Project: Movie statistics

Table of Contents

- Introduction
- Data Wrangling
- Data cleaning
- Exploratory Data Analysis
- Conclusions

Introduction

The analysis of the data set "TMDb movies dataset" will be shown below. Some statistical data, min, max and average values, and the dependence of some indicators on others in the form of tables and graphs are shown.

The data set contained a set of movie data in the form of 10866 lines in 21 columns. During the analysis, uninformative columns were deleted, filled and sorted empty cells, some data types were converted.

Research Questions:

1. Maximum, minimum and average films budget
2. Maximum, minimum and average films revenue
3. Maximum, minimum and average films profit
4. Most popular genres
5. Most popular actors
6. Most popular directors
7. TOP 10 Films rated
8. Most frequent runtime (hist)
9. Most frequent dates of release (graph)
10. Ehe dependence of the film's vote on the budget

In [1]:

```
#importing necessary files and packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
% matplotlib inline
```

In [2]:

```
#output in bold
from IPython.display import Markdown, display
def printmd(string):
    display(Markdown(string))
```

Data Wrangling

Load and preparing of data for cleaning and analysis:

In [3]:

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

Out[3]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan	http://www.jurassicworld.com/	C

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	http://www.madmaxmovie.com/
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.thedivergentseries.movie/#insurgent
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	http://www.starwars.com/films/star-wars-episod...
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	http://www.furious7.com/

5 rows × 21 columns

In [4]:

```
#dataset info
df.shape
```

Out[4]:

(10866, 21)

In [5]:

```
df.describe()
```

Out[5]:

	id	popularity	budget	revenue	runtime	vote_count	vote_average	release_year	budge
count	10866.000000	10866.000000	1.086600e+04	1.086600e+04	10866.000000	10866.000000	10866.000000	10866.000000	1.08660
mean	66064.177434	0.646441	1.462570e+07	3.982332e+07	102.070863	217.389748	5.974922	2001.322658	1.75510
std	92130.136561	1.000185	3.091321e+07	1.170035e+08	31.381405	575.619058	0.935142	12.812941	3.43061
min	5.000000	0.000065	0.000000e+00	0.000000e+00	0.000000	10.000000	1.500000	1960.000000	0.00000
25%	10596.250000	0.207583	0.000000e+00	0.000000e+00	90.000000	17.000000	5.400000	1995.000000	0.00000
50%	20669.000000	0.383856	0.000000e+00	0.000000e+00	99.000000	38.000000	6.000000	2006.000000	0.00000
75%	75610.000000	0.713817	1.500000e+07	2.400000e+07	111.000000	145.750000	6.600000	2011.000000	2.08532
max	417859.000000	32.985763	4.250000e+08	2.781506e+09	900.000000	9767.000000	9.200000	2015.000000	4.25000

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
id                    10866 non-null int64
imdb_id               10856 non-null object
popularity            10866 non-null float64
```

```
budget                10866 non-null int64
revenue               10866 non-null int64
original_title        10866 non-null object
cast                  10790 non-null object
homepage               2936 non-null object
director              10822 non-null object
tagline               8042 non-null object
keywords              9373 non-null object
overview              10862 non-null object
runtime               10866 non-null int64
genres                10843 non-null object
production_companies    9836 non-null object
release_date          10866 non-null object
vote_count            10866 non-null int64
vote_average           10866 non-null float64
release_year          10866 non-null int64
budget_adj            10866 non-null float64
revenue_adj           10866 non-null float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

Data Cleaning

1. Removing unused and uninformative columns
2. Adding new columns needed to answer research questions
3. Filling or deleting empty cells and values with "NAN"
4. Converting data to the required formats

In [7]:

```
#Deleting unusing columns
df.drop(['id', 'imdb_id', 'popularity', 'homepage', 'keywords', 'overview', 'vote_count', 'budget_adj', 'revenue_adj'], axis=1, inplace=True)
```

In [8]:

```
#Create a new column
df['profit'] = df['revenue'] - df['budget']
```

In [9]:

```
#cleaning and transformation of zero values and "NAN"
df['budget'] = df['budget'].replace(0, np.nan)
```

In [10]:

```
df['revenue'] = df['revenue'].replace(0, np.nan)
```

In [11]:

```
df['runtime'] = df['runtime'].replace(0, np.nan)
```

In [12]:

```
df.dropna(subset = ['budget', 'revenue', 'runtime'], inplace = True)
```

In [13]:

```
#deleting unreliable information
df.drop(df[df.revenue < 1000].index, inplace=True)
```

In [14]:

```
df.drop(df[df.budget < 1000].index, inplace=True)
```

In [15]:

```
rows, col = df.shape
```

In [16]:

```
#cnanging of data format
df.release_date = pd.to_datetime(df['release_date'])
```

In [17]:

```
df['budget'] = df['budget'].apply(np.int64)
```

In [18]:

```
df['revenue'] = df['revenue'].apply(np.int64)
```

In [19]:

```
df.dtypes
```

Out[19]:

```
budget                        int64
revenue                       int64
original_title               object
cast                         object
director                     object
tagline                      object
runtime                     float64
genres                       object
production_companies         object
release_date         datetime64[ns]
vote_average                float64
release_year                  int64
profit                        int64
dtype: object
```

Exploratory Data Analysis

Research Question 1: Maximum, Minimum and Average films budget

In [20]:

```
printmd("**Movie with a Maximum budget:**")
```

Movie with a Maximum budget:

In [21]:

```
print("\n", df.loc[df['budget'].idxmax()])
```

```
 budget                                                425000000
revenue                                                11087569
original_title                          The Warrior's Way
cast                  Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann...
director                                          Sngmoo Lee
tagline                                 Assassin. Hero. Legend.
runtime                                                   100
genres                Adventure|Fantasy|Action|Western|Thriller
production_companies                      Boram Entertainment Inc.
release_date                           2010-12-02 00:00:00
vote_average                                              6.4
release_year                                             2010
profit                                            -413912431
Name: 2244, dtype: object
```

In [22]:

```
printmd("**Movie with a Minimum budget:**")
```

Movie with a Minimum budget:

In [23]:

```
print("\n", df.loc[df['budget'].idxmin()])
```

```
budget                                  6000
revenue                                 6000
original_title          Loose Change: Final Cut
cast                                     NaN
director                          Dylan Avery
tagline                                  NaN
runtime                                  129
genres                           Documentary
production_companies          Louder Than Words
release_date                2007-11-11 00:00:00
vote_average                            5.1
release_year                           2007
profit                                    0
Name: 7813, dtype: object
```

In [24]:

```
printmd("**Average budget of movies:**")
```

Average budget of movies:

In [25]:

```
print(int(df['budget'].mean()))
```

```
37460064
```

Research Question 2: Maximum, Minimum and Average films revenue

In [26]:

```
printmd("**Movie with a Maximum revenue:**")
```

Movie with a Maximum revenue:

In [27]:

```
print("\n", df.loc[df['revenue'].idxmax()])
```

```
budget                                     237000000
revenue                                   2781505847
original_title                                Avatar
cast                  Sam Worthington|Zoe Saldana|Sigourney Weaver|S...
director                                 James Cameron
tagline                       Enter the World of Pandora.
runtime                                          162
genres                Action|Adventure|Fantasy|Science Fiction
production_companies    Ingenious Film Partners|Twentieth Century Fox ...
release_date                        2009-12-10 00:00:00
vote_average                                    7.1
release_year                                   2009
profit                                    2544505847
Name: 1386, dtype: object
```

In [28]:

```
printmd("**Movie with a Minimum revenue:**")
```

Movie with a Minimum revenue:

In [29]:

```
print(df.loc[df['revenue'].idxmin()])
```

```
budget                                     1500000
revenue                                       1938
original_title                       Best Man Down
cast                  Justin Long|Jess Weixler|Tyler Labine|Addison ...
director                                 Ted Koland
```

```
tagline                                NaN
runtime                                 90
genres                         Comedy|Drama
production_companies          KODA Entertainment
release_date           2012-10-20 00:00:00
vote_average                           5.9
release_year                          2012
profit                            -1498062
Name: 4668, dtype: object
```

In [30]:

```
printmd("**Average revenue of movies:**")
```

Average revenue of movies:

In [31]:

```
print(int(df['revenue'].mean()))
```

```
108612795
```

Research Question 3: Maximum, Minimum and Average films profit

In [32]:

```
printmd("**Movie with a Maximum profit:**")
```

Movie with a Maximum profit:

In [33]:

```
print("\n", df.loc[df['profit'].idxmax()])
```

```
 budget                                237000000
revenue                              2781505847
original_title                           Avatar
cast                Sam Worthington|Zoe Saldana|Sigourney Weaver|S...
director                           James Cameron
tagline                  Enter the World of Pandora.
runtime                                    162
genres                 Action|Adventure|Fantasy|Science Fiction
production_companies    Ingenious Film Partners|Twentieth Century Fox ...
release_date                2009-12-10 00:00:00
vote_average                                7.1
release_year                               2009
profit                               2544505847
Name: 1386, dtype: object
```

In [34]:

```
printmd("**Movie with a Minimum profit:**")
```

Movie with a Minimum profit:

In [35]:

```
print("\n", df.loc[df['profit'].idxmin()])
```

```
 budget                                425000000
revenue                                11087569
original_title                The Warrior's Way
cast                Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann...
director                             Sngmoo Lee
tagline                    Assassin. Hero. Legend.
runtime                                    100
genres                 Adventure|Fantasy|Action|Western|Thriller
production_companies                Boram Entertainment Inc.
release_date                2010-12-02 00:00:00
vote_average                                6.4
release_year                               2010
```

```
profit                  -413912431
Name: 2244, dtype: object
```

In [36]:

```
printmd("**Average profit of movies:**")
```

Average profit of movies:

In [37]:

```
print(int(df['profit'].mean()))
```

```
71152730
```

Research Question 4: Most popular genres

In [38]:

```
#Create a function to determine the most frequent value in the input column
def df_date(column):
    column_data = df[column].str.cat(sep = '|')
    column_data = pd.Series(column_data.split('|'))
    count = column_data.value_counts(ascending = False)
    return count
```

In [39]:

```
popular_genres = df_date('genres')
popular_genres.head()
```

Out[39]:

```
Drama        1745
Comedy       1340
Thriller     1195
Action       1077
Adventure     743
dtype: int64
```

Research Question 5: Most popular directors

In [40]:

```
popular_director = df_date('director')
popular_director.head()
```

Out[40]:

```
Steven Spielberg    28
Clint Eastwood      24
Ridley Scott        21
Woody Allen         18
Tim Burton          17
dtype: int64
```

Research Question 6: Most popular actors

In [41]:

```
popular_cast = df_date('cast')
popular_cast.head()
```

Out[41]:

```
Robert De Niro      52
Bruce Willis        46
Nicolas Cage        43
Samuel L. Jackson   43
Matt Damon          36
dtype: int64
```

Research Question 7: TOP 10 films rated

In [42]:

```
#grading the vote from higher to least. First 10
top_10 = df.sort_values(['vote_average'], ascending = False)
top_10.head(10)
```

Out[42]:

	budget	revenue	original_title	cast	director	tagline	runtime	genres	production_
7948	1200000	4978922	Stop Making Sense	David Byrne|Tina Weymouth|Chris Frantz|Jerry H...	Jonathan Demme	Why stop making sense? Why a movie? Why a big ...	88.0	Documentary|Music	Talking Head Stiefel Compa
4178	25000000	28341469	The Shawshank Redemption	Tim Robbins|Morgan Freeman|Bob Gunton|William ...	Frank Darabont	Fear can hold you prisoner. Hope can set you f...	142.0	Drama|Crime	Castle Rock Entertainmen
7269	6000000	245066411	The Godfather	Marlon Brando|Al Pacino|James Caan|Richard S. ...	Francis Ford Coppola	An offer you can't refuse.	175.0	Drama|Crime	Paramount Pictures|Alfra Productions
650	3300000	13993093	Whiplash	Miles Teller|J.K. Simmons|Melissa Benoist|Aust...	Damien Chazelle	The road to greatness can take you to the edge.	105.0	Drama|Music	Bold Films|Bl Productions|F Way ...
4177	8000000	213928762	Pulp Fiction	John Travolta|Samuel L. Jackson|Uma Thurman|Br...	Quentin Tarantino	Just because you are a character doesn't mean ...	154.0	Thriller|Crime	Miramax Film Apart|Jersey
2409	63000000	100853753	Fight Club	Edward Norton|Brad Pitt|Meat Loaf|Jared Leto|H...	David Fincher	How much can you know about yourself if you've...	139.0	Drama	Regency Enterprises|F Pictures|Taur
4179	55000000	677945399	Forrest Gump	Tom Hanks|Robin Wright|Gary Sinise|Mykelti Wil...	Robert Zemeckis	The world will never be the same, once you've ...	142.0	Comedy|Drama|Romance	Paramount P
10222	22000000	321265768	Schindler's List	Liam Neeson|Ben Kingsley|Ralph Fiennes|Carolin...	Steven Spielberg	Whoever saves one life, saves the world entire.	195.0	Drama|History|War	Universal Pictures|Amb Entertainmen
2875	185000000	1001921825	The Dark Knight	Christian Bale|Michael Caine|Heath Ledger|Aaro...	Christopher Nolan	Why So Serious?	152.0	Drama|Action|Crime|Thriller	DC Comics|L Pictures|Warr Bros.|Syncop
9758	13000000	47542841	The Godfather:	Al Pacino|Robert Duvall|Diane	Francis Ford	I don't feel I have to wipe	200.0	Drama|Crime	Paramount P

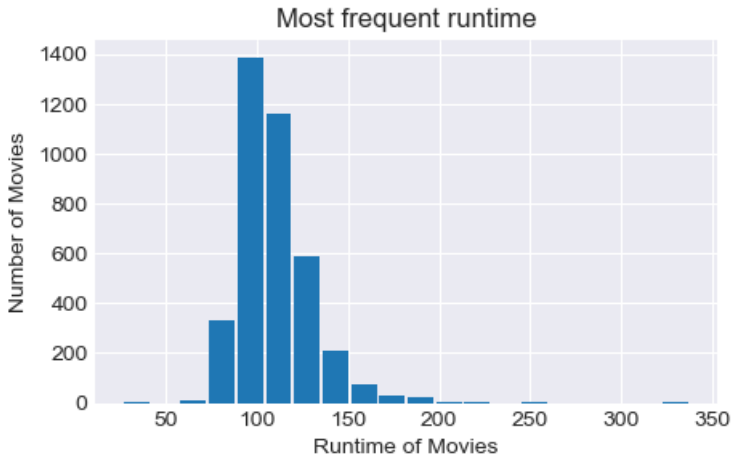
	budget	revenue	original_title	cast	director	tagline	runtime	genres	production_
			Part II	Keaton|Robert De...	Coppola	everybody out, Tom...			Coppola Com

Research Question 8: Most frequent runtime (hist)

In [43]:

```
#chaging the label size and chart visualization
plt.rc('xtick', labelsize = 10)
plt.rc('ytick', labelsize = 10)
sns.set_style('darkgrid')

#changing the figure size
plt.figure(figsize=(5,3), dpi = 100)
#giving a histogram plot
plt.hist(df['runtime'], rwidth = 0.9, bins =20)
#displays the plot
plt.ylabel('runtime')
#title
plt.title('Most frequent runtime')
#Y axis name
plt.ylabel('Number of Movies')
#X axis name
plt.xlabel('Runtime of Movies')
plt.show()
```



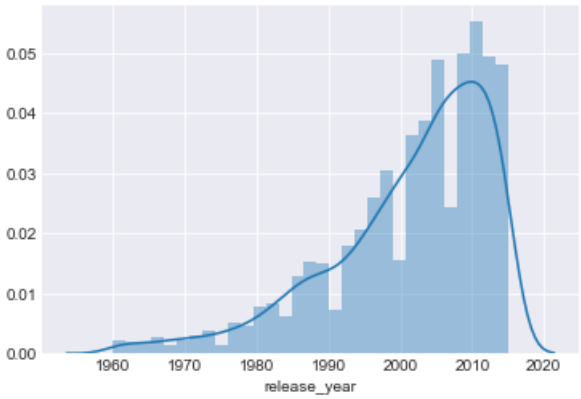
Research Question 9: Most frequent date of release

In [44]:

```
sns.distplot(df.release_year)
```

Out[44]:

<matplotlib.axes._subplots.AxesSubplot at 0x1a13f3d9e8>



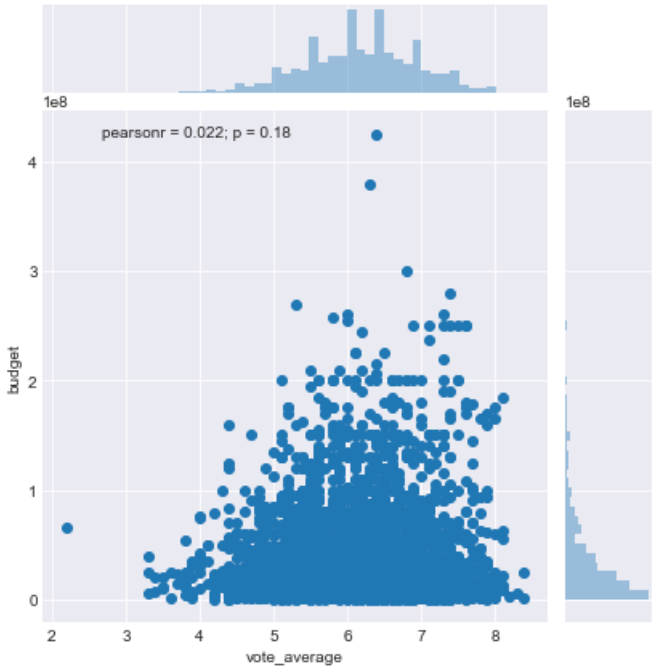
Research Question 10: Most frequent dates of release

In [45]:

```
#application of visualization Seaborn
sns.jointplot(df.vote_average, df.budget)
```

Out[45]:

<seaborn.axisgrid.JointGrid at 0x1a1d583908>



Conclusions

Based on this data analysis, you can draw some conclusions about financial performance, see the most and least successful and popular actors, directors, genres. To see what indicators depend on each other, and which do not. For example: A large budget does not mean that the film will have a high rating and vice versa. A good director or actors often provide a good evaluation of the film.

I think the most useful for users will be to make a forecast about the quality of the newly appeared film, based on its actors, budget and director.