Project: Movies Dataset Analysis

Table of Contents

- Introduction
- Data Wrangling
- Exploratory Data Analysis
- Conclusions

Introduction

Dataset Description

This is movie data from the Movie database(TMDb) that contains over 10,000 records an 21 columns.
The dataset contains columns the following columns:

id : movie id
imdb_id: contains the movie id as in IDMB database
popularity: shows the movie rating
budget: shows the expenses incurrred in production
revenue: shows the revenue generated after movie releases
original_title: describes the movie's original title
cast: shows the movie's cast
homepage: movies webpage
director: the movie directors involved in production
tagline : movie taglines

keywords : keywords used to identify the movies
overview : shows he general descript
runtime : shows how long a movie is
genres : the different genres the movies belong to
production_companies : the companies involved in the movie's production release_date : tthe date the movie was released
vote_count : the count of the votes that deyermined the movie's popularity
vote_average :
release_year : the year the movie was released
budget_adj: budget adjustments as per the inflation rates in terms of the 2010 dollars
revenue_adj : revenue adjustments as per the inflation rates in terms of the 2010 dollars

Question(s) for Analysis

1. What is the average runtime for most movies? and What was the highest revenue generated?
2. Which genre of movies are the most produced?
3. Which production companies produced the most movies?
4. Which director directed most movies?
5. Which actor was casted the most movies?
6. which month had the highest number of movie releases?
7. Does a Movies' Popularity affects its revenue?
8. Which decade had the most movie releases?
9. Did most movies have higher ratings and votes?
10. What was the highest profits made?

```
#import required libraries

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
```

Data Wrangling

General Properties

```
#load the idmb CSV file into a pandas dataframe
#The file is on my github repository
df = pd.read_csv('https://raw.github.com/FMurunga/TMDB-Movie
df.head()
```

	id	imdb_id	popularity	budget	revenue
0	135397	tt0369610	32.985763	150000000	151352881(
1	76341	tt1392190	28.419936	150000000	37843635
2	262500	tt2908446	13.112507	110000000	29523820

```
#Check the shape of the dataframe
df.shape
```

```
(10866, 21)
```

The dataset has 10,866 rows and 21 columns

4	168259	tt2820852	9.335014	190000000	150624936(

```
#check the information about the dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    10866 non-null  int64
```

```
 1   imdb_id                 10856 non-null  object
 2   popularity              10866 non-null  float64
 3   budget                  10866 non-null  int64
 4   revenue                 10866 non-null  int64
 5   original_title          10866 non-null  object
 6   cast                    10790 non-null  object
 7   homepage                2936 non-null   object
 8   director                10822 non-null  object
 9   tagline                 8042 non-null   object
 10  keywords                9373 non-null   object
 11  overview                10862 non-null  object
 12  runtime                 10866 non-null  int64
 13  genres                  10843 non-null  object
 14  production_companies    9836 non-null   object
 15  release_date            10866 non-null  object
 16  vote_count              10866 non-null  int64
 17  vote_average            10866 non-null  float64
 18  release_year            10866 non-null  int64
 19  budget_adj              10866 non-null  float64
 20  revenue_adj             10866 non-null  float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

From the information above the following can be observed:

- Columns such as: idmbId,cast, homepage,director,tagline,keywords,,overview,runtime,genres,production_companies have missing values
- The datatypes of each column: most values in the dataset have object attributes and a few have float and integer data types
- The release_date column is an object data type
-

idmbId,cast,
homepage,director,tagline,keywords,,overview,runtime,genres,pr
oduction_companies have missing values

Note: **Most of these columns might not be useful in our analysis and thus can be dropped**

```
#cchecking for Unique values to determine the analysis apprr
df.nunique()
```

```
id                    10865
imdb_id               10855
popularity            10814
budget                  557
revenue                4702
original_title        10571
cast                  10719
homepage               2896
director               5067
tagline                7997
keywords               8804
overview              10847
runtime                 247
genres                 2039
production_companies   7445
release_date           5909
vote_count             1289
vote_average             72
release_year             56
budget_adj             2614
revenue_adj            4840
dtype: int64
```

Most of the categorical values have too many distinct values thus can be analysed individually

```
#checking for null values in the dataset
df.isnull().sum()
```

```
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
```

```
#Correlation between features check
df.corr()
```

	id	popularity	budget	revenu
id	1.000000	-0.014350	-0.141351	-0.09922
popularity	-0.014350	1.000000	0.545472	0.66335
budget	-0.141351	0.545472	1.000000	0.73490

The id column has a negative correlation with all features therefore it can be dropped

- budget has a high correlation with revenue
- revenue has a high correlation with budget, popularity,vot_count, revenue_adj and budget_adj
- runtime does not have a high correlation with any of the other features
  popularity

Data Cleaning

```
#converting the release_date column from object type to date
df['release_date'] = pd.to_datetime(df['release_date'])
```

```
#dropping some columns
df=df.drop(['imdb_id','id','original_title','homepage','tagl
df.head()
```

	popularity	budget	revenue	cast
0	32.985763	150000000	1513528810	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...
1	28.419936	150000000	378436354	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...
2	13.112507	110000000	295238201	Shailene Woodley|Theo James|Kate Winslet|Ansel...
3	11.173104	200000000	2068178225	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...
4	9.335014	190000000	1506249360	Vin Diesel|Paul Walker|Jason Statham|Michelle

```
#Removing the pipe symbol on the genres column
genres_conc = df['genres'].str.cat(sep='|')
genres_conc
```

```
'Action|Adventure|Science Fiction|Thriller|Action|Adve
nture|Science Fiction|Thriller|Adventure|Science Ficti
on|Thriller|Action|Adventure|Science Fiction|Fantasy|A
ction|Crime|Thriller|Western|Drama|Adventure|Thriller|
Science Fiction|Action|Thriller|Adventure|Drama|Advent
ure|Science Fiction|Family|Animation|Adventure|Comedy|
Comedy|Animation|Family|Action|Adventure|Crime|Science
Fiction|Fantasy|Action|Adventure|Drama|Science Fiction
```

```
genres =pd.Series(genres_conc.split(sep='|'))
```

```
genres
```

```
0                 Action
1              Adventure
2        Science Fiction
3               Thriller
4                 Action
                  ...
26955            Mystery
26956             Comedy
26957             Action
26958             Comedy
26959             Horror
Length: 26960, dtype: object
```

```
#removing the pipe(|) symbol on the production companies
def Production_Companies(x):
          #concatenate production companies into one string
    companies_conc =df['production_companies'].str.cat(sep='

          #breakdown the production companies string into si
    companies =pd.Series(companies_conc.split('|'))
    #companies=companies.value_counts(ascending=False)
    return(companies)


x= Production_Companies(df)
x
```

```
0                  Universal Studios
1               Amblin Entertainment
2                 Legendary Pictures
3            Fuji Television Network
4                            Dentsu
                     ...
23222               Joel Productions
23223    Douglas & Lewis Productions
23224                        Mosfilm
23225        Benedict Pictures Corp.
```

```
23226                       Norm-Iris
Length: 23227, dtype: object
```

```
#removing the pipe(|) symbol from the directors column
def Directors(x):
          #concatenate movie directors into one string
    directors =df['director'].str.cat(sep='|')

          #breakdown the directors string into single rows
    movie_directors =pd.Series(directors.split('|'))
    #movie_directors =movie_directors.value_counts(ascending
    return(movie_directors)
```

```
Directors(df)
```

```
0           Colin Trevorrow
1             George Miller
2          Robert Schwentke
3               J.J. Abrams
4                 James Wan
                  ...
11887           Bruce Brown
11888    John Frankenheimer
11889        Eldar Ryazanov
11890           Woody Allen
11891      Harold P. Warren
Length: 11892, dtype: object
```

```
#removing the pipe(|) symbol from the cast column

def Movie_Cast(x):
          #concatenate movie directors into one string
    cast =df['cast'].str.cat(sep='|')

          #breakdown the directors string into single rows
    movie_cast =pd.Series(cast.split('|'))
```

```
    #movie_cast=movie_cast.value_counts(ascending=False)
    return(movie_cast)
```

```
Movie_Cast(df)
```

```
0                Chris Pratt
1        Bryce Dallas Howard
2                Irrfan Khan
3          Vincent D'Onofrio
4              Nick Robinson
                   ...
52568       Harold P. Warren
52569             Tom Neyman
52570          John Reynolds
52571           Diane Mahree
52572      Stephanie Nielson
Length: 52573, dtype: object
```

Exploratory Data Analysis

Exploratory Data Analysis

Research Question 1 What is the average movie runtime?

```
#statistical information about the dataset
df.describe()
```

	popularity	budget	revenue	r
count	10866.000000	1.086600e+04	1.086600e+04	10866.
mean	0.646441	1.462570e+07	3.982332e+07	102.
std	1.000185	3.091321e+07	1.170035e+08	31.
min	0.000065	0.000000e+00	0.000000e+00	0.
25%	0.207583	0.000000e+00	0.000000e+00	90.
50%	0.383856	0.000000e+00	0.000000e+00	99.

Answer: The average movie runtime is 102

max	32.985763	4.250000e+08	2.781506e+09	900.

From the above stastical information the following can be identified about the dataset:

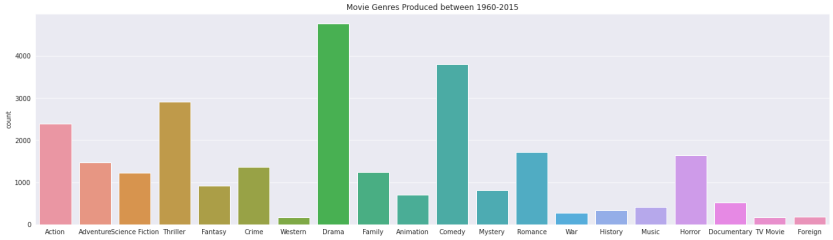
1. The movies with the highest budget was 425,000,000
2. The movie with the highest revenue was 2,781,506,000
3. The latest movie release year is 2015 and the earliest movie release was in 1960
4. movie with the longest runtime was 900
5. The movie with the highest rating had a 9.2 rating, the lowest rating was a 1.5 and the average movie rating was a 5.5
6. The movie with the highest vote count had 9767 votes while the lowest had at least 10 votes.
7. The most popular movie had a 32 and the least popular movie had at least a 0

Research Question 2: Which genre of movies are the most produced?

```
#Get the genres movie counts
Genre_conc =df['genres'].str.cat(sep='|')
Genres =pd.Series(Genre_conc.split('|'))
Genres.value_counts(ascending=False)
```

```
Drama              4761
Comedy             3793
Thriller           2908
Action             2385
Romance            1712
Horror             1637
Adventure          1471
Crime              1355
Family             1231
Science Fiction    1230
Fantasy             916
Mystery             810
Animation           699
Documentary         520
Music               408
History             334
War                 270
Foreign             188
TV Movie            167
Western             165
dtype: int64
```

```
#countplot for movie genres

fig, ax = plt.subplots(figsize=(22, 6))
sns.countplot(x=Genres,ax=ax).set(title='Movie Genres Produc
plt.show()
```



Answer: Dramas were the most produced genres while TV movies,Foreign and Western genres had the lowest production

---

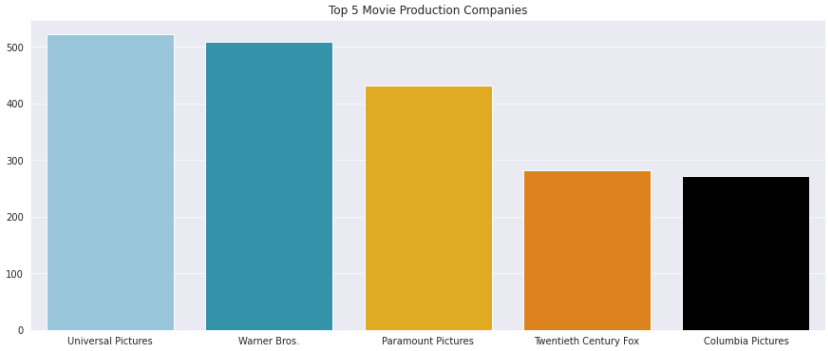
Research Question 3:Which production companies produced the most movies?

```
#get the indivudual production companies and movies produced
def Production_Companies_count(x):
        #concatenate production companies into one string
    companies_conc =df['production_companies'].str.cat(sep='

        #breakdown the production companies string into si
```

```
    companies =pd.Series(companies_conc.split('|'))
    companies=companies.value_counts(ascending=False)
    return(companies)


#count of movies produced by individual production companies
prod_companies = Production_Companies_count(df)
prod_companies
```

```
Universal Pictures                          522
Warner Bros.                                509
Paramount Pictures                          431
Twentieth Century Fox Film Corporation      282
Columbia Pictures                           272
                                           ...
Monophonic Inc.                               1
Populist Pictures                             1
Qatsi Productions                             1
CineEvelyn                                    1
Norm-Iris                                     1
Length: 7879, dtype: int64
```

```
# x-axis plot
x_values = ["Universal Pictures","Warner Bros.","Paramount P
x_axis = prod_companies[:5]

#y-axis plot
y=x_axis.values

#bars color pallette
colors=("#8ecae6","#219ebc","#ffb703","#fb8500","#000000")

#bar plot
sns.set_style("darkgrid")
plt.figure(figsize=(15,6))
sns.barplot(x=x_values, y=y,palette= colors).set(title='Top
plt.show()
```



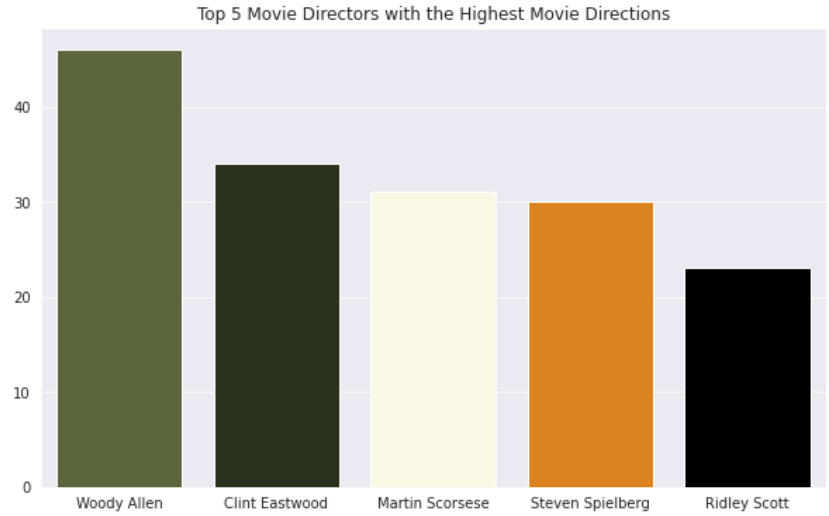
Answer: **Universal pictures** had the highest movie productions with a count of 522 movies followed by Warner Bros. with 509

Research Question 4: Which Director directed the most movies?

```
#get the individual movies directors have directed
def directors_count(x):
          #concatenate movie directors into one string
    directors =df['director'].str.cat(sep='|')

          #breakdown the directors string into single rows
```

```
    movie_directors =pd.Series(directors.split('|'))
    #get the count of movie directions by a director
    movie_directors =movie_directors.value_counts(ascending=

    return(movie_directors)


#count of movies directed by each director
dir_count= directors_count(df)
dir_count

     Woody Allen         46
     Clint Eastwood      34
     Martin Scorsese     31
     Steven Spielberg    30
     Ridley Scott        23
                         ..
     Mike Maguire         1
     Tom Kuntz            1
     John Simpson         1
     Simon Hunter         1
     Harold P. Warren     1
     Length: 5362, dtype: int64


# Graph Representation of the top 5 movie directors
# x-axix plot
x_values = ["Woody Allen","Clint Eastwood","Martin Scorsese"
x_axis = dir_count[:5]

#y-axis plot
y=x_axis.values

#bar color pallette
colors=("#606c38","#283618","#fefae0","#fb8500","#000000")

#bar plot
sns.set_style("darkgrid")
plt.figure(figsize=(10,6))
```

```
sns.barplot(x=x_values, y=y,palette= colors).set(title='Top
plt.show()
```



Answer: **Woody Allen Directed most movies with a count of 46 movies**

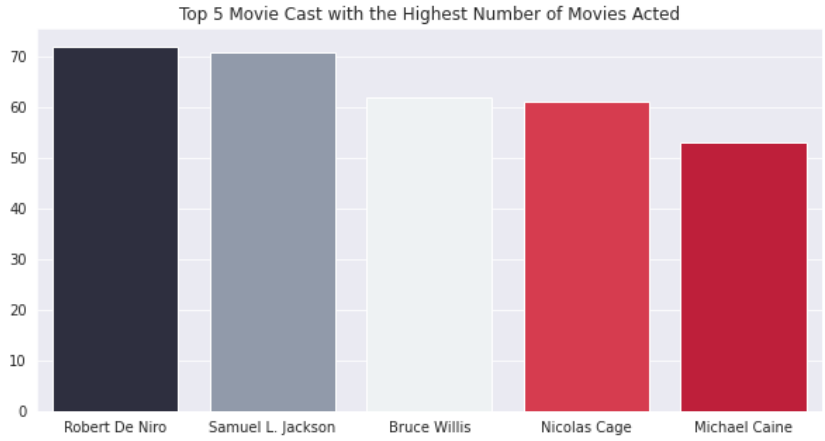
Research Question 5: Who was the most Casted Actor?

```
#get the number of times an actor was casted
```

```
def Cast_count(x):

    cast =df['cast'].str.cat(sep='|')
    movie_cast =pd.Series(cast.split('|'))
    movie_cast=movie_cast.value_counts(ascending=False)
    return(movie_cast)



actors = Cast_count(df)
actors
```

```
Robert De Niro       72
Samuel L. Jackson    71
Bruce Willis         62
Nicolas Cage         61
Michael Caine        53
                     ..
Andy Milonakis        1
Samantha Cope         1
Cynthia Watros        1
Satya Bhabha          1
Stephanie Nielson     1
Length: 19026, dtype: int64
```

```
#x and y values for the graph
x_values = ["Robert De Niro","Samuel L. Jackson","Bruce Will
x_axis = actors[:5]

y=x_axis.values
#colors  for the bars
colors=("#2b2d42","#8d99ae","#edf2f4","#ef233c","#d90429")

#plot
sns.set_style("darkgrid")
plt.figure(figsize=(10,5))
sns.barplot(x=x_values, y=y,palette= colors).set(title='Top
plt.show()
```



Answer: **Robert De Niro** was the most casted actor follwed closely by Samuel L.Jackson both with over 70 movies

---

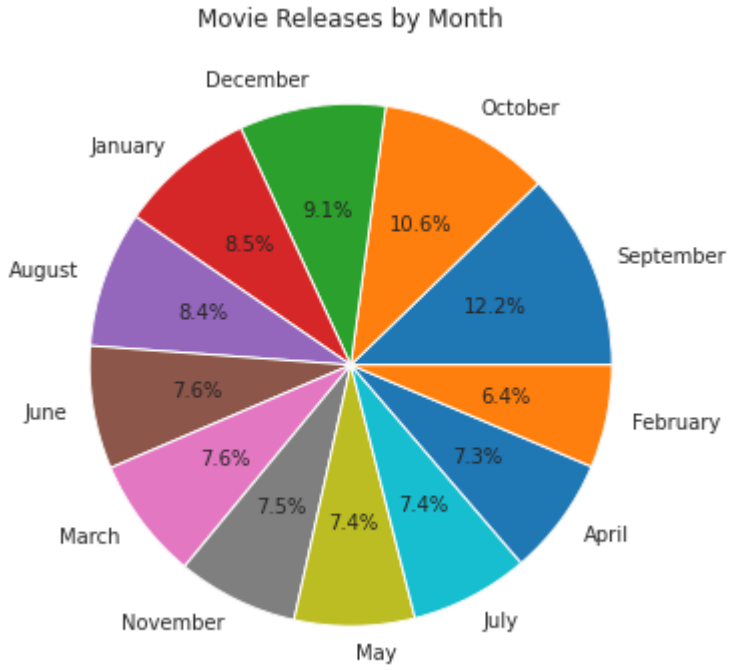
Research Question 6: Which month had the highest movie releases?

```
#get month name from the date column and count movies releas
print(df['release_date'].dt.month_name().value_counts(ascend
```

```
September    1331
October      1153
December      985
January       919
August        918
June          827
```

```
March          823
November       814
May            809
July           799
April          797
February       691
Name: release_date, dtype: int64
```

```
monthly_percentage = (df['release_date'].dt.month_name().val
#explode = (0.1,0.0)

_, ax = plt.subplots(figsize=(10,6))
ax = monthly_percentage \
.plot(kind='pie', autopct='%.1f%%')
ax.set_title("Movie Releases by Month")
ax.set_ylabel('')
plt.show()
```



Answer: Most movies were released in the month of September

```
#drop categorical values to reduce the dataset
df1=df.drop(['cast','director','release_date','genres','prod
df1.head()
```

	popularity	budget	revenue	runtime	vote_cc
0	32.985763	150000000	1513528810	124	5
1	28.419936	150000000	378436354	120	6
2	13.112507	110000000	295238201	119	2
3	11.173104	200000000	2068178225	136	5
4	9.335014	190000000	1506249360	137	2

```
#check for null values
df1.isnull().sum()
```

```
popularity      0
budget          0
revenue         0
runtime         0
vote_count      0
vote_average    0
release_year    0
budget_adj      0
revenue_adj     0
dtype: int64
```

Research Question 7: Does a Movies' Popularity affects its revenue?

```
#correlation between features
plt.figure(figsize = (10,10))
sns.heatmap(df1.corr(), annot =True)
```

```
<matplotlib.axes._subplots.AxesSubplot at
0x7f7240792710>
```

Answer: Yes, since there is a high correlation between revenue and popularity and also the vote_count

---

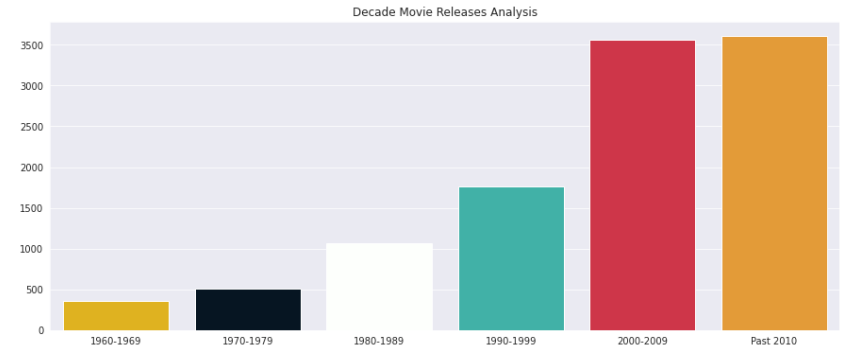
Other Features correlation observation

- Popularity has a high corrrelationwith vote_count, revenue
- budget: has a high corrrelation with budget_adj and revenue
- runtime has a low correlation with most of the features
- vote_count has a high correlation with revenue_adj,revenue and popularity
- vote_average has a low correlation with all other features
- release_year has a low correlation with all other features
- budget_adj has a high corelation with budget_adj,revenue_adj,budget,revenue
- revenue_adj has a high corelation with budget_adj,vote_count,budget,revenue

Research Question 8: Which decade had the most movie releases?

```
#get the decades
the_sixties = df1.release_year[(df1.release_year>= 1960) & (
the_seventies = df1.release_year[(df1.release_year>= 1970) &
the_eighties = df1.release_year[(df1.release_year>= 1980) &
```

```
the_nineties = df1.release_year[(df1.release_year>= 1990) &
two_thousands = df1.release_year[(df1.release_year >=2000) &
past_2010 = df.release_year[df.release_year >= 2010]

#set the x and y values
x_values= ["1960-1969","1970-1979","1980-1989","1990-1999","
y = [the_sixties.count(),the_seventies.count(),len(the_eight
colors=("#ffc300","#011627","#fdfffc","#2ec4b6","#e71d36","#

#plot the graph
sns.set_style("darkgrid")
plt.figure(figsize=(15,6))
sns.barplot(x=x_values, y=y,palette= colors).set(title='Deca
plt.show()
```

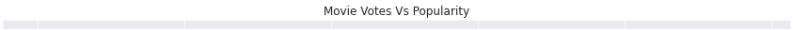


Answer:The two thousand decade had the most movie releases
observation: Movie releases increased by decade and also increased rapidly by the 2000s decade

---

Research Question 9: Did most movies have higher ratings and votes?

```
#votes and popularity  comparison
plt.figure(figsize= (15,6))
sns.scatterplot(x=df1['vote_count'],y=df1['popularity']).set
plt.show()
```



Answer : Most movies got lower votes and thus lower popularity/ratings

Observation: Movies with lower vote count were less popular compared to movies with high vote count

---

Research Question 10:What was the highest profits made?

```
# create the profits column that shows the movie profits
df1=df1.assign(Profits=lambda x: x.revenue - x.budget)
df1.head()
```

	popularity	budget	revenue	runtime	vote_cc
0	32.985763	150000000	1513528810	124	5
1	28.419936	150000000	378436354	120	6
2	13.112507	110000000	295238201	119	2
3	11.173104	200000000	2068178225	136	5
4	9.335014	190000000	1506249360	137	2

```
df1.describe()
```

	popularity	budget	revenue	r
count	10866.000000	1.086600e+04	1.086600e+04	10866.
mean	0.646441	1.462570e+07	3.982332e+07	102.
std	1.000185	3.091321e+07	1.170035e+08	31.
min	0.000065	0.000000e+00	0.000000e+00	0.
25%	0.207583	0.000000e+00	0.000000e+00	90.

Answer: Max profits made on movies production was $2,544,506,000

max	[illegible]	[illegible]	[illegible]	[illegible]

Conclusions

In conclusion, from the analysis the following was determined:

- Most movies were produced in the Two thousands while there was lesser releases in the sixties and seventies
- Universal Pictures production company produced most movies
- Movies with less vote count were less popular compared to movies with higher votes
- A movie's runtime does not affect its popularity or revenue
- The month of September had the most movie releases and there was no month without a movie release
- Dramas were the most popular genres

*challenges encountered with exploration**

- unable to determine if the movie genre, directors, production company or cast determined a movie's popularity or revenue generation because of how the data was represented