Investigating a Dataset

August 6, 2022

1 Project: Investigating a Dataset - IMDB Movie Dataset from Kaggle

2 Table of Content

3 Introduction

This data set contains information about 10,000 movies collected from The Movie Database (TMDb), including user ratings and revenue. Certain columns, like 'cast' and 'genres', contain multiple values eparated by pipe (|) characters. The final two columns ending with "_adj" show the budget and revenue of the associated movie in terms of 2010 dollars, accounting for inflation over time.

4 Questions to ask

The main goal of this analysis would be to explore the movie dataset and see what trends can be identified. To do this, we would ask questions like: 1. Which movie has the highest profit 2. What movie is the most popular 3. Which Actor is the most popular 4. Which studio makes the most films 5. Which genre is the most profitable? 6. Which director is the most popular?

---

[1]:
```
#Importing Modules
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from importlib import reload
import datetime
import seaborn as sns
```

[2]:
```
#Reading the Dataset
mov = pd.read_csv('tmdb-movies.csv')
```

5 Data Wrangling

The first step taken is to check the structure of the data.To do this, I would use general pandas methods to explore the data.

[3]:
```
#Checking the sample of the data to get a general idea of the structure
mov.head()
```

[3]:
```
       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           6.5          2015  1.379999e+08  1.392446e+09
1           7.1          2015  1.379999e+08  3.481613e+08
2           6.3          2015  1.012000e+08  2.716190e+08
3           7.5          2015  1.839999e+08  1.902723e+09
4           7.3          2015  1.747999e+08  1.385749e+09

[5 rows x 21 columns]
```

[4]:
```
#Checking the general Information of the data to see the data-types and␣
 ↪non-null count
mov.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
```

[5]:
```
#Checking the shape of the data
mov.shape
```

[5]:
```
(10866, 21)
```

From the .info() method called, we can see the characteristics of every column with the non-null count and the type of data. From the non-null count we see that there are some columns with missing data. This would be addressed later. This dataset 10866 rows and 21 columns. Some columns are not useful to the analysis and would be dropped.

A summary of columns - id- This is a unique identifier for each movie. - imdb_id- This is a unique identifier for each movie from the IMDB website - popularity- This is a measure of how popular a movie is - budget- The budget for the movie - revenue- Revenue made from the movie - original_title- The title of the movie - cast- The actors in the movie - homepage- The websiste for the movie - director- The director of the movie - tagline- The tagline of the movie - keywords- Words used to identofy the movie - overview- The description of the movid - runtime- Total runtime in minutes - genres- The genres that the movie belong to - production_companies- Companies that produce the movie - release_date- Date the movie was released - vote_count- Number of people that voted for the movie - vote_average- Average voting of each movie - release_year- The year the movie was released - budget_adj- Adjusted budget to 2010 dollar value - revenue_adj- Adjusted revenue to 2010 dollar value

[6]:
```
#Describing the statistics of the movie
mov.describe()
```

[6]:

	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

From the dataset description, some problems can identify. - Problem 1: The IMBD id column is not included because it contains letters. - Problem 2: From the popularity column, 75% of the dataset have popularity below or equals to 0.7, while the maximum number is 32. This signifies the existence of outliers - Problem 3: More than 50% of the figures in the popularity and bidget columns as well as their adjusted counterpart have the value of 0 -Problem 4: The release year is interpreted as a figure and not a date-time type. All these problem would be addressed in the data cleaning section.

[7]:
```
#Checking all the null value in the dataset
mov.isnull().sum()
```

[7]:
```
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
```

The most of the null values are in the homepage column. This column is going to be deleted. Some other non-needed columns would be deleted while other useful columns would be filled

[8]:
```
#Checking the number of unique values in each columns
mov.nunique()
```

[8]:
```
id                      10865
imdb_id                 10855
popularity              10814
budget                    557
revenue                  4702
original_title          10571
cast                    10719
homepage                 2896
director                 5067
tagline                  7997
keywords                 8804
overview                10847
runtime                   247
genres                   2039
production_companies     7445
release_date             5909
vote_count               1289
vote_average               72
release_year               56
budget_adj               2614
revenue_adj              4840
dtype: int64
```

There are 10866 rows in the dataset. There are 10865 unique id in the dataset, this implies that one row is duplicated. It would be dropped in the cleaning phase. One would expect that original_title would tally with the id but it doesn't. This could indicate that one movie has two different identification number.

[9]:
```
#Viewing the duplicated columns
mov[mov.duplicated(['original_title'])].sort_values('original_title')
```

[9]:
```
          id    imdb_id  popularity    budget   revenue  \
5748  176068  tt2395385    0.336081         0         0
6514   98622  tt0443424    0.128484         0         0
7917   13189  tt0087056    0.542315         0         0
2489   16716  tt0216621    0.521573         0         0
7891     377  tt0087800    1.331432   1800000  25504513
…      …        …          …          …        …
4124   39227  tt0324532    0.212966         0         0
3636    9364  tt1181614    0.414629   8000000    100915
8332   25095  tt0104181    0.269621         0         0
7423    1949  tt0443706    1.464555  65000000  84785914
9893   14433  tt0058777    0.306872         0         0

                 original_title  \
```

```
5748                          1
6514                          9
7917          A Christmas Carol
2489          A Christmas Carol
7891  A Nightmare on Elm Street
…                           …
4124               When in Rome
3636          Wuthering Heights
8332          Wuthering Heights
7423                     Zodiac
9893                       Zulu

                                                    cast  \
5748  Rhys Wakefield|Logan Miller|Ashley Hinshaw|Nat…
6514                                                NaN
7917  George C. Scott|Roger Rees|David Warner|Susann…
2489  Patrick Stewart|Richard E. Grant|Joel Grey|Ian…
7891  John Saxon|Ronee Blakley|Heather Langenkamp|Am…
…                                                   …
4124  Mary-Kate Olsen|Ashley Olsen|Leslie Danon|Juli…
3636  Kaya Scodelario|James Northcote|Amy Wren|Nicho…
8332  Juliette Binoche|Ralph Fiennes|Jeremy Northam|…
7423  Jake Gyllenhaal|Robert Downey Jr.|Mark Ruffalo…
9893  Stanley Baker|Jack Hawkins|Ulla Jacobsson|Jame…

                                              homepage           director  \
5748                                               NaN     Dennis Iliadis
6514                           http://www.shaneacker.com        Shane Acker
7917                                               NaN      Clive Donner
2489                                               NaN  David Hugh Jones
7891                                               NaN         Wes Craven
…                                                  …                …
4124                                               NaN     Steve Purcell
3636  http://www.artificial-eye.com/film.php?cinema=…     Andrea Arnold
8332                                               NaN   Peter Kosminsky
7423                                               NaN     David Fincher
9893                                               NaN       Cy Endfield

                                             tagline  …  \
5748                                Everyone wants one.  …
6514                                                NaN  …
7917  A new powerful presentation of the most loved …  …
2489                                                NaN  …
7891  If Nancy Doesn't Wake Up Screaming, She Won't …  …
…                                                   …  …
4124                                                NaN  …
3636                          Love is a force of nature.  …
```

```
8332  A passion. An obsession. A love that destroyed…  …
7423  There's more than one way to lose your life to…  …
9893  Dwarfing the mightiest! Towering over the grea…  …

                                                 overview runtime  \
5748  Three college friends go to the biggest party …      95
6514  A rag doll fights a monster that has been stea…      11
7917  An old bitter miser who makes excuses for his …     100
2489  Scrooge is a miserly old businessman in 1840's…      95
7891  Teenagers in a small town are dropping like fl…      91
…                                                    …     …
4124  Teenage sisters Charli and Lola are on the ver…      94
3636  A poor boy of unknown origins is rescued from …     129
8332  Young orphan Heathcliff is adopted by the weal…     105
7423  The true story of the investigation of 'The Zo…     157
9893  In 1879, during the Zulu wars, man of the peop…     138

                                    genres  \
5748             Thriller|Science Fiction
6514                    Animation|Fantasy
7917  TV Movie|Fantasy|Drama|Comedy|Family
2489                         Drama|Fantasy
7891                                Horror
…                                      …
4124  Action|Adventure|Comedy|Drama|Family
3636                         Drama|Romance
8332                         Drama|Romance
7423         Crime|Drama|Mystery|Thriller
9893             Action|Drama|History|War

                                    production_companies release_date  \
5748                                Process Productions      9/20/13
6514                                                NaN      4/21/05
7917                       Entertainment Partners Ltd.      12/17/84
2489  Turner Network Television (TNT)|Hallmark Enter…      12/5/99
7891                 New Line Cinema|Smart Egg Pictures     11/15/84
…                                                    …          …
4124                                                NaN     11/26/02
3636                    Ecosse Films|Film4 Productions       9/5/11
8332                               Paramount Pictures     10/16/92
7423   Paramount Pictures|Warner Bros.|Phoenix Pictures       3/2/07
9893                                    Diamond Films      1/22/64

     vote_count  vote_average  release_year    budget_adj   revenue_adj
5748         49           5.1          2013  0.000000e+00  0.000000e+00
6514         49           7.3          2005  0.000000e+00  0.000000e+00
7917         29           6.6          1984  0.000000e+00  0.000000e+00
```

```
2489         23           6.1          1999  0.000000e+00  0.000000e+00
7891        625           7.1          1984  3.778276e+06  5.353504e+07
…           …             …            …              …             …
4124         38           5.7          2002  0.000000e+00  0.000000e+00
3636         36           5.6          2011  7.755184e+06  9.782680e+04
8332         30           7.0          1992  0.000000e+00  0.000000e+00
7423       1042           7.1          2007  6.835846e+07  8.916668e+07
9893         65           6.6          1964  0.000000e+00  0.000000e+00

[295 rows x 21 columns]
```

Upon further inspection, while they may share attributes such as title and sometimes storyline, they have different attributes such as actors and directors

---

6 Data Cleaning

In this part of the analysis process, I am going to be cleaning the data, droping unnessery columns, adjusting and removing outliers

7 Dropping duplicates

[10]:
```
#Dropping duplicates
mov.drop_duplicates(inplace=True)

#Confirming changes
mov.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10865 entries, 0 to 10865
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    10865 non-null  int64
 1   imdb_id               10855 non-null  object
 2   popularity            10865 non-null  float64
 3   budget                10865 non-null  int64
 4   revenue               10865 non-null  int64
 5   original_title        10865 non-null  object
 6   cast                  10789 non-null  object
 7   homepage              2936 non-null   object
 8   director              10821 non-null  object
 9   tagline               8041 non-null   object
 10  keywords              9372 non-null   object
 11  overview              10861 non-null  object
 12  runtime               10865 non-null  int64
 13  genres                10842 non-null  object
```

```
 14  production_companies  9835 non-null   object
 15  release_date          10865 non-null  object
 16  vote_count            10865 non-null  int64
 17  vote_average          10865 non-null  float64
 18  release_year          10865 non-null  int64
 19  budget_adj            10865 non-null  float64
 20  revenue_adj           10865 non-null  float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.8+ MB
```

8 Dropping columns that are not needed

The following colunms would be dropped imdb_id: since each movie has an id, there is no need for the imdb_id budget: there is a value for adjusted budget revenue: there is a value for adjusted revenue homepage: missing the most data and is not necessary to the analysis tagline: not needed for the analysis overview: not needed for the analysis

[11]:
```
#Droping columns
drop_columnms = ['imdb_id', 'budget', 'revenue', 'homepage', 'tagline',␣
 ↪'overview']
mov = mov.drop(columns=drop_columnms)
#Confirming drop
mov.head()
```

[11]:

	id	popularity	original_title	cast	director	keywords	runtime	genres
0	135397	32.985763	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…	Colin Trevorrow	monster|dna|tyrannosaurus rex|velociraptor|island	124	Action|Adventure|Science Fiction|Thriller
1	76341	28.419936	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic…	George Miller	future|chase|post-apocalyptic|dystopia|australia	120	
2	262500	13.112507	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel…	Robert Schwentke	based on novel|revolution|dystopia|sequel|dyst…	119	
3	140607	11.173104	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D…	J.J. Abrams	android|spaceship|jedi|space opera|3d	136	
4	168259	9.335014	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle …	James Wan	car race|speed|revenge|suspense|car	137	

```
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           6.5          2015  1.379999e+08  1.392446e+09
1           7.1          2015  1.379999e+08  3.481613e+08
2           6.3          2015  1.012000e+08  2.716190e+08
3           7.5          2015  1.839999e+08  1.902723e+09
4           7.3          2015  1.747999e+08  1.385749e+09
```

Now the columns have been dropped. I'm going to be adjusting the data format for the dates

9 Adjusting the data format for the date

[12]:
```
#Converting to datetime
mov['release_date'] = pd.to_datetime(mov['release_date'])

#Confirming change
mov.info()
mov.sort_values(['release_date'])
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10865 entries, 0 to 10865
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    10865 non-null  int64
 1   popularity            10865 non-null  float64
 2   original_title        10865 non-null  object
 3   cast                  10789 non-null  object
 4   director              10821 non-null  object
 5   keywords              9372 non-null   object
 6   runtime               10865 non-null  int64
 7   genres                10842 non-null  object
 8   production_companies  9835 non-null   object
 9   release_date          10865 non-null  datetime64[ns]
 10  vote_count            10865 non-null  int64
 11  vote_average          10865 non-null  float64
```

```
 12  release_year          10865 non-null  int64
 13  budget_adj            10865 non-null  float64
 14  revenue_adj           10865 non-null  float64
dtypes: datetime64[ns](1), float64(4), int64(4), object(6)
memory usage: 1.3+ MB
```

[12]:

	id	popularity	original_title	cast	director	keywords	runtime	genres
7280	15516	0.384193	The Last House on the Left	Sandra Peabody|Lucy Grantham|David Hess|Fred J…	Wes Craven	rape|gun|birthday|concert|death of a child	84	Horror|Thriller
7283	26119	0.338541	Chato's Land	Charles Bronson|Jack Palance|James Whitmore|Si…	Michael Winner	rape|posse|dead horse|canteen|carcass	110	
7287	15573	0.265891	The Cowboys	John Wayne|Roscoe Lee Browne|Bruce Dern|Collee…	Mark Rydell	boy|beef|cattle drive|rancher|cattle	131	
7272	10784	0.669045	Cabaret	Liza Minnelli|Michael York|Helmut Griem|Joel G…	Bob Fosse	berlin|sex|entertainer|cabaret|gramophone	124	
7304	40022	0.152463	Milano Calibro 9	Gastone Moschin|Barbara Bouchet|Mario Adorf|Fr…	Fernando Di Leo	organized crime	100	
…	…	…	…	…	…	…	…	
9977	32082	0.000188	The Hospital	George C. Scott|Diana Rigg|Richard Dysart|Barn…	Arthur Hiller	hospital|malpratice	103	
9923	185	3.072555	A Clockwork Orange	Malcolm McDowell|Patrick Magee|Adrienne Corri|…	Stanley Kubrick	prison|street gang|rape|adolescence|sexuality	136	
9933	343	0.625188	Harold and Maude	Ruth Gordon|Bud Cort|Cyril Cusack|Charles Tyne…	Hal Ashby	suicide|life and death|depression|age differen…	91	
9927	984	0.885391	Dirty Harry	Clint Eastwood|Harry Guardino|Reni Santoni|Joh…	Don Siegel	ambush|san francisco|detective|ransom|stadium	102	
9929	994	0.735185	Straw Dogs	Dustin Hoffman|Susan George|Peter Vaughan|T. P…	Sam Peckinpah	england|rape|country life|primal fear|revenge	118	

```
7283        Action|Adventure|Western
7287  Action|Adventure|Drama|Western
7272             Drama|Music|Romance
7304                   Crime|Action
…                                …
9977            Mystery|Comedy|Drama
9923          Science Fiction|Drama
9933            Comedy|Drama|Romance
9927           Action|Crime|Thriller
9929     Crime|Drama|Thriller|Mystery

                                  production_companies release_date  \
7280  Lobster Enterprises|Sean S. Cunningham Films|T…   1972-01-01
7283                                    Scimitar Films   1972-01-01
7287                                      Warner Bros.   1972-01-13
7272  ABC Pictures|Bavaria Film|Allied Artists Pictures   1972-02-13
7304                                               NaN   1972-02-15
…                                                  …          …
9977                               Simcha Productions   2071-12-14
9923                           Hawk Films|Warner Bros.   2071-12-18
9933                               Paramount Pictures   2071-12-20
9927                      Warner Bros.|Malpaso Company   2071-12-22
9929                                    ABC Pictures   2071-12-29

      vote_count  vote_average  release_year    budget_adj   revenue_adj
7280          46           6.1          1972  4.693105e+05  0.000000e+00
7283          15           6.6          1972  0.000000e+00  0.000000e+00
7287          30           6.6          1972  0.000000e+00  3.910921e+07
7272          76           6.9          1972  3.128737e+07  0.000000e+00
7304          11           6.6          1972  0.000000e+00  0.000000e+00
…          …            …            …            …            …
9977          10           6.4          1971  0.000000e+00  0.000000e+00
9923        1786           7.7          1971  1.184743e+07  1.431869e+08
9933         112           7.4          1971  6.462234e+06  0.000000e+00
9927         300           7.2          1971  2.154078e+07  1.937378e+08
9929          84           6.8          1971  1.184743e+07  1.751154e+07

[10865 rows x 15 columns]
```

Unfortunately, in the converted date, 2071 is gotten as a year when data from the .describe() method used earlier showed the maximum year as 2015. This is due to the date formatted as YY instead of YYYY. I need to write a function to change that would subtract all the years above 2015 by 100 to get the correct year

[13]:
```
#Confirming the dates above 2016
mov[mov['release_date'] > '2016-01-01'].head(10)
```

[13]:

	id	popularity	original_title	cast	director	keywords	runtime
9719	62	3.309196	2001: A Space Odyssey	Keir Dullea|Douglas Rain|Gary Lockwood|William…	Stanley Kubrick	moon|jupiter|artificial intelligence|man vs ma…	149
9720	871	1.152937	Planet of the Apes	Charlton Heston|Roddy McDowall|Kim Hunter|Maur…	Franklin J. Schaffner	human evolution|gorilla|bondage|space marine|c…	112
9721	10331	1.058272	Night of the Living Dead	Duane Jones|Judith O'Dea|Karl Hardman|Marilyn …	George A. Romero	loss of father|siblings|midnight movie|zombies	96
9722	26690	0.891163	The Boston Strangler	Tony Curtis|Henry Fonda|George Kennedy|Mike Ke…	Richard Fleischer	detective|double life|boston|strangle|serial m…	116
9723	8069	0.867192	Barbarella	Jane Fonda|John Phillip Law|Anita Pallenberg|M…	Roger Vadim	sexual fantasy|alien planet|distant future|les…	98
9724	916	0.786854	Bullitt	Steve McQueen|Jacqueline Bisset|Robert Vaughn|…	Peter Yates	san francisco|hotel|detective|based on novel|a…	113
9725	14136	0.757746	The Love Bug	Dean Jones|Michele Lee|Buddy Hackett|Joe Flynn…	Robert Stevenson	car race|suicide attempt|sport|golden gate bri…	107
9726	18988	0.747337	The Lion in Winter	Peter O'Toole|Katharine Hepburn|Anthony Hopkin…	Anthony Harvey	england|infidelity|famous score|queen|castle	134
9727	805	0.724527	Rosemary's Baby	Mia Farrow|John Cassavetes|Ruth Gordon|Sidney …	Roman Polanski	commercial|anti-christ|contemporary setting|la…	136
9728	12105	0.698337	Yellow Submarine	Paul Angelis|John Lennon|Paul McCartney|George…	George Dunning	submarine|fab four|colours|music|blase	90

```
                                genres  \
9719  Science Fiction|Mystery|Adventure
9720  Adventure|Science Fiction|Mystery
9721                            Horror
9722       Crime|Drama|Mystery|Thriller
9723                    Science Fiction
9724        Action|Crime|Drama|Thriller
9725              Comedy|Family|Fantasy
9726                     Drama|History
9727               Horror|Drama|Mystery
9728            Animation|Fantasy|Music

                                  production_companies release_date  \
9719  Stanley Kubrick Productions|Metro-Goldwyn-Maye…   2068-04-05
9720  Twentieth Century Fox Film Corporation|APJAC P…   2068-02-07
9721  Laurel Group|Off Color Films|Image Ten|Market …   2068-10-01
9722            Twentieth Century Fox Film Corporation   2068-10-08
9723  Dino de Laurentiis Cinematografica|Marianne Pr…   2068-10-10
9724       Solar Productions|Warner Brothers/Seven Arts   2068-10-17
9725                           Walt Disney Productions   2068-12-22
9726          AVCO Embassy Pictures|Haworth Productions   2068-01-01
9727      Paramount Pictures|William Castle Productions   2068-06-12
9728                Apple Corps|King Features Production   2068-07-17

      vote_count  vote_average  release_year    budget_adj   revenue_adj
9719        1708           7.7          1968  7.522756e+07  3.555466e+08
9720         469           7.2          1968  3.635999e+07  2.093547e+08
9721         279           7.2          1968  7.146619e+05  0.000000e+00
9722          15           6.6          1968  0.000000e+00  0.000000e+00
9723          63           5.4          1968  0.000000e+00  0.000000e+00
9724         143           6.7          1968  3.447930e+07  2.651826e+08
9725          62           5.8          1968  0.000000e+00  0.000000e+00
9726          37           6.9          1968  0.000000e+00  0.000000e+00
9727         448           7.3          1968  2.006068e+07  2.093547e+08
9728          63           7.0          1968  0.000000e+00  0.000000e+00
```

There are more than 400 entries that need to be corrected

[14]:
```python
def fix_year(date):
    '''This function subtracts a century from date above 2016'''
    if date.year > 2016:
        year = date.year - 100
    else:
        year = date.year
    return pd.to_datetime(datetime.date(year, date.month, date.day))
```

```
mov['release_date'] = mov['release_date'].apply(fix_year)
```

[15]:
```
#Checking the conversion
mov['release_date']
mov.info()
mov[mov['release_date'] > '2016-01-01']
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10865 entries, 0 to 10865
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    10865 non-null  int64
 1   popularity            10865 non-null  float64
 2   original_title        10865 non-null  object
 3   cast                  10789 non-null  object
 4   director              10821 non-null  object
 5   keywords              9372 non-null   object
 6   runtime               10865 non-null  int64
 7   genres                10842 non-null  object
 8   production_companies  9835 non-null   object
 9   release_date          10865 non-null  datetime64[ns]
 10  vote_count            10865 non-null  int64
 11  vote_average          10865 non-null  float64
 12  release_year          10865 non-null  int64
 13  budget_adj            10865 non-null  float64
 14  revenue_adj           10865 non-null  float64
dtypes: datetime64[ns](1), float64(4), int64(4), object(6)
memory usage: 1.3+ MB
```

[15]:
```
Empty DataFrame
Columns: [id, popularity, original_title, cast, director, keywords, runtime,
genres, production_companies, release_date, vote_count, vote_average,
release_year, budget_adj, revenue_adj]
Index: []
```

The date has been corrected and confirmed.

10 Correcting outliers on the popularity column

More than 75% of values in the popularity columns are below 0.75. The maximum in the column is 32.9. running the code `mov[mov['popularity'] > 1].describe()` It was discovered that 1756 entries are above 1 and 75% of them are below 2. It would have been easier to drop these columns, but they contain recent and popular movies. I would write a function to normalise the data.

[16]:
```
mov[mov['popularity'] > 1].describe()
```

[16]:
```
                  id   popularity      runtime   vote_count  vote_average  \
count    1756.000000  1756.000000  1756.000000  1756.000000   1756.000000
mean    58178.353645     2.101489   110.608200   979.667995      6.387358
std     87624.072436     1.836949    22.872874  1115.521459      0.783752
min         5.000000     1.000194     0.000000    10.000000      3.300000
25%      2162.250000     1.207331    96.000000   304.000000      5.900000
50%     11004.500000     1.552851   108.000000   600.500000      6.400000
75%     73776.500000     2.294167   122.000000  1172.000000      6.900000
max    417859.000000    32.985763   366.000000  9767.000000      8.400000

       release_year    budget_adj   revenue_adj
count   1756.000000  1.756000e+03  1.756000e+03
mean    2003.872437  5.748676e+07  2.201003e+08
std       10.850642  5.452117e+07  2.799882e+08
min     1960.000000  0.000000e+00  0.000000e+00
25%     1999.000000  1.655467e+07  4.233404e+07
50%     2007.000000  4.000231e+07  1.307588e+08
75%     2012.000000  8.374146e+07  2.894069e+08
max     2015.000000  3.683713e+08  2.827124e+09
```

[17]:
```
def normalise(pop):
    '''This function normalises extreme values in the popularity column'''
    if pop < 1:
        return pop
    elif pop < 10:
        return pop/10
    elif pop > 10:
        return pop/100

mov['popularity'] = mov['popularity'].apply(normalise)
```

[18]:
```
#Checking the change
mov['popularity'].describe()
```

[18]:
```
count    10865.000000
mean         0.339257
std          0.231501
min          0.000065
25%          0.154214
50%          0.280055
75%          0.474818
max          0.999866
Name: popularity, dtype: float64
```

[19]:

```
#Filling all the columms
#fill_columms = {'cast': "Not Available",  'director': "Not Available",
 ↪'keywords': "Not Available", 'genres' :"Not Available",
 ↪'production_companies':"Not Available"}
#mov.fillna(value=fill_columms, inplace=True)

#Checking the fill
#mov.isnull().sum()
```

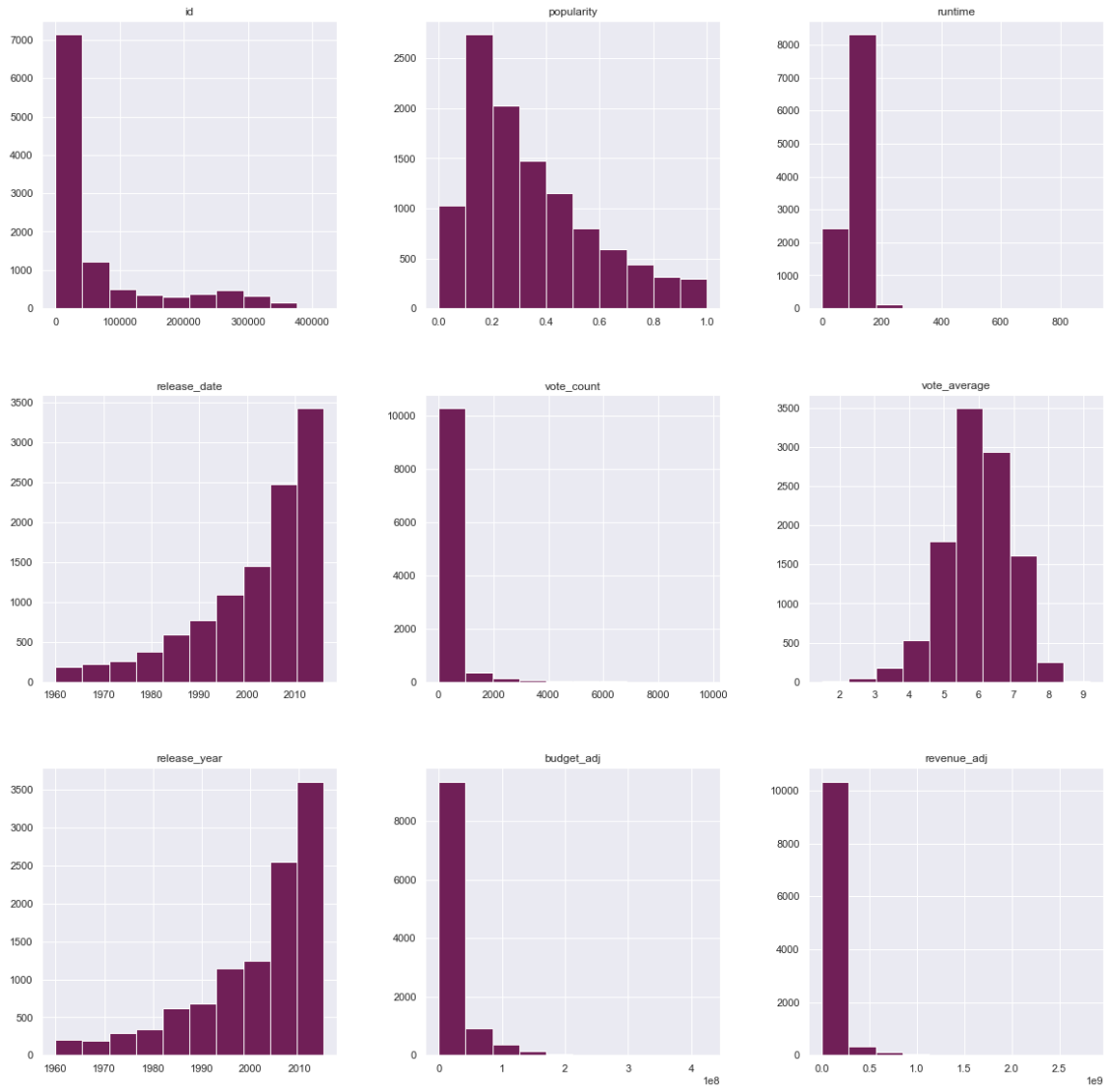
---

11 Exploratory Data Analysis

We are going to explore the dataset to see general trends and patterns

[20]:
```
%matplotlib inline
#Initializing Seaborn
sns.set_style('darkgrid')
sns.set(rc={"figure.figsize": (15,15)})
```

[21]:
```
#Creating a general histogram to see the distribution clearly
mov.hist(figsize=[20,20], color='#701F57');
```



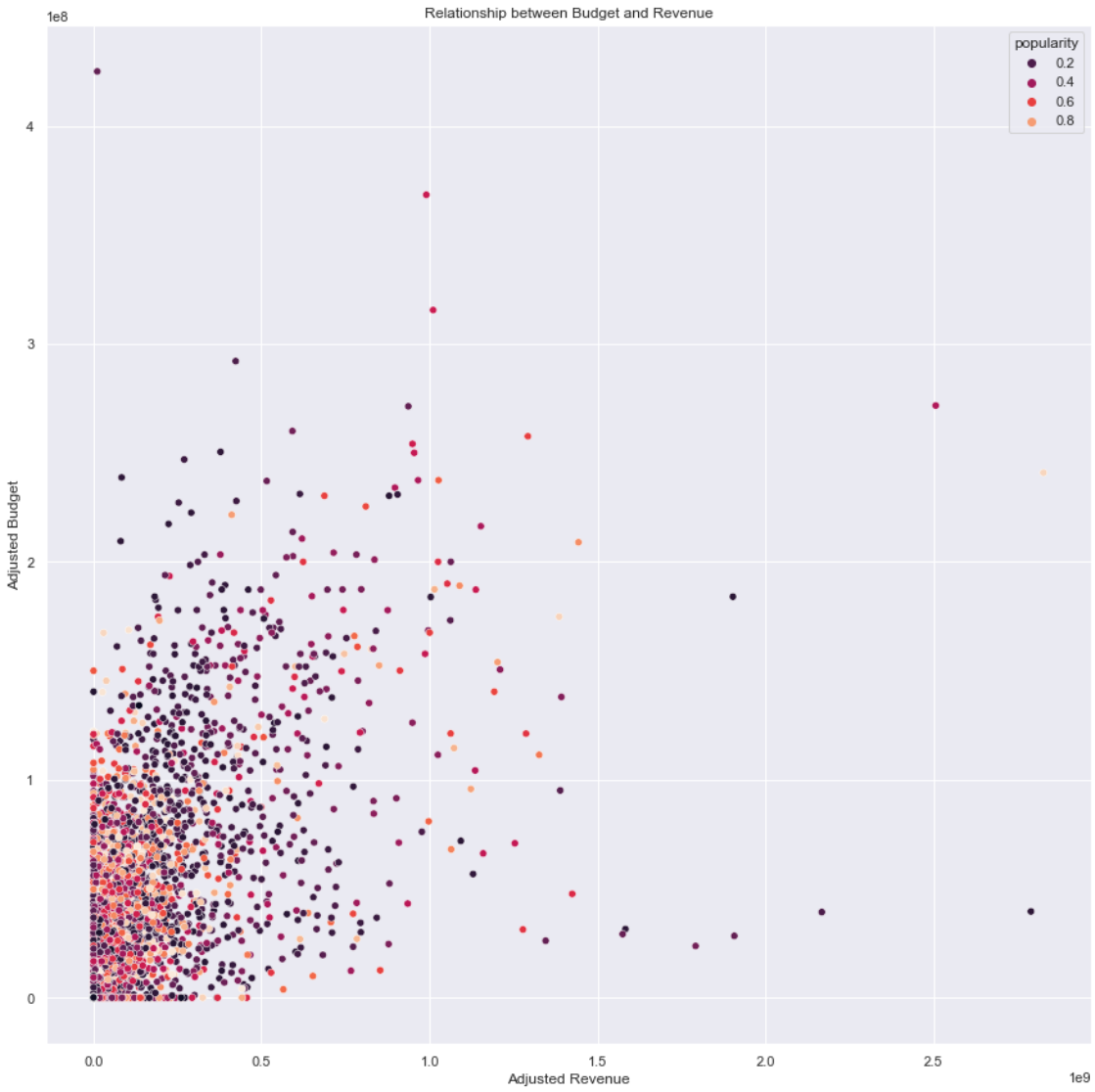
From the dataset histogram plot, we can see the general distribution of the data. A few key findings are: - The popularity column is right skewed, - Most movies have a run time of less than 200 minutes, - Most movies received a less than 1000 people vote, - The vote average is a normal distribution, - The number of movies increases as the years do, - Most movies have budget less than 50,000,000 dollars - Most movies made revenue of less than 250,000,000 dollars

[22]:
```
#Creating a general correlation matrix to see if any numeric figure correlates
mov.corr().style.background_gradient(cmap='rocket')
```

[22]:
```
<pandas.io.formats.style.Styler at 0x238534faeb0>
```

The correlation matrix shows us that there's a strong positive relationship between - Adjusted Budget and Adjusted Revenue - Release year and Id - Vote count and Adjusted budget - Vote count and Adjusted revenue

[23]:
```
#Creating a scatterplot to view clearly the relationship between budget revenue␣
 ↪and popularity
ax = sns.scatterplot(data=mov, y='budget_adj', x='revenue_adj',␣
 ↪hue='popularity', palette='rocket')
ax.set(xlabel='Adjusted Revenue', ylabel='Adjusted Budget', title='Relationship␣
 ↪between Budget and Revenue');
```

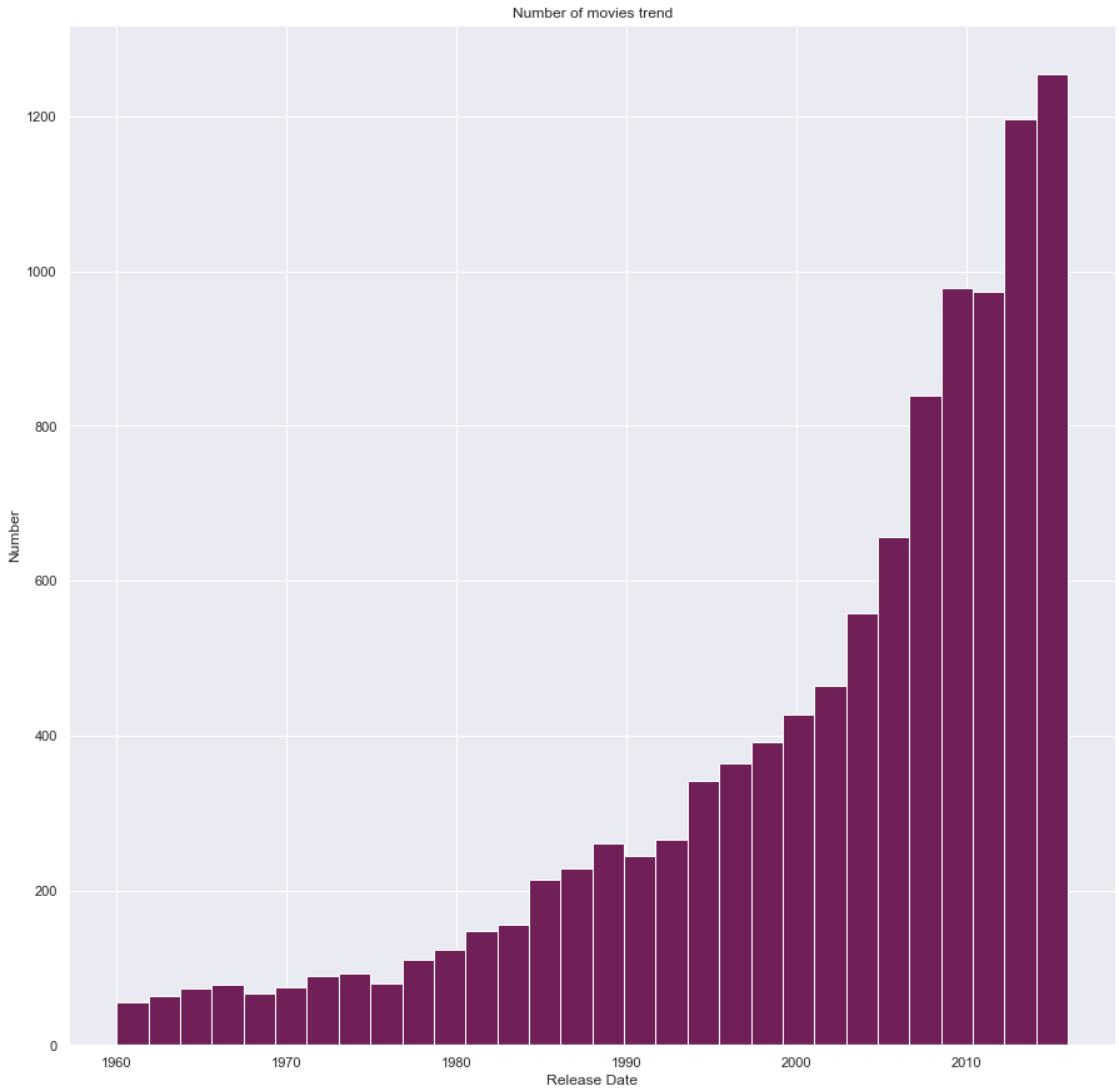


[24]:
```
#Creating a scatterplot to view clearly the relationship between budget revenue␣
 ↪and average votes
ax = sns.scatterplot(data=mov, y='budget_adj', x='revenue_adj',␣
 ↪hue='vote_average', palette='rocket')
```

```
ax.set(xlabel='Adjusted Revenue', ylabel='Adjusted Budget', title='Relationship␣
↪between Budget and Revenue');
```

There is a positive correlation between Revenue and Budget. This means that when the budget increases, revenue also increases. Higher voter averages are seen in movies that have larger budgets and revenue

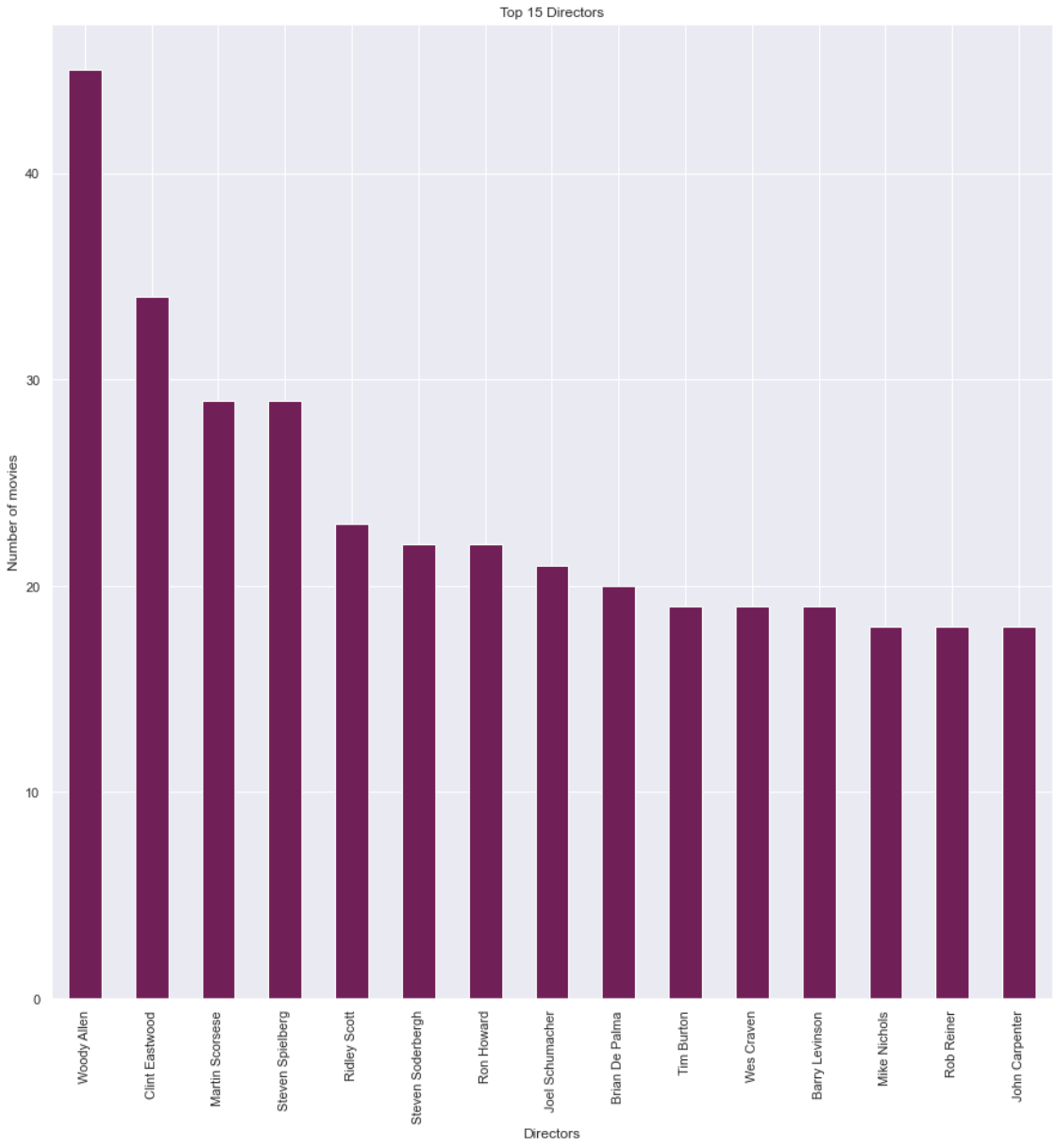
[25]:
```
#Creating a histogram to see the distribution on movies by their released date
ax = sns.histplot(mov['release_date'], color='#701F57', bins=30, alpha=1)
ax.set(xlabel='Release Date', ylabel='Number', title='Number of movies trend');
```



Now to answer some important questions # 1: Relationships with Directors This section tries to answer questions such as which director has the most movies.

[26]:
```
#Getting the top 15 directors
Top_15_dir = mov['director'].value_counts()[:15]
Top_15_dir


#Plotting the top 15 directors
plt=reload(plt)
Top_15_dir.plot(kind='bar',color='#701F57');

#Labelling the Data
plt.ylabel('Number of movies')
```

```
plt.title('Top 15 Directors')
plt.xlabel('Directors');
```



The director with the most movies is Woody Allen with 45 movies directed followed by Clint Eastwood and Martin Scorsese

12 2: Relationships with Genres

This section tries to answer questions such as which genre is the most popular. To answer genre specific questions, we'd have to split and explode the genre column as movies fall into multiple

genres.

[27]:
```
#Copying the data to create a new dataset that would be exploded
mov_genre = mov.copy()

#Splitting by the string '|'
mov_genre['genres'] = mov_genre['genres'].str.split('|')

#Confirming the split
mov_genre.head(2)
```

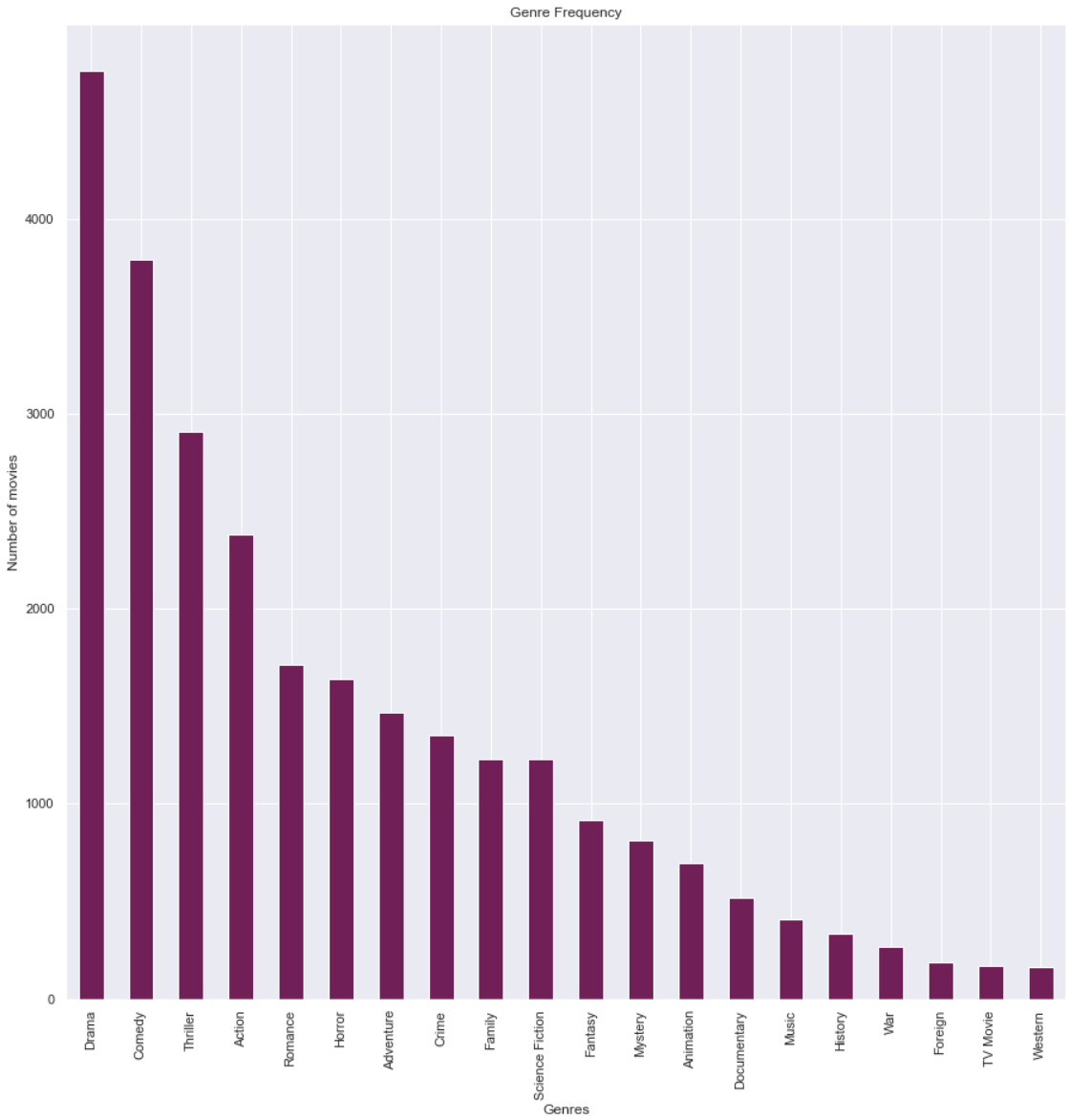
[27]:
```
      id  popularity      original_title  \
0  135397    0.329858      Jurassic World
1   76341    0.284199  Mad Max: Fury Road

                                               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

                                          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…   2015-06-09        5562
1  Village Roadshow Pictures|Kennedy Miller Produ…   2015-05-13        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
```

[28]:
```
#Exploding the dataset by the genres column
mov_genre = mov_genre.explode('genres').reset_index(drop=True)

#Confirming the data
mov_genre.head(2)
```

[28]:
```
      id  popularity  original_title  \
0  135397    0.329858  Jurassic World
1  135397    0.329858  Jurassic World

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

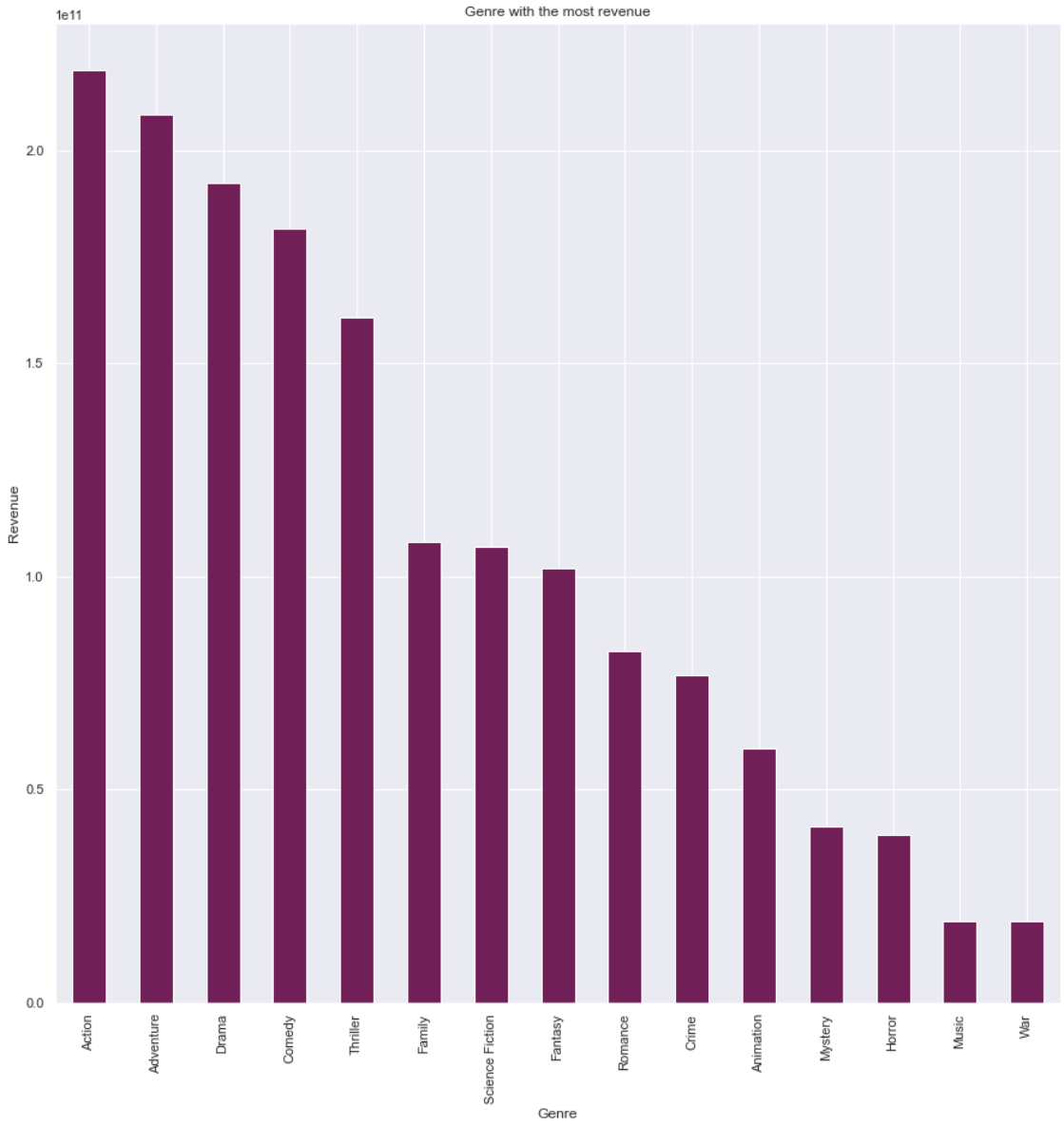
```
                                              keywords  runtime     genres  \
0  monster|dna|tyrannosaurus rex|velociraptor|island       124     Action
1  monster|dna|tyrannosaurus rex|velociraptor|island       124  Adventure

                               production_companies release_date  vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda…   2015-06-09        5562
1  Universal Studios|Amblin Entertainment|Legenda…   2015-06-09        5562

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

[29]:
```
#Plotting the genres frequency graph
genre_grp = mov_genre['genres'].value_counts()
genre_grp.plot(kind='bar', color='#701F57')

#Labelling the Data
plt.ylabel('Number of movies')
plt.title('Genre Frequency')
plt.xlabel('Genres');
```



Most movies belong to the Drama genre followed by Comedy, Thriller and Action.

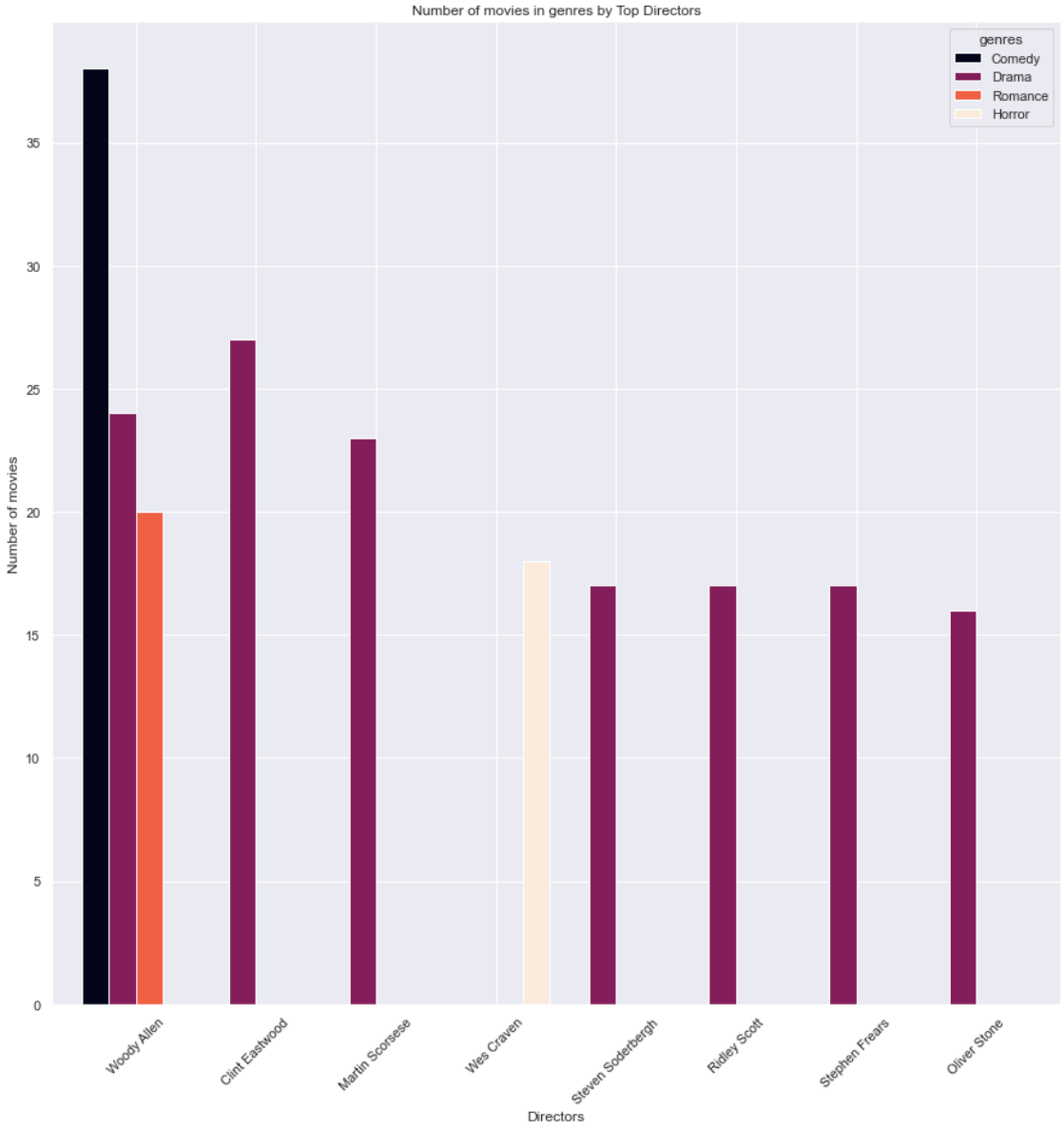
[30]:
```
#Finding the genres that have the most revenue
gnre_revnu = mov_genre.groupby(['genres'])['revenue_adj'].sum().
  ↪sort_values(ascending=False)[:15]

#Plotting the genre with the most revenue
gnre_revnu.plot(kind='bar', color='#701F57');
#Labelling the data
plt.ylabel('Revenue')
plt.title('Genre with the most revenue')
plt.xlabel('Genre');
```



Action genre has the most revenue followed by Adventure, Drama and Comedy

[31]:
```
#Grouping the data by directors and genres
dir_gnr = mov_genre.groupby(['director','genres'])['original_title'].count().
  ↪sort_values(ascending=False)[:10].unstack()
dir_gnr.plot(kind='bar', cmap = 'rocket', width = 0.9,)

#Labeling the data
plt.xticks(rotation=45);
plt.ylabel('Number of movies')
plt.title('Number of movies in genres by Top Directors')
```

```
plt.xlabel('Directors');
```



[32]:
```
dir_gnr
```

[32]:

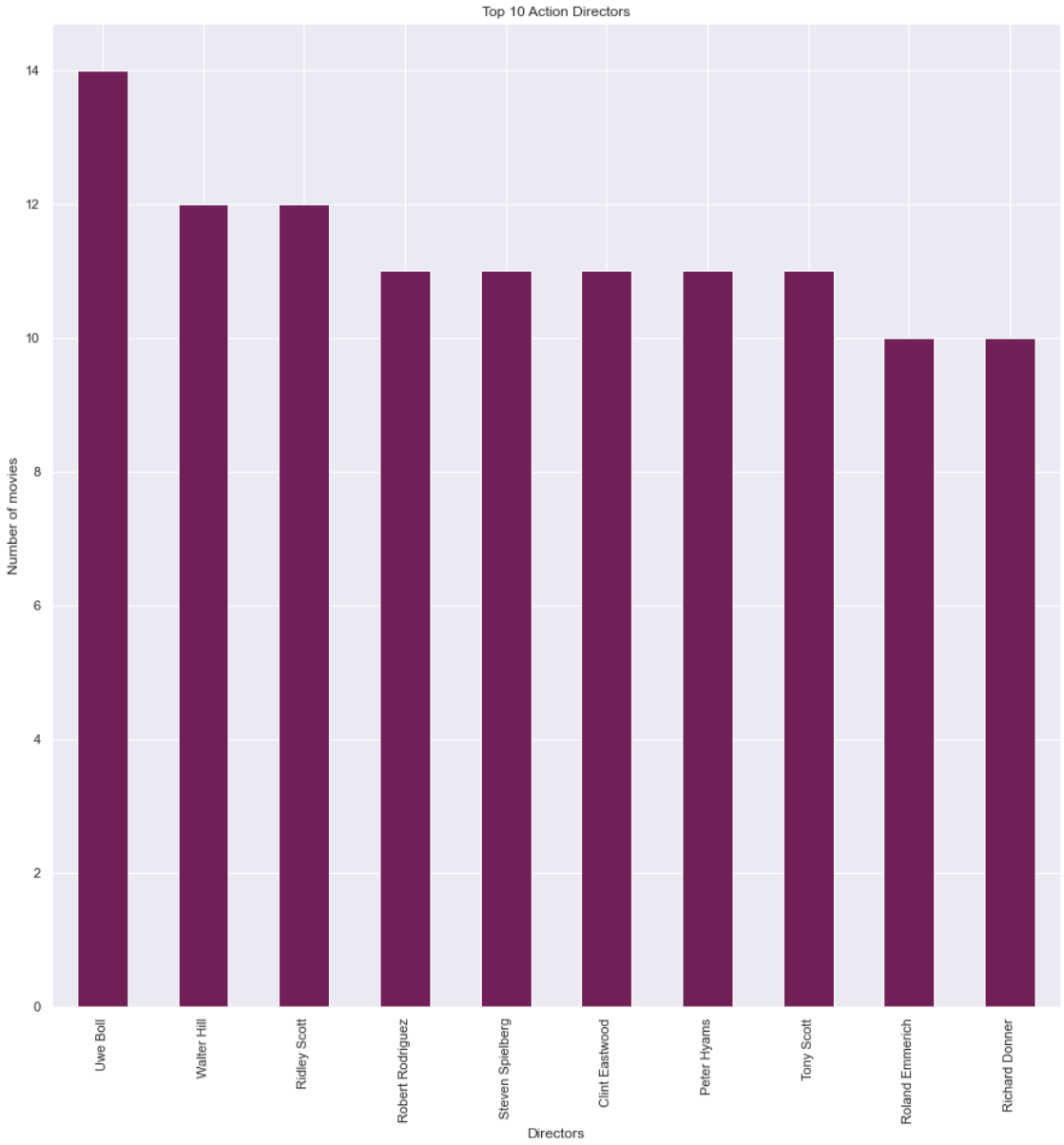
genres	Comedy	Drama	Romance	Horror
director				
Woody Allen	38.0	24.0	20.0	NaN
Clint Eastwood	NaN	27.0	NaN	NaN
Martin Scorsese	NaN	23.0	NaN	NaN
Wes Craven	NaN	NaN	NaN	18.0
Steven Soderbergh	NaN	17.0	NaN	NaN

```
Ridley Scott          NaN   17.0      NaN      NaN
Stephen Frears        NaN   17.0      NaN      NaN
Oliver Stone          NaN   16.0      NaN      NaN
```

From the table and graph above, the director with the most movies is Woody Allen with most movies in the comedy, drama and romance genres. Other directors dominated the drama genre with Wes Craven dominating the horror genre. Further exploration by genres and director

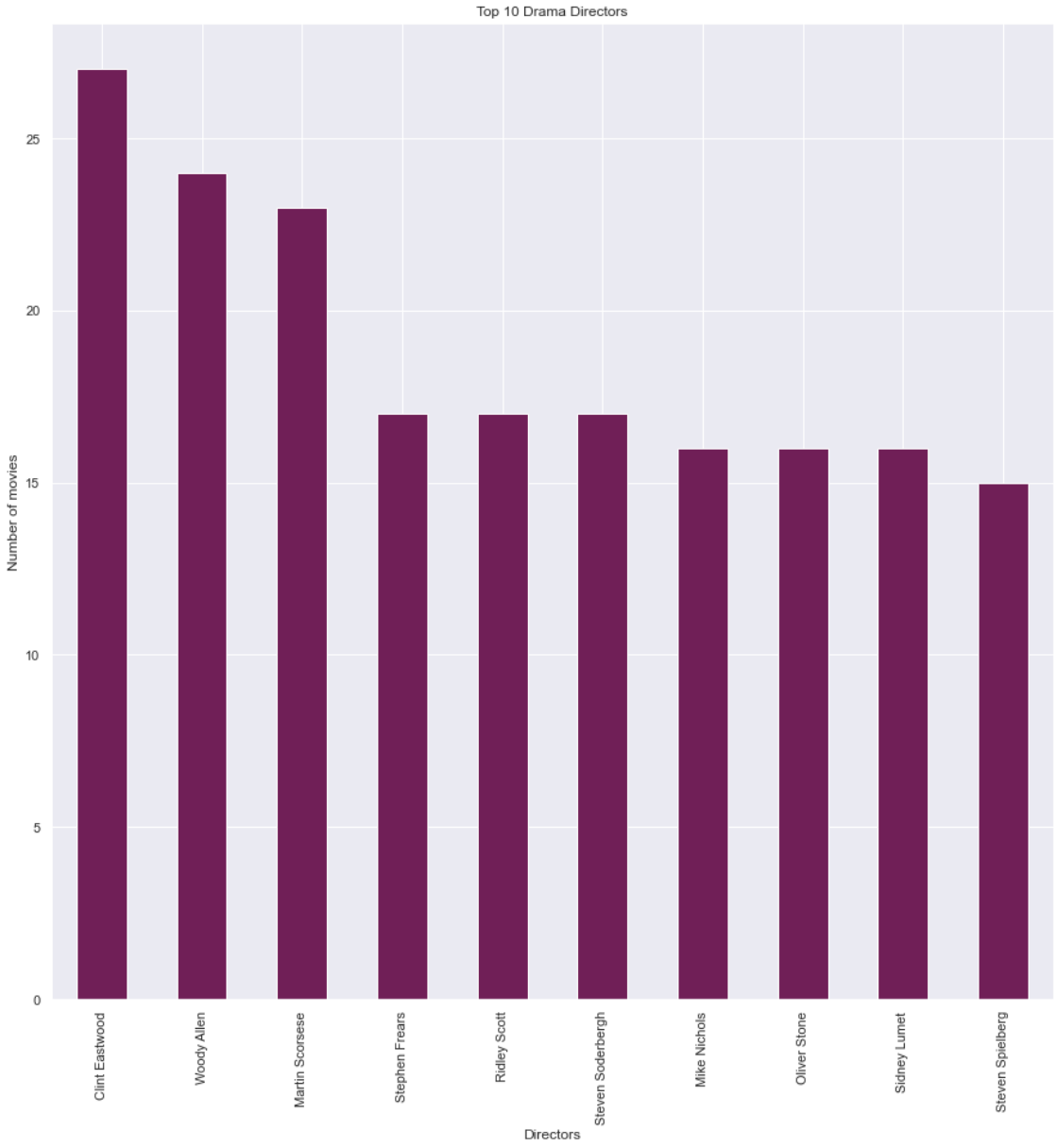
12.1 Top 10 Action Director

[33]:
```
Top_10_act_dir = mov_genre[mov_genre['genres']=='Action'].
↪groupby('director')['original_title'].count().sort_values(ascending=False)[:
↪10]
Top_10_act_dir.plot(kind='bar', color='#701F57')
plt.ylabel('Number of movies')
plt.title('Top 10 Action Directors')
plt.xlabel('Directors');
```



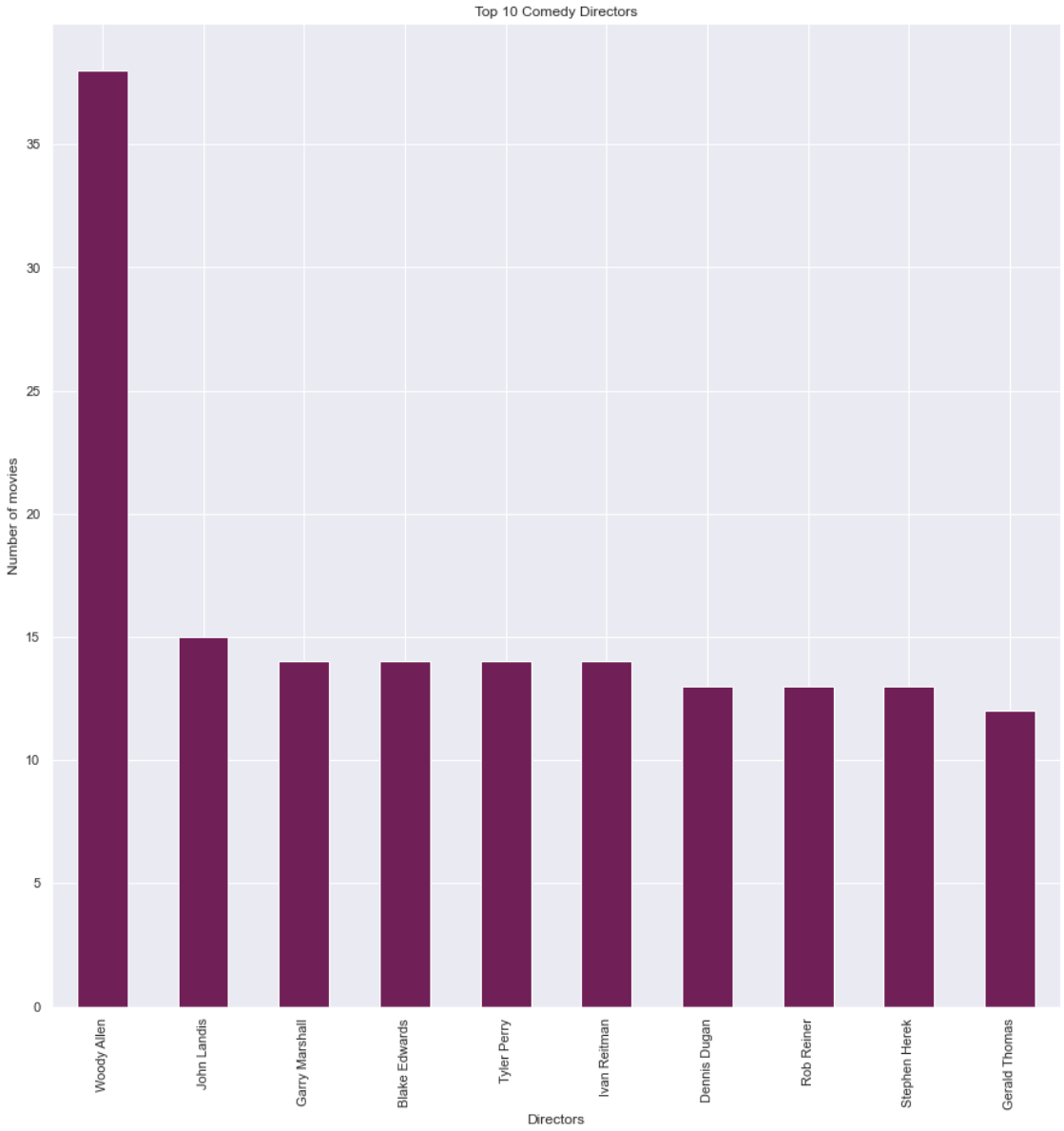
12.2 Top 10 Drama Directors

[34]:
```
Top_10_dra_dir = mov_genre[mov_genre['genres']=='Drama'].
↪groupby('director')['original_title'].count().sort_values(ascending=False)[:
↪10]
Top_10_dra_dir.plot(kind='bar', color='#701F57')
plt.ylabel('Number of movies')
plt.title('Top 10 Drama Directors')
plt.xlabel('Directors');
```



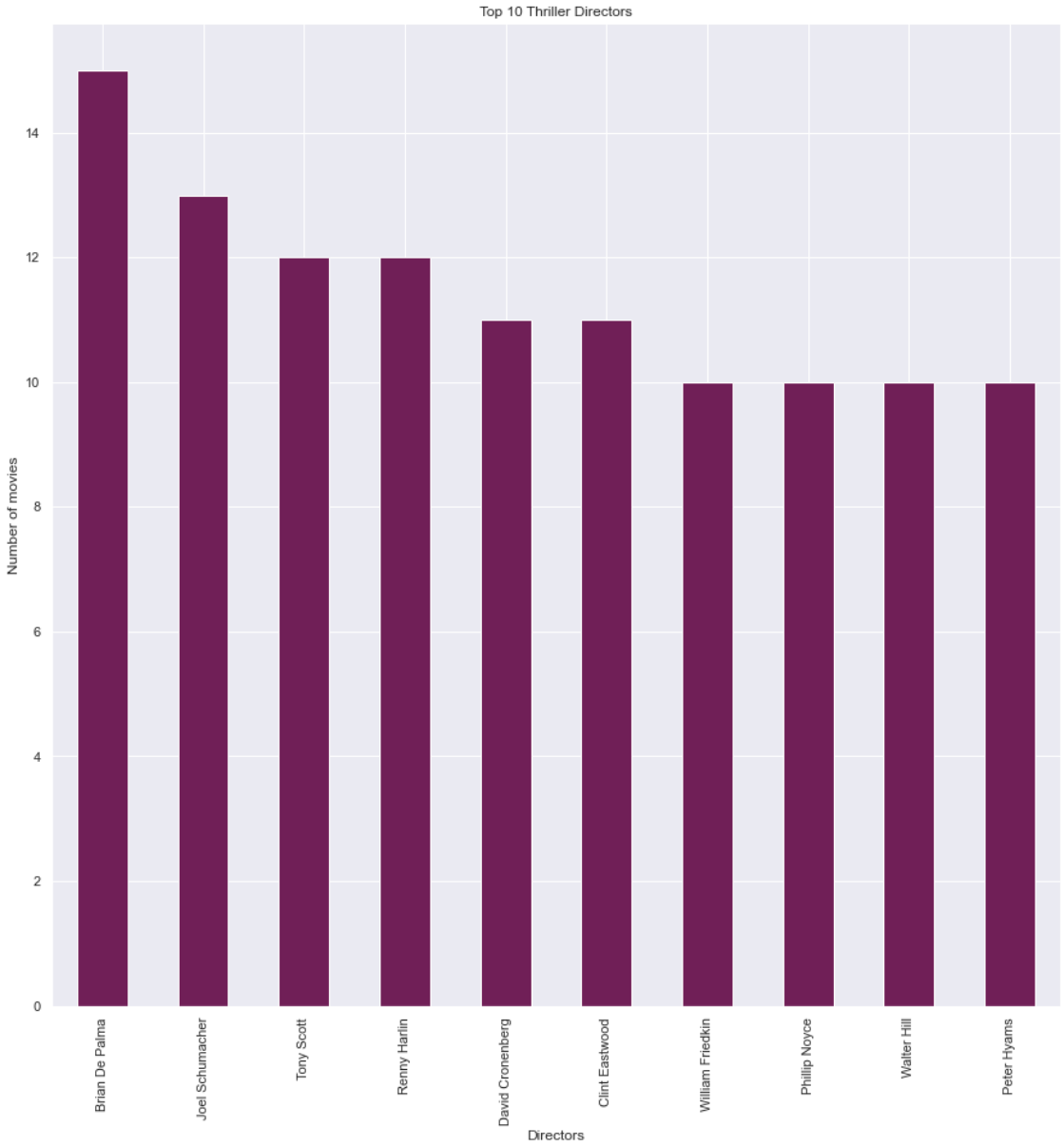
12.3 Top 10 Comedy Directors

[35]:
```
Top_10_cmd_dir = mov_genre[mov_genre['genres']=='Comedy'].
↪groupby('director')['original_title'].count().sort_values(ascending=False)[:
↪10]
Top_10_cmd_dir.plot(kind='bar', color='#701F57')
plt.ylabel('Number of movies')
plt.title('Top 10 Comedy Directors')
plt.xlabel('Directors');
```



12.4 Top 10 Thriller Directors

[36]:
```
Top_10_Trl_dir = mov_genre[mov_genre['genres']=='Thriller'].
↪groupby('director')['original_title'].count().sort_values(ascending=False)[:
↪10]
Top_10_Trl_dir.plot(kind='bar', color='#701F57')
plt.ylabel('Number of movies')
plt.title('Top 10 Thriller Directors')
plt.xlabel('Directors');
```



13 3: Relationships with Actors

This question aims to see the most casted actor, the actors and their respective genres.

[37]:
```
#Copying the data to create a new dataset that would be exploded
mov_genre_cast = mov_genre.copy()

#Splitting by the string '|'
mov_genre_cast['cast'] = mov_genre_cast['cast'].str.split('|')
```

```
mov_genre_cast.head(2)
```

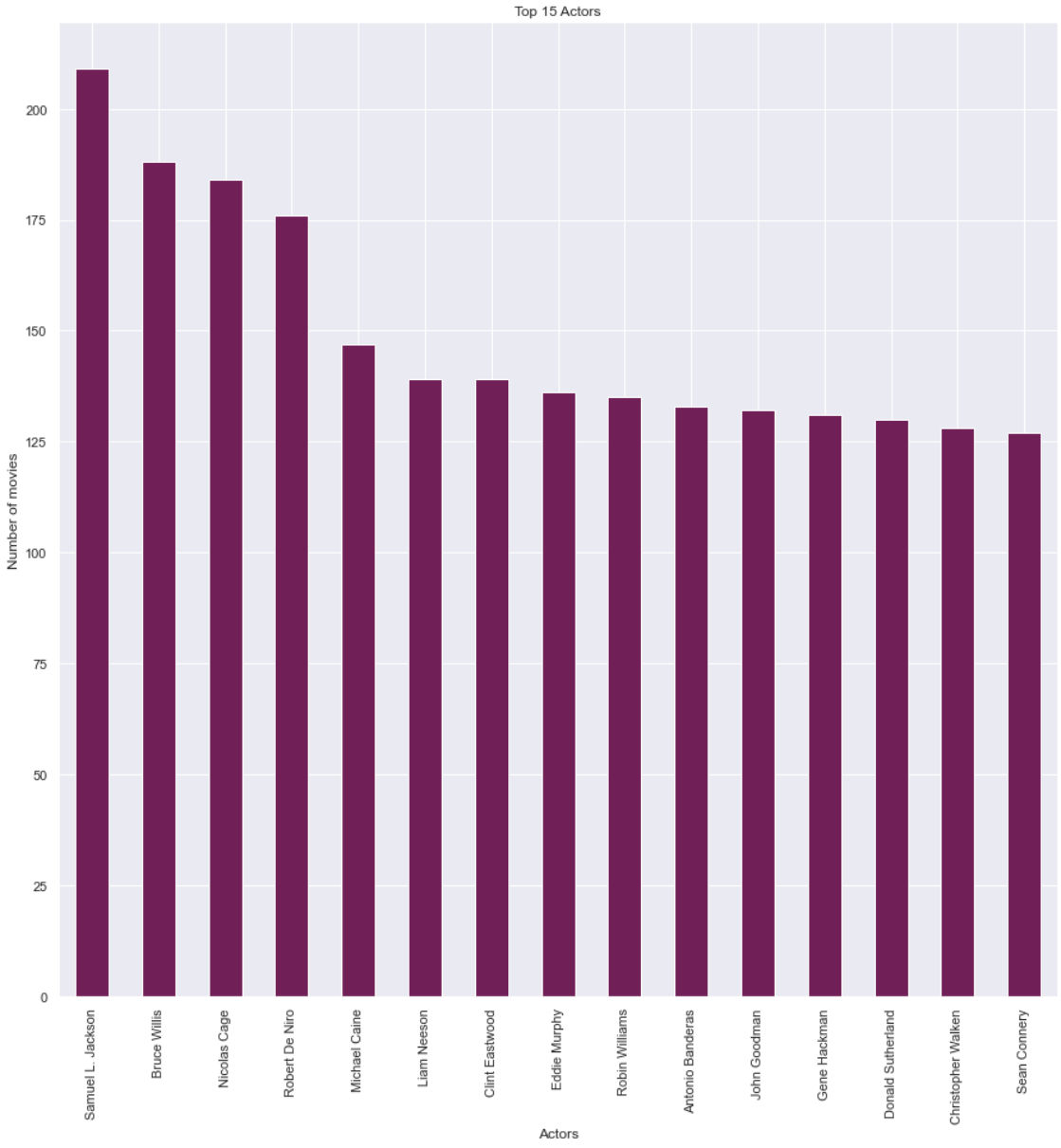
[37]:
```
       id  popularity  original_title  \
0  135397    0.329858  Jurassic World
1  135397    0.329858  Jurassic World

                                                cast         director  \
0  [Chris Pratt, Bryce Dallas Howard, Irrfan Khan…  Colin Trevorrow
1  [Chris Pratt, Bryce Dallas Howard, Irrfan Khan…  Colin Trevorrow

                                        keywords  runtime     genres  \
0  monster|dna|tyrannosaurus rex|velociraptor|island      124     Action
1  monster|dna|tyrannosaurus rex|velociraptor|island      124  Adventure

                                production_companies release_date  vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda…   2015-06-09        5562
1  Universal Studios|Amblin Entertainment|Legenda…   2015-06-09        5562

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

[38]:
```
#Exploding the dataset by the cast column
mov_genre_cast = mov_genre_cast.explode('cast').reset_index(drop=True)

#Confirming the data
mov_genre_cast.head(2)
```

[38]:
```
       id  popularity  original_title                 cast         director  \
0  135397    0.329858  Jurassic World          Chris Pratt  Colin Trevorrow
1  135397    0.329858  Jurassic World  Bryce Dallas Howard  Colin Trevorrow

                                        keywords  runtime  genres  \
0  monster|dna|tyrannosaurus rex|velociraptor|island      124  Action
1  monster|dna|tyrannosaurus rex|velociraptor|island      124  Action

                                production_companies release_date  vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda…   2015-06-09        5562
1  Universal Studios|Amblin Entertainment|Legenda…   2015-06-09        5562

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

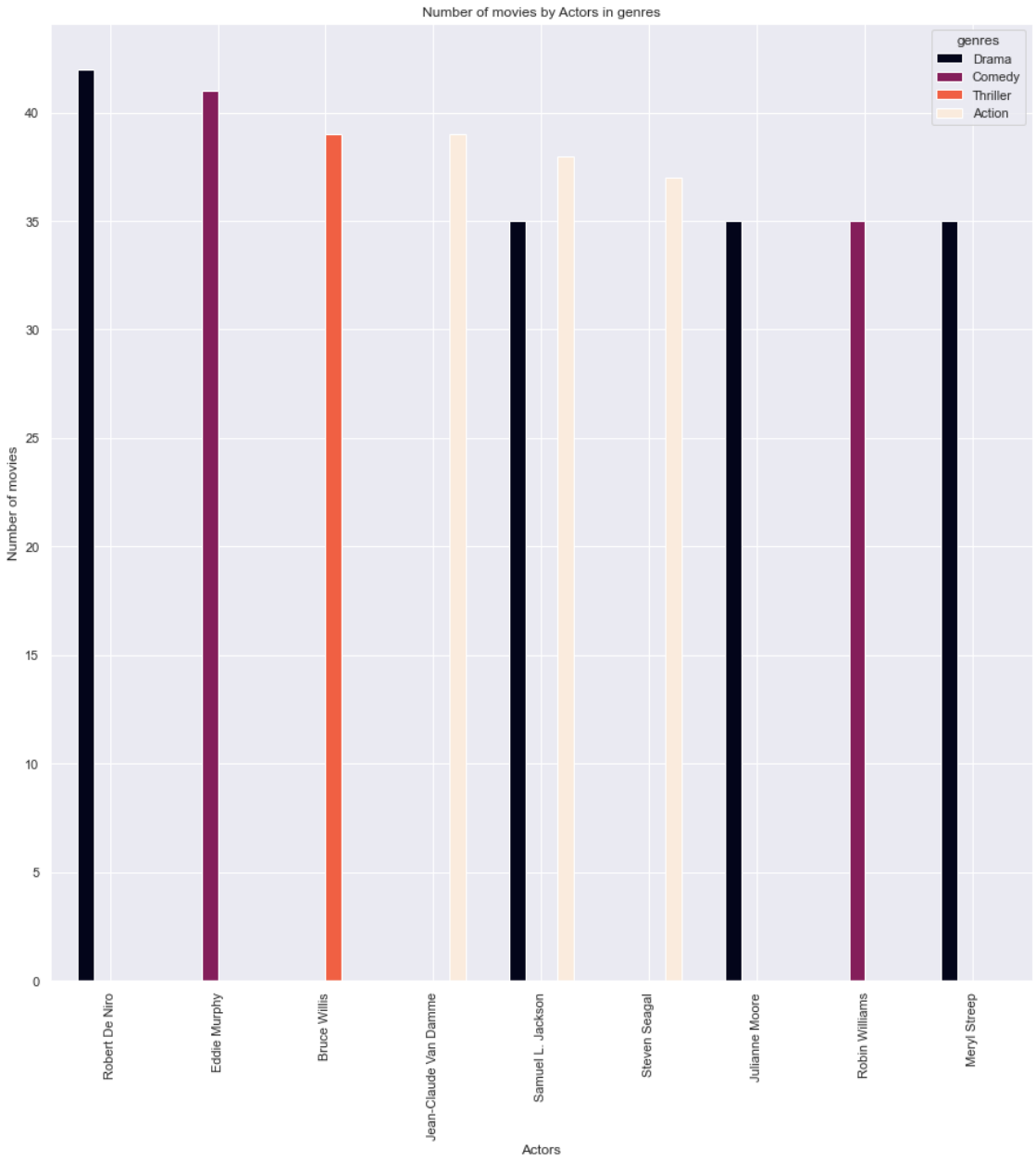
[39]:
```
#Getting the top 15 Actors
Top_15_act = mov_genre_cast['cast'].value_counts()[:15]
```

```
#Plotting the top 15 Actors
Top_15_act.plot(kind='bar', color='#701F57')

#Labelling the data
plt.ylabel('Number of movies')
plt.title('Top 15 Actors')
plt.xlabel('Actors');
```



The actor with the most movies is Samuel L Jackson followed by Bruce Willis, Nicholas Cage, Robert De Niro, and Michael Caine

[40]:
```
#Grouping the data by actors and genres
act_gnr = mov_genre_cast.groupby(['cast','genres'])['original_title'].count().
 ↪sort_values(ascending=False)[:10].unstack()
act_gnr.plot(kind='bar', cmap = 'rocket', width = 0.6,)

#Labeling the data
plt.xticks(rotation=90);
plt.ylabel('Number of movies')
plt.title('Number of movies by Actors in genres')
plt.xlabel('Actors');
```



The grouping of the actors and genres shows the most popular genres to be Drama, Comedy, Thriller and Action.

[41]:
```
act_gnr
```

[41]:

genres	Drama	Comedy	Thriller	Action
cast				
Robert De Niro	42.0	NaN	NaN	NaN
Eddie Murphy	NaN	41.0	NaN	NaN
Bruce Willis	NaN	NaN	39.0	NaN
Jean-Claude Van Damme	NaN	NaN	NaN	39.0
Samuel L. Jackson	35.0	NaN	NaN	38.0
Steven Seagal	NaN	NaN	NaN	37.0
Julianne Moore	35.0	NaN	NaN	NaN
Robin Williams	NaN	35.0	NaN	NaN
Meryl Streep	35.0	NaN	NaN	NaN

I have a theory that actors with the most films star in films across genres. I would test it by looking at the genre stats for the top five actors

[42]:
```
#Samuel L Jackson's stats
mov_genre_cast[mov_genre_cast['cast'].isin(['Samuel L. Jackson'])].
 ↪groupby('genres')['original_title'].count().sort_values(ascending=False)

#Bruce Willis's stats
mov_genre_cast[mov_genre_cast['cast'].isin(['Bruce Willis'])].
 ↪groupby('genres')['original_title'].count().sort_values(ascending=False)

#Nicholas Cage's stats
mov_genre_cast[mov_genre_cast['cast'].isin(['Nicolas Cage'])].
 ↪groupby('genres')['original_title'].count().sort_values(ascending=False)

#Robert De Niro's stats
mov_genre_cast[mov_genre_cast['cast'].isin(['Robert De Niro'])].
 ↪groupby('genres')['original_title'].count().sort_values(ascending=False)

#Michael Caine's stats
mov_genre_cast[mov_genre_cast['cast'].isin(['Michael Caine'])].
 ↪groupby('genres')['original_title'].count().sort_values(ascending=False)
```

[42]:

genres	
Drama	31
Thriller	22
Comedy	18
Action	17
Crime	14

```
Romance             8
Adventure           7
Mystery             5
Family              4
Fantasy             4
Science Fiction     4
War                 4
History             3
Horror              3
Animation           2
Documentary         1
Name: original_title, dtype: int64
```

After querying the top five actors, it is evident that the top actors act in a wide range of movie genres.

14 4: Relationships with production companies

This section would attempt to answer questions like what is the relationship between genres and companies, and which company is the most popular.

[43]:
```
#Copying the data to create a new dataset that would be exploded
mov_genre_prod = mov_genre.copy()

#Splitting by the string '|'
mov_genre_prod['production_companies'] = mov_genre_prod['production_companies'].
 ↪str.split('|')

mov_genre_prod.head(2)
```

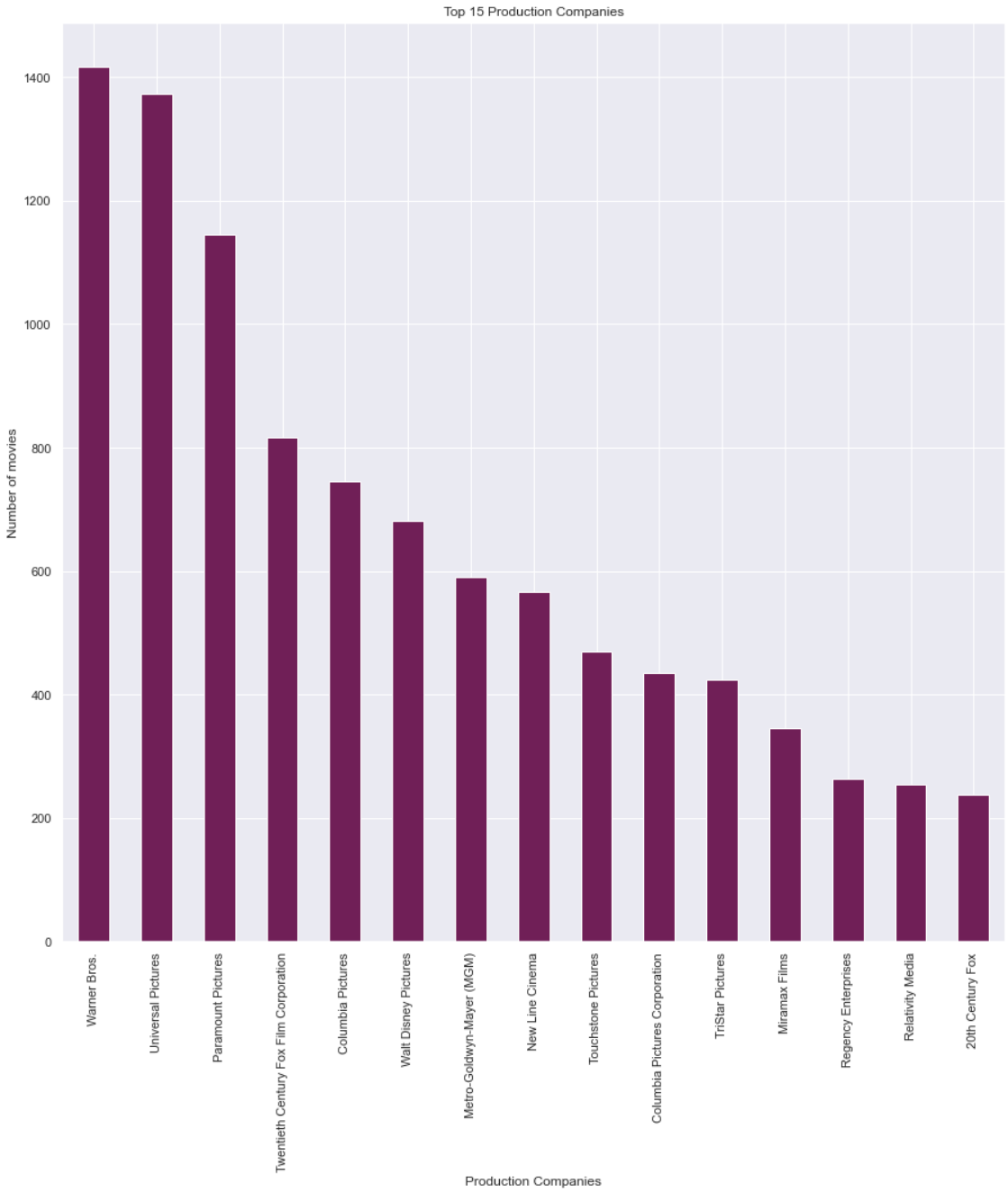
[43]:
```
       id  popularity  original_title  \
0  135397    0.329858  Jurassic World
1  135397    0.329858  Jurassic World

                                               cast         director  \
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…  Colin Trevorrow
1  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…  Colin Trevorrow

                                        keywords  runtime     genres  \
0  monster|dna|tyrannosaurus rex|velociraptor|island      124     Action
1  monster|dna|tyrannosaurus rex|velociraptor|island      124  Adventure

                              production_companies release_date  vote_count  \
0  [Universal Studios, Amblin Entertainment, Lege…   2015-06-09        5562
1  [Universal Studios, Amblin Entertainment, Lege…   2015-06-09        5562

   vote_average  release_year    budget_adj   revenue_adj
```

```
0           6.5          2015  1.379999e+08  1.392446e+09
1           6.5          2015  1.379999e+08  1.392446e+09
```

[44]:
```
#Exploding the dataset by the cast column
mov_genre_prod = mov_genre_prod.explode('production_companies').
 ↪reset_index(drop=True)

#Confirming the data
mov_genre_prod.head(2)
```

[44]:
```
       id  popularity  original_title  \
0  135397    0.329858  Jurassic World
1  135397    0.329858  Jurassic World

                                                cast         director  \
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…  Colin Trevorrow
1  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…  Colin Trevorrow

                                        keywords  runtime  genres  \
0  monster|dna|tyrannosaurus rex|velociraptor|island      124  Action
1  monster|dna|tyrannosaurus rex|velociraptor|island      124  Action

   production_companies release_date  vote_count  vote_average  release_year  \
0     Universal Studios   2015-06-09        5562           6.5          2015
1  Amblin Entertainment   2015-06-09        5562           6.5          2015

     budget_adj   revenue_adj
0  1.379999e+08  1.392446e+09
1  1.379999e+08  1.392446e+09
```

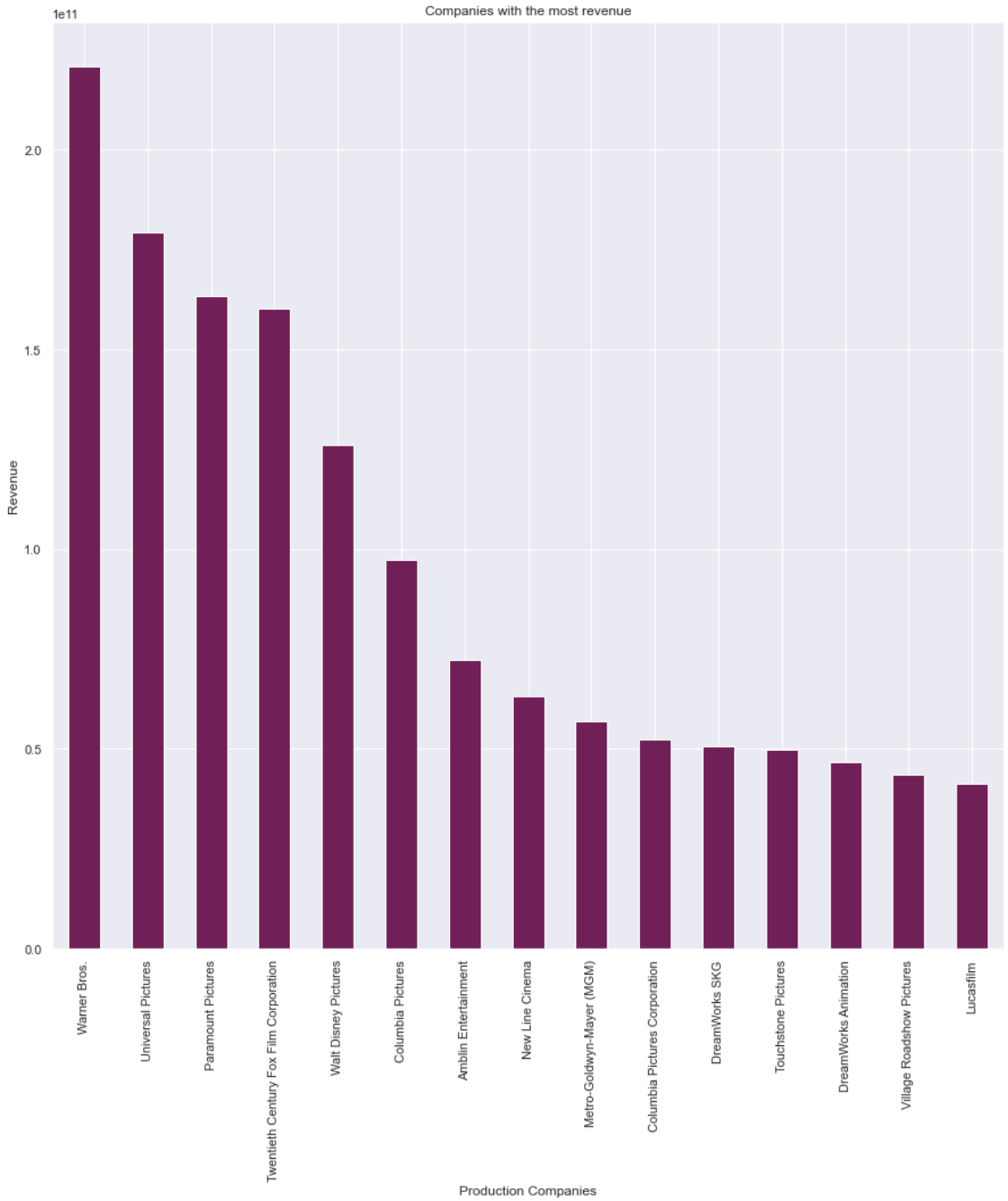
[45]:
```
#Getting the top 15 production companies
Top_15_prod = mov_genre_prod['production_companies'].value_counts()[:15]

#Plotting the top 15 Production Companies
Top_15_prod.plot(kind='bar', color='#701F57')


#Labelling the data
plt.ylabel('Number of movies')
plt.title('Top 15 Production Companies')
plt.xlabel('Production Companies');
```



The most popular production company is Warner Bros followed by Universal Pictures, Paramount Pictures, Twentieth Century Fox and Columbia Pictures

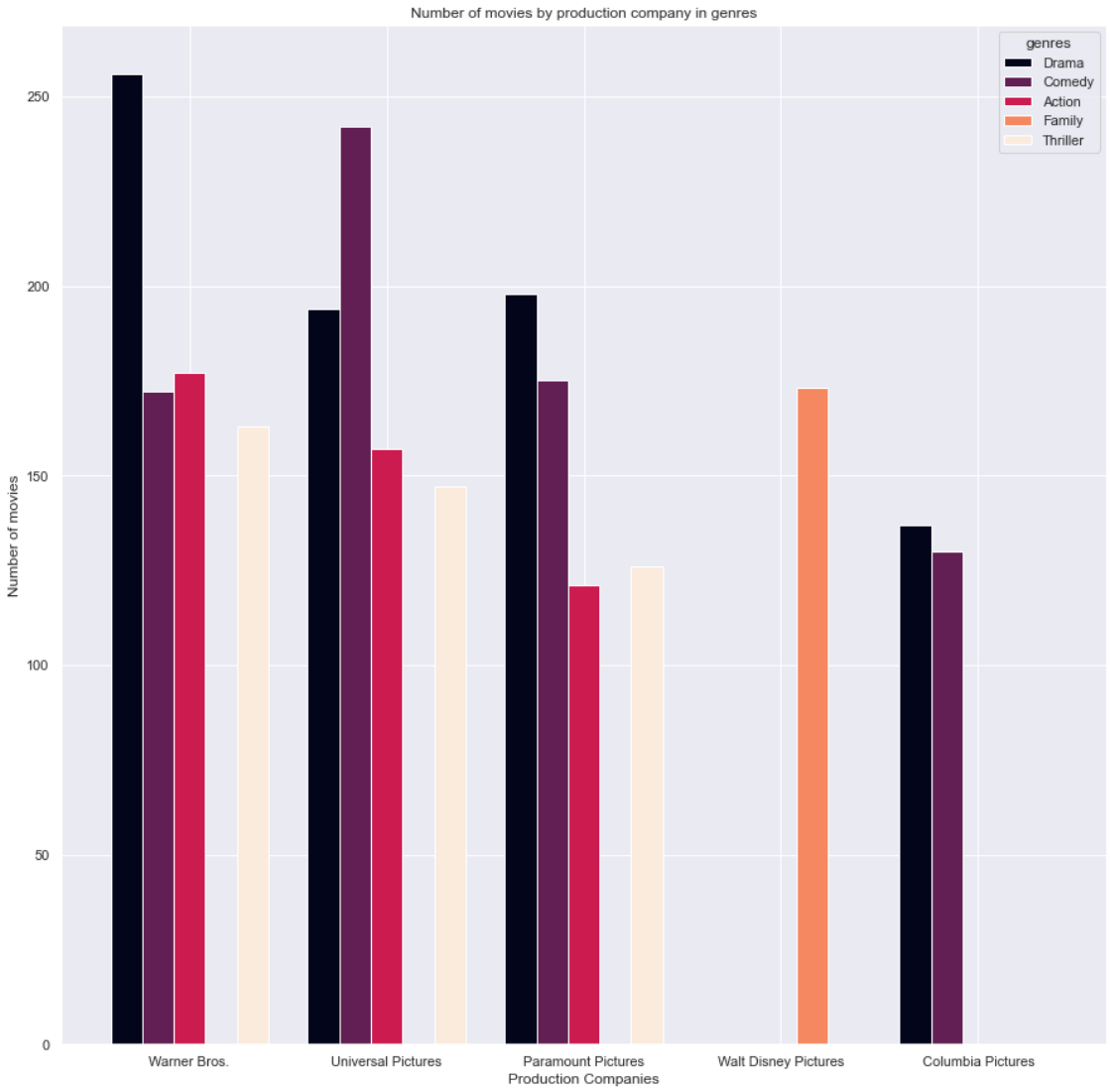
[46]:
```
#Finding the companies that have the most revenue
most_revnu = mov_genre_prod.groupby(['production_companies'])['revenue_adj'].
  ↪sum().sort_values(ascending=False)[:15]
```

```
#Plotting the most profitable company
most_revnu.plot(kind='bar', color='#701F57');
#Labelling the data
plt.ylabel('Revenue')
plt.title('Companies with the most revenue')
plt.xlabel('Production Companies');
```



Comparing the companies with the most movies with companies with the most revenues, there are exceptions in the list. Amblin Entertainment is a notable one. It brings in the 7th most revenue despite not being in the top 15 companies with the most movies. Village Roadshow Pictures and Lucasfilm also follow the trend of being in the top 15 profitable despite not being in the top 15 companies with the most movies

[47]:
```
#Grouping by production and genre
prod_grn = mov_genre_prod.
 ↪groupby(['production_companies','genres'])['original_title'].count().
 ↪sort_values(ascending=False)[:15].unstack()

#Plotting the data
prod_grn.plot(kind='bar', cmap = 'rocket', width = 0.8,)

#Labeling the data
plt.xticks(rotation=0)
plt.ylabel('Number of movies')
plt.title('Number of movies by production company in genres')
plt.xlabel('Production Companies');
```



Most of the top companies produce a diverse range of movies except for Walt Disney Pictures which produces mainly family and animated movies

---

15 Conclusion

The movie dataset has been analysed and questions were asked and answered. There was quite a lot of information that was gotten from the dataset. The major conclusions from the data are - Most movies have a run time of fewer than 200 minutes - Most movies received less than 1000 people's vote - The number of movies increases as the years do - Most movies have a budget less than 50,000,000 dollars - Most movies made revenue of fewer than 250,000,000 dollars - There is a strong positive correlation between Adjusted Budget and Adjusted Revenue, Release year and Id, Vote count and Adjusted budget, and Vote count and Adjusted revenue - The higher the budget,

the higher the revenue. If production companies want to make more profit, they'd have to spend more on making the movies. - Action genre has the most revenue followed by Adventure, Drama and Comedy. - Production companies with a large number of films also have high revenue except for a few exceptions - The top actors act in movies in a very wide range of genres. If anyone is aspiring to become a top actor, they have to spread their range of movies - Most movies are in the Drama genre and most production companies that want to profit would profit more in the drama genre. This is with the exception of Disney which is into the family and animation genre.

---

16 Limitations

Unfortunately, there were some limitations while I was analysing the data.

- Limitation 1: In the popularity column, there were quite a number of outliers as more than 90% of the dataset had values below 1 and the remainder ten percent had values ranging from 1 to 32. I tried to overcome this limitation by checking the IMDB website and other sources to see how the popularity was scored and how I can adjust the outliers accordingly. I couldn't find a suitable solution and the percentage of outliers are too much to drop. I decided to normalise the data by dividing by 10 if the number is in it's 10th and dividing by 100 if the number is in it's 100ths. Because the popularity columns were normalised by me, I couldn't carry out extensive EDA on it so as not to end up with wrong conclusions.
- Limitation 2: Due to the how the data is packed, I had to unpack the data by the respective column I was performing analysis. This was because if I had unpacked the columns simultaneously, some attributes would be duplicated and this would lead to wrong analysis whereby, the numbers analysed would be more than what it's supposed to be. Even with my method of unpacking one by one, there were still some forms of analysis that I was not able to carryout
- Limitation 3: Some of the data in the budget colunm were missing while having counterpart in the revenue colunm were available and vice versa. Because of this, I was not able to calculate the profit colunm as planned because that would give us profits that were not accurate. Due to this limitation, I wasn't able to compare profits.