Sygnific

Project - 2

Uncovering suprising Facts from World Wide Movie Database using Data Cleaning & Data Visualization

OVERVIEW

A project to overlook at the movie's database and interpret various finding using Data cleaning, Data wrangling and Data Visualization

Software Requirements

1. Programming Language : Python
2. Environemnt: Jupyter Notebooks / Google Collab
3. Database: CSV(export type)
4. Operation System: Windows XP or above
5. Librarires Used: Pandas,Folium, Seaborn, Scikit, SKLEARN, Wordcount

6.Datasets used: TMDB Dataset

1. Open a New Notebook and import the required libraires and read the csv file

```
import numpy as np
import pandas as pd
pd.set_option('max_columns', None)
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
plt.style.use('ggplot')
import datetime
from scipy import stats
from wordcloud import WordCloud
from collections import Counter
from nltk.corpus import stopwords
from nltk.util import ngrams
import nltk
nltk.download('stopwords')
stop = set(stopwords.words('english'))
import os
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
from PIL import Image
```

Description:Here we are importing all the required libraries like numpy, pandas, matplotlib, seaborn, datetime, wordcloud etc. Pandas is used for data manipulation and analysis. We are using NumPy library for scientific computing applications. matplotlib, seaborn and wordcloud for data visualization.

2. Loading the training & testing Dataset

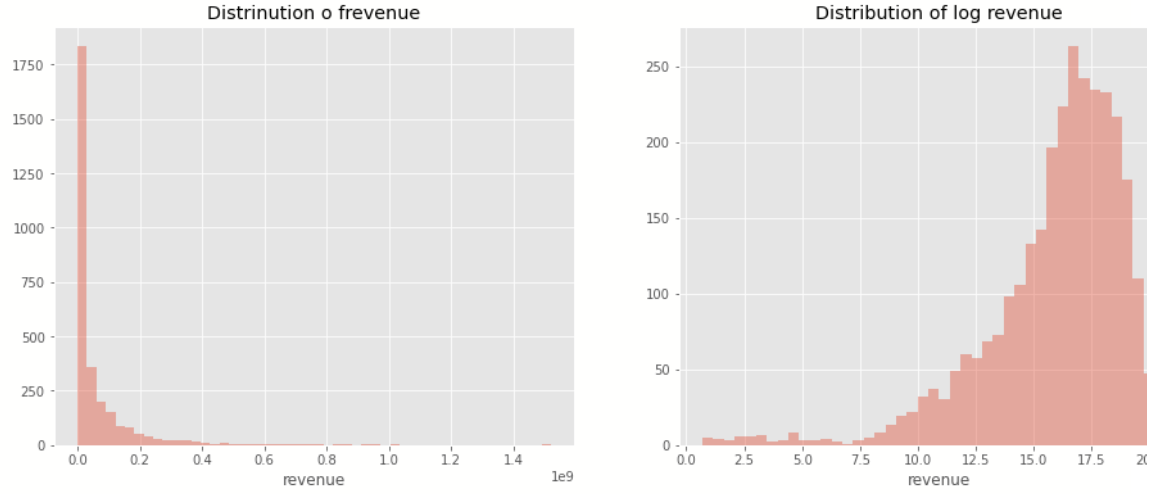
```
train = pd.read_csv('/data.csv')
```

Description:We are loading all the data from data.csv file to variable train using pandas.

3. Visualizing the Distribution of Revenue with & without Log

```
fig, ax = plt.subplots(figsize=(16,6))
plt.subplot(1, 2, 1)
sns.distplot(train['revenue'], kde=False);
plt.title('Distrinution o frevenue');
plt.subplot(1, 2, 2)
sns.distplot(np.log1p(train['revenue']), kde=False);
plt.title('Distribution of log revenue')
```

Output:



Description:Here are ploting a distplot of revenue for all the movies using seaborn library. If we observe this distplot we find that there are more than 1750 movies with revenue of zero. This is technically not posible. This distplot is wrong because we have huge data and we may not be able to plot it directly. Now we use numpy library's method .log1p(x). This will calculate the natural logarithmic value of x+1 where x belongs to all the input array elements. Now if we plot a distplot we get a proper graph.

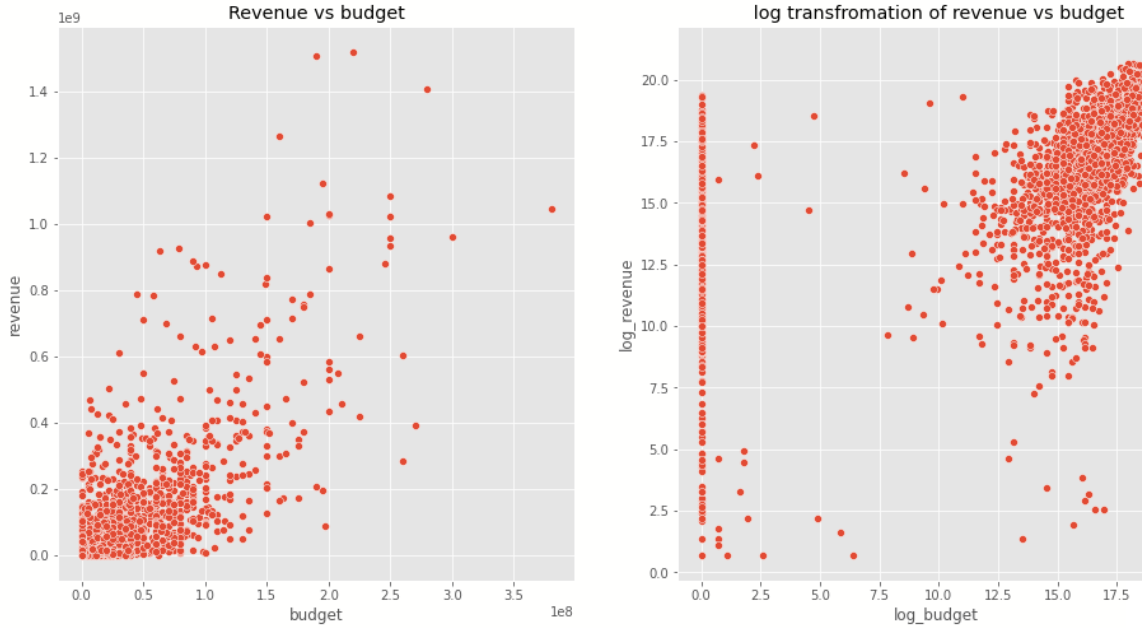
4. Finding the Relationship between Movie Revenue & Budget

```
train['log_revenue'] = np.log1p(train['revenue'])
train['log_budget'] = np.log1p(train['budget'])

plt.figure(figsize=(16, 8))
plt.subplot(1, 2, 1)
sns.scatterplot(train['budget'], train['revenue'])
plt.title('Revenue vs budget');
plt.subplot(1, 2, 2)
sns.scatterplot(train['log_budget'], train['log_revenue'])
plt.title('log transfromation of revenue vs budget');
```

Description: Now we are trying to find the relation between Movie revenue and Budget by ploting a scatterplot. Fig 1 shows the scatter plot of revenue and Budget of different movies. If we observe this plot we find that this plot is wrong as it is interpreting data ina wrong way. If we consider the log transformation of both revenue and Budget and then plot a scatterplot we get a plot which is more meaningful. If we observe this second plot we still find some point that show movies with zero Budget have huge revenue. This is because of anomalies. In general, we either use some methods and remove these anamolies or just ignore them.

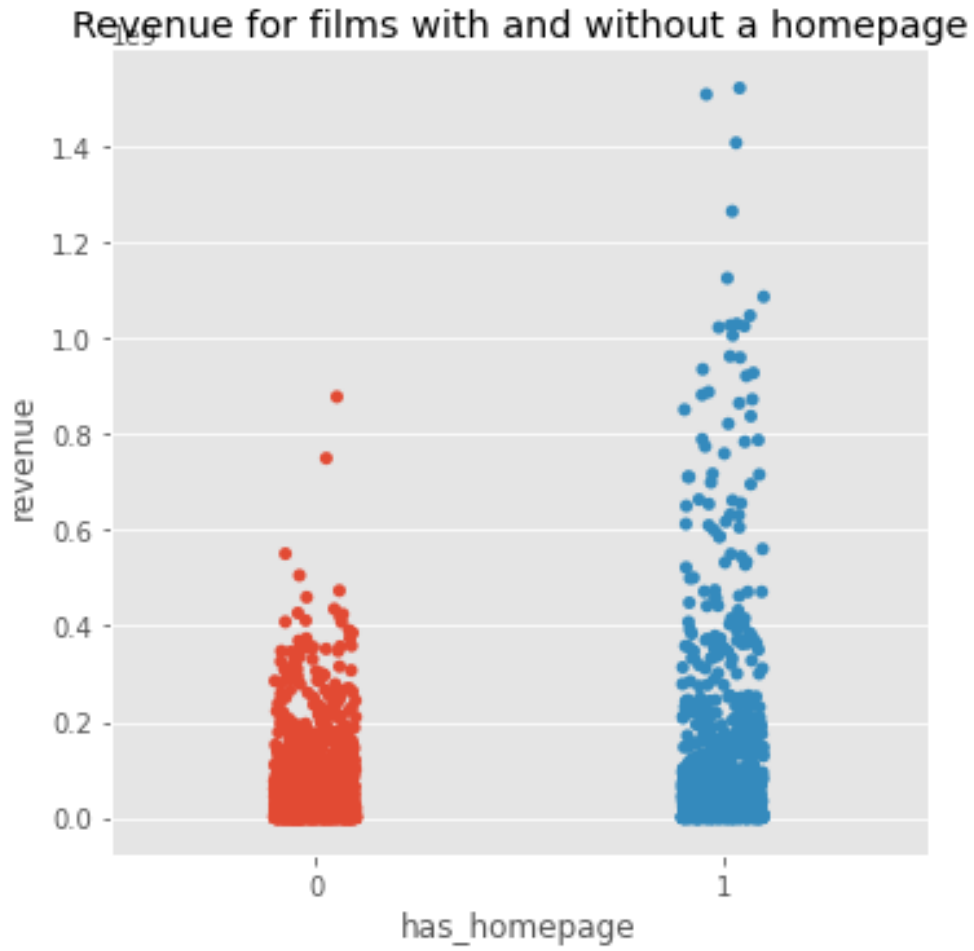
Output:



5. Impact of Film's Revenue with or without Homepage

```
train['has_homepage'] = 0
train.loc[train['homepage'].isnull() == False, 'has_homepage'] = 1
sns.catplot(x='has_homepage', y='revenue', data=train);
plt.title('Revenue for films with and without a homepage');
```

Output:



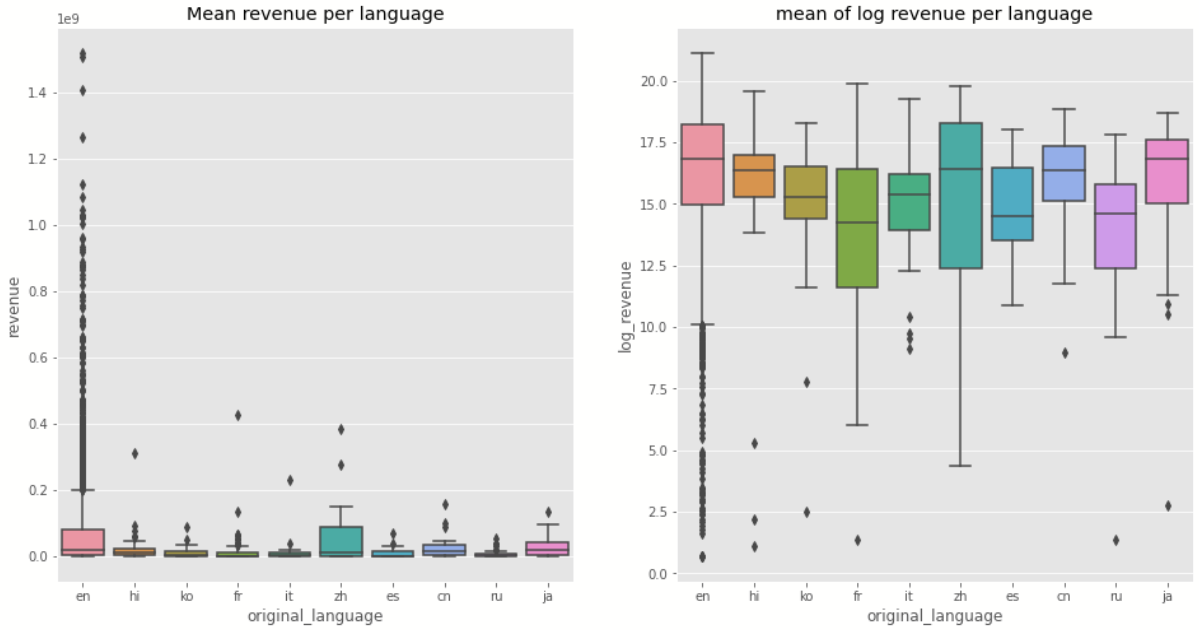
Description: Here we are trying to find out what is impact of having a homepage to a movie. First we divide the all movies into two categories namely movies with homepage and without homepage. Movies with homepage are marked with 1 and without homepage are marked 0 in has_homepage column of train dataframe. Now we are ploting a catplot for has_homepage column. If we observe the plot we find that very few movie that have a homepage has some effect on revenue. But it is not enough to say that homepage of a movie will impact its revenue.

6. Films Revenue in various Languages

```
language_data = train.loc[train['original_language'].isin(train['original_language'].value_counts().head(10).index)]

plt.figure(figsize=(16,8))
plt.subplot(1, 2, 1)
sns.boxplot(x='original_language', y = 'revenue', data=language_data )
plt.title('Mean revenue per language')
plt.subplot(1, 2, 2)
sns.boxplot(x='original_language', y = 'log_revenue', data=language_data)
plt.title('mean of log revenue per language')
```

Output:



Description: Here we are ploting a box plot for original_language and revenue. We are using log transformation of revenue for better visulization. After ploting we find that english language movie has got highest revenue but when we consider as a whole zurish language movies have generated more revenue tan any other language.

7. Frequent Words in Movie Titles

```
plt.figure(figsize=(12, 12))
text =  ' '.join(train['original_title'].values)
wordcloud = WordCloud(max_font_size=None,
                      background_color ='white',
                      width =1200, height =1000).generate(text)
plt.imshow(wordcloud)
plt.title('Top word across movie titles')
plt.axis('off')
plt.show()
```

Description: Wordcloud helps us to find out the most freaquent words. We are using the wordcloud to find out the most frequent words used in movie titles. First we are joining all the words in original_title column by using .join method and then we are creating a wordcloud using this text. Plt.imshow() is used to display the Word cloud.
Here we find that words like man, last, love, day etc are most frequently used words in movie titles.

Output:

8. **Frequent Words in Movie Overviews**

```
plt.figure(figsize=(12, 12))
text =  ' '.join(train['overview'].fillna('').values)
wordcloud = WordCloud(max_font_size=None,
                     background_color ='white',
                     width =1200, height =1000).generate(text)
plt.imshow(wordcloud)
plt.title('Top word across movie overviews')
plt.axis('off')
plt.show()
```

Description: we use the same wordcloud to find the frequent words in movie overviews. When we generate the Word cloud of movie overviews we find that words like life, find, one, family, world, take etc are used frequently.

Output: