Investigation

February 4, 2018

1 Investigate a Dataset (TMDb Movie Dataset)

1.1 Table of Contents

Introduction

The TMDb dataset contains most of the necessary information about a movie like rating, revenue, cast etc.

This data helps us to analyze the movies for trends and answer some interesting questions

1.1.1 Things explored (Questions 1: During years, how are runtime, popularity and average are trending?)

How Runtime is trending over the years
How popularity is trending over the years
How revenue is trending over the years

1.1.2 Associations explored (Question 2: What are some factors that are effecting the revenue of movies)

Directors and revenue in their movies
Genre and revenue in those genre movies
lead actor and revenue in their movies

*Associations and exploration stated are tentative, and the investigation is performed for basic correlation, detailed statistical analysis are yet to be performed.

```
In [41]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         %pylab inline
         import seaborn as sns
```

```
Populating the interactive namespace from numpy and matplotlib
```

Data Wrangling

Tip: In this section of the report, you will load in the data, check for cleanliness, and then trim and clean your dataset for analysis. Make sure that you document your steps carefully and justify your cleaning decisions.

1.1.3 General Properties

```
In [42]: # Loading the data from csv file
         tmdb_data = pd.read_csv('data/tmdb-movies.csv')
         # check to see how data frame looks like
         tmdb_data.head()
```

```
Out[42]:        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

                      original_title  \
         0                Jurassic World
         1            Mad Max: Fury Road
         2                     Insurgent
         3  Star Wars: The Force Awakens
         4                     Furious 7

                                                          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 ...

                                                     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
```

```
                         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       ...

                                            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

                                      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

                                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
2  Summit Entertainment|Mandeville Films|Red Wago...      3/18/15       2480
3          Lucasfilm|Truenorth Productions|Bad Robot     12/15/15       5292
4  Universal Pictures|Original Film|Media Rights ...       4/1/15       2947

   vote_average  release_year       budget_adj       revenue_adj
0      6.500000          2015 137999939.280026 1392445892.523800
1      7.100000          2015 137999939.280026  348161292.489031
2      6.300000          2015 101199955.472019  271619025.407628
3      7.500000          2015 183999919.040035 1902723129.801820
4      7.300000          2015 174799923.088033 1385748801.470520

[5 rows x 21 columns]
```

Info about the data Using head() we were able to see what and observe what exactly are we dealing with here

Now that we can see that there are 21 columns which are already names to access them and the indexex are 0,1,2.

But we can use imdb_id to uniquely identify movies.

Lets check for any null or missing values in the datase

Replacing the null values with mean values

```
In [43]: tmdb_data.describe()
```

Out[43]:
```
                 id   popularity          budget          revenue  \
count  10866.000000 10866.000000    10866.000000     10866.000000
mean   66064.177434     0.646441  14625701.094147   39823319.793392
std    92130.136561     1.000185  30913213.831437  117003486.582085
min        5.000000     0.000065        0.000000         0.000000
25%    10596.250000     0.207583        0.000000         0.000000
50%    20669.000000     0.383856        0.000000         0.000000
75%    75610.000000     0.713817 15000000.000000  24000000.000000
max   417859.000000    32.985763 425000000.000000 2781505847.000000

            runtime   vote_count  vote_average  release_year       budget_adj  \
count  10866.000000 10866.000000  10866.000000  10866.000000     10866.000000
mean     102.070863   217.389748      5.974922   2001.322658  17551039.822887
std       31.381405   575.619058      0.935142     12.812941  34306155.722844
min        0.000000    10.000000      1.500000   1960.000000         0.000000
25%       90.000000    17.000000      5.400000   1995.000000         0.000000
50%       99.000000    38.000000      6.000000   2006.000000         0.000000
75%      111.000000   145.750000      6.600000   2011.000000  20853251.084404
max      900.000000  9767.000000      9.200000   2015.000000 425000000.000000

            revenue_adj
count      10866.000000
mean    51364363.253251
std    144632485.039975
min            0.000000
25%            0.000000
50%            0.000000
75%     33697095.717312
max   2827123750.411890
```

In [44]:
```
mean_data = tmdb_data.mean(skipna=True)
```

In [45]:
```
tmdb_data['budget'] = tmdb_data.budget.mask(tmdb_data.budget < 100, mean_data.budget)
tmdb_data['revenue'] = tmdb_data.revenue.mask(tmdb_data.revenue < 100, mean_data.reve
tmdb_data['budget_adj'] = tmdb_data.budget_adj.mask(tmdb_data.budget_adj < 100, mean_
tmdb_data['revenue_adj'] = tmdb_data.revenue_adj.mask(tmdb_data.revenue_adj < 100, mea
tmdb_data['runtime'] = tmdb_data.runtime.mask(tmdb_data.runtime < 5, mean_data.runtime
```

In [46]:
```
tmdb_data.describe()
```

Out[46]:
```
                 id   popularity          budget          revenue  \
count  10866.000000 10866.000000    10866.000000     10866.000000
mean   66064.177434     0.646441  22354467.382164   62007247.080656
std    92130.136561     1.000185  27979250.085824  110969027.254057
min        5.000000     0.000065      108.000000       100.000000
25%    10596.250000     0.207583  14625701.094147   39823319.793392
50%    20669.000000     0.383856  14625701.094147   39823319.793392
75%    75610.000000     0.713817  15000000.000000   39823319.793392
max   417859.000000    32.985763 425000000.000000 2781505847.000000
```

```
             runtime    vote_count  vote_average  release_year        budget_adj  \
count  10866.000000  10866.000000  10866.000000  10866.000000      10866.000000
mean     102.661838    217.389748      5.974922   2001.322658   26820818.623043
std       30.415838    575.619058      0.935142     12.812941   30467464.342343
min        5.000000     10.000000      1.500000   1960.000000        103.900086
25%       90.000000     17.000000      5.400000   1995.000000   17551039.822887
50%       99.000000     38.000000      6.000000   2006.000000   17551039.822887
75%      111.000000    145.750000      6.600000   2011.000000   20853251.084404
max      900.000000   9767.000000      9.200000   2015.000000  425000000.000000

            revenue_adj
count      10866.000000
mean    79972602.296036
std    136493479.100997
min          114.196069
25%     51364363.253251
50%     51364363.253251
75%     51364363.253251
max   2827123750.411890
```

In [47]:
```
tmdb_data.isnull().sum()
```

Out[47]:
```
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
```

1.1.4 Data Cleaning (Removing or droping null imdb_id row)

> From the above code cell we can see some missing data. >As the missing information is already padded as NaN and Null numberical values as 0, There is not much of a work is to be done here. Most of the data that is missing is not relevant for my analysis anyway

We can see that there are 10 imdb_id's that are null, and we don't need those 10 rows with no imdb_id's.

In [48]:
```
# After discussing the structure of the data and any problems that need to be
#   cleaned, perform those cleaning steps in the second part of this section.

tmdb_data.dropna(axis=0, subset=['imdb_id'], inplace=True)
tmdb_data.isnull().sum()
```

Out[48]:
```
id                         0
imdb_id                    0
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      76
homepage                7922
director                  40
tagline                 2817
keywords                1487
overview                   3
runtime                    0
genres                    21
production_companies    1025
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

1.1.5 Data Cleaning (Removing unwanted columns)

In [49]:
```
tmdb_data = tmdb_data.drop(['keywords','production_companies','tagline', 'homepage',
tmdb_data.info();
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10856 entries, 0 to 10865
Data columns (total 16 columns):
id                  10856 non-null int64
imdb_id             10856 non-null object
```

```
popularity        10856 non-null float64
budget            10856 non-null float64
revenue           10856 non-null float64
original_title    10856 non-null object
cast              10780 non-null object
director          10816 non-null object
runtime           10856 non-null float64
genres            10835 non-null object
release_date      10856 non-null object
vote_count        10856 non-null int64
vote_average      10856 non-null float64
release_year      10856 non-null int64
budget_adj        10856 non-null float64
revenue_adj       10856 non-null float64
dtypes: float64(7), int64(3), object(6)
memory usage: 1.4+ MB
```

In [50]:
```
tmdb_data[tmdb_data.isnull().any(axis=1)]
```

Out[50]:

	id	imdb_id	popularity	budget	revenue	\
371	345637	tt4661600	0.422901	14625701.094147	39823319.793392	
424	363869	tt4835298	0.244648	14625701.094147	39823319.793392	
441	355020	tt4908644	0.220751	14625701.094147	39823319.793392	
465	321109	tt4393514	0.201696	14625701.094147	39823319.793392	
532	320996	tt4073952	0.126594	14625701.094147	39823319.793392	
536	333350	tt3762974	0.122543	14625701.094147	39823319.793392	
538	224972	tt3983674	0.114264	14625701.094147	39823319.793392	
556	321160	tt3908634	0.100910	14625701.094147	39823319.793392	
587	319091	tt4185572	0.062536	14625701.094147	39823319.793392	
600	332479	tt4550996	0.047256	14625701.094147	39823319.793392	
620	361043	tt5022680	0.129696	14625701.094147	39823319.793392	
1032	259910	tt3591568	0.291253	14625701.094147	39823319.793392	
1054	253675	tt3711030	0.269468	14625701.094147	39823319.793392	
1088	169607	tt2714900	0.226028	14625701.094147	1503616.000000	
1173	261041	tt3576038	0.159037	14625701.094147	39823319.793392	
1177	269711	tt3723996	0.153047	14625701.094147	39823319.793392	
1190	250761	tt3279124	0.137962	14625701.094147	39823319.793392	
1203	256561	tt3203290	0.119891	150000.000000	39823319.793392	
1208	282297	tt3302820	0.116190	117.000000	39823319.793392	
1236	250665	tt3399112	0.093062	14625701.094147	39823319.793392	
1241	296370	tt3024964	0.135376	14625701.094147	39823319.793392	
1256	299729	tt3995006	0.076280	14625701.094147	39823319.793392	
1288	301235	tt4217172	0.038364	14625701.094147	39823319.793392	
1315	250745	tt2171902	0.008000	14625701.094147	39823319.793392	
1316	245158	tt2925642	0.007622	14625701.094147	39823319.793392	
1319	320420	tt3249478	0.005844	14625701.094147	39823319.793392	
1326	250668	tt2548738	0.023872	14625701.094147	39823319.793392	

```
1327   258614  tt2966760    0.003504 14625701.094147 39823319.793392
1385    20785  tt0075988    0.002457 14625701.094147 39823319.793392
1712    21634  tt1073510    0.302095 14625701.094147 39823319.793392
...       ...        ...         ...             ...             ...
6760    38580  tt0816562    0.371028 14625701.094147 39823319.793392
6870    14518  tt0863136    0.194447  2300000.000000 39823319.793392
6930    53215  tt1051713    0.076078 14625701.094147 39823319.793392
7579    58432  tt0484273    0.443952 14625701.094147 39823319.793392
7650    12172  tt1093824    0.383253 14625701.094147 39823319.793392
7723    13016  tt1166827    0.197715  7000000.000000 39823319.793392
7767   282758  tt0827573    0.126603 14625701.094147 39823319.793392
7813    22887  tt0914809    0.065543     6000.000000     6000.000000
7814    25565  tt1236486    0.040311 14625701.094147 39823319.793392
7905    13924  tt0086855    0.647261 14625701.094147 39823319.793392
8234    56804  tt0114844    0.028874 14625701.094147 39823319.793392
8292    14002  tt0103767    0.521669  4000000.000000 39823319.793392
8614    65595  tt0117880    0.273934 14625701.094147 39823319.793392
8824    48617  tt0279079    0.191631 14625701.094147 39823319.793392
8878    92208  tt0250593    0.038045 14625701.094147 39823319.793392
8880    48868  tt0400231    0.032577 14625701.094147 39823319.793392
9251    13928  tt0097674    0.471351 14625701.094147 39823319.793392
9307   141859  tt0097446    0.094652 14625701.094147 39823319.793392
9529    13927  tt0096273    0.236514 14625701.094147 39823319.793392
9564    24348  tt0095895    0.168545  2500000.000000   589244.000000
9593    46188  tt0220698    0.001662 14625701.094147 39823319.793392
9677    13926  tt0093832    0.253376 14625701.094147 39823319.793392
9755    48714  tt0061402    0.046272 14625701.094147 39823319.793392
9799    48847  tt0193716    0.175008 14625701.094147 39823319.793392
10386  225804  tt1028555    0.118854 14625701.094147 39823319.793392
10426   34038  tt0061937    0.114034 14625701.094147 39823319.793392
10434   48784  tt0060984    0.146906      200.000000 39823319.793392
10550   13925  tt0091455    0.306425 14625701.094147 39823319.793392
10659    4255  tt0065904    0.344172     5000.000000 39823319.793392
10754    3171  tt0064064    0.002757 14625701.094147 39823319.793392

                                      original_title  \
371                              Sanjay's Super Team
424                                     Belli di papÃ
441         Winter on Fire: Ukraine's Fight for Freedom
465                                      Bitter Lake
532                    Iliza Shlesinger: Freezing Hot
536                                   A Faster Horse
538                             The Mask You Live In
556                                   With This Ring
587                               The Hunting Ground
600                             Star Wars: TIE Fighter
620                                All Hallows' Eve 2
1032             Marvel Studios: Assembling a Universe
```

1054	Unlocking Sherlock
1088	Finding Vivian Maier
1173	The Search for General Tso
1177	JohnnyExpress
1190	Last Days in Vietnam
1203	Free to Play
1208	Cowspiracy: The Sustainability Secret
1236	No No: A Dockumentary
1241	Dance-Off
1256	Banksy Does New York
1288	Top Gear: The Perfect Road Trip 2
1315	Happy Valley
1316	Kids for Cash
1319	Love Me
1326	Rich Hill
1327	Pantani: The Accidental Death of a Cyclist
1385	Emmet Otter's Jug-Band Christmas
1712	Prayers for Bobby
...	...
6760	The Little Matchgirl
6870	Peter & the Wolf
6930	Kiwi!
7579	La hora frÃŋa
7650	Encounters at the End of the World
7723	Zeitgeist
7767	Doctor Who: The Runaway Bride
7813	Loose Change: Final Cut
7814	Transformers: Beginnings
7905	The Adventures of AndrÃľ and Wally B.
8234	Viaggi di nozze
8292	Baraka
8614	T2 3-D: Battle Across Time
8824	Father and Daughter
8878	Mom's Got a Date With a Vampire
8880	The World of Stainboy
9251	Knick Knack
9307	Goldeneye
9529	Tin Toy
9564	Powaqqatsi
9593	Peter Pan
9677	Red's Dream
9755	The Big Shave
9799	The Amputee
10386	The Making of 'The Nightmare Before Christmas'
10426	Magical Mystery Tour
10434	Six Men Getting Sick
10550	Luxo Jr.
10659	The Party at Kitty and Stud's

```
10754                    Bambi Meets Godzilla

                                                   cast  \
371                                                 NaN
424   Diego Abatantuono|Matilde Gioli|Andrea Pisani|...
441                                                 NaN
465                                                 NaN
532                                    Iliza Shlesinger
536                                                 NaN
538                                                 NaN
556   Regina Hall|Jill Scott|Eve|Brooklyn Sudano|Dei...
587                                                 NaN
600                                                 NaN
620                                                 NaN
1032  Robert Downey Jr.|Chris Hemsworth|Chris Evans|...
1054  Benedict Cumberbatch|Martin Freeman|Steven Mof...
1088                                                NaN
1173                                                NaN
1177                                                NaN
1190                                                NaN
1203          Benedict Lim|Danil Ishutin|Clinton Loomis
1208                                                NaN
1236                                                NaN
1241  Kathryn McCormick|Shane Harper|Finola Hughes|C...
1256                                                NaN
1288                    Jeremy Clarkson|Richard Hammond
1315                                                NaN
1316                                                NaN
1319                                                NaN
1326                                                NaN
1327                                                NaN
1385                                                NaN
1712  Ryan Kelley|Sigourney Weaver|Henry Czerny|Dan ...
...                                                 ...
6760                                                NaN
6870                                                NaN
6930                                                NaN
7579  Silke|Omar MuÃśoz|Pepo Oliva|Carola Manzanares...
7650                                                NaN
7723                                                NaN
7767                       David Tennant|Catherine Tate
7813                                                NaN
7814  Peter Cullen|Frank Welker|Mark Ryan|Patrick Ha...
7905                                                NaN
8234  Carlo Verdone|Claudia Gerini|Veronica Pivetti|...
8292                                                NaN
8614  Arnold Schwarzenegger|Linda Hamilton|Edward Fu...
8824                                                NaN
```

```
8878   Matt O'Leary|Laura Vandervoort|Myles Jeffrey|C...
8880                                                 NaN
9251                                                 NaN
9307   Charles Dance|Phyllis Logan|Patrick Ryecart|La...
9529                                                 NaN
9564                                                 NaN
9593                                                 NaN
9677                                                 NaN
9755                                                 NaN
9799                   Catherine E. Coulson|David Lynch
10386  Mike Belzer|Tim Burton|Bonita DeCarlo|Greg Dyk...
10426  John Lennon|Paul McCartney|George Harrison|Rin...
10434                                                NaN
10550                                                NaN
10659  Sylvester Stallone|Henrietta Holm|Nicholas War...
10754                                                NaN

                                                director    runtime  \
371                                         Sanjay Patel   7.000000
424                                         Guido Chiesa 100.000000
441                                    Evgeny Afineevsky  98.000000
465                                          Adam Curtis 135.000000
532                                                  NaN  71.000000
536                                           David Gelb  90.000000
538                               Jennifer Siebel Newsom  88.000000
556                                                  NaN 105.000000
587                                           Kirby Dick 103.000000
600                                         Paul Johnson   7.000000
620    Antonio Padovan|Bryan Norton|Marc Roussel|Ryan...  90.000000
1032                                                 NaN  43.000000
1054                                                 NaN  60.000000
1088                         John Maloof|Charlie Siskel  83.000000
1173                                          Ian Cheney  71.000000
1177                                        Kyungmin Woo   5.000000
1190                                        Rory Kennedy  98.000000
1203                                                 NaN  75.000000
1208                           Kip Anderson|Keegan Kuhn  85.000000
1236                                      Jeffrey Radice 100.000000
1241                                                 NaN 102.070863
1256                                     Chris Moukarbel  79.000000
1288                                                 NaN  94.000000
1315                                        Amir Bar-Lev  98.000000
1316                                          Robert May 102.000000
1319                                   Jonathon Narducci  94.000000
1326             Tracy Droz Tragos|Andrew Droz Palermo  91.000000
1327                                       James Erskine  90.000000
1385                                          Jim Henson  48.000000
1712                                     Russell Mulcahy  88.000000
```

...	...	...
6760	Roger Allers	7.000000
6870	Suzie Templeton	32.000000
6930	Dony Permedi	102.070863
7579	NaN	92.000000
7650	Werner Herzog	99.000000
7723	Peter Joseph	118.000000
7767	NaN	60.000000
7813	Dylan Avery	129.000000
7814	NaN	22.000000
7905	Alvy Ray Smith	102.070863
8234	Carlo Verdone	103.000000
8292	Ron Fricke	96.000000
8614	James Cameron	12.000000
8824	Michael Dudok de Wit	8.000000
8878	Steve Boyum	85.000000
8880	Tim Burton	30.000000
9251	John Lasseter	102.070863
9307	Don Boyd	105.000000
9529	John Lasseter	5.000000
9564	Godfrey Reggio	99.000000
9593	NaN	52.000000
9677	John Lasseter	102.070863
9755	Martin Scorsese	6.000000
9799	David Lynch	5.000000
10386	NaN	25.000000
10426	NaN	55.000000
10434	David Lynch	102.070863
10550	John Lasseter	102.070863
10659	Morton Lewis	71.000000
10754	Marv Newland	102.070863

	genres	release_date \
371	Animation	11/25/15
424	NaN	10/29/15
441	Documentary	10/9/15
465	Documentary	1/24/15
532	Comedy	1/23/15
536	Documentary	10/8/15
538	Documentary	1/1/15
556	Comedy|Romance	1/24/15
587	Documentary	2/27/15
600	Science Fiction|Action|Animation	3/24/15
620	NaN	10/6/15
1032	TV Movie|Documentary	3/18/14
1054	TV Movie|Documentary	1/19/14
1088	Documentary	3/28/14
1173	Documentary	4/20/14

1177	Animation|Comedy|Science Fiction	5/8/14
1190	War|Documentary	9/5/14
1203	Documentary	3/19/14
1208	Documentary	7/1/14
1236	Documentary	1/20/14
1241	Romance|Music|Comedy	1/1/14
1256	Documentary	11/17/14
1288	Documentary	11/17/14
1315	Documentary	11/14/14
1316	Documentary|Thriller	2/7/14
1319	Documentary	4/6/14
1326	Documentary	1/19/14
1327	Documentary	2/17/14
1385	Drama|Comedy|Family	1/1/77
1712	NaN	2/27/09
...	...	...
6760	Drama|Animation	9/7/06
6870	Animation|Family|Music	9/23/06
6930	Animation|Action	1/1/06
7579	Horror|Thriller|Science Fiction|Mystery|Foreign	9/14/07
7650	Documentary	9/1/07
7723	Documentary|History	6/1/07
7767	Science Fiction|TV Movie	7/6/07
7813	Documentary	11/11/07
7814	Animation|Action|Thriller|Science Fiction	10/16/07
7905	Animation	12/17/84
8234	NaN	12/15/95
8292	Documentary	9/15/92
8614	NaN	1/1/96
8824	Animation|Drama	1/1/00
8878	NaN	10/13/00
8880	Animation	1/1/00
9251	Animation|Family	1/1/89
9307	NaN	8/26/89
9529	Animation|Family	8/1/88
9564	Documentary|Drama|Music	4/29/88
9593	Action|Adventure|Animation|Family|Fantasy	1/1/88
9677	Animation	8/17/87
9755	Drama	1/1/68
9799	NaN	1/1/74
10386	Documentary	10/3/93
10426	Music	12/25/67
10434	Animation	1/1/67
10550	Animation|Family	8/17/86
10659	NaN	2/10/70
10754	Animation|Comedy	1/1/69

```
      vote_count  vote_average  release_year     budget_adj     revenue_adj
```

371	47	6.900000	2015	17551039.822887	51364363.253251
424	21	6.100000	2015	17551039.822887	51364363.253251
441	37	8.200000	2015	17551039.822887	51364363.253251
465	19	7.800000	2015	17551039.822887	51364363.253251
532	14	6.600000	2015	17551039.822887	51364363.253251
536	12	8.000000	2015	17551039.822887	51364363.253251
538	11	8.900000	2015	17551039.822887	51364363.253251
556	14	6.500000	2015	17551039.822887	51364363.253251
587	39	7.800000	2015	17551039.822887	51364363.253251
600	29	7.600000	2015	17551039.822887	51364363.253251
620	13	5.000000	2015	17551039.822887	51364363.253251
1032	32	6.300000	2014	17551039.822887	51364363.253251
1054	11	7.200000	2014	17551039.822887	51364363.253251
1088	70	7.800000	2014	17551039.822887	1384967.241397
1173	14	6.900000	2014	17551039.822887	51364363.253251
1177	14	7.800000	2014	17551039.822887	51364363.253251
1190	20	5.900000	2014	17551039.822887	51364363.253251
1203	40	7.000000	2014	138163.657616	51364363.253251
1208	66	7.600000	2014	107.767653	51364363.253251
1236	13	6.700000	2014	17551039.822887	51364363.253251
1241	18	5.700000	2014	17551039.822887	51364363.253251
1256	22	6.800000	2014	17551039.822887	51364363.253251
1288	12	6.800000	2014	17551039.822887	51364363.253251
1315	14	6.000000	2014	17551039.822887	51364363.253251
1316	12	7.100000	2014	17551039.822887	51364363.253251
1319	13	7.000000	2014	17551039.822887	51364363.253251
1326	14	7.100000	2014	17551039.822887	51364363.253251
1327	11	6.500000	2014	17551039.822887	51364363.253251
1385	10	7.500000	1977	17551039.822887	51364363.253251
1712	57	7.400000	2009	17551039.822887	51364363.253251
...	...	...	...	...	...
6760	15	6.500000	2006	17551039.822887	51364363.253251
6870	11	6.700000	2006	2487839.064071	51364363.253251
6930	15	6.700000	2006	17551039.822887	51364363.253251
7579	10	4.900000	2007	17551039.822887	51364363.253251
7650	38	6.700000	2007	17551039.822887	51364363.253251
7723	104	6.900000	2007	7361680.078421	51364363.253251
7767	18	7.600000	2007	17551039.822887	51364363.253251
7813	12	5.100000	2007	6310.011496	6310.011496
7814	34	5.800000	2007	17551039.822887	51364363.253251
7905	32	5.300000	1984	17551039.822887	51364363.253251
8234	44	6.700000	1995	17551039.822887	51364363.253251
8292	89	7.600000	1992	6216097.018356	51364363.253251
8614	14	6.700000	1996	17551039.822887	51364363.253251
8824	18	6.900000	2000	17551039.822887	51364363.253251
8878	16	5.400000	2000	17551039.822887	51364363.253251
8880	10	6.300000	2000	17551039.822887	51364363.253251
9251	77	7.100000	1989	17551039.822887	51364363.253251

```
9307            10       5.300000          1989 17551039.822887 51364363.253251
9529            51       6.100000          1988 17551039.822887 51364363.253251
9564            18       7.200000          1988  4609727.557726  1086501.722010
9593            28       6.600000          1988 17551039.822887 51364363.253251
9677            44       6.600000          1987 17551039.822887 51364363.253251
9755            12       6.700000          1968 17551039.822887 51364363.253251
9799            11       5.000000          1974 17551039.822887 51364363.253251
10386           15       7.500000          1993 17551039.822887 51364363.253251
10426           15       5.800000          1967 17551039.822887 51364363.253251
10434           16       5.200000          1967     1307.352748 51364363.253251
10550           81       7.300000          1986 17551039.822887 51364363.253251
10659           10       3.000000          1970    28081.841720 51364363.253251
10754           12       5.600000          1969 17551039.822887 51364363.253251

[129 rows x 16 columns]
```

Data is missing in some rows but those are already padded with NaN or 0, so that won't be a problem.

1.1.6 Data Cleaning (Editing column names)

Renaming release_date as release_month, with only month instead of having date which might be irrelevent for the analysis.

```
In [51]: from datetime import datetime as d
         def time_change(data):
             return d.strftime(d.strptime(data, "%m/%d/%y"), "%B")

         # There is not much of a need with release date exactly, so we will only absorb relea
         tmdb_data["release_month"] = tmdb_data["release_date"].apply(time_change)

In [52]: tmdb_data.head()

Out[52]:        id    imdb_id  popularity           budget            revenue  \
         0  135397  tt0369610   32.985763 150000000.000000 1513528810.000000
         1   76341  tt1392190   28.419936 150000000.000000  378436354.000000
         2  262500  tt2908446   13.112507 110000000.000000  295238201.000000
         3  140607  tt2488496   11.173104 200000000.000000 2068178225.000000
         4  168259  tt2820852    9.335014 190000000.000000 1506249360.000000

                      original_title  \
         0                 Jurassic World
         1            Mad Max: Fury Road
         2                     Insurgent
         3  Star Wars: The Force Awakens
         4                     Furious 7

                                                      cast          director  \
         0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...   Colin Trevorrow
```

```
1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...     George Miller
2  Shailene Woodley|Theo James|Kate Winslet|Ansel...  Robert Schwentke
3  Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...       J.J. Abrams
4  Vin Diesel|Paul Walker|Jason Statham|Michelle ...         James Wan

     runtime                                    genres release_date  \
0 124.000000  Action|Adventure|Science Fiction|Thriller       6/9/15
1 120.000000  Action|Adventure|Science Fiction|Thriller      5/13/15
2 119.000000         Adventure|Science Fiction|Thriller      3/18/15
3 136.000000   Action|Adventure|Science Fiction|Fantasy     12/15/15
4 137.000000                     Action|Crime|Thriller       4/1/15

   vote_count  vote_average  release_year        budget_adj        revenue_adj  \
0        5562      6.500000          2015 137999939.280026 1392445892.523800
1        6185      7.100000          2015 137999939.280026  348161292.489031
2        2480      6.300000          2015 101199955.472019  271619025.407628
3        5292      7.500000          2015 183999919.040035 1902723129.801820
4        2947      7.300000          2015 174799923.088033 1385748801.470520

  release_month
0          June
1           May
2         March
3      December
4         April
```

Exploratory Data Analysis ### Research Question 1 >### During years, how are runtime, popularity and average are trending?

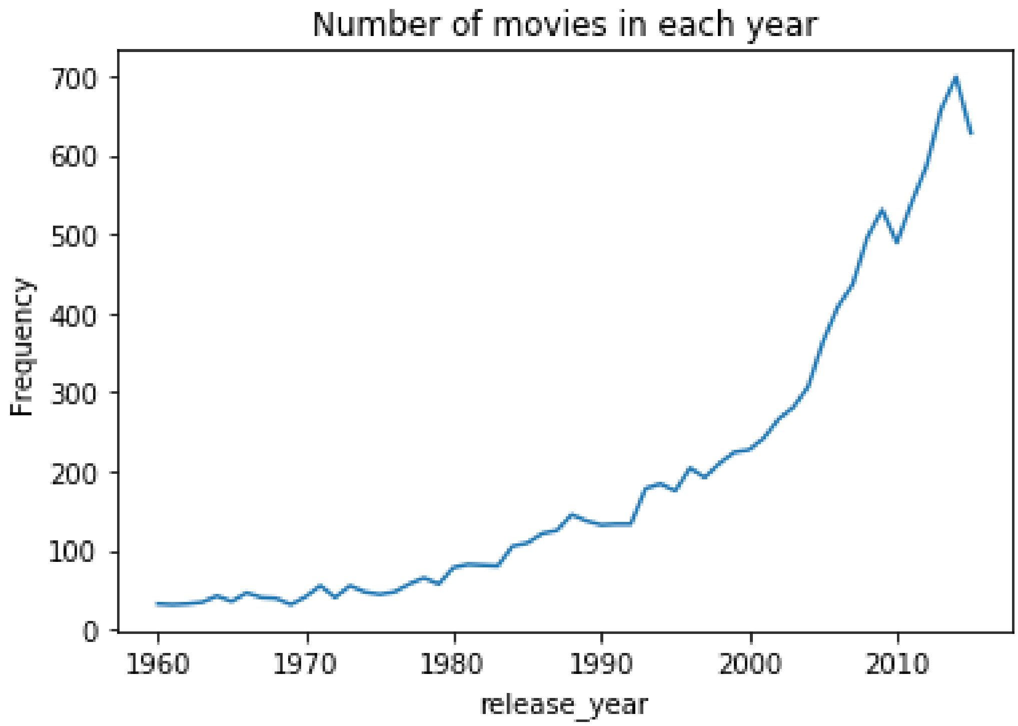
```
In [53]: years_data = tmdb_data.groupby("release_year").mean()
```

Now we have movies grouped by their respective release_year, now we can answer the question

Lets observe how many movies are in each year.

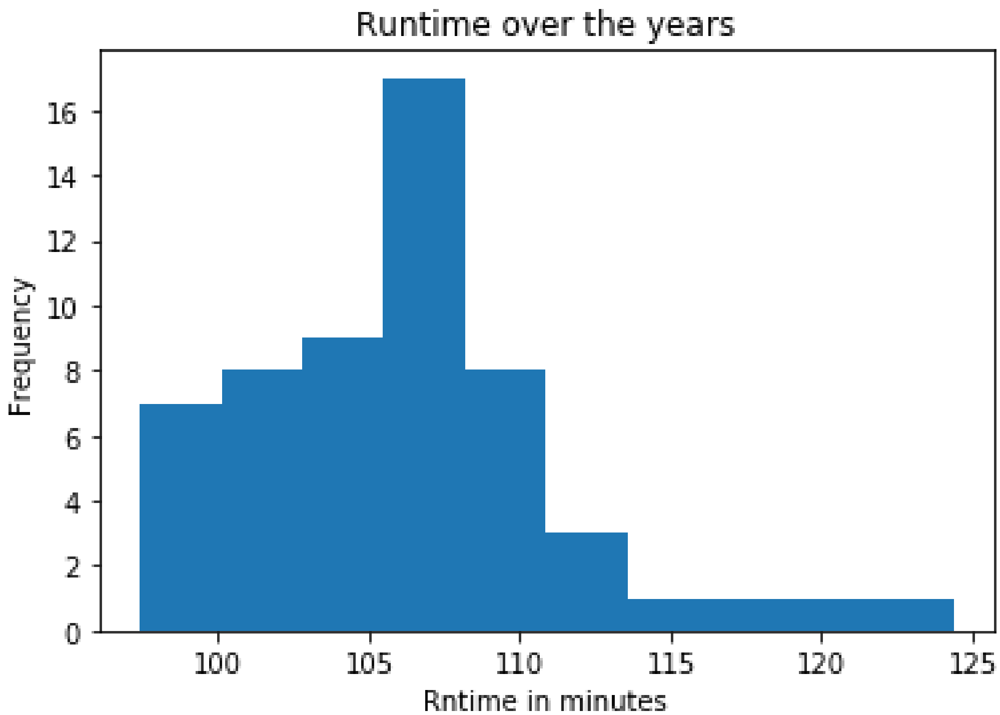
```
In [54]: # To get number of movies in each year
         movie_year_count = tmdb_data.groupby("release_year").describe()['budget']['count']
```

```
In [55]: movie_year_count.plot( title='Number of movies in each year')
         plt.ylabel("Frequency");
```



It can be clearly inferred that the number of movies has been increased drastically after 1990 >#### Runtime over the years

In [16]:
```
years_data['runtime'].plot.hist()
plt.xlabel("Rntime in minutes")
plt.title("Runtime over the years");
```




```
In [17]: years_data['runtime'].describe()
```

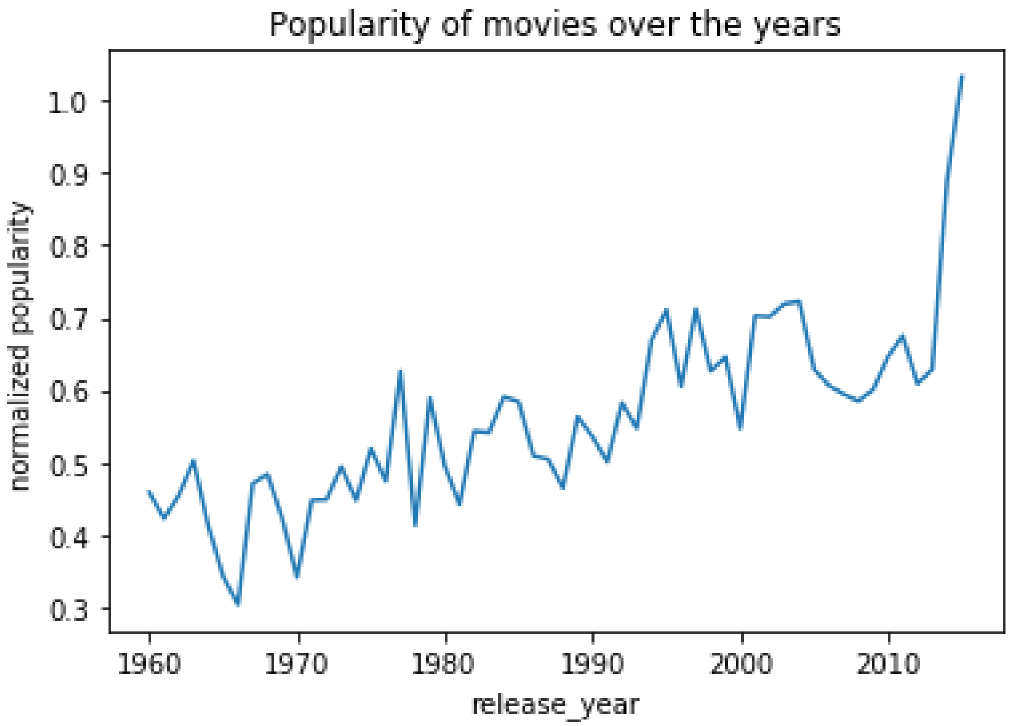
```
Out[17]: count     56.000000
         mean     106.143784
         std        5.318237
         min       97.405117
         25%      102.120755
         50%      105.785868
         75%      108.851172
         max      124.343750
         Name: runtime, dtype: float64
```

Observation

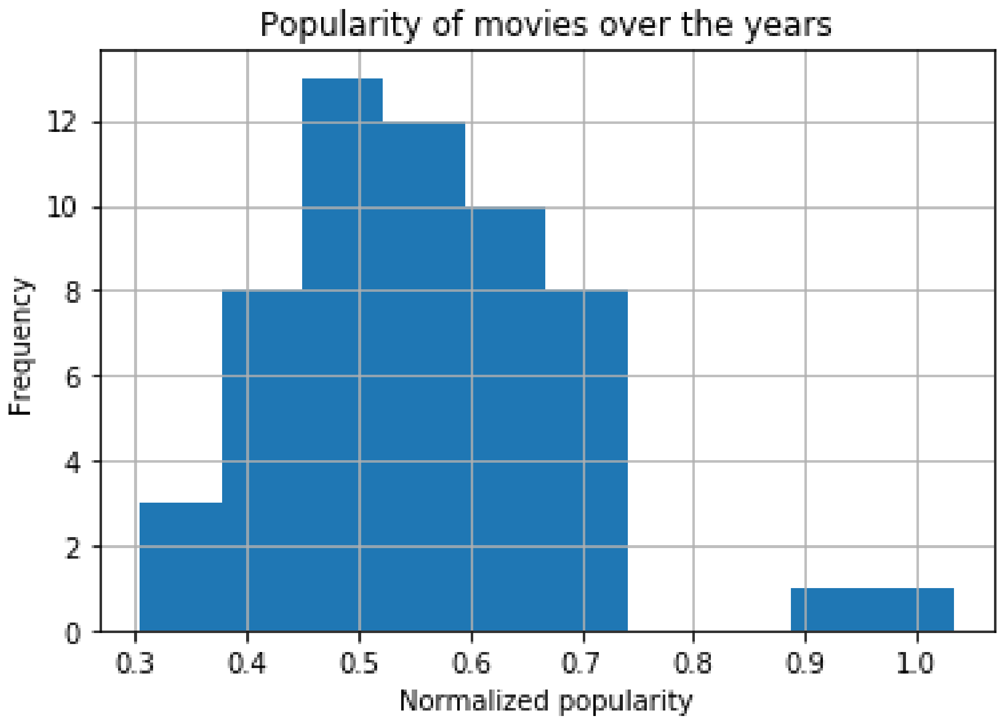
It can be observed that the runtime over the years is not much changed and mostly lied between 102 and 108 minutes.
Distribution is right skewed.

>#### Popularity over the years

```
In [18]: years_data['popularity'].plot()
         plt.ylabel('normalized popularity')
         plt.title("Popularity of movies over the years");
```




```
In [19]: years_data['popularity'].hist(bins=10)
         plt.xlabel('Normalized popularity')
         plt.ylabel("Frequency")
         plt.title("Popularity of movies over the years");
```




```
In [20]: years_data['popularity'].describe()
```

```
Out[20]: count    56.000000
         mean      0.559691
         std       0.128433
         min       0.304112
         25%       0.469625
         50%       0.546928
         75%       0.626934
         max       1.032126
         Name: popularity, dtype: float64
```

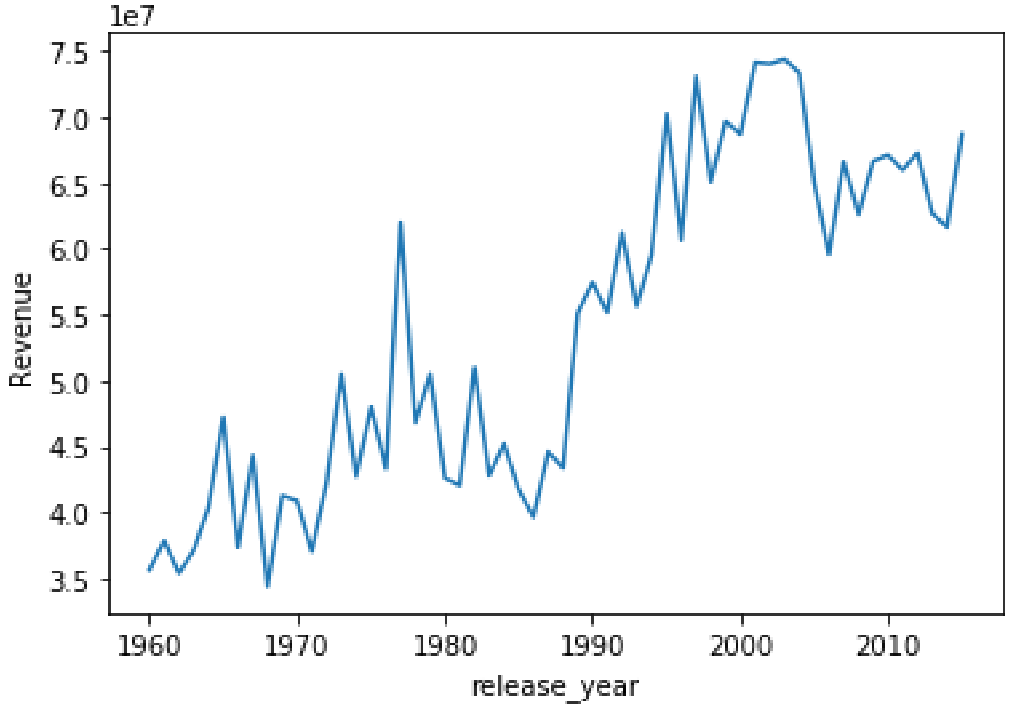
Observations It can be observed that the popularity for movies has been increasing gradually and there was a sudden rise in popularity for movies after 2010. Rise of social media and promotions through it explains the rise.

The distribution is right skewed

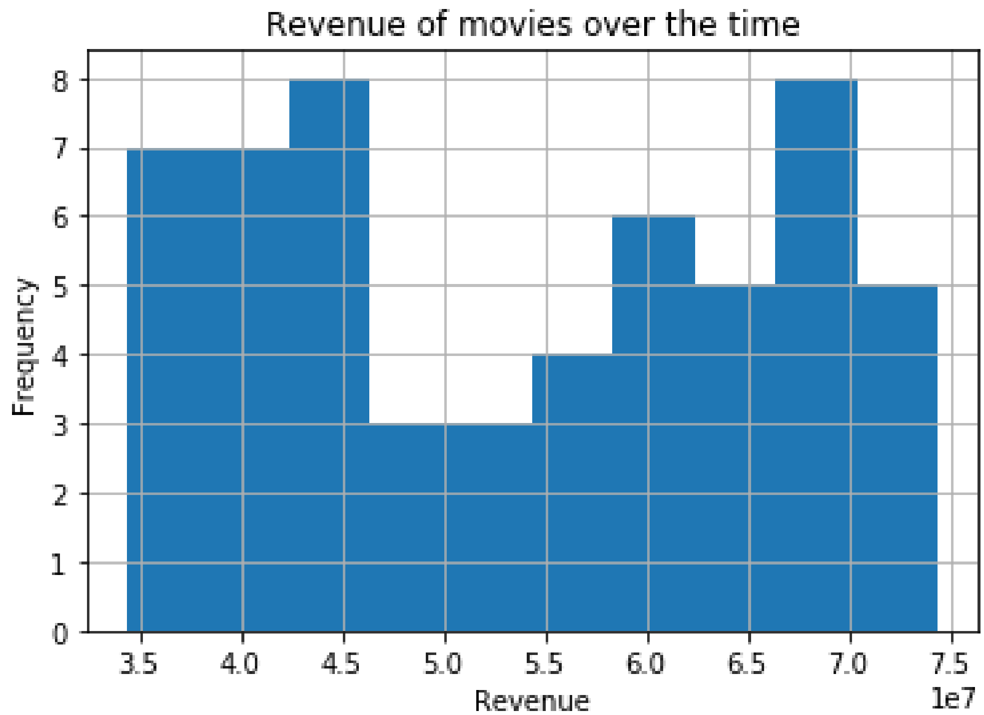
It is also observed that there are no movies between 0.7 and 0.9 popularity rating which is questionable

>### Revenue over the years

```
In [32]: years_data['revenue'].plot()
         plt.ylabel('Revenue');
```




```
In [22]: years_data['revenue'].hist()
         plt.xlabel("Revenue")
         plt.ylabel("Frequency")
         plt.title("Revenue of movies over the time");
```




```
In [23]: pd.set_option('float_format', '{:f}'.format)
         years_data['revenue'].describe()
```

```
Out[23]: count         56.000000
         mean    53746908.783461
         std     12647350.408113
         min     34358015.754400
         25%     42499050.058302
         50%     53070665.178745
         75%     65339138.993231
         max     74422152.605131
         Name: revenue, dtype: float64
```

Observations The observations that can be inferred are the revenue from movies acquired over the years is mostly concentrated between 4.5 billion to 7 billion dollars

The distribution is left skewed.

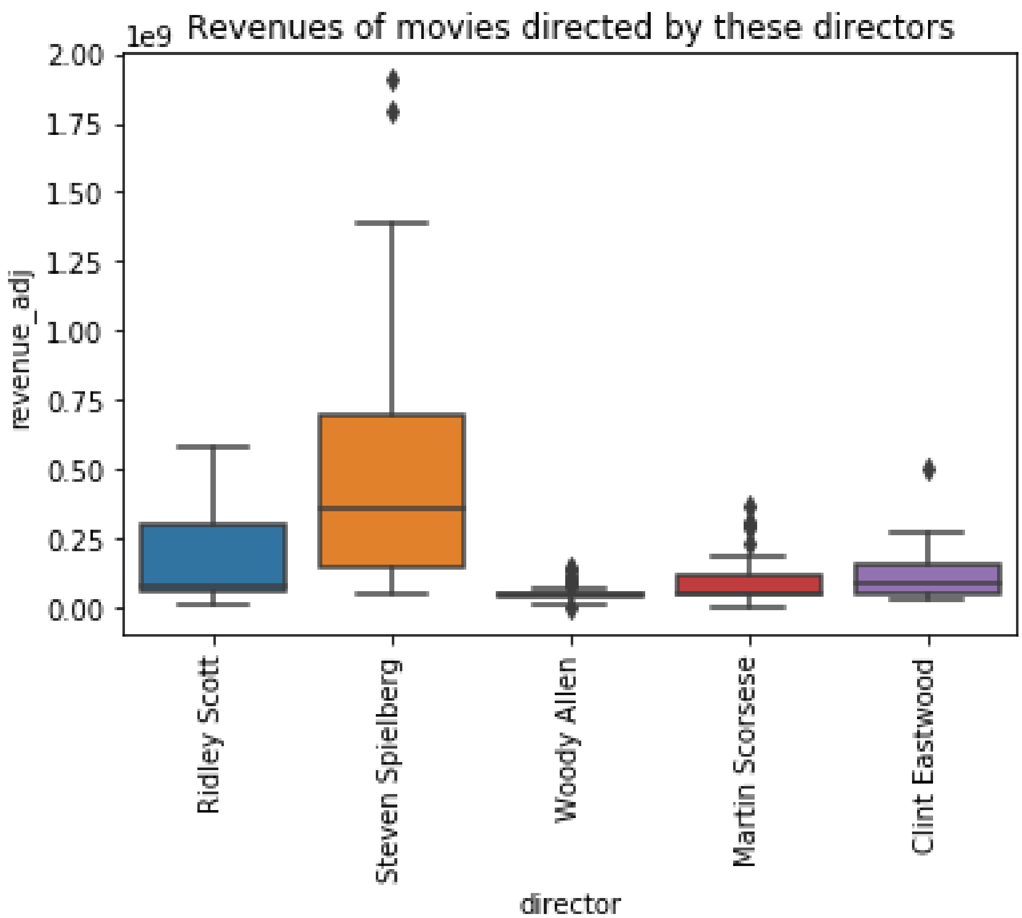
It can also be observed that the revenue is increased considerably after 1990.

Research Question 2

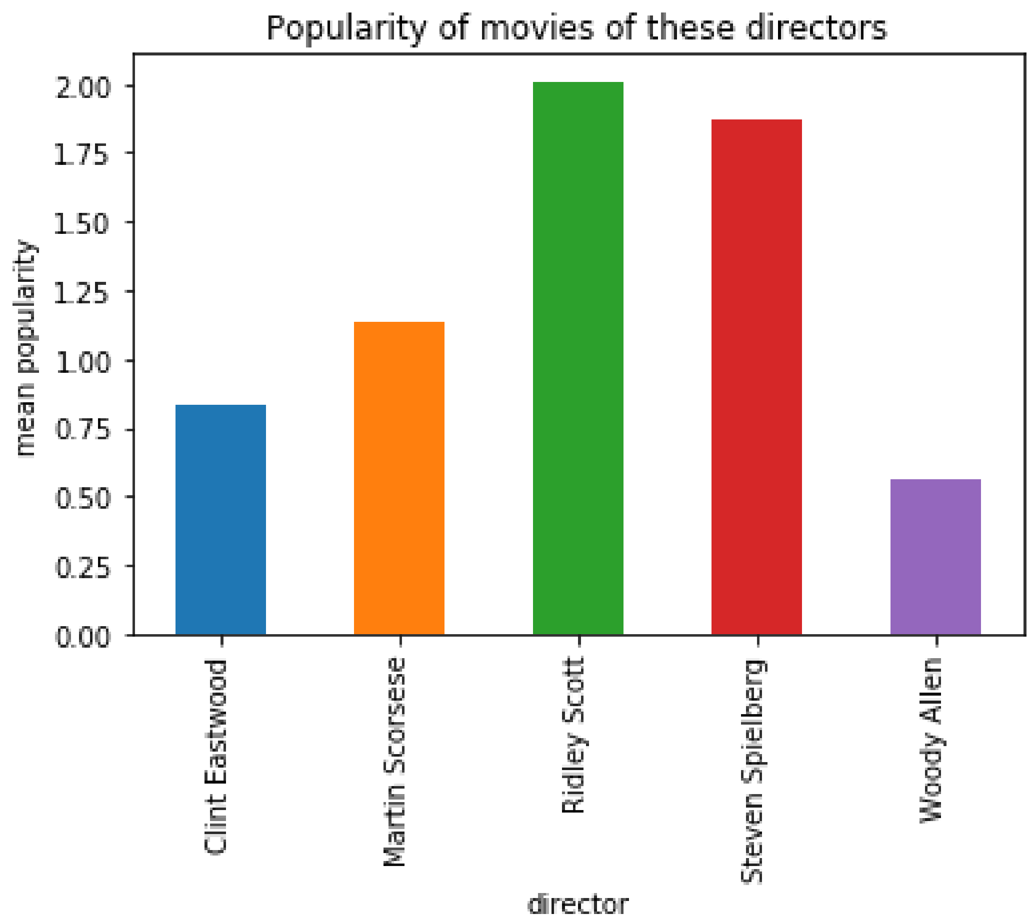
1.1.7 What are some important factors that are effecting the revenue of movies

How top directors are influencing revenue and who is best?

In [36]:
```
# Getting the directors name who has most movies in their name
dir_5 = tmdb_data['director'].value_counts().sort_values().index[-5:]

# collecting all the movies that are directed by these directors
director_analysis = tmdb_data.loc[(tmdb_data.director.isin(dir_5))]

# Box Plot
bx_plt = sns.boxplot(x='director',y='revenue_adj',data=director_analysis)
bx_plt.set_xticklabels(bx_plt.get_xticklabels(), rotation=90)
bx_plt.set_title("Revenues of movies directed by these directors")
bx_plt.plot();
```



In [34]:
```
# To see and analyze other variables
d = director_analysis.groupby('director').mean()
d['popularity'].plot(kind='bar', title='Popularity of movies of these directors')
plt.ylabel("mean popularity");
```



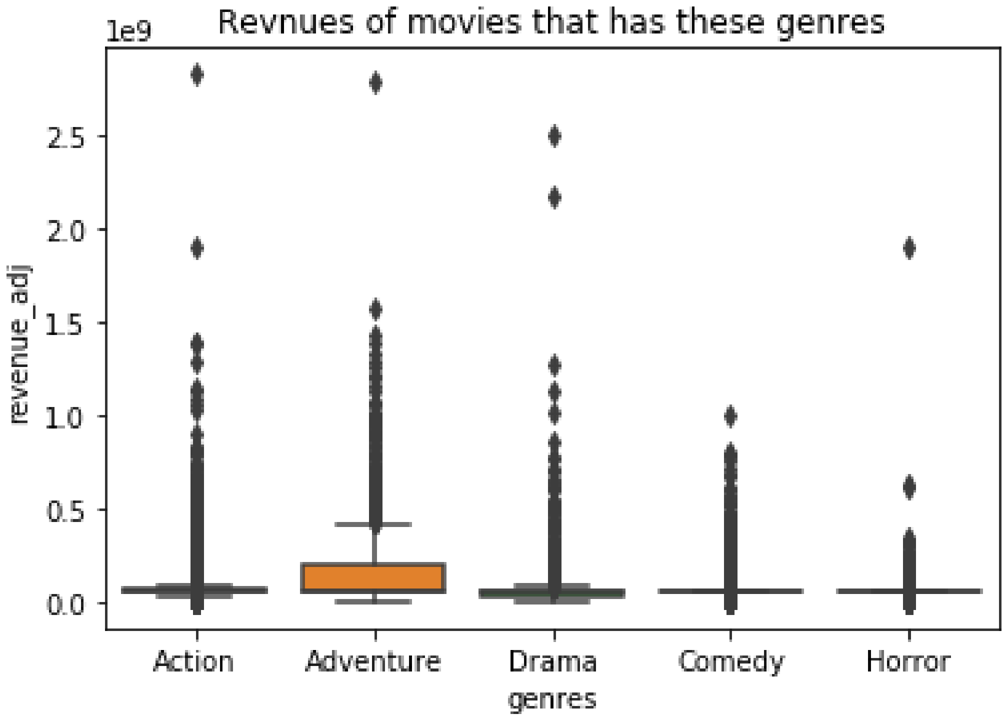
1.1.8 Observation

Steven Spielberg is wearing the crown being a director to create movies with most revenue than any director
Ridley Scott is second best director, concering revenue
It can be observed that the movies of Steven Spielberg and Ridley Scott are ecpected to have high revenue
Interesting to observe that the popularity for Ridley Scott is more that that of Steven Spielberg movies

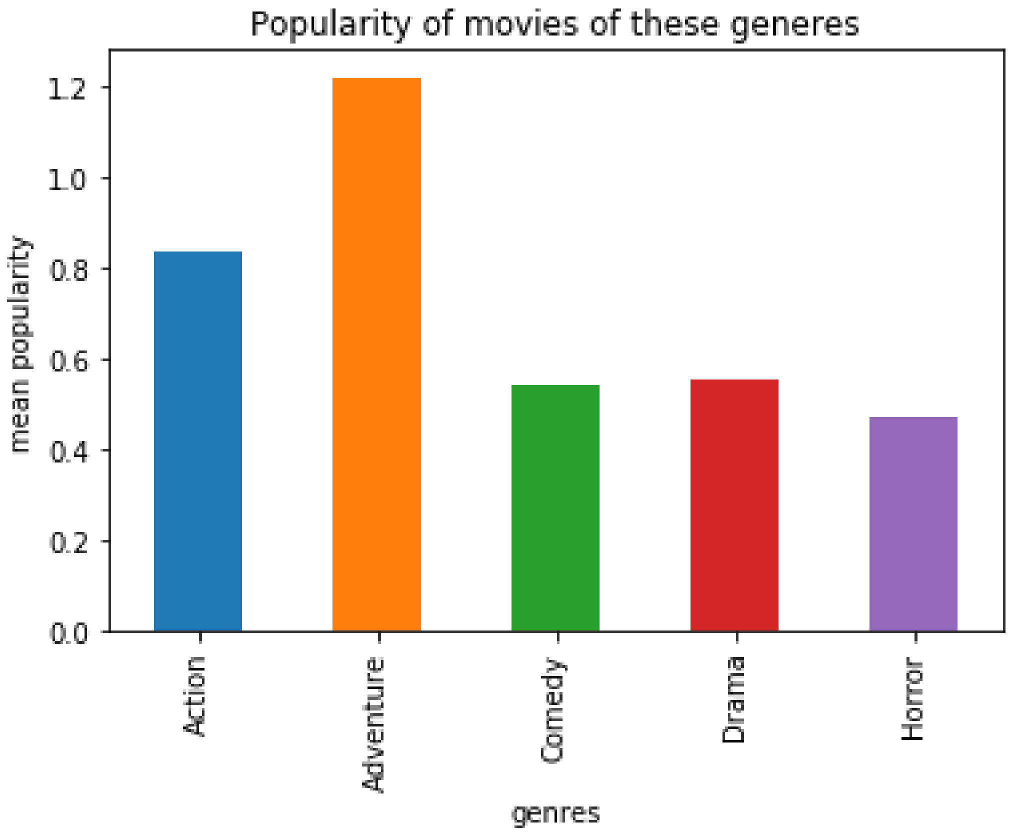
How genre is influencing revenue and which genre is best?

In [26]:
```
# having a copy of orginal data
genre_data = tmdb_data.copy()

# As genres are seperated by '|' we will split then and consider only the first genre
genre_analysis = genre_data['genres'].str.split('|', expand=True)#[0].value_counts().
```

```
genre_data['genres'] = genre_data['genres'].str.split('|', expand=True)[0]#.value_cou

# Collecting only the top 5 genres with most frequency in movies
gen_5 = genre_analysis.groupby(0).size().sort_values().index[-5:]

# Movies with genre which has anyone of the most frequent genres
genre_analysis = genre_data.loc[(genre_data.genres.isin(gen_5))]

# Boxplot
bx_plt = sns.boxplot(x='genres',y='revenue_adj',data=genre_analysis)
bx_plt.set_title("Revnues of movies that has these genres")
bx_plt.plot();
```



In [27]:
```
# To see and analyze other variables
d = genre_analysis.groupby('genres').mean()
plt.ylabel("mean popularity")
d['popularity'].plot(kind='bar', title='Popularity of movies of these generes');
```



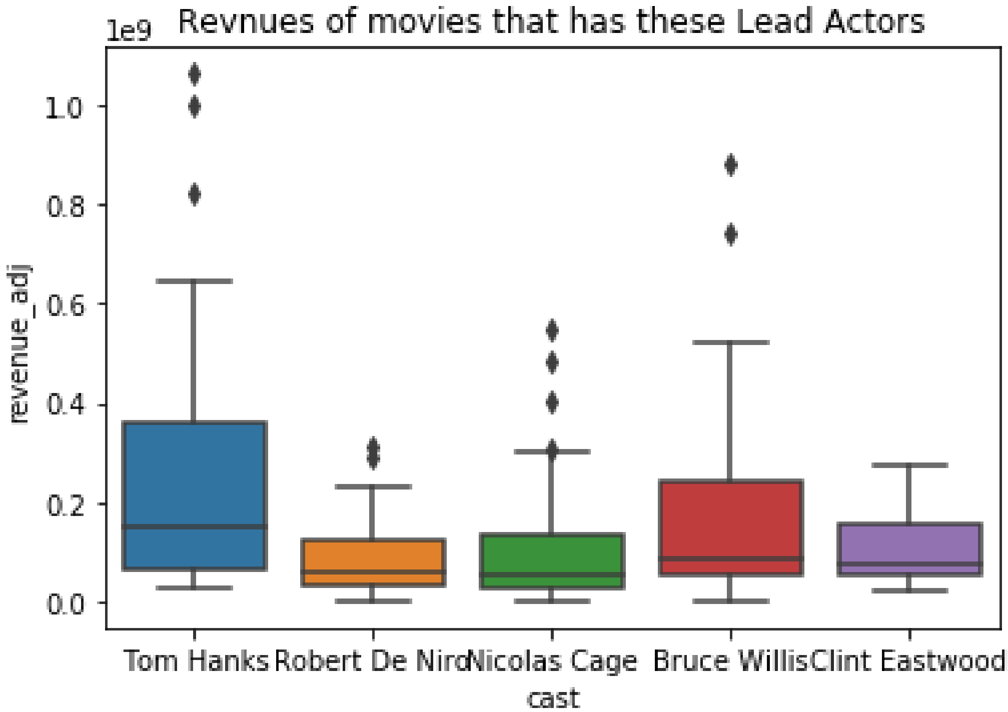
1.1.9 Observation

Movies with Adventure as genre is having most revenue than any genre
Action is second best genre, concering revenue
To second the first observation, even popularity for adventure movies is higher than any other genre

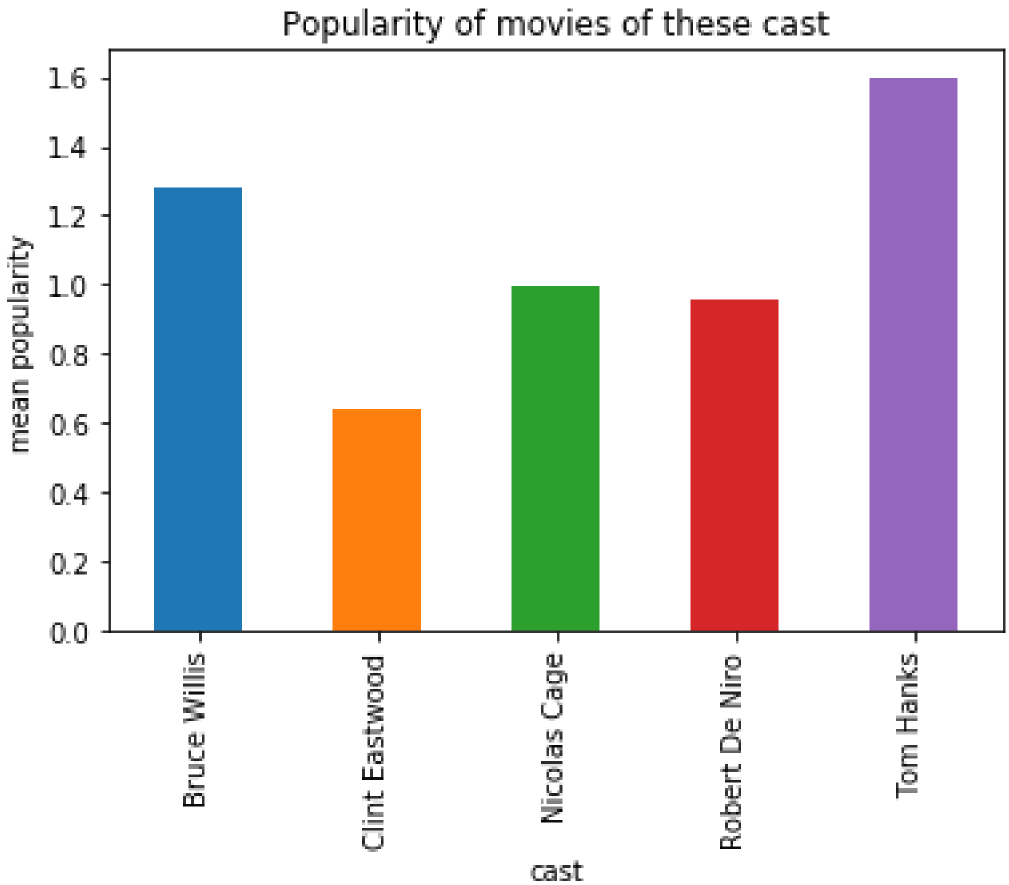
How lead actor is influencing revenue and which Actor is best, concerning revenues?

```
In [28]: # having a copy of orginal data
         cast_data = tmdb_data.copy()

         # As casts are seperated by '|' we will split then and consider only the first genre
         cast_analysis = cast_data['cast'].str.split('|', expand=True)
         cast_data['cast'] = cast_data['cast'].str.split('|', expand=True)

         # Collecting only the top 5 genres with most frequency in movies
         cast_5 = cast_analysis.groupby(0).size().sort_values().index[-5:]

         # Movies with genre which has anyone of the most frequent genres
```

```
cast_analysis = cast_data.loc[(cast_data.cast.isin(cast_5))]

# Boxplot
bx_plt = sns.boxplot(x='cast',y='revenue_adj',data=cast_analysis)
bx_plt.set_title("Revnues of movies that has these Lead Actors")
bx_plt.plot();
```




```
In [29]: # To see and analyze other variables
         d = cast_analysis.groupby('cast').mean()
         plt.ylabel("mean popularity")
         d['popularity'].plot(kind='bar', title='Popularity of movies of these cast');
```



1.1.10 Observation

Movies with Tom Hanks as lead actor are having most revenue than any actors

Bruce Wills is second best actor, concering revenue

To second the first observation, even popularity for Tom Hanks acted movies is higher than any other actor

Conclusions

Limitations Some of the data is incomplete and replaced by mean, so the analysis is tentative

The data contains 21 columns and most of them are irrelevant moreover we got only 10561 rows and the data in many of them were incomplete

Having an extra column stating how much money was spent on marketing, given the amount invested in reaching out to poeple may have influenced the revenue.

Voting and popularity may have sourced from irrelevant sources so that can be a limitation

Having cast and genres in the same column as a collection is a bit difficult, but I attempted to solve it by taking first genre and lead actor into consideration.

Some Analysis Trend: Over the years from 1960 to 2015:

Runtime has been considerately same and there was not much difference

Popularity was increased drastically, this can be explained with advent of social network promotions.

Revenue collected from movies has been decreased in recent movies than earlier movies from 1960-1990 period

Factors: Good revenue in movies

Director plays a crucial role in creating a good movie, so Steven Spielberg movies were considered to get good revenue than other directors

Adventure movies has highest revenue than any other genre

Movies with Tom Hanks as lead actor got more revenue than any other actor.

Final Words First question, how is the trend for movies over the years, we can understand that over the years runtime has been considrate over the years.

Revenue was decreased over the years, this can be explained as the number of movies over the time also increased so people have lot more options and may not spend on single movie.

popularity obviosly increased due to advent of internet and social network or may be better marketing strategies.

Second questions, the factors that influenced good revenue.

The analysis I perfermed suggests that a proved good director will perform good in his next movie.

Movie genre greatlty influences a movie and from my analysis, adventure turned out to be best genre interms of revenue generation

A good cast or actors definetly influences a movie, considering lead actor, my analysis gave an insight that some actors are great influencers in creating a good revenue

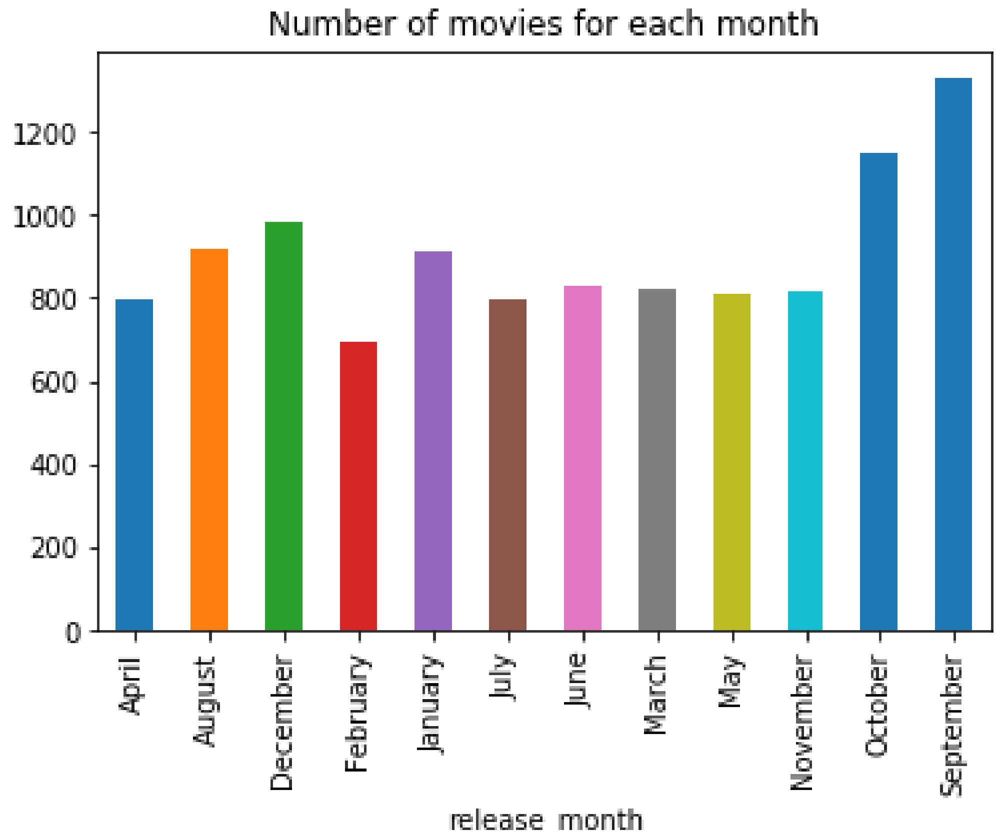
While the correlations above might not be the exact causes and further statistical tests needed to be performed on those variables to prove the point.

1.1.11 Some more interesting analysis

Months to have most number of releases.

```
In [30]: month_analysis =  tmdb_data.groupby("release_month")
         month_analysis.describe()['budget']['count'].plot.bar(title="Number of movies for eacl
```

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x23eb441db38>
```



References: https://stackoverflow.com was used extensively for many queries in using pandas. https://pandas.pydata.org was used as documentation reference.