bad
998 482204  My Sister-in-Law's Job  2017-08-31  ['Drama', 'Romance']  Korean           5.0         5  10.425  370000.0  []  bad
999 444539      The Bookshop  2017-11-10  ['Drama']  English          6.5       382  12.525  5400000.0  ['Zephyr Films', 'A Contracorriente Films', 'D...  good

379 rows x 11 columns
```

- On line 268, we used the “shape” command to show how many rows and how many columns we have.
- On line 269, we used the “ds” command to print all dataset.

```
In [268]: ds.shape
```

```
Out[268]: (9979, 11)
```

```
In [269]: ds
```

```
Out[269]:
```

	id	title	release_date	genres	original_language	vote_average	vote_count	popularity	budget	production_companies	vote_category
0	758323	The Pope's Exorcist	2023-04-05	['Horror', 'Mystery', 'Thriller']	English	7.4	619	5089.969	18000000.0	['Screen Gems', '2.0 Entertainment', 'Jesus & ...']	good
1	640146	Ant-Man and the Wasp: Quantumania	2023-02-15	['Action', 'Adventure', 'Science Fiction']	English	6.6	2294	4665.438	200000000.0	['Marvel Studios', 'Kevin Feige Productions']	good
2	502356	The Super Mario Bros. Movie	2023-04-05	['Animation', 'Adventure', 'Family', 'Fantasy']	English	7.5	1861	3935.550	100000000.0	['Universal Pictures', 'Illumination', 'Ninten...']	good
3	868759	Ghosted	2023-04-18	['Action', 'Comedy', 'Romance']	English	7.2	652	2791.532	370000.0	['Skydance Media', 'Apple Studios']	good
4	594767	Shazam! Fury of the Gods	2023-03-15	['Action', 'Comedy', 'Fantasy', 'Adventure']	English	6.8	1510	2702.593	125000000.0	['New Line Cinema', 'The Safran Company', 'DC ...']	good
...
9995	374473	I, Daniel Blake	2016-10-21	['Drama']	English	7.7	1220	10.774	370000.0	['Why Not Productions', 'Wild Bunch', 'Sixteen...']	good
9996	16774	Hellboy Animated: Sword of Storms	2008-10-28	['TV Movie', 'Fantasy', 'Animation', 'Action']	English	6.3	99	12.739	370000.0	['IDT Entertainment', 'Film Roman']	good
9997	13564	Return to House on Haunted Hill	2007-10-03	['Horror', 'Thriller']	English	5.6	263	12.769	370000.0	['Dark Castle Entertainment', 'Warner Premiere']	bad
9998	482204	My Sister-in-Law's Job	2017-08-31	['Drama', 'Romance']	Korean	5.0	5	10.425	370000.0	[]	bad
9999	444530	The Bookshop	2017-11-10	['Drama']	English	6.5	382	12.525	5400000.0	['Zephyr Films', 'A Contracorriente Films', 'D...']	good

979 rows × 11 columns

- On line 270, we used the “shape” command to show how many rows and how many columns we have.

In [270]: ds.shape ds											
Out[270]:											
	id	title	release_date	genres	original_language	vote_average	vote_count	popularity	budget	production_companies	vote_category
0	758323	The Pope's Exorcist	2023-04-05	['Horror', 'Mystery', 'Thriller']	English	7.4	619	5089.969	18000000.0	['Screen Gems', '2.0 Entertainment', 'Jesus & ...']	good
1	640146	Ant-Man and the Wasp: Quantumania	2023-02-15	['Action', 'Adventure', 'Science Fiction']	English	6.6	2294	4665.438	200000000.0	['Marvel Studios', 'Kevin Feige Productions']	good
2	502356	The Super Mario Bros. Movie	2023-04-05	['Animation', 'Adventure', 'Family', 'Fantasy']	English	7.5	1881	3935.550	100000000.0	['Universal Pictures', 'Illumination', 'Ninten...']	good
3	868759	Ghosted	2023-04-18	['Action', 'Comedy', 'Romance']	English	7.2	652	2791.532	370000.0	['Skydance Media', 'Apple Studios']	good
4	594767	Shazam! Fury of the Gods	2023-03-15	['Action', 'Comedy', 'Fantasy', 'Adventure']	English	6.8	1510	2702.593	125000000.0	['New Line Cinema', 'The Safran Company', 'DC ...']	good
...	
9995	374473	I, Daniel Blake	2016-10-21	['Drama']	English	7.7	1220	10.774	370000.0	['Why Not Productions', 'Wild Bunch', 'Sixteen...']	good
9996	16774	Hellboy: Animated: Sword of Storms	2008-10-28	['TV Movie', 'Fantasy', 'Animation', 'Action']	English	6.3	99	12.739	370000.0	['IDT Entertainment', 'Film Roman']	good
9997	13564	Return to House on Haunted Hill	2007-10-03	['Horror', 'Thriller']	English	5.6	263	12.769	370000.0	['Dark Castle Entertainment', 'Warner Premiere']	bad
9998	482204	My Sister-in-law's Job	2017-08-31	['Drama', 'Romance']	Korean	5.0	5	10.425	370000.0	[]	bad
9999	444539	The Bookshop	2017-11-10	['Drama']	English	6.5	382	12.525	5400000.0	['Zephyr Films', 'A Contracorriente Films', 'D...']	good

979 rows × 11 columns

- On line 345 we use command ".replace" to replace "[]" to "company x" and print the data set

```
In [345]: ds.replace('[]', 'Company X', inplace=True)

In [346]: ds

Out[346]:
   id      title release_date    genres original_language  vote_average  vote_count popularity budget production_companies  vote_catego
0  758323  The Pope's Exorcist  2023-04-05  ['Horror', 'Mystery', 'Thriller']        English       7.4        619  5089.969  18000000.0  ['Screen Gems', '2.0 Entertainment', 'Jesus & ...']  gox
1  640146  Ant-Man and the Wasp: Quantumania  2023-02-15  ['Action', 'Adventure', 'Science Fiction']        English       6.6       2294  4865.438  200000000.0  ['Marvel Studios', 'Kevin Feige Productions']  gox
2  502356  The Super Mario Bros. Movie  2023-04-05  ['Animation', 'Adventure', 'Family', 'Fantasy']...        English       7.5       1881  3935.550  100000000.0  ['Universal Pictures', 'Illumination', 'Ninten...']  gox
3  868759     Ghosted  2023-04-18  ['Action', 'Comedy', 'Romance']        English       7.2        652  2791.532  370000.0  ['Skydance Media', 'Apple Studios']  gox
4  594767  Shazam! Fury of the Gods  2023-03-15  ['Action', 'Comedy', 'Fantasy', 'Adventure']        English       6.8       1510  2702.593  125000000.0  ['New Line Cinema', 'The Safran Company', 'DC ...']  gox
... ...
9995 374473     I, Daniel Blake  2016-10-21  ['Drama']        English       7.7       1220  10.774  370000.0  ['Why Not Productions', 'Wild Bunch', 'Sixteen...']  gox
9996 16774  Hellboy Animated: Sword of Storms  2008-10-28  ['TV Movie', 'Fantasy', 'Animation', 'Action']...        English       6.3        99  12.739  370000.0  ['IDT Entertainment', 'Film Roman']  gox
9997 13564  Return to Haunted Hill  2007-10-03  ['Horror', 'Thriller']        English       5.6        263  12.769  370000.0  ['Dark Castle Entertainment', 'Warner Premiere']  bix
9998 482204  My Sister-in-Law's Job  2017-08-31  ['Drama', 'Romance']        Korean       5.0         5  10.425  370000.0  Company X  bix
9999 444539     The Bookshop  2017-11-10  ['Drama']        English       6.5        382  12.525  5400000.0  ['Zephyr Films', 'A Contracorriente Films', 'D...']  gox
9979 rows × 11 columns
```

- On line 346, so save the changes in new excel

```
In [795]: ds.to_csv('C:\\\\Users\\\\El-Wattaneya\\\\Desktop\\\\popular_10000_movies_tmdb.csv', index=False)
```

Respond the questions

- 1) Number of movies after 2020
- 2) Number of movies before 2020
- 3) The first 15 movies with vote greater than 7

```
In [ ]: ###### Q #####
In [271]: #Q1
movies_after2020 = ds.loc[ds['release_date'] > '1/1/2020', 'title']
movies_after2020
Out[271]: 0           The Pope's Exorcist
1           Ant-Man and the Wasp: Quantumania
2           The Super Mario Bros. Movie
3           Ghosted
4           Shazam! Fury of the Gods
...
9974           Silver Skates
9975           Heart Shot
9991   The Witcher Season One Recap: From the Beginning
9992           The Lost King
9993           Good on Paper
Name: title, Length: 2497, dtype: object

In [272]: #Q2
movies_before2020 = ds.loc[ds['release_date'] < '1/1/2020', 'title']
movies_before2020
Out[272]: 37          Guardians of the Galaxy Vol. 2
78    Demon Slayer: Kimetsu no Yaiba Sibling's Bond
82          John Wick: Chapter 3 - Parabellum
83          Guardians of the Galaxy
89    The Forbidden Legend: Sex & Chopsticks 2
...
9995           I, Daniel Blake
9996    Hellboy Animated: Sword of Storms
9997    Return to House on Haunted Hill
9998           My Sister-in-Law's Job
9999           The Bookshop
Name: title, Length: 7479, dtype: object

In [273]: #Q3
vote_gt7 = ds.loc[ds['vote_average'] > 7, 'title'].head(15)
vote_gt7
Out[273]: 0           The Pope's Exorcist
2           The Super Mario Bros. Movie
3           Ghosted
5           Avatar: The Way of Water
6           Guardians of the Galaxy Volume 3
7           Scream VI
8           Creed III
9    Dungeons & Dragons: Honor Among Thieves
10   The Last Kingdom: Seven Kings Must Die
12           John Wick: Chapter 4
15           Puss in Boots: The Last Wish
22  Justice League x RWBY: Super Heroes & Huntsmen...
24           Black Panther: Wakanda Forever
25           Mummies
33           Black Adam
Name: title, dtype: object
```

- 4) The first 15 movies with vote less than 4
- 5) Is there a correlation between budget, popularity and vote average.

```
In [277]: #Q4
vote_lt4 = ds.loc[ds['vote_average'] < 4, 'title'].head(15)
vote_lt4
```

```
Out[277]: 23           Fast X
75      Yuku et la fleur de l'Himalaya
101          The Little Mermaid
128  Transformers: Rise of the Beasts
150      Ryoma! The Prince of Tennis
184          The Flash
209  Shabash Feluda: Gangtokey Gondogol
217          Strays
287  Cocaine Bear: The True Story
339  Spider-Man: Across the Spider-Verse
408          The Rape
471          No Hard Feelings
610          Snake Beauty
641  Soltera, casada, viuda, divorciada
766          Barbie
Name: title, dtype: object
```

```
In [281]: #Q5
correlation = ds[['budget', 'popularity', 'vote_average']].corr()
correlation
```

```
Out[281]:
budget  popularity  vote_average
budget  1.000000  0.143559  0.071361
popularity  0.143559  1.000000  0.041081
vote_average  0.071361  0.041081  1.000000
```

6) The biggest rating for a movie

7) What is the correlation between popularity and original language

```
In [282]: #Q6
average_popularity_by_language = ds.groupby('original_language')['popularity'].mean()
average_popularity_by_language
```

```
Out[282]: original_language
Arabic      43.169000
Basque     55.991333
Bengali     57.061500
Catalan    23.054000
Chinese    21.217533
Czech       13.226000
Danish      19.516133
Dutch       52.384250
Dzongkha    11.366000
English     32.870146
Estonian   197.218000
Finnish     94.556000
French      25.468060
Galician    48.490000
German      28.252476
Greek       14.642750
Hebrew      15.617000
Hindi        15.083750
Hungarian   14.230667
Icelandic   77.614500
Indonesian  24.875706
Irish        12.405000
Italian     23.938814
Japanese    28.910547
Kannada     12.449000
Khmer       10.681000
Korean      26.199825
Macedonian   62.567000
Malayalam    15.762567
Norwegian   34.792969
Norwegian Bokmal 16.935000
Panjabi     10.082000
Persian      27.867333
Polish       43.287036
Portuguese   18.492545
Romanian    54.136000
Russian      24.687174
Serbian      21.419000
Spanish      36.218121
Sundanese    9.660000
Swedish      24.620478
Tagalog      25.523043
Tamil         28.539300
Telugu       27.553111
Thai          28.855877
Turkish      34.906778
Ukrainian    79.999750
Vietnamese   21.236667
cn           22.407372
sh           13.323000
xx           9.991500
Name: popularity, dtype: float64
```

```
In [288]: #Q7
max_value = ds['vote_average'].max()
max_film = ds.loc[ds['vote_average'].idxmax(), 'title']
max_film
```

```
Out[288]: 'Orgasm Lecture 2'
```

- 8) The biggest budget for a movie
- 9) The smallest budget for a movie
- 10) Number of English, Spanish and french movies
- 11) The most language movies were made in
- 12) number of films with production_companies
- 13) What is the lowest company produced films?
- 14) What is the name of the company that produced Evil Dead Rise?
- 15) The largest number in the rate count column
- 16) The lowest number in the rate count column
- 17) What is the vote for “Avatar”?



```
In [290]: #Q8
max_value = ds['budget'].max()
max_film = ds.loc[ds['budget'].idxmax(), 'title']
max_film
```

```
Out[290]: 'Operation Red Sea'
```

```
In [291]: #Q9
min_value = ds['budget'].min()
min_film = ds.loc[ds['budget'].idxmin(), 'title']
min_film
```

```
Out[291]: 'Down'
```

```
In [ ]:
```

```
In [292]: #Q10
languages = ['English', 'Spanish', 'French']
num_movies = ds[ds['original_language'].isin(languages)].shape[0]
num_movies
```

```
Out[292]: 7869
```

```
In [294]: #Q11
most_common_language = ds['original_language'].value_counts().idxmax()
most_common_language
```

```
Out[294]: 'English'
```

```
In [305]: #Q12
num_movies = sum(ds['production_companies'] == 'Marvel Studios')
num_movies
```

```
Out[305]: 0
```

```
In [302]: #Q13
less_common_company = ds['production_companies'].value_counts().idxmin()
less_common_company
```

```
Out[302]: "['Levins-Henenlotter', 'Shapiro-Glickenhaus Entertainment']"
```

```
In [306]: #Q14
movie_title = "Evil Dead Rise"
production_companies = ds[ds['title'] == movie_title]['production_companies']
movie_title
```

```
Out[306]: 'Evil Dead Rise'
```

```
In [309]: #Q15
max_vote_count = ds['vote_count'].max()
max_vote_count
```

```
Out[309]: 33633
```

```
In [310]: #Q16
min_vote_count = ds['vote_count'].min()
min_vote_count
```

```
Out[310]: 0
```

```
In [312]: #Q17
movie_title = "Avatar"
vote_count = ds[ds['title'] == movie_title]['vote_count'].values[0]
vote_count
```

```
Out[312]: 29840
```

- 18) Which movie won popularity = 1107.559**
- 19) What is the original language of “just Retired 2”?**
- 20) What is the number of action movies in the dataset**
- 19) What is the original language of Just Retired 2?**
- 20) What is the number of action movies in the dataset?**
- 21) The number of films that received a zero rating**
- 22) Which is the largest category of films with a high rating.**
- 23) Number of films produced in recent years (eg in the last five years)**



```
In [313]: #Q18
popularity_value = 1107.559
movie_title = ds[ds['popularity'] == popularity_value]['title'].values[0]
movie_title

Out[313]: 'Evil Dead Rise'

In [315]: #Q19
movie_title = "Just Retired 2"
original_language = ds[ds['title'] == movie_title]['original_language'].values[0]
original_language

Out[315]: 'French'

In [316]: #Q20
num_action_movies = ds[ds['genres'].str.contains('action', case=False)].shape[0]
num_action_movies

Out[316]: 2687

In [318]: #Q21
num_movies_with_zero_vote_count = ds[ds['vote_count'] == 0].shape[0]
num_movies_with_zero_vote_count

Out[318]: 229

In [319]: #Q22
highest_voted_genre = ds.groupby('genres')['vote_average'].mean().idxmax()
highest_voted_genre

Out[319]: "[Fantasy", 'Drama', 'Crime']"

In [322]: #Q23
from datetime import datetime
current_year = datetime.now().year
five_years_ago = current_year - 5

num_movies_last_five_years = ds[ds['release_date'].dt.year >= five_years_ago].shape[0]
num_movies_last_five_years

Out[322]: 3435
```

24) What are the names of all the production companies?

25) what is the sorting for IDs

```
In [326]: #B#4
# استخرج أسماء شركات الانتاج المميزة
production_companies = ds['production_companies'].unique()

# عرض أسماء شركات الانتاج
for company in production_companies:
    print(company)

[Screen Gems, '2.0 Entertainment', 'Jesus & Mary', 'Worldwide Katz', 'Loyola Productions', 'FFILME.RO']
['Marvel Studios', 'Kevin Feige Productions']
['Universal Pictures', 'Illumination', 'Nintendo']
['Skydance Media', 'Apple Studios']
['New Line Cinema', 'The Safran Company', 'DC Films', 'Warner Bros. Pictures']
['20th Century Studios', 'Lightstorm Entertainment']
['Radio Silence', 'Project X Entertainment', 'Spyglass Media Group', 'Paramount']
['Metro-Goldwyn-Mayer', 'Proximity Media', 'Balboa Productions', 'Outlier Society Productions', 'Chartoff-Winkler Productions']
['Entertainment One', 'Paramount', 'Allspark Pictures', 'Sierra/Affinity']
['Carnival Films']
['Walt Disney Pictures', 'Whitaker Entertainment']
['Thunder Road', '87Eleven', 'Summit Entertainment', 'Studio Babelsberg', 'Lionsgate']
['New Line Cinema', 'Ghost House Pictures', 'Renaissance Pictures', 'Pacific Renaissance Pictures', 'Wild Atlantic Pictures']
['Universal Pictures', 'Brownstone Productions', 'Lord Miller Productions']
['DreamWorks Animation', 'Universal Pictures']
['Happy Madison Productions', 'Echo Films', 'Vinson Films', 'Endgame Entertainment']
['Inox Filmes', 'Nolita']

In [328]: #Q25
# Sort the IDs in ascending order
sorted_data = ds.sort_values('id')

# Display the sorted IDs
sorted_ids = sorted_data['id'].tolist()
print(sorted_ids)

[5, 11, 12, 13, 14, 15, 16, 18, 19, 22, 24, 25, 27, 28, 33, 35, 38, 55, 58, 59, 62, 63, 64, 65, 66, 68, 69, 70, 71, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 85, 87, 88, 89, 90, 93, 95, 96, 97, 98, 101, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 115, 116, 117, 118, 120, 121, 122, 123, 128, 129, 134, 136, 137, 138, 141, 142, 143, 144, 145, 146, 147, 149, 150, 152, 153, 154, 155, 157, 158, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 172, 173, 174, 175, 176, 178, 179, 180, 181, 185, 186, 187, 189, 192, 194, 196, 197, 199, 200, 201, 203, 205, 207, 213, 214, 215, 217, 218, 219, 221, 223, 226, 227, 229, 231, 233, 234, 235, 236, 238, 239, 240, 241, 242, 243, 244, 247, 251, 252, 253, 254, 257, 262, 268, 269, 272, 275, 277, 278, 279, 280, 284, 285, 287, 288, 289, 290, 293, 296, 297, 298, 301, 306, 310, 311, 314, 319, 320, 322, 326, 329, 332, 334, 335, 336, 338, 345, 346, 348, 350, 363, 364, 377, 378, 379, 380, 387, 388, 389, 391, 392, 393, 395, 396, 401, 402, 403, 404, 405, 406, 408, 409, 411, 414, 415, 421, 422, 423, 424, 425, 426, 429, 431, 433, 435, 437, 439, 440, 451, 452, 453, 454, 455, 458, 462, 468, 470, 475, 479, 482, 483, 488, 489, 490, 492, 496, 497, 500, 503, 504, 506, 508, 509, 510, 521, 524, 525, 526, 530, 531, 533, 534, 535, 539, 540, 542, 544, 547, 550, 551, 553, 557, 558, 559, 561, 562, 563, 564, 565, 567, 568, 571, 573, 576, 578, 579, 580, 581, 582, 583, 585, 586, 587, 588, 591, 593, 594, 595, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 612, 613, 615, 616, 617, 619, 620, 621, 622, 623, 624, 627, 628, 629, 630, 634, 635, 637, 638, 639, 640, 641, 642, 644, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 657, 658, 659, 660, 663, 664, 665, 667, 668, 670, 671, 672, 673, 674, 675, 676, 679, 680, 681, 682, 686, 687, 688, 691, 692, 693, 694, 698, 699, 700, 702, 703, 705, 707, 708, 709, 710, 711, 712, 713, 714, 744, 745, 746, 747, 752, 754, 755, 756, 759, 762, 763, 764, 765, 766, 767, 768, 769, 770, 772, 773, 775, 780, 782, 783, 786, 787, 788, 790, 791, 792, 793, 794, 795, 796, 797, 801, 802, 804, 805, 806, 807, 808, 809, 810, 812, 813, 814, 816, 817, 818, 819, 820, 821, 823, 824, 826, 829, 830, 832, 834, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 849, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 871, 872, 873, 877, 878, 881, 884, 887, 890, 893, 896, 899, 902, 905, 908, 911, 914, 917, 920, 923, 926, 929, 932, 935, 938, 941, 944, 947, 950, 953, 956, 959, 962, 965, 968, 971, 974, 977, 980, 983, 986, 989, 992, 995, 998, 1001, 1004, 1007, 1010, 1013, 1016, 1019, 1022, 1025, 1028, 1031, 1034, 1037, 1040, 1043, 1046, 1049, 1052, 1055, 1058, 1061, 1064, 1067, 1070, 1073, 1076, 1079, 1082, 1085, 1088, 1091, 1094, 1097, 1098, 1101, 1104, 1107, 1110, 1113, 1116, 1119, 1122, 1125, 1128, 1131, 1134, 1137, 1140, 1143, 1146, 1149, 1152, 1155, 1158, 1161, 1164, 1167, 1170, 1173, 1176, 1179, 1182, 1185, 1188, 1191, 1194, 1197, 1199, 1202, 1205, 1208, 1211, 1214, 1217, 1220, 1223, 1226, 1229, 1232, 1235, 1238, 1241, 1244, 1247, 1251, 1254, 1257, 1262, 1268, 1269, 1272, 1275, 1277, 1278, 1281, 1284, 1287, 1288, 1291, 1294, 1297, 1300, 1303, 1306, 1309, 1312, 1315, 1318, 1321, 1324, 1327, 1330, 1333, 1336, 1339, 1342, 1345, 1348, 1351, 1354, 1357, 1360, 1363, 1366, 1369, 1372, 1375, 1378, 1381, 1384, 1387, 1390, 1393, 1396, 1399, 1402, 1405, 1408, 1411, 1414, 1417, 1420, 1423, 1426, 1429, 1432, 1435, 1438, 1441, 1444, 1447, 1450, 1453, 1456, 1459, 1462, 1465, 1468, 1471, 1474, 1477, 1480, 1483, 1486, 1489, 1492, 1495, 1498, 1501, 1504, 1507, 1510, 1513, 1516, 1519, 1522, 1525, 1528, 1531, 1534, 1537, 1540, 1543, 1546, 1549, 1552, 1555, 1558, 1561, 1564, 1567, 1570, 1573, 1576, 1579, 1582, 1585, 1588, 1591, 1594, 1597, 1598, 1599, 1600, 1601, 1604, 1607, 1610, 1613, 1616, 1619, 1622, 1625, 1628, 1631, 1634, 1637, 1640, 1643, 1646, 1649, 1652, 1655, 1658, 1661, 1664, 1667, 1670, 1673, 1676, 1679, 1682, 1685, 1688, 1691, 1694, 1697, 1700, 1703, 1705, 1707, 1708, 1711, 1712, 1713, 1714, 1744, 1745, 1746, 1747, 1752, 1754, 1755, 1756, 1759, 1762, 1763, 1764, 1765, 1766, 1767, 1768, 1769, 1770, 1772, 1773, 1775, 1780, 1782, 1783, 1786, 1787, 1788, 1790, 1791, 1792, 1793, 1794, 1795, 1796, 1797, 1801, 1802, 1804, 1805, 1806, 1807, 1808, 1809, 1810, 1812, 1813, 1814, 1816, 1817, 1818, 1819, 1820, 1821, 1823, 1824, 1826, 1829, 1830, 1832, 1834, 1837, 1838, 1839, 1840, 1841, 1842, 1843, 1844, 1845, 1846, 1847, 1849, 1853, 1854, 1855, 1856, 1857, 1858, 1859, 1860, 1861, 1862, 1863, 1864, 1865, 1866, 1867, 1868, 1869, 1871, 1872, 1873, 1877, 1878, 1881, 1884, 1887, 1890, 1893, 1896, 1899, 1902, 1905, 1908, 1911, 1914, 1917, 1920, 1923, 1926, 1929, 1932, 1935, 1938, 1941, 1944, 1947, 1950, 1953, 1956, 1959, 1962, 1965, 1968, 1971, 1974, 1977, 1980, 1983, 1986, 1989, 1992, 1995, 1998, 2001, 2004, 2007, 2010, 2013, 2016, 2019, 2022, 2025, 2028, 2031, 2034, 2037, 2040, 2043, 2046, 2049, 2052, 2055, 2058, 2061, 2064, 2067, 2070, 2073, 2076, 2079, 2082, 2085, 2088, 2091, 2094, 2097, 2100, 2103, 2106, 2109, 2112, 2115, 2118, 2121, 2124, 2127, 2130, 2133, 2136, 2139, 2142, 2145, 2148, 2151, 2154, 2157, 2160, 2163, 2166, 2169, 2172, 2175, 2178, 2181, 2184, 2187, 2190, 2193, 2196, 2199, 2202, 2205, 2208, 2211, 2214, 2217, 2220, 2223, 2226, 2229, 2232, 2235, 2238, 2241, 2244, 2247, 2250, 2253, 2256, 2259, 2262, 2265, 2268, 2271, 2274, 2277, 2280, 2283, 2286, 2289, 2292, 2295, 2298, 2301, 2306, 2310, 2313, 2316, 2319, 2322, 2325, 2328, 2331, 2334, 2337, 2340, 2343, 2346, 2349, 2352, 2355, 2358, 2361, 2364, 2367, 2370, 2373, 2376, 2379, 2382, 2385, 2388, 2391, 2394, 2397, 2400, 2403, 2406, 2409, 2412, 2415, 2418, 2421, 2424, 2427, 2430, 2433, 2436, 2439, 2442, 2445, 2448, 2451, 2454, 2457, 2460, 2463, 2466, 2469, 2472, 2475, 2478, 2481, 2484, 2487, 2490, 2493, 2496, 2499, 2502, 2505, 2508, 2511, 2514, 2517, 2520, 2523, 2526, 2529, 2532, 2535, 2538, 2541, 2544, 2547, 2550, 2553, 2556, 2559, 2562, 2565, 2568, 2571, 2574, 2577, 2580, 2583, 2586, 2589, 2592, 2595, 2598, 2601, 2604, 2607, 2610, 2613, 2616, 2619, 2622, 2625, 2628, 2631, 2634, 2637, 2640, 2643, 2646, 2649, 2652, 2655, 2658, 2661, 2664, 2667, 2670, 2673, 2676, 2679, 2682, 2685, 2688, 2691, 2694, 2697, 2700, 2703, 2705, 2707, 2708, 2711, 2712, 2713, 2714, 2744, 2745, 2746, 2747, 2752, 2754, 2755, 2756, 2759, 2762, 2763, 2764, 2765, 2766, 2767, 2768, 2769, 2770, 2772, 2773, 2775, 2780, 2782, 2783, 2786, 2787, 2788, 2790, 2791, 2792, 2793, 2794, 2795, 2796, 2797, 2801, 2802, 2804, 2805, 2806, 2807, 2808, 2809, 2810, 2812, 2813, 2814, 2816, 2817, 2818, 2819, 2820, 2821, 2823, 2824, 2826, 2828, 2829, 2830, 2832, 2834, 2836, 2838, 2839, 2840, 2841, 2843, 2845, 2846, 2847, 2849, 2853, 2855, 2856, 2857, 2858, 2859, 2860, 2861, 2862, 2863, 2864, 2865, 2866, 2867, 2868, 2869, 2871, 2872, 2873, 2877, 2878, 2881, 2884, 2887, 2890, 2893, 2896, 2899, 2902, 2905, 2908, 2911, 2914, 2917, 2920, 2923, 2926, 2929, 2932, 2935, 2938, 2941, 2944, 2947, 2950, 2953, 2956, 2959, 2962, 2965, 2968, 2971, 2974, 2977, 2980, 2983, 2986, 2989, 2992, 2995, 2998, 3001, 3004, 3007, 3010, 3013, 3016, 3019, 3022, 3025, 3028, 3031, 3034, 3037, 3040, 3043, 3046, 3049, 3052, 3055, 3058, 3061, 3064, 3067, 3070, 3073, 3076, 3079, 3082, 3085, 3088, 3091, 3094, 3097, 3100, 3103, 3106, 3109, 3112, 3115, 3118, 3121, 3124, 3127, 3130, 3133, 3136, 3139, 3142, 3145, 3148, 3151, 3154, 3157, 3160, 3163, 3166, 3169, 3172, 3175, 3178, 3181, 3184, 3187, 3190, 3193, 3196, 3199, 3202, 3205, 3208, 3211, 3214, 3217, 3220, 3223, 3226, 3229, 3232, 3235, 3238, 3241, 3244, 3247, 3250, 3253, 3256, 3259, 3262, 3265, 3268, 3271, 3274, 3277, 3280, 3283, 3286, 3289, 3292, 3295, 3298, 3301, 3304, 3307, 3310, 3313, 3316, 3319, 3322, 3325, 3328, 3331, 3334, 3337, 3340, 3343, 3346, 3349, 3352, 3355, 3358, 3361, 3364, 3367, 3370, 3373, 3376, 3379, 3382, 3385, 3388, 3391, 3394, 3397, 3390, 3393, 3396, 3399, 3402, 3405, 3408, 3411, 3414, 3417, 3420, 3423, 3426, 3429, 3432, 3435, 3438, 3441, 3444, 3447, 3450, 3453, 3456, 3459, 3462, 3465, 3468, 3471, 3474, 3477, 3480, 3483, 3486, 3489, 3492, 3495, 3498, 3501, 3504, 3507, 3510, 3513, 3516, 3519, 3522, 3525, 3528, 3531, 3534, 3537, 3540, 3543, 3546, 3549, 3552, 3555, 3558, 3561, 3564, 3567, 3570, 3573, 3576, 3579, 3582, 3585, 3588, 3591, 3594, 3597, 3598, 3599, 3600, 3601, 3604, 3607, 3610, 3613, 3616, 3619, 3622, 3625, 3628, 3631, 3634, 3637, 3640, 3643, 3646, 3649, 3652, 3655, 3658, 3661, 3664, 3667, 3670, 3673, 3676, 3679, 3682, 3685, 3688, 3691, 3694, 3697, 3690, 3693, 3696, 3699, 3702, 3705, 3708, 3711, 3712, 3713, 3714, 3744, 3745, 3746, 3747, 3752, 3754, 3755, 3756, 3759, 3762, 3763, 3764, 3765, 3766, 3767, 3768, 3769, 3770, 3772, 3773, 3775, 3780, 3782, 3783, 3786, 3787, 3788, 3790, 3791, 3792, 3793, 3794, 3795, 3796, 3797, 3801, 3802, 3804, 3805, 3806, 3807, 3808, 3810, 3812, 3813, 3814, 3816, 3817, 3818, 3819, 3820, 3821, 3823, 3824, 3826, 3828, 3829, 3830, 3832, 3834, 3837, 3838, 3839, 3840, 3841, 3842, 3843, 3844, 3845, 3846, 3847, 3849, 3853, 3855, 3856, 3857, 3858, 3859, 3860, 3861, 3862, 3863, 3864, 3865, 3866, 3867, 3868, 3869, 3871, 3872, 3873, 3877, 3878, 3881, 3884, 3887, 3890, 3893, 3896, 3899, 3902, 3905, 3908, 3911, 3914, 3917, 3920, 3923, 3926, 3929, 3932, 3935, 3938, 3941, 3944, 3947, 3950, 3953, 3956, 3959, 3962, 3965, 3968, 3971, 3974, 3977, 3980, 3983, 3986, 3989, 3992, 3995, 3998, 4001, 4004, 4007, 4010, 4013, 4016, 4019, 4022, 4025, 4028, 4031, 4034, 4037, 4040, 4043, 4046, 4049, 4052, 4055, 4058, 4061, 4064, 4067, 4070, 4073, 4076, 4079, 4082, 4085, 4088, 4091, 4094, 4097, 4090, 4093, 4096, 4099, 4102, 4105, 4108, 4111, 4114, 4117, 4120, 4123, 4126, 4129, 4132, 4135, 4138, 4141, 4144, 4147, 4150, 4153, 4156, 4159, 4162, 4165, 4168, 4171, 4174, 4177, 4180, 4183, 4186, 4189, 4192, 4195, 4198, 4201, 4204, 4207, 4210, 4213, 4216, 4219, 4222, 4225, 4228, 4231, 4234, 4237, 4240, 4243, 4246, 4249, 4252, 4255, 4258, 4261, 4264, 4267, 4270, 4273, 4276, 4279, 4282, 4285, 4288, 4291, 4294, 4297, 4290, 4293, 4296, 4299, 4302, 4305, 4308, 4311, 4314, 4317, 4320, 4323, 4326, 4329, 4332, 4335, 4338, 4341, 4344, 4347, 4350, 4353, 4356, 4359, 4362, 4365, 4368, 4371, 4374, 4377, 4380, 4383, 4386, 4389, 4392, 4395, 4398, 4401, 4404, 4407, 4410, 4413, 4416, 4419, 4422, 4425, 4428, 4431, 4434, 4437, 4440, 4443, 4446, 4449, 4452, 4455, 4458, 4461, 4464, 4467, 4470, 4473, 4476, 4479, 4482, 4485, 4488, 4491, 4494, 4497, 4500, 4503, 4506, 4509, 4512, 4515, 4518, 4521, 4524, 4527, 4530, 4533, 4536, 4539, 4542, 4545, 4548, 4551, 4554, 4557, 4560, 4563, 4566, 4569, 4572, 4575, 4578, 4581, 4584, 4587, 4590, 4593, 4596, 4599, 4602, 4605, 4608, 4611, 4614, 4617, 4620, 4623, 4626, 4629, 4632, 4635, 4638, 4641, 4644, 4646, 4649, 4652, 4655, 4658, 4661, 4664, 4667, 4670, 4673, 4676, 4679, 4682, 4685, 4688, 4691, 4694, 4697, 4700, 4703, 4705, 4708, 4711, 4712, 4713, 4714, 4744, 4745, 4746, 4747, 4752, 4754, 4755, 4756, 4759, 4762, 4763, 4764, 4765, 4766, 4767, 4768, 4769, 4770, 4772, 4773, 4775, 4780, 4782, 4783, 4786, 4787, 4788, 4790, 4791, 4792, 4793, 4794, 4795, 4796, 4797, 4801, 4802, 4804, 4805, 4806, 4807, 4808, 4810, 4812, 4813, 4814, 4816, 4817, 4818, 4819, 4820, 4821, 4823, 4824, 4826, 4828, 4829, 4830, 4832, 4834, 4837, 4838, 4839, 4840, 4841, 4842, 4843, 4844, 4845, 4846, 4847, 4849, 4851, 4853, 4854, 4855, 4856, 4857, 4858, 4859, 4860, 4861, 4862, 4863, 4864, 4865, 4866, 4867, 4868, 4869, 4871, 4872, 4873, 4877, 4878, 4881, 4884, 4887, 4890, 4893, 4896, 4899, 4902, 4905, 4908, 4911, 4914, 4917, 4920, 4923, 4926, 4929, 4932, 4935, 4938, 4941, 4944, 4947, 4950, 4953, 4956, 4959, 4962, 4965, 4968, 4971, 4974, 4977, 4980, 4983, 4986, 4989, 4992, 4995, 4998, 5001, 5004, 5007, 5010, 5013, 5016, 5019, 5022, 5025, 5028, 5031, 5034, 5037, 5040, 5043, 5046, 5049, 5052, 5055, 5058, 5061, 5064, 5067, 5070, 5073, 5076, 5079, 5082, 5085, 5088, 5091, 5094, 5097, 5090, 5093, 5096, 5099, 5102, 5105, 5108, 5111, 5114, 5117, 5120, 5123, 5126, 5129
```

26) What are the high budget low rate movies?

```
In [331]: #Q26
high_budget_low_rate = ds.query('budget > 100000000 and vote_average < 5')
high_budget_low_rate
```

Out[331]:

	id	title	release_date	genres	original_language	vote_average	vote_count	popularity	budget	production_companies	vote_category	
	23	385687	Fast X	2023-05-17	['Action', 'Crime', 'Thriller']	English	0.0	0	732,647	340000000.0	['Universal Pictures', 'Original Film', 'One R...']	bad
	128	667538	Transformers: Rise of the Beasts	2023-06-07	['Action', 'Adventure', 'Science Fiction']	English	0.0	0	222,430	200000000.0	['Skydance Media', 'Paramount', 'di Bonaventur...']	bad
	184	298618	The Flash	2023-06-14	['Science Fiction', 'Action', 'Adventure']	English	0.0	0	123,659	220000000.0	['Warner Bros. Pictures', 'Double Dream', 'Thei...']	bad
	1142	10196	The Last Airbender	2010-06-30	['Action', 'Adventure', 'Fantasy']	English	4.7	3585	36,621	150000000.0	['Paramount', 'Nickelodeon Movies', 'Blinding ...']	bad
	1266	335977	Indiana Jones and the Dial of Destiny	2023-06-28	['Adventure', 'Action']	English	0.0	0	42,876	294700000.0	['Lucasfilm Ltd.', 'Walt Disney Pictures', 'Pa...']	bad
	1849	166424	Fantastic Four	2015-08-05	['Action', 'Adventure', 'Science Fiction']	English	4.4	5525	32,330	120000000.0	['Moving Picture Company', 'Marvel Entertainment...']	bad
	2408	575264	Mission: Impossible - Dead Reckoning Part One	2023-07-10	['Action', 'Adventure', 'Thriller']	English	0.0	0	30,128	290000000.0	['Paramount', 'Skydance Media', 'New Republic ...']	bad
	3476	565770	Blue Beetle	2023-08-16	['Action', 'Science Fiction']	English	0.0	0	22,946	120000000.0	['Warner Bros. Pictures', 'The Safran Company']	bad
	3483	415	Batman & Robin	1997-06-20	['Action', 'Science Fiction', 'Adventure', 'Co...']	English	4.3	4485	24,492	125000000.0	['DC Comics', 'PolyGram Filmed Entertainment', ...]	bad
	4798	1639	Speed 2: Cruise Control	1997-06-13	['Action', 'Adventure', 'Thriller']	English	4.6	1429	20,277	160000000.0	['Blue Tulip Productions', '20th Century Fox']	bad
	4953	393209	Avatar 5	2028-12-20	['Action', 'Adventure', 'Science Fiction']	English	0.0	0	16,033	250000000.0	['Lightstorm Entertainment', '20th Century Stu...']	bad
	3586	787699	Wonka	2023-12-13	['Adventure', 'Comedy', 'Family']	English	0.0	0	14,106	125000000.0	['Warner Bros. Pictures', 'Heyday Films']	bad
	3587	406420	Killers of the Flower Moon	2023-10-06	['Crime', 'Drama', 'Thriller']	English	0.0	0	14,445	200000000.0	['Appian Way', 'Imperative Entertainment', 'Si...']	bad
	5939	83533	Avatar 3	2024-12-18	['Action', 'Science Fiction', 'Adventure']	English	0.0	0	11,996	250000000.0	['Lightstorm Entertainment', '20th Century Stu...']	bad



27) What are the high rate low budget movies?

28) What are the films that took a rating = 8 ?

In [332]: #Q27
high_rate_low_budget = ds.query('vote_average > 8 and budget < 10000000')
high_rate_low_budget

		id	title	release_date	genres	original_language	vote_average	vote_count	popularity	budget	production_companies	vote_category
56	820067	The Quintessential Quintuplets Movie		2022-05-20	['Animation', 'Comedy', 'Romance']	Japanese	8.8	235	287.143	370000.0	['DAX Production', 'Pony Canyon', 'Nichion', ...]	good
164	637920	Miracle in Cell No. 7		2019-10-10	['Drama']	Turkish	8.3	4125	128.270	370000.0	['Lanistar Media', 'Motion Content Group', 'CJ...']	good
192	372058	Your Name.		2016-08-26	['Romance', 'Animation', 'Drama']	Japanese	8.5	9777	139.896	370000.0	['CoMix Wave Films', 'TOHO', 'KADOKAWA', 'East...']	good
207	851644	20th Century Girl		2022-10-06	['Romance', 'Drama']	Korean	8.4	448	138.051	110.0	['Yong Film']	good
237	400928	Gifted		2017-04-12	['Drama', 'Comedy']	English	8.1	4788	98.766	700000.0	['Grade A Entertainment', 'Dayday Films', 'Fil...']	good
...
9814	15804	A Brighter Summer Day		1991-07-27	['Crime', 'Drama', 'Romance']	Chinese	8.3	245	11.905	370000.0	['Yang & His Gang Filmmakers']	good
9817	823754	Bo Burnham: Inside		2021-07-22	['Comedy', 'Drama']	English	8.2	353	10.014	370000.0	[]	good
9847	612304	Divaldo: O Mensageiro da Paz		2019-09-12	['Drama']	Portuguese	8.3	59	16.640	370000.0	['20th Century Fox Brazil', 'CINE Cinematogr...']	good
9861	654299	Out of the Clear Blue Sky		2019-12-24	['Comedy', 'Drama', 'Music']	Spanish	8.2	314	11.296	370000.0	['Esparza Caldera', 'La Victoria Films']	good
9883	532753	Dying to Survive		2018-07-06	['Drama', 'Comedy']	Chinese	8.1	117	10.290	370000.0	['Huanxi Media Group', 'Wanda Pictures', 'Alib...']	good

208 rows × 11 columns



In [339]: #Q28
movies_with_rating_8 = ds[ds['vote_average'] == 8]['title']
movies_with_rating_8

22	Justice League x RWBY: Super Heroes & Huntsmen...
68	Spider-Man: No Way Home
78	Demon Slayer: Kimetsu no Yaiba Sibling's Bond
103	Suzume
133	Swapping: A Divorce Trip Between Two Couples
...	
9207	La Strada
9283	Autumn Sonata
9380	Date A Live: Mayuri Judgment
9437	Inazuma Eleven Go vs. Danball Senki W
9809	La leyenda del Charro Negro
Name: title, Length: 113, dtype: object	

29) The most recently released movie in the RELEASE_DATA column

30) Oldest released movie in the RELEASE_DATA column

```
In [340]: #Q29
min_release_date = ds['release_date'].min()
movie_with_min_release_date = ds[ds['release_date'] == min_release_date]['title']
movie_with_min_release_date
```

```
Out[340]: 5527    A Trip to the Moon
Name: title, dtype: object
```

```
In [341]: #Q30
max_release_date = ds['release_date'].max()
movie_with_max_release_date = ds[ds['release_date'] == max_release_date]['title']
movie_with_max_release_date
```

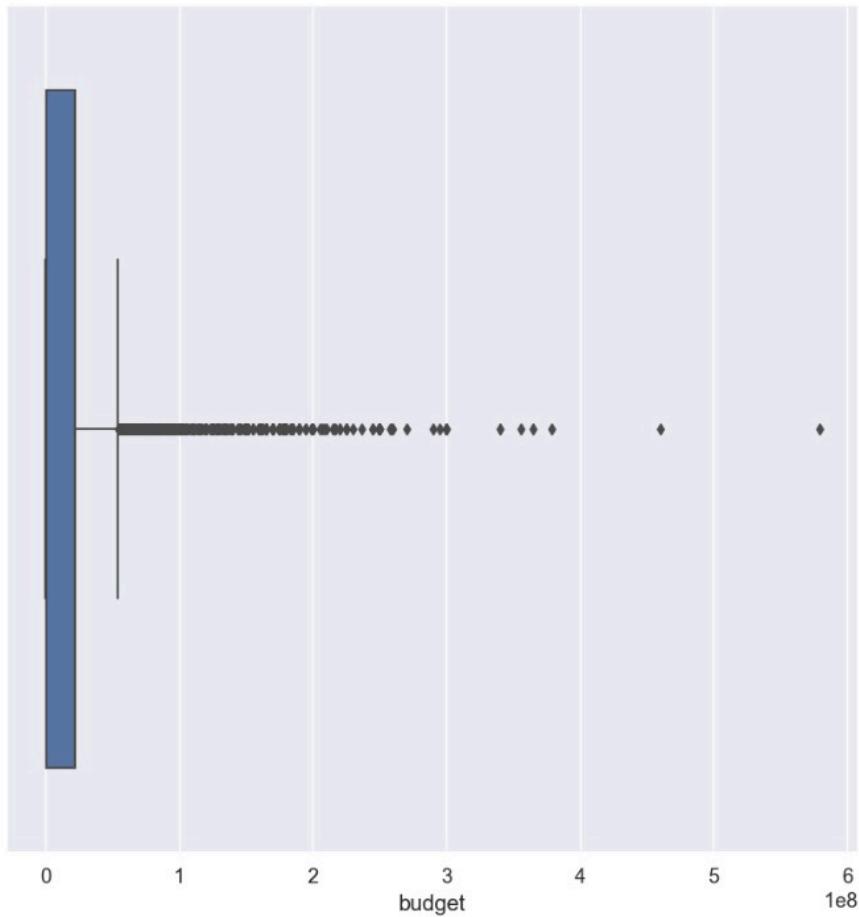
```
Out[341]: 4953    Avatar 5
Name: title, dtype: object
```

Visualization

1) Before handling the outlier

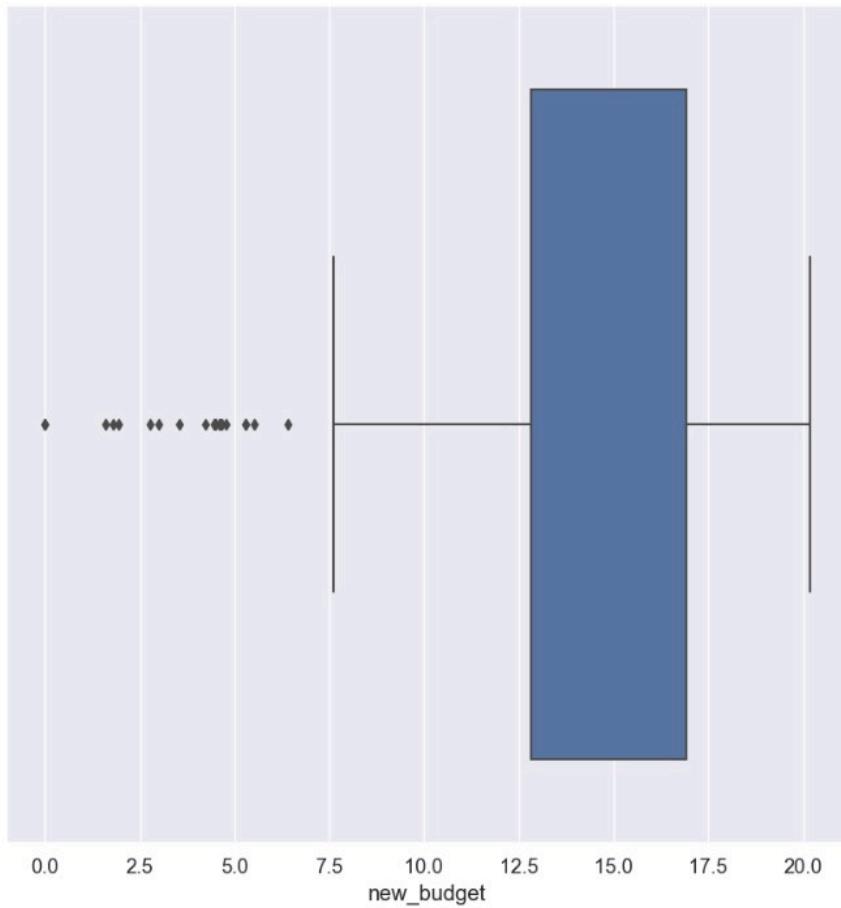
```
In [786]: import seaborn as sns  
sns.boxplot(x="budget", data=ds)
```

```
Out[786]: <Axes: xlabel='budget'>
```



2) After handling the outlier

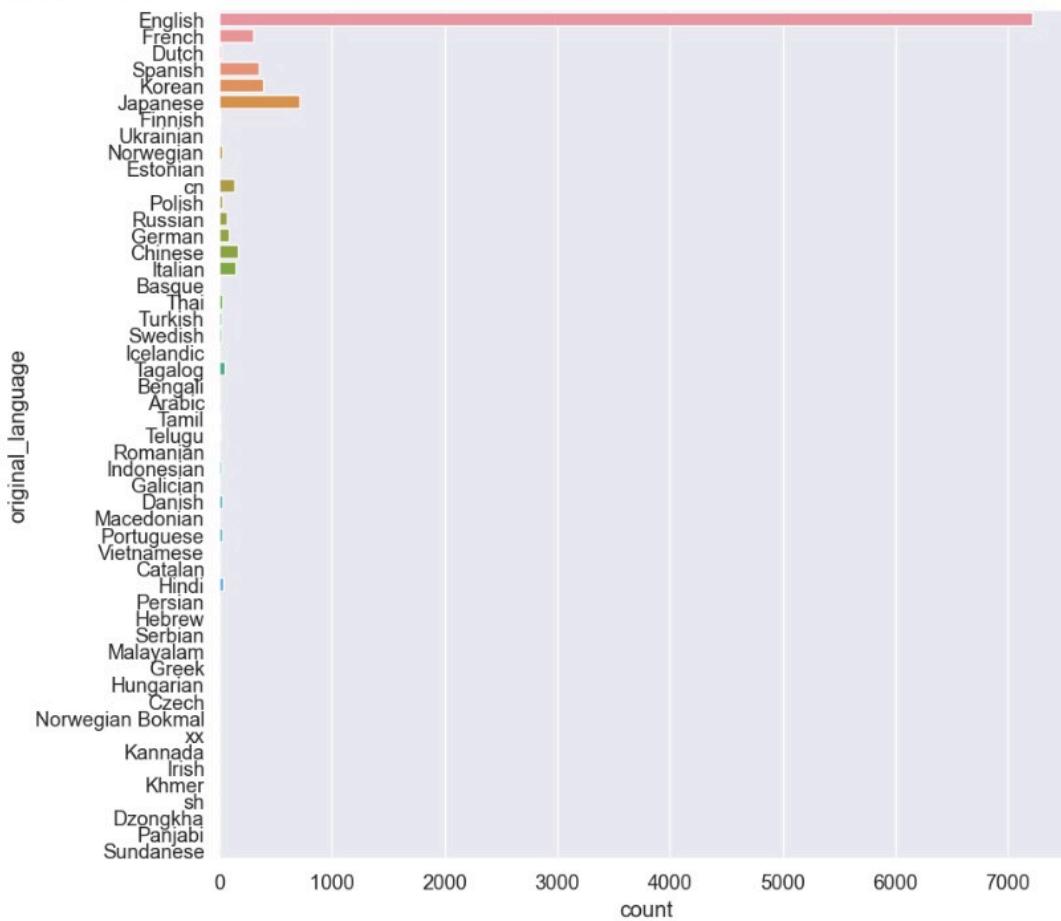
```
In [788]: import seaborn as sns  
sns.boxplot (x="new_budget", data=ds)  
Out[788]: <Axes: xlabel='new_budget'>
```



3) Show the most used language

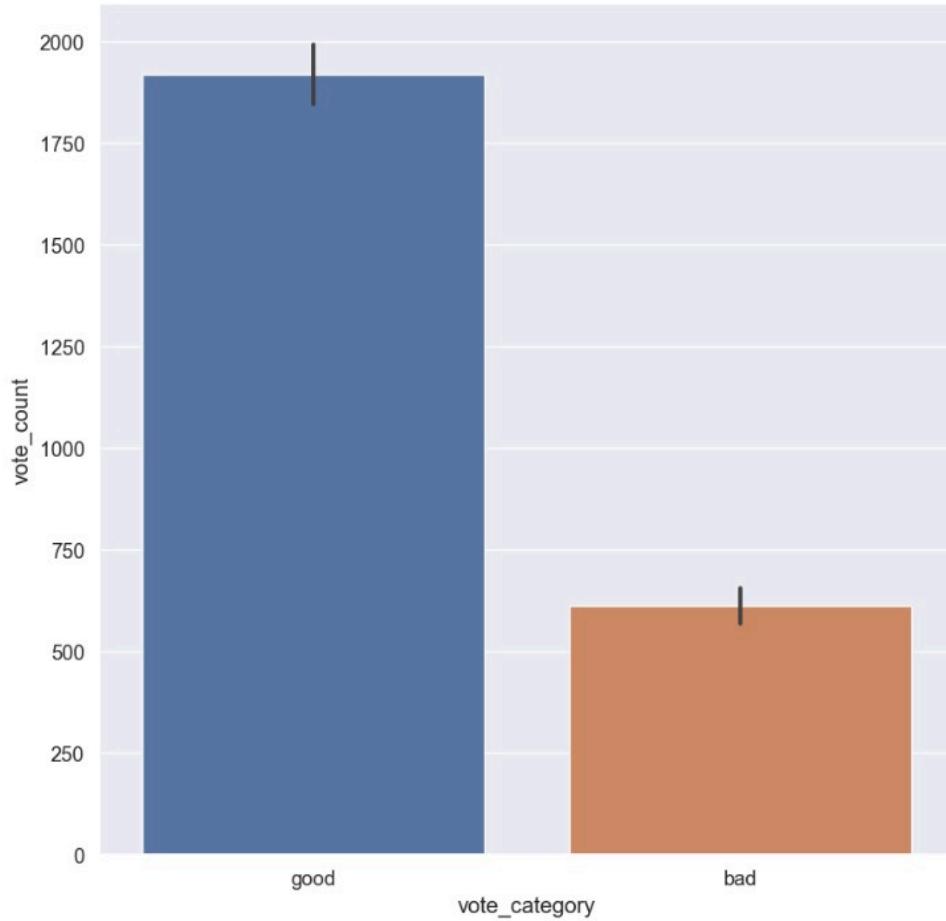
```
In [797]: sns.set(rc={'figure.figsize':[10,10]},font_scale=1.2)
sns.countplot(y='original_language',data=ds)

Out[797]: <Axes: xlabel='count', ylabel='original_language'>
```

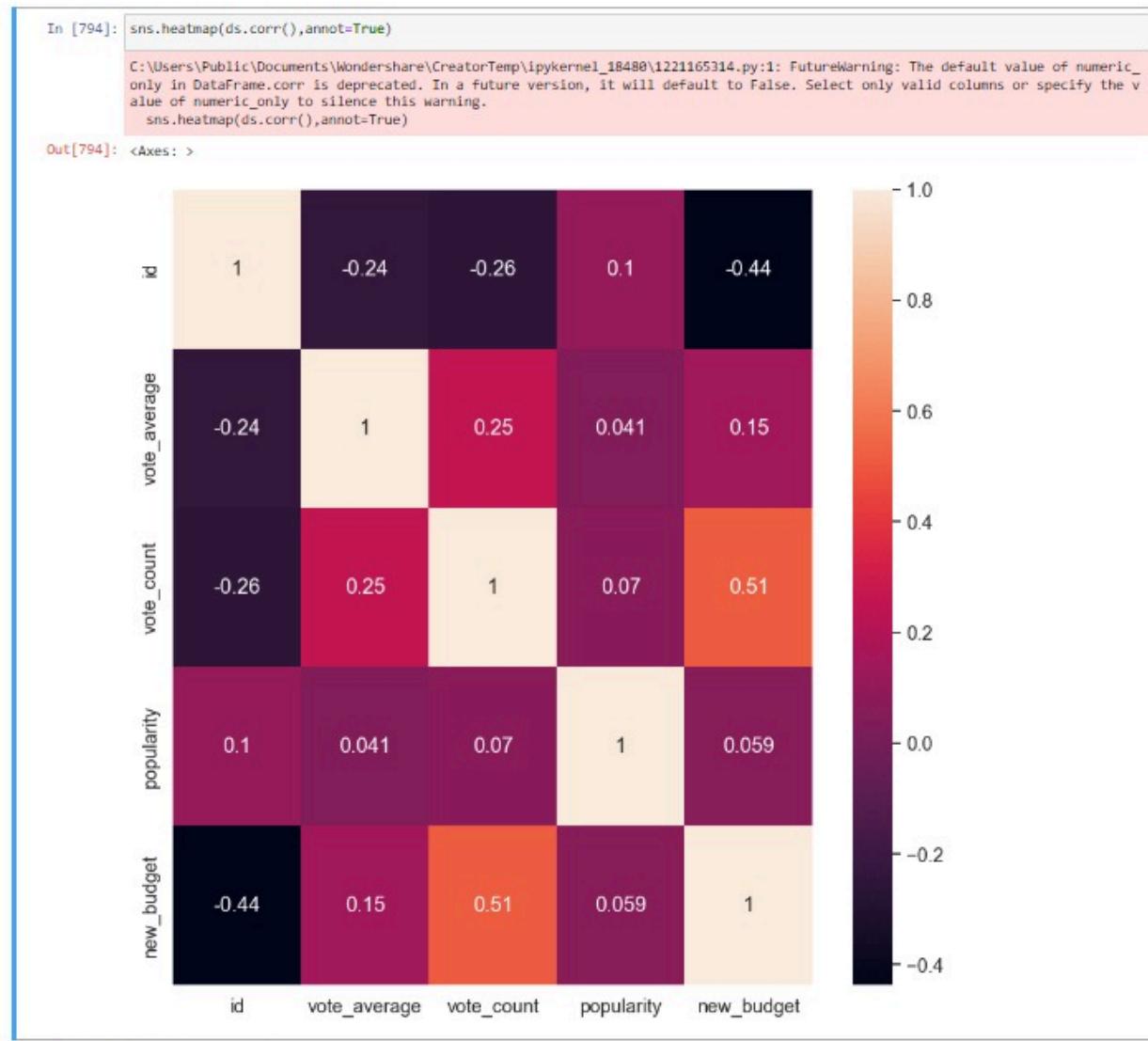


4) Show relation between vote_category & vote_count

```
In [793]: sns.barplot(x='vote_category',y='vote_count',data=ds,estimator=np.mean)
Out[793]: <Axes: xlabel='vote_category', ylabel='vote_count'>
```



5) show the heatmap correlation

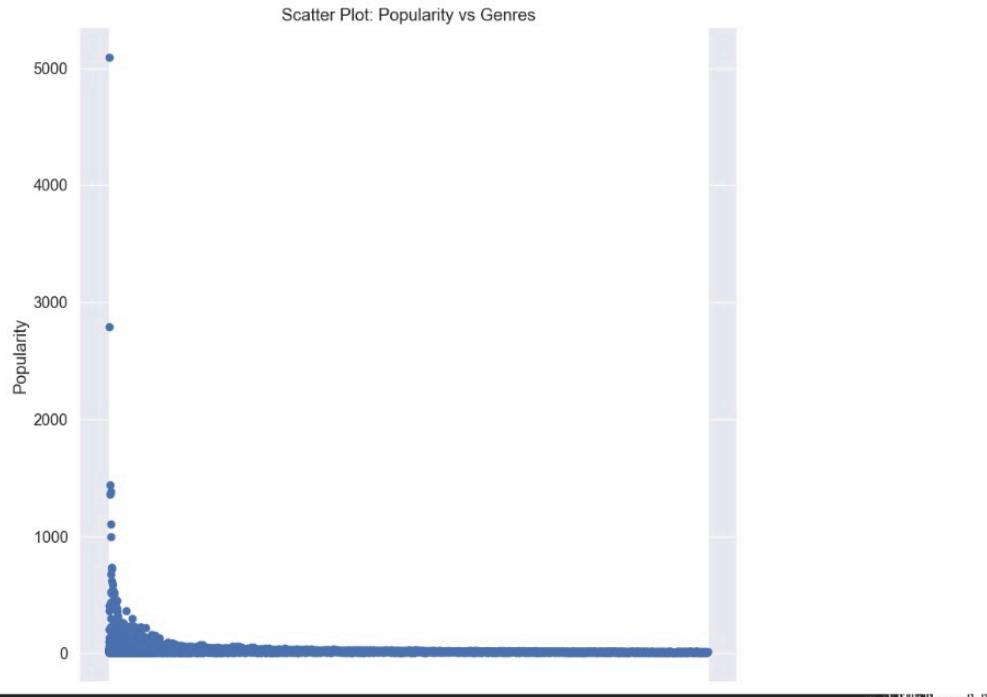


In [801]: `ds['vote_category'].value_counts()`

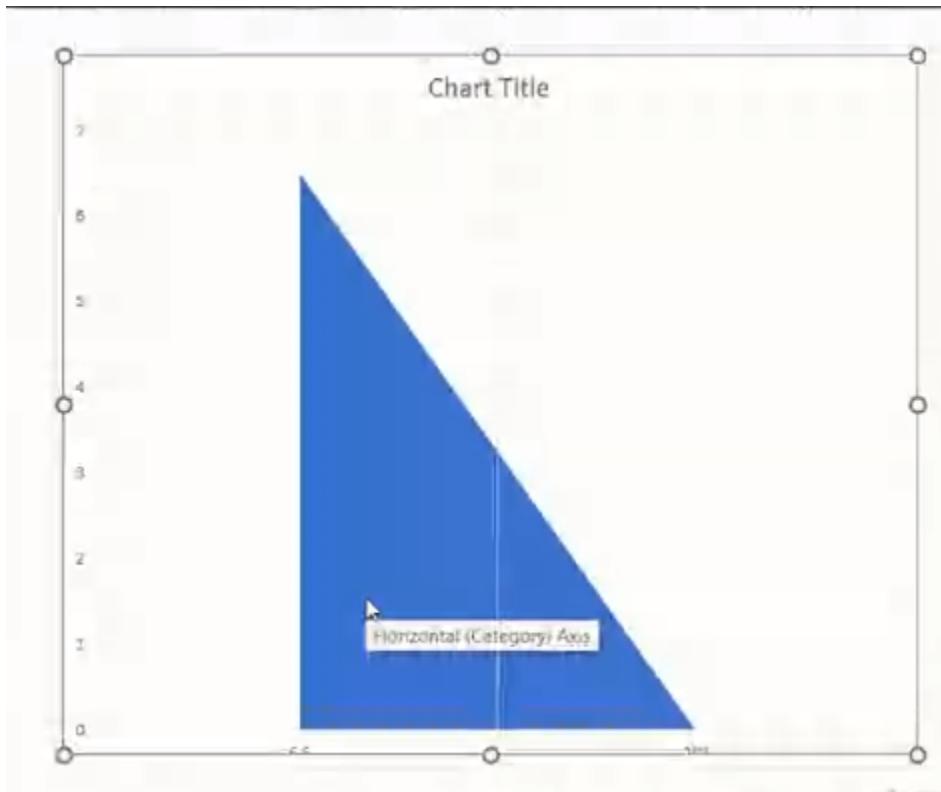
Out[801]: good 7010
bad 2969
Name: vote_category, dtype: int64

6) Show the scatter ploting between popularity & genres

```
plt.scatter(df['genres'], df['popularity'])
plt.xlabel('Genres')
plt.ylabel('Popularity')
plt.title('Scatter Plot: Popularity vs Genres')
plt.show()
```



7) correlation between vote average & vote category



Power BI

Untitled - Power BI Desktop

File Home Insert Modeling View Optimize Help Format Data / Drill

Paste Get data Excel Data Server SQL Enter Dataverse Recent sources Transform data Refresh data New visual Text box More Quick measure Sensitivity Publish

Clipboard Sum of popularity by original_language

original_language: English (0.32%), Japanese (0.11%), Spanish (0.1%), Korean (0.07%), French (0.06%), Italian (0.05%), Chinese (0.04%), Russian (0.03%), German (0.02%), Dutch (0.01%), Polish (0.01%)

Sum of vote_average by vote_category

Sum of vote_average, Sum of vote_count, Sum of new_budget and Sum of popularity by vote_category

Sum of vote_average, Sum of vote_count, Sum of new_budget and Sum of popularity by vote_category

Sum of vote_average, Sum of vote_count, Sum of new_budget and Sum of popularity by vote_category

Sum of vote_average by vote_category

Sum of vote_average, Sum of vote_count, Sum of new_budget and Sum of popularity by vote_category

Sum of new_budget, Sum of popularity, Sum of vote_average and Sum of vote_count by title

vote_category: good (237.16K), bad (10.3K)

Y Axis: Sum of vote_count

X Axis: Sum of vote_average

Filters on this visual:

- Sum of new_budget is (All)
- Sum of popularity is (All)
- Sum of vote_average is (All)
- Sum of vote_count is (All)
- vote_category is (All)

Filters on this page:

- Add data fields here

Filters on all pages:

- Add data fields here

Visualizations

Data

Build visual

Filters

Search

Play Axis

Stop sharing Hide

meet.google.com is sharing your screen.

Page 1 of 1

Type here to search

Windows Taskbar: 25°C, 11:19 PM, 5/29/2023

Machine Learning

```
In [1]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder

ds = pd.read_csv("popular_10000_movies_tmdb.csv")

# Step 1: Prepare the Data
# Extract the feature (X) and target variable (y)
X = ds["vote_average"]
y = ds["vote_category"]

# Step 2: Split the Data into Training and Test Sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Step 3: Encode the Target Variable
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)

# Reshape the input data from 1D to 2D
X_train = X_train.values.reshape(-1, 1)
X_test = X_test.values.reshape(-1, 1)

# Step 4: Initialize the Classifier
classifier = LogisticRegression()

# Step 5: Train the Model
classifier.fit(X_train, y_train_encoded)

# Step 6: Predict and Evaluate
y_pred = classifier.predict(X_test)

# Step 7: Calculate Accuracy
accuracy = accuracy_score(y_test_encoded, y_pred)
print("Accuracy:", accuracy)
```

Accuracy: 1.0