Assignment no.04

members:

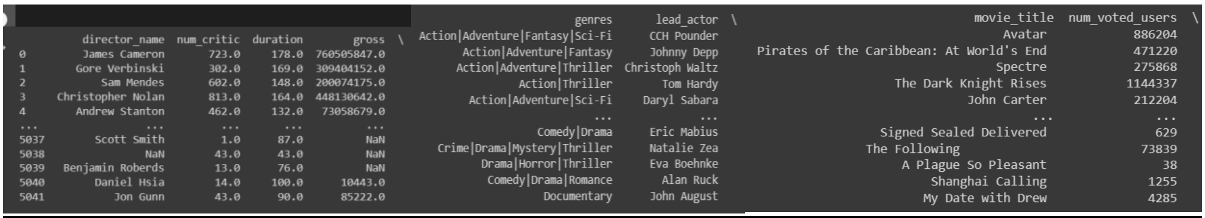
Amrikt Bhadra(203)

Tejashri Darade(213)

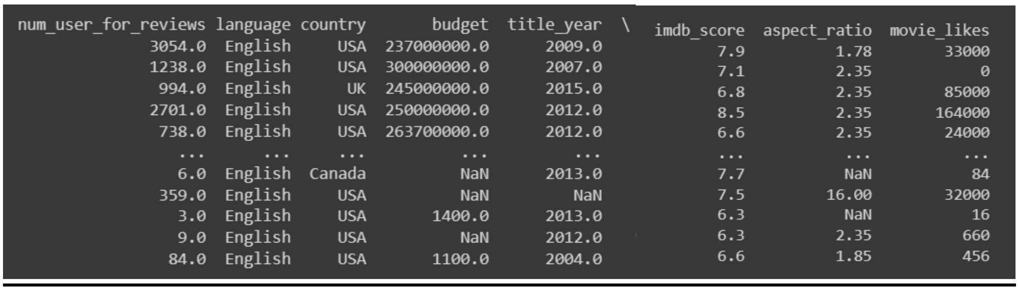
Prajwal Ganar(220)

Code:

```
import pandas as pd

df  = pd.read_csv("/content/movie_data.csv")
#print all records of dataset
print(df)
```

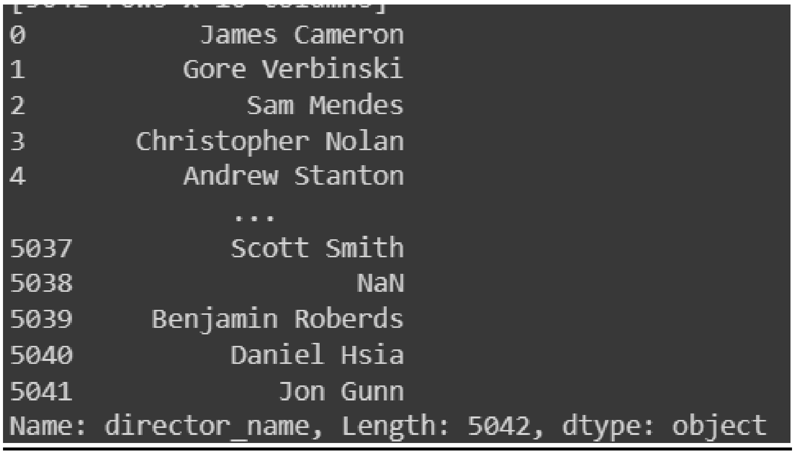


```
         director_name  num_critic  duration        gross  \                       genres      lead_actor  \                               movie_title  num_voted_users  \
0        James Cameron       723.0     178.0  760505847.0      Action|Adventure|Fantasy|Sci-Fi     CCH Pounder                                        Avatar           886204
1       Gore Verbinski       302.0     169.0  309404152.0             Action|Adventure|Fantasy     Johnny Depp    Pirates of the Caribbean: At World's End           471220
2           Sam Mendes       602.0     148.0  200074175.0            Action|Adventure|Thriller Christoph Waltz                                       Spectre           275868
3    Christopher Nolan       813.0     164.0  448130642.0                      Action|Thriller       Tom Hardy                         The Dark Knight Rises          1144337
4       Andrew Stanton       462.0     132.0   73058679.0              Action|Adventure|Sci-Fi    Daryl Sabara                                   John Carter           212204
...                ...         ...       ...          ...                                  ...             ...                                           ...              ...
5037       Scott Smith         1.0      87.0          NaN                         Comedy|Drama     Eric Mabius                      Signed Sealed Delivered              629
5038               NaN        43.0      43.0          NaN          Crime|Drama|Mystery|Thriller     Natalie Zea                    The Following                         73839
5039  Benjamin Roberds        13.0      76.0          NaN               Drama|Horror|Thriller     Eva Boehnke                         A Plague So Pleasant               38
5040       Daniel Hsia        14.0     100.0      10443.0              Comedy|Drama|Romance       Alan Ruck                             Shanghai Calling             1255
5041          Jon Gunn        43.0      90.0      85222.0                          Documentary     John August                             My Date with Drew             4285
```



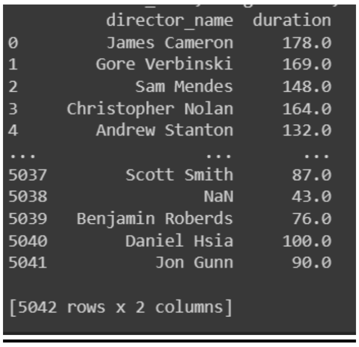
```
num_user_for_reviews language country       budget  title_year  \  imdb_score  aspect_ratio  movie_likes
              3054.0  English     USA  237000000.0      2009.0            7.9          1.78        33000
              1238.0  English     USA  300000000.0      2007.0            7.1          2.35            0
               994.0  English      UK  245000000.0      2015.0            6.8          2.35        85000
              2701.0  English     USA  250000000.0      2012.0            8.5          2.35       164000
               738.0  English     USA  263700000.0      2012.0            6.6          2.35        24000
                 ...      ...     ...          ...         ...            ...           ...          ...
                 6.0  English  Canada          NaN      2013.0            7.7           NaN           84
               359.0  English     USA          NaN         NaN            7.5         16.00        32000
                 3.0  English     USA       1400.0      2013.0            6.3           NaN           16
                 9.0  English     USA          NaN      2012.0            6.3          2.35          660
                84.0  English     USA       1100.0      2004.0            6.6          1.85          456
```

```
#1 print Names of all employees
print(df['director_name'])
```



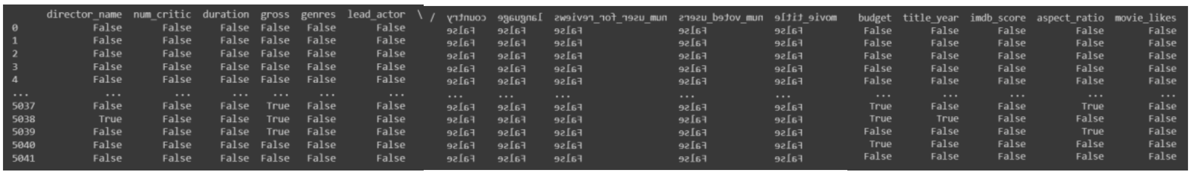
```
[5042 rows x 10 columns]
0           James Cameron
1          Gore Verbinski
2              Sam Mendes
3       Christopher Nolan
4          Andrew Stanton
              ...
5037          Scott Smith
5038                  NaN
5039     Benjamin Roberds
5040          Daniel Hsia
5041             Jon Gunn
Name: director_name, Length: 5042, dtype: object
```

```
#2 print name and duration
print(df[['director_name','duration']])
```



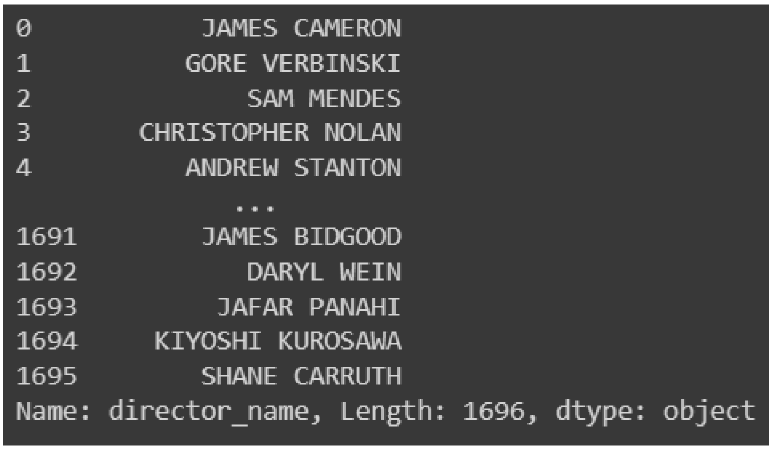
```
         director_name  duration
0        James Cameron     178.0
1       Gore Verbinski     169.0
2           Sam Mendes     148.0
3    Christopher Nolan     164.0
4       Andrew Stanton     132.0
...                ...       ...
5037       Scott Smith      87.0
5038               NaN      43.0
5039  Benjamin Roberds      76.0
5040       Daniel Hsia     100.0
5041          Jon Gunn      90.0

[5042 rows x 2 columns]
```

```
#1 Data cleaning
#check for missing values
print(df.isnull())

# #drop rows with missing values
df.dropna(inplace=True
```

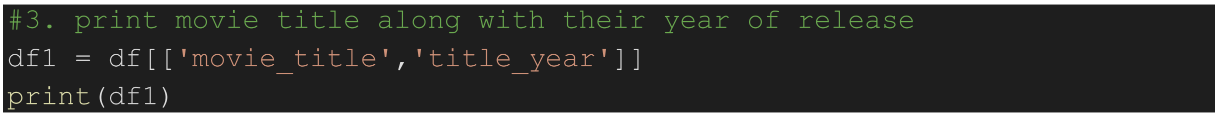


```
      director_name  num_critic  duration  gross  genres  lead_actor  \ /  country  language  num_user_for_reviews  num_voted_users  movie_title   budget  title_year  imdb_score  aspect_ratio  movie_likes
0             False       False     False  False   False       False       False    False     False                 False            False          False       False       False         False        False
1             False       False     False  False   False       False       False    False     False                 False            False          False       False       False         False        False
2             False       False     False  False   False       False       False    False     False                 False            False          False       False       False         False        False
3             False       False     False  False   False       False       False    False     False                 False            False          False       False       False         False        False
4             False       False     False  False   False       False       False    False     False                 False            False          False       False       False         False        False
...             ...         ...       ...    ...     ...         ...       ...      ...       ...                   ...              ...            ...         ...         ...           ...          ...
5037          False       False     False   True   False       False       False    False     False                 False            False           True       False       False          True        False
5038           True       False     False   True   False       False       False    False     False                 False            False           True        True       False         False        False
5039          False       False     False   True   False       False       False    False     False                 False            False          False       False       False          True        False
5040          False       False     False  False   False       False       False    False     False                 False            False           True       False       False         False        False
5041          False       False     False  False   False       False       False    False     False                 False            False          False       False       False         False        False
```

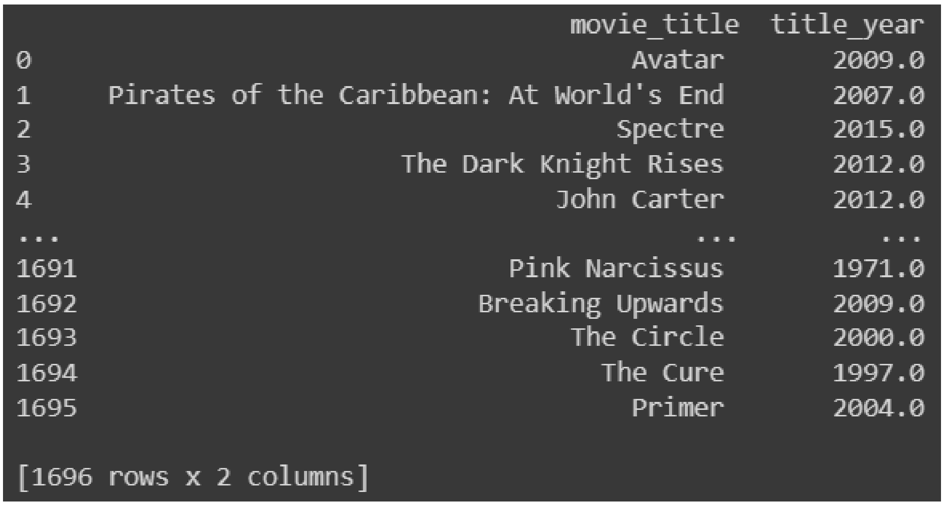
```
#2 convert string to upper case
df['director_name'].str.upper()
```



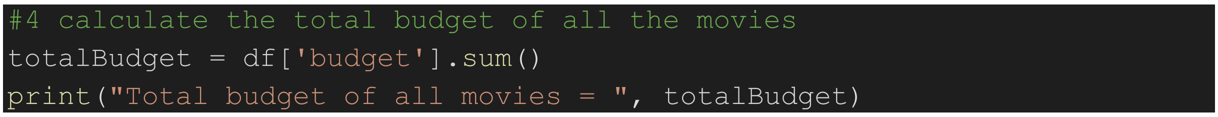
```
0           JAMES CAMERON
1          GORE VERBINSKI
2              SAM MENDES
3       CHRISTOPHER NOLAN
4          ANDREW STANTON
              ...
1691        JAMES BIDGOOD
1692           DARYL WEIN
1693         JAFAR PANAHI
1694     KIYOSHI KUROSAWA
1695        SHANE CARRUTH
Name: director_name, Length: 1696, dtype: object
```



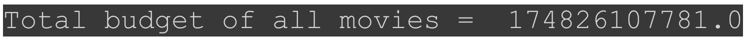
```
#3. print movie title along with their year of release
df1 = df[['movie_title','title_year']]
print(df1)
```



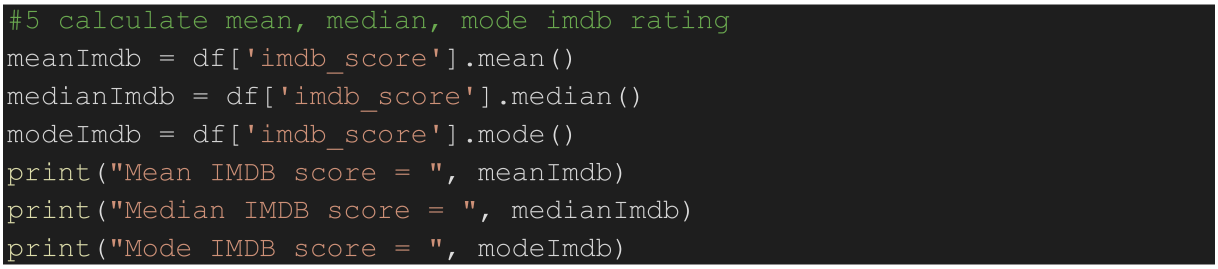
```
                                    movie_title  title_year
0                                        Avatar      2009.0
1      Pirates of the Caribbean: At World's End      2007.0
2                                       Spectre      2015.0
3                         The Dark Knight Rises      2012.0
4                                   John Carter      2012.0
...                                         ...         ...
1691                             Pink Narcissus      1971.0
1692                           Breaking Upwards      2009.0
1693                                 The Circle      2000.0
1694                                   The Cure      1997.0
1695                                     Primer      2004.0

[1696 rows x 2 columns]
```



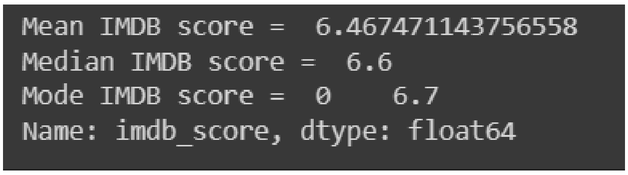
```
#4 calculate the total budget of all the movies
totalBudget = df['budget'].sum()
print("Total budget of all movies = ", totalBudget)
```



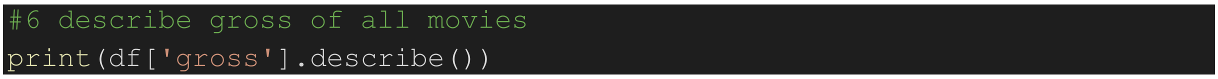
```
Total budget of all movies =  174826107781.0
```



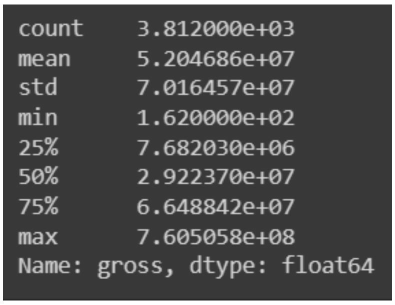
```
#5 calculate mean, median, mode imdb rating
meanImdb = df['imdb_score'].mean()
medianImdb = df['imdb_score'].median()
modeImdb = df['imdb_score'].mode()
print("Mean IMDB score = ", meanImdb)
print("Median IMDB score = ", medianImdb)
print("Mode IMDB score = ", modeImdb)
```



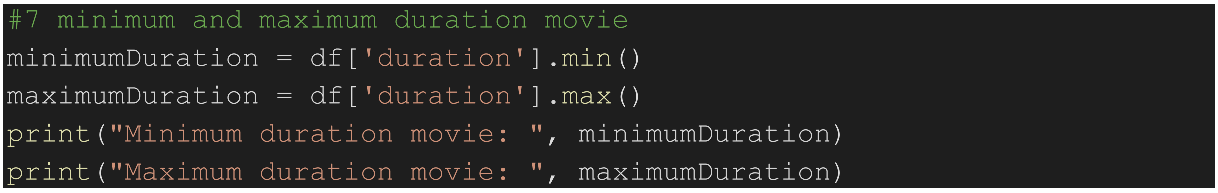
```
Mean IMDB score =  6.467471143756558
Median IMDB score =  6.6
Mode IMDB score =  0    6.7
Name: imdb_score, dtype: float64
```



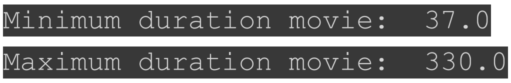
```
#6 describe gross of all movies
print(df['gross'].describe())
```



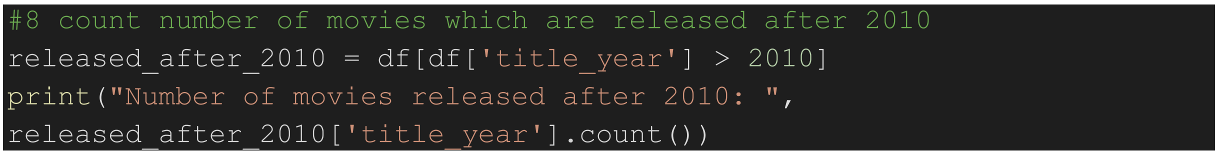
```
count    3.812000e+03
mean     5.204686e+07
std      7.016457e+07
min      1.620000e+02
25%      7.682030e+06
50%      2.922370e+07
75%      6.648842e+07
max      7.605058e+08
Name: gross, dtype: float64
```



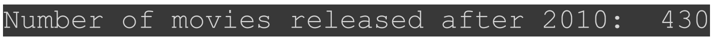
```
#7 minimum and maximum duration movie
minimumDuration = df['duration'].min()
maximumDuration = df['duration'].max()
print("Minimum duration movie: ", minimumDuration)
print("Maximum duration movie: ", maximumDuration)
```



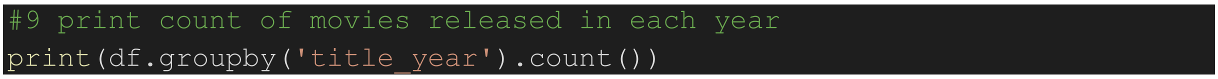
```
Minimum duration movie:  37.0
Maximum duration movie:  330.0
```



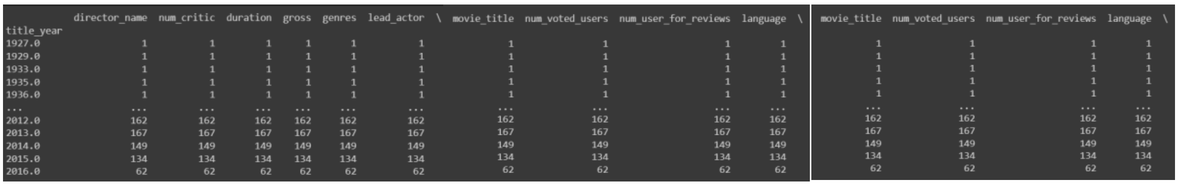
```
#8 count number of movies which are released after 2010
released_after_2010 = df[df['title_year'] > 2010]
print("Number of movies released after 2010: ",
released_after_2010['title_year'].count())
```



```
Number of movies released after 2010:  430
```

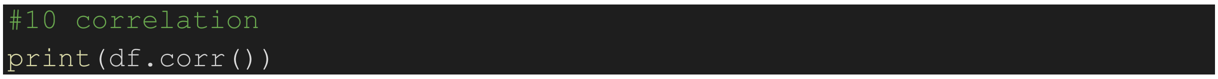


```
#9 print count of movies released in each year
print(df.groupby('title_year').count())
```

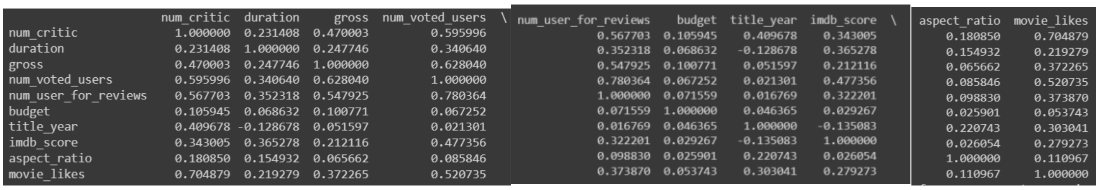


```
            director_name  num_critic  duration  gross  genres  lead_actor  \  movie_title  num_voted_users  num_user_for_reviews  language  \
title_year
1927.0                  1           1         1      1       1           1               1                1                     1         1
1929.0                  1           1         1      1       1           1               1                1                     1         1
1933.0                  1           1         1      1       1           1               1                1                     1         1
1935.0                  1           1         1      1       1           1               1                1                     1         1
1936.0                  1           1         1      1       1           1               1                1                     1         1
...                   ...         ...       ...    ...     ...         ...             ...              ...                   ...       ...
2012.0                162         162       162    162     162         162             162              162                   162       162
2013.0                167         167       167    167     167         167             167              167                   167       167
2014.0                149         149       149    149     149         149             149              149                   149       149
2015.0                134         134       134    134     134         134             134              134                   134       134
2016.0                 62          62        62     62      62          62              62               62                    62        62
```

```
movie_title  num_voted_users  num_user_for_reviews  language  \

          1                1                     1         1
          1                1                     1         1
          1                1                     1         1
          1                1                     1         1
          1                1                     1         1
        ...              ...                   ...       ...
        162              162                   162       162
        167              167                   167       167
        149              149                   149       149
        134              134                   134       134
         62               62                    62        62
```



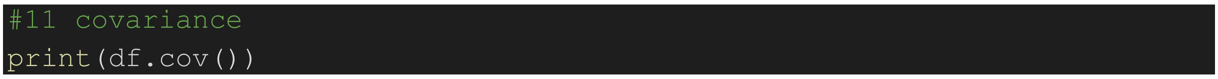
```
#10 correlation
print(df.corr())
```



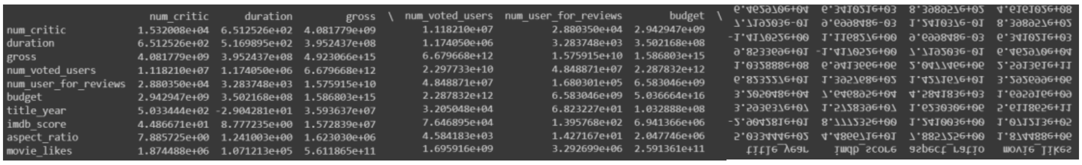
```
                      num_critic  duration     gross  num_voted_users  \
num_critic              1.000000  0.231408  0.470003         0.595996
duration                0.231408  1.000000  0.247746         0.340640
gross                   0.470003  0.247746  1.000000         0.628040
num_voted_users         0.595996  0.340640  0.628040         1.000000
num_user_for_reviews    0.567703  0.352318  0.547925         0.780364
budget                  0.105945  0.068632  0.100771         0.067252
title_year              0.409678 -0.128678  0.051597         0.021301
imdb_score              0.343005  0.365278  0.212116         0.477356
aspect_ratio            0.180850  0.154932  0.065662         0.085846
movie_likes             0.704879  0.219279  0.372265         0.520735
```

```
num_user_for_reviews    budget  title_year  imdb_score  \
            0.567703  0.105945    0.409678    0.343005
            0.352318  0.068632   -0.128678    0.365278
            0.547925  0.100771    0.051597    0.212116
            0.780364  0.067252    0.021301    0.477356
            1.000000  0.071559    0.016769    0.322201
            0.071559  1.000000    0.046365    0.029267
            0.016769  0.046365    1.000000   -0.135083
            0.322201  0.029267   -0.135083    1.000000
            0.098830  0.025901    0.220743    0.026054
            0.373870  0.053743    0.303041    0.279273
```

```
aspect_ratio  movie_likes
    0.180850     0.704879
    0.154932     0.219279
    0.065662     0.372265
    0.085846     0.520735
    0.098830     0.373870
    0.025901     0.053743
    0.220743     0.303041
    0.026054     0.279273
    1.000000     0.110967
    0.110967     1.000000
```

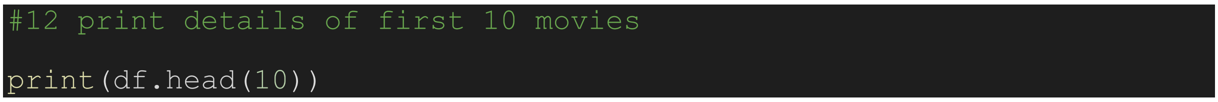


```
#11 covariance
print(df.cov())
```

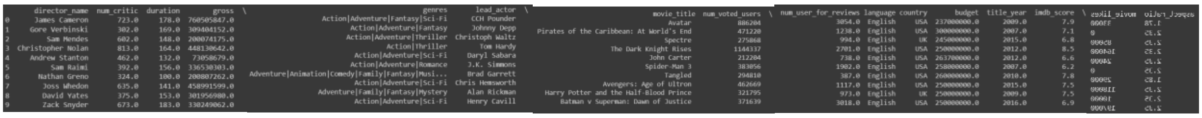


```
                        num_critic      duration         gross  \  num_voted_users  num_user_for_reviews        budget  \
num_critic            1.532008e+04  6.512526e+02  4.081779e+09        1.118210e+07          2.880350e+04  2.942947e+09
duration              6.512526e+02  5.169895e+02  3.952437e+08        1.174050e+06          3.283748e+03  3.502168e+08
gross                 4.081779e+09  3.952437e+08  4.923066e+15        6.679668e+12          1.575915e+10  1.586803e+15
num_voted_users       1.118210e+07  1.174050e+06  6.679668e+12        2.297733e+10          4.848871e+07  2.287832e+12
num_user_for_reviews  2.880350e+04  3.283748e+03  1.575915e+10        4.848871e+07          1.680301e+05  6.583046e+09
budget                2.942947e+09  3.502168e+08  1.586803e+15        2.287832e+12          6.583046e+09  5.036664e+16
title_year            5.033444e+02 -2.904281e+01  3.593637e+07        3.205048e+04          6.823227e+01  1.032888e+08
imdb_score            4.486671e+01  8.777235e+00  1.572839e+07        7.646895e+04          1.395768e+02  6.941366e+06
aspect_ratio          7.885725e+00  1.241003e+00  1.623030e+06        4.584183e+03          1.427167e+01  2.047746e+06
movie_likes           1.874488e+06  1.071213e+05  5.611865e+11        1.695916e+09          3.292699e+06  2.591361e+11
```

```
  title_year    imdb_score  aspect_ratio   movie_likes
5.033444e+02  4.486671e+01  7.885725e+00  1.874488e+06
-2.904281e+01 8.777235e+00  1.241003e+00  1.071213e+05
3.593637e+07  1.572839e+07  1.623030e+06  5.611865e+11
3.205048e+04  7.646895e+04  4.584183e+03  1.695916e+09
6.823227e+01  1.395768e+02  1.427167e+01  3.292699e+06
1.032888e+08  6.941366e+06  2.047746e+06  2.591361e+11
9.853396e+01 -1.417052e+00  7.719203e-01  6.462970e+04
-1.417052e+00 1.116827e+00  9.699848e-03  6.341021e+03
7.719203e-01  9.699848e-03  1.241037e-01  8.398927e+02
6.462970e+04  6.341021e+03  8.398927e+02  4.616102e+08
```



```
#12 print details of first 10 movies

print(df.head(10))
```



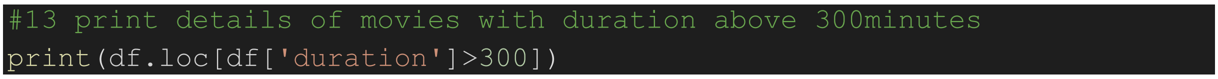
```
      director_name  num_critic  duration        gross  \
0     James Cameron       723.0     178.0  760505847.0
1    Gore Verbinski       302.0     169.0  309404152.0
2        Sam Mendes       602.0     148.0  200074175.0
3 Christopher Nolan       813.0     164.0  448130642.0
4    Andrew Stanton       462.0     132.0   73058679.0
5         Sam Raimi       392.0     156.0  336530303.0
6      Nathan Greno       324.0     100.0  200807262.0
7       Joss Whedon       635.0     141.0  458991599.0
8       David Yates       375.0     153.0  301956980.0
9       Zack Snyder       673.0     183.0  330249062.0
```

```
                                                 genres          lead_actor  \
                        Action|Adventure|Fantasy|Sci-Fi         CCH Pounder
                               Action|Adventure|Fantasy         Johnny Depp
                              Action|Adventure|Thriller     Christoph Waltz
                                        Action|Thriller           Tom Hardy
                                Action|Adventure|Sci-Fi        Daryl Sabara
                               Action|Adventure|Romance        J.K. Simmons
Adventure|Animation|Comedy|Family|Fantasy|Musi...       Brad Garrett
                                Action|Adventure|Sci-Fi   Chris Hemsworth
                       Adventure|Family|Fantasy|Mystery         Alan Rickman
                                Action|Adventure|Sci-Fi         Henry Cavill
```

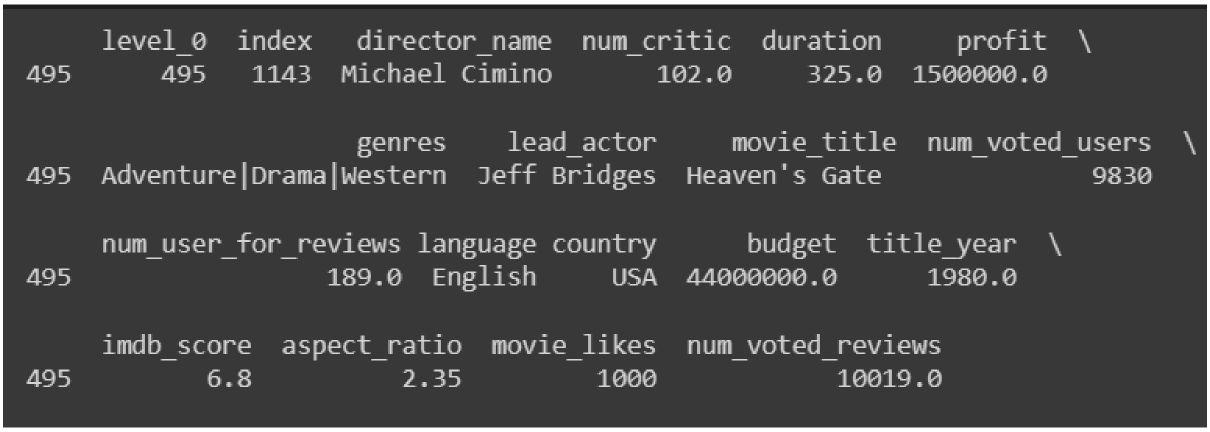
```
                                movie_title  num_voted_users  \
                                     Avatar           886204
   Pirates of the Caribbean: At World's End           471220
                                    Spectre           275868
                      The Dark Knight Rises          1144337
                                John Carter           212204
                               Spider-Man 3           383056
                                    Tangled           294810
                    Avengers: Age of Ultron           462669
     Harry Potter and the Half-Blood Prince           321795
         Batman v Superman: Dawn of Justice           371639
```

```
num_user_for_reviews language country       budget  title_year  imdb_score  \
              3054.0  English     USA  237000000.0      2009.0         7.9
              1238.0  English     USA  300000000.0      2007.0         7.1
               994.0  English      UK  245000000.0      2015.0         6.8
              2701.0  English     USA  250000000.0      2012.0         8.5
               738.0  English     USA  263700000.0      2012.0         6.6
              1902.0  English     USA  258000000.0      2007.0         6.2
               387.0  English     USA  260000000.0      2010.0         7.8
              1117.0  English     USA  250000000.0      2015.0         7.5
               973.0  English      UK  250000000.0      2009.0         7.5
              3018.0  English     USA  250000000.0      2016.0         6.9
```

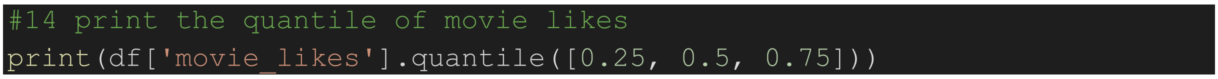
```
aspect_ratio  movie_likes
        1.78        33000
        2.35            0
        2.35        85000
        2.35       164000
        2.35        24000
        2.35            0
        1.85        29000
        2.35       118000
        2.35        10000
        2.35       197000
```



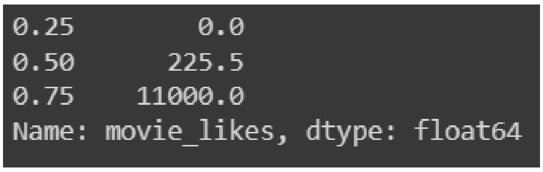
```
#13 print details of movies with duration above 300minutes
print(df.loc[df['duration']>300])
```



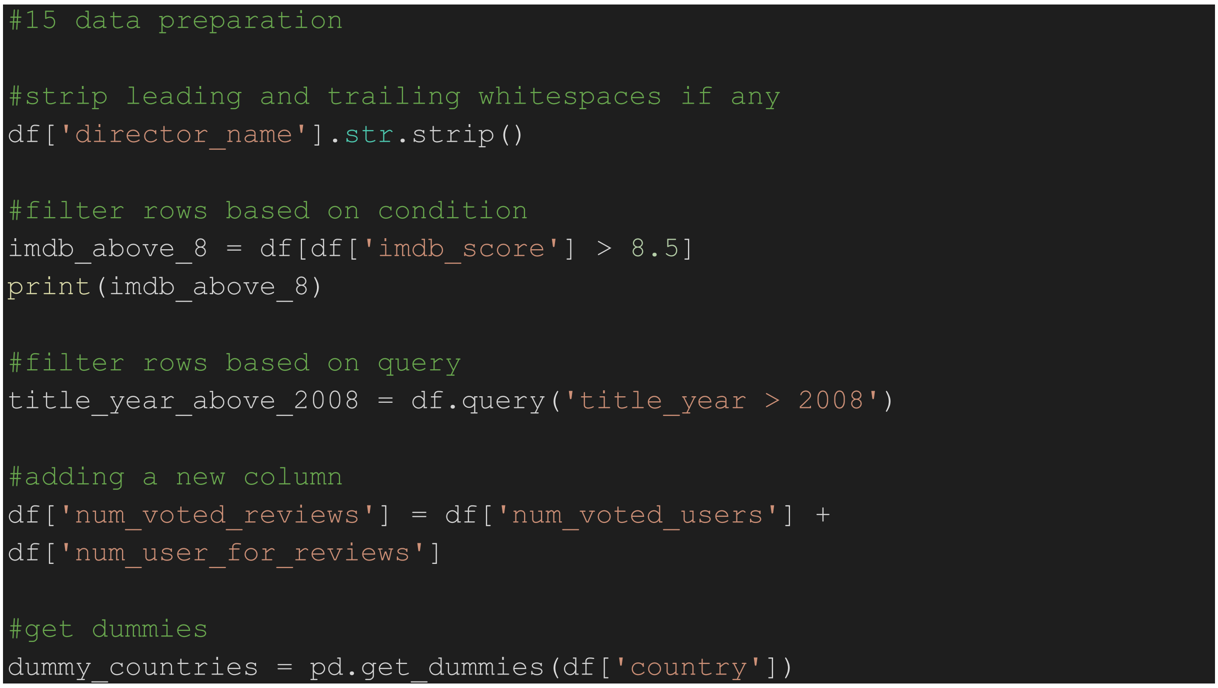
```
     level_0  index    director_name  num_critic  duration     profit  \
495      495   1143  Michael Cimino       102.0     325.0  1500000.0

                      genres     lead_actor     movie_title  num_voted_users  \
495  Adventure|Drama|Western  Jeff Bridges  Heaven's Gate              9830

     num_user_for_reviews language country      budget  title_year  \
495                 189.0  English     USA  44000000.0      1980.0

     imdb_score  aspect_ratio  movie_likes  num_voted_reviews
495         6.8          2.35         1000            10019.0
```



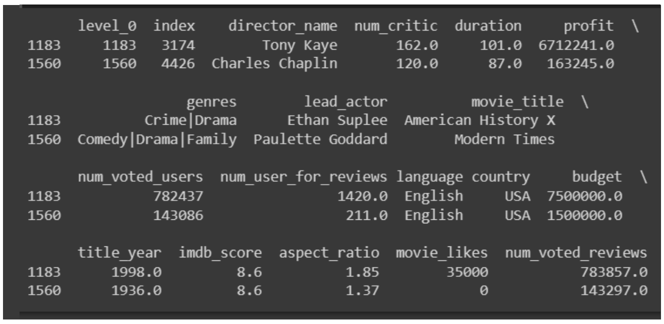
```
#14 print the quantile of movie likes
print(df['movie_likes'].quantile([0.25, 0.5, 0.75]))
```



```
0.25        0.0
0.50      225.5
0.75    11000.0
Name: movie_likes, dtype: float64
```

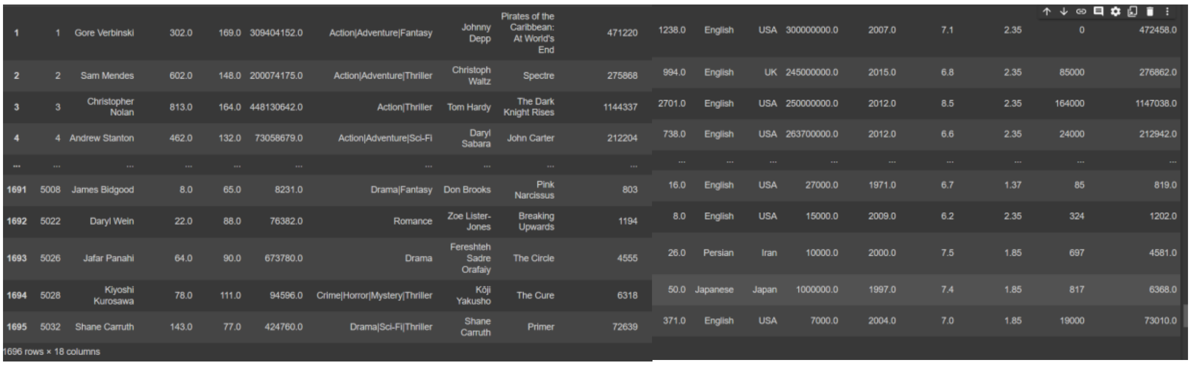


```
#15 data preparation

#strip leading and trailing whitespaces if any
df['director_name'].str.strip()

#filter rows based on condition
imdb_above_8 = df[df['imdb_score'] > 8.5]
print(imdb_above_8)

#filter rows based on query
title_year_above_2008 = df.query('title_year > 2008')

#adding a new column
df['num_voted_reviews'] = df['num_voted_users'] +
df['num_user_for_reviews']

#get dummies
dummy_countries = pd.get_dummies(df['country'])
```



```
      level_0  index    director_name  num_critic  duration     profit  \
1183     1183   3174        Tony Kaye       162.0     101.0  6712241.0
1560     1560   4426  Charles Chaplin       120.0      87.0   163245.0

                   genres        lead_actor         movie_title  \
1183          Crime|Drama      Ethan Suplee  American History X
1560  Comedy|Drama|Family  Paulette Goddard        Modern Times

      num_voted_users  num_user_for_reviews language country     budget  \
1183           782437                1420.0  English     USA  7500000.0
1560           143086                 211.0  English     USA  1500000.0

      title_year  imdb_score  aspect_ratio  movie_likes  num_voted_reviews
1183      1998.0         8.6          1.85        35000           783857.0
1560      1936.0         8.6          1.37            0           143297.0
```

```
#16  data aggregation
#renaming our gross column as profit
df.rename(columns={'gross':'profit'},inplace=True)
df
```



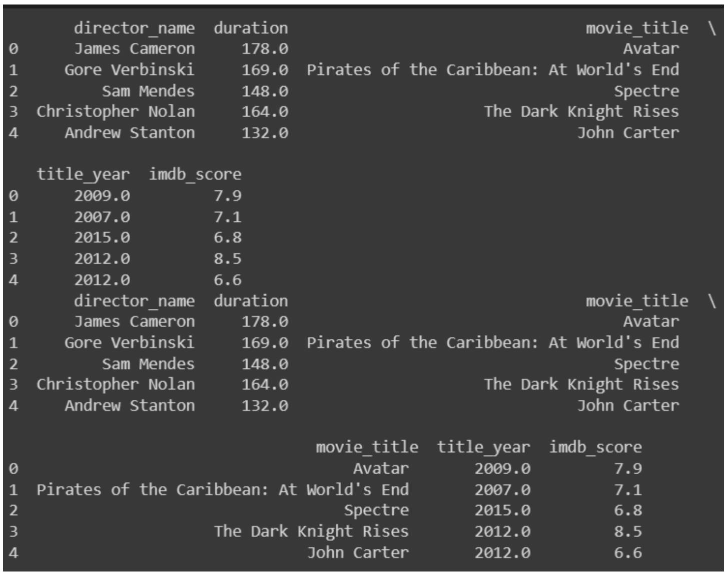
1	1	Gore Verbinski	302.0	169.0	309404152.0	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	471220	1238.0	English	USA	300000000.0	2007.0	7.1	2.35	0	472458.0
2	2	Sam Mendes	602.0	148.0	200074175.0	Action|Adventure|Thriller	Christoph Waltz	Spectre	275868	994.0	English	UK	245000000.0	2015.0	6.8	2.35	85000	276862.0
3	3	Christopher Nolan	813.0	164.0	448130642.0	Action|Thriller	Tom Hardy	The Dark Knight Rises	1144337	2701.0	English	USA	250000000.0	2012.0	8.5	2.35	164000	1147038.0
4	4	Andrew Stanton	462.0	132.0	73058679.0	Action|Adventure|Sci-Fi	Daryl Sabara	John Carter	212204	738.0	English	USA	263700000.0	2012.0	6.6	2.35	24000	212942.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1691	5008	James Bidgood	8.0	65.0	8231.0	Drama|Fantasy	Don Brooks	Pink Narcissus	803	16.0	English	USA	27000.0	1971.0	6.7	1.37	85	819.0
1692	5022	Daryl Wein	22.0	88.0	76382.0	Romance	Zoe Lister-Jones	Breaking Upwards	1194	8.0	English	USA	15000.0	2009.0	6.2	2.35	324	1202.0
1693	5026	Jafar Panahi	64.0	90.0	673780.0	Drama	Fereshteh Sadre Orafaiy	The Circle	4555	26.0	Persian	Iran	10000.0	2000.0	7.5	1.85	697	4581.0
1694	5028	Kiyoshi Kurosawa	78.0	111.0	94596.0	Crime|Horror|Mystery|Thriller	Kōji Yakusho	The Cure	6318	50.0	Japanese	Japan	1000000.0	1997.0	7.4	1.85	817	6368.0
1695	5032	Shane Carruth	143.0	77.0	424760.0	Drama|Sci-Fi|Thriller	Shane Carruth	Primer	72639	371.0	English	USA	7000.0	2004.0	7.0	1.85	19000	73010.0

1696 rows × 18 columns

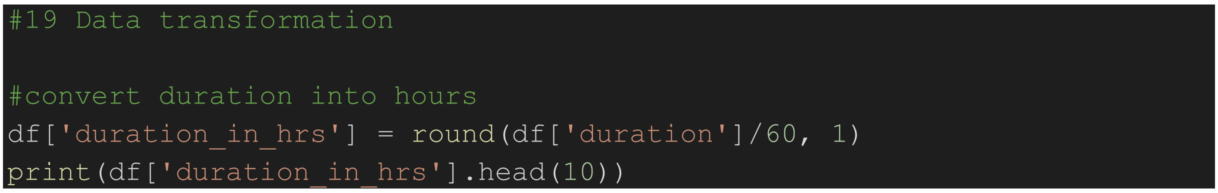
```
#17 Datatype conversion
df['duration'] = df['duration'].astype('float')
print(type(df['duration'][0]))
```

```
<class 'numpy.float64'>
```

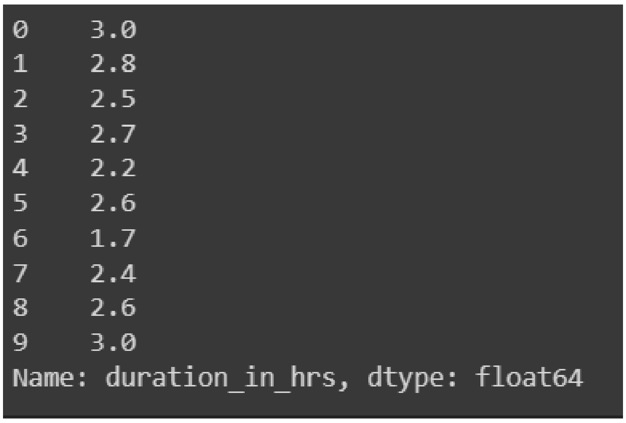
```
#18 data wrangling

newdf1 = pd.DataFrame(df[['director_name', 'duration', 'movie_title']])
newdf2 = pd.DataFrame(df[['movie_title', 'title_year', 'imdb_score']])

# merge dataframes
merged_df = pd.merge(newdf1, newdf2)
print(merged_df.head())

#concat dataframes
concatenated_df = pd.concat([newdf1, newdf2], axis=1)
print(concatenated_df.head())
```



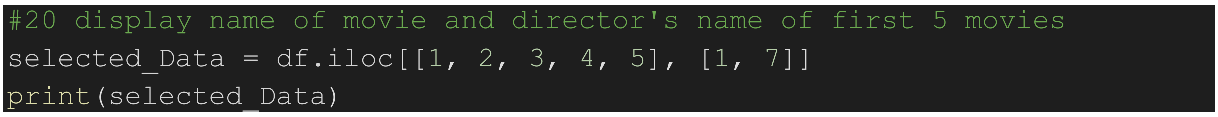
```
      director_name  duration                               movie_title  \
0     James Cameron     178.0                                   Avatar
1    Gore Verbinski     169.0  Pirates of the Caribbean: At World's End
2        Sam Mendes     148.0                                   Spectre
3 Christopher Nolan     164.0                     The Dark Knight Rises
4    Andrew Stanton     132.0                               John Carter

  title_year  imdb_score
0     2009.0         7.9
1     2007.0         7.1
2     2015.0         6.8
3     2012.0         8.5
4     2012.0         6.6
      director_name  duration                               movie_title  \
0     James Cameron     178.0                                   Avatar
1    Gore Verbinski     169.0  Pirates of the Caribbean: At World's End
2        Sam Mendes     148.0                                   Spectre
3 Christopher Nolan     164.0                     The Dark Knight Rises
4    Andrew Stanton     132.0                               John Carter

                               movie_title  title_year  imdb_score
0                                   Avatar      2009.0         7.9
1 Pirates of the Caribbean: At World's End      2007.0         7.1
2                                  Spectre      2015.0         6.8
3                    The Dark Knight Rises      2012.0         8.5
4                              John Carter      2012.0         6.6
```



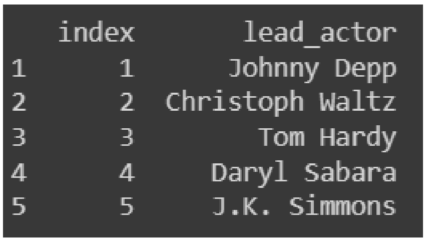
```
#19 Data transformation

#convert duration into hours
df['duration_in_hrs'] = round(df['duration']/60, 1)
print(df['duration_in_hrs'].head(10))
```



```
0    3.0
1    2.8
2    2.5
3    2.7
4    2.2
5    2.6
6    1.7
7    2.4
8    2.6
9    3.0
Name: duration_in_hrs, dtype: float64
```



```
#20 display name of movie and director's name of first 5 movies
selected_Data = df.iloc[[1, 2, 3, 4, 5], [1, 7]]
print(selected_Data)
```



```
   index       lead_actor
1      1      Johnny Depp
2      2  Christoph Waltz
3      3        Tom Hardy
4      4     Daryl Sabara
5      5     J.K. Simmons
```