PROJECT 03

IMDB MOVIES

1. Data Preprocessing
2. Exploratory Data Analysis (EDA)
3. Correlation Analysis
4. Data Visualization

By:
Mehrdad Mansourdehghan

GitHUb
YouTube Channel
LinkedIn Profile
Send me E-mail

1. Project Goals

In this project, I'm going to preprocess and clean data, run exploratory data analysis (EDA) and find descriptive statistical measurements, finding the correlation among variables as well as analysis data with making an interactive dashboard for a dataset which contains nearly 7000 movies' data.

2. Language, libraries, tools:

Language: Python, DAX
Libraries: Pandas, NumPy, Seaborn, Matplotlib, Regular Expression, StatsModels, SweetViz
IDE: Jupyter Notebook
Application: Microsoft Excel, Microsoft PowerBI

3. Data

There are 6820 movies in the dataset (220 movies per year, 1986-2016). Each movie has the following attributes:

- **budget:** the budget of a movie. Some movies don't have this, so it appears as 0
- **company:** the production company
- **country:** country of origin
- **director:** the director
- **genre:** main genre of the movie.
- **gross:** revenue of the movie
- **rating:** rating of the movie (R, PG, etc.)
- **released:** release date (YYYY-MM-DD)
- **runtime:** duration of the movie
- **score:** IMDb user rating
- **votes:** number of user votes
- **star:** main actor/actress
- **writer:** writer of the movie
- **year:** year of release

Data Preprocessing

Here you can follow all steps that were taken in this project. Moreover, the codes in Jupyter notebook are exactly based in these processes.

1. Import Packages

First, I import needed packages. I use Pandas because the structure of the dataset is in tabular format. Also, I use NumPy to have this opportunity to run numerical analysis much easier. Finally, I use Seaborn and Matplotlib for visualization during EDA process. I set the size of all figures and visualizes in this project, at the beginning. Also, the style of visualization is "ggplot" based.

```
import pandas as pd
import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import matplotlib
plt.style.use('ggplot')
from matplotlib.pyplot import figure

%matplotlib inline
```

2. load dataset:

Then I load the dataset using with pandas

```
df = pd.read_csv('movie.csv')
```

3. look at data:

just to make sure the data is imported correctly, let's see its 3 first rows:

```
df.head(3)
```

	name	rating	genre	year	released	score	votes	director	writer	star	country	budget	gross	company	runtime
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Jack Nicholson	United Kingdom	19000000.0	46998772.0	Warner Bros.	146.0
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	Brooke Shields	United States	4500000.0	58853106.0	Columbia Pictures	104.0
2	Star Wars: Episode V - The Empire Strikes Back	PG	Action	1980	June 20, 1980 (United States)	8.7	1200000.0	Irvin Kershner	Leigh Brackett	Mark Hamill	United States	18000000.0	538375067.0	Lucasfilm	124.0

4. handle missing values:

I create a for loop to iterate among all columns to realize whether they have null values or not. I'm looking for the number of null values in every single column as well as the percentage of null values.

```
for col in df.columns:
    number_null = df.loc[: , col].isnull().sum()
    perc_null = (number_null / df.shape[0]) * 100
    print('{} - {} - %{}'.format(col, number_null, round(perc_null,3)))
```

```
name - 0 - %0.0
rating - 77 - %1.004
genre - 0 - %0.0
year - 0 - %0.0
released - 2 - %0.026
score - 3 - %0.039
votes - 3 - %0.039
director - 0 - %0.0
writer - 3 - %0.039
star - 1 - %0.013
country - 3 - %0.039
budget - 2171 - %28.312
gross - 189 - %2.465
company - 17 - %0.222
runtime - 4 - %0.052
```

The result shows we must have a different approach to handling null values. For some columns that the percentage of null values are less than 5%, we can drop the records, and for those have more than 5%, we should impute. Let's drop the null values in "rating, released, score, votes, writer, star, country, gross, company and runtime".

```
#drop the null values
print("Dimension before: " , df.shape)
df = df.dropna(subset = ['rating','released','score','votes','writer','star','country','gross','company','runtime'])
print("Dimension after: " , df.shape)
```

```
Dimension before:  (7668, 15)
Dimension after:  (7412, 15)
```

And now we should impute null values for "budget". But before doing this, we must make share, about distribution shape of this column to see whether it's right-skewed or left-skewed. It can be helpful when we want to decide choosing mean or median for imputing.

```
#find distribution shape
print('Skewness :' , round(df['budget'].skew() ,3))

mean_budget = df['budget'].mean()
median_budget = df['budget'].median()

if mean_budget > median_budget:
    print('Mean is bigger than Median. Left Skewed. Median for imputing')
else:
    print('Mean is smaller than Median. Right Skewed. Mean for imputing')
```

```
Skewness : 2.443
Mean is bigger than Median. Left Skewed. Median for imputing
```

So, we choose median:

```
#impute with median
df['budget'] = df['budget'].fillna(median_budget).round(0)
```

Finally, we check the null values again:

```
#check null again
for col in df.columns:
    number_null = df.loc[: , col].isnull().sum()
    perc_null = (number_null / df.shape[0]) * 100
    print('{} - {} - %{}'.format(col, number_null, round(perc_null,3)))
```

```
name - 0 - %0.0
rating - 0 - %0.0
genre - 0 - %0.0
year - 0 - %0.0
released - 0 - %0.0
score - 0 - %0.0
votes - 0 - %0.0
director - 0 - %0.0
writer - 0 - %0.0
star - 0 - %0.0
country - 0 - %0.0
budget - 0 - %0.0
gross - 0 - %0.0
company - 0 - %0.0
runtime - 0 - %0.0
```

5. Sanity checks on “Year”:

Since we have two columns for the year, we must check the sanity (correctness) to realizer whether the year column is based on the release date or not. By the way, we want to replace the new year column (that is extracted from release date) with the old year column.

```
import re

# Create a new column 'year' in the DataFrame
df['Years'] = ''
df = df.reset_index(drop=True)

# Define a regular expression pattern to match the year
pattern = r"\b\d{4}\b"

# Iterate over the rows in the DataFrame
for i in range(df.shape[0]):
    date_string = df.iloc[i, 4]  # Assuming the date is in the 5th column (index 4)
    # Search for the year using the regular expression pattern
    match = re.search(pattern, date_string)
    if match:
        year = match.group(0)
        df.at[i, 'Years'] = year  # Assign the extracted year to the 'year' column
    else:
        df.at[i, 'Years'] = 'Year not found'  # Assign a default value when year is not found
```

Now we can drop the old year column.

```
df = df.drop('year',axis=1)
```

6. Handle duplicate rows:

Now we should handle duplicate rows. Since all values might be same, we just we need to check whether there are two rows that all values in all columns are the same or not.

```
def has_duplicate_rows(data):
    df = pd.DataFrame(data)
    duplicate_rows = df.duplicated()
    return any(duplicate_rows)

has_duplicate_rows(df)
```

```
False
```

The result shows fortunately we don’t have duplicate rows.

Exploratory Data Analysis (EDA)

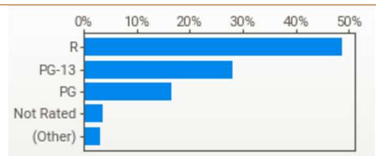
In this section we do descriptive statistical analysis to know data much more. By doing this, we can realize their distribution, central tendency measurement, the dispersion measurement and shape of the data. First, we import SweetViz package that is very helpful in descriptive statistical analysis. Moreover, we write some functions for Matplotlib and Seaborn to add more information about the statistical analysis by drawing distribution plots (quantitative variables) and bar charts (qualitative variables).

1. Qualitative Variables

Here we analyze qualitative variables, and we see each label in every single variable account for the highest frequency. Then, we can see the statistical interpretation for categorical variables:

rating

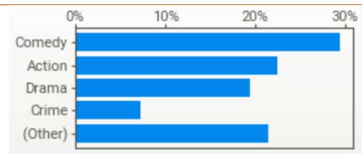
R	3,608	49%
PG-13	2,089	28%
PG	1,220	16%
Not Rated	258	3%
G	152	2%
Unrated	45	<1%
NC-17	23	<1%
TV-MA	9	<1%



The "R" , "PG-13" and "PG" account for the most frequent rating.

genre

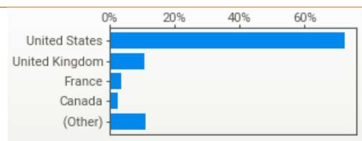
Comedy	2,182	29%
Action	1,666	22%
Drama	1,439	19%
Crime	536	7%
Biography	429	6%
Adventure	419	6%
Animation	331	4%
Horror	304	4%



The "Comedy", "Action" and "Drama" were the types of genres that directors made.

country

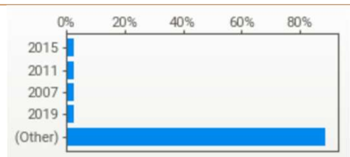
United States	5,358	72%
United Kingdom	790	11%
France	255	3%
Canada	181	2%
Germany	114	2%
Australia	85	1%
Japan	70	<1%



The most movies are made in "US" and "United Kingdom".

year

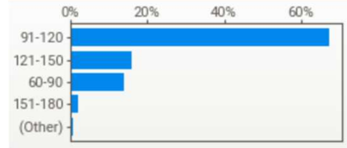
2015	210	3%
2011	210	3%
2007	210	3%
2019	209	3%
2003	204	3%
2008	203	3%
2018	203	3%
1994	203	3%



The in "2015", "2011" and "2007" companies made more movies compared to other years.

runtime

91-120	4,990	67%
121-150	1,190	16%
60-90	1,034	14%
151-180	151	2%
181-210	38	<1%
211-240	5	<1%



The most movies' durations long between "91 to 120" minutes.

director

38	<1%	Woody Allen
31	<1%	Clint Eastwood
27	<1%	Steven Spielberg
25	<1%	Directors
24	<1%	Ron Howard
23	<1%	Steven Soderbergh
23	<1%	Ridley Scott
22	<1%	Joel Schumacher

The "Woody Allen", "Clint Eastwood" and "Steven Spielberg" made many more than other.

writer

37	<1%	Woody Allen
31	<1%	Stephen King
25	<1%	Luc Besson
25	<1%	John Hughes
15	<1%	William Shakespeare
15	<1%	David Mamet
13	<1%	Pedro Almodóvar

The most movies are made with the scenarios that are written by "Woody Allen", "Stephen King", "Luc Besson" and "John Hughes".

star

43	<1%	Nicolas Cage
41	<1%	Tom Hanks
41	<1%	Robert De Niro
37	<1%	Denzel Washington
34	<1%	Bruce Willis
33	<1%	Tom Cruise
33	<1%	Johnny Depp

The "Nicolas Cage", "Tom Hanks" and" "Robert De Niro" played in the most movies compared to other actresses or actors.

companu

376	5%	Universal Pictures
332	4%	Warner Bros.
332	4%	Columbia Pictures
319	4%	Paramount Pictures
240	3%	Twentieth Century Fox
173	2%	New Line Cinema
132	2%	Touchstone Pictures

The most movies are made by "Universal Pictures", "Warner Bros" and "Columbia Pictures".

2. Quantitative Variables

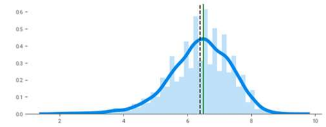
Now, let's analyze descriptive statics for quantitative variables. In this section we can find central tendency, dispersion, and shape measurements. Then we draw the distribution plot to compare with normal distribution. Also, we can see how those variables are related to other variables.

```python
def kde_plot(x):
    import seaborn as sns
    import matplotlib.pyplot as plt

    plt.figure(figsize = (8,3))
    sns.distplot(df[x], kde_kws={"lw": 5}, hist_kws = {'alpha': 0.25})
    sns.despine(left = True)

    mean_age = df[x].mean()
    median_age = df[x].median()

    plt.axvline(mean_age, color ='black', linestyle ='dashed')
    plt.axvline(median_age, color ='green', linestyle ='solid')
    plt.xlabel('')
    plt.ylabel('')

    return plt.show()
```

Now we can see the statistical interpretation for quantitative variables:

score

MAX	9.30	RANGE	7.40
95%	7.80	IQR	1.30
Q3	7.10	STD	0.963
MEDIAN	6.50	VAR	0.928
AVG	6.40		
Q1	5.80	KURT.	0.956
5%	4.70	SKEW	-0.613
MIN	1.90	SUM	47,414



The average score of movies is 6.40.

Half of the movies scored less than 6.5

The difference between the maximum and minimum score is 7.4.

25% of the movies are scored less than or equal to 5.8

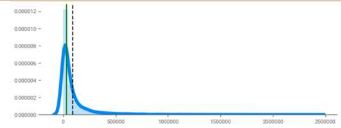
75% of the movies are scored less than or equal to 7.1

50% of the movies are scored between 5.8 and 7.1

The distribution is almost normal.

votes

MAX	2.4M	RANGE	2.4M
95%	0.4M	IQR	86,000
Q3	0.1M	STD	165k
AVG	0.1M	VAR	27.3B
MEDIAN	0.0M		
Q1	0.0M	KURT.	36.1
5%	0.0M	SKEW	4.85
MIN	0.0M	SUM	672.7M



The average votes of movies is 0.1M

Half of the movies voted less than 0.034M

The difference between the maximum and minimum votes is 2.4M.

25% of the movies are voted less than or equal to 0.010M

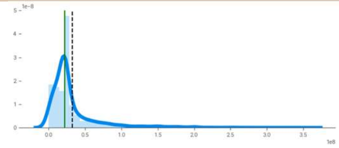
75% of the movies are scored less than or equal to 0.096M

50% of the movies are scored between 0.010M and 0.096M

The distribution is considerably right skewed (big outlier).

budget

MAX	356.0M	RANGE	356.0M
95%	105.0M	IQR	19.0M
Q3	33.0M	STD	36.1M
AVG	32.2M	VAR	1303.7T
MEDIAN	21.8M		
Q1	14.0M	KURT.	11.6
5%	3.0M	SKEW	3.03
MIN	0.0M	SUM	238.6B



The average budget for producing movies is 32.2M.

Half of the movies are made with the amount of budget less than 21.8M

The difference between the maximum and minimum budget is 356M.

25% of the movies are made with budget less than or equal to 14M

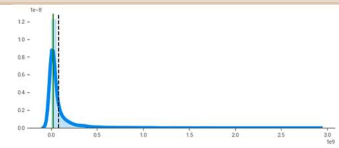
75% of the movies are made with budget less than or equal to 33M

50% of the movies are made with budget between 14M and 33M

The distribution is considerably right skewed (big outlier).

gross

MAX	2.8B	RANGE	2.8B
95%	0.4B	IQR	71.8M
Q3	0.1B	STD	166.2M
AVG	0.1B	VAR	27627.9T
MEDIAN	0.0B		
Q1	0.0B	KURT.	45.3
5%	0.0B	SKEW	5.30
MIN	0.0B	SUM	585.5B



The average gross of movies is 0.1B

Half of the movies could make money less than 0.020B

The difference between the maximum and minimum gross is 2.8B

25% of the movies could bring benefit less than or equal to 0.0046B

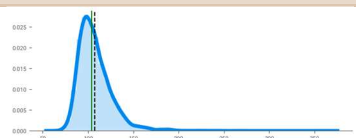
75% of the movies could bring benefit less than or equal to 0.0764B

50% of the movies could bring benefit between 0.0046B and 0.0764B

The distribution is considerably right skewed (big outlier).

runtime

MAX	366	RANGE	303
95%	140	IQR	21.0
Q3	116	STD	18.5
AVG	107	VAR	343
MEDIAN	104		
Q1	95	KURT.	13.8
5%	85	SKEW	2.13
MIN	63	SUM	796k



The average duration of movies is 107 minutes

Half of the movies last less than 104 minutes

The difference between the maximum and minimum duration is 303 minutes.

25% of the movies last less than or equal to 95 minutes

75% of the movies last less than or equal to 116 minutes

50% of the movies last between 95 minutes and 116 minutes

The distribution is considerably right skewed (big outlier).

3. Maximum & Minimum Values:

We can also, see the minimum and maximum values for each quantitative variable:

```python
#minimum score
df[df['score'] == df['score'].min()]
```

	name	rating	genre	released	score	votes	director	writer	star	country	budget	gross	company	runtime	Years
4386	Superbabies: Baby Geniuses 2	PG	Comedy	August 27, 2004 (United States)	1.9	30000.0	Bob Clark	Robert Grasmere	Jon Voight	Germany	20000000.0	9448644.0	ApolloMedia Distribution	88.0	2004
5094	Disaster Movie	PG-13	Comedy	August 29, 2008 (United States)	1.9	88000.0	Jason Friedberg	Jason Friedberg	Carmen Electra	United States	20000000.0	34816824.0	Lionsgate	87.0	2008
5142	The Hottie & the Nottie	PG-13	Comedy	February 21, 2008 (Russia)	1.9	36000.0	Tom Putnam	Heidi Ferrer	Paris Hilton	United States	21800000.0	1596232.0	Purple Pictures	91.0	2008

```python
#maximum score
df[df['score'] == df['score'].max()]
```

	name	rating	genre	released	score	votes	director	writer	star	country	budget	gross	company	runtime	Years
2273	The Shawshank Redemption	R	Drama	October 14, 1994 (United States)	9.3	2400000.0	Frank Darabont	Stephen King	Tim Robbins	United States	25000000.0	28817291.0	Castle Rock Entertainment	142.0	1994

```python
#minimum votes
df[df['votes'] == df['votes'].min()]
```

	name	rating	genre	released	score	votes	director	writer	star	country	budget	gross	company	runtime	Years
582	Petit Con	R	Comedy	April 19, 1985 (United States)	6.2	105.0	Gérard Lauzier	Gérard Lauzier	Guy Marchand	France	21800000.0	127426.0	Gaumont International	90.0	1985

```python
#maximum votes
df[df['votes'] == df['votes'].max()]
```

	name	rating	genre	released	score	votes	director	writer	star	country	budget	gross	company	runtime	Years
2273	The Shawshank Redemption	R	Drama	October 14, 1994 (United States)	9.3	2400000.0	Frank Darabont	Stephen King	Tim Robbins	United States	25000000.0	2.881729e+07	Castle Rock Entertainment	142.0	1994
5032	The Dark Knight	PG-13	Action	July 18, 2008 (United States)	9.0	2400000.0	Christopher Nolan	Jonathan Nolan	Christian Bale	United States	185000000.0	1.005974e+09	Warner Bros.	152.0	2008

```python
#minimum budget
df[df['budget'] == df['budget'].min()]
```

	name	rating	genre	released	score	votes	director	writer	star	country	budget	gross	company	runtime	Years
3136	Following	R	Crime	November 5, 1999 (United Kingdom)	7.5	89000.0	Christopher Nolan	Christopher Nolan	Jeremy Theobald	United Kingdom	6000.0	48482.0	Next Wave Films	69.0	1999

```python
#maximum budget
df[df['budget'] == df['budget'].max()]
```

	name	rating	genre	released	score	votes	director	writer	star	country	budget	gross	company	runtime	Years
7221	Avengers: Endgame	PG-13	Action	April 26, 2019 (United States)	8.4	903000.0	Anthony Russo	Christopher Markus	Robert Downey Jr.	United States	356000000.0	2.797501e+09	Marvel Studios	181.0	2019

```python
#minimum gross
df[df['gross'] == df['gross'].min()]
```

	name	rating	genre	released	score	votes	director	writer	star	country	budget	gross	company	runtime	Years
3021	Trojan War	PG-13	Comedy	October 1, 1997 (Brazil)	5.7	5800.0	George Huang	Andy Burg	Will Friedle	United States	15000000.0	309.0	Daybreak	85.0	1997

```python
#maximum gross
df[df['gross'] == df['gross'].max()]
```

	name	rating	genre	released	score	votes	director	writer	star	country	budget	gross	company	runtime	Years
5233	Avatar	PG-13	Action	December 18, 2009 (United States)	7.8	1100000.0	James Cameron	James Cameron	Sam Worthington	United States	237000000.0	2.847246e+09	Twentieth Century Fox	162.0	2009

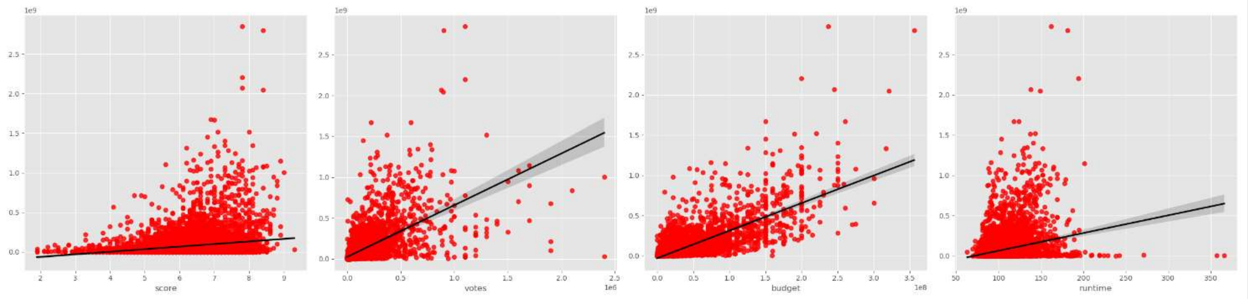
Correlation Analysis

In this section, I'm going to see whether there is relationship between variables with gross. Since the gross is very crucial for each producer, it makes sense that we see how many other factors are correlated to gross. For doing this, we should consider two different ways. One way is correlation between numeric variables with gross, and the second way is correlation between categorical variables with gross. For the first one, we use Pearson correlation and for the second one, we use ANOVA test.

1. Pearson Correlation

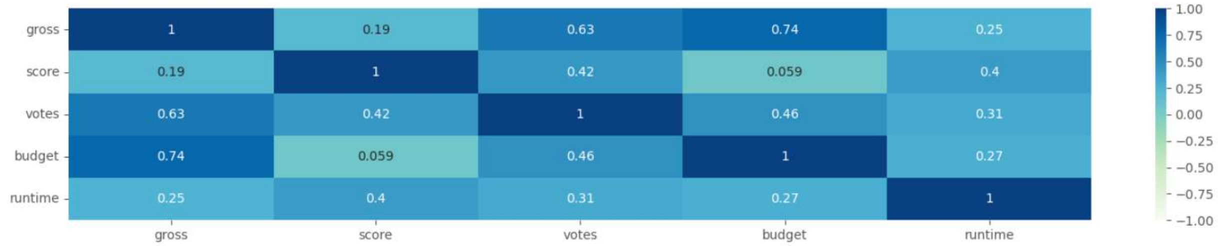
First, based on OLS method, we analyze the correlation and regression between variables on plot.

```
#declare numeric variable
numeric = ['score','votes','budget','runtime']

# Create a grid of subplots
fig, axes = plt.subplots(1, 4, figsize=(25, 6))

# Flatten the axes array to make it 1D
axes = axes.ravel()

# Loop through each subplot and plot sns.regplot
for i, col in enumerate(numeric):
    sns.regplot(x=col, y='gross', data=df, ax=axes[i], scatter_kws={"color": "red"}, line_kws={"color":"black"})
    axes[i].set_xlabel(col)
    axes[i].set_ylabel('')

# Adjust spacing between subplots
plt.tight_layout()

# Show the plot
plt.show()
```



For all numeric variables, there is a positive relationship between them and gross. However, this relationship between gross with votes and budget are strong, but with scores are weak. For knowing the exact correlation, we can use Pearson function:

```python
pearson_cor = df[['gross','score','votes','budget','runtime']].corr(method = 'pearson')

plt.figure(figsize = (18,3))

sns.heatmap(pearson_cor,
            vmin = -1,
            vmax = 1,
            cmap = "GnBu",
            annot = True)
plt.show()
```



As we can see, we can consider the budget as the most numeric variable for having more gross, and if we budget more, we will have more gross. Even though having a higher score and runtime of a movie can increase the gross, it is not very considerable.

2. ANOVA Test

For categorical variables, first we should see whether a particular categorical variable impacts gross or not (it is significant or not). For doing this, we run ANOVA test to compare the meaningful difference between means. After that, for the variables that are significant, we run pairwise descriptive analysis for all labels in the particular categorical variable and then compare their impacts on the gross.

```python
cat_list = ['name', 'rating', 'genre','director', 'writer', 'star', 'country', 'company', 'Years']

import statsmodels.api as sm
from statsmodels.formula.api import ols

for i in cat_list:
    formula = 'gross ~ {}'.format(i)
    model = ols(formula, data=df).fit()
    anova = sm.stats.anova_lm(model, typ=2)
    p_value = anova.iloc[0,3]

    print('P-value for gross ~ {}: {}'.format(i , p_value))
```

```
P-value for gross ~ name: 0.9496631408836501
P-value for gross ~ rating: 1.308189908902382e-99
P-value for gross ~ genre: 2.1239925631751138e-179
P-value for gross ~ director: 4.231152789248918e-89
P-value for gross ~ writer: 0.004341822809523021
P-value for gross ~ star: 5.204885271191884e-09
P-value for gross ~ country: 1.2175041260852342e-17
P-value for gross ~ company: 1.414118645393978e-15
P-value for gross ~ Years: 1.3632496446607623e-81
```

According to results, we can come up that name is not significant variables to explain how a movie can make gross, because the p-value is more than 0.05 and we cannot reject null hypothesis. So, in the next step, we want to see for each label in the above categorical variables, which of them has the most impact on gross. Thus, I make s function to calculate the mean for each label in every single categorical variable, and then shows just first top positive influencer:

```
def mean_pairwise(cat_var):
    mean_by = df.groupby(cat_var)['gross'].mean()
    mean_by = pd.DataFrame(mean_by)
    mean_by = mean_by.sort_values(by=['gross'], inplace=False, ascending=False)

    return mean_by.head(5)
```

rating	gross
G	1.420433e+08
PG-13	1.309839e+08
TV-PG	1.202498e+08
PG	1.066129e+08
TV-MA	7.917078e+07

The most gross can be gained by rate "G"

genre	gross
Animation	2.413567e+08
Family	2.157876e+08
Action	1.458350e+08
Adventure	1.095587e+08
Mystery	1.011835e+08

The most gross can be gained by genre "Animation"

director	gross
Anthony Russo	1.368850e+09
Kyle Balda	1.097122e+09
Josh Cooley	1.073395e+09
Chris Buck	1.059909e+09
Lee Unkrich	9.373943e+08

The most gross can be gained by director "A. Russo"

writer	gross
Christopher Markus	1.083883e+09
Irene Mecchi	1.083721e+09
Rick Jaffa	1.076159e+09
Byron Howard	1.024121e+09
J.R.R. Tolkien	9.970720e+08

The most gross can be gained by writer "Ch. Markus"

star	gross
Donald Glover	1.670728e+09
Daisy Ridley	1.120174e+09
Neel Sethi	9.665549e+08
Craig T. Nelson	9.381233e+08
Chris Pratt	8.797427e+08

The most gross can be gained by star "D. Glover"

country	gross
Malta	3.527941e+08
New Zealand	2.647805e+08
China	2.177334e+08
Finland	1.691938e+08
United States	9.020570e+07

The most gross can be gained by country "Malta"

company	gross
Marvel Studios	1.255466e+09
Illumination Entertainment	1.097122e+09
Fairview Entertainment	9.665549e+08
B24	8.806815e+08
Avi Arad Productions	8.560852e+08

The most gross can be gained by company "Marvel Studios"

Years	gross
2020	1.668662e+08
2017	1.475836e+08
2016	1.410022e+08
2018	1.407065e+08
2019	1.402180e+08

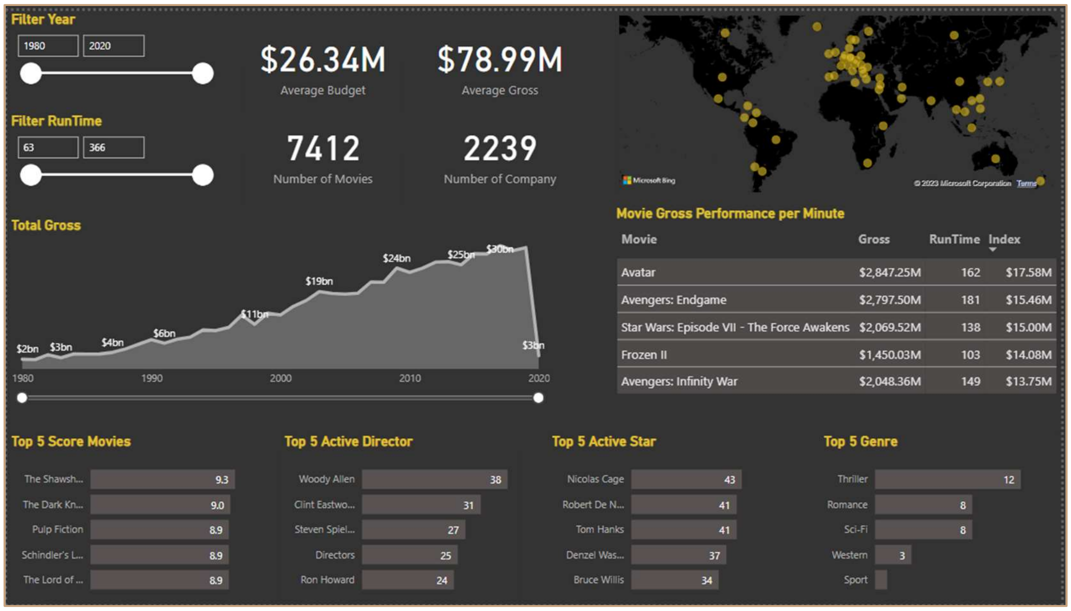
The most gross can be gained by year "2020"

Data Visualization

In this section we want to analyze data based on visualization on "data_cleaned" file, because I believe the best way to analyze the data is in visualized way. So, by doing this we can answer some ad-hoc questions that might be asked in daily-basis business. So far, we have a good image of the data, and we can help managers or users who are willing to have insight about the data. I use Microsoft Power BI to create an interactive dashboard.

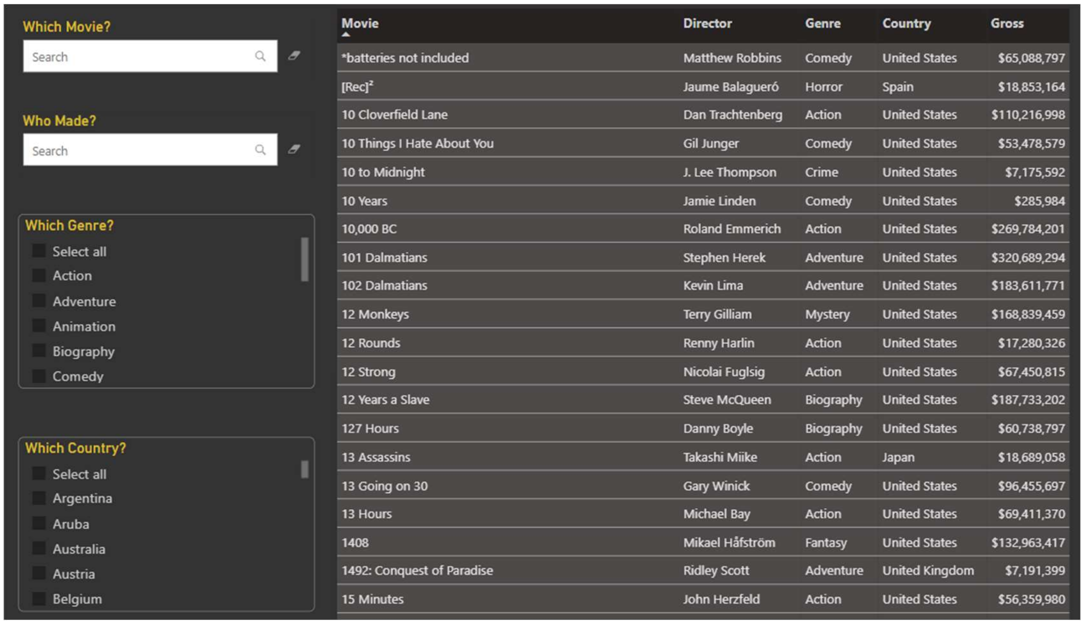
1. Dashboard

The dashboard contains so many elements that indicate information about the data. In the top left, we have two carousel slicers we can use to filter data based on year and movie's length. Next to them, there are some cards where we can find some statistical information about movie(s), and on the top right, we have a filled map to have better view about the geographical distribution of movies all around the world. In the middle of the report, we have a trend chart that indicates the total gross for every year and you can use the small carousel (bottom of the chart) to filter the years. Also, the table shows the performance of the top-5 movies in terms of how much they can benefit and earn money per minute. Finally, at the bottom of the dashboard, we can see some bar charts that show top-5 movie, active director, stars, and the best genre (in terms of number of produced movies). Having said that, if you hover mouse on the top-5 movies, you can see their more information there.



2. Movie Table

Having a whole table is always good practice to give this chance to users for iterative among data. In the next page, there is a table that show list of all movies with some information about them. Moreover, there are some filters that can help you to get closer to your target.



Movie	Director	Genre	Country	Gross
*batteries not included	Matthew Robbins	Comedy	United States	$65,088,797
[Rec]²	Jaume Balagueró	Horror	Spain	$18,853,164
10 Cloverfield Lane	Dan Trachtenberg	Action	United States	$110,216,998
10 Things I Hate About You	Gil Junger	Comedy	United States	$53,478,579
10 to Midnight	J. Lee Thompson	Crime	United States	$7,175,592
10 Years	Jamie Linden	Comedy	United States	$285,984
10,000 BC	Roland Emmerich	Action	United States	$269,784,201
101 Dalmatians	Stephen Herek	Adventure	United States	$320,689,294
102 Dalmatians	Kevin Lima	Adventure	United States	$183,611,771
12 Monkeys	Terry Gilliam	Mystery	United States	$168,839,459
12 Rounds	Renny Harlin	Action	United States	$17,280,326
12 Strong	Nicolai Fuglsig	Action	United States	$67,450,815
12 Years a Slave	Steve McQueen	Biography	United States	$187,733,202
127 Hours	Danny Boyle	Biography	United States	$60,738,797
13 Assassins	Takashi Miike	Action	Japan	$18,689,058
13 Going on 30	Gary Winick	Comedy	United States	$96,455,697
13 Hours	Michael Bay	Action	United States	$69,411,370
1408	Mikael Håfström	Fantasy	United States	$132,963,417
1492: Conquest of Paradise	Ridley Scott	Adventure	United Kingdom	$7,191,399
15 Minutes	John Herzfeld	Action	United States	$56,359,980

- END -

https://www.linkedin.com/in/mehrdad-dehghan-527b553a/
https://www.youtube.com/@mehrdad_analyst