Homework 3 - TMDB Box Office Prediction

For all parts below, answer all parts as shown in the Google document for Homework 3. Be sure to include both code that justifies your answer as well as text to answer the questions. Show runtime results for each cell. We also ask that code be commented to make it easier to follow.

Part 1 - Data Cleaning and Reformatting

In [73]:

```
#task 1
#import jieba
from collections import Counter
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
#display all outputs
import pandas as pd
import numpy as np
#pd.set_option(‘display.max_rows’, None)
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
from os import path
import requests
from io import StringIO
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from ast import literal_eval
train = pd.read_csv('C:/Users/user/Desktop/CSE519/hw3/hw3train.csv')
test = pd.read_csv('C:/Users/user/Desktop/CSE519/hw3/hw3test.csv')
sample_submission = pd.read_csv('C:/Users/user/Desktop/CSE519/hw3/sample_submission.csv')
pd.options.display.max_rows=120
pd.options.display.max_rows
#train.genres[470].replace(np.NaN, '[]',inplace=True)
train.genres[470]="[{'id': 12, 'name': 'Adventure'}, {'id': 18, 'name': 'Drama'}]"#replace Nan with...
train.genres[1622]="[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'name': 'Drama'}]"#replace Nan with...
train.genres[1814]="[{'id': 35, 'name': 'Comedy'}]"#replace Nan with...
train.genres[1819]="[{'id': 10749, 'name': 'Romance'}, {'id': 18, 'name': 'Drama'}]"
train.genres[2423]="[{'id': 28, 'name': 'Action'}, {'id': 10749, 'name': 'Romance'}]"
train.genres[2686]="[{'id': 53, 'name': 'Thriller'}]"
train.genres[2900]="[{'id': 18, 'name': 'Drama'}]"
train.overview[390]="Paolo needs to reach the castle of Alberto Caccia, where he is invited to spend Christmas holidays with his wife Margherita, at the ninth month of pregnancy, and her family. Among various mishaps and blunders, Paolo will come to make everyone believe, because of a misunderstanding, that Alberto is dead because of his fault."
#train.runtime.describe()
#train.runtime.hist(bins=50)
#train[train.id==391]['runtime']=86
train.runtime[390]=86
train.runtime[591]=90
train.runtime[924]=86
train.runtime[977]=93
train.runtime[1255]=91
train.runtime[1541]=93
train.runtime[1874]=86
train.runtime[2150]=108
train.runtime[2498]=86
train.runtime[2645]=98
train.runtime[2785]=111
train.runtime[2865]=96
train.hist()
plt.tight_layout()
plt.show()
pd.set_option('display.max_rows', None)
train.revenue[312]=12009070
train.revenue[15]=273683
#(198 on train date makes no sense,
#171052 pound on wiki is equal to 171052*1.6=273683 us dollars)
train.budget[15]=800000
# the budget on train data is 500000 in English Pounds. I convert it to 500000x1.6=800000(2011 currency rate)
train.revenue[450]=12000000
train.revenue[1281]=46789413
train.revenue[280]=10200000
train.budget[280]=5250000
# row 280 bats budget and revenue based on this link below
#https://en.wikipedia.org/wiki/Bats_(film)
train.revenue[1541]=3514780
#link for 3514780:  https://www.imdb.com/title/tt3805180/
train.revenue[1884]=23700000
#link for $23700000   https://en.wikipedia.org/wiki/In_the_Cut_(film)
train.revenue[2490]=6858261
train.loc[1200]
#train.budget.corr(train.revenue)
train.budget.hist(bins=100)
train['belongs_to_collection3']=[list() for x in range(len(train['belongs_to_collection']))]
train['belongs_to_collection4']=[list() for x in range(len(train['belongs_to_collection']))]
train['belongs_to_collection1']=train.belongs_to_collection.replace(np.NaN, '[]',inplace=False)
train['belongs_to_collection2']=train['belongs_to_collection1'].apply(literal_eval)
train['belongs_to_collection3']=train['belongs_to_collection2'].apply(lambda x:x[0]['name'] if x != [] else 0)
train['belongs_to_collection4']=train['belongs_to_collection2'].apply(lambda x:1 if x != [] else 0)
list6=Counter(train['original_language'])
language6=dict(list6)
train['original_language1']=[list() for x in range(len(train['original_language']))]
for i in range(len(train['original_language'])):
    train['original_language1'][i]=int(language6[train['original_language'][i]])#train.belongs_to_collection4
#need to delete this row below because the budget is very high and the revenue is too low only $100
train['original_language2']=pd.to_numeric(train['original_language1'])
train.drop(train.index[2864],axis=0,inplace=True)
#$100 show here : https://www.themoviedb.org/movie/42481-die-angst-des-tormanns-beim-elfmeter?language=en-US
#need to drop this row below because the budget is 2000000, but the revenue is in deed 30$
train.drop(train.index[2090],axis=0,inplace=True)#see #train.drop([1006])#see https://en.wikipedia.org/wiki/Zyzzyx_Road
#row 2090, Deadfall, has high budget and very low revenue,so I need to delete it.
train.drop(train.index[1006],axis=0,inplace=True)#see https://en.wikipedia.org/wiki/Zyzzyx_Road
train.reset_index(drop=True,inplace=True)
for i in range(0,len(train.budget)):
    if train.budget[i]<1000:
        train.budget[i]=8000000
```

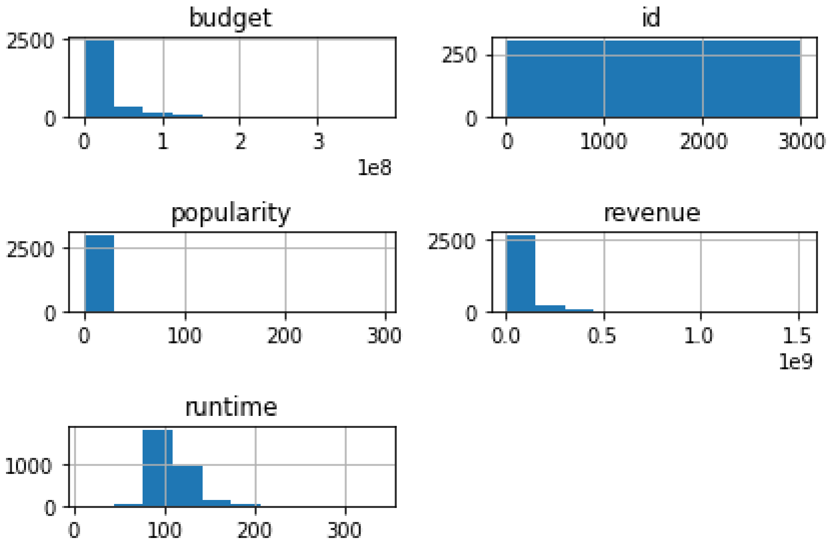
```
Out[73]:

120
```

```
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:24: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:25: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:26: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:27: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:28: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:29: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:30: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:31: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:35: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:36: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:37: Sett
```

```
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:38: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:39: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:40: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:41: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:42: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:43: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:44: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:45: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:46: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[73]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x00000218EA55E
688>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000218DF1ED
D48>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x00000218DF1D5
E88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000218DF0F3
0C8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x00000218FEF22
DC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000218DF130
8C8>]],
      dtype=object)
```



```
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:51: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:52: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:55: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:57: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:58: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:59: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:60: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:63: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:65: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:67: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

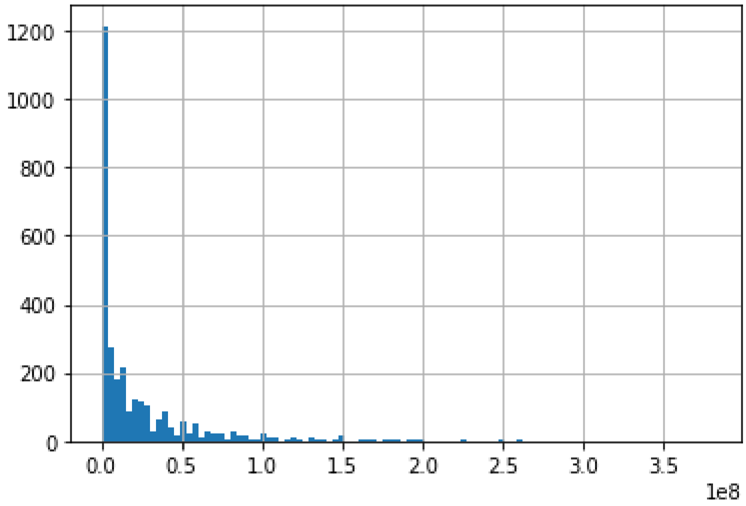
Out[73]:

```
id                                                                    12
01
belongs_to_collection    [{'id': 222639, 'name': 'Detective Dee Collec
t...
budget                                                            130000
00
genres                   [{'id': 28, 'name': 'Action'}, {'id': 12, 'na
m...
homepage                                                               N
aN
imdb_id                                                          tt11233
73
original_language
zh
original_title                                                    狄仁傑
之通天帝國
overview                 An exiled detective is recruited to solve a s
e...
popularity                                                         7.175
48
poster_path                               /2PHpd9dMhrvEaeQk0zRTUQUm2EO.j
pg
production_companies              [{'name': 'Huayi Brothers', 'id': 339
3}]
production_countries     [{'iso_3166_1': 'HK', 'name': 'Hong Kong'},
{'...
release_date                                                       9/18/
10
runtime                                                                1
19
spoken_languages         [{'iso_639_1': 'zh', 'name': '普通话'}, {'iso_63
9...
status                                                            Releas
ed
tagline                              The Fate of an Empire Is in His Han
ds
title                    Detective Dee and the Mystery of the Phantom
F...
Keywords                 [{'id': 703, 'name': 'detective'}, {'id': 150
3...
cast                     [{'cast_id': 1, 'character': 'Detective Dee',
...
crew                     [{'credit_id': '52fe4768c3a36847f8133ea7', 'd
e...
revenue                                                           517232
85
Name: 1200, dtype: object
```

Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x218f7d05988>
```

```
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:81: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:93: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```



Write your answer here 1. The genres for 'The Book of Mormon Movie, Volume 1: The Journey' is NaN. On Wikipedia, it indicats that this film is a adventure drama. Here is the link: https://en.wikipedia.org/wiki/The_Book_of_Mormon_Movie (https://en.wikipedia.org/wiki/The_Book_of_Mormon_Movie)
I replaced the NaN in genres column with Adventure and Drama. The code is: train.genres[470]="[{'id': 12, 'name': 'Adventure'}, {'id': 18, 'name': 'Drama'}]"

2. Jackpot released on 7/26/2001 has no genres shown. Wikipedia indicates this is a comedy and drama film. Here is the link: https://en.wikipedia.org/wiki/Jackpot_(2001_film (https://en.wikipedia.org/wiki/Jackpot_(2001_film)) I replaced the NaN in genres column with comedy and Drama. The code is: train.genres[1622]="[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'name': 'Drama'}]"

3. Ryaba, My Chicken relased on 10/1/1994 has no genres shown. Wikipedia indicates this is a comedy. Here is the link:
https://en.wikipedia.org/wiki/Assia_and_the_Hen_with_the_Golden_Eggs#:~:text=Asya%20and%20the%20He
(https://en.wikipedia.org/wiki/Assia_and_the_Hen_with_the_Golden_Eggs#:~:text=Asya%20and%20the%20H
I replaced the NaN in genres column with comedy. The code is: train.genres[1814]="[{'id': 35, 'name': 'Comedy'}]"

4. Sky. Plane. Girl relased on 9/2/2002 has no genres shown. The follwoing webpage indicates it is a Romance and Drama film. Here is the link: https://sovietmoviesonline.com/melodrama/480-nebo-samolet-devushka.html (https://sovietmoviesonline.com/melodrama/480-nebo-samolet-devushka.html) I replaced the NaN in genres column with Romance and Drama. The code is: train.genres[1819]="[{'id': 10749, 'name': 'Romance'}, {'id': 18, 'name': 'Drama'}]"

5. Amarkalam release on 8/25/1999 has no genres shown. The wikipedia indicates it is a Action and Romance movie. Here is the link: https://en.wikipedia.org/wiki/Amarkalam (https://en.wikipedia.org/wiki/Amarkalam) I replaced the NaN in genres column with Action and Romance. The code is: train.genres[2423]="[{'id': 28, 'name': 'Action'}, {'id': 10749, 'name': 'Romance'}]"

6. Lift released on 7/1/2006 has no genres shown. The follwoing webpage indicates it is a Thriller film. Here is the link: https://www.imdb.com/title/tt0833448/ (https://www.imdb.com/title/tt0833448/) I replaced the NaN in genres column with Thriller. The code is: train.genres[2686]="[{'id': 53, 'name': 'Thriller'}]"

7.Rita's Last Fairy Tale released on 11/1/2012 has no genres shown. The following webpage indicates it is a Drama film. Here is the link: https://en.wikipedia.org/wiki/Rita%27s_Last_Fairy_Tale (https://en.wikipedia.org/wiki/Rita%27s_Last_Fairy_Tale) I replaced the NaN in genres column with Drama. The code is: train.genres[2900]="[{'id': 18, 'name': 'Drama'}]"

8.The Worst Christmas of My Life released on 12/22/2012 has no overivew shown. The Wikipedia has its plot, below is the link https://en.wikipedia.org/wiki/The_Worst_Christmas_of_My_Life (https://en.wikipedia.org/wiki/The_Worst_Christmas_of_My_Life) I replace the Nan in overview column with the plot found in wikipedia, the code is: train.overview[390]="Paolo needs to reach the castle of Alberto Caccia, where he is invited to spend Christmas holidays with his wife Margherita, at the ninth month of pregnancy, and her family. Among various mishaps and blunders, Paolo will come to make everyone believe, because of a misunderstanding, that Alberto is dead because of his fault."

9.I change the runtime of The Worst Christmas of My Life released (released on 12/22/2012) from 0 minutes to 86 minutes from info on wikipedia.The wikipedia link is below: https://en.wikipedia.org/wiki/The_Worst_Christmas_of_My_Life (https://en.wikipedia.org/wiki/The_Worst_Christmas_of_My_Life) The code is: train.runtime[390]=86

1. I change the runtime of А поутру они проснулись (2003) from 0 minutes to 90 minutes based on info from this website below: https://www.kinopoisk.ru/film/252021/ (https://www.kinopoisk.ru/film/252021/) the code is: train.runtime[591]=90
2. I change the runtime of ¿Quién mató a Bambi? (2013) from 0 minutes to 86 minutes based on info from this website below: https://www.imdb.com/title/tt2604346/ (https://www.imdb.com/title/tt2604346/) the code is: train.runtime[924]=86
3. I change the runtime of ¿Quién mató a Bambi? (2013) from 0 minutes to 93 minutes based on info from this website below: https://www.imdb.com/title/tt2076251/ (https://www.imdb.com/title/tt2076251/) the code is: train.runtime[977]=93
4. I change the runtime of ¿Quién mató a Bambi? (2013) from 0 minutes to 91 minutes based on info from this website below: https://en.wikipedia.org/wiki/Cry,_Onion! (https://en.wikipedia.org/wiki/Cry,_Onion!) the code is: train.runtime[1255]=91
5. I change the runtime of ¿Quién mató a Bambi? (2013) from 0 minutes to 93 minutes based on info from this website below: https://en.wikipedia.org/wiki/All_at_Once_(2014_film (https://en.wikipedia.org/wiki/All_at_Once_(2014_film)) the code is: train.runtime[1541]=93
6. I change the runtime of Missing (2013) from 0 minutes to 86 minutes based on info from this website below: https://www.rottentomatoes.com/m/vermist (https://www.rottentomatoes.com/m/vermist) the code is: train.runtime[1874]=86
7. I change the runtime of Missing (2013) from 0 minutes to 108 minutes based on info from this website below: https://www.imdb.com/title/tt0477337/ (https://www.imdb.com/title/tt0477337/) the code is: train.runtime[2150]=108
8. I change the runtime of Hooked on the Game 2. The Next Level(2010) from 0 minutes to 86 minutes based on info from this website below: https://www.themoviedb.org/movie/37851-na-igre-2-novyy-uroven?language=en-US (https://www.themoviedb.org/movie/37851-na-igre-2-novyy-uroven?language=en-US) the code is: train.runtime[2498]=86
9. I change the runtime of My Old Classmate(2010) from 0 minutes to 98 minutes based on info from this website below: https://en.wikipedia.org/wiki/My_Old_Classmate (https://en.wikipedia.org/wiki/My_Old_Classmate) the code is: train.runtime[2645]=98
10. I change the runtime of My Old Classmate(2010) from 0 minutes to 111 minutes based on info from this website below: https://www.imdb.com/title/tt0278675/ (https://www.imdb.com/title/tt0278675/) the code is: train.runtime[2785]=111
11. I change the runtime of My Old Classmate(2010) from 0 minutes to 96 minutes based on info from this website below: https://www.imdb.com/title/tt24567209 (https://www.imdb.com/title/tt24567209) the code is: train.runtime[2865]=96
12. I change the revenue of The Cookout from 12(I assume the measure here mean to be in million) to 12,009,070 in Cumulative Worldwide Gross. This is the link: https://www.imdb.com/title/tt0380277/ (https://www.imdb.com/title/tt0380277/)

I droped 3 row where budget is very high and revenue is very low. Then I reset_index of the train data. The train data reduced to 2997 rows.

Part 2 - Word Cloud

In [45]:

```
# TODO: code for generating word clouds
#2.1
#This part is for 'genres' wordcloud
train['genres2']=[list() for x in range(len(train['genres']))]
train['genres1']=train['genres'].apply(literal_eval)
text=[]
for i in range(0,len(train['genres1'])):
    for j in range(0,len(train['genres1'][i])):
            train['genres2'][i].append(train['genres1'][i][j]['name'])
            text.append(train['genres1'][i][j]['name'])
text1=[]
text1=','.join(text)
#mask = np.array(Image.open("spiderman.jpg"))
stopwords = set(STOPWORDS)
word_could_dict = Counter(text)
wordcloud = WordCloud().generate_from_frequencies(word_could_dict)
wordcloud = WordCloud(min_font_size=13,colormap='Set2',random_state=1,background_col
or='black', collocations=False,
                      width=1000, height=1000).generate_from_frequencies(word_could_
dict)
#add to above parameter: mask = mask
wordcloud.to_file('box1.jpg')
plt.imshow(wordcloud,interpolation='bilinear')
plt.title('Genres Wordcloud')
plt.axis("off")
plt.show()
```

Out[45]:

```
<wordcloud.wordcloud.WordCloud at 0x218d9533b88>
```

Out[45]:

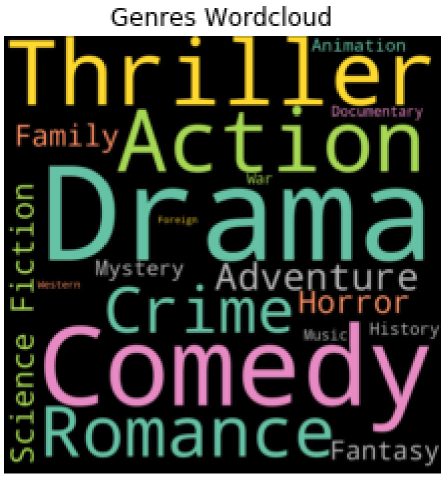
```
<matplotlib.image.AxesImage at 0x218d988b348>
```

Out[45]:

```
Text(0.5, 1.0, 'Genres Wordcloud')
```

Out[45]:

```
(-0.5, 999.5, 999.5, -0.5)
```



In [46]:

```
#2.2
#This part is for 'Keywords' WordCloud
train['Keywords1']=train.Keywords.replace(np.NaN, '[]',inplace=False)
train['Keywords2']=[list() for x in range(len(train['Keywords1']))]
train['Keywords2']=train['Keywords1'].apply(literal_eval)
text=[]
train['Keywords3']=[list() for x in range(len(train['Keywords2']))]
for i in range(0,len(train['Keywords2'])):
    for j in range(0,len(train['Keywords2'][i])):
            train['Keywords3'][i].append(train['Keywords2'][i][j]['name'])
            text.append(train['Keywords2'][i][j]['name'])
text1=[]
text1=','.join(text)
#mask = np.array(Image.open("map1.jpg"))
stopwords = set(STOPWORDS)
word_could_dict = Counter(text)
wordcloud = WordCloud().generate_from_frequencies(word_could_dict)
wordcloud = WordCloud(min_font_size=13,colormap='Set2', random_state=1,background_co
lor='black', collocations=False,
                      width=1000, height=1000).generate_from_frequencies(word_could_
dict)
#add to above parameter: mask = mask
wordcloud.to_file('keywords.jpg')
plt.imshow(wordcloud,interpolation='bilinear')
plt.title('Keywords Wordcloud')
plt.axis("off")
plt.show()
```

Out[46]:

```
<wordcloud.wordcloud.WordCloud at 0x218f646a108>
```

Out[46]:

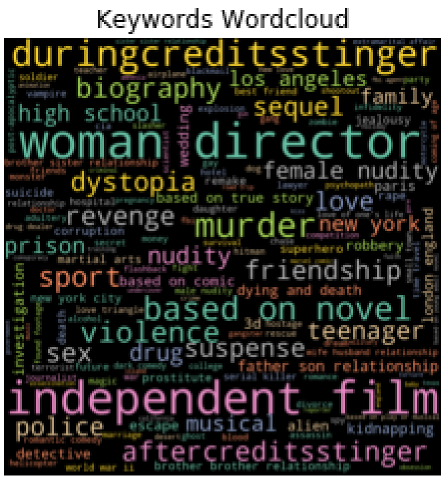
```
<matplotlib.image.AxesImage at 0x218f63d2408>
```

Out[46]:

```
Text(0.5, 1.0, 'Keywords Wordcloud')
```

Out[46]:

```
(-0.5, 999.5, 999.5, -0.5)
```



Write your answer here

In [47]:

```
#2.3
#WordCloud for original_title
text=' '.join(train['original_title'])
stopwords = set(STOPWORDS)
#word_could_dict = Counter(text)
wordcloud = WordCloud().generate_from_frequencies(word_could_dict)
wordcloud = WordCloud(min_font_size=13,colormap='Set2', random_state=1, background_c
olor='black', collocations=False,
                      width=1000, height=1000).generate(text)
#add to above parameter: mask = mask
wordcloud.to_file('box3.jpg')
plt.imshow(wordcloud,interpolation='bilinear')
plt.title('orignial_title Wordcloud')
plt.axis("off")
plt.show()
```

Out[47]:

```
<wordcloud.wordcloud.WordCloud at 0x218fefe9588>
```

Out[47]:

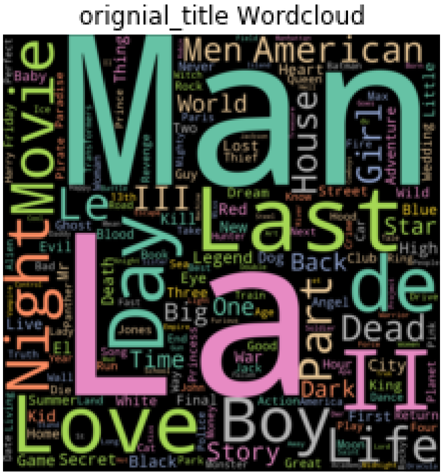
```
<matplotlib.image.AxesImage at 0x218feafe348>
```

Out[47]:

```
Text(0.5, 1.0, 'orignial_title Wordcloud')
```

Out[47]:

```
(-0.5, 999.5, 999.5, -0.5)
```



In [48]:

```
#2.4
#WordCloud for overview column
train['overview1']=""
train['overview1']=train.overview.replace(np.NaN, '[]',inplace=False)
text=' '.join(train['overview1'])
stopwords = set(STOPWORDS)
wordcloud = WordCloud().generate_from_frequencies(word_could_dict)
wordcloud = WordCloud(min_font_size=13,colormap='Set2', random_state=1, background_color='black', collocations=False,
                      width=1000, height=1000).generate(text)
#add to above parameter: mask = mask
wordcloud.to_file('box4.jpg')
plt.imshow(wordcloud,interpolation='bilinear')
plt.title('overview Wordcloud')
plt.axis("off")
plt.show()
```

Out[48]:

```
<wordcloud.wordcloud.WordCloud at 0x218f64a6448>
```

Out[48]:

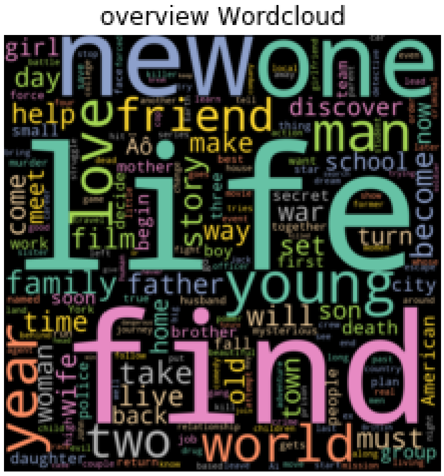
```
<matplotlib.image.AxesImage at 0x218df6a05c8>
```

Out[48]:

```
Text(0.5, 1.0, 'overview Wordcloud')
```

Out[48]:

```
(-0.5, 999.5, 999.5, -0.5)
```



Part 3 - Time Series Analysis

In [76]:

```
# TODO: code for time series analysis
#task3.1 week plot
import datetime
from datetime import datetime,date
train['release_date1']=train['release_date'].apply(lambda x: datetime.strptime(x,'%m/%d/%y'))
train['release_date2']=''
for i in range(0,len(train['release_date1']
                    )):
    if train['release_date1'][i].year >=2020:
        year = train['release_date1'][i].year-100
    else:
        year=train['release_date1'][i].year
    train['release_date2'][i]=date(year,train['release_date1'][i].month,train['release_date1'][i].day)
train['release_weekday']=train['release_date2'].apply(lambda x: x.strftime('%A'))
train['release_date1']=train['release_date'].apply(lambda x: datetime.strptime(x,'%m/%d/%y'))
train['release_date2']=''
for i in range(0,len(train['release_date1']
                    )):
    if train['release_date1'][i].year >=2020:
        year = train['release_date1'][i].year-100
    else:
        year=train['release_date1'][i].year
    train['release_date2'][i]=date(year,train['release_date1'][i].month,train['release_date1'][i].day)
train['release_weekday']=train['release_date2'].apply(lambda x: x.strftime('%A'))
weekday=train['release_weekday'].value_counts()
weekday1=weekday[['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']]
plt.plot(weekday1, linestyle='-', marker='o', markersize=10)
plt.title('# of Movies Released by Day of Week in the Training Dataset From 1921 to 2017')
plt.xlabel('Day of Week')
plt.ylabel('Number of Movies Count')
plt.xticks(rotation=25)
plt.show()
```

```
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:13: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  del sys.path[0]
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:23: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[76]:

```
[<matplotlib.lines.Line2D at 0x218f62d0048>]
```

Out[76]:

```
Text(0.5, 1.0, '# of Movies Released by Day of Week in the Training Data
set From 1921 to 2017')
```

Out[76]:

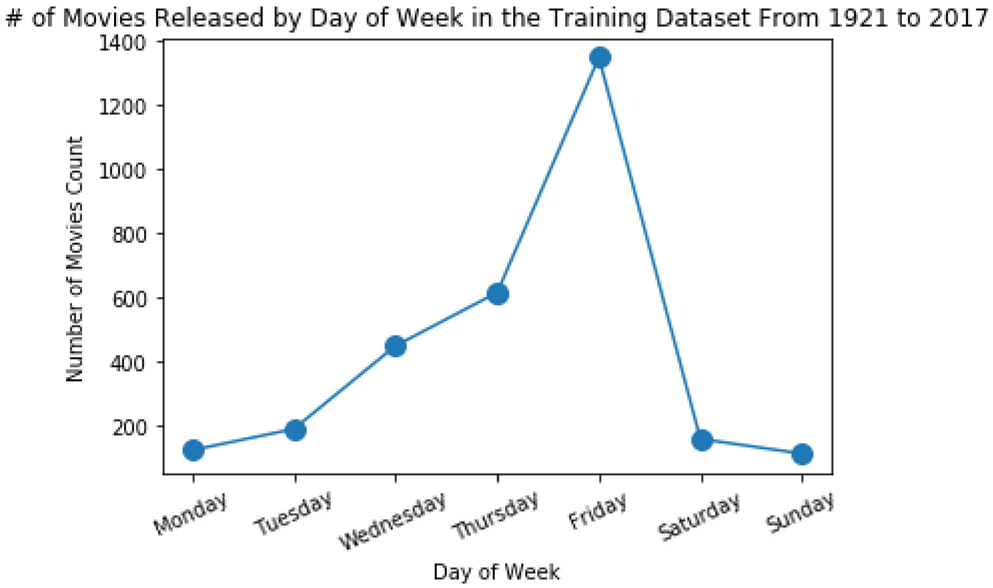
```
Text(0.5, 0, 'Day of Week')
```

Out[76]:

```
Text(0, 0.5, 'Number of Movies Count')
```

Out[76]:

```
([0, 1, 2, 3, 4, 5, 6], <a list of 7 Text xticklabel objects>)
```



Parttern observed

We notice that most movies are released on Fridays. This is because most people finish with their works at Friday evening. They start relaxing by going to to watch movies. Follow that are Thursdays and Wednesdays.Mondays and Sundays have least movies released.

In [52]:

```
#Q3.2 - Plot by Month
train['release_weekday']=train['release_date2'].apply(lambda x: x.strftime('%A'))
train['release_month']=train['release_date2'].apply(lambda x: x.strftime('%b'))
month=train['release_month'].value_counts()
month1=month[['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','De
c']]
plt.plot(month1, linestyle='-', marker='o', markersize=10)
plt.title('# of Movies Released by Month in the Training Dataset From 1921 to 2017')
plt.xlabel('Month')
plt.ylabel('Number of Movies Count')
plt.show()
```

Out[52]:

```
[<matplotlib.lines.Line2D at 0x218f6250648>]
```

Out[52]:

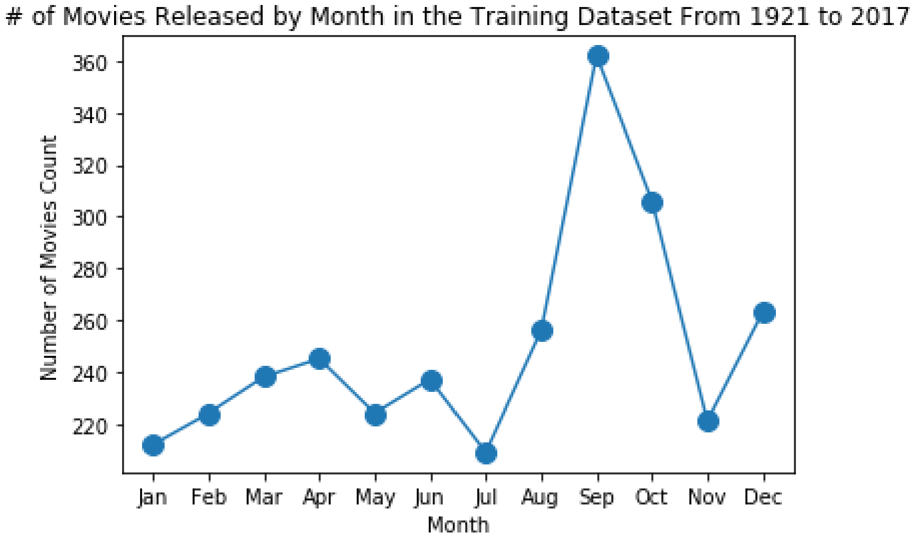
```
Text(0.5, 1.0, '# of Movies Released by Month in the Training Dataset Fr
om 1921 to 2017')
```

Out[52]:

```
Text(0.5, 0, 'Month')
```

Out[52]:

```
Text(0, 0.5, 'Number of Movies Count')
```



Pattern observed

September seems to have the highest count of movies released. Follow by that is October, December and August.The first half year has lower release rate.

In [78]:

```
## Q3.3 Plot by Year
train['release_year']=train['release_date'].apply(lambda x: datetime.strptime(x,'%m/
%d/%y').strftime('%Y'))
for i in range(0,len(train.release_year)):
    if int(train.release_year[i])>=2020:
        train.release_year[i]=int(train.release_year[i])-100
        train.release_year[i]=str(train.release_year[i])
year=train['release_year'].value_counts()
a=[]
c=[]
#year.index.max()    #2068  2017
#year.index.min()    #1969  1921
for i in range(1921,2018):#2069
   a.append(str(i));
b=year.index
for j in a:
    if j not in b:
        c.append(j);
for i in c:
    year.loc[i]=0;
year1=year.loc[a]
plt.plot(year1, linestyle='-', marker='o', markersize=2)
plt.title('# of Movies Released by Year in the Training Dataset From 1921 to 2017')
plt.xlabel('Year')
plt.ylabel('Number of Movies Count')
plt.xticks(ticks=['1921','1940','1960','1980', '2000', '2017'],rotation=30)
plt.show()
```

```
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:5: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:6: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[78]:

```
[<matplotlib.lines.Line2D at 0x218f6258e48>]
```

Out[78]:

```
Text(0.5, 1.0, '# of Movies Released by Year in the Training Dataset Fro
m 1921 to 2017')
```

Out[78]:

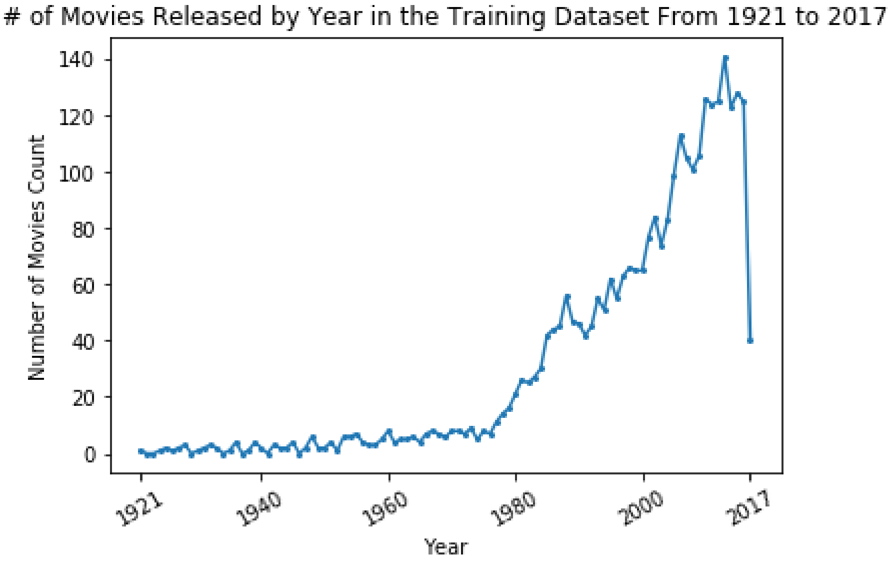
```
Text(0.5, 0, 'Year')
```

Out[78]:

```
Text(0, 0.5, 'Number of Movies Count')
```

Out[78]:

```
([<matplotlib.axis.XTick at 0x218df2e8e48>,
  <matplotlib.axis.XTick at 0x218df3983c8>,
  <matplotlib.axis.XTick at 0x218f6209fc8>,
  <matplotlib.axis.XTick at 0x218f6250e48>,
  <matplotlib.axis.XTick at 0x218df0a9dc8>,
  <matplotlib.axis.XTick at 0x218df09d548>],
 <a list of 6 Text xticklabel objects>)
```



pattern observed

The first movie released on the training data is 'The kid', the role 'The Tramp' was played by star Charlie Chaplin.

Part 4 - Cast Power

In [97]:

```python
# TODO: code for measuring cast power
from collections import Counter
import statistics
from statistics import mean
train['cast1']=train['cast'].replace(np.NaN,'[]',inplace=False)
train['cast2']=train['cast1'].apply(literal_eval)
list4=[]
train['cast3']=[list() for x in range(len(train['cast']))]
for i in range(len(train['cast2'])):
    for j in range(len(train['cast2'][i])):
        list4.append(str(train['cast2'][i][j]['id']))
        train['cast3'][i].append(str(train['cast2'][i][j]['id']))
star=Counter(list4)
star1=dict(star)
train['cast4']=train['cast3']
for i in range(len(train['cast3'])):
    for j in range(len(train['cast3'][i])):
        train['cast3'][i][j]=star1[str(train['cast3'][i][j])]
        #print(train['cast3'][i][j],i,j,train['cast4'][i][j])
train['castPower']=train['cast4'].apply(lambda x: 0 if x == [] else mean(x))#if change mean(x) to sum(x), corr become 0.4636
a=mean(train['castPower'])*2997/(2997-26)
for i in range(len(train['castPower'])):
    if train['castPower'][i]==0:
        train['castPower'][i]=a
train['castPower5']=pd.to_numeric(train['castPower'])
train['logCastPower1']=np.log(train['castPower5'])
train['castPower5'].corr(train['revenue'])
```

```
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:24: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[97]:

```
0.11132182803589868
```

The correlation between calculated castPower and revenue is 0.1113. The correlation is under 0.5 and is weak. corr=0.1113 is bery low, not very useful of predicting the revenue. Methed: I count number of each star appear in all movies as their individual star power. Then I average all individual starpowers in a movie, and I take this average as the final starpower for that movie.

If I change castPower to without division by the number of cast in that movie. The corr would rise and become 0.46.

Part 5 - External Dataset

In [80]:

```
# TODO: code for integrating external dataset
original_url='https://drive.google.com/file/d/1f5GU2BDXU43a3QkBL2CXTngjX2tMkHxc/vie
w?usp=sharing'
file_id = original_url.split('/')[-2]
dwn_url='https://drive.google.com/uc?export=download&id=' + file_id
url = requests.get(dwn_url).text
csv_data1 = StringIO(url)
external2 = pd.read_csv(csv_data1)
train1v=pd.merge(train, external2, how='left',left_on='imdb_id', right_on='tconst')
```

Write your answer here

The external data named 'data.csv' public downloadable link is: https://drive.google.com/file/d/1f5GU2BDXU43a3QkBL2CXTngjX2tMkHxc/view?usp=sharing (https://drive.google.com/file/d/1f5GU2BDXU43a3QkBL2CXTngjX2tMkHxc/view?usp=sharing)

The data.csv was downloaded from https://datasets.imdbws.com/ (https://datasets.imdbws.com/) inside the https://www.imdb.com/interfaces/ (https://www.imdb.com/interfaces/) webpage. and the file I download originally was title.ratings.tsv.gz I download the above .tsv.gz file and unziped it locally on my computer and derived a data.tsv file. I used online converter to convert it to data.csv file. Then I upload this data.csv to my google drive with a public downloadable share link shown above. In my code, I access this data.csv file direclty from the google drive public downloadable share link.

In the data.csv file there are 3 columns

1. tconst which is the same as imdb_id column in train data set. 2.averageRating is the average of rating viewers provided ranging from 1 to 10.
2. numVotes is the number of votes each movie receive from public and the range is from 5 to 2.292763e+06.

The higher the averageRating and numVotes the higher the revenue expected.They are each positive correlated with revenue. These two features act as two important feafures for revenue prediction.

Part 6 - Informative Plots

In [81]:

```
# TODO: code for producing informative plots
#task 6.1
train1v.year1=''
train1v['year1']=train1v['release_date2'].apply(lambda x: x.strftime('%Y'))
train1v1=train1v.groupby('year1').mean()
plt.plot(train1v1.averageRating, linestyle='-', marker='o', markersize=2)
plt.title('Average rating by Year in the Training Dataset From 1921 to 2017')
plt.xlabel('Year')
plt.ylabel('averageRating per movie')
plt.xticks(ticks=['1921','1940','1960','1980', '2000', '2017'],rotation=30)
plt.show()
```

Out[81]:

```
[<matplotlib.lines.Line2D at 0x21893d555c8>]
```

Out[81]:

```
Text(0.5, 1.0, 'Average rating by Year in the Training Dataset From 1921
to 2017')
```

Out[81]:

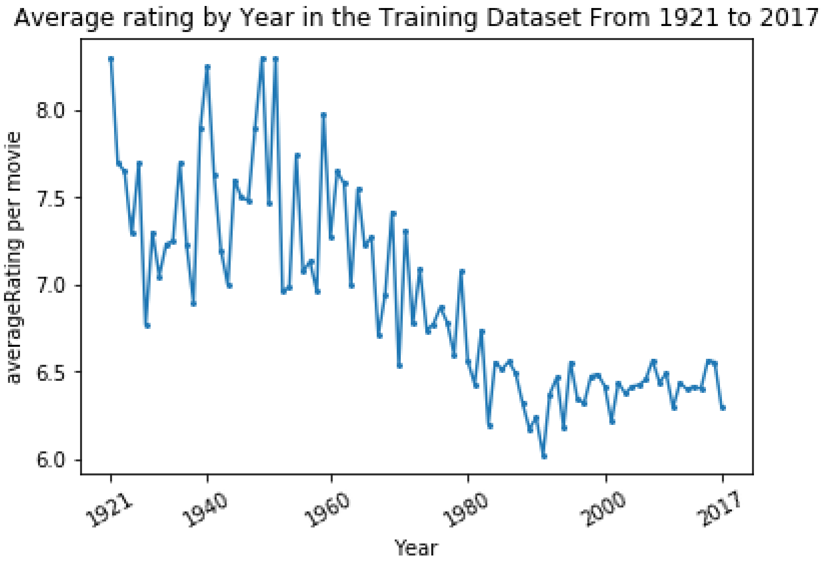
```
Text(0.5, 0, 'Year')
```

Out[81]:

```
Text(0, 0.5, 'averageRating per movie')
```

Out[81]:

```
([<matplotlib.axis.XTick at 0x21893d63748>,
  <matplotlib.axis.XTick at 0x218f6439288>,
  <matplotlib.axis.XTick at 0x21893d5c248>,
  <matplotlib.axis.XTick at 0x21893d48a48>,
  <matplotlib.axis.XTick at 0x21893d48448>,
  <matplotlib.axis.XTick at 0x21893d35b88>],
 <a list of 6 Text xticklabel objects>)
```



Task 6.1 Althought there are ups and downs in the consecutive years, overall this plot shows that the average rating decreases over time. Perhaps people like older styles movie over new ones.

In [82]:

```
#task 6.2
plt.scatter(np.log(train1v.revenue), np.log(train1v.numVotes),s=1)
plt.xlabel('Log(revenue in US Dollar)')
plt.ylabel('log(Number of Votes per movie)')
train1v.numVotes.corr(train1v.revenue)
```

Out[82]:

```
<matplotlib.collections.PathCollection at 0x21893cbc388>
```

Out[82]:

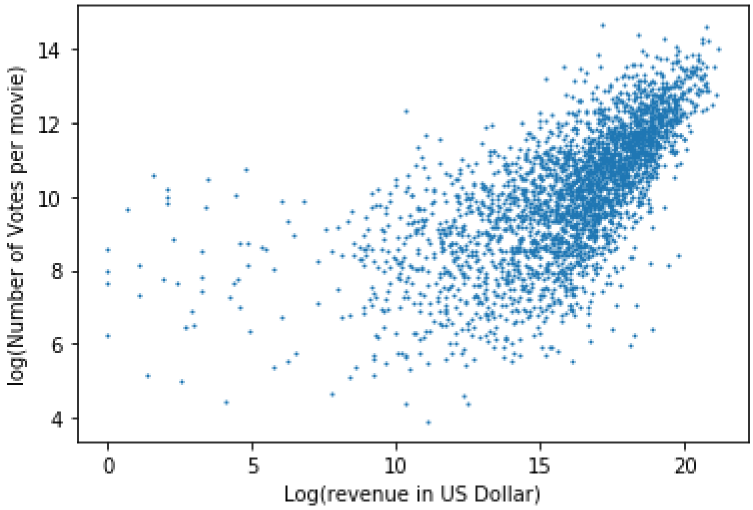
```
Text(0.5, 0, 'Log(revenue in US Dollar)')
```

Out[82]:

```
Text(0, 0.5, 'log(Number of Votes per movie)')
```

Out[82]:

```
0.6293124879744174
```



task 6.2 scatter plot

As we can see from the scatter plot that log(revenue) and log(Number of Votes) are highly correlated with correlated coeffienct of 0.6294. This mean the higher the revenue the more people are voting it.

In [83]:

```
#task 6.3
from matplotlib.pyplot import figure
plt.figure(figsize=(20,5))
train1v.groupby('original_language')['numVotes'].mean().plot(kind='bar')
plt.xlabel('Short Names for original_language',fontsize=18)
plt.ylabel('Average Number of Votes per movie',fontsize=18)
plt.xticks(rotation=0,fontsize=16)
plt.yticks(fontsize=14)
```

Out[83]:

```
<Figure size 1440x360 with 0 Axes>
```

Out[83]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21893c86208>
```

Out[83]:

```
Text(0.5, 0, 'Short Names for original_language')
```

Out[83]:

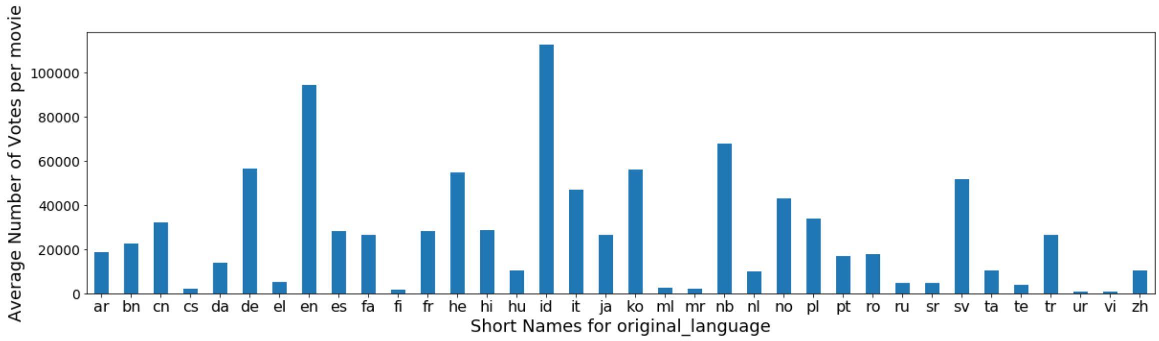
```
Text(0, 0.5, 'Average Number of Votes per movie')
```

Out[83]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32,
33,
        34, 35]),
 <a list of 36 Text xticklabel objects>)
```

Out[83]:

```
(array([     0.,  20000.,  40000.,  60000.,  80000., 100000., 120000.]),
 <a list of 7 Text yticklabel objects>)
```



Task 6.3 Indonesia(id) movies has the highest number of votes. Only 1 movie is in Indonesia language. US is the second place. Movie in vi language has the least average number of votes.

In [84]:

```
#task 6.4
plt.scatter(np.log(train1v.numVotes),train1v.averageRating,s=1)
plt.xlabel('np.log(train1v.numVotes)')
plt.ylabel('averageRating per movie')
train1v.numVotes.corr(train1v.revenue)
train1v.averageRating.corr(train1v.revenue)
train1v.describe()
```

Out[84]:

```
<matplotlib.collections.PathCollection at 0x218ee3e1708>
```

Out[84]:

```
Text(0.5, 0, 'np.log(train1v.numVotes)')
```

Out[84]:

```
Text(0, 0.5, 'averageRating per movie')
```

Out[84]:

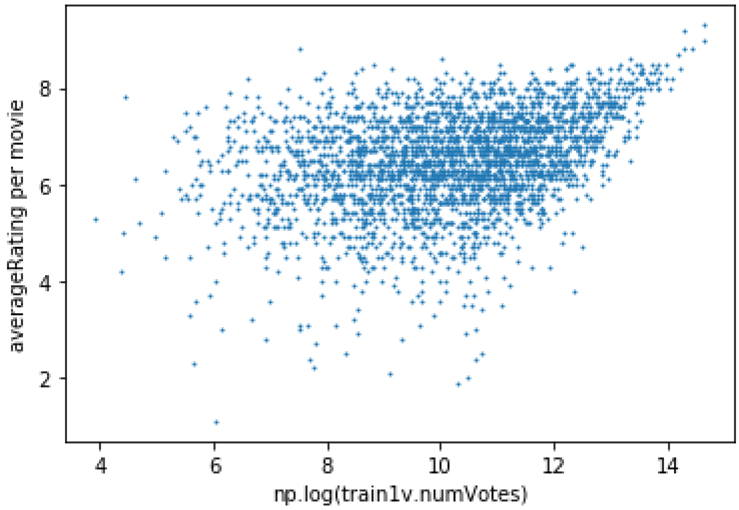
```
0.6293124879744174
```

Out[84]:

```
0.14399579299533832
```

Out[84]:

	id	budget	popularity	runtime	revenue	belongs_to_colle
count	2997.000000	2.997000e+03	2997.000000	2995.000000	2.997000e+03	2997.0
mean	1500.012346	2.476448e+07	8.470868	108.243406	6.683113e+07	0.2
std	866.130412	3.584469e+07	12.107675	21.034243	1.375701e+08	0.4
min	1.000000	2.500000e+03	0.000001	11.000000	1.000000e+00	0.0
25%	750.000000	8.000000e+06	4.037707	94.000000	2.452566e+06	0.0
50%	1500.000000	8.000000e+06	7.390012	104.000000	1.692814e+07	0.0
75%	2250.000000	2.900000e+07	10.893224	118.000000	6.892915e+07	0.0
max	3000.000000	3.800000e+08	294.337037	338.000000	1.519558e+09	1.0



task6.4

The averageRating and numVotes are positively correlated with correlation coefficient of 0.14399. I was expecting the coefficient to be higher.

In [85]:

```
#task 6.5
plt.plot(train1v1.numVotes, linestyle='-', marker='o', markersize=2)
plt.title('average number of votes by Year in the Training Dataset From 1921 to 201
7')
plt.xlabel('Year')
plt.ylabel('average number of votes per movie')
plt.xticks(ticks=['1921','1940','1960','1980', '2000', '2017'],rotation=30)
plt.show()
```

Out[85]:

```
[<matplotlib.lines.Line2D at 0x218ee53fe48>]
```

Out[85]:

```
Text(0.5, 1.0, 'average number of votes by Year in the Training Dataset
From 1921 to 2017')
```

Out[85]:

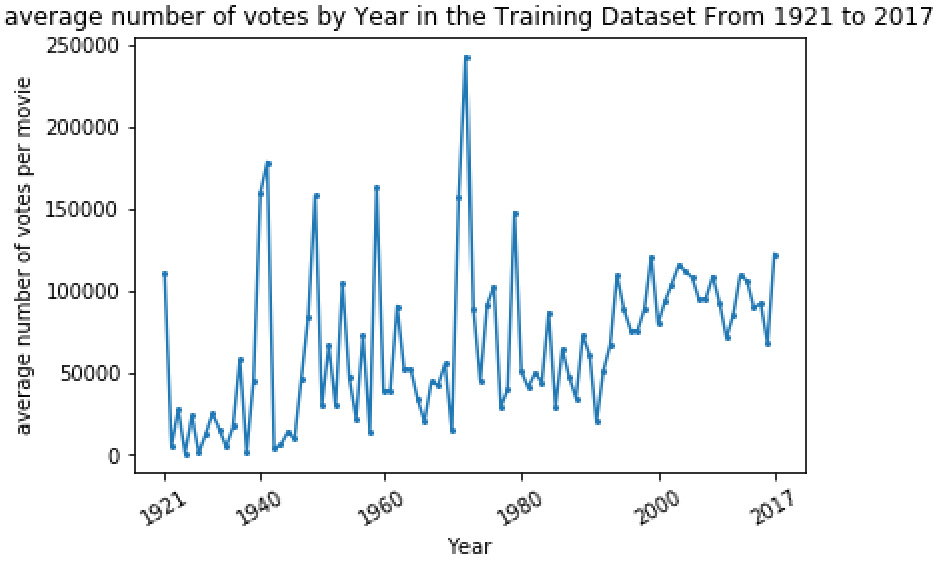
```
Text(0.5, 0, 'Year')
```

Out[85]:

```
Text(0, 0.5, 'average number of votes per movie')
```

Out[85]:

```
([<matplotlib.axis.XTick at 0x218ee52fe48>,
  <matplotlib.axis.XTick at 0x218ee52f4c8>,
  <matplotlib.axis.XTick at 0x218ee534f88>,
  <matplotlib.axis.XTick at 0x218ee540a08>,
  <matplotlib.axis.XTick at 0x218ee544248>,
  <matplotlib.axis.XTick at 0x218ee5448c8>],
 <a list of 6 Text xticklabel objects>)
```



The highest movie average rating occured in early 1970. The overall average rating per movie by year has an increasing trend.

Part 7 - Pairwise Pearson Correlation

In [88]:

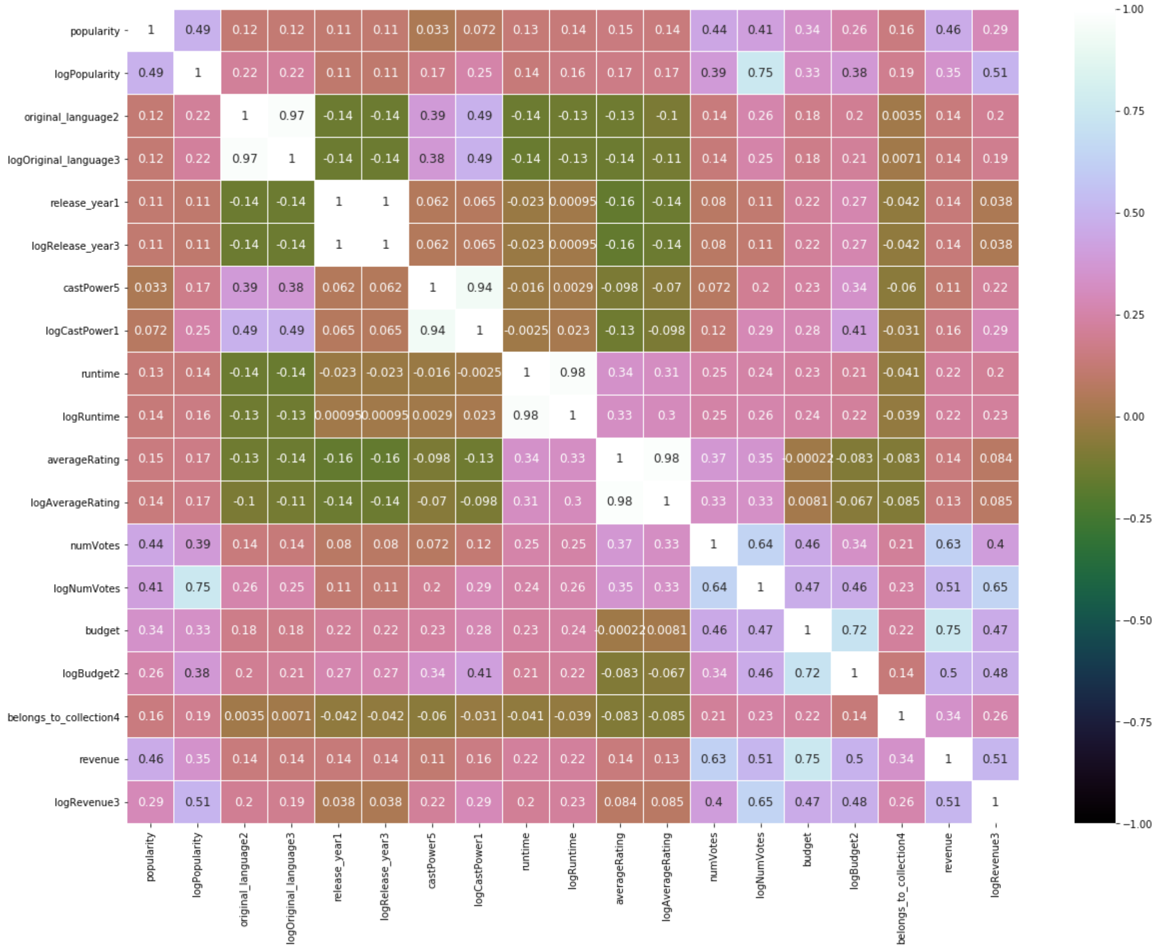
```
# TODO: code for pairwise Pearson correlation
import seaborn as sns
#for i in range(len(train1v['release_year'])):
train1v['logRuntime']=np.log(train1v['runtime'])
train1v['logAverageRating']=np.log(train1v['averageRating'])
train1v['logNumVotes']=np.log(train1v['numVotes'])
train1v['logPopularity']=np.log(train1v['popularity'])
train1v['logOriginal_language3']=np.log(train1v['original_language2'])
train1v['logRelease_year3']=pd.to_numeric(train1v['release_year'], errors='coerce')
train1v['logRevenue2']=pd.to_numeric(train1v['revenue'])
train1v['logRevenue3']=np.log(train1v['logRevenue2'])
train1v['logBudget2']=np.log(train1v['budget'])
train1v['release_year1']=pd.to_numeric(train1v['release_year'])
train2v=train1v[['popularity','logPopularity','original_language2','logOriginal_language3','release_year1','logRelease_year3','castPower5','logCastPower1','runtime','logRuntime','averageRating','logAverageRating','numVotes','logNumVotes','budget','logBudget2','belongs_to_collection4','revenue','logRevenue3']]
train2v.corr(method='pearson')
plt.subplots(figsize=(20,15))
Heatmap=sns.heatmap(train2v.corr(method='pearson'),vmin=-1, vmax=1, linewidths=0.01, annot=True,cmap='cubehelix',annot_kws={"fontsize":12})
figure5 = Heatmap.get_figure()
#figure5.savefig('heatmap1.jpg', dpi=400)
```

Out[88]:

	popularity	logPopularity	original_language2	logOriginal_language3	rele
popularity	1.000000	0.489311	0.122701	0.121770	
logPopularity	0.489311	1.000000	0.217569	0.220485	
original_language2	0.122701	0.217569	1.000000	0.974556	
logOriginal_language3	0.121770	0.220485	0.974556	1.000000	
release_year1	0.109416	0.105903	-0.138962	-0.139992	
logRelease_year3	0.109416	0.105903	-0.138962	-0.139992	
castPower5	0.033235	0.173673	0.386012	0.380493	
logCastPower1	0.071621	0.253103	0.487721	0.486033	
runtime	0.129778	0.135175	-0.138887	-0.140837	
logRuntime	0.136831	0.160215	-0.129203	-0.132203	
averageRating	0.153683	0.168185	-0.125271	-0.137929	
logAverageRating	0.140718	0.171153	-0.101184	-0.113447	
numVotes	0.443942	0.385808	0.144250	0.140237	
logNumVotes	0.409824	0.751242	0.264409	0.253341	
budget	0.336394	0.331793	0.179328	0.177277	
logBudget2	0.263356	0.376922	0.199755	0.212980	
belongs_to_collection4	0.155631	0.187597	0.003493	0.007111	
revenue	0.461300	0.346377	0.142103	0.139302	
logRevenue3	0.291317	0.505794	0.196350	0.193141	

Out[88]:

```
(<Figure size 1440x1080 with 1 Axes>,
 <matplotlib.axes._subplots.AxesSubplot at 0x218edf19e08>)
```



	popularity	logPopularity	original_language2	logOriginal_language3	release_year1	logRelease_year3	castPower5	logCastPower1	runtime	logRuntime	averageRating	logAverageRating	numVotes	logNumVotes	budget	logBudget2	belongs_to_collection4	revenue	logRevenue3
popularity	1	0.49	0.12	0.12	0.11	0.11	0.033	0.072	0.13	0.14	0.15	0.14	0.44	0.41	0.34	0.26	0.16	0.46	0.29
logPopularity	0.49	1	0.22	0.22	0.11	0.11	0.17	0.25	0.14	0.16	0.17	0.17	0.39	0.75	0.33	0.38	0.19	0.35	0.51
original_language2	0.12	0.22	1	0.97	-0.14	-0.14	0.39	0.49	-0.14	-0.13	-0.13	-0.1	0.14	0.26	0.18	0.2	0.0035	0.14	0.2
logOriginal_language3	0.12	0.22	0.97	1	-0.14	-0.14	0.38	0.49	-0.14	-0.13	-0.14	-0.11	0.14	0.25	0.18	0.21	0.0071	0.14	0.19
release_year1	0.11	0.11	-0.14	-0.14	1	1	0.062	0.065	-0.023	0.00095	-0.16	-0.14	0.08	0.11	0.22	0.27	-0.042	0.14	0.038
logRelease_year3	0.11	0.11	-0.14	-0.14	1	1	0.062	0.065	-0.023	0.00095	-0.16	-0.14	0.08	0.11	0.22	0.27	-0.042	0.14	0.038
castPower5	0.033	0.17	0.39	0.38	0.062	0.062	1	0.94	-0.016	0.0029	-0.098	-0.07	0.072	0.2	0.23	0.34	-0.06	0.11	0.22
logCastPower1	0.072	0.25	0.49	0.49	0.065	0.065	0.94	1	-0.0025	0.023	-0.13	-0.098	0.12	0.29	0.28	0.41	-0.031	0.16	0.29
runtime	0.13	0.14	-0.14	-0.14	-0.023	-0.023	-0.016	-0.0025	1	0.98	0.34	0.31	0.25	0.24	0.23	0.21	-0.041	0.22	0.2
logRuntime	0.14	0.16	-0.13	-0.13	0.00095	0.00095	0.0029	0.023	0.98	1	0.33	0.3	0.25	0.26	0.24	0.22	-0.039	0.22	0.23
averageRating	0.15	0.17	-0.13	-0.14	-0.16	-0.16	-0.098	-0.13	0.34	0.33	1	0.98	0.37	0.35	-0.00022	-0.083	-0.083	0.14	0.084
logAverageRating	0.14	0.17	-0.1	-0.11	-0.14	-0.14	-0.07	-0.098	0.31	0.3	0.98	1	0.33	0.33	0.0081	-0.067	-0.085	0.13	0.085
numVotes	0.44	0.39	0.14	0.14	0.08	0.08	0.072	0.12	0.25	0.25	0.37	0.33	1	0.64	0.46	0.34	0.21	0.63	0.4
logNumVotes	0.41	0.75	0.26	0.25	0.11	0.11	0.2	0.29	0.24	0.26	0.35	0.33	0.64	1	0.47	0.46	0.23	0.51	0.65
budget	0.34	0.33	0.18	0.18	0.22	0.22	0.23	0.28	0.23	0.24	-0.00022	0.0081	0.46	0.47	1	0.72	0.22	0.75	0.47
logBudget2	0.26	0.38	0.2	0.21	0.27	0.27	0.34	0.41	0.21	0.22	-0.083	-0.067	0.34	0.46	0.72	1	0.14	0.5	0.48
belongs_to_collection4	0.16	0.19	0.0035	0.0071	-0.042	-0.042	-0.06	-0.031	-0.041	-0.039	-0.083	-0.085	0.21	0.23	0.22	0.14	1	0.34	0.26
revenue	0.46	0.35	0.14	0.14	0.14	0.14	0.11	0.16	0.22	0.22	0.14	0.13	0.63	0.51	0.75	0.5	0.34	1	0.51
logRevenue3	0.29	0.51	0.2	0.19	0.038	0.038	0.22	0.29	0.2	0.23	0.084	0.085	0.4	0.65	0.47	0.48	0.26	0.51	1

The runtime and logRuntime are highly correlated with coefficient 0.98.

But this is not interested.

The different features with highest postive correlation is 0.75 from both (logNumVotes vs logPopularity) and (budget and revenue). These two pairs both have corr=0.75.

The most negative correlation coefficient of -0.16 from both (averageRating vs release_year1) and (averageRating vs logRelease_year3). These two pairs both have corr=-0.16.

Write your answer here

Part 8 - Regression and Permutation Test

In [104]:

```
# TODO: code for your regression models and permutation tests
import random
from math import sqrt
from sklearn.utils import shuffle
from sklearn.linear_model import LinearRegression
# random.seed(1)
train1v['logBudget']=np.log(train1v['budget'])
train1v['logRevenue']=np.log(train1v['revenue'])
train4v = train1v[:2398]
test4v = train1v[2398:]
X=train4v[['logBudget']]
y=train4v['logRevenue']
X.loc[0]
Z=test4v[['logBudget']]
reg = LinearRegression().fit(X,y)
y1=y.sample(frac=1).reset_index(drop=True)
result1=reg.predict(Z)
realRmse=sqrt(mean((result1-test4v['logRevenue'])**2))
permRmse=[]
for i in range(1000):
    #set_seed(i)
    y1=y.sample(frac=1,random_state=i).reset_index(drop=True)
    reg = LinearRegression().fit(X,y1)
    result2=reg.predict(Z)
    permRmse.append(sqrt(mean((result2-test4v['logRevenue'])**2)))
import scipy
count=0
for i in range(len(permRmse)):
    if permRmse[i]<realRmse:
        count+=1
PVALUE1=count/len(permRmse)
from scipy.stats import gaussian_kde
density = gaussian_kde(permRmse)
x = np.arange(2.25, 3, 0.001)
plt.bar(x, density(x),width=0.001,label='y permuted rmse count')
plt.bar(realRmse,8,width=0.01,color='r',label='real rmse count')
plt.xlabel('RMSE Values\n(note: red line means real Rmse=2.3157 with 1 count)\n figu
re 1',fontsize=11)
plt.ylabel('Number of counts',fontsize=11)
plt.xticks(rotation=0,fontsize=16)
plt.yticks(fontsize=11)
plt.legend()
plt.title('Single Real Rmse(Red) compared to 1000 y permuted Rmses(blue)\n Underlyin
g model uses log(budget) to predict log(revenue)\n corr(logBudget,logRevenue)=0.48;
 pValue is 0',fontsize=12)
```

```
Out[104]:
```

```
logBudget    16.454568
Name: 0, dtype: float64
```

```
Out[104]:
```

```
<BarContainer object of 750 artists>
```

```
Out[104]:
```

```
<BarContainer object of 1 artists>
```

```
Out[104]:
```

```
Text(0.5, 0, 'RMSE Values\n(note: red line means real Rmse=2.3157 with 1
count)\n figure 1')
```

```
Out[104]:
```

```
Text(0, 0.5, 'Number of counts')
```

```
Out[104]:
```

```
(array([2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3. , 3.1]),
 <a list of 10 Text xticklabel objects>)
```

```
Out[104]:
```

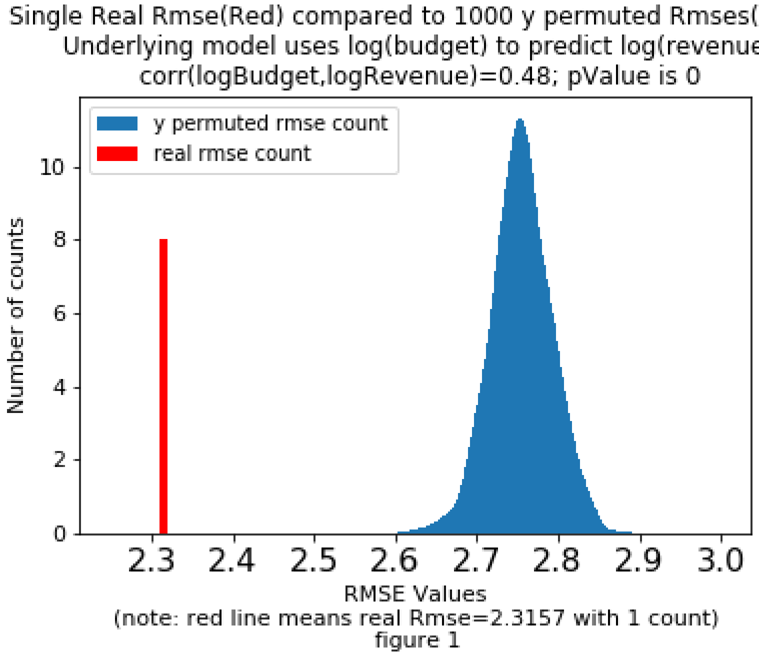
```
(array([ 0.,  2.,  4.,  6.,  8., 10., 12.]),
 <a list of 7 Text yticklabel objects>)
```

```
Out[104]:
```

```
<matplotlib.legend.Legend at 0x2188a5e1f08>
```

```
Out[104]:
```

```
Text(0.5, 1.0, 'Single Real Rmse(Red) compared to 1000 y permuted Rmses
(blue)\n Underlying model uses log(budget) to predict log(revenue)\n cor
r(logBudget,logRevenue)=0.48; pValue is 0')
```

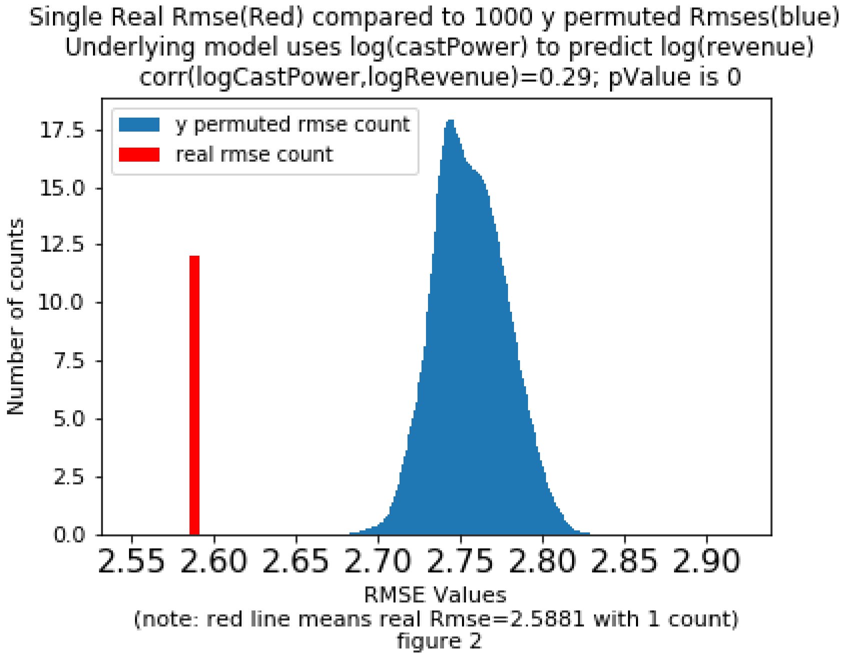


Corr = 0.48 and pvalue = 0 ; real rmse = 2.3157

In [106]:

```
#task8.2 log(castPower) vs log(revenue)
train1v['logCastPower']=np.log(train1v['castPower'])
train4v = train1v[:2398]
test4v = train1v[2398:]
X=train4v[['logCastPower']]
y=train4v['logRevenue']
Z=test4v[['logCastPower']]
reg = LinearRegression().fit(X,y)
y1=y.sample(frac=1).reset_index(drop=True)
result1=reg.predict(Z)
realRmse=sqrt(mean((result1-test4v['logRevenue'])**2))
permRmse=[]
for i in range(1000):
    #set_seed(i)
    y1=y.sample(frac=1,random_state=i).reset_index(drop=True)
    reg = LinearRegression().fit(X,y1)
    result2=reg.predict(Z)
    permRmse.append(sqrt(mean((result2-test4v['logRevenue'])**2)))
import scipy
count=0
for i in range(len(permRmse)):
    if permRmse[i]<realRmse:
        count+=1
PVALUE2=count/len(permRmse)
from scipy.stats import gaussian_kde
density = gaussian_kde(permRmse)
x = np.arange(2.55, 2.92, 0.001)
plt.bar(x, density(x),width=0.001,label='y permuted rmse count')
plt.bar(realRmse,12,width=0.005,color='r',label='real rmse count')
plt.xlabel('RMSE Values\n(note: red line means real Rmse=2.5881 with 1 count)\n figu
re 2',fontsize=11)
plt.ylabel('Number of counts',fontsize=11)
plt.xticks(rotation=0,fontsize=16)
plt.yticks(fontsize=11)
plt.legend()
plt.title('Single Real Rmse(Red) compared to 1000 y permuted Rmses(blue)\n Underlyin
g model uses log(castPower) to predict log(revenue)\n corr(logCastPower,logRevenue)=
0.29; pValue is 0',fontsize=12)
```

```
Out[106]:

<BarContainer object of 371 artists>

Out[106]:

<BarContainer object of 1 artists>

Out[106]:

Text(0.5, 0, 'RMSE Values\n(note: red line means real Rmse=2.5881 with 1
count)\n figure 2')

Out[106]:

Text(0, 0.5, 'Number of counts')

Out[106]:

(array([2.5 , 2.55, 2.6 , 2.65, 2.7 , 2.75, 2.8 , 2.85, 2.9 , 2.95]),
 <a list of 10 Text xticklabel objects>)

Out[106]:

(array([ 0. ,  2.5,  5. ,  7.5, 10. , 12.5, 15. , 17.5, 20. ]),
 <a list of 9 Text yticklabel objects>)


Out[106]:

<matplotlib.legend.Legend at 0x218f66fb0c8>

Out[106]:

Text(0.5, 1.0, 'Single Real Rmse(Red) compared to 1000 y permuted Rmses
(blue)\n Underlying model uses log(castPower) to predict log(revenue)\n
corr(logCastPower,logRevenue)=0.29; pValue is 0')
```



Corr = 0.29 and pvalue = 0; real rmse =2.5881

In [107]:

```
#8.3
#task8.2 log(numVotes) vs log(revenue)
train1v['logRelease_year']=pd.to_numeric(train1v['release_year'], errors='coerce')
train4v = train1v[:2398]
test4v = train1v[2398:]
X=train4v[['logRelease_year']]
y=train4v['logRevenue']
Z=test4v[['logRelease_year']]
#Z.size
reg = LinearRegression().fit(X,y)
y1=y.sample(frac=1).reset_index(drop=True)
result1=reg.predict(Z)
realRmse=sqrt(mean((result1-test4v['logRevenue'])**2))
permRmse=[]
for i in range(1000):
    #set_seed(i)
    y1=y.sample(frac=1,random_state=i).reset_index(drop=True)
    reg = LinearRegression().fit(X,y1)
    result2=reg.predict(Z)
    permRmse.append(sqrt(mean((result2-test4v['logRevenue'])**2)))
import scipy
count=0
for i in range(len(permRmse)):
    if permRmse[i]<realRmse:
        count+=1
PVALUE3=count/len(permRmse)
from scipy.stats import gaussian_kde
density = gaussian_kde(permRmse)
x = np.arange(2.735, 2.785, 0.001)
plt.bar(x, density(x),width=0.001,label='y permuted rmse count')
plt.bar(realRmse,80,width=0.001,color='r',label='real rmse count')
plt.xlabel('RMSE Values\n(note: red line means real Rmse=2.7503 with 1 count)\n figu
re 3',fontsize=11)
plt.ylabel('Number of counts',fontsize=11)
plt.xticks(rotation=0,fontsize=16)
plt.yticks(fontsize=11)
plt.legend()
plt.title('Single Real Rmse(Red) compared to 1000 y permuted Rmses(blue)\n Underlyin
g model uses log(release_year) to predict log(revenue)\n corr(logRelease_year,logRev
enue)=0.038; pValue=0.072',fontsize=12)
```

Out[107]:

```
<BarContainer object of 51 artists>
```

Out[107]:

```
<BarContainer object of 1 artists>
```

Out[107]:

```
Text(0.5, 0, 'RMSE Values\n(note: red line means real Rmse=2.7503 with 1
count)\n figure 3')
```

Out[107]:

```
Text(0, 0.5, 'Number of counts')
```

Out[107]:

```
(array([2.73, 2.74, 2.75, 2.76, 2.77, 2.78, 2.79]),
 <a list of 7 Text xticklabel objects>)
```

Out[107]:

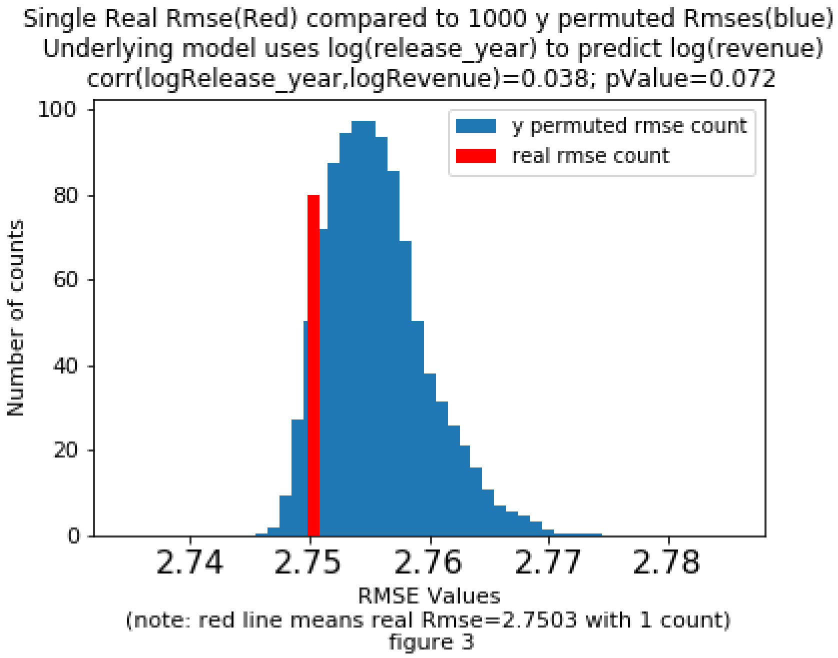
```
(array([  0.,  20.,  40.,  60.,  80., 100., 120.]),
 <a list of 7 Text yticklabel objects>)
```

Out[107]:

```
<matplotlib.legend.Legend at 0x218f68d1e88>
```

Out[107]:

```
Text(0.5, 1.0, 'Single Real Rmse(Red) compared to 1000 y permuted Rmses
(blue)\n Underlying model uses log(release_year) to predict log(revenue)
\n corr(logRelease_year,logRevenue)=0.038; pValue=0.072')
```



Corr = 0.038 ; pvalue = 0.072 ; real_rmse = 2.7503

Part 9 - Predicton

In [115]:

```
# TODO: code for your prediction models
train1v['runtime'][1334]=np.nanmean(train1v['runtime'])
train1v['runtime'][2300]=np.nanmean(train1v['runtime'])
train1v['runtime'][train1v['runtime'].isnull()]
train1v['numVotes'][355]=np.nanmean(train1v['numVotes'])
train1v['numVotes'][355]=np.nanmean(train1v['numVotes'])
train1v['averageRating'][355]=np.nanmean(train1v['averageRating'])
list6=Counter(test['original_language'])
language6=dict(list6)
test['original_language1']=[list() for x in range(len(test['original_language']))]
for i in range(len(test['original_language'])):
    test['original_language1'][i]=int(language6[test['original_language'][i]])#trai
n.belongs_to_collection4
#need to delete this row below because the budget is very high and the revenue is to
o low only $100
test['original_language2']=pd.to_numeric(test['original_language1'])
test['release_date'][828]='5/15/00'
test['release_date1']=test['release_date'].apply(lambda x: datetime.strptime(x,'%m/%
d/%y'))
test['release_date2']=''
for i in range(0,len(test['release_date1'])):
    if test['release_date1'][i].year >=2020:
        year = test['release_date1'][i].year-100
    else:
        year=test['release_date1'][i].year
    test['release_date2'][i]=date(year,test['release_date1'][i].month,test['release_
date1'][i].day)
test['release_year']=[list() for x in range(len(test['release_date2']))]
for i in range(0,len(test['release_date2'])):
    test['release_year'][i]=test['release_date2'][i].year
test['release_year']=pd.to_numeric(test['release_year'])
test['cast1']=test['cast'].replace(np.NaN,'[]',inplace=False)
test['cast2']=test['cast1'].apply(literal_eval)
list4=[]
test['cast3']=[list() for x in range(len(test['cast']))]
for i in range(len(test['cast2'])):
    for j in range(len(test['cast2'][i])):
        list4.append(str(test['cast2'][i][j]['id']))
        test['cast3'][i].append(str(test['cast2'][i][j]['id']))
star=Counter(list4)
star1=dict(star)
test['cast4']=test['cast3']
for i in range(len(test['cast3'])):
    for j in range(len(test['cast3'][i])):
        test['cast3'][i][j]=star1[str(test['cast3'][i][j])]
test['castPower']=test['cast4'].apply(lambda x: 0 if x == [] else mean(x))
a=mean(test['castPower'])*2997/(2997-26)
for i in range(len(test['castPower'])):
    if test['castPower'][i]==0:
        test['castPower'][i]=a
test['castPower5']=pd.to_numeric(test['castPower'])
test['logCastPower1']=np.log(test['castPower5'])
test1v=pd.merge(test, external2, how='left',left_on='imdb_id', right_on='tconst')
test1v['release_year1']=test1v['release_year']
test1v['averageRating'][713]=np.nanmean(test1v['averageRating'])
test1v['averageRating'][1975]=np.nanmean(test1v['averageRating'])
test1v['numVotes'][713]=np.nanmean(test1v['numVotes'])
test1v['numVotes'][1975]=np.nanmean(test1v['numVotes'])
```

```
test1v['runtime'][243]=np.nanmean(test1v['runtime'])
test1v['runtime'][1489]=np.nanmean(test1v['runtime'])
test1v['runtime'][1632]=np.nanmean(test1v['runtime'])
test1v['runtime'][3817]=np.nanmean(test1v['runtime'])
test1v['runtime'][test1v['runtime'].isnull()]
X=train1v[['release_year1','original_language2','popularity','budget','logCastPower1','averageRating','numVotes','runtime']]
y=train1v['revenue']
Z=test1v[['release_year1','original_language2','popularity','budget','logCastPower1','averageRating','numVotes','runtime']]
reg = LinearRegression().fit(X,y)
result1=reg.predict(Z)
#result1
result2=[list() for x in range(len(result1))]
for i in range(len(result1)):
    if result1[i]<0:
        result2[i]=0
    else:
        result2[i]=result1[i]
sample_submission['revenue']=result2
sample_submission.to_csv('sample_submission13.csv')
```

```
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:2: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:3: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  This is separate from the ipykernel package so we can avoid doing impo
rts until
```

Out[115]:

```
Series([], Name: runtime, dtype: float64)
```

```
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:5: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:6: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:7: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  import sys
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:12: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  if sys.path[0] == '':
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:15: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  from ipykernel import kernelapp as app
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:23: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:26: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:46: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:51: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:52: Sett
ingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:53: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:54: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:55: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:56: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:57: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
C:\Users\user\anaconda3\lib\site-packages\ipykernel_launcher.py:58: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-d
ocs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[115]:

```
Series([], Name: runtime, dtype: float64)
```

I train linear regression model using the following features in the training data set. I fit the following features in test data set to the trained model in order to predict the revenue. ['release_year1','original_language2','popularity','budget','logCastPower1','averageRating','numVotes','runtim

Part 10 - Final Result

Report your highest score. Include a snapshot of your best score after submission as confirmation. Be sure to provide a link to your Kaggle profile. Make sure your profile includes your face and affiliation with SBU.

Kaggle Link: https://www.kaggle.com/yanian (https://www.kaggle.com/yanian)

Highest Score:2.99761

Number of entries: FILL HERE

3 entries

INCLUDE IMAGE OF YOUR HIGHEST SCORE imagelink: https://drive.google.com/file/d/1UaB92jIJ9fepWPQFTgr2TT0mdBEzBbvO/view?usp=sharing (https://drive.google.com/file/d/1UaB92jIJ9fepWPQFTgr2TT0mdBEzBbvO/view?usp=sharing)