analysis-with-plotly-and-python-2

October 12, 2023

Data Visualization and Analysis of Worldwide Box Office Revenue

0.0.1 (Part 1) Libraries

[1]:
```
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
import lightgbm as lgb
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 xgboost as xgb
import lightgbm as lgb
from sklearn import model_selection
from sklearn.metrics import accuracy_score
import json
import ast
from urllib.request import urlopen
from PIL import Image
```

```
from sklearn.preprocessing import LabelEncoder
import time
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn import linear_model
```

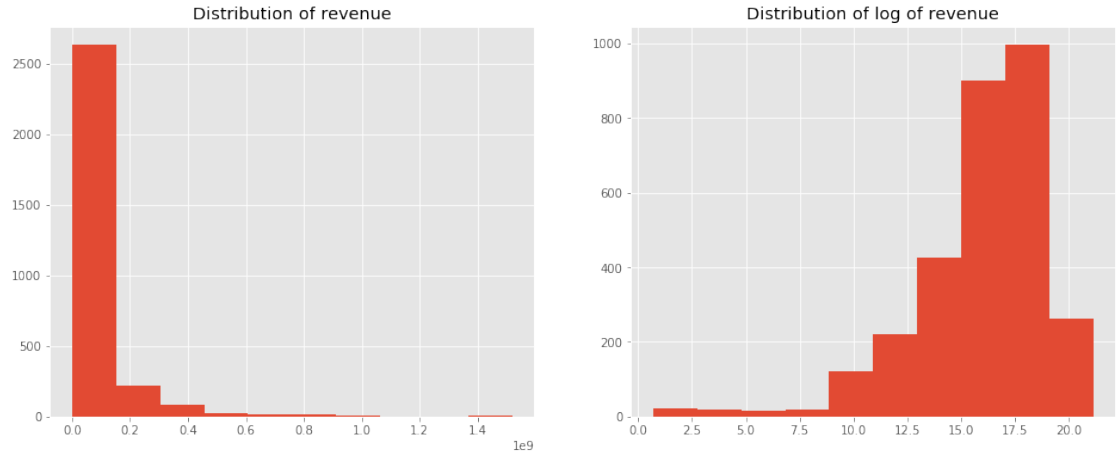
```
[nltk_data] Downloading package stopwords to /home/cicada/nltk_data…

[nltk_data]   Package stopwords is already up-to-date!
```

0.0.2 (Part 1) Data Loading and Exploration

[2]:
```
train = pd.read_csv('data/train.csv')
test = pd.read_csv('data/test.csv')
```

0.0.3 (Part 1) Visualizing the Target Distribution

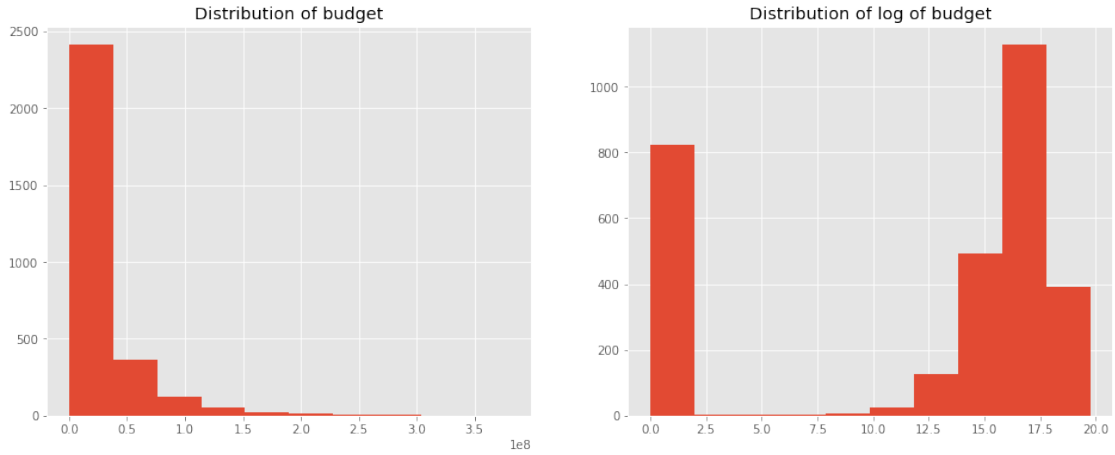
[4]:
```
fig, ax = plt.subplots(figsize = (16, 6))
plt.subplot(1, 2, 1)
plt.hist(train['revenue']);
plt.title('Distribution of revenue');
plt.subplot(1, 2, 2)
plt.hist(np.log1p(train['revenue']));
plt.title('Distribution of log of revenue');
```



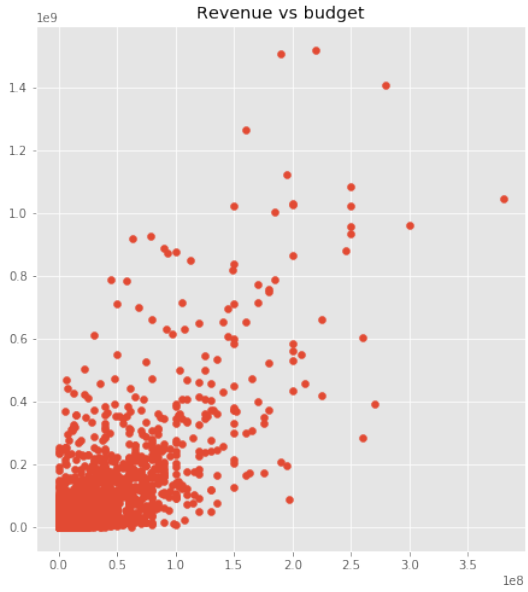
[5]:
```
train['log_revenue'] = np.log1p(train['revenue'])
```

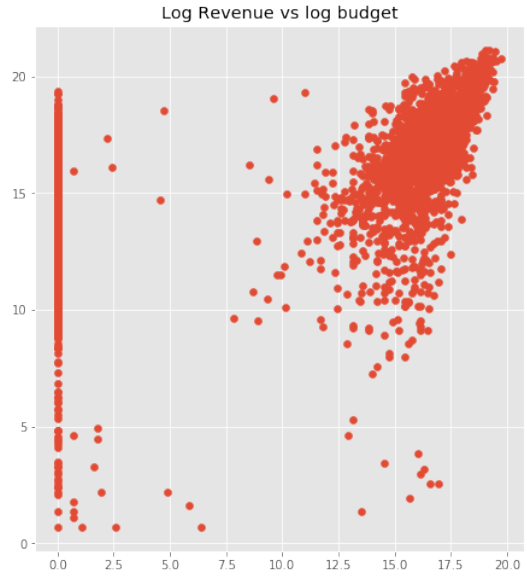
0.0.4 (Part 1) Relationship between Film Revenue and Budget

[6]:
```
fig, ax = plt.subplots(figsize = (16, 6))
plt.subplot(1, 2, 1)
plt.hist(train['budget']);
plt.title('Distribution of budget');
plt.subplot(1, 2, 2)
plt.hist(np.log1p(train['budget']));
plt.title('Distribution of log of budget');
```



[7]:
```
plt.figure(figsize=(16, 8))
plt.subplot(1, 2, 1)
plt.scatter(train['budget'], train['revenue'])
plt.title('Revenue vs budget');
plt.subplot(1, 2, 2)
plt.scatter(np.log1p(train['budget']), train['log_revenue'])
plt.title('Log Revenue vs log budget');
```





[8]:
```
train['log_budget'] = np.log1p(train['budget'])
test['log_budget'] = np.log1p(test['budget'])
```

0.0.5 (Part 1) Does having an Official Homepage Affect Revenue?

[9]:
```
train['homepage'].value_counts().head(10)
```

[9]:
```
http://www.transformersmovie.com/
4
http://www.lordoftherings.net/
2
http://www.thehobbit.com/
2
http://www.universalstudiosentertainment.com/lock-stock-and-two-smoking-barrels/
1
http://www.worldwarzmovie.com
1
http://www.sonypictures.com/movies/residentevil/index.html
1
http://www.filminfocus.com/focusfeatures/film/sin_nombre
1
http://www.dineshdsouza.com/movies/hillarys-america/
1
http://todolistmovie.com/
1
http://www.everybodyloveswhales.com/
1
```

```
Name: homepage, dtype: int64
```

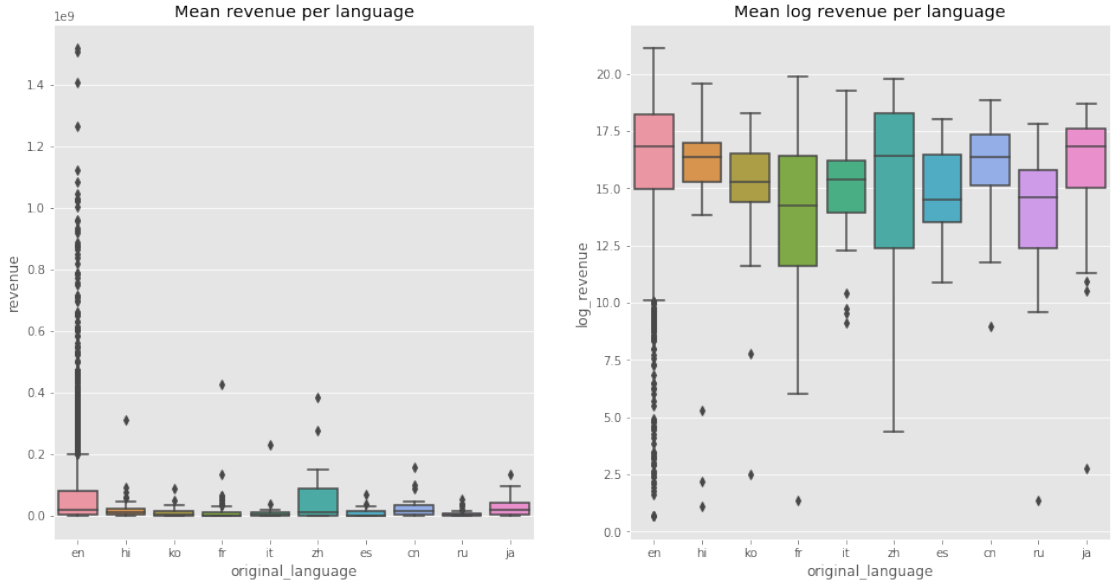
[10]:
```
train['has_homepage'] = 0
train.loc[train['homepage'].isnull() == False, 'has_homepage'] = 1
test['has_homepage'] = 0
test.loc[test['homepage'].isnull() == False, 'has_homepage'] = 1
```

[11]:
```
sns.catplot(x='has_homepage', y='revenue', data=train);
plt.title('Revenue for film with and without homepage');
```

0.0.6 (Part 1) Distribution of Languages in Film

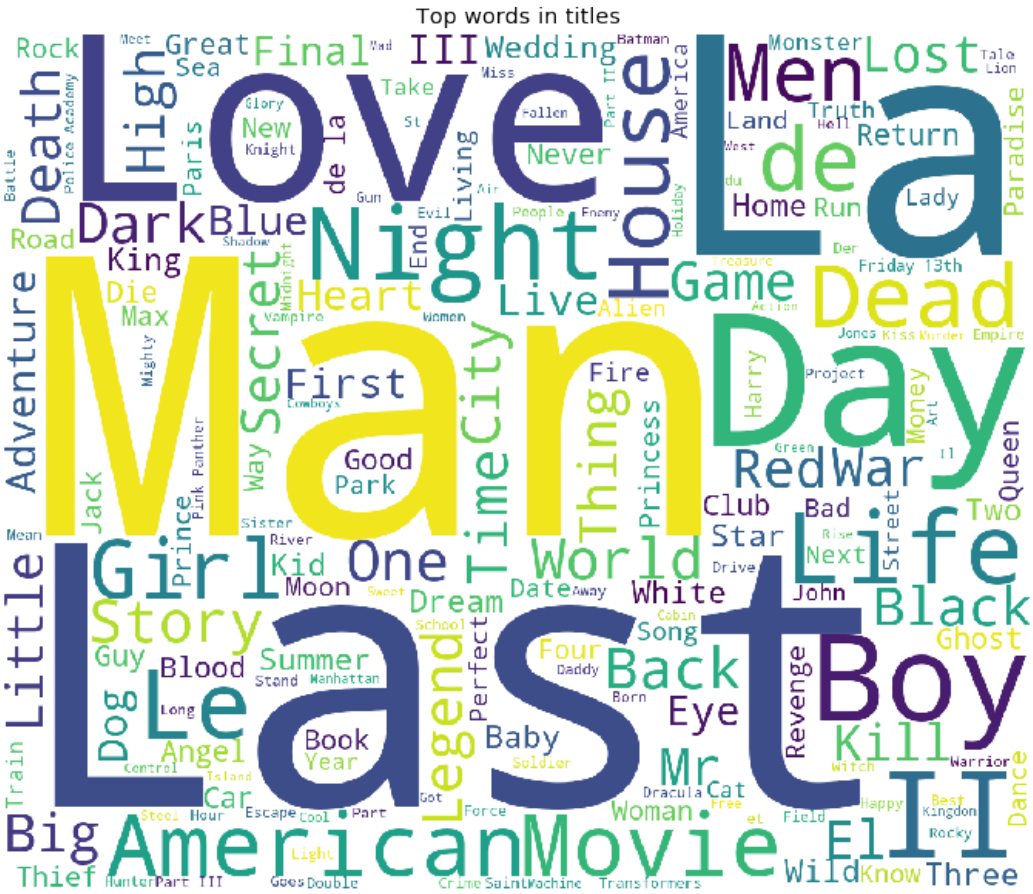
[12]:
```
plt.figure(figsize=(16, 8))
plt.subplot(1, 2, 1)
sns.boxplot(x='original_language', y='revenue', data=train.
 ↪loc[train['original_language'].isin(train['original_language'].
 ↪value_counts().head(10).index)]);
```

```
plt.title('Mean revenue per language');
plt.subplot(1, 2, 2)
sns.boxplot(x='original_language', y='log_revenue', data=train.
 ↪loc[train['original_language'].isin(train['original_language'].
 ↪value_counts().head(10).index)]);
plt.title('Mean log revenue per language');
```

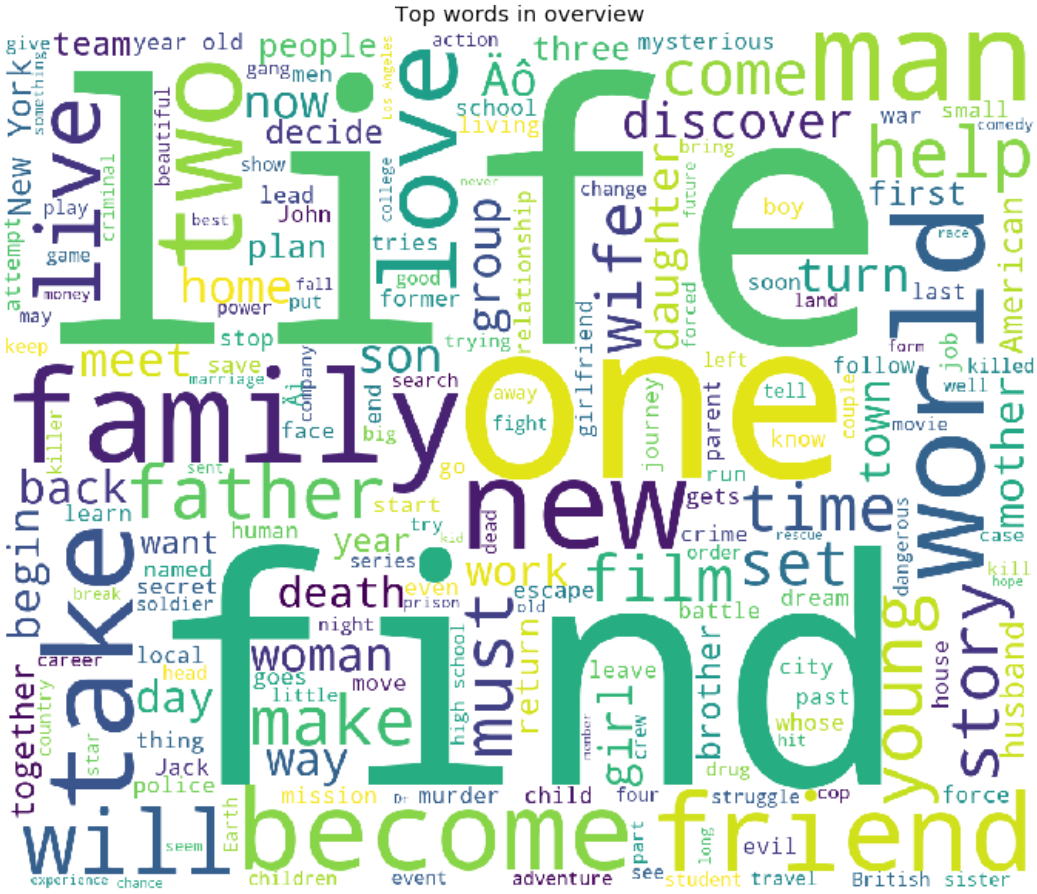


0.0.7 (Part 1) Frequent Words in Film Titles and Discriptions

[13]:
```
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 words in titles')
plt.axis("off")
plt.show()
```



[14]:
```
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 words in overview')
plt.axis("off")
plt.show()
```



0.0.8 (Part 1) Do Film Descriptions Impact Revenue?

[ ]:
```
import eli5

vectorizer = TfidfVectorizer(
            sublinear_tf=True,
            analyzer='word',
            token_pattern=r'\w{1,}',
            ngram_range=(1, 2),
            min_df=5)

overview_text = vectorizer.fit_transform(train['overview'].fillna(''))
linreg = LinearRegression()
linreg.fit(overview_text, train['log_revenue'])
eli5.show_weights(linreg, vec=vectorizer, top=20, feature_filter=lambda x: x !=␣
 ↪'<BIAS>')
```

[ ]: <IPython.core.display.HTML object>

[ ]:
```
print('Target value:', train['log_revenue'][1000])
eli5.show_prediction(linreg, doc=train['overview'].values[1000], vec=vectorizer)
```

```
Target value: 16.44583954907521
```

[ ]:
```
<IPython.core.display.HTML object>
```

0.0.9 Task 1: Analyzing Movie Release Dates

[17]:
```
test.loc[test['release_date'].isnull() == False, 'release_date'].head()
```

[17]:
```
0    7/14/07
1    5/19/58
2    5/23/97
3     9/4/10
4    2/11/05
Name: release_date, dtype: object
```

[ ]:
```

```

0.0.10 Task 2: Preprocessing Features

[18]:
```
def fix_date(x):
    year = x.split('/')[2]
    if int(year) <= 19:
        return x[:-2] + '20' + year
    else:
        return x[:-2] + '19' + year
```

[19]:
```
test.loc[test['release_date'].isnull() == True].head()
```

[19]:
```
       id  budget homepage    imdb_id original_language  \
828  3829       0      NaN  tt0210130                en

                 original_title  \
828  Jails, Hospitals & Hip-Hop

                                              overview  popularity  \
828  Jails, Hospitals &amp;amp; Hip-Hop is a cinematic …    0.009057

    poster_path release_date  runtime status  \
828         NaN          NaN     90.0    NaN

                                          tagline  \
828  three worlds / two million voices / one genera…

                         title Keywords collection_name  has_collection  \
```

```
828  Jails, Hospitals & Hip-Hop       {}               0               0

     num_genres all_genres  genre_Drama  genre_Comedy  genre_Thriller  \
828          1      Drama             1             0               0

     genre_Action  genre_Romance  genre_Crime  genre_Adventure  genre_Horror  \
828             0              0            0                0             0

     genre_Science Fiction  genre_Family  genre_Fantasy  genre_Mystery  \
828                      0             0              0              0

     genre_Animation  genre_History  genre_Music  num_companies  \
828                0              0            0              0

     production_company_Warner Bros.  production_company_Universal Pictures  \
828                                 0                                        0

     production_company_Paramount Pictures  \
828                                      0

     production_company_Twentieth Century Fox Film Corporation  \
828                                                          0

     production_company_Columbia Pictures  \
828                                      0

     production_company_Metro-Goldwyn-Mayer (MGM)  \
828                                              0

     production_company_New Line Cinema  \
828                                    0

     production_company_Touchstone Pictures  \
828                                       0

     production_company_Walt Disney Pictures  \
828                                        0

     production_company_Columbia Pictures Corporation  \
828                                                  0

     production_company_TriStar Pictures  production_company_Relativity Media  \
828                                    0                                       0

     production_company_Canal+  production_company_United Artists  \
828                          0                                    0
```

```
     production_company_Miramax Films  \
828                                  0

     production_company_Village Roadshow Pictures  \
828                                             0

     production_company_Regency Enterprises  production_company_BBC Films  \
828                                       0                               0

     production_company_Dune Entertainment  \
828                                       0

     production_company_Working Title Films  \
828                                       0

     production_company_Fox Searchlight Pictures  \
828                                             0

     production_company_StudioCanal  production_company_Lionsgate  \
828                                0                             0

     production_company_DreamWorks SKG  production_company_Fox 2000 Pictures  \
828                                  0                                       0

     production_company_Summit Entertainment  \
828                                       0

     production_company_Hollywood Pictures  production_company_Orion Pictures  \
828                                       0                                   0

     production_company_Amblin Entertainment  \
828                                       0

     production_company_Dimension Films  num_countries  \
828                                   0              0

     production_country_United States of America  \
828                                             0

     production_country_United Kingdom  production_country_France  \
828                                  0                           0

     production_country_Germany  production_country_Canada  \
828                           0                           0

     production_country_India  production_country_Italy  \
828                         0                         0
```

```
     production_country_Japan  production_country_Australia  \
828                         0                             0

     production_country_Russia  production_country_Spain  \
828                         0                         0

     production_country_China  production_country_Hong Kong  \
828                         0                             0

     production_country_Ireland  production_country_Belgium  \
828                          0                             0

     production_country_South Korea  production_country_Mexico  \
828                               0                          0

     production_country_Sweden  production_country_New Zealand  \
828                         0                                0

     production_country_Netherlands  production_country_Czech Republic  \
828                               0                                    0

     production_country_Denmark  production_country_Brazil  \
828                          0                          0

     production_country_Luxembourg  production_country_South Africa  \
828                             0                                 0

     num_languages  language_English  language_Français  language_Español  \
828              0                0                  0                 0

     language_Deutsch  language_P     language_Italiano  language_    \
828                 0                0                  0            0

     language_    language_     language_  language_Português  \
828            0               0          1                   0

     language_      language_ /    language_   /     language_     \
828                 0                 0                   0              0

     language_Polski  language_Magyar  language_Latin  language_svenska  \
828                0                0               0                 0

     language_      language_Český  language_     language_      \
828                 0               0                  0                 0

     language_Türkçe  language_Dansk  language_Nederlands  language_     \
```

```
828               0              0                    0              0

     language_Tiếng Việt  language_    language_Română  num_cast  \
828                    0             0                0         0

     cast_name_Samuel L. Jackson  cast_name_Robert De Niro  \
828                            0                          0

     cast_name_Morgan Freeman  cast_name_J.K. Simmons  cast_name_Bruce Willis  \
828                         0                       0                       0

     cast_name_Liam Neeson  cast_name_Susan Sarandon  cast_name_Bruce McGill  \
828                      0                         0                       0

     cast_name_John Turturro  cast_name_Forest Whitaker  \
828                        0                          0

     cast_name_Willem Dafoe  cast_name_Bill Murray  cast_name_Owen Wilson  \
828                      0                      0                      0

     cast_name_Nicolas Cage  cast_name_Sylvester Stallone  genders_0_cast  \
828                      0                              0               0

     genders_1_cast  genders_2_cast  cast_character_  cast_character_Himself  \
828              0               0                1                       0

     cast_character_Herself  cast_character_Dancer  \
828                       0                      0

     cast_character_Additional Voices (voice)  cast_character_Doctor  \
828                                         0                      0

     cast_character_Reporter  cast_character_Waitress  cast_character_Nurse  \
828                       0                       0                     0

     cast_character_Bartender  cast_character_Jack  cast_character_Debutante  \
828                        0                    0                         0

     cast_character_Security Guard  cast_character_Paul  cast_character_Frank  \
828                             0                    0                      0

     num_crew  crew_name_Avy Kaufman  crew_name_Robert Rodriguez  \
828        0                      0                           0

     crew_name_Deborah Aquila  crew_name_James Newton Howard  \
828                        0                               0
```

```
     crew_name_Mary Vernieu  crew_name_Steven Spielberg  crew_name_Luc Besson  \
828                       0                           0                     0

     crew_name_Jerry Goldsmith  crew_name_Francine Maisler  \
828                          0                           0

     crew_name_Tricia Wood  crew_name_James Horner  crew_name_Kerry Barden  \
828                      0                       0                       0

     crew_name_Bob Weinstein  crew_name_Harvey Weinstein  \
828                        0                           0

     crew_name_Janet Hirshenson  genders_0_crew  genders_1_crew  \
828                           0              0               0

     genders_2_crew  jobs_Producer  jobs_Executive Producer  jobs_Director  \
828               0              0                        0              0

     jobs_Screenplay  jobs_Editor  jobs_Casting  jobs_Director of Photography  \
828                0            0             0                             0

     jobs_Original Music Composer  jobs_Art Direction  jobs_Production Design  \
828                             0                   0                       0

     jobs_Costume Design  jobs_Writer  jobs_Set Decoration  \
828                    0            0                    0

     jobs_Makeup Artist  jobs_Sound Re-Recording Mixer  \
828                   0                              0

     departments_Production  departments_Sound  departments_Art  \
828                       0                  0                0

     departments_Crew  departments_Writing  departments_Costume & Make-Up  \
828                 0                    0                              0

     departments_Camera  departments_Directing  departments_Editing  \
828                   0                     0                    0

     departments_Visual Effects  departments_Lighting  departments_Actors  \
828                           0                     0                   0

     log_budget  has_homepage
828         0.0             0
```

[20]:
```
test.loc[test['release_date'].isnull() == True, 'release_date'] = '05/01/00'
```

[21]:
```
train['release_date'] = train['release_date'].apply(lambda x: fix_date(x))
test['release_date'] = test['release_date'].apply(lambda x: fix_date(x))
```

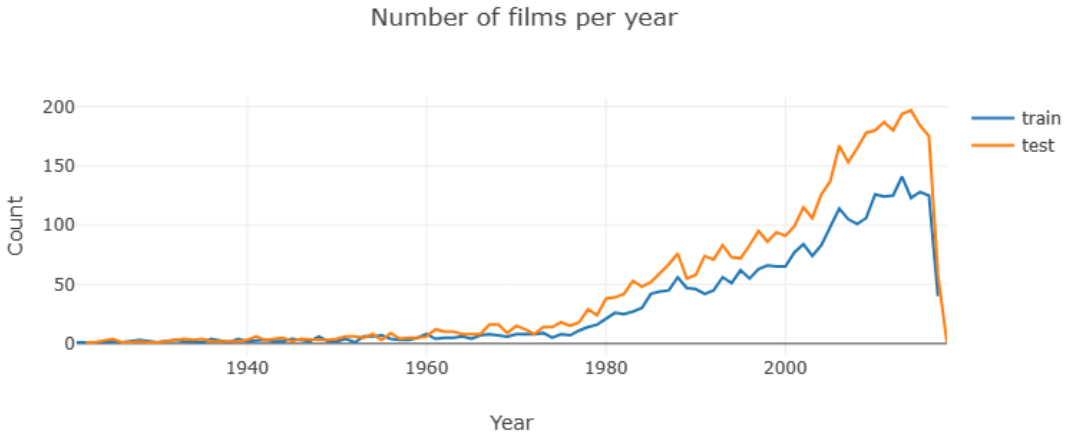
0.0.11 Task 3: Creating Features Based on Release Date

[22]:
```
train['release_date'] = pd.to_datetime(train['release_date'])
test['release_date'] = pd.to_datetime(test['release_date'])
```

[23]:
```
def process_date(df):
    date_parts = ["year", "weekday", "month", 'weekofyear', 'day', 'quarter']
    for part in date_parts:
        part_col = 'release_date' + "_" + part
        df[part_col] = getattr(df['release_date'].dt, part).astype(int)

    return df

train = process_date(train)
test = process_date(test)
```

0.0.12 Task 4: Using Plotly to Visualize the Number of Films Per Year

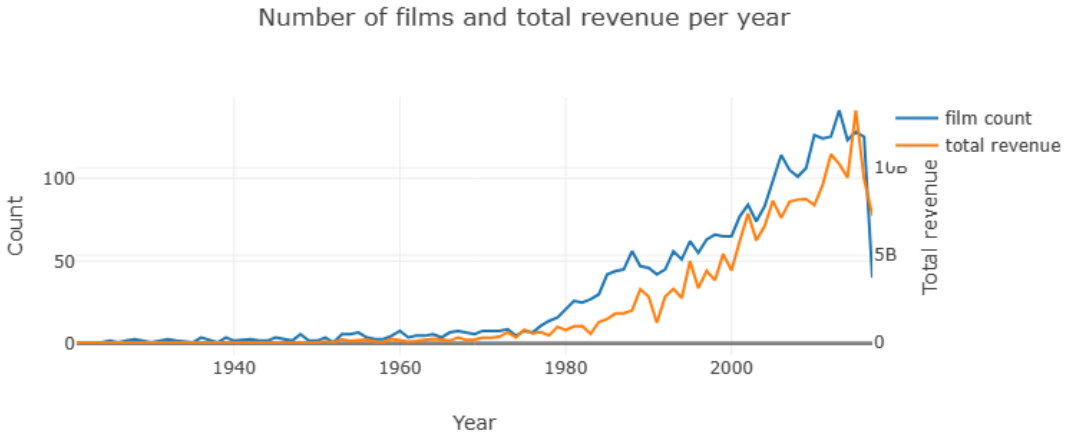
[24]:
```
# Count no.of films released per year and sort the years in ascending order
# Do this for both Train and Test Sets
d1 = train['release_date_year'].value_counts().sort_index()
d2 = test['release_date_year'].value_counts().sort_index()

import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go

# x values are years, and y values are movie counts, name=legend
data = [go.Scatter(x=d1.index, y=d1.values, name='train'),
        go.Scatter(x=d2.index, y=d2.values, name='test')]

layout = go.Layout(dict(title = "Number of films per year",
                  xaxis = dict(title = 'Year'),
                  yaxis = dict(title = 'Count'),
                  ),legend=dict(
                orientation="v"))
py.iplot(dict(data=data, layout=layout))
```



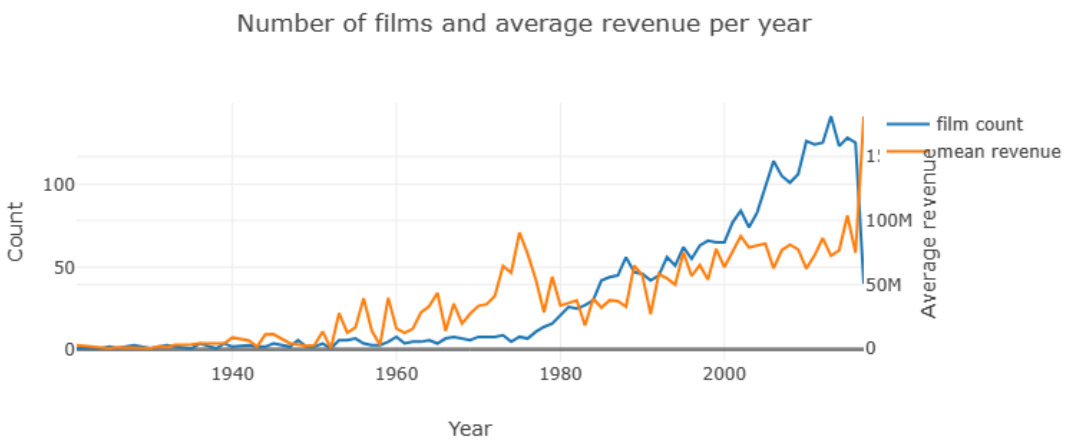
[ ]:

0.0.13 Task 5: Number of Films and Revenue Per Year

[25]:
```
d1 = train['release_date_year'].value_counts().sort_index()
d2 = train.groupby(['release_date_year'])['revenue'].sum()

data = [go.Scatter(x=d1.index, y=d1.values, name='film count'),
        go.Scatter(x=d2.index, y=d2.values, name='total revenue', yaxis='y2')]

layout = go.Layout(dict(title = "Number of films and total revenue per year",
                        xaxis = dict(title = 'Year'),
                        yaxis = dict(title = 'Count'),
                        yaxis2=dict(title='Total revenue', overlaying='y',␣
 ↪side='right')),
                   legend=dict(orientation="v"))

py.iplot(dict(data=data, layout=layout))
```

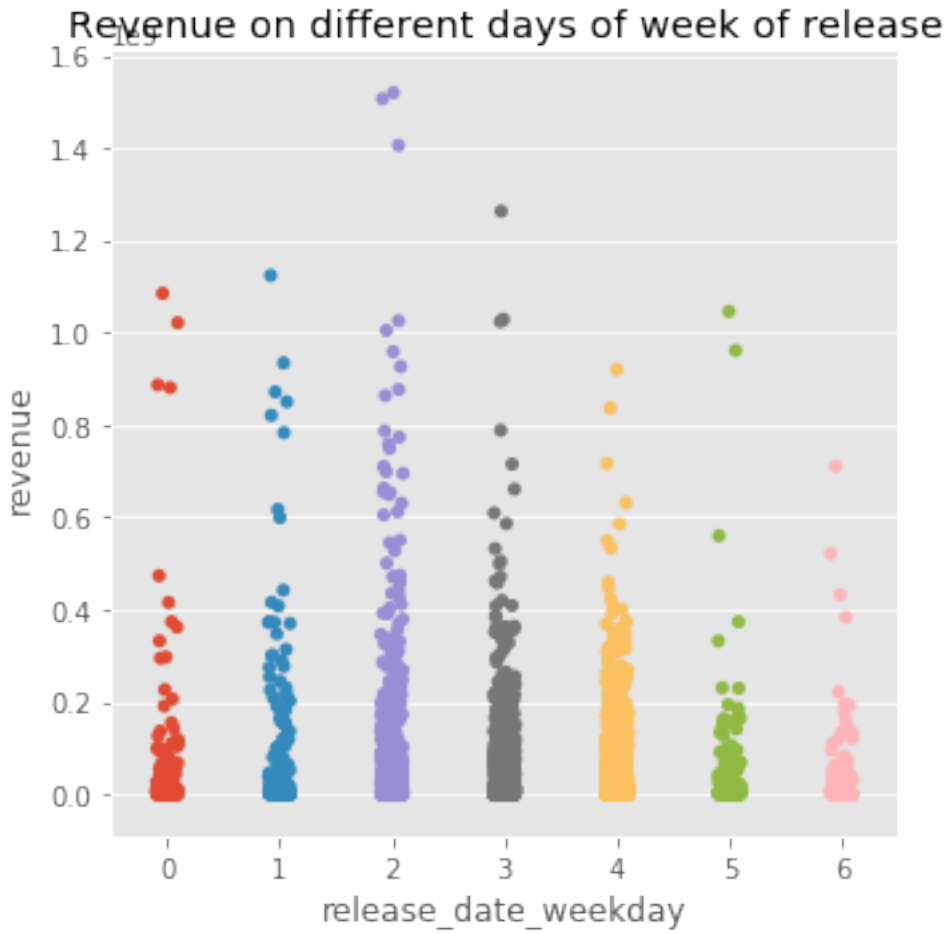


[26]:
```
d1 = train['release_date_year'].value_counts().sort_index()
d2 = train.groupby(['release_date_year'])['revenue'].mean()

data = [go.Scatter(x=d1.index, y=d1.values, name='film count'),
        go.Scatter(x=d2.index, y=d2.values, name='mean revenue', yaxis='y2')]

layout = go.Layout(dict(title = "Number of films and average revenue per year",
                  xaxis = dict(title = 'Year'),
                  yaxis = dict(title = 'Count'),
                  yaxis2=dict(title='Average revenue', overlaying='y',␣
 ↪side='right')
                  ),legend=dict(
                orientation="v"))
py.iplot(dict(data=data, layout=layout))
```



0.0.14 Task 6: Do Release Days Impact Revenue?

[27]:
```
sns.catplot(x='release_date_weekday', y='revenue', data=train);
plt.title('Revenue on different days of week of release');
```

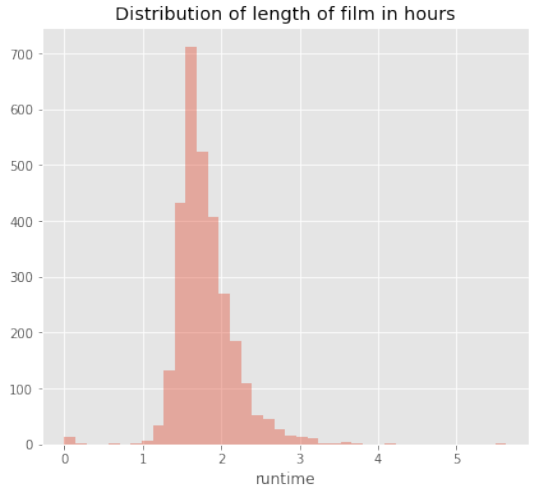


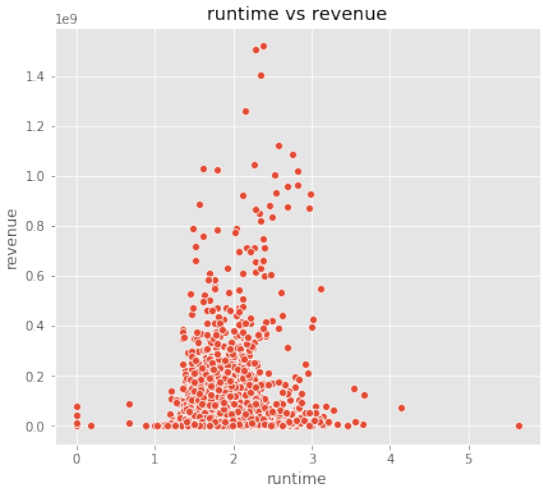
[ ]:
```

```

0.0.15 Task 7: Relationship between Runtime and Revenue

[28]:
```
plt.figure(figsize=(15, 6))
plt.subplot(1, 2, 1)
sns.distplot(train['runtime'].fillna(0) / 60, bins=40, kde=False);
plt.title('Distribution of length of film in hours');
plt.subplot(1, 2, 2)
```

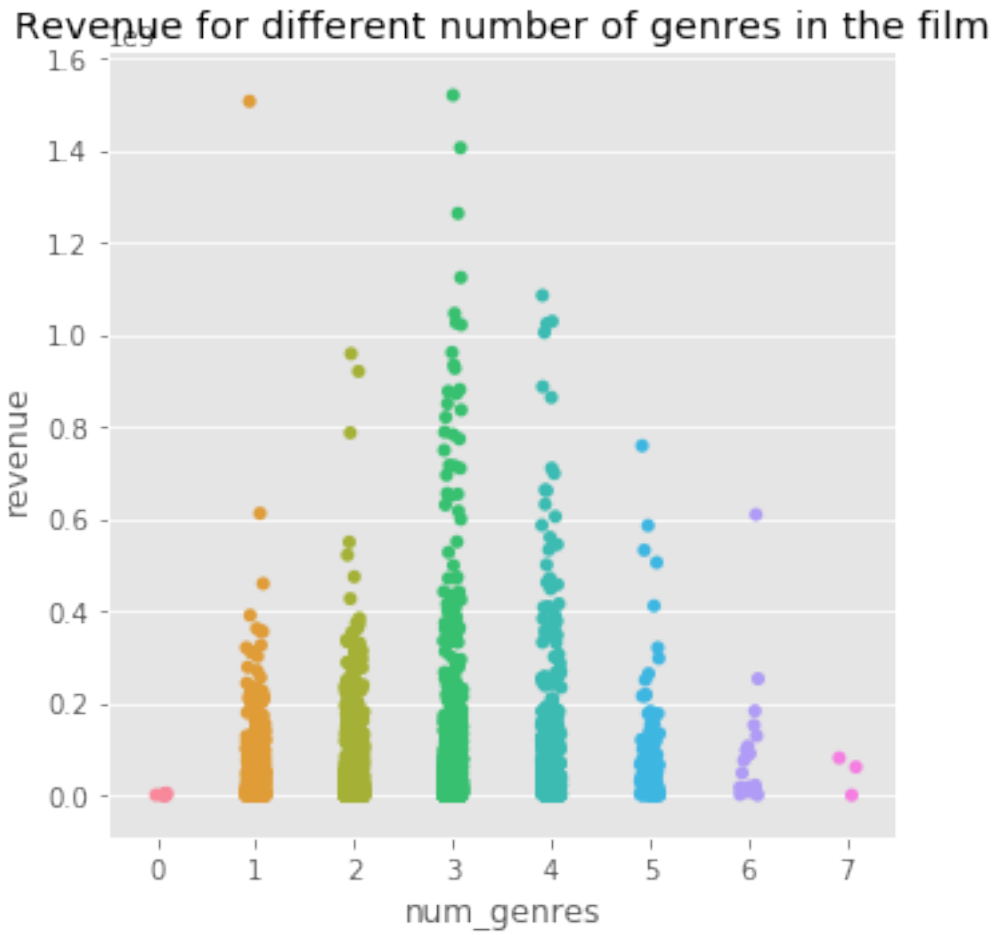
```
sns.scatterplot(train['runtime'].fillna(0)/60, train['revenue'])
plt.title('runtime vs revenue');
```





0.0.16 Task 8: Highest Grossing Genres

[29]:
```
sns.catplot(x='num_genres', y='revenue', data=train);
plt.title('Revenue for different number of genres in the film');
```


Revenue for different number of genres in the film
1e9
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
revenue
0
1
2
3
4
5
6
7
num_genres