Investigating a Dataset (TMDb_Movies Dataset)

UDACITY - DATA ANALYST NANODEGREE

Mohamed Balam |Project-2|Benghazi, Libya |23rd-Feb-2020

Overview:

At this project I'll test my analysis skills which I gained from this course, to analyze and investigate as well as write observations about the "TMDb Movies Dataset" which is supported from Udacity at this link below:

https://docs.google.com/document/d/e/2PACX-1vTlVmknRRnfy_4eTrjw5hYGaiQim5ctr9naaRd4V9du2B5bxpd8FEH3KtDgp8qVekw7Cj1GLk1IXdZi/pub?embedded=True

This dataset had a lot of information, it's nearly to 11 thousand movies collected from The Movie Database (TMDb), it includes a lot of important data to analyze like 'Revenue, Budget, and user rating, etc. So I'll try to ask questions and answer it after analysis process.

Goals:

1. Take a look at the given dataset.
2. Write questions that I'll answer it after analyzing the given dataset.
3. Cleaning dataset by deleting non-needed columns as well as nan-values.
4. Write functions that will help my analysis process.
5. Make analysis to answer my question one after a one.
6. Write observations as well as conclusion about analysis process.

Tools Used:

1. Python: To write the code that allows me to read data, analyze it, and draw charts using libraries like pandas, numpy, and matplotlib.
2. ANACONDA and Jupyter Lab: To install packages, write and modify the Python code and print the results.
3. Microsoft excel: To open and make a first look at the dataset.

My questions:

1. Which movies had the highest and lowest popularity?
2. What is the best number of voters that effects on movies profit?
3. Which movies had the highest and lowest earned profit?
4. Which genre had the greatest number of movies?
5. Who is the top 5 directors that directed the greatest number of movies?

Data wrangling:

Lesson after a lesson my experience in increase, so I just followed the analysis process steps that I learned from your course, and filled the knowledge gap by searching about auxiliary functions which helped me to complete this project, so I'll mention every helping recourse at references section.

Step 1- Importing needed packages as well as the dataset

Here I imports the packages I used in this project, and also, I import the dataset which I'll analyze in next steps:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


df = pd.read_csv( " tmdb-movies.csv" )
```

Step 2- Cleaning dataset

Here I'll clean the dataset by using some functions to show and delete duplicated data, show and drop zero values as well as fill some of it with Nan, and drop unneeded columns:

```
#show sum of missy data for each column
[37]:    df.isna().sum()

[37]:
        id                      0
        imdb_id                10
        popularity              0
        budget                  0
        revenue                 0
        original_title          0
        cast                   76
        homepage             7930
        director               44
        tagline              2824
        keywords             1493
        overview                4
        runtime                 0
        genres                 23
```

```
production_companies    1030
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

[38]:

```
#show sum of duplicated values
sum(df.duplicated())
```

[38]:

```
1
```

[39]:

```
#change "release_data" type to datetime
df['release_date'] = pd.to_datetime(df['release_date'])
df['release_date'].head()
```

[39]:

```
0   2015-06-09
1   2015-05-13
2   2015-03-18
3   2015-12-15
4   2015-04-01
Name: release_date, dtype: datetime64[ns]
```

Now let’s drop duplicated and null values:

[40]:

```
#Drop dublicated values
df.drop_duplicates(inplace=True)
df.shape()
```

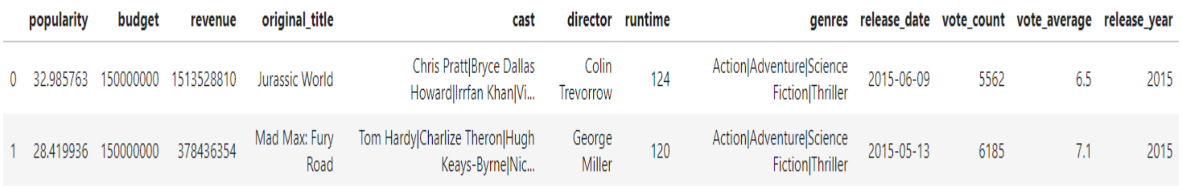
[40]:

```
(10865, 21)
```

[42]:

```
# Dropping all unneeded columns.
df.drop(['id', 'imdb_id', 'budget_adj', 'revenue_adj', 'homepage',  'tagline', 'keywords',
'overview','production_companies'], axis=1, inplace=True)
df.head(2)
```

[42]:

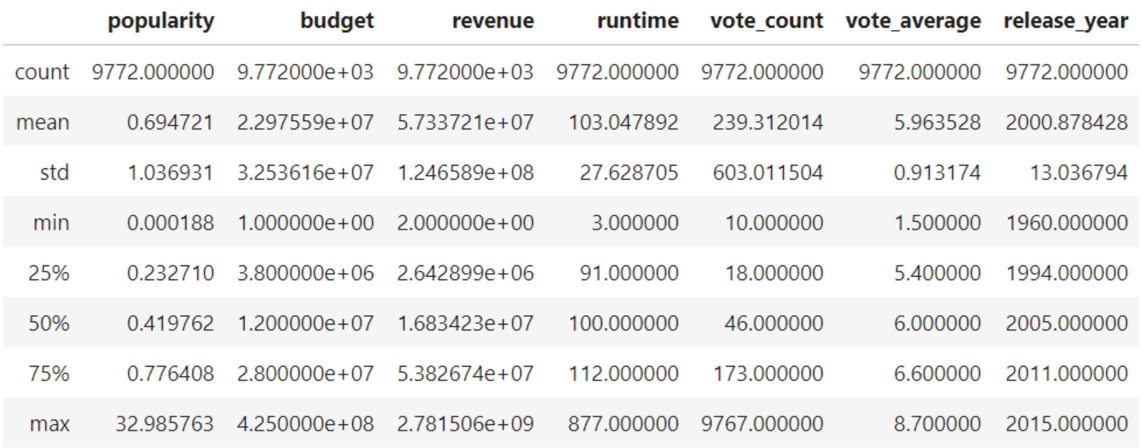


	popularity	budget	revenue	original_title	cast	director	runtime	genres	release_date	vote_count	vote_average	release_year
0	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124	Action|Adventure|Science Fiction|Thriller	2015-06-09	5562	6.5	2015
1	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	120	Action|Adventure|Science Fiction|Thriller	2015-05-13	6185	7.1	2015

[43]:

```
 #Drop missy data
df = df.replace(0,None)
df.dropna(how='any',axis=0,inplace=True)
df.describe()
```

[43]:

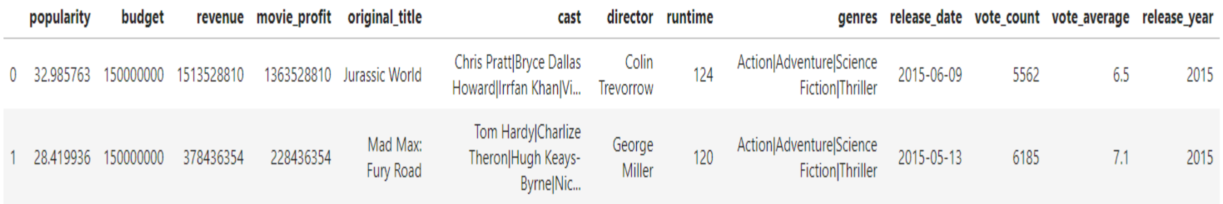


	popularity	budget	revenue	runtime	vote_count	vote_average	release_year
count	9772.000000	9.772000e+03	9.772000e+03	9772.000000	9772.000000	9772.000000	9772.000000
mean	0.694721	2.297559e+07	5.733721e+07	103.047892	239.312014	5.963528	2000.878428
std	1.036931	3.253616e+07	1.246589e+08	27.628705	603.011504	0.913174	13.036794
min	0.000188	1.000000e+00	2.000000e+00	3.000000	10.000000	1.500000	1960.000000
25%	0.232710	3.800000e+06	2.642899e+06	91.000000	18.000000	5.400000	1994.000000
50%	0.419762	1.200000e+07	1.683423e+07	100.000000	46.000000	6.000000	2005.000000
75%	0.776408	2.800000e+07	5.382674e+07	112.000000	173.000000	6.600000	2011.000000
max	32.985763	4.250000e+08	2.781506e+09	877.000000	9767.000000	8.700000	2015.000000

[44]:

```
# Creating 'movie_profit' column by subtracting values in 'budget' column
from those in 'revenue' column.

df.insert(3, 'movie_profit', df['revenue'] - df['budget'])
df.head(2)
```

[44]:



	popularity	budget	revenue	movie_profit	original_title	cast	director	runtime	genres	release_date	vote_count	vote_average	release_year
0	32.985763	150000000	1513528810	1363528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	124	Action|Adventure|Science Fiction|Thriller	2015-06-09	5562	6.5	2015
1	28.419936	150000000	378436354	228436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	120	Action|Adventure|Science Fiction|Thriller	2015-05-13	6185	7.1	2015

All is done now, so let’s begin to analyze, write functions, and answer all questions.

Step 3- Analyze dataset and answering questions

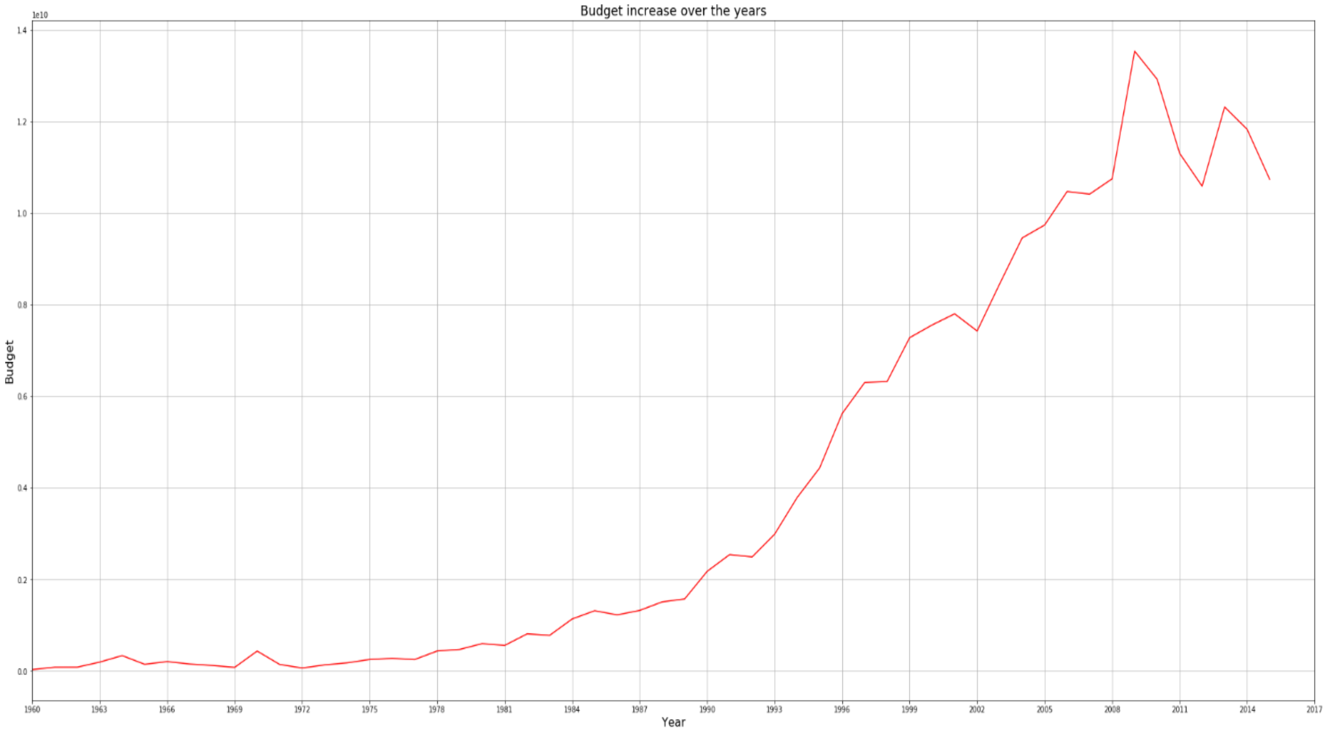
Now after cleaning, and trimming, the dataset is ready now to compute statistics, create visualizations to answer all questions.

Before answering any questions, I just did some statistic about “budget increase over the years” so I write this code below that shows me the chart after compiling.

[45]:

```
# Creating a Budget vs Years plot.

df.groupby('release_year')['budget'].sum().plot(figsize = (35,
16),xticks=np.arange(1960,2018,3), color='r')

plt.grid(True)
plt.title('Budget increase over the years', fontsize = 18)
plt.xlabel('Year', fontsize = 16)
plt.ylabel('Budget', fontsize = 16);
```

[45]:



From this simple plot you can found that year after a year, movies budget is increasing, o guess that is because the tools to make a good movie with high quality and nearly to real life is expensive.

Now, let’s answer some questions.

Q1: Which movies had the highest and lowest popularity?

To answer this kind of questions you need to write a function that helps you to make it very smooth, so after searching about some auxiliary functions I found some Ideas helped me to write this function:

[46]:

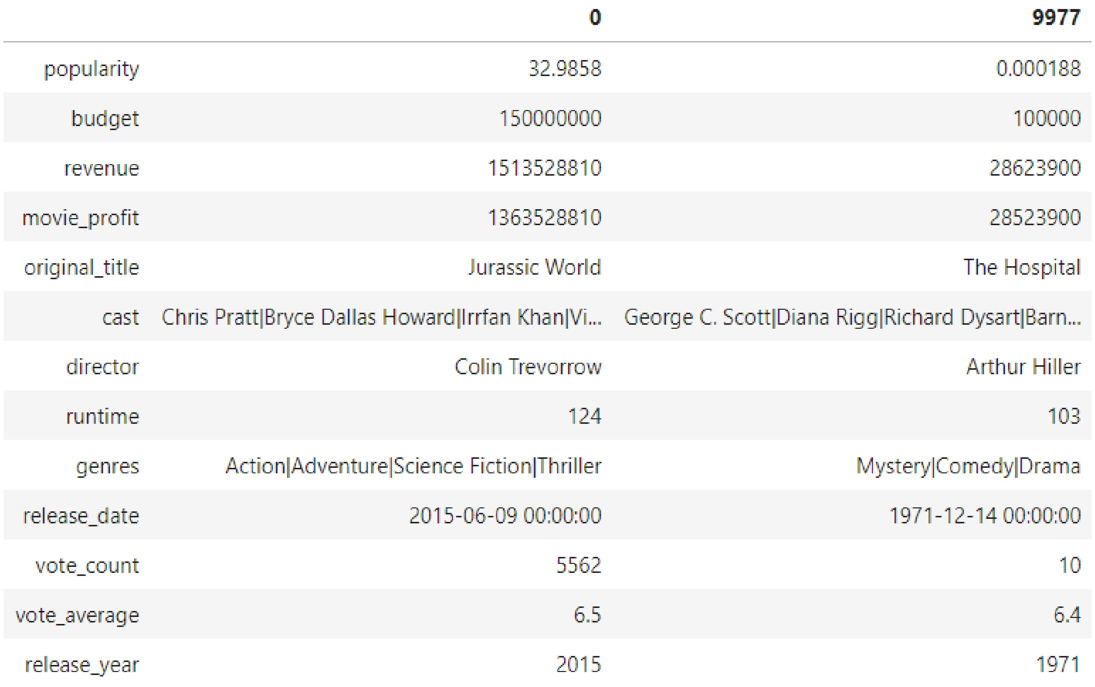
```
# Define high_vs_low function to calculate and return the highest and lowest values from dataset

def high_vs_low(column_name):

    low_value = df[column_name].idxmin()

    high_value = df[column_name].idxmax()

    high = pd.DataFrame(df.loc[high_value,:])

    low = pd.DataFrame(df.loc[low_value,:])

    # Print ing solutions

    print(df['original_title'][high_value],"is highest in "+ column_name )

    print(df['original_title'][low_value],"is the lowest "+ column_name )

    return pd.concat([high,low],axis = 1)
```

[47]:

```
high_vs_low('popularity')
```

```
Jurassic World is highest in popularity
The Hospital is the lowest popularity
```

[47]:



	0	9977
popularity	32.9858	0.000188
budget	150000000	100000
revenue	1513528810	28623900
movie_profit	1363528810	28523900
original_title	Jurassic World	The Hospital
cast	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	George C. Scott|Diana Rigg|Richard Dysart|Barn...
director	Colin Trevorrow	Arthur Hiller
runtime	124	103
genres	Action|Adventure|Science Fiction|Thriller	Mystery|Comedy|Drama
release_date	2015-06-09 00:00:00	1971-12-14 00:00:00
vote_count	5562	10
vote_average	6.5	6.4
release_year	2015	1971

After calling this function, we found that the most popular movie is "**Jurassic World**", and the most unpopular movie is "**The Hospital**".

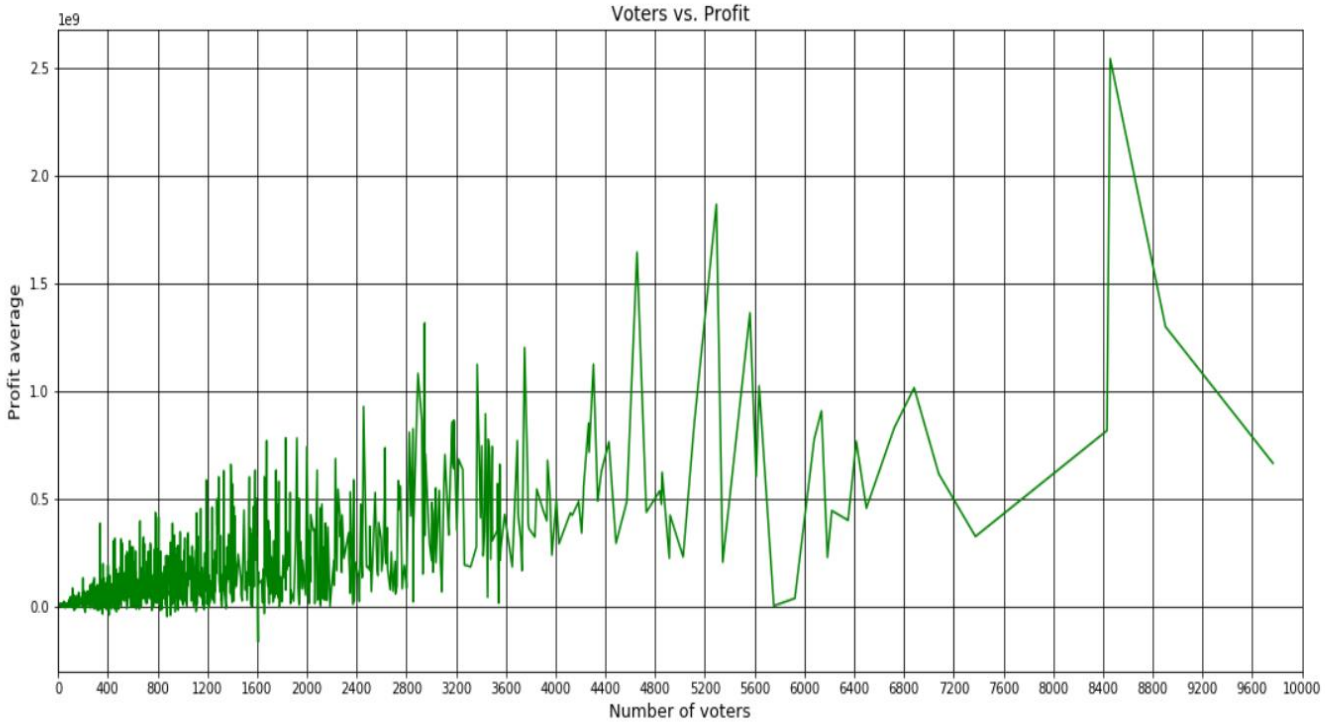
Q2: What is the best number of voters that effects on movies profit?

Here we need to use groupby function to calculate the solution, and after that making a plot depending on groupby solution, and now let's show the answer:

[48]:

```
# Use the group by vote_count and find the mean of movie_profit and make
plot for this calculating.
df.groupby('vote_count')['movie_profit'].mean().plot(figsize =
(18,8),xticks=np.arange(0,10300,400),color='g')
plt.grid(True,color='k')
plt.title("Voters vs. Profit",fontsize = 14)
plt.xlabel('Number of voters',fontsize = 13)
plt.ylabel('Profit average',fontsize = 13)
```

[48]:



From this plot you can find that there is a relation between voting and movie profits, to prove that look at the plot again you will show the profit was increased by increasing of number of voters, specially when voters number equal to 8400 voters.

Q3: Which movies had the highest and lowest earned profit?

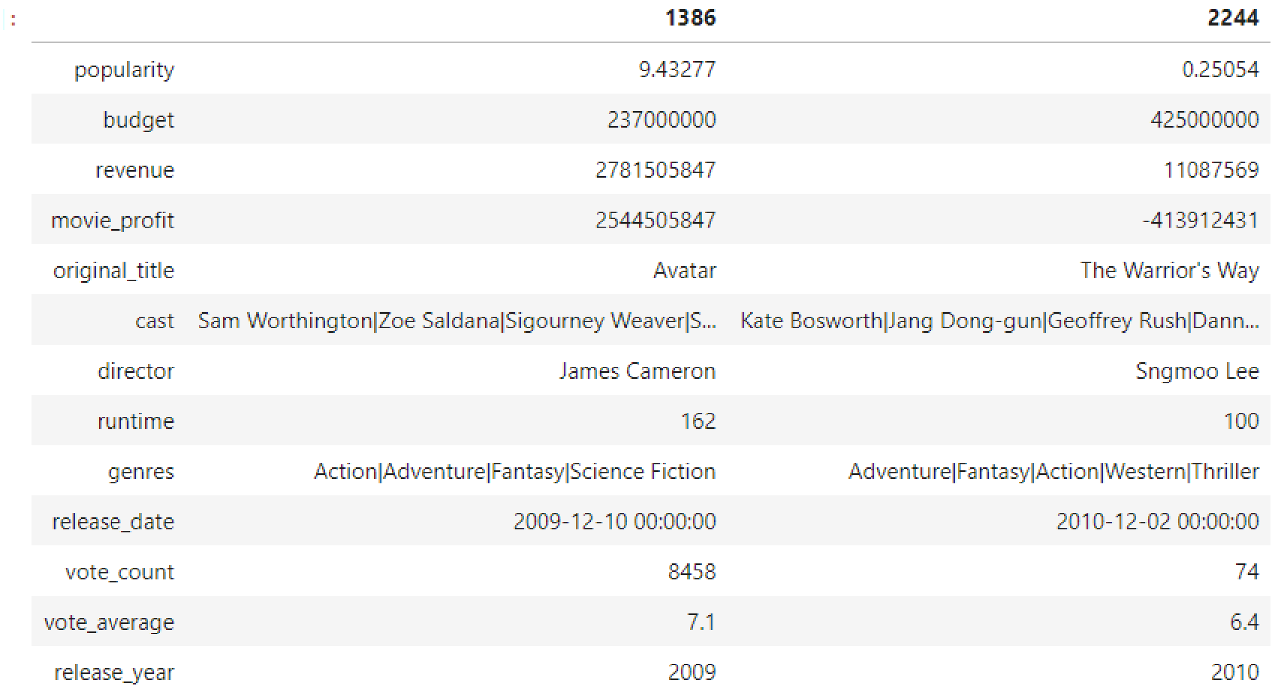
To answer this question, you need to use the high_vs_low function again, which is defined at Q1.

[49]:

```
high_vs_low('movie_profit')
```

```
Avatar is highest in movie_profit
The Warrior's Way is the lowest in movie_profit
```

[49]:



:

	1386	2244
popularity	9.43277	0.25054
budget	237000000	425000000
revenue	2781505847	11087569
movie_profit	2544505847	-413912431
original_title	Avatar	The Warrior's Way
cast	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...	Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann...
director	James Cameron	Sngmoo Lee
runtime	162	100
genres	Action|Adventure|Fantasy|Science Fiction	Adventure|Fantasy|Action|Western|Thriller
release_date	2009-12-10 00:00:00	2010-12-02 00:00:00
vote_count	8458	74
vote_average	7.1	6.4
release_year	2009	2010

After calling this function, we found that the highest movie in profit is “**Avatar**”, and the lowest movie in profit is “**The Warrior's Way**”.

Q4: Which genre had the greatest number of movies?

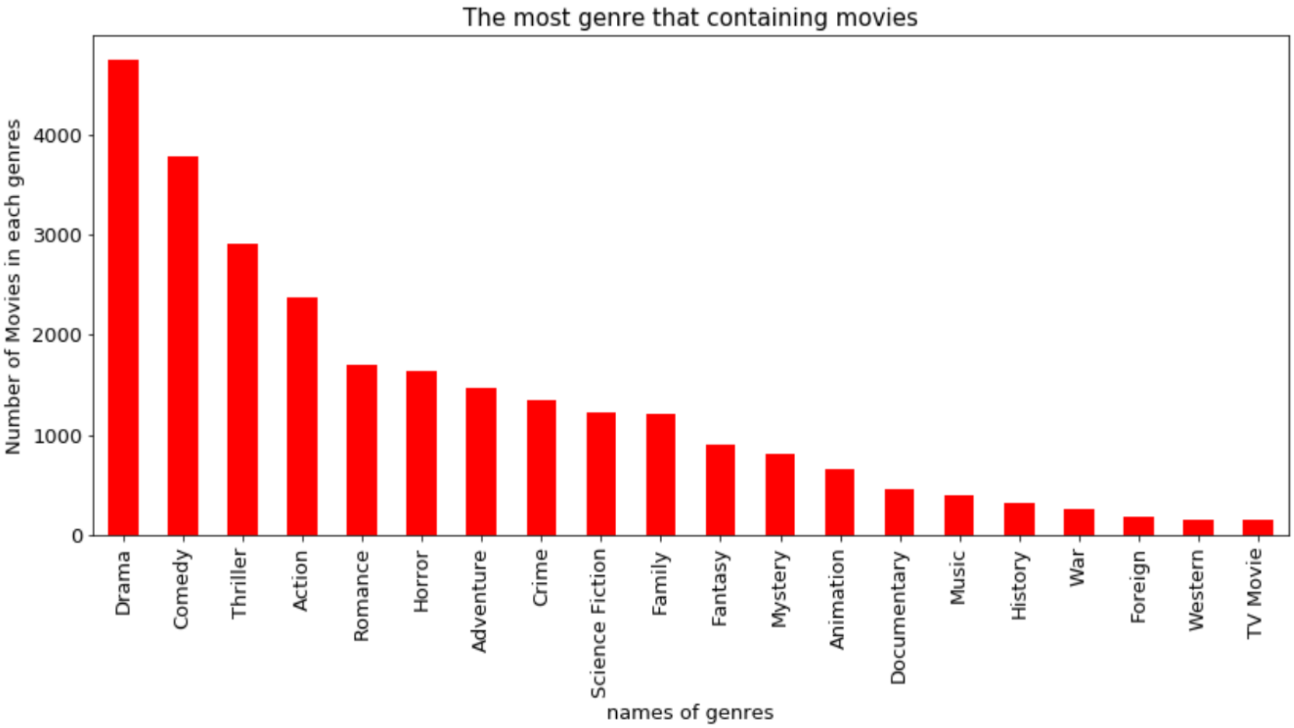
Here I answered this question after searching about auxiliary functions and ideas, so I found some ideas and similar functions which is helped me to write this:

[50]:

```
 #Fuction that calculates the number of movies in each genre

def movies_count_per_genre(column_name):

    splited_plot = df[column_name].str.cat(sep = '|')

    data= pd.Series(splited_plot.split('|'))

    sol = data.value_counts(ascending=False)

    return sol


#Calling the function.

per_genre_count = movies_count_per_genre('genres')
```

```
#draw a chart to show the solution using plot

per_genre_count.plot(kind= 'bar',figsize = (14,6),fontsize=13,color='r')


#setup the title and the labels of the plot.

plt.title("The most genre that containing movies",fontsize=15)

plt.xlabel('names of genres',fontsize=13)

plt.ylabel("Number of Movies in each genres",fontsize= 13)
```

[50]:



From this chart you can find that the genre that had the greatest number of movies is “**Drama**”, and the genre that had the lowest number of movies is “**TV Movie**”

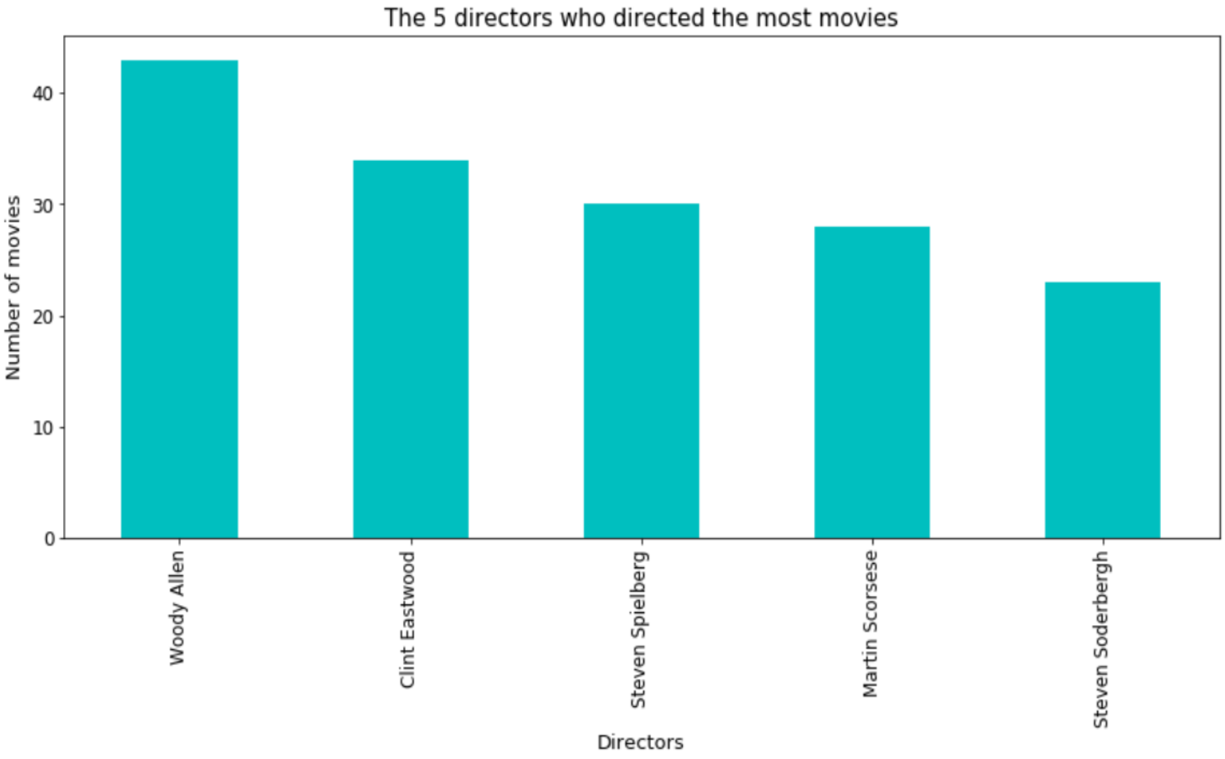
Q5: Who is the top 5 directors that directed the greatest number of movies?

To answer this question, you need to call the previous function to calculate the number of movies for each director and then make a plot to show the top 5 directors who were had the greatest number of directed movies.

[51]:

```
#calling movies_count_per_genre function to calculate and answer this question
top_directors = movies_count_per_genre ('director')
top_directors.iloc[:5].plot.bar(figsize=(14,6),fontsize=12,color = 'c')

plt.title("The 5 directors who directed the most movies",fontsize=15)
plt.xlabel('Directors',fontsize=13)
plt.ylabel("Number of movies",fontsize= 13)
```

[51]:



From this chart you can find the top 5 directors, and the director who had the greatest number of movies is “**Woody Allen**”, and the director who had the fifth greatest number of movies is “**Steven Soderbergh**”.

Conclusion:

After completing this project, I gained new knowledge about movies, like, budget is increasing year after a year, number of voters is one of important things that effect on movies profit, and the genre that had the greatest number of movies is “Drama”.

Limitations: I used TMDb dataset and I found some limitations like, there is a lot of zero values, the data isn’t updated because we are in 2020 and last statistics on this dataset is 2018, to prove this I made an analysis for the most movie had a profit and I saw “Avatar” is the answer, and we all know that is old statistic because “Avengers: Endgame” get the most profit in 2019.

References:

1. Auxiliary functions to calculate some statistics like high_vs_low: https://www.geeksforgeeks.org/
2. Debugging and find some solutions for pandas: https://stackoverflow.com/questions/tagged/pandas
3. Some helped tutorials for plots from: https://matplotlib.org/tutorials/index.html