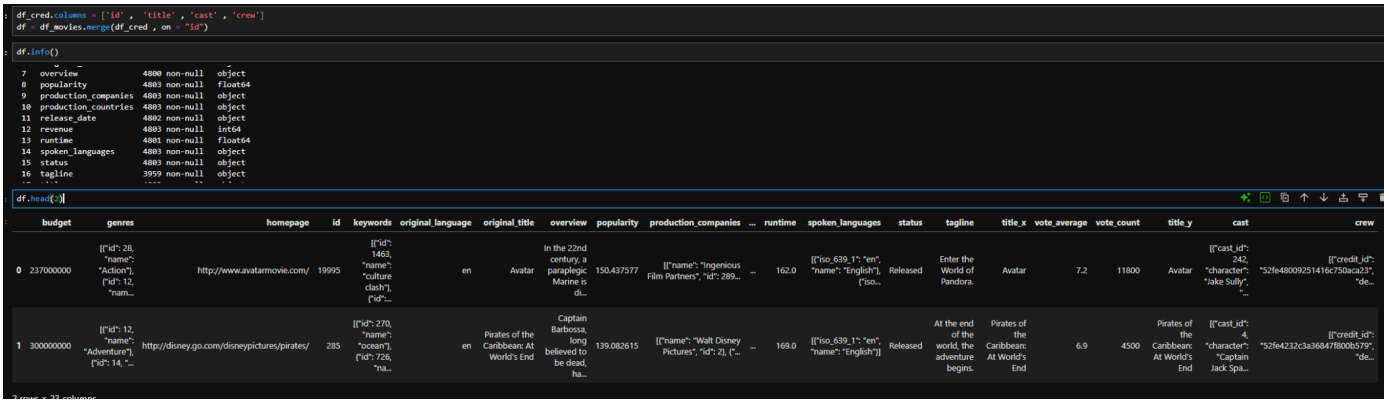
TMDb movie data investigation

Data overview :

Metadata on ~5,000 movies from TMDb. This dataset was generated from The Movie Database API. This product uses the TMDb API but is not endorsed or certified by TMDb. Their API also provides access to data on many additional movies, actors and actresses, crew members, and TV shows.

Main Investigation : Budget and popularity relation and factors.

First we started to explore the two datasets in the file, we have a tmdb_5000_credits.csv and tmdb_5000_movies.csv. Secondly we have merged the two data sets and get the following result:



```
df_cred.columns = ['id' , 'title' , 'cast' , 'crew']
df = df_movies.merge(df_cred , on = "id")
```

```
df.info()
```

```
 7   overview              4800 non-null   object
 8   popularity            4803 non-null   float64
 9   production_companies  4803 non-null   object
 10  production_countries  4803 non-null   object
 11  release_date          4802 non-null   object
 12  revenue               4803 non-null   int64
 13  runtime               4801 non-null   float64
 14  spoken_languages      4803 non-null   object
 15  status                4803 non-null   object
 16  tagline               3959 non-null   object
```

```
df.head(2)
```

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_companies	...	runtime	spoken_languages	status	tagline	title_x	vote_average	vote_count	title_y	cast	crew
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "Ingenious Film Partners", "id": 289...	...	162.0	[{"iso_639_1": "en", "name": "English"}, {"iso...	Released	Enter the World of Pandora.	Avatar	7.2	11800	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe4800925141 6c750aca23", "de...
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[{"name": "Walt Disney Pictures", "id": 2}, {"...	...	169.0	[{"iso_639_1": "en", "name": "English"}]	Released	At the end of the world, the adventure begins.	Pirates of the Caribbean: At World's End	6.9	4500	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...

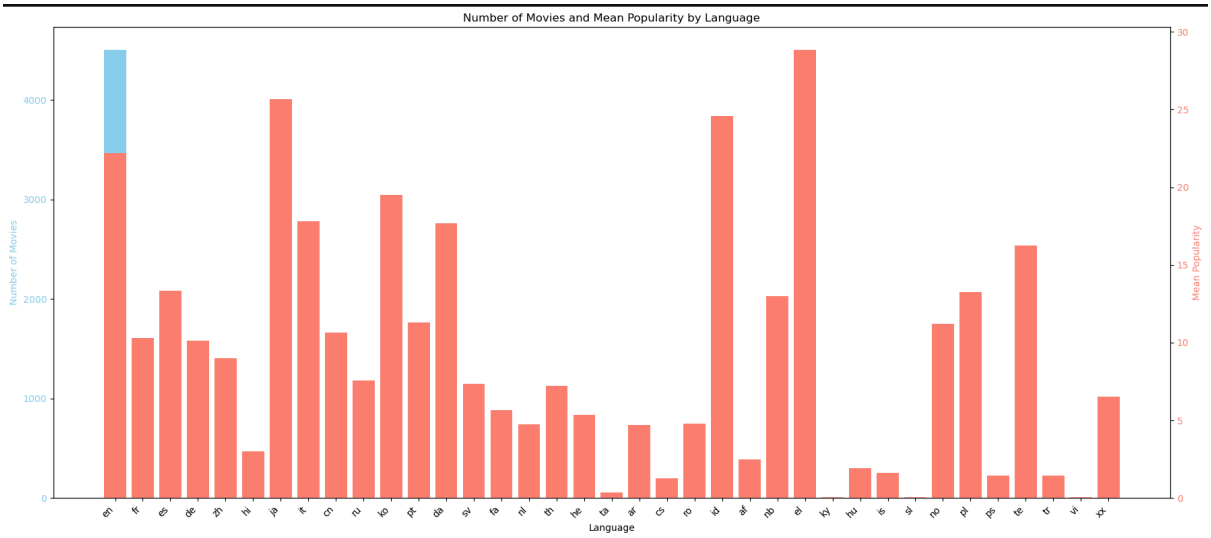
As we can see we need to do some cleaning and tidy our data. We have deleted the repeated columns and renamed others for better use.
Our main question is : **how the budget may affect the popularity around the years , in general and according to the original language**.

First investigation: Original language vs popularity

We have created our dataset using group by and aggregation to get the following:
The total number of movies and the mean and popularity for each language.

```
y = df_copy.groupby('original_language').agg(movie_count=('id', 'size'), mean_popularity=('popularity', 'mean') ).reset_index()
```

We have got the following chart:



Salmon : mean of popularity
Sky blue : number of movies

Conclusion :
Some languages have a high popularity compared to the number of movies while language like english is not the highest even though it has the highest number of movies.

Second investigation: Budget Vs Popularity

To get the first insight we have created the dataset for the top 20 movies to make our first check :

```
# Budget Vs Popularity
```

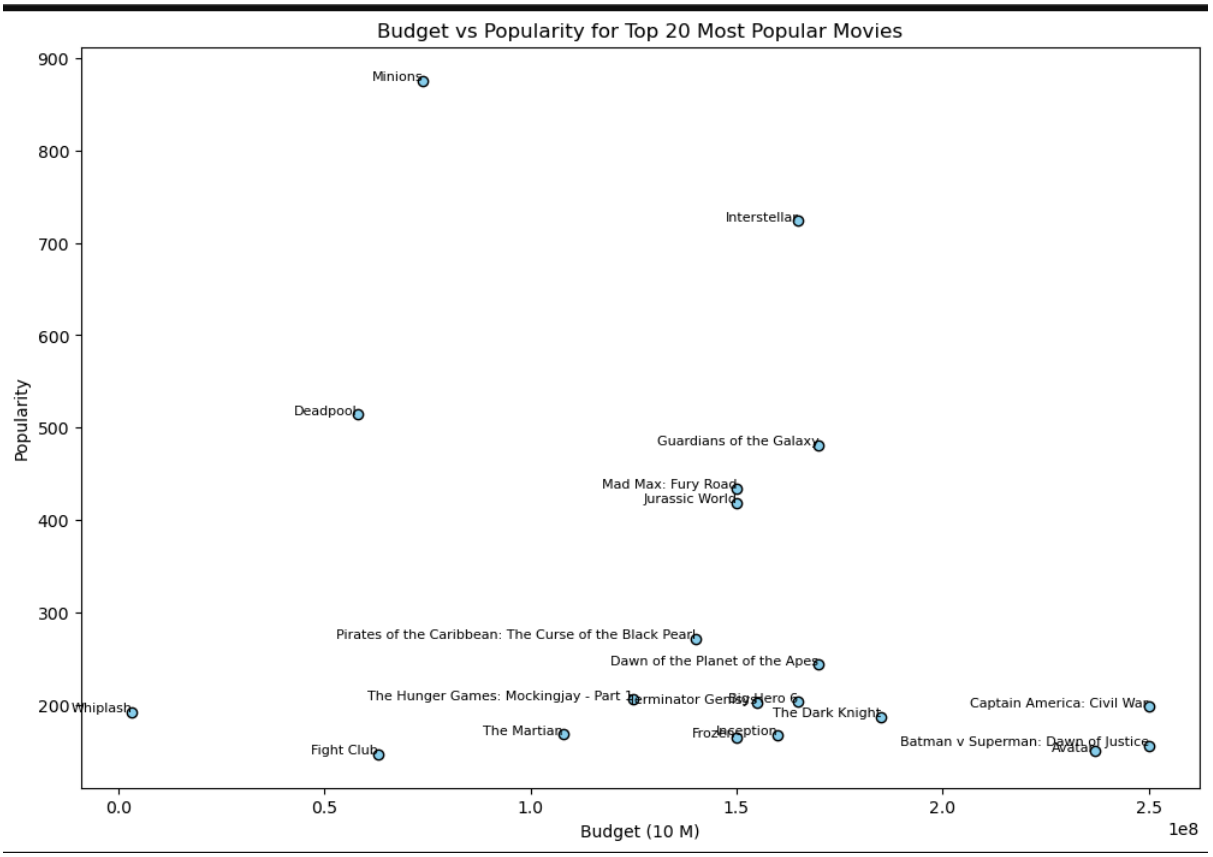
```
df_budget_vs_pop = df.groupby(['popularity', 'title', 'production_companies','release_date']).agg(budget_mean=('budget', 'mean')).reset_index().sort_values(by='popularity', ascending=False).head(20)
```

```
df_budget_vs_pop.columns
```

```
Index(['popularity', 'title', 'production_companies', 'release_date',
       'budget_mean'],
      dtype='object')
```

```
df_budget_vs_pop.describe()
```

We got the following chart by using the scatter plot to be able to see the three factors:



Conclusion :
Budget is not the main cause for movies to be popular , some movies have the highest budget but least popularity .

Let’s check the budget along with the time

Third investigation: Budget Vs Popularity along time

Does the budget increase or decrease during time and does that affect the popularity?

First let's see the release date for budget and popularity for top 20 movies to get first insights. To be able to do that we can see that the release date is by dd/mm/yyyy
We need to make it by year and sum the total movies and mean of budget.
To get the year we have used the datetime function to transfer it and then add only the year column as following :

```
df_copy['release_date'] = pd.to_datetime(df['release_date'])
df_copy['year'] = df_copy['release_date'].dt.year
```

We got the following result:

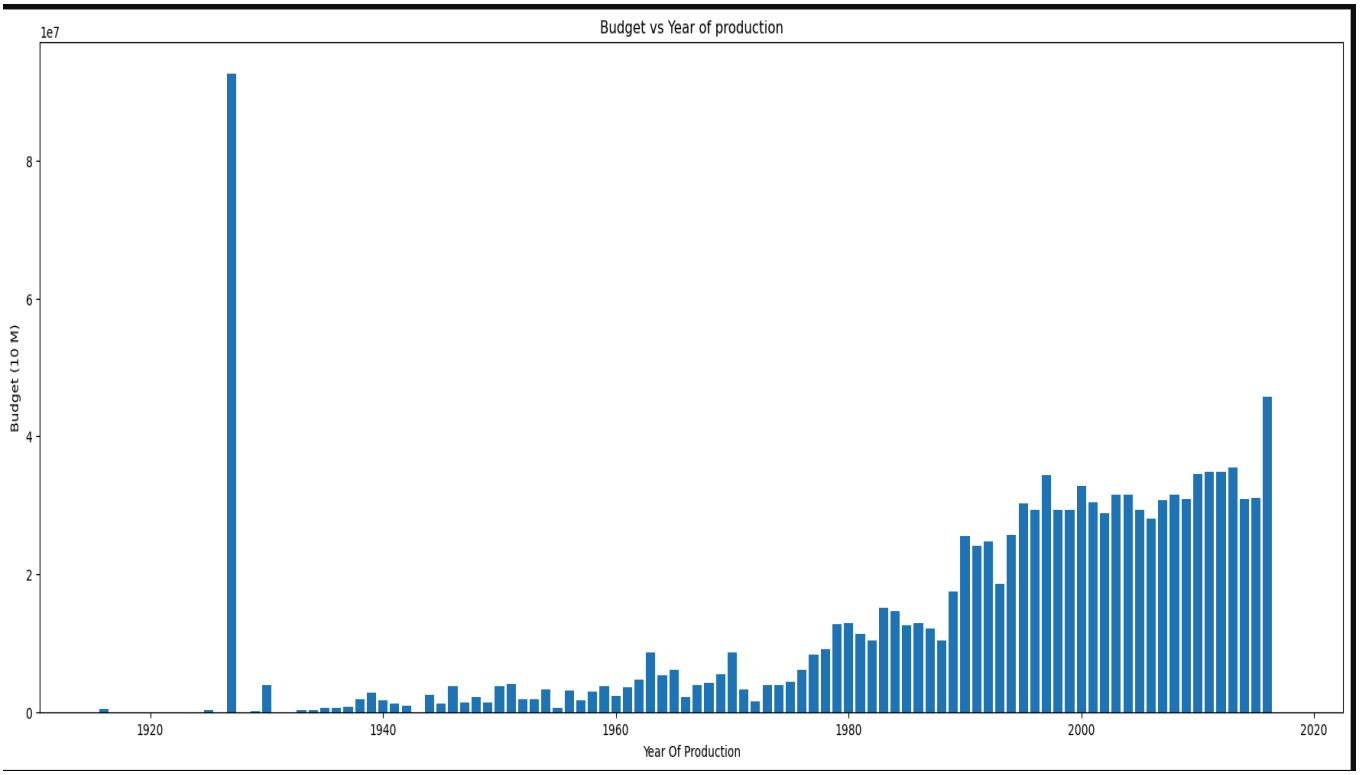
cast	crew	year
{"cast_id": 242, haracter": ake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...	2009.0

Now we create out dataset as following:

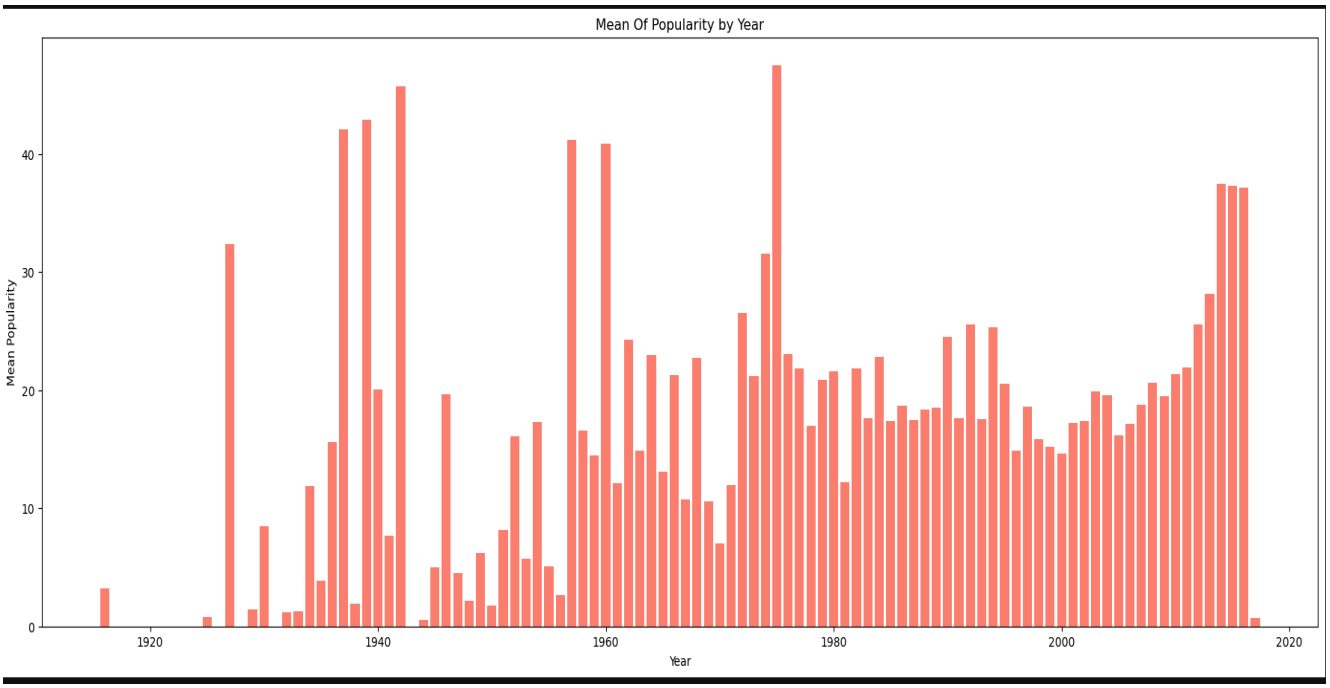
```
y4 = df_copy.groupby(['year']).agg(mean_budget=('budget', 'mean'),
    sum_movies=('id', 'sum'),
    mean_popularity=('popularity', 'mean')).reset_index()
```

We used the bar plot and got the following results :

1- first figure Year of production vs budget:



2- second figure Mean Of Popularity by Year :



Conclusion :
1- Popularity of movies is the period between 1930-1960 at the peak while on the other hand budget was at its lowest at these times

2- after 1960 the budget started to increase same as the popularity

Future Studies for budget vs popularity :

Study budget and popularity according to genres and production companies.

Some methods to enhance the code:

We have created functions for the most reused charts in our investigation like bar and scatter as the following and reused when needed:

Wriring a functions for plotting:

```
def bar_pltot(df_1, x_para, y_para , bar_color , x_label , y_label ,title):
    plt.figure(figsize=(22, 8))
    plt.bar(df_1[x_para],df_1[y_para], color=bar_color)
    plt.title(title)
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.show()

def scatter_plot(df_1 , x_para, y_para , point_row , scatter_color , x_label , y_label  ,title):
    plt.figure(figsize=(12, 8))
    plt.scatter(df_1[x_para], df_1[y_para], color= scatter_color, edgecolor='black')
    plt.title(title)
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    for i, row in df_1.iterrows():
        plt.text(row[x_para], row[y_para], row[point_row], fontsize=8, ha='right')
    plt.show()
```

Resources used :

- [ ] https://medium.com/@veronica.isiaho/writing-python-functions-for-plotting-graphs-da52c3998d84
- [ ] https://pandas.pydata.org/docs/user_guide/visualization.html
- [ ] https://www.w3schools.com/python/matplotlib_plotting.asp
- [ ] https://saturncloud.io/blog/how-to-split-a-date-column-into-separate-day-month-year-columns-in-pandas/