Mini Project -3

Box office Data Analysis &Interpretation

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 scipy.sparse import hstack, csr_matrix
from sklearn.model_selection import train_test_split, KFold
from wordcloud import WordCloud
from collections import Counter
from nltk.corpus import stopwords
from nltk.util import ngrams
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.preprocessing import StandardScaler
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
import json
import ast
from urllib.request import urlopen
from PIL import Image
```

Firstly we import all the required libraries. Seaborn for data visualization, Matplotlib for data plotting ,Numpy for complex computing and dealing with multidimensional arrays, Pandas for data manipulation, Datetime module for dealing with dates and time, Scipy(scientific python) for complex data computation and it has modules like linear algebra, integration and some special functions. Sklearn(sci-kit learn) it has algorithams like classsification, regression and clustering. Nltk(natural language toolkit) has libraries and toolkits for symbolic and statistical natural language processing.

2. Loading the training & testing Dataset

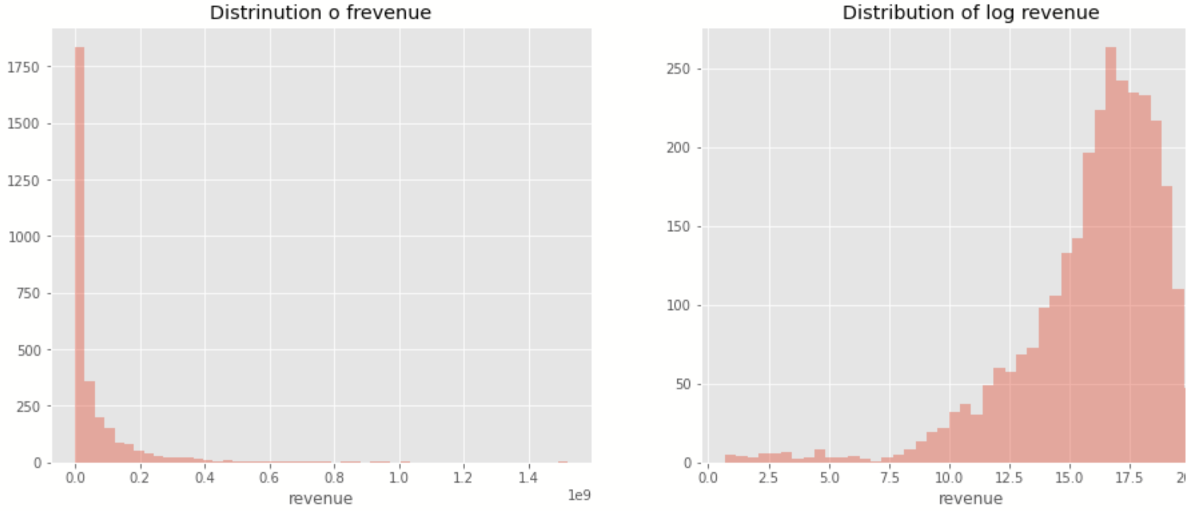
```
train = pd.read_csv('/train.csv')
test = pd.read_csv('/test.csv')
```

Here we load the training and testing data.

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:



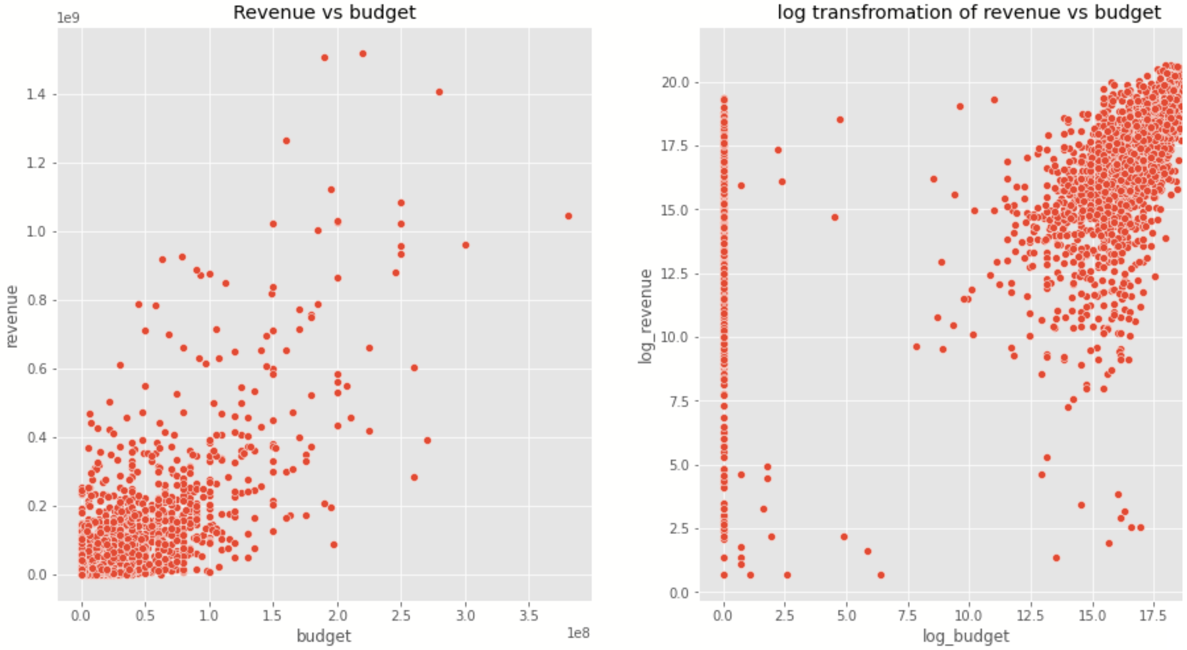
Here we plot the data using subplots and displots. Subplots for plotting multiple plots at one place. Distplot shows histogram and it combines the matplotlib hist with seaborn kdeplot and regplot funtions. title function displays the title over image.

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');
```

Output:

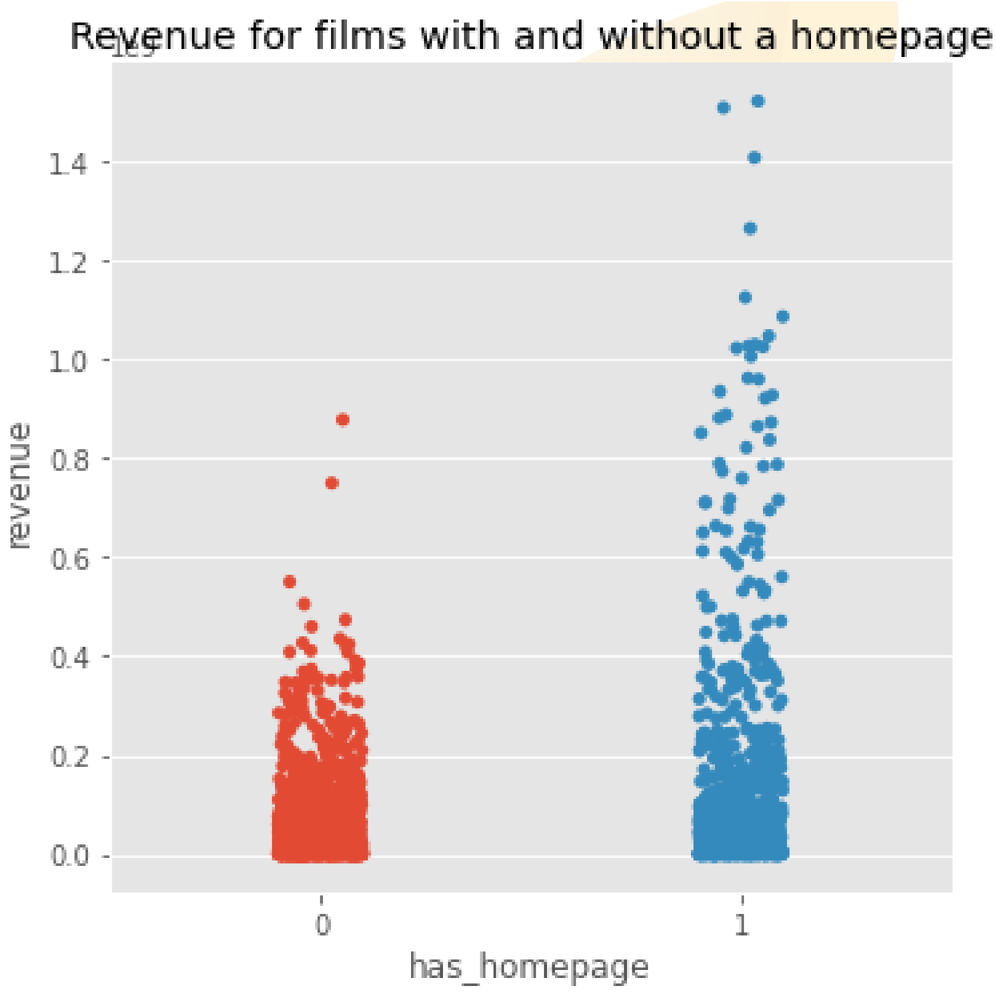


We use plt.figure(figsize) for giving dimensions. Scatterplot displays the data in scattered form. scatterplot(train['budget'], train['revenue']) plots the graph against two values

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:
```



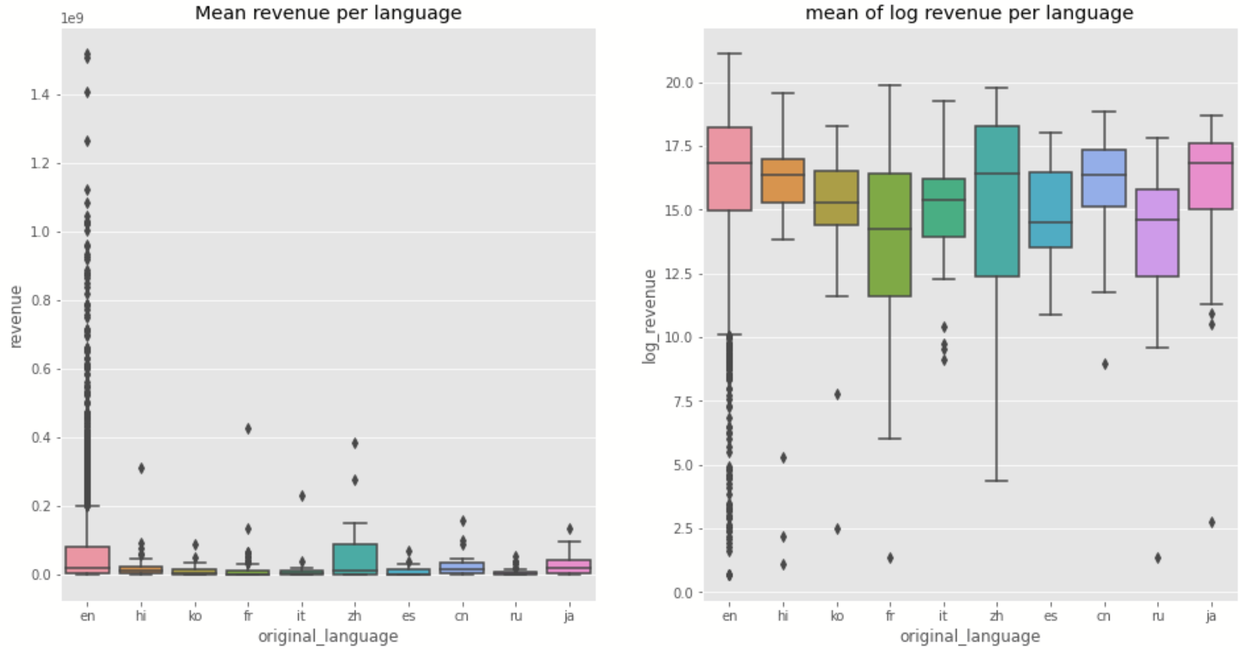
Firstly we assign the has_hompage value to zero. loc method takes only index labels and returns value if label exsits. So here we check if the value is zero and return true or false. Catplot returns the frequency of the categories and here we assign the x and y axis values to it. Title function for assigning title to it.

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:

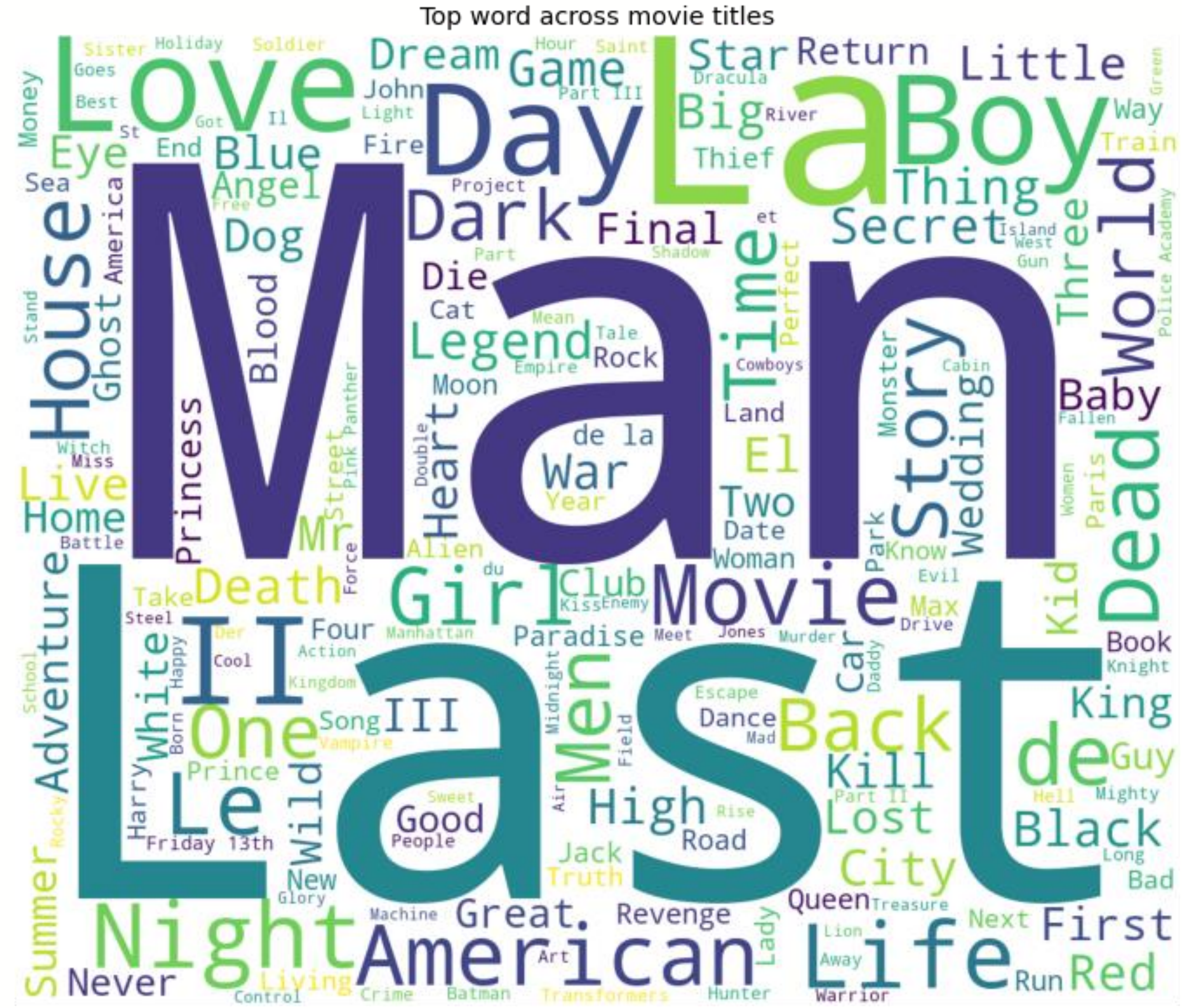


As mentioned above figsize is used to assign dimensions to figure. Subplot for assigning space for the graph plots. Boxplot plots shows the distribution of numerical data and skewness through displaying the data quartiles and averages.

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()
```

Output



Top word across movie titles

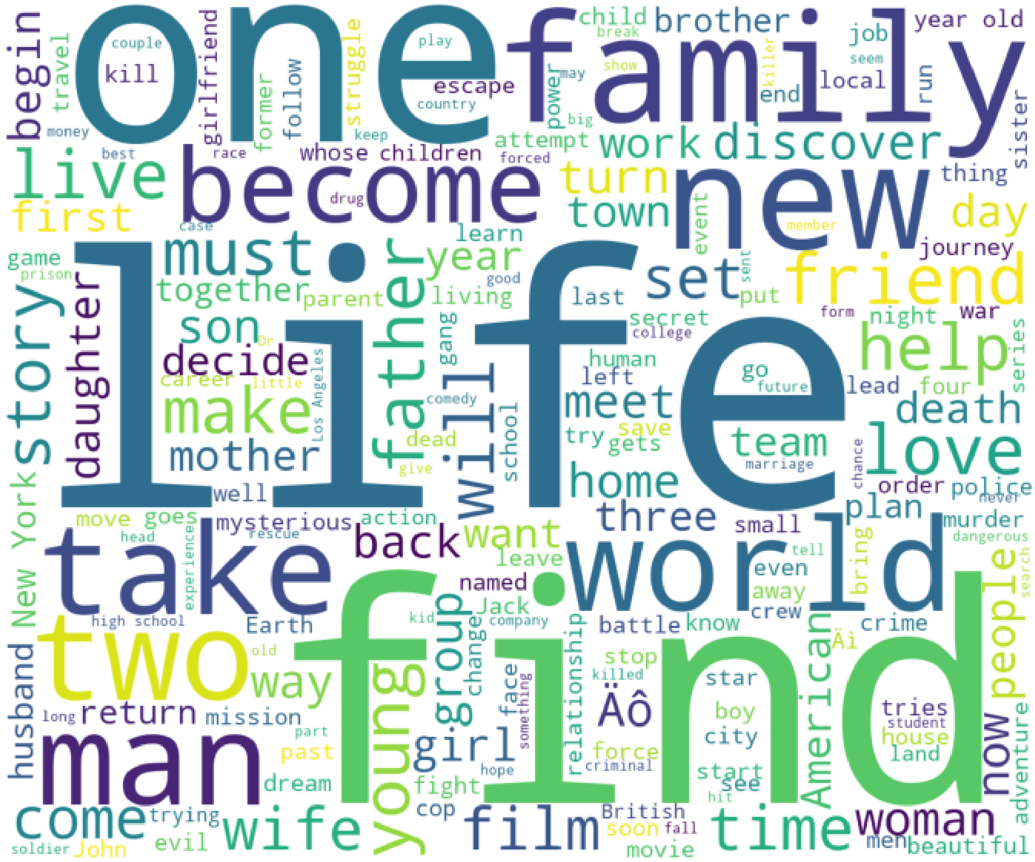
Here we are making a word cloud, which means it clusters all the words into one single cloud format. We give all the attributes like size and text format and print the text. imshow displays the grey scale image in figure

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()
```

Output:

Top word across movie overviews



Conclusion:

We have interpreted and analyzed data of movies.