Investigating-TMDB-Movies-Dataset

December 20, 2020

1 Project: Investigate a Dataset (TMDb Movies Dataset)

1.1 Table of Contents

Introduction

Data Wrangling

Exploratory Data Analysis

Conclusions

Resource

Introduction

In this section of the report, we will introduce the dataset

This dataset was generated from The Movie Database API. This product uses the TMDb API but is not endorsed or certified by TMDb

What can we say about the success of a movie before it is released? Are there certain companies (Pixar?) that have found a consistent formula? Given that major films costing over $100 million to produce can still flop, this question is more important than ever to the industry. Film aficionados might have different interests. Can we predict which films will be highly rated, whether or not they are a commercial success?

[1]:
```
# Use this cell to set up import statements for all of the packages that you
#   plan to use
# Remember to include a 'magic word' so that your visualizations are plotted
#   inline with the notebook. See this page for more:
#   http://ipython.readthedocs.io/en/stable/interactive/magics.html

import os              #this module routines for NT or Posix depending on what␣
 ↪system we're on.
import pandas as pd    # this module
import numpy as np     # library used for working with arrays. It also has␣
 ↪functions for working in domain of linear algebra
import types
import csv
import seaborn as sns         # provides a high-level interface for drawing␣
 ↪attractive and informative statistical graphics.
```

```
import time             # This module provides various functions to manipulate␣
 ↪time values.
import pandocfilters    # This Functions to aid writing python scripts that␣
 ↪process the pandoc AST serialized as JSON.
import nbconvert        # This module converting notebooks to and from different␣
 ↪formats
import pyppeteer        # Generate screenshots and PDFs of pages
import pip


# Matplotlib and associated plotting modules
import matplotlib.cm as cm
import matplotlib.colors as colors
import matplotlib.pyplot as plt
%matplotlib inline

print ("libarary-Imported")
```

```
libarary-Imported
```

[2]:
```
from IPython.display import HTML
from timeit import default_timer as timer
from pandas.core.tools.datetimes import to_datetime
from pandas.core.tools.timedeltas import to_timedelta
from nbconvert import LatexExporter
from nbconvert import PDFExporter
from nbconvert import webpdf
from nbconvert import nbconvertapp
from nbconvert import templates

print ("functions-Imported")
```

```
functions-Imported
```

[3]:
```
import os
os.getcwd()
```

[3]:
```
'C:\\Users\\Abdelrazek\\Downloads\\investigate-tmdb-movies-dataset'
```

Data Wrangling

1.1.1 General Properties

[4]:
```
# Load your data and print out a few lines. Perform operations to inspect data
#   types and look for instances of missing or possibly errant data.
#loading the csv file and storing it in 'df'
df = pd.read_csv('tmdb-movies.csv')
```

[5]:
```
# printing first two rows
df.head(2)
```

[5]:
```
       id    imdb_id  popularity     budget     revenue      original_title  \
0  135397  tt0369610   32.985763  150000000  1513528810      Jurassic World
1   76341  tt1392190   28.419936  150000000   378436354  Mad Max: Fury Road

                                                cast  \
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…
1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic…

                        homepage         director             tagline  …  \
0  http://www.jurassicworld.com/  Colin Trevorrow   The park is open.  …
1    http://www.madmaxmovie.com/    George Miller  What a Lovely Day.  …

                                            overview runtime  \
0  Twenty-two years after the events of Jurassic …     124
1  An apocalyptic story set in the furthest reach…     120

                                      genres  \
0  Action|Adventure|Science Fiction|Thriller
1  Action|Adventure|Science Fiction|Thriller

                                production_companies release_date vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda…       6/9/15       5562
1  Village Roadshow Pictures|Kennedy Miller Produ…      5/13/15       6185

   vote_average  release_year    budget_adj   revenue_adj
0           6.5          2015  1.379999e+08  1.392446e+09
1           7.1          2015  1.379999e+08  3.481613e+08

[2 rows x 21 columns]
```

[6]:
```
# printing the last two rows
df.tail(2)
```

[6]:
```
          id    imdb_id  popularity  budget  revenue  \
10864  21449  tt0061177    0.064317       0        0
10865  22293  tt0060666    0.035919   19000        0

                original_title  \
10864    What's Up, Tiger Lily?
10865  Manos: The Hands of Fate

                                                cast homepage  \
10864  Tatsuya Mihashi|Akiko Wakabayashi|Mie Hama|Joh…      NaN
10865  Harold P. Warren|Tom Neyman|John Reynolds|Dian…      NaN
```

```
              director                                        tagline  …  \
10864       Woody Allen                    WOODY ALLEN STRIKES BACK!  …
10865  Harold P. Warren  It's Shocking! It's Beyond Your Imagination!  …

                                                 overview runtime  \
10864  In comic Woody Allen's film debut, he took the…      80
10865  A family gets lost on the road and stumbles up…      74

              genres     production_companies release_date vote_count  \
10864  Action|Comedy  Benedict Pictures Corp.      11/2/66         22
10865         Horror                 Norm-Iris     11/15/66         15

       vote_average  release_year     budget_adj  revenue_adj
10864           5.4          1966       0.000000          0.0
10865           1.5          1966  127642.279154          0.0

[2 rows x 21 columns]
```

[7]:
```
# The dimensions of the dataset
df.shape
```

[7]:
```
(10866, 21)
```

[8]:
```
# display a summary of the dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    10866 non-null  int64
 1   imdb_id               10856 non-null  object
 2   popularity            10866 non-null  float64
 3   budget                10866 non-null  int64
 4   revenue               10866 non-null  int64
 5   original_title        10866 non-null  object
 6   cast                  10790 non-null  object
 7   homepage              2936 non-null   object
 8   director              10822 non-null  object
 9   tagline               8042 non-null   object
 10  keywords              9373 non-null   object
 11  overview              10862 non-null  object
 12  runtime               10866 non-null  int64
 13  genres                10843 non-null  object
 14  production_companies  9836 non-null   object
```

```
 15  release_date          10866 non-null  object
 16  vote_count            10866 non-null  int64
 17  vote_average          10866 non-null  float64
 18  release_year          10866 non-null  int64
 19  budget_adj            10866 non-null  float64
 20  revenue_adj           10866 non-null  float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

[9]:
```
# view missing value count for archive
df.isnull().sum()
```

[9]:
```
id                         0
imdb_id                   10
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      76
homepage                7930
director                  44
tagline                 2824
keywords                1493
overview                   4
runtime                    0
genres                    23
production_companies    1030
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

[10]:
```
# number not in the rating range(0-10)
df.vote_average.value_counts()
```

[10]:
```
6.1    496
6.0    495
5.8    486
5.9    473
6.2    464
      ...
8.9      1
8.6      1
9.2      1
```

```
8.7      1
2.0      1
Name: vote_average, Length: 72, dtype: int64
```

[11]:
```
# counts keywords
df.keywords .value_counts()
```

[11]:
```
woman director                                          134
independent film                                         82
sport                                                    25
musical                                                  24
suspense                                                 24
                                                       ...
los angeles|gangster                                      1
family|christmas                                          1
new york|film director|legend|song|public                 1
celebrity|thief|fame|burglary|hollywood                   1
female nudity|monster|spacecraft|experiment|nudity        1
Name: keywords, Length: 8804, dtype: int64
```

[12]:
```
# counts production by companies
df.production_companies .value_counts()
```

[12]:
```
Paramount Pictures                                    156
Universal Pictures                                    133
Warner Bros.                                           84
Walt Disney Pictures                                   76
Columbia Pictures                                      72
                                                     ...
Wessler Entertainment                                   1
Trimark Pictures|Anglo-American Film Corporation        1
Studio Canal|Mandarin Film                              1
Turner Network Television                               1
Scott Free Productions|YouTube|LG                       1
Name: production_companies, Length: 7445, dtype: int64
```

[13]:
```
#check for duplicates in the data by imdb_id
df.imdb_id.duplicated
```

[13]:
```
<bound method Series.duplicated of 0        tt0369610
1        tt1392190
2        tt2908446
3        tt2488496
4        tt2820852
           ...
10861    tt0060371
10862    tt0060472
```

```
10863    tt0060161
10864    tt0061177
10865    tt0060666
Name: imdb_id, Length: 10866, dtype: object>
```

[14]:
```
#check for duplicates in the data by id
df.id.duplicated()
```

[14]:
```
0        False
1        False
2        False
3        False
4        False
         …
10861    False
10862    False
10863    False
10864    False
10865    False
Name: id, Length: 10866, dtype: bool
```

[15]:
```
#check unique in keywords col
df.keywords.unique
```

[15]:
```
<bound method Series.unique of 0        monster|dna|tyrannosaurus
rex|velociraptor|island
1         future|chase|post-apocalyptic|dystopia|australia
2        based on novel|revolution|dystopia|sequel|dyst…
3                     android|spaceship|jedi|space opera|3d
4                       car race|speed|revenge|suspense|car
                                 …
10861                             surfer|surfboard|surfing
10862                           car race|racing|formula 1
10863                            car|trolley|stealing car
10864                                                spoof
10865                   fire|gun|drive|sacrifice|flashlight
Name: keywords, Length: 10866, dtype: object>
```

[16]:
```
# display a sample of the dataframe
df.sample
```

[16]:
```
<bound method NDFrame.sample of            id    imdb_id  popularity     budget
revenue  \
0      135397  tt0369610   32.985763  150000000  1513528810
1       76341  tt1392190   28.419936  150000000   378436354
2      262500  tt2908446   13.112507  110000000   295238201
3      140607  tt2488496   11.173104  200000000  2068178225
```

```
4       168259  tt2820852    9.335014  190000000  1506249360
…          …         …           …         …          …
10861      21  tt0060371    0.080598          0           0
10862   20379  tt0060472    0.065543          0           0
10863   39768  tt0060161    0.065141          0           0
10864   21449  tt0061177    0.064317          0           0
10865   22293  tt0060666    0.035919      19000           0

                    original_title  \
0                   Jurassic World
1                Mad Max: Fury Road
2                         Insurgent
3       Star Wars: The Force Awakens
4                         Furious 7
…                               …
10861             The Endless Summer
10862                    Grand Prix
10863            Beregis Avtomobilya
10864         What's Up, Tiger Lily?
10865       Manos: The Hands of Fate

                                                  cast  \
0       Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…
1       Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic…
2       Shailene Woodley|Theo James|Kate Winslet|Ansel…
3       Harrison Ford|Mark Hamill|Carrie Fisher|Adam D…
4       Vin Diesel|Paul Walker|Jason Statham|Michelle …
…                                                     …
10861   Michael Hynson|Robert August|Lord 'Tally Ho' B…
10862   James Garner|Eva Marie Saint|Yves Montand|Tosh…
10863   Innokentiy Smoktunovskiy|Oleg Efremov|Georgi Z…
10864   Tatsuya Mihashi|Akiko Wakabayashi|Mie Hama|Joh…
10865   Harold P. Warren|Tom Neyman|John Reynolds|Dian…

                                          homepage            director  \
0                         http://www.jurassicworld.com/     Colin Trevorrow
1                           http://www.madmaxmovie.com/       George Miller
2          http://www.thedivergentseries.movie/#insurgent    Robert Schwentke
3       http://www.starwars.com/films/star-wars-episod…         J.J. Abrams
4                               http://www.furious7.com/           James Wan
…                                                     …                 …
10861                                              NaN         Bruce Brown
10862                                              NaN  John Frankenheimer
10863                                              NaN      Eldar Ryazanov
10864                                              NaN         Woody Allen
10865                                              NaN    Harold P. Warren
```

```
                                                tagline  …  \
0                                      The park is open.  …
1                                    What a Lovely Day.  …
2                                 One Choice Can Destroy You  …
3                               Every generation has a story.  …
4                                         Vengeance Hits Home  …
…                                                        …  …
10861                                                   NaN  …
10862  Cinerama sweeps YOU into a drama of speed and …  …
10863                                                   NaN  …
10864                              WOODY ALLEN STRIKES BACK!  …
10865       It's Shocking! It's Beyond Your Imagination!  …

                                                 overview runtime  \
0      Twenty-two years after the events of Jurassic …     124
1      An apocalyptic story set in the furthest reach…     120
2      Beatrice Prior must confront her inner demons …     119
3      Thirty years after defeating the Galactic Empi…     136
4      Deckard Shaw seeks revenge against Dominic Tor…     137
…                                                    …     …
10861  The Endless Summer, by Bruce Brown, is one of …      95
10862  Grand Prix driver Pete Aron is fired by his te…     176
10863  An insurance agent who moonlights as a carthie…      94
10864  In comic Woody Allen's film debut, he took the…      80
10865  A family gets lost on the road and stumbles up…      74

                                          genres  \
0      Action|Adventure|Science Fiction|Thriller
1      Action|Adventure|Science Fiction|Thriller
2             Adventure|Science Fiction|Thriller
3       Action|Adventure|Science Fiction|Fantasy
4                       Action|Crime|Thriller
…                                          …
10861                          Documentary
10862                Action|Adventure|Drama
10863                       Mystery|Comedy
10864                        Action|Comedy
10865                              Horror

                                    production_companies release_date  \
0      Universal Studios|Amblin Entertainment|Legenda…       6/9/15
1      Village Roadshow Pictures|Kennedy Miller Produ…      5/13/15
2      Summit Entertainment|Mandeville Films|Red Wago…      3/18/15
3              Lucasfilm|Truenorth Productions|Bad Robot     12/15/15
4      Universal Pictures|Original Film|Media Rights …       4/1/15
…                                                    …          …
10861                              Bruce Brown Films      6/15/66
```

```
10862  Cherokee Productions|Joel Productions|Douglas …     12/21/66
10863                                            Mosfilm       1/1/66
10864                                Benedict Pictures Corp.      11/2/66
10865                                            Norm-Iris     11/15/66

      vote_count  vote_average  release_year    budget_adj   revenue_adj
0           5562           6.5          2015  1.379999e+08  1.392446e+09
1           6185           7.1          2015  1.379999e+08  3.481613e+08
2           2480           6.3          2015  1.012000e+08  2.716190e+08
3           5292           7.5          2015  1.839999e+08  1.902723e+09
4           2947           7.3          2015  1.747999e+08  1.385749e+09
…            …             …             …             …             …
10861         11           7.4          1966  0.000000e+00  0.000000e+00
10862         20           5.7          1966  0.000000e+00  0.000000e+00
10863         11           6.5          1966  0.000000e+00  0.000000e+00
10864         22           5.4          1966  0.000000e+00  0.000000e+00
10865         15           1.5          1966  1.276423e+05  0.000000e+00

[10866 rows x 21 columns]>
```

[17]:
```
# display a static summary of the dataframe
df.describe()
```

[17]:
```
                 id    popularity        budget       revenue       runtime  \
count   10866.000000  10866.000000  1.086600e+04  1.086600e+04  10866.000000
mean    66064.177434      0.646441  1.462570e+07  3.982332e+07    102.070863
std     92130.136561      1.000185  3.091321e+07  1.170035e+08     31.381405
min         5.000000      0.000065  0.000000e+00  0.000000e+00      0.000000
25%     10596.250000      0.207583  0.000000e+00  0.000000e+00     90.000000
50%     20669.000000      0.383856  0.000000e+00  0.000000e+00     99.000000
75%     75610.000000      0.713817  1.500000e+07  2.400000e+07    111.000000
max    417859.000000     32.985763  4.250000e+08  2.781506e+09    900.000000

         vote_count  vote_average  release_year    budget_adj   revenue_adj
count  10866.000000  10866.000000  10866.000000  1.086600e+04  1.086600e+04
mean     217.389748      5.974922   2001.322658  1.755104e+07  5.136436e+07
std      575.619058      0.935142     12.812941  3.430616e+07  1.446325e+08
min       10.000000      1.500000   1960.000000  0.000000e+00  0.000000e+00
25%       17.000000      5.400000   1995.000000  0.000000e+00  0.000000e+00
50%       38.000000      6.000000   2006.000000  0.000000e+00  0.000000e+00
75%      145.750000      6.600000   2011.000000  2.085325e+07  3.369710e+07
max     9767.000000      9.200000   2015.000000  4.250000e+08  2.827124e+09
```

1.2 Data Cleaning

Cleaning steps in the second part of this section.

1.2.1 1. Rearrange columns

[18]:
```
df.columns
```

[18]:
```
Index(['id', 'imdb_id', 'popularity', 'budget', 'revenue', 'original_title',
       'cast', 'homepage', 'director', 'tagline', 'keywords', 'overview',
       'runtime', 'genres', 'production_companies', 'release_date',
       'vote_count', 'vote_average', 'release_year', 'budget_adj',
       'revenue_adj'],
      dtype='object')
```

[19]:
```
col = ['original_title','genres','release_year','id', 'imdb_id', 'popularity',␣
 ↪'budget', 'revenue',
       'cast', 'homepage', 'director', 'tagline', 'keywords', 'overview',
       'runtime', 'production_companies', 'release_date',
       'vote_count', 'vote_average', 'budget_adj',
       'revenue_adj']

df= df.reindex(columns=col)
```

[20]:
```
df.columns
```

[20]:
```
Index(['original_title', 'genres', 'release_year', 'id', 'imdb_id',
       'popularity', 'budget', 'revenue', 'cast', 'homepage', 'director',
       'tagline', 'keywords', 'overview', 'runtime', 'production_companies',
       'release_date', 'vote_count', 'vote_average', 'budget_adj',
       'revenue_adj'],
      dtype='object')
```

1.2.2 2. Removing columns with unnecessary Values

[21]:
```
df.drop(['id', 'imdb_id','tagline','homepage','overview'], axis=1, inplace=True)
```

1.2.3 3. Changing the release date column into standard date format

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

1.2.4 4. Changing format of budget and revenue columns

[23]:
```
cols = ['budget', 'revenue','budget_adj', 'revenue_adj']
df[cols] = df[cols].applymap(np.int64)
```

[24]:
```
df.dtypes
```

[24]:
```
original_title                 object
genres                         object
release_year                    int64
popularity                     float64
budget                          int64
revenue                         int64
cast                           object
director                       object
keywords                       object
runtime                         int64
production_companies           object
release_date           datetime64[ns]
vote_count                      int64
vote_average                   float64
budget_adj                      int64
revenue_adj                     int64
dtype: object
```

5.Replace value "0" with mean

[25]:
```
df.revenue.value_counts()
```

[25]:
```
0           6016
12000000      10
10000000       8
11000000       7
6000000        6
            ...
53676580       1
617000         1
13001257       1
504050219      1
20518905       1
Name: revenue, Length: 4702, dtype: int64
```

1.2.5 mean value before Replace "0"value

[26]:
```
revenue_mean= df.revenue.mean()
revenue_mean
```

[26]:
```
39823319.79339223
```

[27]:
```
budget_mean= df.budget.mean()
budget_mean
```

[27]:
```
14625701.09414688
```

[28]:
```
revenue_adj_mean= df.revenue_adj.mean()
revenue_adj_mean
```

[28]:
```
51364363.03929689
```

[29]:
```
budget_adj_mean= df.budget_adj.mean()
budget_adj_mean
```

[29]:
```
17551039.596999817
```

[30]:
```
#calculate Profit for each of the movie
#add a new column in the dataframe name 'Profit
df['Profit_1'] = df['revenue'] -df['budget']

df.head(2)
```

[30]:

	original_title	genres	release_year	popularity	budget	revenue	cast	director	keywords	runtime	production_companies	release_date	vote_count	vote_average	budget_adj	revenue_adj	Profit_1
0	Jurassic World	Action|Adventure|Science Fiction|Thriller	2015	32.985763	150000000	1513528810	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…	Colin Trevorrow	monster|dna|tyrannosaurus rex|velociraptor|island	124	Universal Studios|Amblin Entertainment|Legenda…	2015-06-09	5562	6.5	137999939	1392445892	1363528810
1	Mad Max: Fury Road	Action|Adventure|Science Fiction|Thriller	2015	28.419936	150000000	378436354	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic…	George Miller	future|chase|post-apocalyptic|dystopia|australia	120	Village Roadshow Pictures|Kennedy Miller Produ…	2015-05-13	6185	7.1	137999939	348161292	228436354

[31]:
```
df.revenue.value_counts()
```

[31]:
```
0           6016
12000000      10
10000000       8
11000000       7
6000000        6
```

```
              …
53676580      1
617000        1
13001257      1
504050219     1
20518905      1
Name: revenue, Length: 4702, dtype: int64
```

[32]:
```
# Explore the Data frame 'keywords' ='woman director'
df2=df.loc[df['keywords'] =='woman director']
df2
```

[32]:
```
                            original_title                genres  \
174                        Jenny's Wedding          Comedy|Drama
227               Careful What You Wish For              Thriller
302                    The Girl in the Book                 Drama
336                                  Zipper        Thriller|Drama
372                     Learning to Drive  Romance|Comedy|Drama
…                                       …                     …
9406                             Pure Luck  Comedy|Crime|Mystery
9422                     He Said, She Said  Comedy|Drama|Romance
9556   La Vie est un long fleuve tranquille                Comedy
9843                           Grey Gardens           Documentary
10102                       Xiaoao jiang hu  Action|Drama|History

       release_year  popularity   budget  revenue  \
174            2015    0.861179        0        0
227            2015    0.586735  8000000        0
302            2015    0.405095        0        0
336            2015    0.357567        0        0
372            2015    0.307395        0  3447633
…               …          …       …        …
9406           1991    0.245828        0        0
9422           1991    0.202054        0        0
9556           1988    0.194418        0        0
9843           1975    0.125881        0        0
10102          1990    0.073640  2068041        0

                                                    cast  \
174    Katherine Heigl|Tom Wilkinson|Alexis Bledel|Gr…
227    Dermot Mulroney|Kandyse McClure|Isabel Lucas|G…
302    Emily VanCamp|Michael Nyqvist|Ana Mulvoy Ten|T…
336    Patrick Wilson|Lena Headey|Richard Dreyfuss|Ra…
372    Patricia Clarkson|Ben Kingsley|Jake Weber|Sari…
…                                                    …
9406   Martin Short|Danny Glover|Harry Shearer|Sheila…
9422   Kevin Bacon|Elizabeth Perkins|Nathan Lane|Anth…
```

```
9556   BenoÃ®t Magimel|ValÃ©rie Lalande|Catherine Hie…
9843   Edith Bouvier Beale|Brooks Hyers|Norman Vincen…
10102  Samuel Hui|Cecilia Yip|Jacky Cheung|Sharla Che…

                                                director        keywords  \
174                                  Mary Agnes Donoghue  woman director
227                               Elizabeth Allen Rosenbaum  woman director
302                                          Marya Cohn  woman director
336                                       Mora Stephens  woman director
372                                       Isabel Coixet  woman director
…                                                   …             …
9406                                        Nadia Tass  woman director
9422                                Ken Kwapis|Marisa Silver  woman director
9556                                      Ã%tienne Chatiliez  woman director
9843   Ellen Hovde|Muffie Meyer|Albert Maysles|David …  woman director
10102  King Hu|Tsui Hark|Ann Hui|Ching Siu-Tung|Raymo…  woman director

       runtime                                  production_companies  \
174         94  MM Productions|Merced Media Partners|PalmStar …
227         91  Hyde Park Entertainment|Troika Pictures|Amasia…
302         86                  Varient Busted Buggy Entertainmen
336        103  Protozoa Pictures|Cargo Entertainment|33 Pictures
372        105             Broad Green Pictures|Lavender Pictures
…          …                                                   …
9406        96  Universal Pictures|Silver Lion Films|Sean Dani…
9422       115                                 Paramount Pictures
9556        90  MK2 Productions|France 3 Cinema|Centre Nationa…
9843       100                                    Portrait Films
10102      120                                               NaN

      release_date  vote_count  vote_average  budget_adj  revenue_adj  \
174     2015-07-31          92           5.2           0            0
227     2015-05-06          57           5.4     7359996            0
302     2015-06-13          23           6.2           0            0
336     2015-08-28          30           5.5           0            0
372     2015-08-06          44           5.9           0      3171820
…              …           …             …           …            …
9406    1991-08-09          13           6.7           0            0
9422    1991-02-22          10           5.2           0            0
9556    1988-02-03          28           7.2           0            0
9843    1975-09-27          22           7.2           0            0
10102   1990-04-05          12           6.9     3451350            0

       Profit_1
174           0
227    -8000000
302           0
```

```
336           0
372     3447633
…           …
9406          0
9422          0
9556          0
9843          0
10102  -2068041

[134 rows x 17 columns]
```

[33]:
```
df2.budget.value_counts()
```

[33]:
```
0           117
6000000       2
12000000      2
8000000       2
15000000      1
300000        1
24000000      1
11000000      1
7000000       1
20000000      1
2500000       1
250000        1
1400000       1
10000000      1
2068041       1
Name: budget, dtype: int64
```

[34]:
```
df2.revenue.value_counts()
```

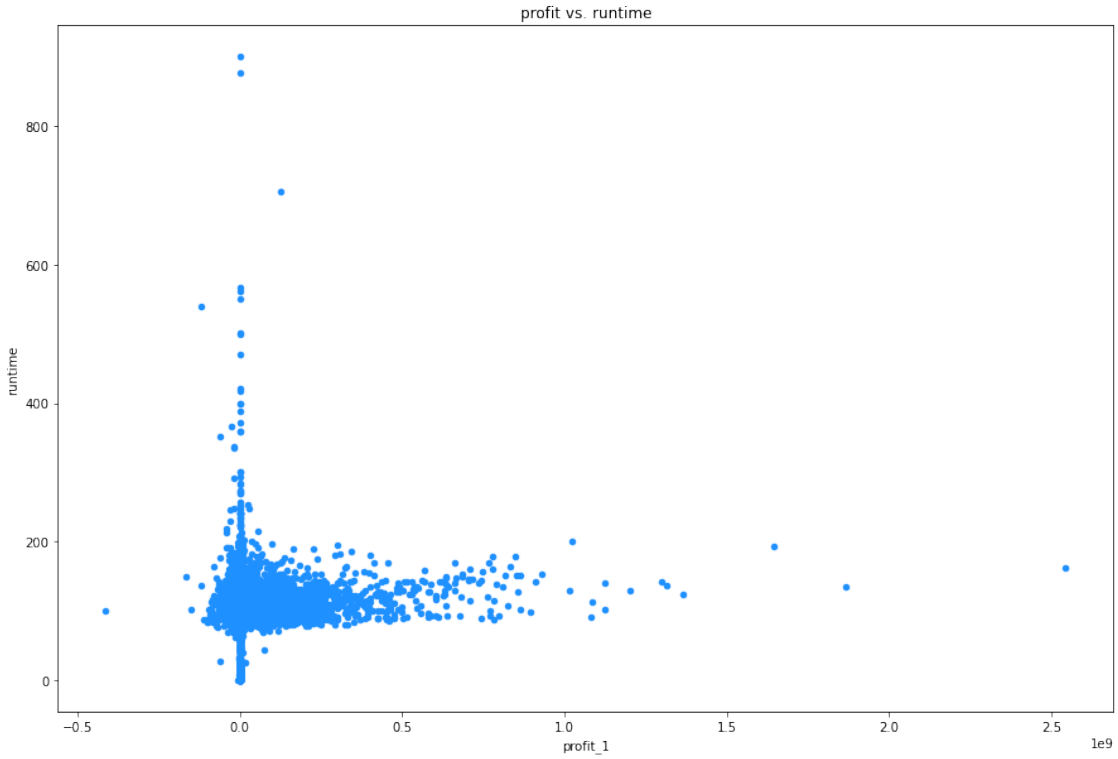
[34]:
```
0           119
32251         1
38105395      1
278000        1
5217498       1
13769         1
32726956      1
24995         1
26488734      1
884100        1
4187          1
3447633       1
21520719      1
201275        1
4186931       1
```

```
9576495       1
Name: revenue, dtype: int64
```

we notice that woman director movies has a multi missing data for budget and revenue and that will reflect in calculate profit

as we see woman director is 134 rows × 17 columns there is 117 budget and 119 revenue value equal to zero or missing

[35]:
```
# Plot scatter plot of Profit_1 and runtime
df.plot(x='Profit_1', y='runtime', kind='scatter',␣
 ↪figsize=(15,10),color='dodgerblue')
plt.title('profit vs. runtime')
plt.xlabel('profit_1')
plt.ylabel('runtime');
```



we make that step to check if replacing Replace value "0" with mean will change our view to the data set

[36]:
```
df.loc[df['revenue'] == 0 , 'revenue'] = revenue_mean
df.loc[df['budget'] == 0 , 'budget'] = budget_mean
df.loc[df['revenue_adj'] == 0 , 'revenue_adj'] = revenue_adj_mean
df.loc[df['budget_adj'] == 0 , 'budget_adj'] = budget_adj_mean
```

1.2.6 mean value after Replace “0”value with the mean

[37]:
```
df.revenue.mean()
```

[37]:
```
61871644.09645201
```

[38]:
```
df.budget.mean()
```

[38]:
```
22292551.216755074
```

[39]:
```
df.revenue_adj.mean()
```

[39]:
```
79802427.46451408
```

[40]:
```
df.budget_adj.mean()
```

[40]:
```
26754594.13626955
```

[41]:
```
df.revenue.value_counts()
```

[41]:
```
3.982332e+07    6016
1.200000e+07      10
1.000000e+07       8
1.100000e+07       7
2.000000e+06       6
                 …
1.150000e+08       1
2.376813e+08       1
4.418098e+08       1
3.123963e+06       1
1.507394e+07       1
Name: revenue, Length: 4702, dtype: int64
```

[42]:
```
cols = ['budget', 'revenue','budget_adj', 'revenue_adj']
df[cols] = df[cols].applymap(np.int64)
```

[43]:
```
#calculate Profit for each of the movie
#add a new column in the dataframe name 'Profit
df['Profit_2'] = df['revenue'] -df['budget']

df.head(2)
```

[43]:
```
        original_title                                     genres  \
0      Jurassic World  Action|Adventure|Science Fiction|Thriller
1  Mad Max: Fury Road  Action|Adventure|Science Fiction|Thriller

   release_year  popularity     budget     revenue  \
```

```
0          2015   32.985763  150000000  1513528810
1          2015   28.419936  150000000   378436354

                                                cast         director  \
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…  Colin Trevorrow
1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic…    George Miller

                                          keywords  runtime  \
0  monster|dna|tyrannosaurus rex|velociraptor|island      124
1   future|chase|post-apocalyptic|dystopia|australia      120

                               production_companies release_date  vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda…   2015-06-09        5562
1  Village Roadshow Pictures|Kennedy Miller Produ…   2015-05-13        6185

   vote_average  budget_adj  revenue_adj    Profit_1    Profit_2
0           6.5   137999939   1392445892  1363528810  1363528810
1           7.1   137999939    348161292   228436354   228436354
```

[44]:
```
df .dtypes
```

[44]:
```
original_title                object
genres                        object
release_year                   int64
popularity                   float64
budget                         int64
revenue                        int64
cast                          object
director                      object
keywords                      object
runtime                        int64
production_companies          object
release_date          datetime64[ns]
vote_count                     int64
vote_average                 float64
budget_adj                     int64
revenue_adj                    int64
Profit_1                       int64
Profit_2                       int64
dtype: object
```

7. making new Data Frame with profit with less Column

[45]:
```
df_profit = df.drop(['revenue',␣
↪'budget','revenue_adj','budget_adj','cast','production_companies','genres','release_date','
↪axis=1, inplace=True)
```

```
df_profit =df.to_csv (r'C:
↪\Users\Abdelrazek\Downloads\investigate-tmdb-movies-dataset\movies-profit.
↪csv', index = False, header=True)
df_profit = pd.read_csv('movies-profit.csv')
df_profit.head()
```

[45]:

	original_title	release_year	popularity	director	keywords	runtime	vote_average	Profit_1	Profit_2
0	Jurassic World	2015	32.985763	Colin Trevorrow	monster|dna|tyrannosaurus rex|velociraptor|island	124	6.5	1363528810	1363528810
1	Mad Max: Fury Road	2015	28.419936	George Miller	future|chase|post-apocalyptic|dystopia|australia	120	7.1	228436354	228436354
2	Insurgent	2015	13.112507	Robert Schwentke	based on novel|revolution|dystopia|sequel|dyst…	119	6.3	185238201	185238201
3	Star Wars: The Force Awakens	2015	11.173104	J.J. Abrams	android|spaceship|jedi|space opera|3d	136	7.5	1868178225	1868178225
4	Furious 7	2015	9.335014	James Wan	car race|speed|revenge|suspense|car	137	7.3	1316249360	1316249360

[46]:
```
df_profit.shape
```

[46]: (10866, 9)

[47]:
```
df_profit.describe()
```

[47]:

	release_year	popularity	runtime	vote_average	Profit_1	Profit_2
count	10866.000000	10866.000000	10866.000000	10866.000000	1.086600e+04	1.086600e+04
mean	2001.322658	0.646441	102.070863	5.974922	2.519762e+07	3.957909e+07
std	12.812941	1.000185	31.381405	0.935142	9.658799e+07	9.339421e+07
min	1960.000000	0.000065	0.000000	1.500000	-4.139124e+08	
25%	1995.000000	0.207583	90.000000	5.400000	0.000000e+00	
50%	2006.000000	0.383856	99.000000	6.000000	0.000000e+00	
75%	2011.000000	0.713817	111.000000	6.600000	9.083358e+06	
max	2015.000000	32.985763	900.000000	9.200000	2.544506e+09	

```
min   -4.139124e+08
25%    1.435662e+07
50%    2.519762e+07
75%    2.837769e+07
max    2.544506e+09
```

1.2.7 8. explore data with keywords 'woman director' and making new dataframe

[48]:
```
df_woman_director=df_profit.loc[df_profit['keywords'] =='woman director']
df_woman_director.to_csv (r'C:
 ↪\Users\Abdelrazek\Downloads\investigate-tmdb-movies-dataset\woman_director.
 ↪csv', index = False, header=True)
df_woman_director = pd.read_csv('woman_director.csv')

df_woman_director.head()
```

[48]:

	original_title	release_year	popularity	director	keywords	runtime	vote_average	Profit_1	Profit_2
0	Jenny's Wedding	2015	0.861179	Mary Agnes Donoghue	woman director	94	5.2	0	25197618
1	Careful What You Wish For	2015	0.586735	Elizabeth Allen Rosenbaum	woman director	91	5.4	-8000000	31823319
2	The Girl in the Book	2015	0.405095	Marya Cohn	woman director	86	6.2	0	25197618
3	Zipper	2015	0.357567	Mora Stephens	woman director	103	5.5	0	25197618
4	Learning to Drive	2015	0.307395	Isabel Coixet	woman director	105	5.9	3447633	-11178068

the multi missing data for budget and revenue reflect in calculate profit as Profit_1 before add mean value

[49]:
```
df_woman_director.drop(['keywords'], axis=1, inplace=True)
```

[50]:
```
df_woman_director.shape
```

[50]:
```
(134, 8)
```

[51]:
```
df_woman_director.describe()
```

[51]:
```
       release_year  popularity     runtime  vote_average      Profit_1  \
count    134.000000  134.000000  134.000000    134.000000  1.340000e+02
mean    2009.529851    0.263397  101.798507      5.934328 -2.096346e+04
std        6.712899    0.202354   44.413434      0.903719  4.778147e+06
min     1975.000000    0.003066    0.000000      3.900000 -2.000000e+07
25%     2008.000000    0.126169   89.000000      5.300000  0.000000e+00
50%     2012.000000    0.214709   96.000000      5.800000  0.000000e+00
75%     2013.000000    0.348612  102.000000      6.475000  0.000000e+00
max     2015.000000    0.938457  470.000000      8.400000  2.710540e+07

           Profit_2
count  1.340000e+02
mean   2.257432e+07
std    1.075968e+07
min   -1.462151e+07
25%    2.519762e+07
50%    2.519762e+07
75%    2.519762e+07
max    3.952332e+07
```

[52]:
```
df_woman_director.dtypes
```

[52]:
```
original_title     object
release_year        int64
popularity        float64
director           object
runtime             int64
vote_average      float64
Profit_1            int64
Profit_2            int64
dtype: object
```

[53]:
```
df_men_director=df_profit.copy()
print ("df_men_director_ready")
df_men_director.shape
```

```
df_men_director_ready
```

[53]:
```
(10866, 9)
```

9. making new Data Frame with out woman director

[54]:
```
df_men_director=df_profit.loc[df_profit['keywords'] !='woman director']
df_men_director.to_csv (r'C:
 ↪\Users\Abdelrazek\Downloads\investigate-tmdb-movies-dataset\men_director.
 ↪csv', index = False, header=True)
df_men_director = pd.read_csv('men_director.csv')
df_men_director.drop(['keywords'], axis=1, inplace=True)
```

```
df_men_director.head()
```

[54]:

	original_title	release_year	popularity	director	runtime	vote_average	Profit_1	Profit_2
0	Jurassic World	2015	32.985763	Colin Trevorrow	124	6.5	1363528810	1363528810
1	Mad Max: Fury Road	2015	28.419936	George Miller	120	7.1	228436354	228436354
2	Insurgent	2015	13.112507	Robert Schwentke	119	6.3	185238201	185238201
3	Star Wars: The Force Awakens	2015	11.173104	J.J. Abrams	136	7.5	1868178225	1868178225
4	Furious 7	2015	9.335014	James Wan	137	7.3	1316249360	1316249360

[55]:
```
df_men_director.shape
```

[55]: (10732, 8)

[56]:
```
df_men_director.describe()
```

[56]:

	release_year	popularity	runtime	vote_average	Profit_1	Profit_2
count	10732.000000	10732.000000	10732.000000	10732.000000	1.073200e+04	1.073200e+04
mean	2001.220183	0.651224	102.074264	5.975429	2.551250e+07	3.979141e+07
std	12.837888	1.005236	31.187195	0.935557	9.714634e+07	9.394843e+07
min	1960.000000	0.000065	0.000000	1.500000	-4.139124e+08	-4.139124e+08
25%	1994.000000	0.209112	90.000000	5.400000	0.000000e+00	1.382332e+07
50%	2006.000000	0.386049	99.000000	6.000000	0.000000e+00	2.519762e+07
75%	2011.000000	0.719615	112.000000	6.600000	9.866447e+06	2.882332e+07
max	2015.000000	32.985763	900.000000	9.200000	2.544506e+09	2.544506e+09

2 Exploratory Data Analysis

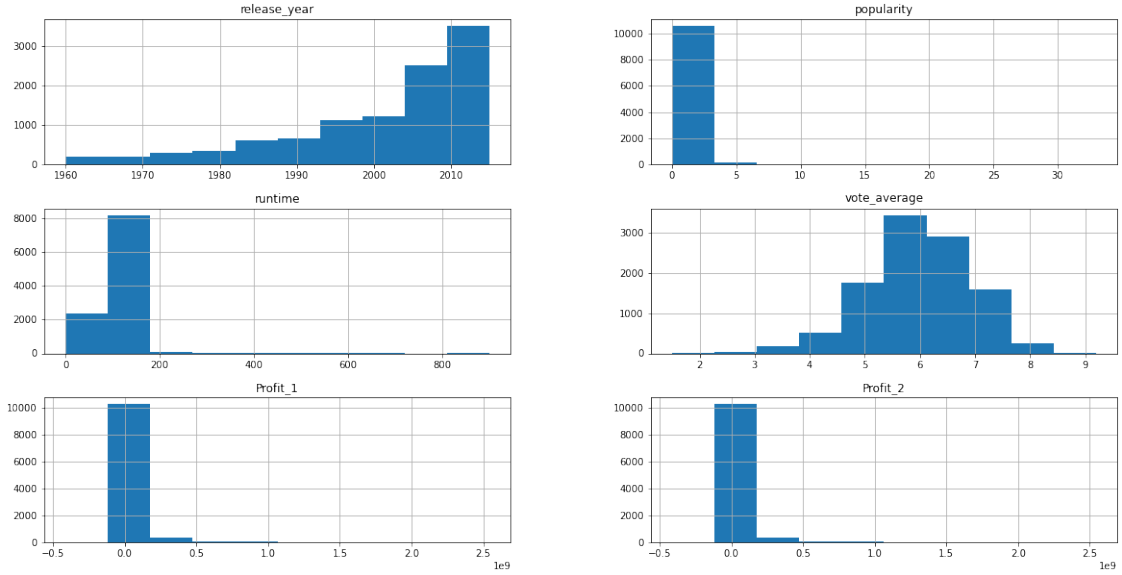
Tip: Now that you've trimmed and cleaned your data, you're ready to move on to exploration. Compute statistics and create visualizations with the goal of addressing the research questions that you posed in the Introduction section. It is recommended

that you be systematic with your approach. Look at one variable at a time, and then follow it up by looking at relationships between variables.

2.0.1 Explore what the histogram of the data looks like

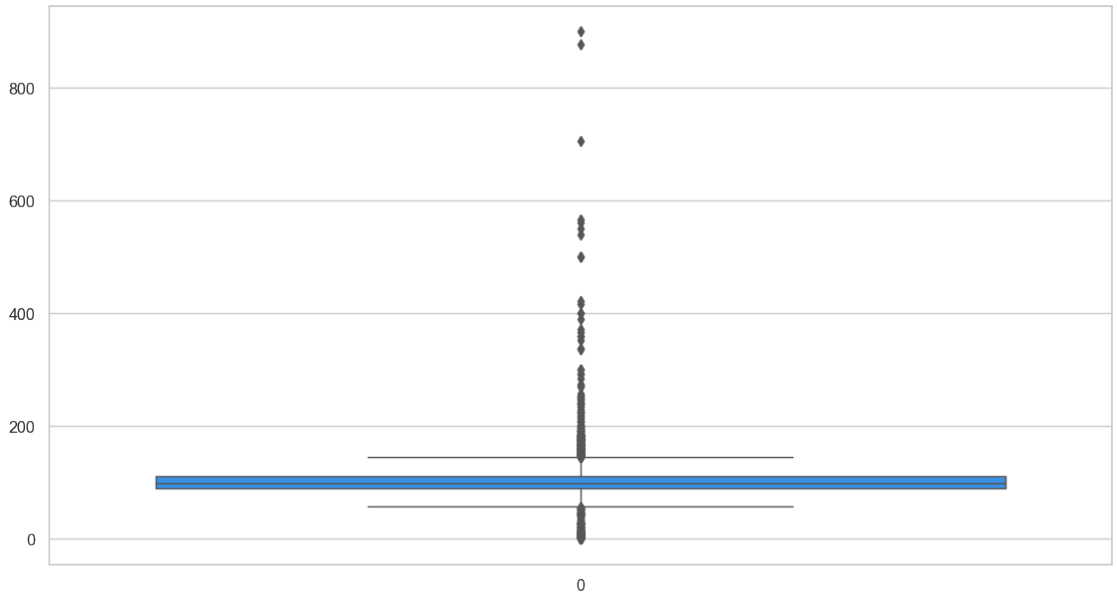
Explore Men_Director

[57]:
```
df_men_director.hist(figsize=(20,10));
```



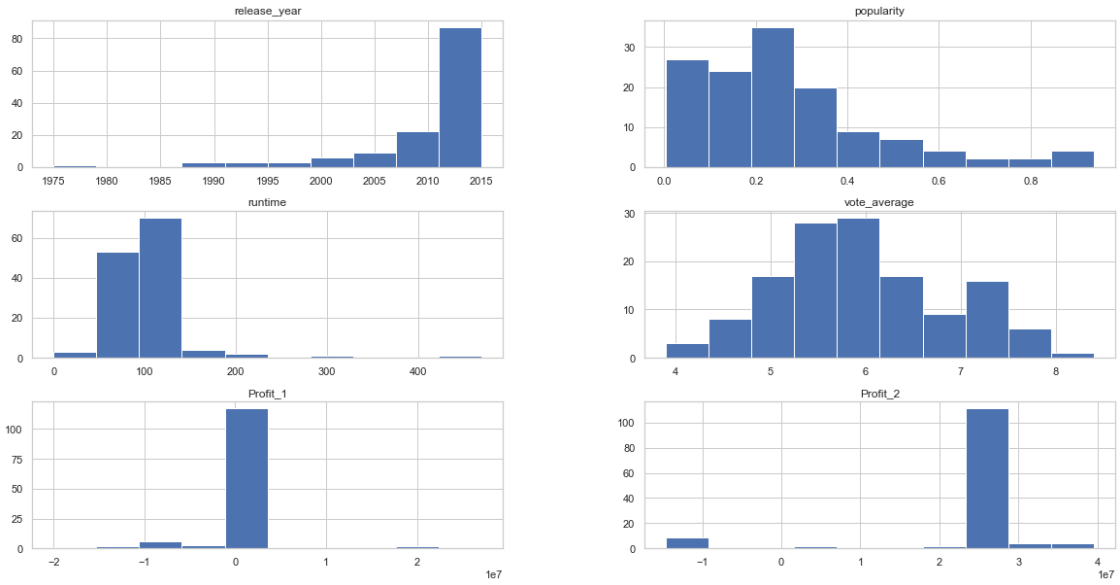
. we can notice that the profit for movies director with men dos not change after adding mean value

[58]:
```
# shoe men_average_films_runtime
plt.figure(figsize=(13,7), dpi = 105)
sns.set_theme(style="whitegrid")
men_average_films_runtime=df_men_director ['runtime']
ax =sns.boxplot(data=men_average_films_runtime, linewidth =␣
 ↪1,color='dodgerblue')
plt.show()
```



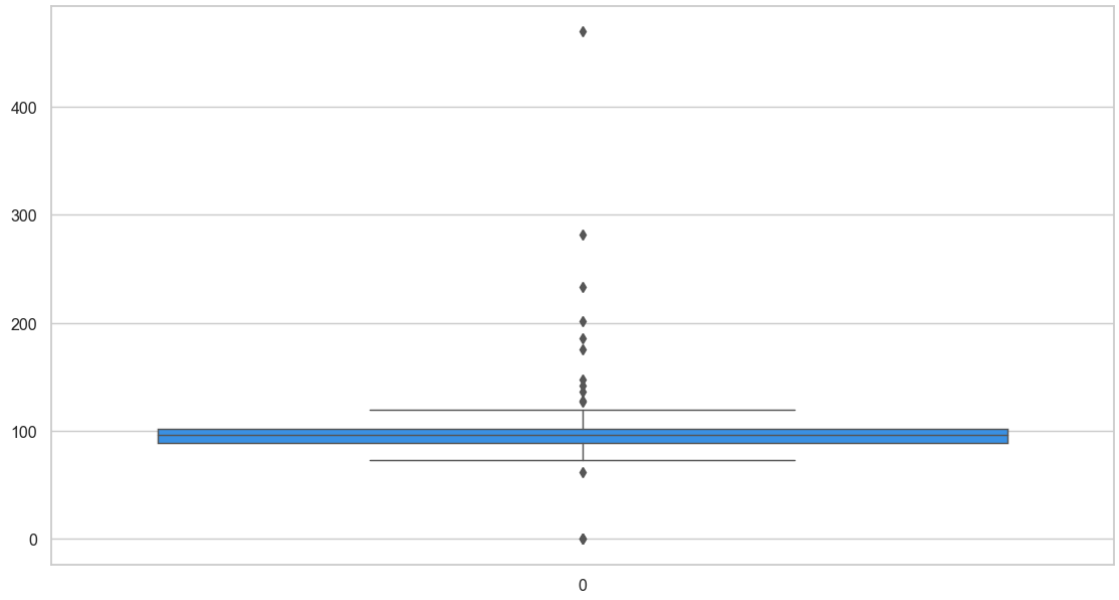
Explore Woman_Director

[59]:
```
df_woman_director.hist(figsize=(20,10));
```



[60]:
```
# shoe woman_average_films_runtime
plt.figure(figsize=(13,7), dpi = 105)
sns.set_theme(style="whitegrid")
woman_average_films_runtime=df_woman_director ['runtime']
```

```
ax =sns.boxplot(data=woman_average_films_runtime, linewidth =␣
↪1,color='dodgerblue')
plt.show()
```



1. we can notice that woman start director films to be notice more than 15 year after 2000

2. women like to directer short film and a little make long film

3. men like to directer films between 120 and 180 min and many make long films

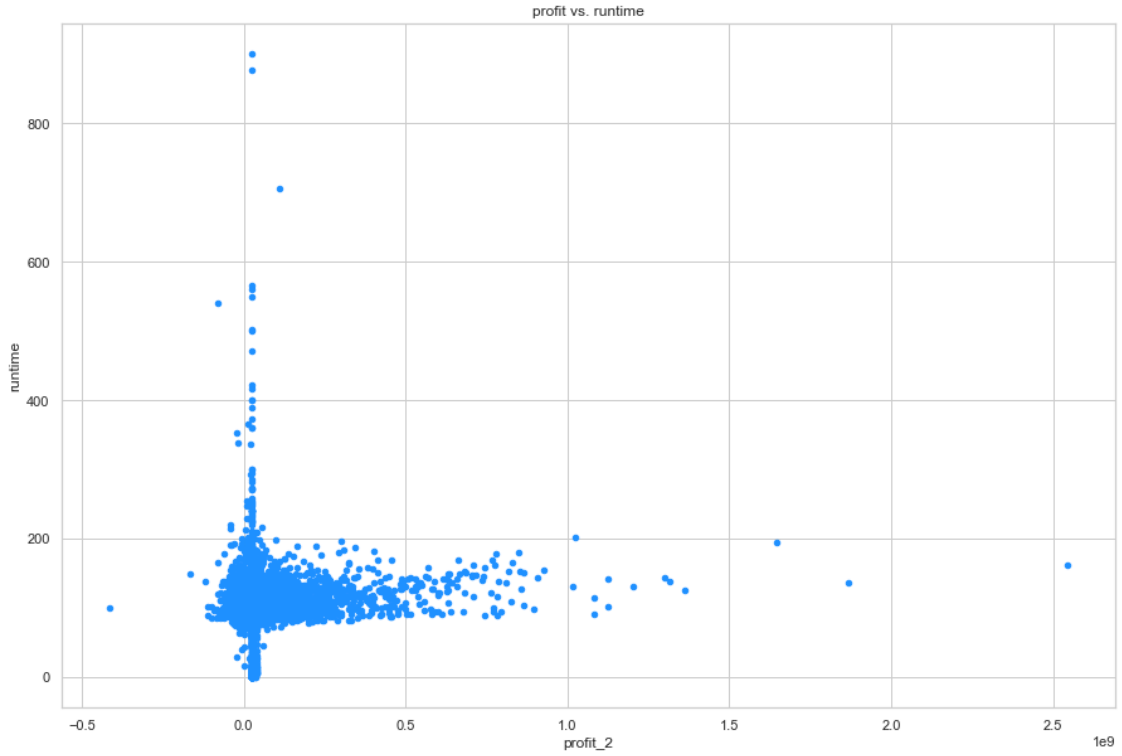
4. men directer Data Frame profit before add mean equal to profit after mean

5. women directer Data Frame profit before add mean not equal to profit after mean and that related to the multi missing value

2.0.2 Research Question 1

What is the relationship between runtime and profit?

[61]:
```
# Plot scatter plot of Profit_2 and runtime
df.plot(x='Profit_2', y='runtime', kind='scatter',␣
↪figsize=(15,10),color='dodgerblue')
plt.title('profit vs. runtime')
plt.xlabel('profit_2')
plt.ylabel('runtime');
plt.show()
```

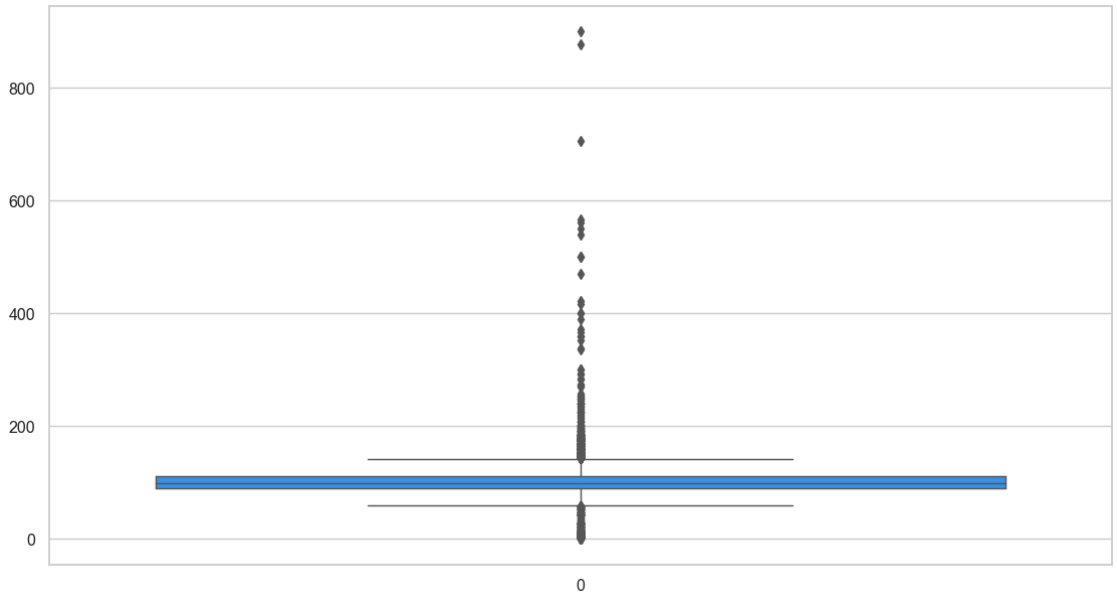


. most films runtime between 100 min to 200 min make profit 0 and o.5 million $

show The average runtime of a movie

[62]:
```
#using seaborn to generate the The average runtime of a movie

plt.figure(figsize=(13,7), dpi = 105)
sns.set_theme(style="whitegrid")
average_runtime=df_profit ['runtime']
ax =sns.boxplot(data=average_runtime, linewidth = 1,color='dodgerblue')
plt.show()
```



2.0.3 Research Question 2

what Movies with most and least profit?

[63]:
```
# Movies_best_profit_info
df_Movies_best_profit=df_profit.sort_values(['Profit_2', 'original_title'],␣
↪ascending=[False, True])
print(df_Movies_best_profit.head(5).to_markdown())
```

```
|      | original_title               |   release_year |   popularity | director
| keywords                                              |   runtime |
vote_average |   Profit_1 |   Profit_2 |
|-----:|:-----------------------------|---------------:|-------------:|:--------
--------|:------------------------------------------------------|----------:|-----
----------:|-----------:|-----------:|
| 1386 | Avatar                       |           2009 |      9.43277 | James
Cameron    | culture clash|future|space war|space colony|society |       162 |
7.1 | 2544505847 | 2544505847 |
|    3 | Star Wars: The Force Awakens |           2015 |     11.1731  | J.J.
Abrams     | android|spaceship|jedi|space opera|3d                |       136 |
7.5 | 1868178225 | 1868178225 |
| 5231 | Titanic                      |           1997 |      4.35522 | James
Cameron    | shipwreck|iceberg|ship|panic|titanic                 |       194 |
7.3 | 1645034188 | 1645034188 |
|    0 | Jurassic World               |           2015 |     32.9858  | Colin
Trevorrow | monster|dna|tyrannosaurus rex|velociraptor|island    |       124 |
6.5 | 1363528810 | 1363528810 |
|    4 | Furious 7                    |           2015 |      9.33501 | James
```

```
Wan       | car race|speed|revenge|suspense|car                  |       137 |
7.3 | 1316249360 | 1316249360 |
```

[64]:
```
# Movies_Least_profit_info
df_Movies_Least_profit=df_profit.sort_values(['Profit_2', 'original_title'],␣
 ↪ascending=[False, True])
print(df_Movies_Least_profit.tail(5).to_markdown())
```

```
|      | original_title    |   release_year |   popularity | director         |
keywords                                                          |   runtime |
vote_average |   Profit_1 |   Profit_2 |
|-----:|:------------------|---------------:|-------------:|:-----------------|:
------------------------------------------------------------------|----------:|--
-----------:|-----------:|-----------:|
| 2069 | The Wolfman       |           2010 |     0.642475 | Joe Johnston     |
father-son relationship|victorian england|rural setting|werewolf |       102 |
5.4 | -150000000 | -110176681 |
| 3484 | Mars Needs Moms   |           2011 |     0.921653 | Simon Wells      |
boy|alien|rescue|martian|alien abduction                          |        88 |
5.5 | -111007242 | -111007242 |
| 7031 | The Alamo         |           2004 |     0.94856  | John Lee Hancock |
texas|officer|uprising|alamo|mexican                              |       137 |
5.9 | -119180039 | -119180039 |
| 5508 | The Lone Ranger   |           2013 |     1.21451  | Gore Verbinski   |
texas|horse|survivor|texas ranger|partner                        |       149 |
6   | -165710090 | -165710090 |
| 2244 | The Warrior's Way |           2010 |     0.25054  | Sngmoo Lee       |
assassin|small town|revenge|deception|super speed                |       100 |
6.4 | -413912431 | -413912431 |
```

[65]:
```
# men_director_Movies_best_profit_info
df_men_director_best_profit=df_men_director.sort_values(['Profit_2',␣
 ↪'original_title'], ascending=[False, True])
print(df_men_director_best_profit.head(1).to_markdown())
```

```
|      | original_title   |   release_year |   popularity | director      |
runtime |   vote_average |   Profit_1 |   Profit_2 |
|-----:|:-----------------|---------------:|-------------:|:--------------|-----
-----:|---------------:|-----------:|-----------:|
| 1358 | Avatar           |           2009 |      9.43277 | James Cameron |
162 |            7.1 | 2544505847 | 2544505847 |
```

Avatar is the most profit in men_director_Movies

[66]:
```
# men_directort_Movies_Least_profit_info
df_men_director_Least_profit=df_men_director.sort_values(['Profit_2',␣
 ↪'original_title'], ascending=[False, True])
print(df_men_director_Least_profit.tail(1).to_markdown())
```

```
|      | original_title    |   release_year |   popularity | director   |
runtime |   vote_average |   Profit_1 |   Profit_2 |
|-----:|:------------------|---------------:|-------------:|:-----------|-------
---:|---------------:|----------:|----------:|
| 2204 | The Warrior's Way |           2010 |      0.25054 | Sngmoo Lee |
100 |            6.4 | -413912431 | -413912431 |
```

The Warrior's Way is the Least profit in men_director_Movies

[67]:
```
# woman_director_Movies_best_profit_info
df_woman_director_best_profit=df_woman_director.sort_values(['Profit_1',␣
↪'original_title'], ascending=[False, True])
HTML(df_woman_director_best_profit.head(1).to_html(classes='table␣
↪table-striped'))
```

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

The Secret Life of Bees is the most profit in woman_director_Movies

[68]:
```
# woman_director_Movies_Least_profit_info
df_woman_director_least_profit=df_woman_director.sort_values(['Profit_1',␣
↪'original_title'], ascending=[False, True])
HTML(df_woman_director_least_profit.tail(1).to_html(classes='table␣
↪table-striped'))
```

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

[69]:
```
# woman_director_Movies_Least_profit_info
df_woman_director_least_profit=df_woman_director.sort_values(['Profit_2',␣
↪'original_title'], ascending=[False, True])
HTML(df_woman_director_least_profit.tail(1).to_html(classes='table␣
↪table-striped'))
```

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

as we mention before about missing data for women_directer

we can see that woman_director_Movies_Least_profit has two different result

1. A Case of You will be the Least_profit Movies if we using col Profit_2 after adding the mean value to our missing data

2. La Rafle will be the Least_profit Movies if we using col Profit_1 before adding the mean value to our missing data

using the static summary of men and women Data Frame and create anew one where release_year' >= 1975 the year of min in woman_director_Movies

2.0.4 Research Question 3

what is the Movies with most and least popularity?

[70]:
```
# Movies with most popularity
df_profit.sort_values(['popularity', 'original_title'], ascending=[False, True])
HTML(df_profit.head(1).to_html(classes='table table-striped'))
```

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

[71]:
```
# Movies with least popularity
df_profit.sort_values(['popularity', 'original_title'], ascending=[False, True])
HTML(df_profit.tail(1).to_html(classes='table table-striped'))
```

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

2.0.5 Research Question 4

what is the top 10 director with most production ?

[72]:
```
# top_10_director_movies_production
df_profit.director.value_counts().head(10)
```

[72]:
```
Woody Allen          45
Clint Eastwood       34
Martin Scorsese      29
Steven Spielberg     29
Ridley Scott         23
Steven Soderbergh    22
Ron Howard           22
Joel Schumacher      21
Brian De Palma       20
Wes Craven           19
Name: director, dtype: int64
```

2.0.6 Research Question 5

which director gain most and least profit all the time?

[73]:
```
# director_Profit_ALL THE TIME
director_Profit=df_profit.groupby(['director'])['Profit_1'].sum().reset_index()
m=director_Profit.sort_values(by='Profit_1',ascending=True,)
```

[74]:
```
# Top_director_Profit
HTML(m.tail(1).to_html(classes='table table-striped'))
```

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

[75]:
```
# least_director_Profit
HTML(m.head(1).to_html(classes='table table-striped'))
```

[75]:
```
<IPython.core.display.HTML object>
```

[76]:
```
# explore Steven Spielberg movies
director_movies= df_profit.loc[df_profit.director == 'Steven Spielberg']
df_top_director_movies=director_movies.sort_values(['release_year',␣
↪'Profit_1'], ascending=[False, False])
df_top_director_movies
```

[76]:

	original_title	release_year	popularity	director
33	Bridge of Spies	2015	3.648210	Steven Spielberg
4425	Lincoln	2012	1.312488	Steven Spielberg
3397	The Adventures of Tintin	2011	2.234300	Steven Spielberg
3414	War Horse	2011	1.592819	Steven Spielberg
2879	Indiana Jones and the Kingdom of the Crystal S…	2008	3.161670	Steven Spielberg
6205	War of the Worlds	2005	1.844731	Steven Spielberg
6265	Munich	2005	0.869394	Steven Spielberg
6988	The Terminal	2004		
3918	Catch Me If You Can	2002		
3921	Minority Report	2002		
2638	A.I. Artificial Intelligence	2001		
8974	Saving Private Ryan	1998		
5391	The Lost World: Jurassic Park	1997		
5387	Amistad	1997		
10223	Jurassic Park	1993		
10222	Schindler's List	1993		
9318	Hook	1991		
9180	Indiana Jones and the Last Crusade	1989		
9219	Always	1989		
9636	Empire of the Sun	1987		
6094	The Color Purple	1985		
7883	Indiana Jones and the Temple of Doom	1984		
8889	E.T. the Extra-Terrestrial	1982		
8375	Raiders of the Lost Ark	1981		
7851	1941	1979		
1334	Close Encounters of the Third Kind	1977		
9806	Jaws	1975		
9770	The Sugarland Express	1974		
9939	Duel	1971		

```
6988     1.682492  Steven Spielberg
3918     2.973115  Steven Spielberg
3921     2.103595  Steven Spielberg
2638     2.971372  Steven Spielberg
8974     2.170136  Steven Spielberg
5391     0.210550  Steven Spielberg
5387     0.221360  Steven Spielberg
10223    2.204926  Steven Spielberg
10222    2.377288  Steven Spielberg
9318     2.326917  Steven Spielberg
9180     3.536655  Steven Spielberg
9219     0.494235  Steven Spielberg
9636     0.460550  Steven Spielberg
6094     1.012186  Steven Spielberg
7883     2.556799  Steven Spielberg
8889     2.900556  Steven Spielberg
8375     4.578300  Steven Spielberg
7851     0.387797  Steven Spielberg
1334     1.104816  Steven Spielberg
9806     2.563191  Steven Spielberg
9770     0.415866  Steven Spielberg
9939     0.462681  Steven Spielberg
```

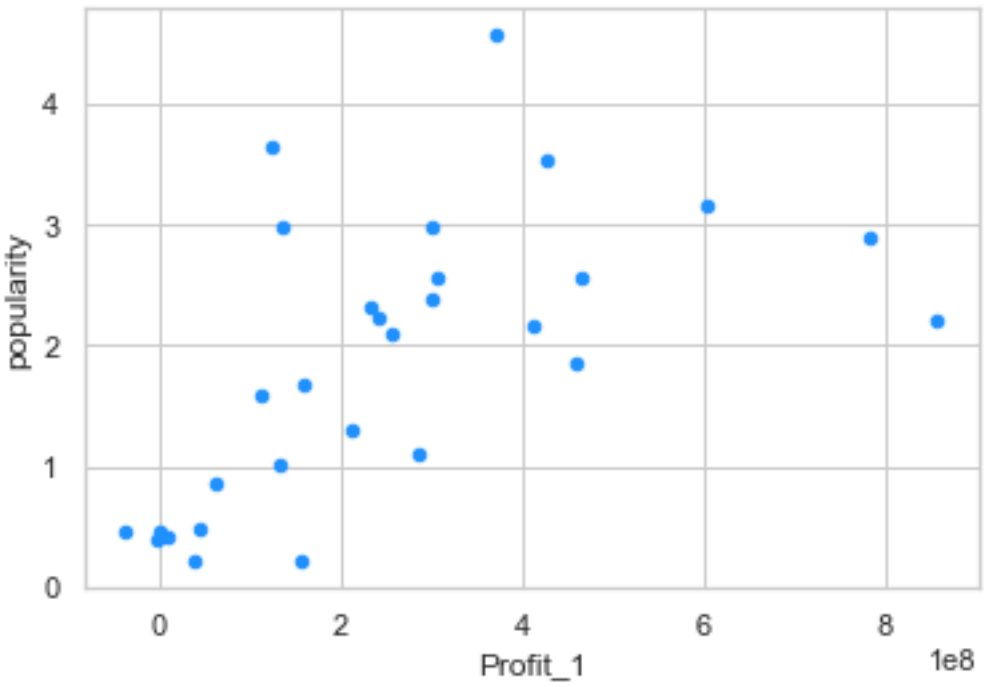
```
                                                keywords  runtime  \
33                           spy|cia|cold war|pilot|lawyer      141
4425   u.s. president|speech|death of a child|battlef…      149
3397        riddle|captain|treasure|liquor|treasure hunt      107
3414       world war i|horse|farm life|execution|trapped      146
2879   saving the world|riddle|whip|treasure|mexico city      122
6205   post traumatic stress  disorder|new jersey|air…      116
6265         paris|assassination|israel|hotel room|1970s      164
6988   new york|airport|marriage proposal|translation…      128
3918   con man|biography|fbi agent|overhead camera sh…      141
3921   self-fulfilling prophecy|evidence|hologram|dys…      145
2638   ocean|artificial intelligence|prophecy|android…      146
8974   killing|loss of brother|loss of lover|war crim…      169
5391   exotic island|dna|paleontology|tyrannosaurus r…      129
5387      cuba|mutiny|slavery|sentence|historical figure      155
10223  exotic island|dna|paleontology|tyrannosaurus r…      127
10222  factory|concentration camp|hero|holocaust|worl…      195
9318           flying|swordplay|sword|peter pan|daughter      144
9180   saving the world|venice|holy grail|library|riddle      127
9219         firemen|pilot|airplane|ghost|guardian angel      122
9636             japan|stadium|harmonica|bravery|peasant      153
6094       prison|africa|southern u.s.|rape|black people      154
7883        riddle|treasure|heart|skeleton|treasure hunt      118
8889   farewell|homesickness|nasa|extraterrestrial te…      115
```

```
8375        saving the world|riddle|nepal|himalaya|cairo      115
7851   submarine|california|world war ii|war ship|pea…      113
1334   indiana|obsession|extraterrestrial technology|…      135
9806   fishing|atlantic ocean|bathing|shipwreck|polic…      124
9770               prison|stadium|texas|hostage|adoption      110
9939   terror|california|falsely accused person|gas s…       90

      vote_average   Profit_1   Profit_2
33             7.1  122610473  122610473
4425           6.6  210293450  210293450
3397           6.6  241940071  241940071
3414           6.9  111584879  111584879
2879           5.6  601636033  601636033
6205           5.9  459739379  459739379
6265           6.7   60358911   60358911
6988           7.0  159417255  159417255
3918           7.6  300114312  300114312
3921           6.9  256372926  256372926
2638           6.6  135926552  135926552
8974           7.7  411840909  411840909
5391           6.1  156074524  156074524
5387           6.6   38000000   38000000
10223          7.4  857100000  857100000
10222          8.1  299265768  299265768
9318           6.5  230854823  230854823
9180           7.4  426171806  426171806
9219           5.8   43134790   43134790
9636           6.9  -38000000    1823319
6094           7.5  131292009  131292009
7883           6.9  305000000  305000000
8889           7.2  782410554  782410554
8375           7.5  371925971  371925971
7851           5.4   -3244258   -3244258
1334           7.0  283788635  283788635
9806           7.3  463654000  463654000
9770           6.1    9800000    9800000
9939           7.0    -450000   39373319
```

[77]:
```
# a scatter plot comparing popularity and Profit Steven Spielberg movies
plt.figure()
df_top_director_movies.
 ↪plot(kind='scatter',x='Profit_1',y='popularity',color='dodgerblue')
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



3 Conclusions

My Research will

1.compare and discuss is the production of the movie affected by who is the director of men or woman

2.try to found what is the successful director all the time

Towards my goal

1.clean and generate df_profit and save as movies-profit.csv 2.split df_profit to two Data Frame df_woman_director and df_men_director and save them as men_director.csv and woman_director.csv

Reading our analyzing summary

woman start director films since 1975 with 143 films and they like to directer short film and a little make long film woman does not present in the Movies best or Least profit list

men start director films since 1960 with 10732 films most film run time between 120 and 180 min and some of them make long film men take all places in the best or Least Movies profit list

director James Cameron take the first and the third place with his film avatar and titanic

but he did not present in the top 10 director movies production list or the most director Profit all the time

Woody Allen take the first place by direct 45 movies although Steven Spielberg take the fourth place with 29 movies but he take the most director Profit all the time

Limitations

Split the data after cleaning to women_directer and Men_Director and discover that women_directer Movies_ missing almost 90% from budget', 'revenue and by add the mean the data change and did not present the actual value so i depend on profit one before add the mean for women_directer data frame to calculate my static and answer my Research Question
and that prevent me from answer the

4 Resource

1.TMDB 5000 Movie Dataset https://www.kaggle.com/tmdb/tmdb-movie-metadata?select=tmdb_5000_credits.csv

2.Set order of columns in pandas Data Frame https://stackoverflow.com/questions/41968732/set-order-of-columns-in-pandas-dataframe

3.Installing nbconvert For converting notebooks to PDF (with –to pdf), nbconvert makes use of LaTeX and the XeTeX as the rendering engine.

https://nbconvert.readthedocs.io/en/latest/install.html#installing-tex

seaborn.boxplot

https://seaborn.pydata.org/generated/seaborn.boxplot.html

color codes

https://www.rapidtables.com/web/color/blue-color.html