

# Analyses on IMDb Movie Data

December 19, 2017

## Contents

<b>1</b>	<b>Group Information</b>	<b>1</b>
<b>2</b>	<b>Collecting Data</b>	<b>2</b>
2.1	Kaggle . . . . .	2
2.2	IMDb . . . . .	2
2.2.1	Movies on IMDb . . . . .	3
<b>3</b>	<b>Analyses</b>	<b>3</b>
3.1	Analysis on Directors . . . . .	4
3.1.1	Arranging Data . . . . .	5
3.1.2	Sort Data . . . . .	5
3.1.3	Genres of Directors' Work . . . . .	6
3.1.4	Summary . . . . .	6
3.2	Analysis on Directors by Rating . . . . .	6
3.2.1	Arranging Data . . . . .	7
3.2.2	Sort Data . . . . .	8
3.2.3	Genres of Directors' Work . . . . .	8
3.2.4	Summary . . . . .	9
3.3	Genre's Analysis . . . . .	9
3.3.1	Load Dataset . . . . .	9
3.3.2	Question . . . . .	11
3.3.3	Question . . . . .	26
3.3.4	Pie Chart of Average Revenue . . . . .	43
3.3.5	Summary . . . . .	46
3.4	Analysis on Rating and Metascore . . . . .	46
3.4.1	Implementation . . . . .	47
3.4.2	Summary . . . . .	49
3.5	Keywords Analyzing . . . . .	49
3.5.1	Implementation . . . . .	49
3.5.2	Summary . . . . .	50

## 1 Group Information

Group members' ID and name are sorted by alphabetical order of family name.

- 1430003004 Yongsheng (Jeff) Fu
- 1430003011 Kedun (Covey) Liu
- 1430003029 Jiaqi (Garfield) Wu
- 1430003030 Kefu (Frank) Wu
- 1430003045 Junru (bill) Zhong

## 2 Collecting Data

The data comes from Kaggle, which collects data of 1000 movies.

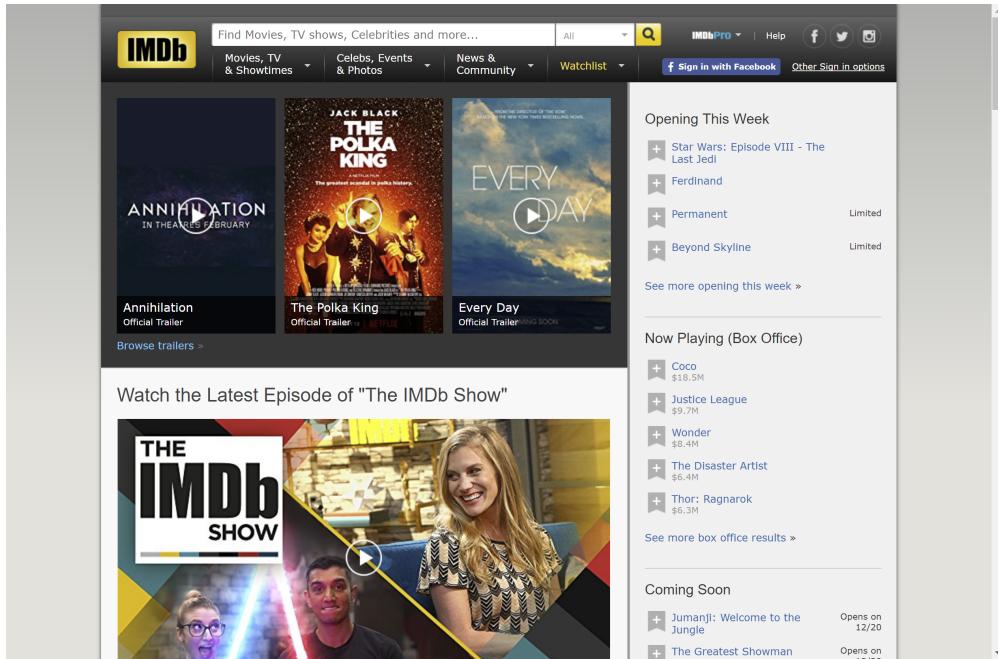
### 2.1 Kaggle

Kaggle is a platform for predictive modelling and analytics competitions in which statisticians and data miners compete to produce the best models for predicting and describing the datasets uploaded by companies and users.[Wikipedia]

The screenshot shows the Kaggle interface for the "IMDB data from 2006 to 2016" dataset. At the top, there's a navigation bar with links for Competitions, Datasets, Kernels, Discussion, Jobs, and more. Below the navigation is a large thumbnail image of movie posters from the dataset. A banner below the thumbnail reads "A data set of 1,000 popular movies on IMDB in the last 10 years". Below the banner, there are buttons for "Overview", "Data", "Kernels", "Discussion", and "Activity". The "Activity" button is highlighted in blue. There are also "Download (133 KB)" and "New Kernel" buttons. Further down, there are sections for "Top Contributors" (Damiano, NoPeriphs, Ajith Panner), "Kernels" (Sunday EDA, IMDB EDA, Simple IMDB Data analysis), and "Discussion" (Dataset suggestions, organi...). At the bottom, there's a "Description" section with a text input field and a "Help us describe this dataset" link.

### 2.2 IMDb

IMDb, formerly known as Internet Movie Database, is an online database of information related to films, television programs and video games, including cast, production crew, fictional characters, biographies, plot summaries, trivia and reviews, operated by IMDb.com, Inc., a subsidiary of Amazon.[Wikipedia]



## 2.2.1 Movies on IMDb

The screenshot shows the IMDb movie page for "Gekijo-ban Sword Art Online: Ordinal Scale" (2017). The page includes the movie's title, rating (7.5/10), director (Tomohiko Itō), and plot summary: "Kirito uncovers a conspiracy within Ordinal Scale, a popular VR game developed for a new system called, The Augma." Below the plot summary are sections for "Photos" (with 26 photos) and "Reviews" (12 user | 16 critic). To the right of the main movie info, there is a sidebar for "The IMDb Show": "Star Wars" Creatures We Love, featuring a video thumbnail of a Porg from Star Wars.

## 3 Analyses

Five analyses were brought out by members in the group.

- Directors and Revenue by Junru (Bill) Zhong
- Directors and Rating by Kefu (Frank) Wu

- Genres and Time by **Yongsheng (Jeff) Fu**
- Rating and Metascore by **Kedun (Covey) Liu**
- Keywords in Films by **Jiaqi (Garfield) Wu**

### 3.1 Analysis on Directors

This analysis was posted by **Junru (Bill) Zhong**, and the following aims were setted.

- Find out which director earned most.
- Find out the genres of the works of the five-most earned directors.
- Estimate how much each of them will get in their next film.

This analysis was brought by Python with package: pandas

```
In [1]: # Import package
    import pandas as pd
    # Read data
    df = pd.read_csv('../Dataset/IMDB-Movie-Data.csv')
    print(df.head())

      Rank          Title           Genre \
0     1  Guardians of the Galaxy  Action, Adventure, Sci-Fi
1     2              Prometheus  Adventure, Mystery, Sci-Fi
2     3                  Split   Horror, Thriller
3     4                  Sing  Animation, Comedy, Family
4     5        Suicide Squad  Action, Adventure, Fantasy

                                         Description           Director \
0  A group of intergalactic criminals are forced ...       James Gunn
1  Following clues to the origin of mankind, a te...       Ridley Scott
2  Three girls are kidnapped by a man with a diag...  M. Night Shyamalan
3  In a city of humanoid animals, a hustling thea... Christophe Lourdelet
4  A secret government agency recruits some of th...       David Ayer

          Actors  Year Runtime (Minutes) \
0  Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...    2014                121
1  Noomi Rapace, Logan Marshall-Green, Michael Fa...    2012                124
2  James McAvoy, Anya Taylor-Joy, Haley Lu Richar...    2016                117
3  Matthew McConaughey, Reese Witherspoon, Seth Ma...    2016                108
4  Will Smith, Jared Leto, Margot Robbie, Viola D...    2016                123

  Rating  Votes Revenue (Millions)  Metascore
0     8.1  757074         333.13      76.0
1     7.0  485820         126.46      65.0
2     7.3  157606         138.12      62.0
3     7.2  60545          270.32      59.0
4     6.2  393727         325.02      40.0
```

### 3.1.1 Arranging Data

- Use groupby method provided by pandas, select all directors and their films' revenue.
- Collect this data with the film count and total revenue of their works.
- Drop the data with value of null.

In [4]: # List all directors.

```
directorList = df.groupby('Director')['Revenue (Millions)'].agg(['sum', 'count']).reset_index()
# Delete invalid data
directorList = directorList.dropna()
# Print out result
print(directorList.head())
```

	Director	sum	count
0	Aamir Khan	1.20	1
1	Abdellatif Kechiche	2.20	1
3	Adam McKay	438.14	4
4	Adam Shankman	157.33	2
5	Adam Wingard	21.07	2

### 3.1.2 Sort Data

Sort the data according to the **average revenue** of each film that the directors directed.

- Calculate the average revenue.
- Sort the list.
- Save the first five directors.

In [6]: # Get average revenue of each director.

```
directorList['avg'] = directorList['sum'] / directorList['count']
print(directorList.head())
```

	Director	sum	count	avg
0	Aamir Khan	1.20	1	1.200
1	Abdellatif Kechiche	2.20	1	2.200
3	Adam McKay	438.14	4	109.535
4	Adam Shankman	157.33	2	78.665
5	Adam Wingard	21.07	2	10.535

In [7]: # Sorting by revenue, list the first five data.

```
firstFive = directorList.sort_values('avg', ascending=False).head()
print(firstFive)
```

	Director	sum	count	avg
261	James Cameron	760.51	1	760.510

114	Colin Trevorrow	652.18	1	652.180
341	Joss Whedon	1082.27	2	541.135
377	Lee Unkrich	414.98	1	414.980
208	Gary Ross	408.00	1	408.000

### 3.1.3 Genres of Directors' Work

Go back to the original dataset, then match all entries of these directors.

```
In [8]: # Searching their works.
directorNames = firstFive['Director']
for directorName in directorNames:
    for index, row in df.iterrows():
        if row['Director'] == directorName:
            print(row['Director'] + ', ' + row['Genre'] + ', ' + row['Title'])
```

James Cameron, Action, Adventure, Fantasy, Avatar  
 Colin Trevorrow, Action, Adventure, Sci-Fi, Jurassic World  
 Joss Whedon, Action, Sci-Fi, The Avengers  
 Joss Whedon, Action, Adventure, Sci-Fi, Avengers: Age of Ultron  
 Lee Unkrich, Animation, Adventure, Comedy, Toy Story 3  
 Gary Ross, Adventure, Sci-Fi, Thriller, The Hunger Games  
 Gary Ross, Action, Biography, Drama, Free State of Jones

*One movie from Gary Ross was not calculated because there is no revenue data.*

### 3.1.4 Summary

- In the first five directors, seven of their works were recorded.
- Six of seven films have the genre of **Action**, that means action films are popular.
- Four of seven films have the genre of **Sci-Fi**, this genre is also popular.

## 3.2 Analysis on Directors by Rating

This analysis was posted by **Kefu (Frank) Wu**, and the following aims were setted.

- Find out which director is praised most by rating, as revenue cannot define how the films are evaluated by the public.
- Find out the genres of the works of the five-most praised directors.
- Estimate the rating that new film may get according to the genres of the film and its director.

This analysis was brought by Python with package: pandas

```
In [3]: # Show origin data
print(df.head())
```

Rank		Title	Genre	\
0	1	Guardians of the Galaxy	Action, Adventure, Sci-Fi	
1	2	Prometheus	Adventure, Mystery, Sci-Fi	
2	3	Split	Horror, Thriller	
3	4	Sing	Animation, Comedy, Family	
4	5	Suicide Squad	Action, Adventure, Fantasy	
Description		Director	\	
0	A group of intergalactic criminals are forced ...	James Gunn		
1	Following clues to the origin of mankind, a te...	Ridley Scott		
2	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan		
3	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet		
4	A secret government agency recruits some of th...	David Ayer		
Actors	Year	Runtime (Minutes)	\	
0	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	
1	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	
2	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	
3	Matthew McConaughey, Reese Witherspoon, Seth Ma...	2016	108	
4	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	
Rating	Votes	Revenue (Millions)	Metascore	
0	8.1	757074	333.13	76.0
1	7.0	485820	126.46	65.0
2	7.3	157606	138.12	62.0
3	7.2	60545	270.32	59.0
4	6.2	393727	325.02	40.0

### 3.2.1 Arranging Data

- Use groupby method provided by pandas, select all directors and their films' rating.
- Collect this data with the film count and sum up the rating of their works.

In [4]: # List all directors.

```
directorList = df.groupby('Director')['Rating'].agg(['sum', 'count']).reset_index()
# Print out result
print(directorList.head())
```

	Director	sum	count
0	Aamir Khan	8.5	1
1	Abdellatif Kechiche	7.8	1
2	Adam Leon	6.5	1
3	Adam McKay	28.0	4
4	Adam Shankman	12.6	2

### 3.2.2 Sort Data

Sort the data according to the **average rating** of each film that the directors directed. \* Calculate the average rating. \* Sort the list. \* Save the top five directors that customers praised.

```
In [5]: # Get average rating of each director.  
directorList['avg'] = directorList['sum'] / directorList['count']  
print(directorList.head())
```

	Director	sum	count	avg
0	Aamir Khan	8.5	1	8.5
1	Abdellatif Kechiche	7.8	1	7.8
2	Adam Leon	6.5	1	6.5
3	Adam McKay	28.0	4	7.0
4	Adam Shankman	12.6	2	6.3

```
In [6]: # Sorting by rating, list the first five data.  
firstFive = directorList.sort_values('avg', ascending=False).head()  
print(firstFive)
```

	Director	sum	count	avg
465	Nitesh Tiwari	8.8	1	8.80
108	Christopher Nolan	43.4	5	8.68
392	Makoto Shinkai	8.6	1	8.60
470	Olivier Nakache	8.6	1	8.60
194	Florian Henckel von Donnersmarck	8.5	1	8.50

### 3.2.3 Genres of Directors' Work

Go back to the original dataset, then match all entries of these directors.

```
In [23]: # Searching their works.  
directorNames = firstFive['Director']  
for directorName in directorNames:  
    for index, row in df.iterrows():  
        if row['Director'] == directorName:  
            print(row['Director'] + ': \n\t' + row['Genre'] + ', ' + row['Title'])
```

Nitesh Tiwari:

Action, Biography, Drama, Dangal

Christopher Nolan:

Adventure, Drama, Sci-Fi, Interstellar

Christopher Nolan:

Action, Crime, Drama, The Dark Knight

Christopher Nolan:

Drama, Mystery, Sci-Fi, The Prestige

Christopher Nolan:

```

Action,Adventure,Sci-Fi, Inception
Christopher Nolan:
    Action,Thriller, The Dark Knight Rises
Makoto Shinkai:
    Animation,Drama,Fantasy, Kimi no na wa
Olivier Nakache:
    Biography,Comedy,Drama, The Intouchables
Florian Henckel von Donnersmarck:
    Drama,Thriller, The Lives of Others

```

### 3.2.4 Summary

- We find out five director that are favored by their audience.
- In the first five directors, nine of their works were recorded.
- Among the directors, Chrostopher Nolan is the only director that can maintain high average rating after producing more than one film.
- In Nolan's five works, there are three of them are action/sci-Fi, which means that Nolan is good at directoring action/sci-Fi film.
- To sum up, we can predict that Chrostopher Nolan's next action/sci-Fi movie will gain a high comment, but for the rest of the top-five directors we cannot predict because there is only one record for each of them.

## 3.3 Genre's Analysis

The following python code analyze the revenue of each kind of moive in past ten years. We can easily learn what kind of moive is easiest to make money.

In [1]: #Import packages

```

import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
import seaborn as sns
sns.set(style='white', color_codes=True, font_scale=1.25)
import itertools

```

### 3.3.1 Load Dataset

First, we read the IMDB dataset csv file, and load it in to a DataFrame. Here are the first 5 row of the dataset.

In [2]: #Import the data into a DF and view the first 5 entries

```

imdb = pd.read_csv('..../Dataset/IMDB-Movie-Data.csv')
imdb.head()

```

```

Out[2]:   Rank          Title           Genre \
0      1  Guardians of the Galaxy  Action, Adventure, Sci-Fi
1      2                  Prometheus  Adventure, Mystery, Sci-Fi
2      3                   Split    Horror, Thriller
3      4                   Sing    Animation, Comedy, Family
4      5        Suicide Squad  Action, Adventure, Fantasy

                                         Description           Director \
0  A group of intergalactic criminals are forced ...       James Gunn
1  Following clues to the origin of mankind, a te...       Ridley Scott
2  Three girls are kidnapped by a man with a diag...     M. Night Shyamalan
3  In a city of humanoid animals, a hustling thea...  Christophe Lourdelet
4  A secret government agency recruits some of th...       David Ayer

                                         Actors   Year Runtime (Minutes) \
0  Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...  2014            121
1  Noomi Rapace, Logan Marshall-Green, Michael Fa...  2012            124
2  James McAvoy, Anya Taylor-Joy, Haley Lu Richar...  2016            117
3  Matthew McConaughey, Reese Witherspoon, Seth Ma...  2016            108
4  Will Smith, Jared Leto, Margot Robbie, Viola D...  2016            123

          Rating   Votes  Revenue (Millions)  Metascore
0      8.1  757074          333.13       76.0
1      7.0  485820          126.46       65.0
2      7.3  157606          138.12       62.0
3      7.2  60545           270.32       59.0
4      6.2  393727          325.02       40.0

```

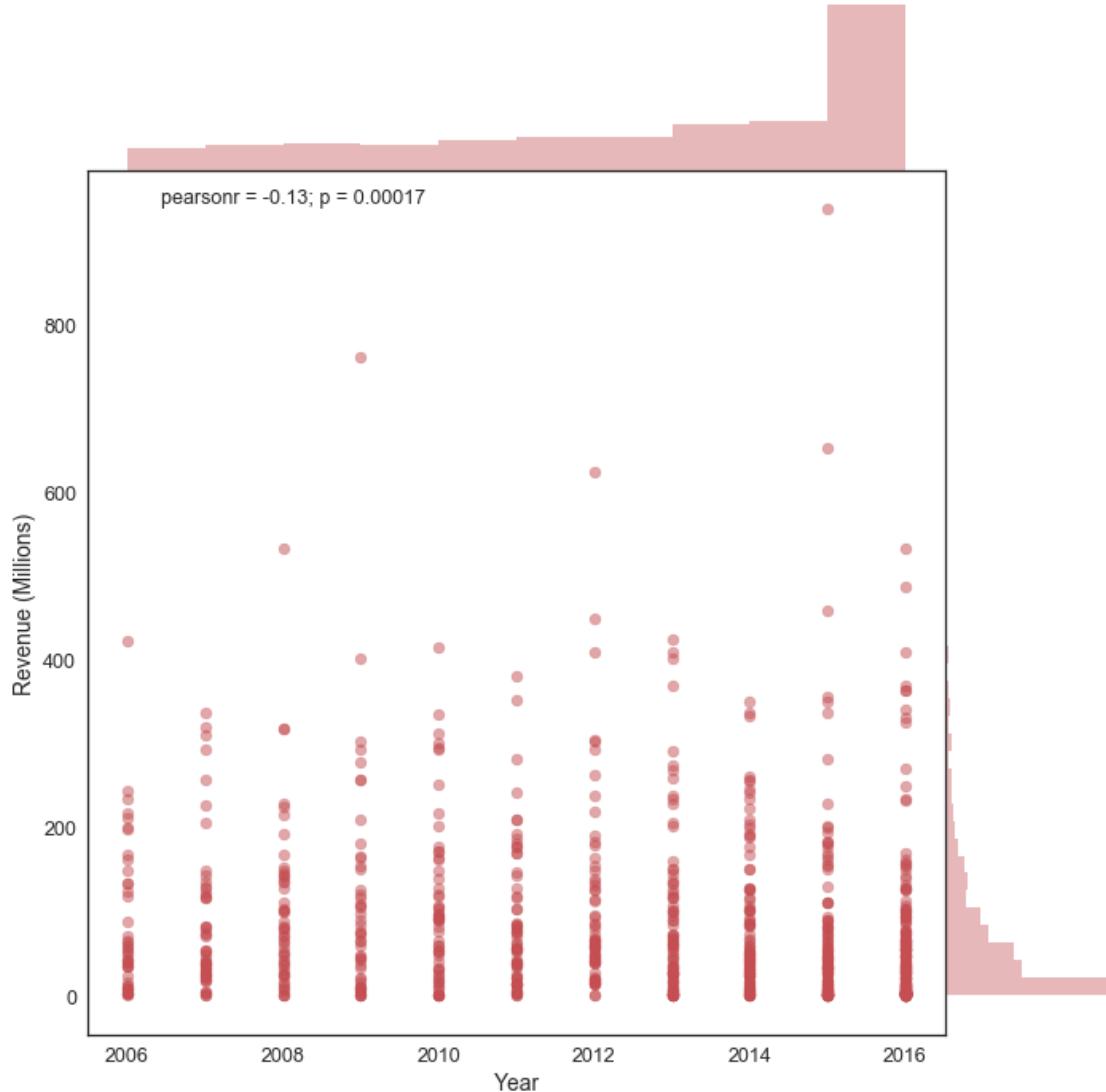
Now, we use above dataset to plot a scatter graph so that we can know the **distribution of movies and their revenue**

```

In [3]: #Plot the Revenue in recent ten years
sns.jointplot(x='Year', y='Revenue (Millions)', data=imdb,
alpha=0.5, color='r', size=10, space=0)

```

```
Out[3]: <seaborn.axisgrid.JointGrid at 0x7fa0c64a0f98>
```



As the result shows, we know that more and more file are made in recent year. Besides, the result also shows us that most movie's revenue is less than 200 Millions dollar.

### 3.3.2 Question

- **What kind of movie are easiest to earn money?**
- **For answering this question, let's divide movie into different genre.**

By using unique() build-in function of pandas package. And record them by iteration, finally we got 20 unique genres. Here are the result.

In [4]: #Get unique genres

```
unique_genres = imdb['Genre'].unique()
```

```

individual_genres = []
for genre in unique_genres:
    individual_genres.append(genre.split(','))
individual_genres = list(itertools.chain.from_iterable(individual_genres))
individual_genres = set(individual_genres)
individual_genres

```

Out [4]:

```

{'Action',
 'Adventure',
 'Animation',
 'Biography',
 'Comedy',
 'Crime',
 'Drama',
 'Family',
 'Fantasy',
 'History',
 'Horror',
 'Music',
 'Musical',
 'Mystery',
 'Romance',
 'Sci-Fi',
 'Sport',
 'Thriller',
 'War',
 'Western'}

```

After we get each individual genres. We plot how many films of each genre were made by year onto a bar graph

In [5]: #Now we can iterate through each genre, counting the number of films that contain that genre  
#then plot how many films of each genre were made by year onto a bar graph

```

print('Number of movies in each genre: \n')

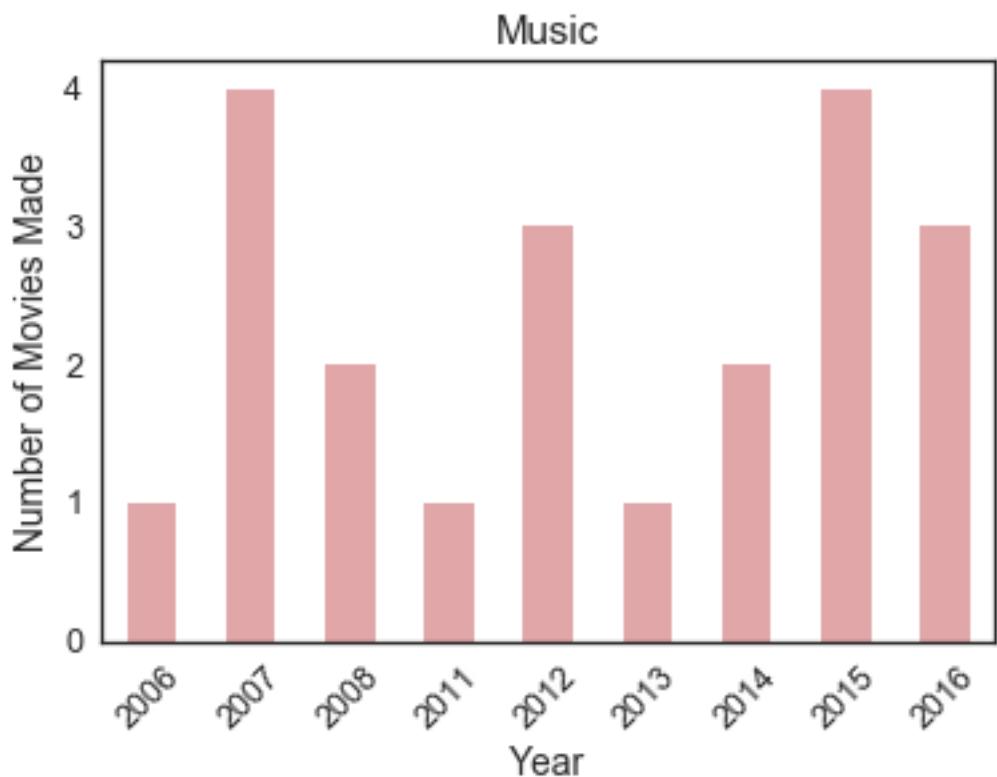
for genre in individual_genres:
    current_genre = imdb['Genre'].str.contains(genre).fillna(False)
    plt.figure()
    plt.xlabel('Year')
    plt.ylabel('Number of Movies Made')
    plt.title(str(genre))
    imdb[current_genre].Year.value_counts().sort_index().
        plot(kind='bar', color='r', alpha=0.5, rot=45)
    print(genre, len(imdb[current_genre]))

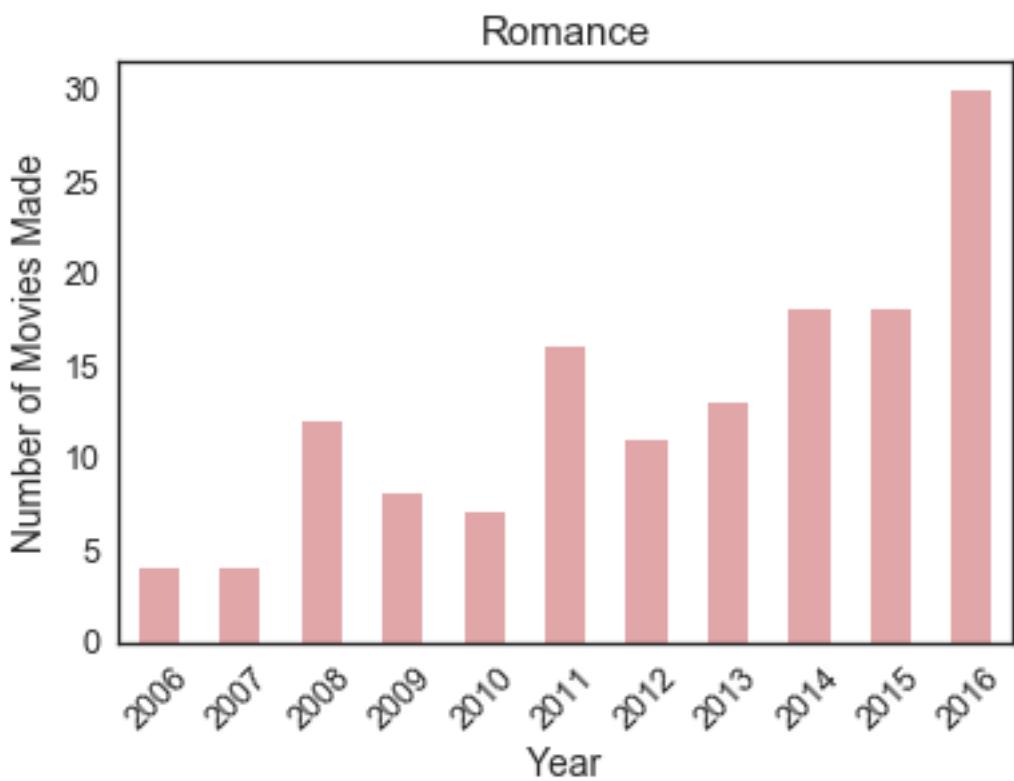
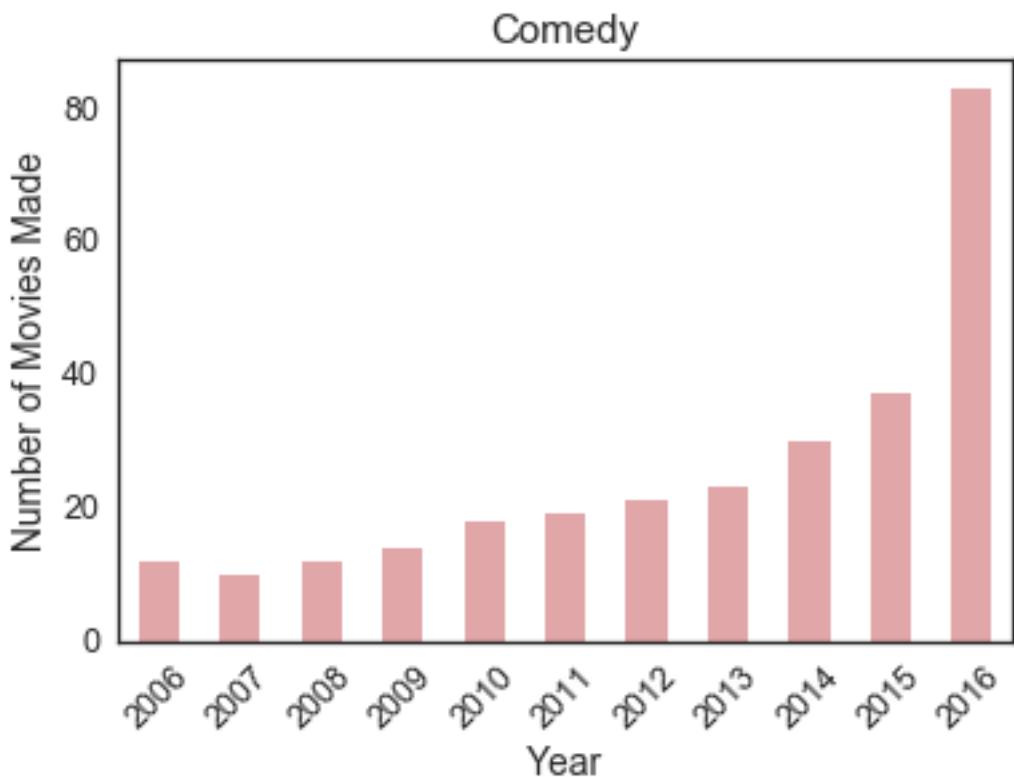
```

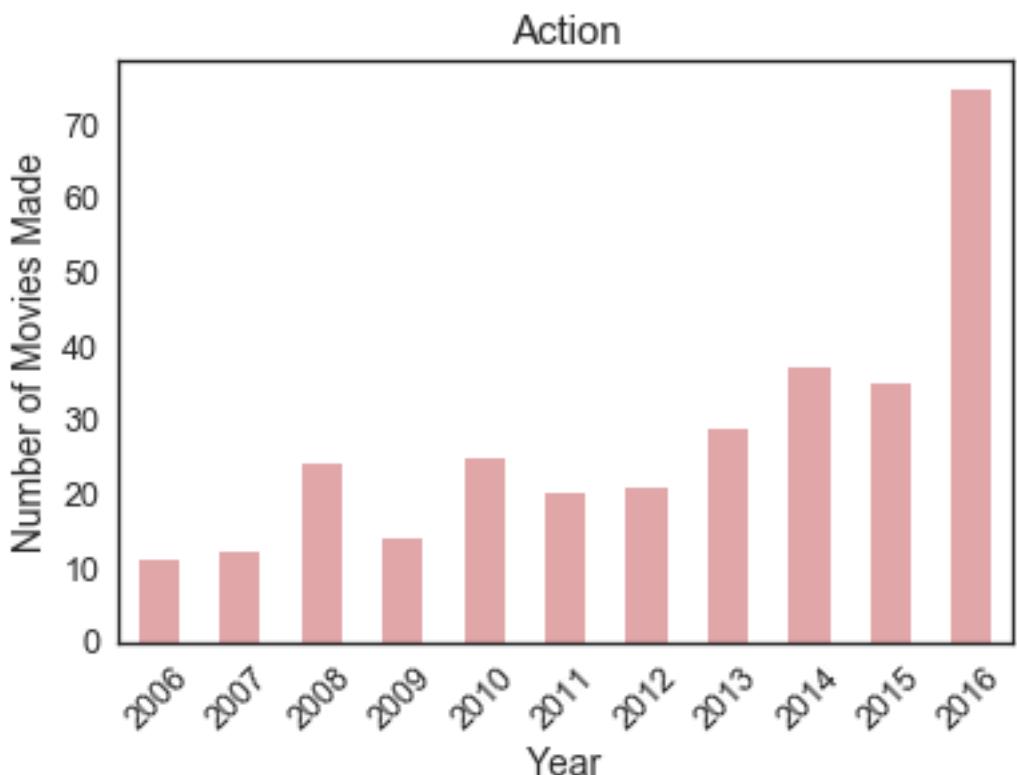
Number of movies in each genre:

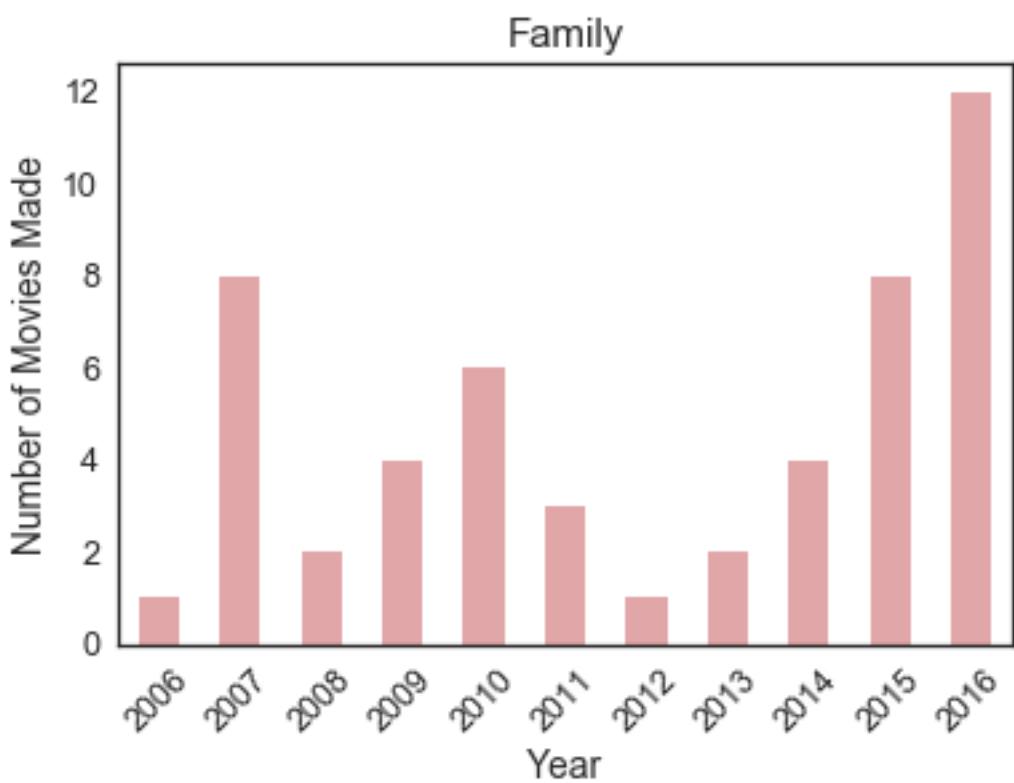
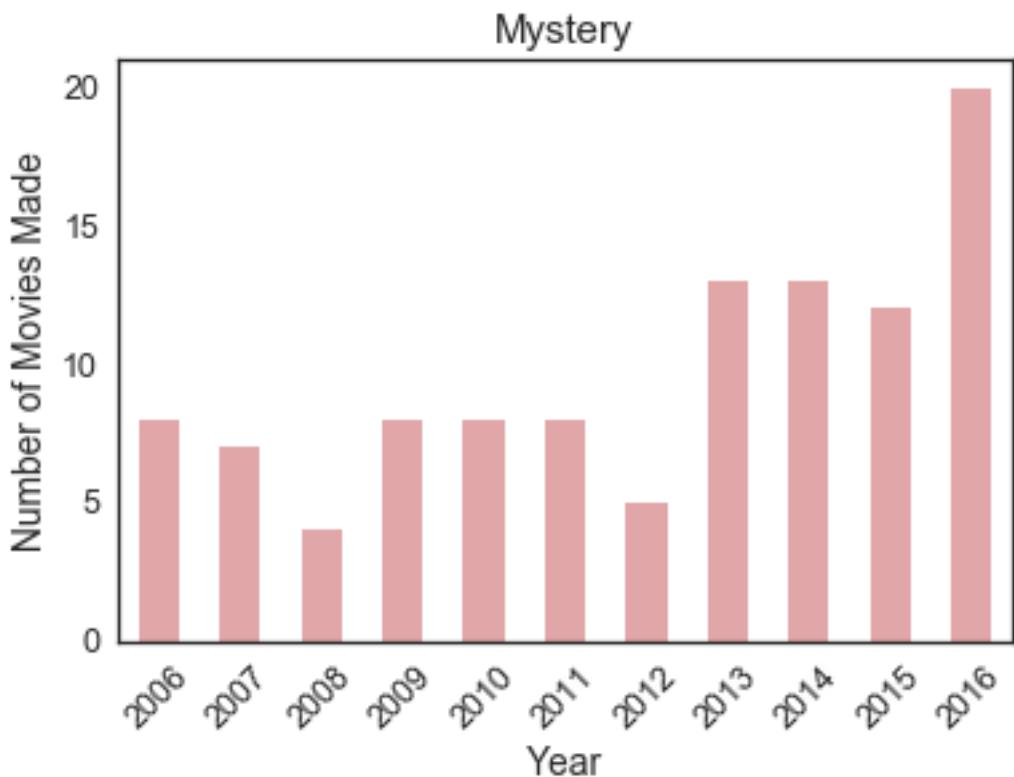
Music 21

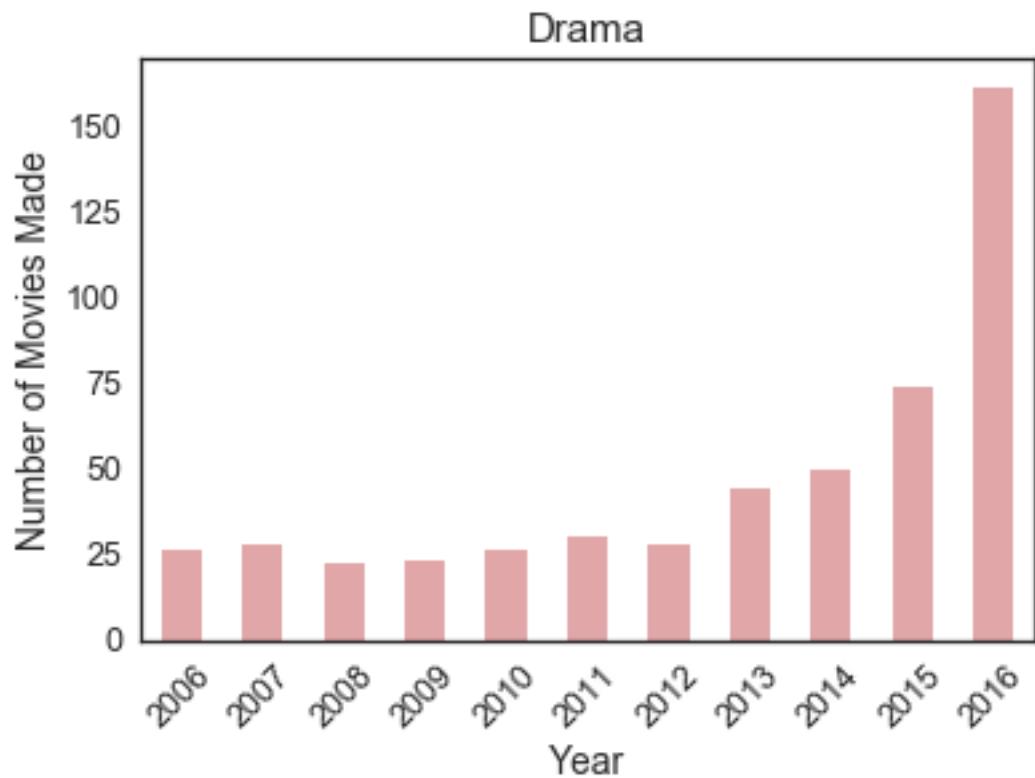
Comedy 279  
Romance 141  
Action 303  
Mystery 106  
Family 51  
Drama 513  
Biography 81  
Sci-Fi 120  
History 29  
Musical 5  
Adventure 259  
Thriller 195  
Fantasy 101  
War 13  
Animation 49  
Sport 18  
Horror 119  
Western 7  
Crime 150



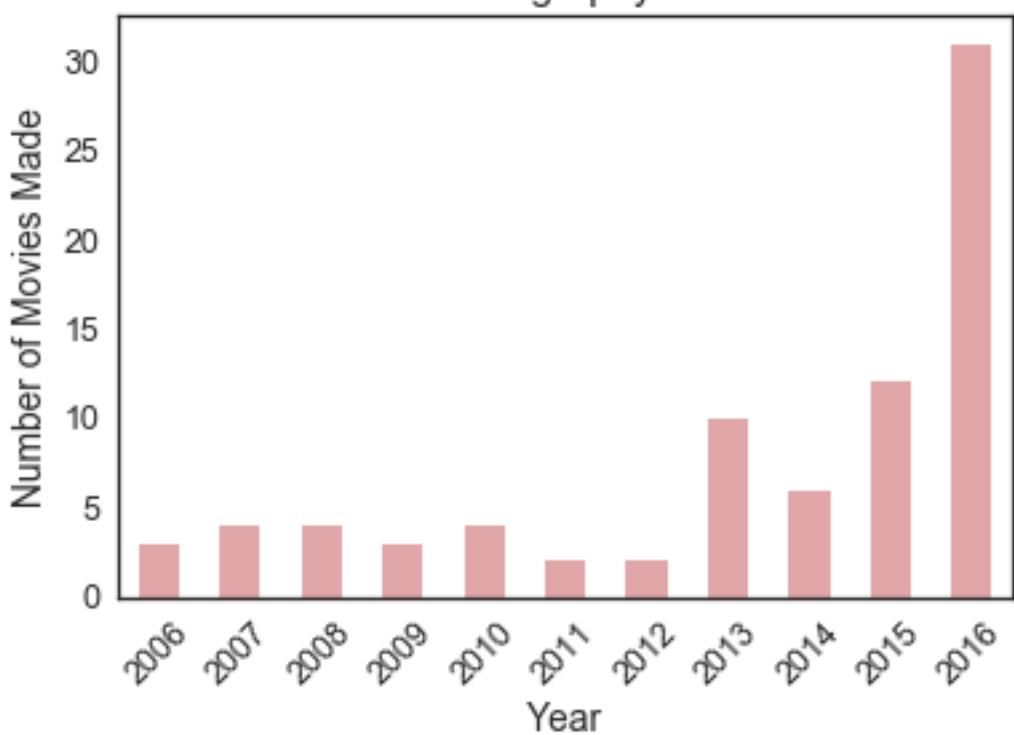




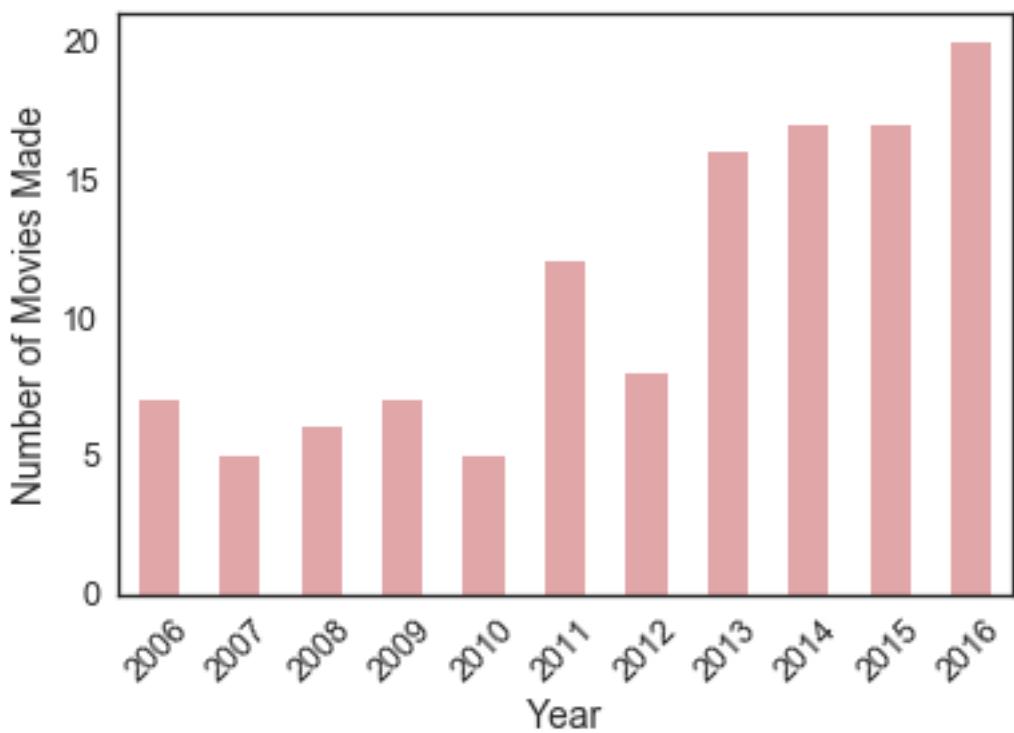


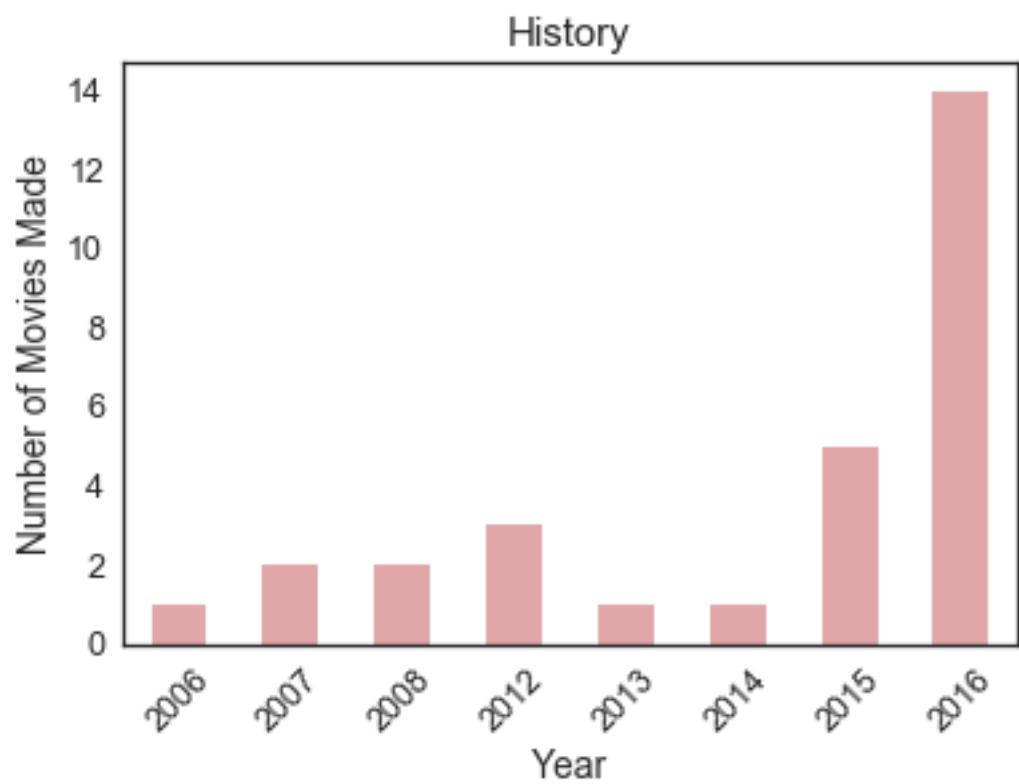


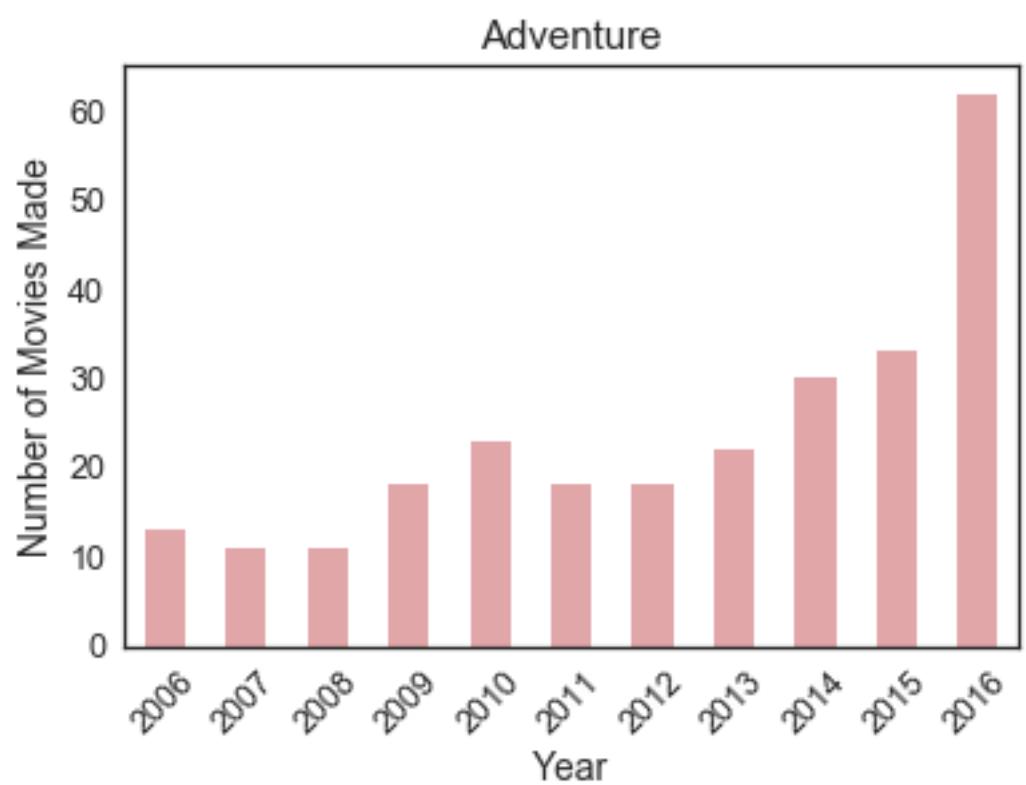
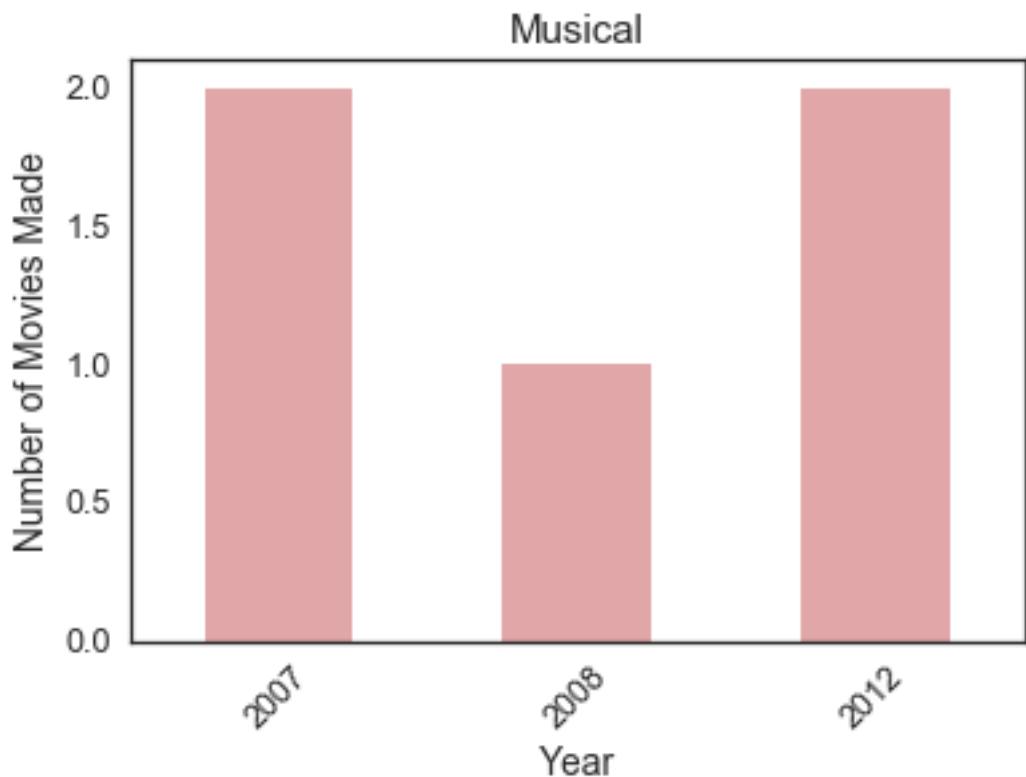
Biography

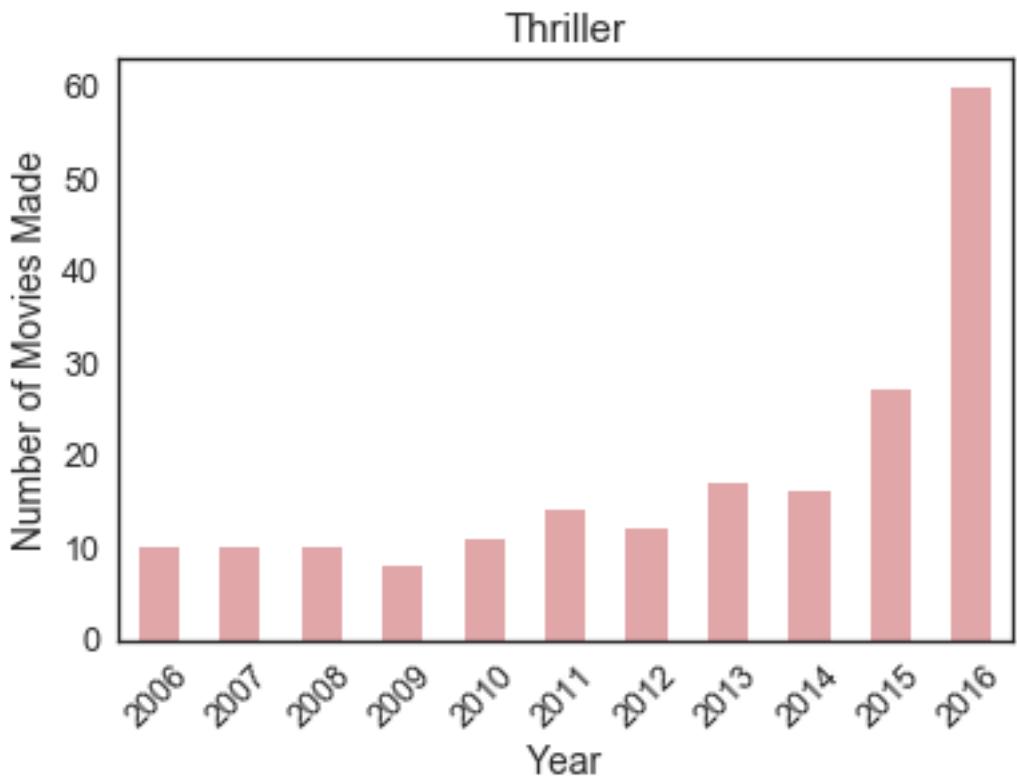


Sci-Fi

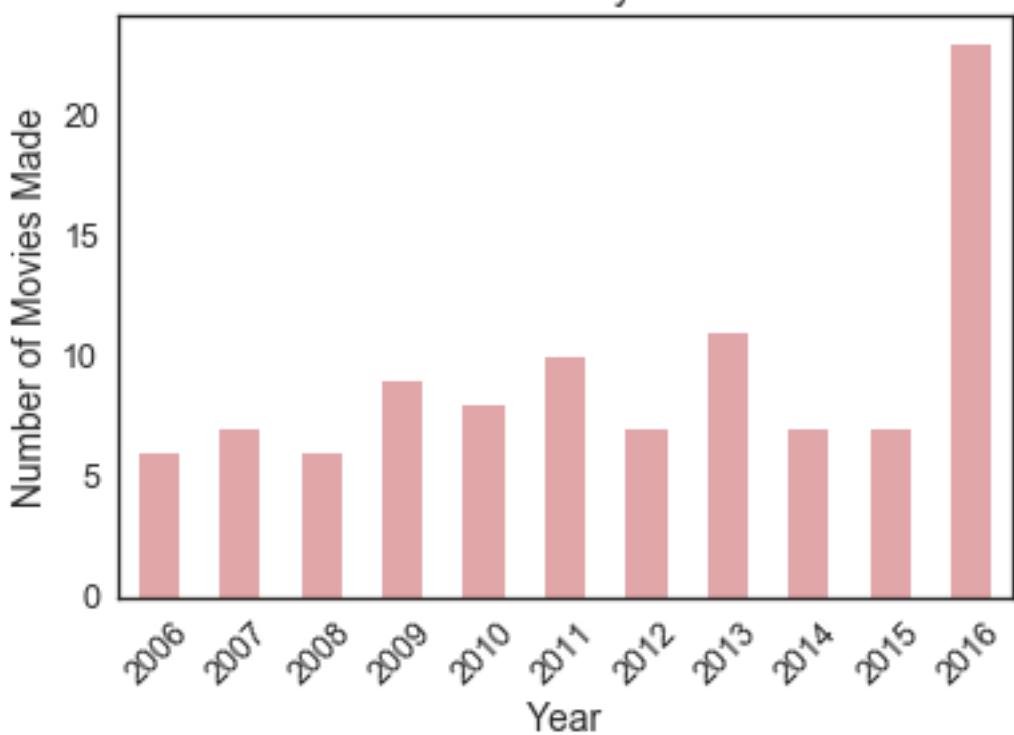




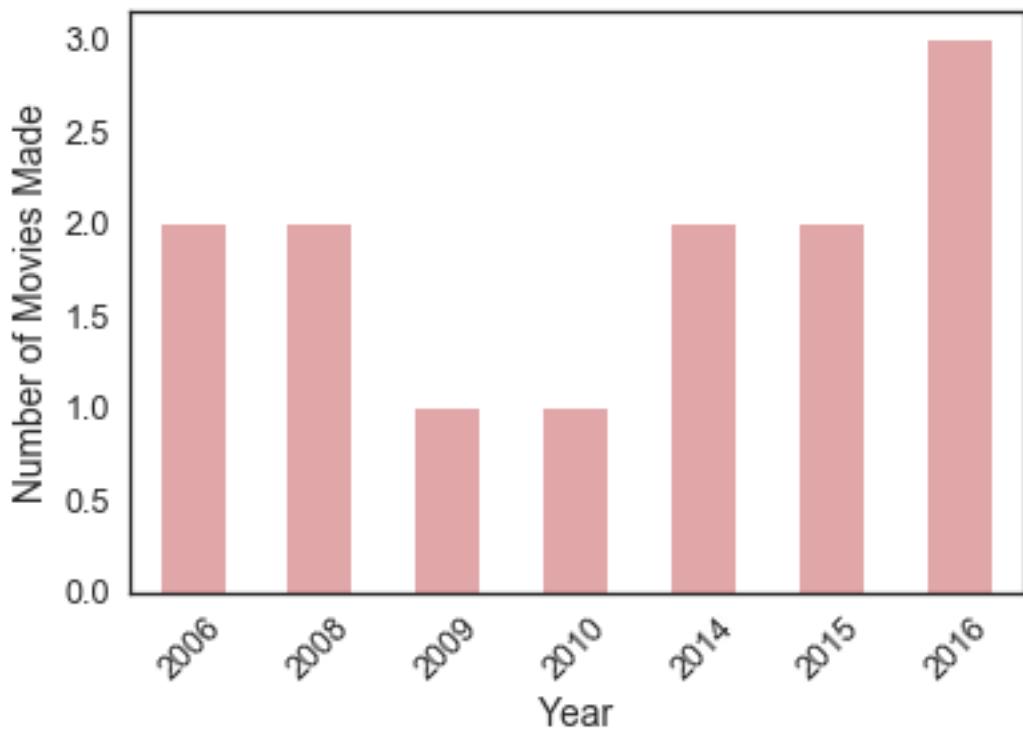


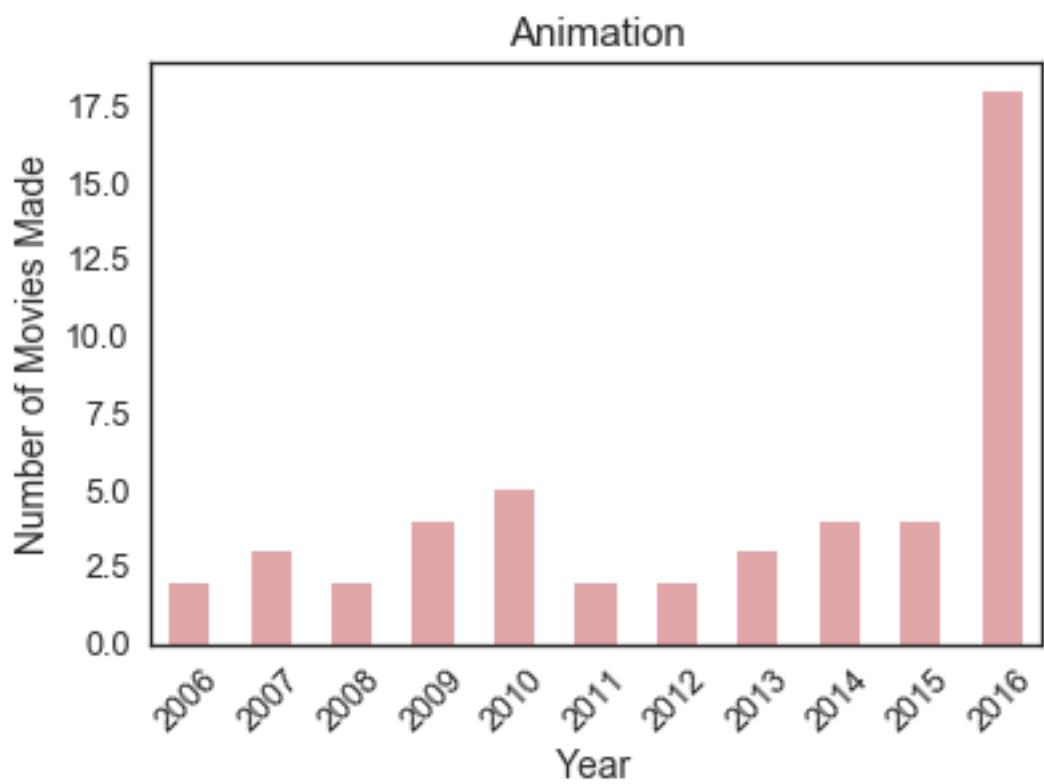


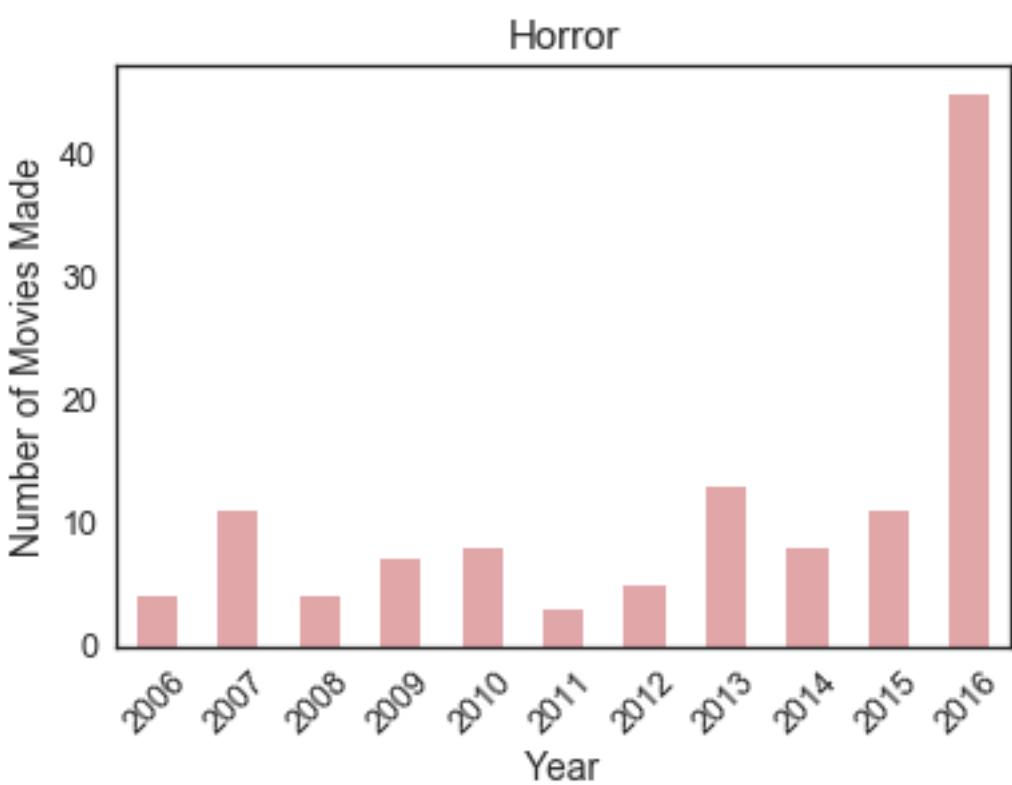
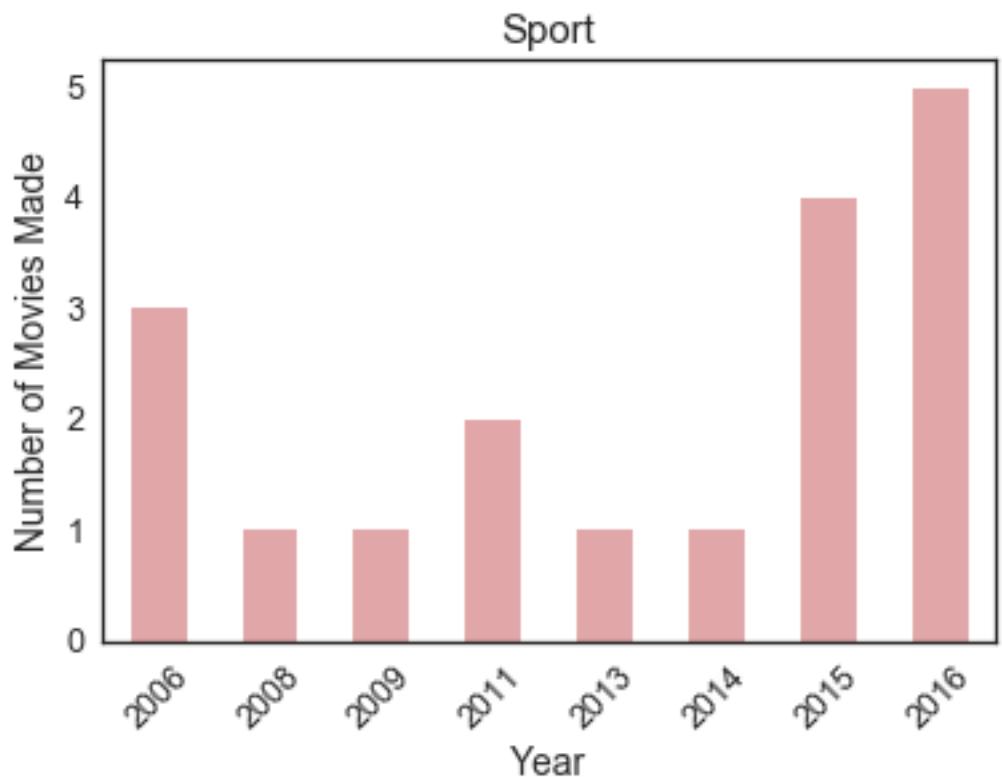
Fantasy

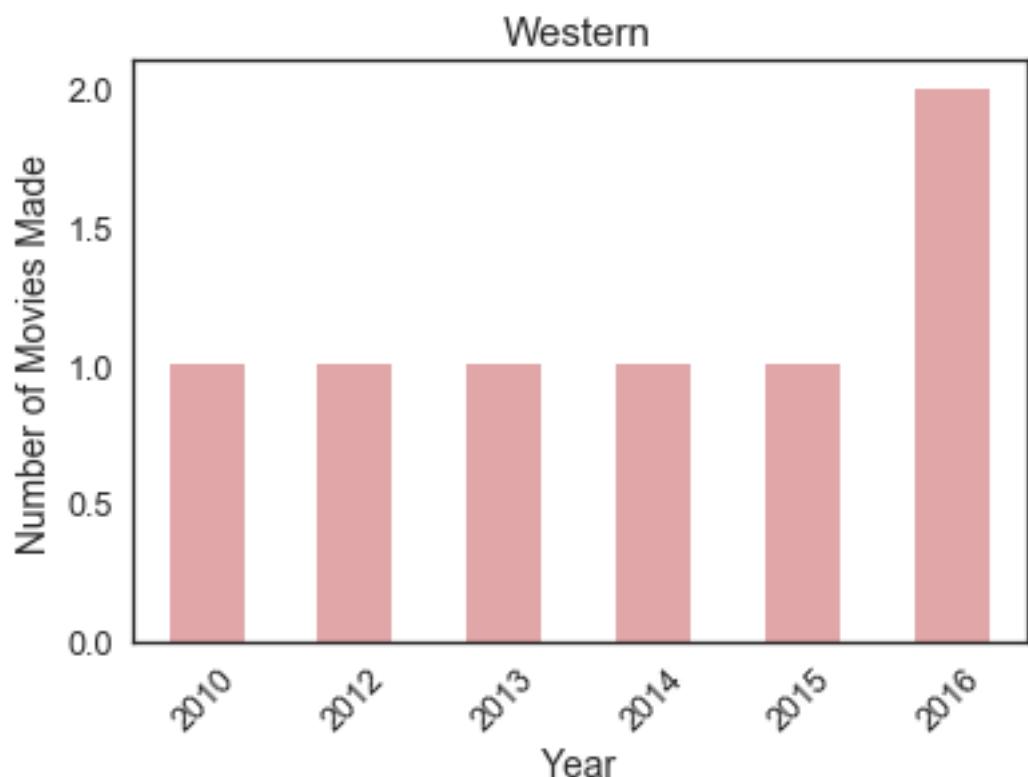


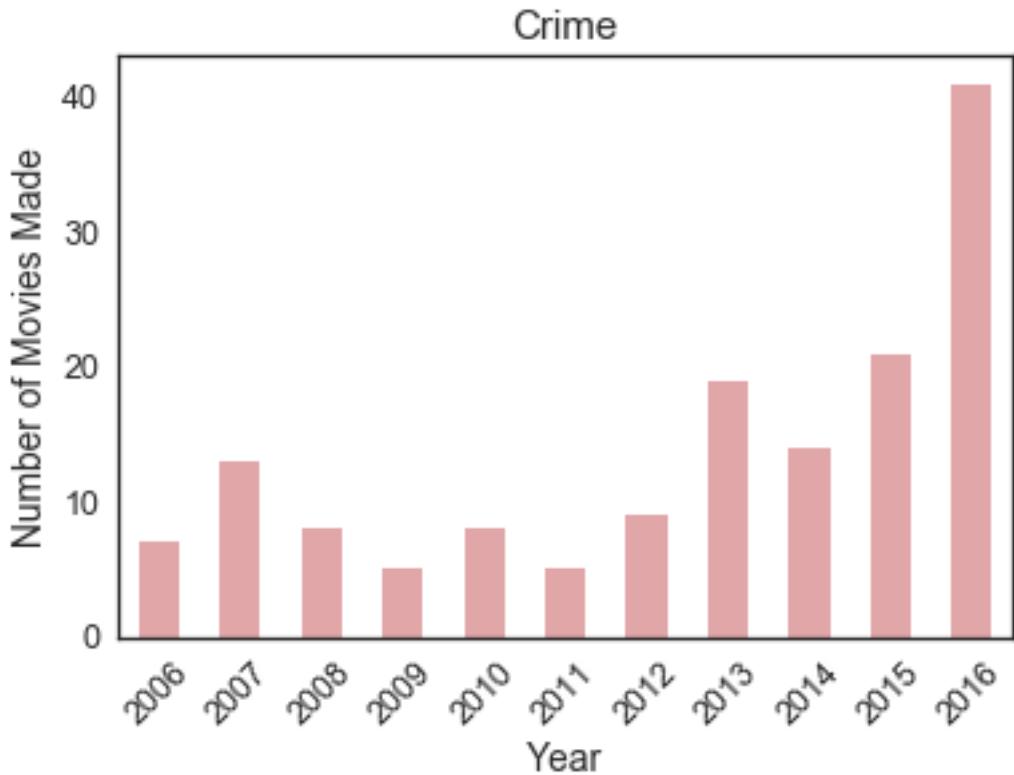
War











As all bar charts show above. Almost all kinds of movie trend to increasing their production as time going.

### 3.3.3 Question

**However, does it means all kinds of movie are become easier to make money than before?**

For answering this question, let's calculate the average revenue of different kind of movies. And sort them by year to draw bar charts.

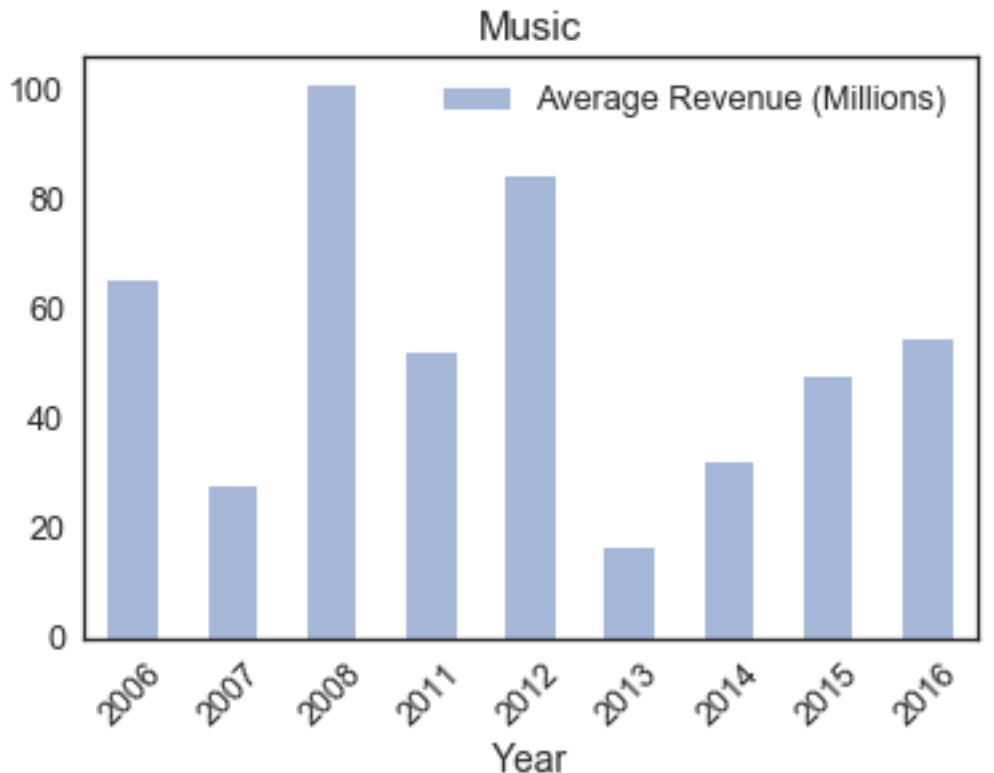
In [6]: #Count the average revenue of each genre of film by year

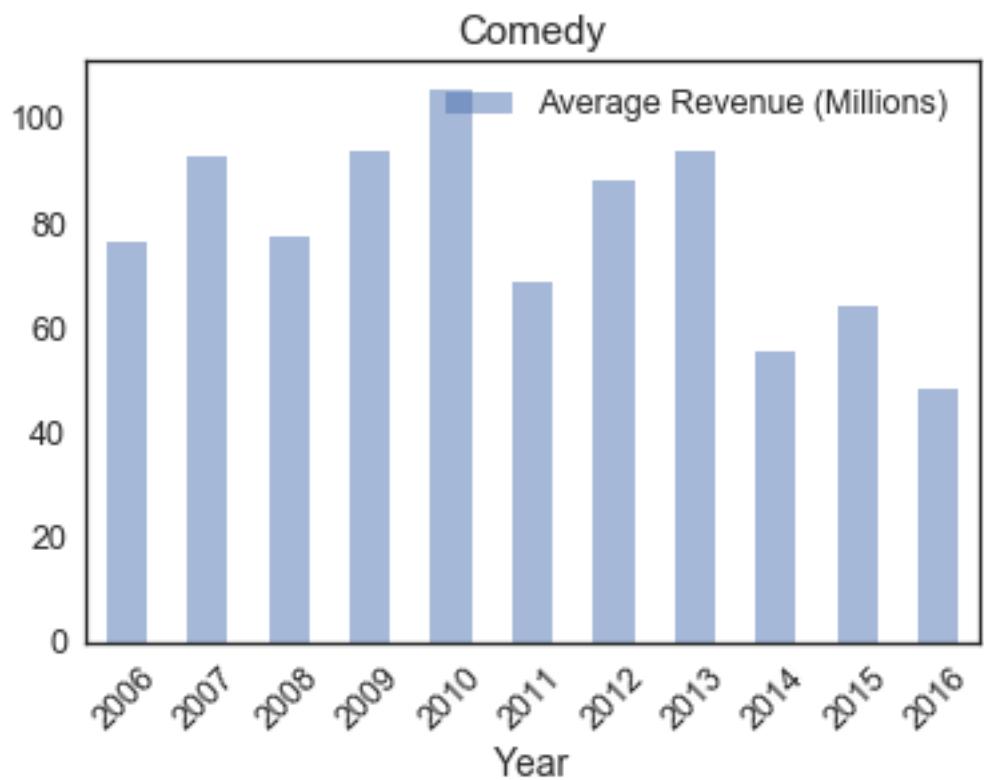
```
year={'Year':[2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016]}
YMR=pd.DataFrame(year)
for genre in individual_genres:
    current_genre = imdb['Genre'].str.contains(genre).fillna(False)
    df = imdb[current_genre]
    genre_revenue=df[['Year','Revenue (Millions)']].fillna(0)
    GR=genre_revenue.groupby('Year')['Revenue (Millions)'].agg(['sum','count'])
    GR.fillna(0)
    GR['Average Revenue (Millions)']=GR['sum']/GR['count']
    GR.plot(y='Average Revenue (Millions)',kind='bar',
             color='b', alpha=0.5, rot=45,title=genre)
    sum = genre_revenue['Revenue (Millions)'].sum()
```

```

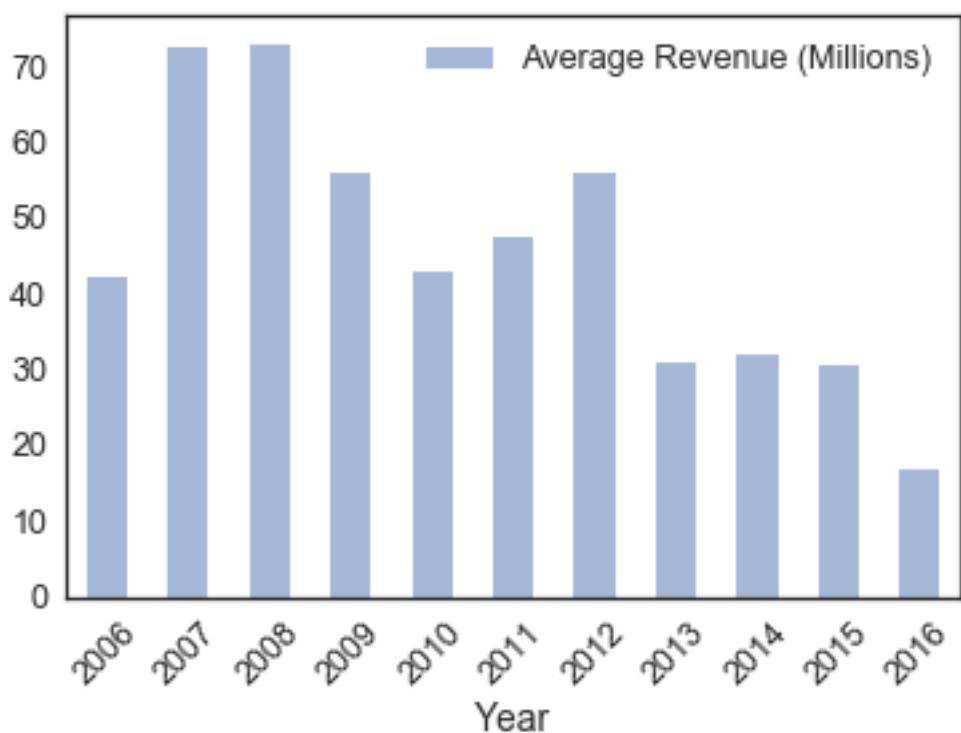
count = len(genre_revenue)
AR=GR['Average Revenue (Millions)'].values
if len(AR) == 11:
    YMR[genre]=[AR[0],AR[1],AR[2],AR[3],AR[4],AR[5],AR[6],AR[7],
                AR[8],AR[9],AR[10]]
elif len(AR) == 6:
    YMR[genre]=[np.nan,np.nan,np.nan,np.nan,np.nan,AR[0],np.nan,
                AR[1],AR[2],AR[3],AR[4],AR[5]]
elif len(AR) ==3:
    YMR[genre]=[np.nan,AR[0],AR[1],np.nan,np.nan,np.nan,
                AR[2],np.nan,np.nan,np.nan,np.nan]
elif len(AR) ==9:
    YMR[genre]=[AR[0],AR[1],AR[2],np.nan,np.nan,np.nan,
                AR[3],AR[4],AR[5],AR[6],AR[7],AR[8]]
elif len(AR) ==7:
    YMR[genre]=[AR[0],np.nan,AR[1],AR[2],AR[3],np.nan,np.nan,np.nan,
                AR[4],AR[5],AR[6]]
elif len(AR) ==8:
    if genre == 'Western':
        YMR[genre]=[np.nan,np.nan,np.nan,np.nan,np.nan,AR[0],np.nan,AR[1],
                    AR[2],AR[3],AR[4],AR[5]]
    else:
        YMR[genre]=[AR[0],AR[1],AR[2],np.nan,np.nan,np.nan,AR[3],AR[4],
                    AR[5],AR[6],AR[7]]

```

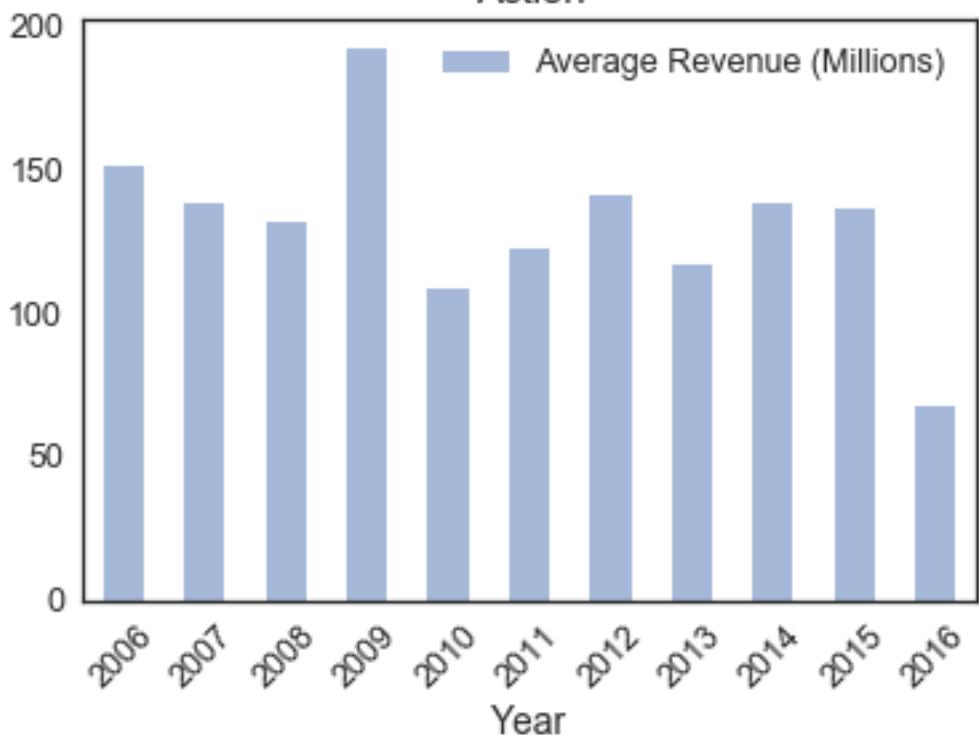


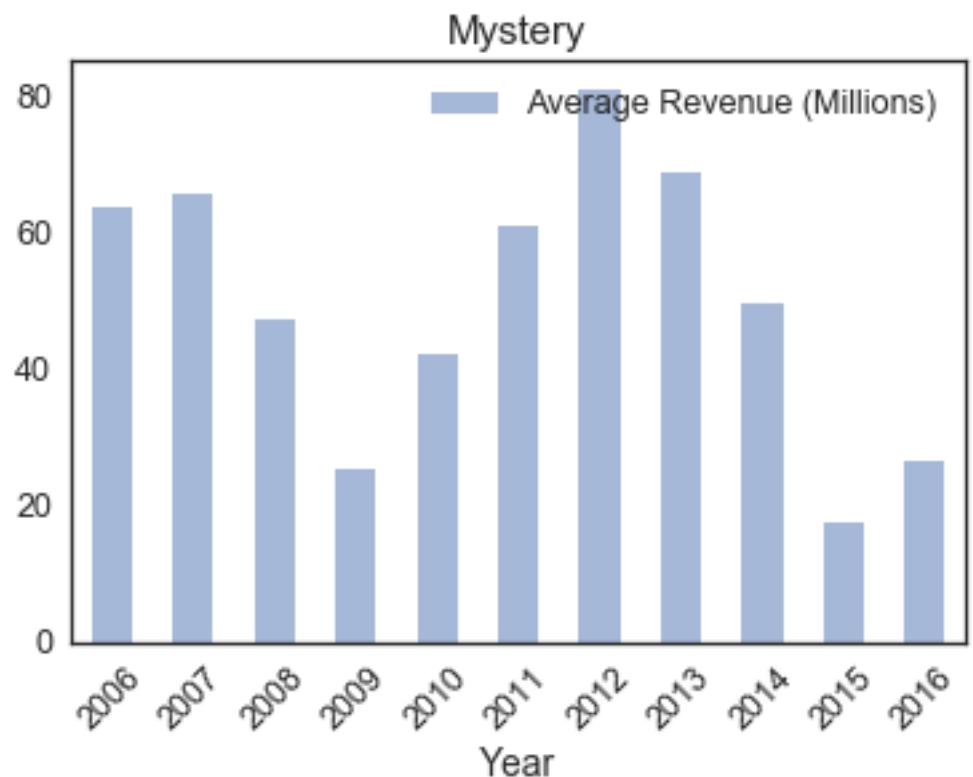


Romance

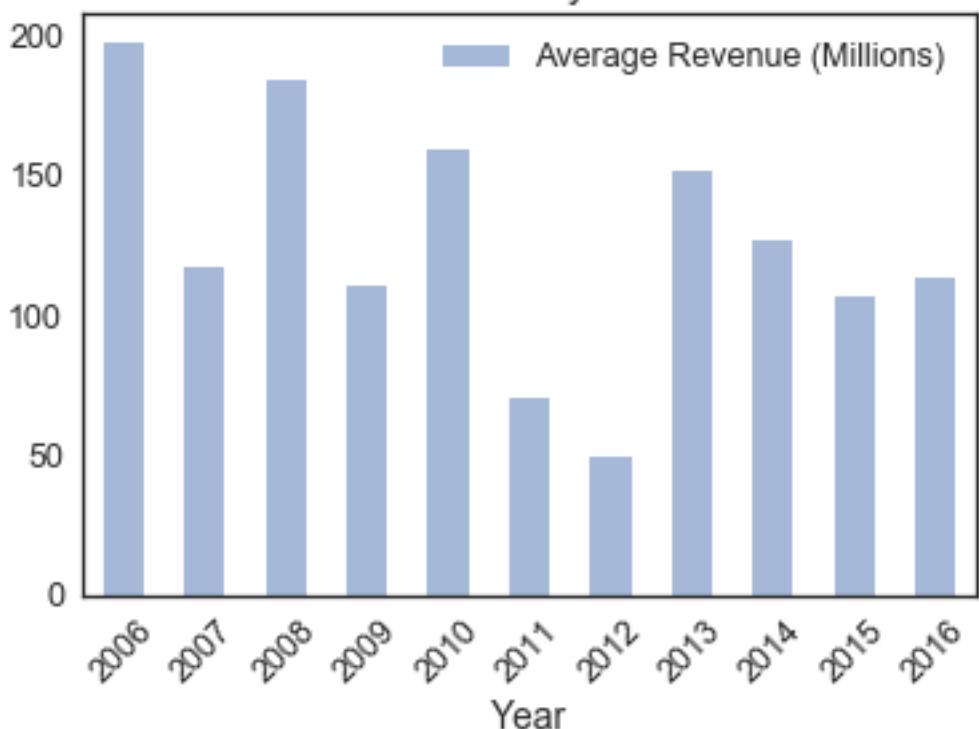


Action

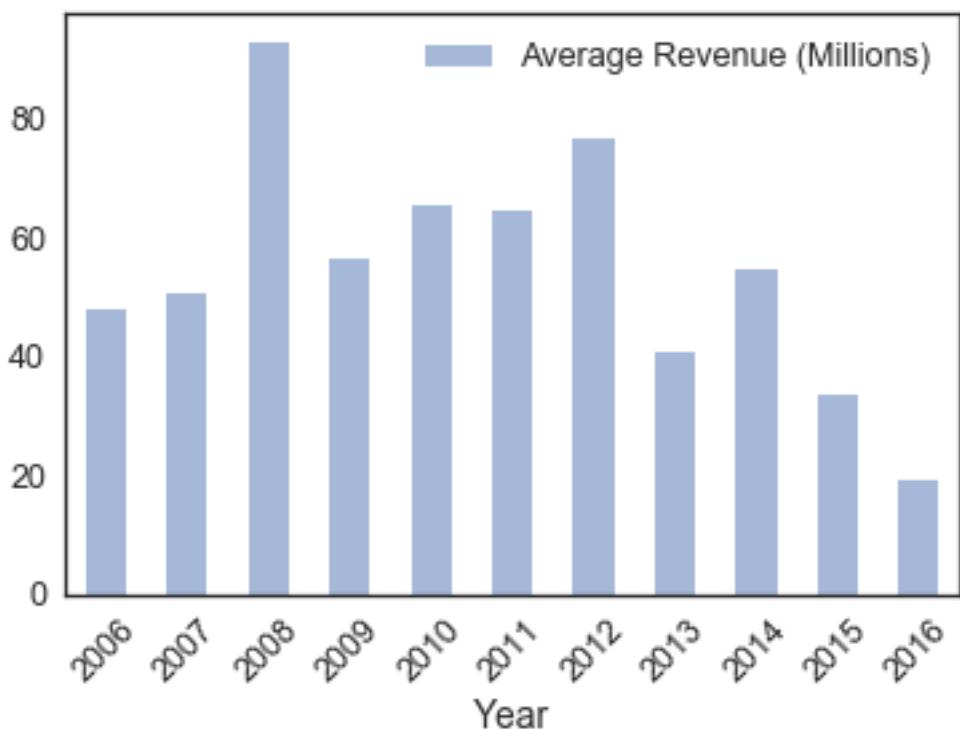




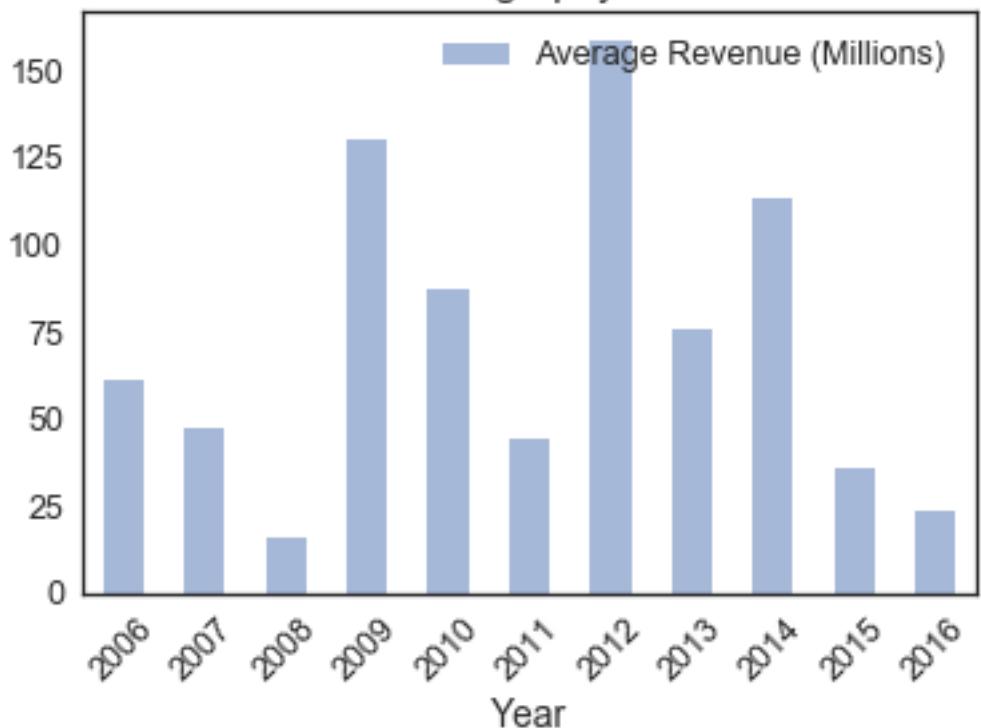
Family



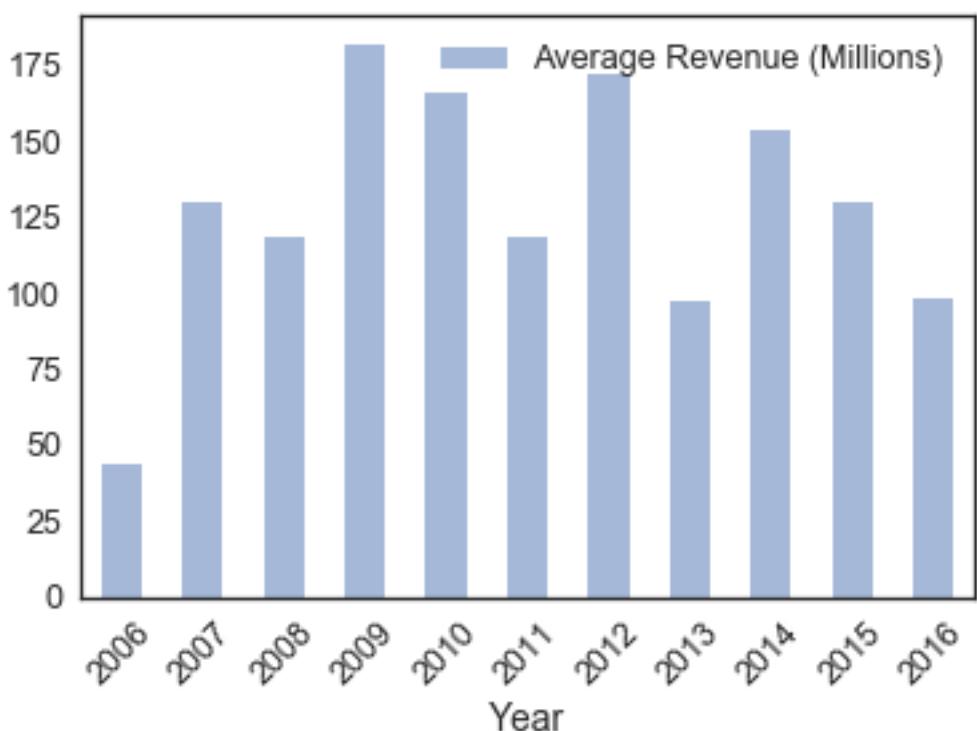
Drama



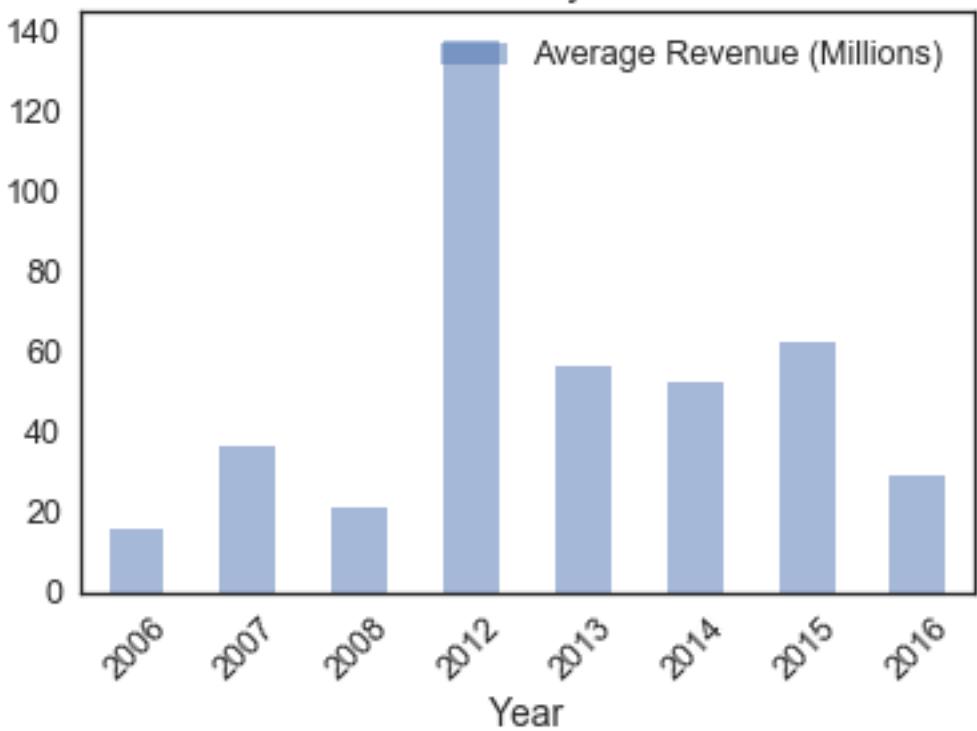
## Biography



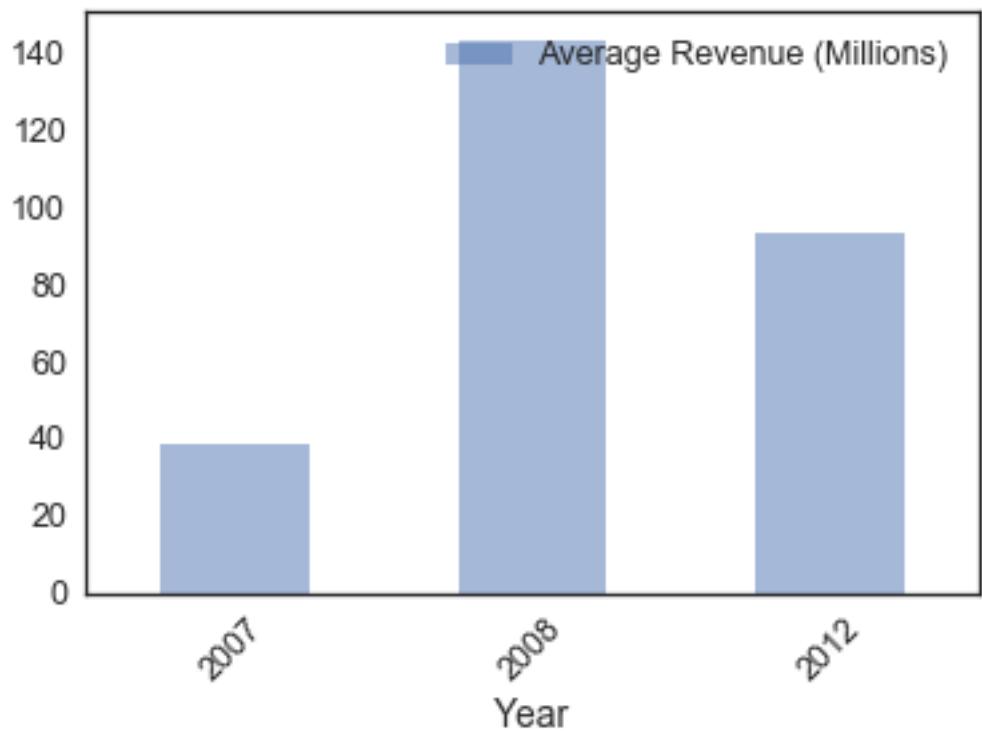
Sci-Fi



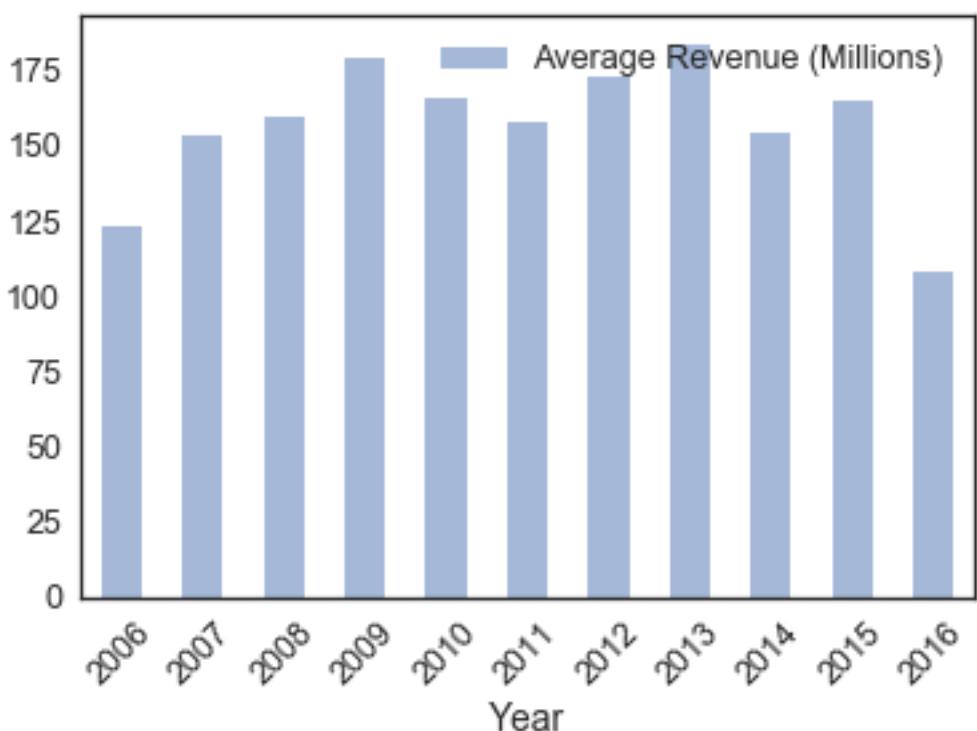
History



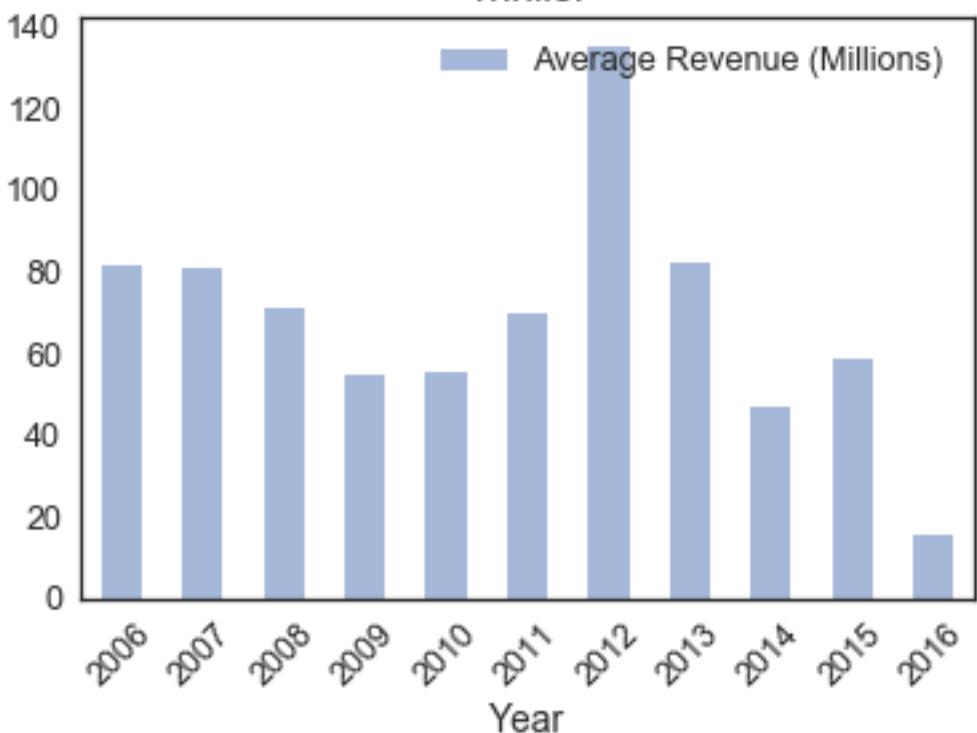
Musical



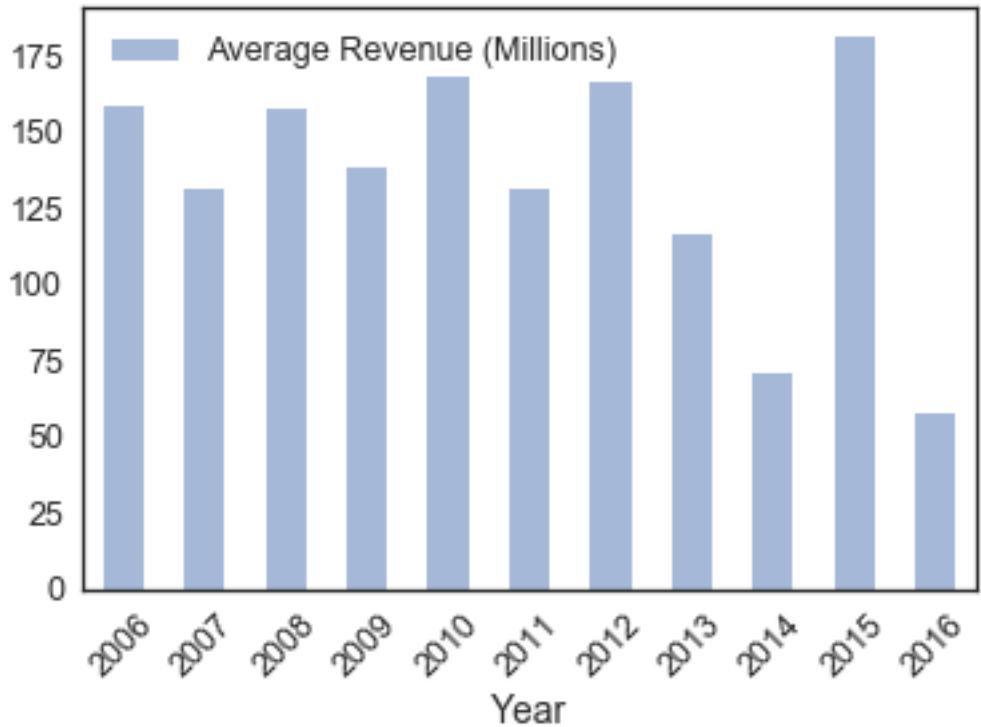
Adventure



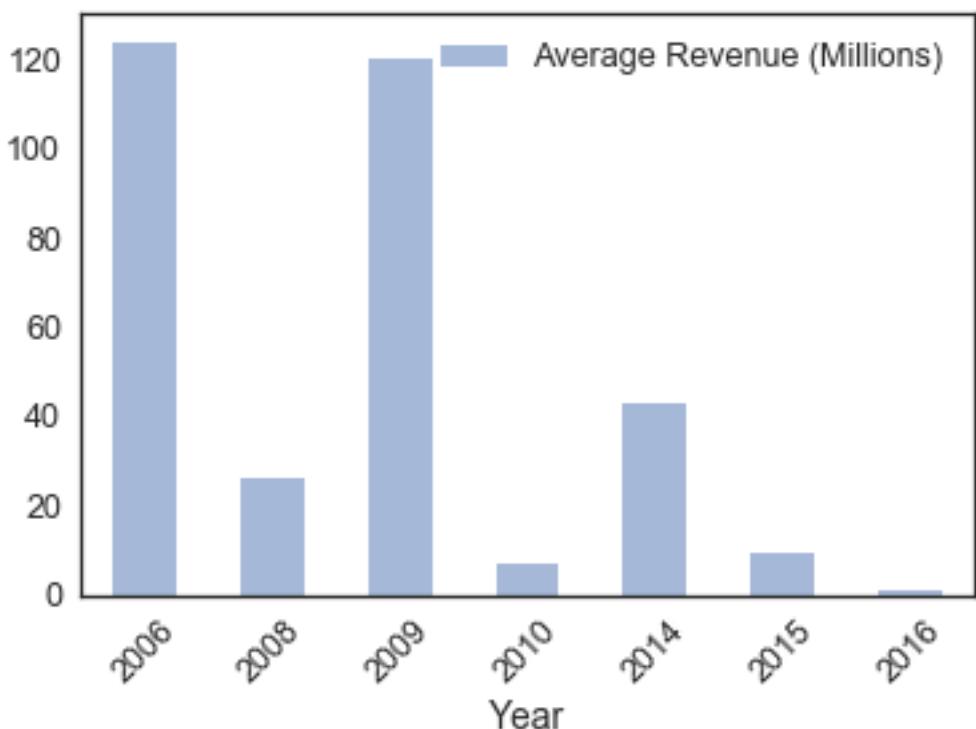
Thriller



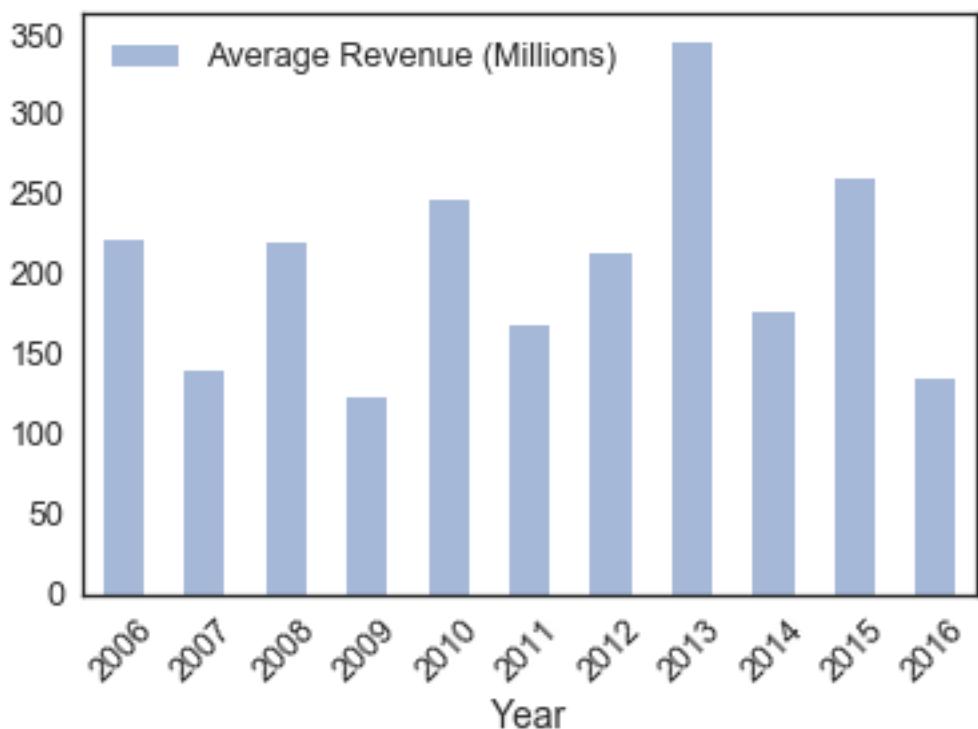
Fantasy

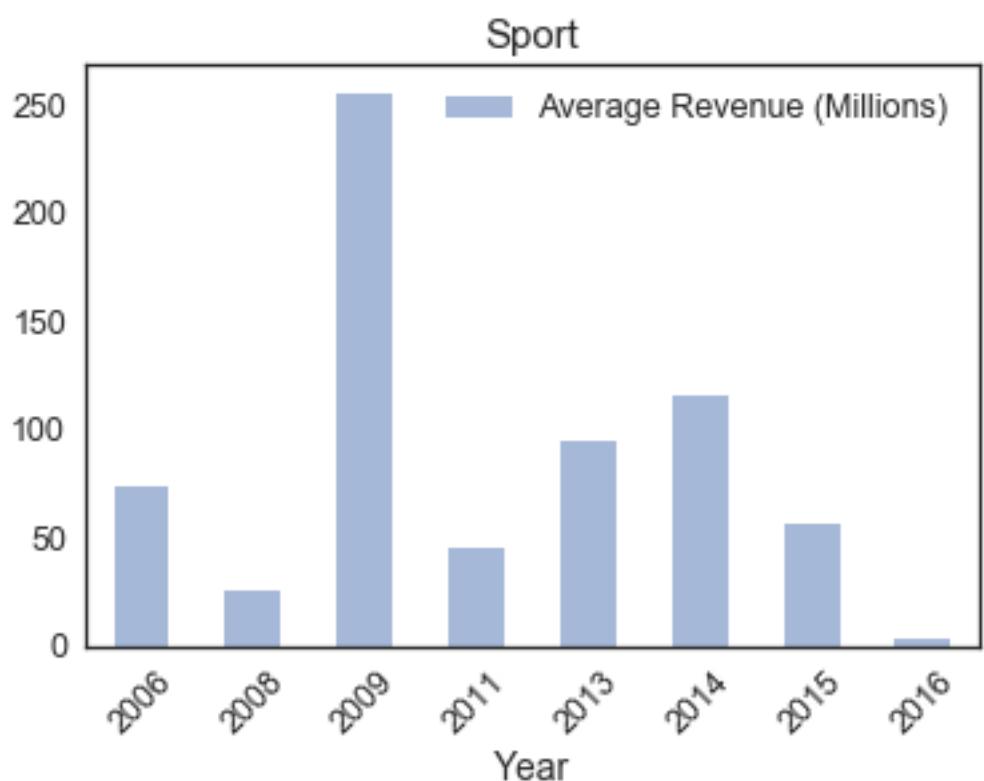


War

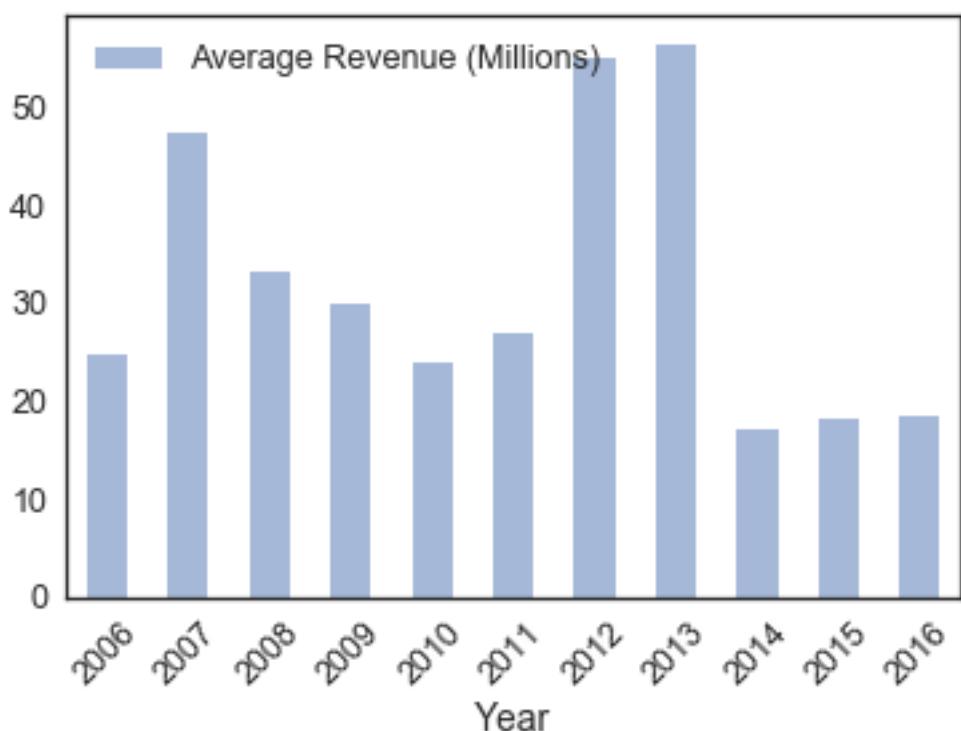


Animation

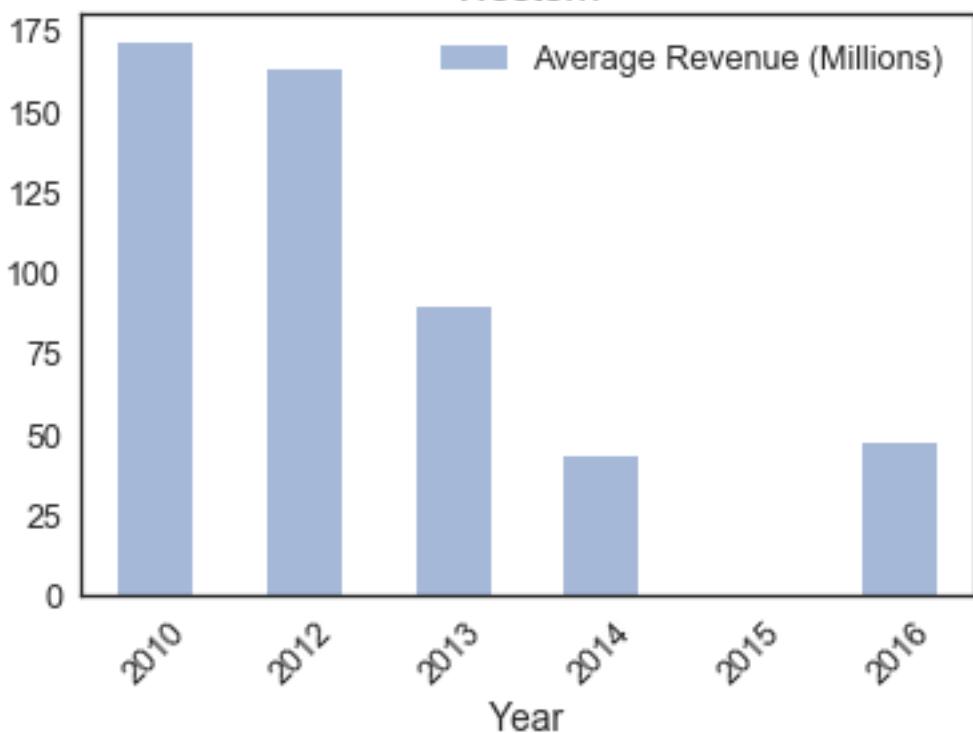


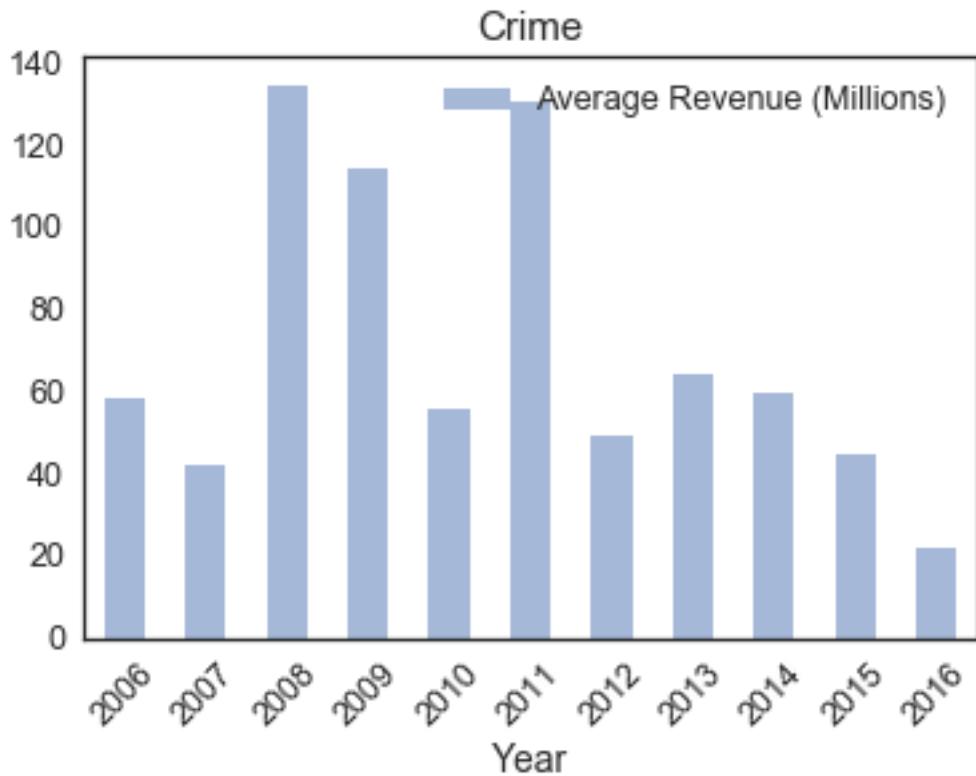


**Horror**



**Western**





Above bar charts show us that opposite to the number of movie made. The revenue of each kind of movie is decreasing as time going.

Although the result shows us that movie industry is falling down, we still want to know which genre's movie can make most money.

Here, we get a table of average revenue of all genres in different years. (**NaN means there have no movie made in that year.**)

In [7]: `YMR.head(11)`

Out[7] :	Year	Music	Comedy	Romance	Action	Mystery	\
0	2006	65.270000	76.336667	42.077500	151.273636	63.836250	
1	2007	27.520000	92.704000	72.525000	138.478333	65.448571	
2	2008	100.855000	77.442500	73.094167	131.270000	47.355000	
3	2009	NaN	93.662857	55.995000	192.460000	25.296250	
4	2010	NaN	105.794444	42.827143	108.455200	41.920000	
5	2011	51.780000	68.814211	47.385625	122.569000	61.068750	
6	2012	84.096667	88.211905	56.084545	140.509524	81.050000	
7	2013	16.170000	93.633043	30.794615	116.272759	68.596154	
8	2014	31.775000	55.254333	31.982222	138.251892	49.429231	
9	2015	47.432500	63.951892	30.486667	136.440571	17.505833	

10	2016	54.560000	48.086145	16.657667	67.448533	26.384000															
0		Family	Drama	Biography	Sci-Fi	...	Musical	\													
0	197.990000	48.170000	61.343333	43.827143	...	...	Nan														
1	117.103750	50.500000	47.240000	129.854000	...	...	38.610														
2	183.755000	93.149545	15.692500	118.931667	...	...	143.700														
3	110.347500	56.290870	129.940000	182.371429	...	...	Nan														
4	159.753333	65.632692	86.905000	165.740000	...	...	Nan														
5	70.656667	64.749333	44.395000	118.510000	...	...	Nan														
6	49.000000	76.583214	159.110000	172.352500	...	...	93.645														
7	151.730000	40.752273	75.484000	97.255000	...	...	Nan														
8	127.000000	54.544000	113.780000	153.545294	...	...	Nan														
9	106.370000	33.529730	36.135833	129.831765	...	...	Nan														
10	113.081667	19.298086	23.669677	98.070000	...	...	Nan														
0		Adventure	Thriller	Fantasy	War	Animation	Sport	\													
0	123.024615	81.673000	158.346667	124.105	221.020000	73.606667															
1	153.278182	80.611000	131.707143	NaN	138.770000	24.850000															
2	159.570909	71.285000	158.075000	25.875	219.605000	255.950000															
3	179.450556	54.971250	138.285556	120.520	123.407500	NaN															
4	165.895652	55.117273	168.007500	6.860	246.604000	NaN															
5	157.611667	69.570000	131.715000	NaN	167.535000	NaN															
6	172.577778	135.587500	166.368571	NaN	213.345000	44.630000															
7	184.101818	82.172353	116.452727	NaN	345.760000	95.000000															
8	154.687333	46.583125	70.882857	42.855	176.850000	115.600000															
9	164.933636	58.363333	181.960000	9.350	259.877500	56.357500															
10	107.889677	15.741167	57.406087	0.860	134.296667	2.754000															
0		Horror	Western	Crime																	
0	24.682500		NaN	58.191429																	
1	47.376364		NaN	42.171538																	
2	33.157500		NaN	134.736250																	
3	29.895714		NaN	114.250000																	
4	23.941250		171.03	55.502500																	
5	26.920000		NaN	130.348000																	
6	54.896000		162.80	49.362222																	
7	56.520769		89.29	64.073684																	
8	17.073750		42.62	59.431429																	
9	18.185455		0.00	44.389048																	
10	18.525556		46.69	22.153171																	

[11 rows x 21 columns]

For easier to do operation on this table, we **pop** the column of 'year'. And take its **transformation**.

In [8]: YMR.pop('Year')  
YMR.T

Out[8]: 0 1 2 3 4 \

Music	65.270000	27.520000	100.855000	NaN	NaN
Comedy	76.336667	92.704000	77.442500	93.662857	105.794444
Romance	42.077500	72.525000	73.094167	55.995000	42.827143
Action	151.273636	138.478333	131.270000	192.460000	108.455200
Mystery	63.836250	65.448571	47.355000	25.296250	41.920000
Family	197.990000	117.103750	183.755000	110.347500	159.753333
Drama	48.170000	50.500000	93.149545	56.290870	65.632692
Biography	61.343333	47.240000	15.692500	129.940000	86.905000
Sci-Fi	43.827143	129.854000	118.931667	182.371429	165.740000
History	15.960000	36.635000	21.255000	NaN	NaN
Musical	NaN	38.610000	143.700000	NaN	NaN
Adventure	123.024615	153.278182	159.570909	179.450556	165.895652
Thriller	81.673000	80.611000	71.285000	54.971250	55.117273
Fantasy	158.346667	131.707143	158.075000	138.285556	168.007500
War	124.105000	NaN	25.875000	120.520000	6.860000
Animation	221.020000	138.770000	219.605000	123.407500	246.604000
Sport	73.606667	24.850000	255.950000	NaN	NaN
Horror	24.682500	47.376364	33.157500	29.895714	23.941250
Western	NaN	NaN	NaN	NaN	171.030000
Crime	58.191429	42.171538	134.736250	114.250000	55.502500

	5	6	7	8	9	\
Music	51.780000	84.096667	16.170000	31.775000	47.432500	
Comedy	68.814211	88.211905	93.633043	55.254333	63.951892	
Romance	47.385625	56.084545	30.794615	31.982222	30.486667	
Action	122.569000	140.509524	116.272759	138.251892	136.440571	
Mystery	61.068750	81.050000	68.596154	49.429231	17.505833	
Family	70.656667	49.000000	151.730000	127.000000	106.370000	
Drama	64.749333	76.583214	40.752273	54.544000	33.529730	
Biography	44.395000	159.110000	75.484000	113.780000	36.135833	
Sci-Fi	118.510000	172.352500	97.255000	153.545294	129.831765	
History	NaN	137.980000	56.670000	52.070000	62.456000	
Musical	NaN	93.645000	NaN	NaN	NaN	
Adventure	157.611667	172.577778	184.101818	154.687333	164.933636	
Thriller	69.570000	135.587500	82.172353	46.583125	58.363333	
Fantasy	131.715000	166.368571	116.452727	70.882857	181.960000	
War	NaN	NaN	NaN	42.855000	9.350000	
Animation	167.535000	213.345000	345.760000	176.850000	259.877500	
Sport	NaN	44.630000	95.000000	115.600000	56.357500	
Horror	26.920000	54.896000	56.520769	17.073750	18.185455	
Western	NaN	162.800000	89.290000	42.620000	0.000000	
Crime	130.348000	49.362222	64.073684	59.431429	44.389048	

	10
Music	54.560000
Comedy	48.086145
Romance	16.657667
Action	67.448533

Mystery	26.384000
Family	113.081667
Drama	19.298086
Biography	23.669677
Sci-Fi	98.070000
History	29.237143
Musical	NaN
Adventure	107.889677
Thriller	15.741167
Fantasy	57.406087
War	0.860000
Animation	134.296667
Sport	2.754000
Horror	18.525556
Western	46.690000
Crime	22.153171

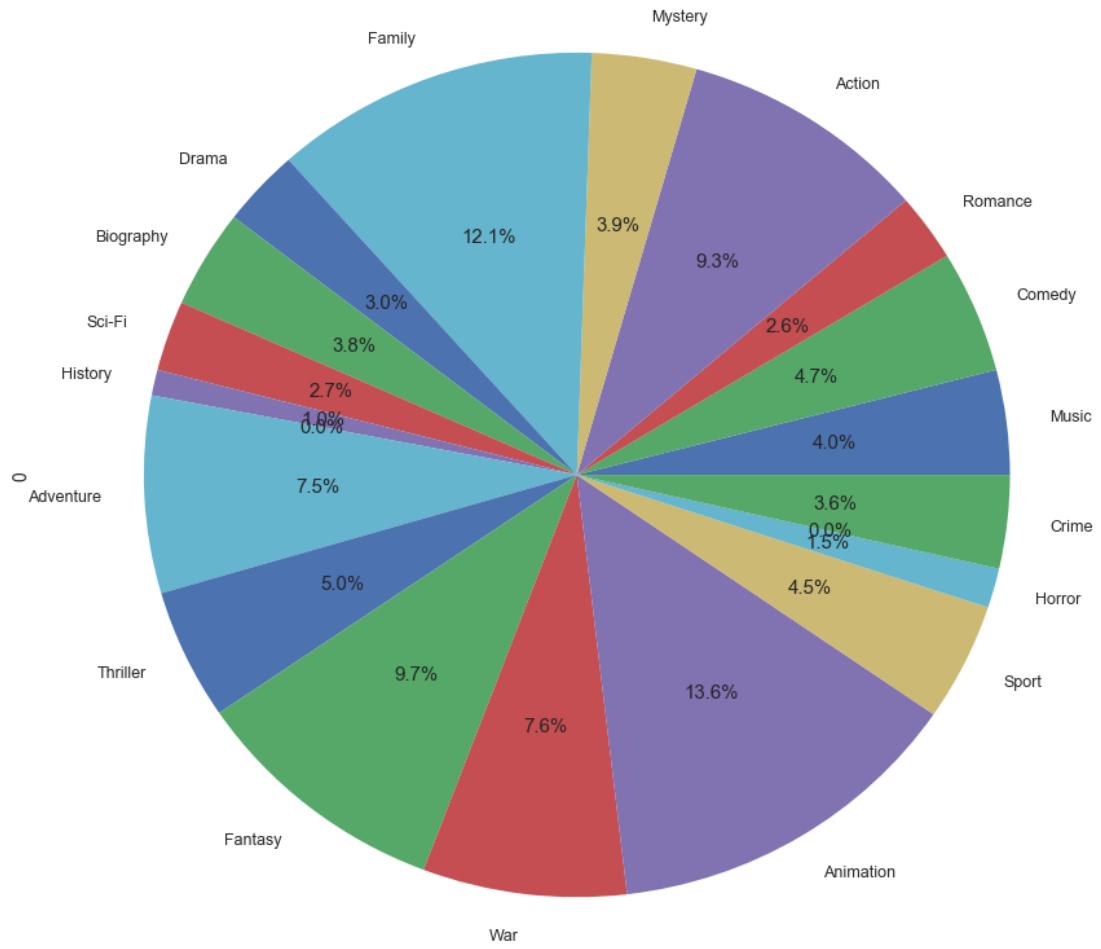
### 3.3.4 Pie Chart of Average Revenue

With the new table we get, we can plot pie charts as below. It can directly show the ranking of average revenue to us. Here, I choose three year to draw pie chart.

```
In [9]: YMR.T[0].plot.pie(subplots=True, figsize=(15, 15),
 autopct='%1.1f%%', title='Average Revenue Percentage in 2006')
```

```
Out[9]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x7fa0c62203c8>], dtype=object)
```

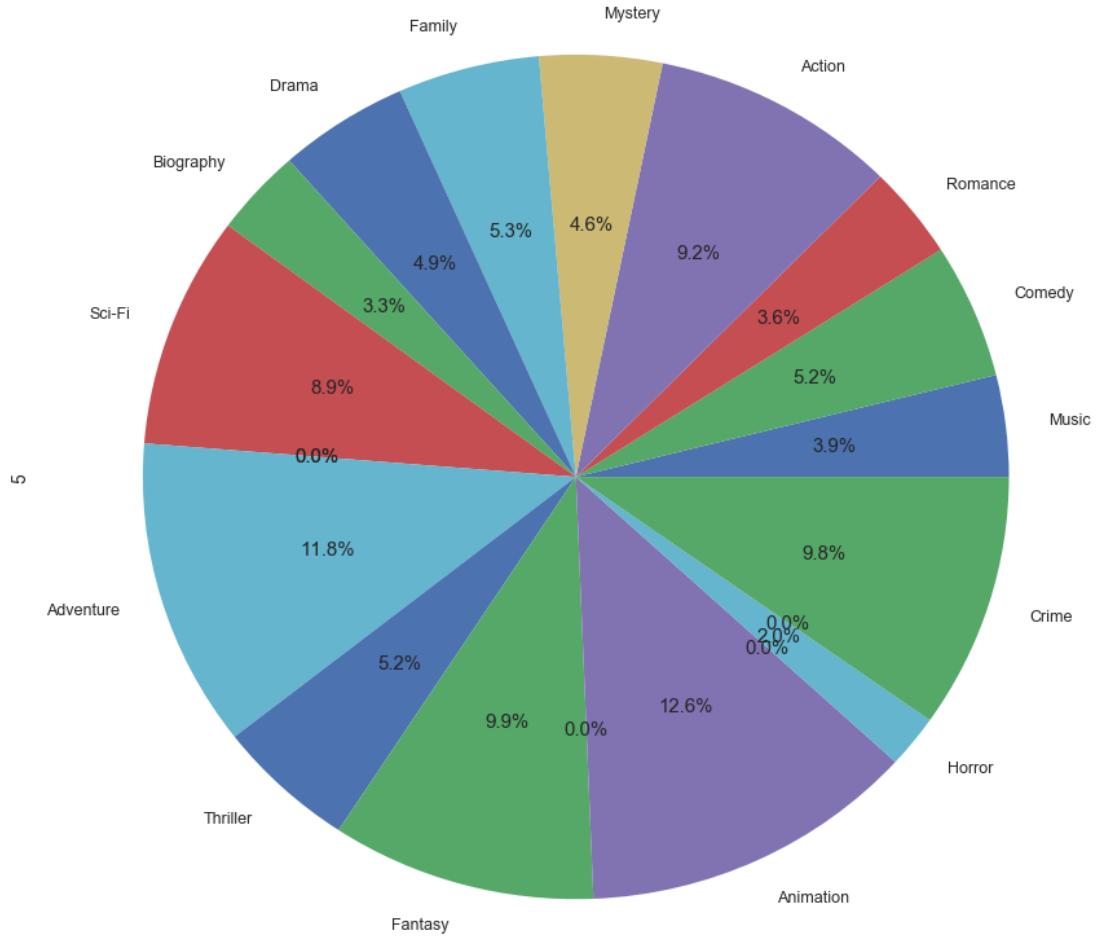
Average Revenue Percentage in 2006



```
In [10]: YMR.T[5].plot.pie(subplots=True, figsize=(15, 15),
                           autopct='%.1f%%', title='Average Revenue Percentage in 2011')
```

```
Out[10]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x7fa0c31c3630>], dtype=object)
```

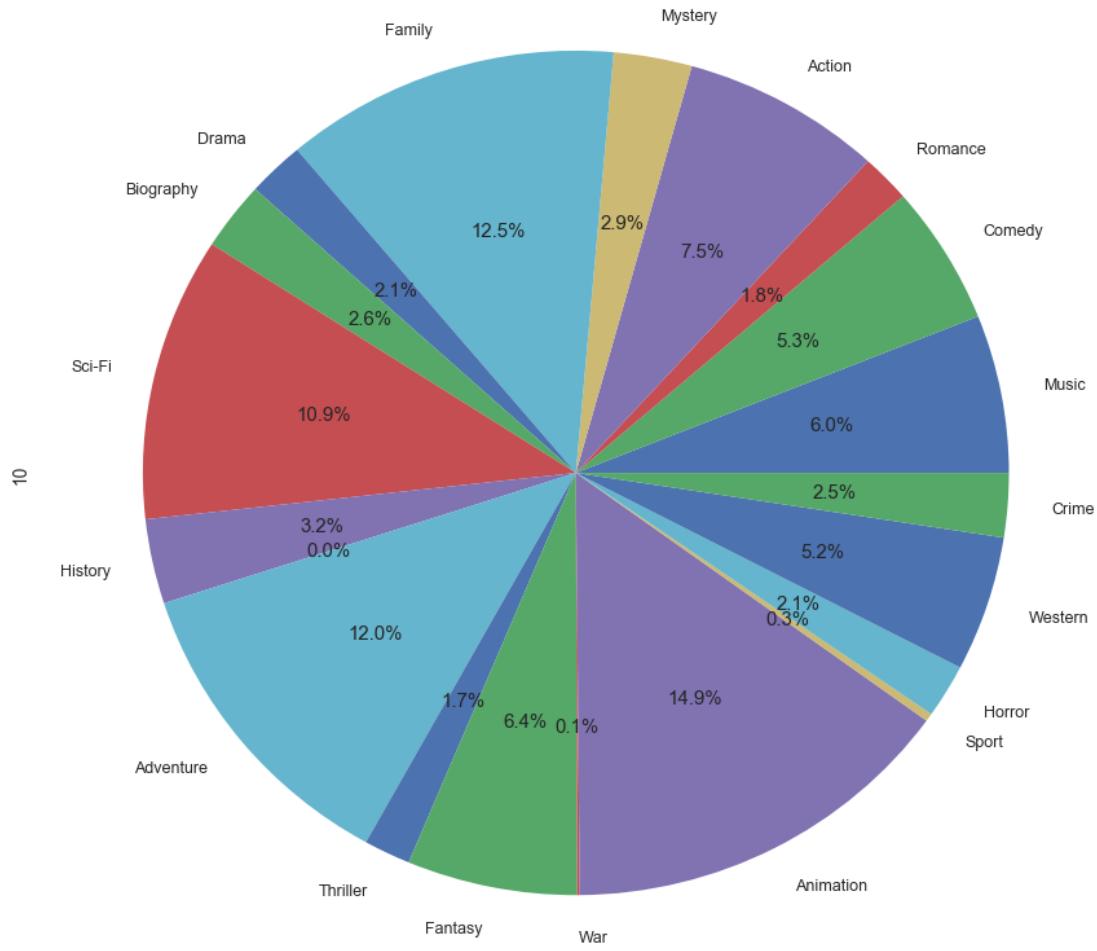
Average Revenue Percentage in 2011



```
In [11]: YMR.T[10].plot.pie(subplots=True, figsize=(15, 15),
                           autopct='%1.1f%%', title='Average Revenue Percentage in 2016')
```

```
Out[11]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x7fa0c311f4e0>], dtype=object)
```

Average Revenue Percentage in 2016



### 3.3.5 Summary

As above pie chart show, we know that Animation movie is easiest to make money. Sci-Fi movie are getting easier to earn money than other kinds of movie. However, the average revenue of all kinds of movie are trend to decreasing. Maybe we need a revolution in movie industry.

## 3.4 Analysis on Rating and Metascore

This analysis is provided by Kedun (Covey) Liu, which aims to find out the relationship between metascore and rating.

- The following Excel table analyze the relationship between Rating and MetaScore and find out the outlier.
- We analysis the outlier of what the genre they belong.
- We can finally learn that what kind of movie has the most controversial comment.

This analysis was done by hand in Excel.

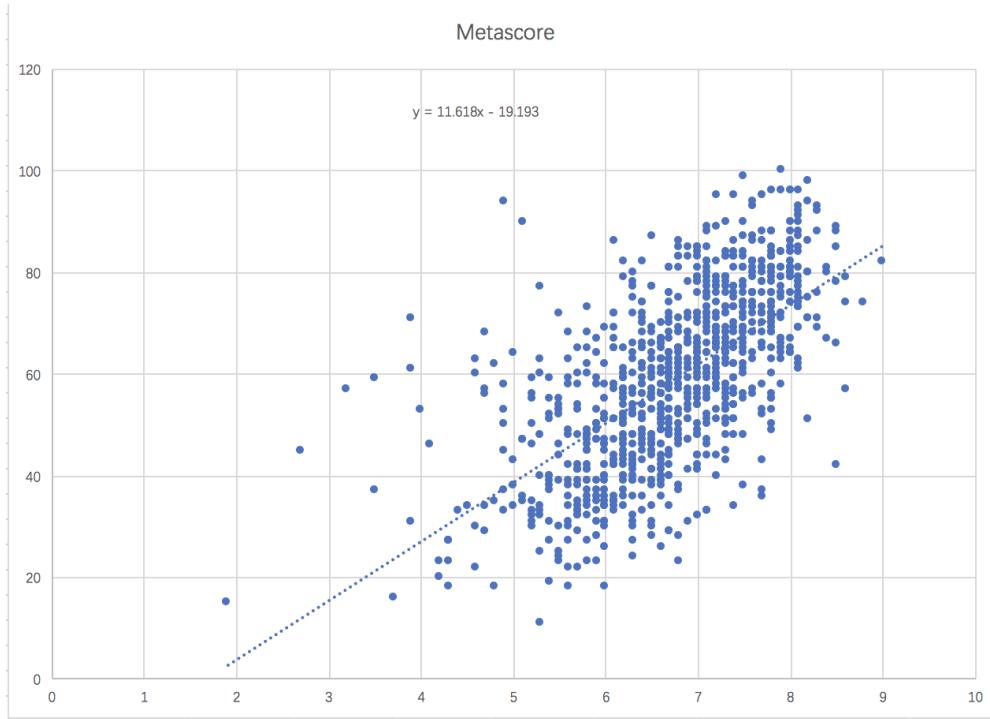
### 3.4.1 Implementation

Import the data into Excel.

Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Mi)	Votes	Revenue (M)	Rating	Metascore	line
1	Guardians o	Action	Adve A group of i	James Gunn	Chris Pratt, \	2014	121	757074	333.13	8.1	76	74.9128
2	Prometheus	Adventure,N	Following cl	Ridley Scott	Noomi Rap	2012	124	485820	126.46	7	65	62.133
3	Split	Horror,Thrill	Three girls a M. Night Sh	James McAv	2016	117	157606	138.12	7.3	62	65.6184	
4	Sing	Animation,C	In a city of h	Christophe l	Matthew Mc	2016	108	60545	270.32	7.2	59	64.4566
5	Suicide Squi	Action	Adve A secret gov	David Ayer	Will Smith, J	2016	123	393727	325.02	6.2	40	52.8386
6	The Great W	Action	Adve European m	Yimou Zhang	Matt Damon	2016	103	56036	45.13	6.1	42	51.6768
7	La La Land	Comedy,Dra	A jazz pianis	Damien Cha	Ryan Goslin	2016	128	258682	151.06	8.3	93	77.2364
8	Mindhorn	Comedy	A has-been Sean Foley	Essie Davis, I	2016	89	2490			6.4	71	55.1622
9	The Lost City	Action,Adve	A true-life d James Gray	Charlie Hun	2016	141	7188	8.01	7.1	78	63.2948	
10	Passengers	Adventure,C	A spacecraft Morten Tyd	Jennifer Law	2016	116	192177	100.01		7	41	62.133
11	Fantastic Be	Adventure,F	The adventu	David Yates	Eddie Redm	2016	133	232072	234.02	7.5	66	67.942
12	Hidden Figu	Biography,D	The story of Theodore M Taraji P. Her	2016	127	93103	169.27	7.8	74	71.4274		
13	Rogue One	Action,Adve	The Rebel A Gareth Edw	Felicity Jone	2016	133	323118	532.17	7.9	65	72.5892	
14	Moana	Animation,A	In Ancient P Ron Clement	Auli'i Craval	2016	107	118151	248.75	7.7	81	70.2656	
15	Colossal	Action,Comi	Gloria is an i Nacho Vigal Anne Hatha	2016	109	8612	2.87		6.4	70	55.1622	
16	The Secret L	Animation,A	The quiet lifi Chris Renaud Louis C.K, E	2016	87	120259	368.31		6.6	61	57.4858	
17	Hacksaw Ric	Biography,D	WWII Ameri Mel Gibson	Andrew Gar	2016	139	211760	67.12	8.2	71	76.0746	
18	Jason Bourn	Action,Thrill	The CIA's m Paul Greeng	Matt Damon	2016	123	150823	162.16	6.7	58	58.6476	
19	Lion	Biography,D	A five-year- Garth Davis	Dev Patel, N	2016	118	102061	51.69	8.1	69	74.9128	
20	Arrival	Drama,Myst	When twelv Denis Villen Amy Adams	2016	116	340798	100.5		8	81	73.751	
21	Gold	Adventure,C	Kenny Wells Stephen Ga	Matthew Mc	2016	120	19053	7.22	6.7	49	58.6476	
22	Manchester Drama	A depressed Kenneth Lor	Casey Afflec	2016	137	134213	47.7	7.9	96	72.5892		
23	Hounds of L	Crime,Dram	A cold-bloo Ben Young	Emma Bootl	2016	108	1115		6.7	72	58.6476	
24	Trolls	Animation,A	After the Be Walt Dohrn	Anna Kendri	2016	92	38552	153.69	6.5	56	56.324	

Then, we remove the line with empty value of MetaScore because in this case, we require the data of MetaScore. After remove, we get 936 valid record.

Then, we user the Excel internal function to draw the scatter diagram of the data and draw the line of **linear regression**



The regression function is  $y = 11.618x - 19.193$ , where  $x$  is rating and  $y$  is metascore.

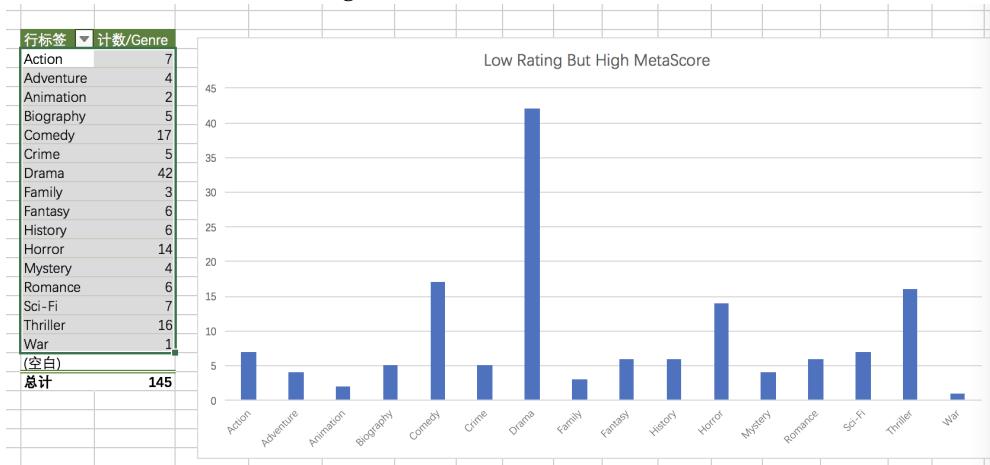
We calculate the absolute value between actual value and linear regression and count the number of record.

num_diff_abs > 10	420
num_diff_abs > 15	240
num_diff_abs > 20	118
...	

We assume that the top 5% and lowest 5% as the outlier.

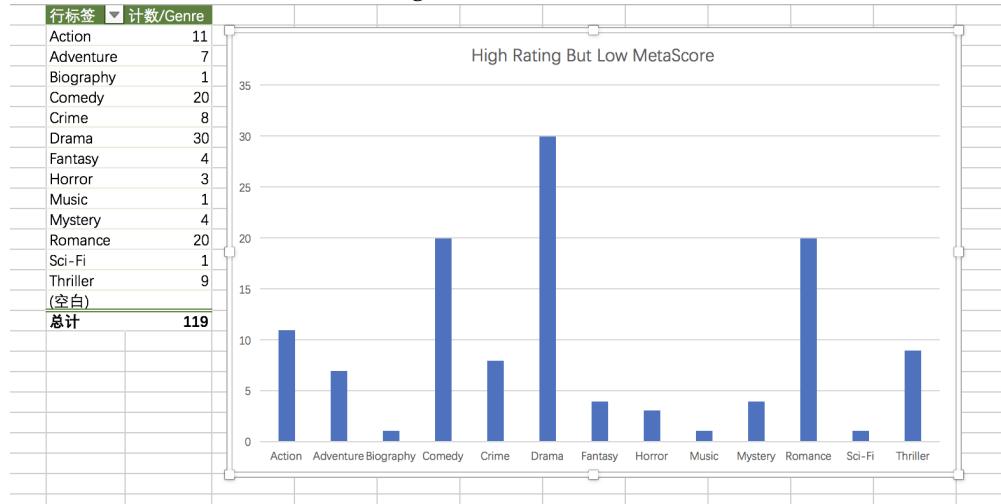
So, we need to get about 10% of data: 20 is a suitable different.

We first analysis the top 5% (different >20) of the data, which mean "Low Rating but Has High MetaScore". Then we can get:



We can clear know that Drama is the most common Genre in this case.

Then, we analysis the lowest 5% (different <-20) of the data, which mean "High Rating but Has Low MetaScore". Then we can get:



In this case, we also find that Drama is the most common Genre.

### 3.4.2 Summary

By analysis the result, we can conclude that Drama Movie has the most controversial comment. Since it has the most gap between Rating and MetaScore. It means the evaluation of Drama Movie is various among people.

For Drama movie, we should carefully consider the Rating and MetaScore, because it has the most possibility that is a stray value.

## 3.5 Keywords Analyzing

This analysis provided by Jiaqi (Garfield) Wu. By using these data, we can create a word cloud based on the descriptions and to find out the most frequent appeared keyword, and find out why the resulting words are appeared so frequently.

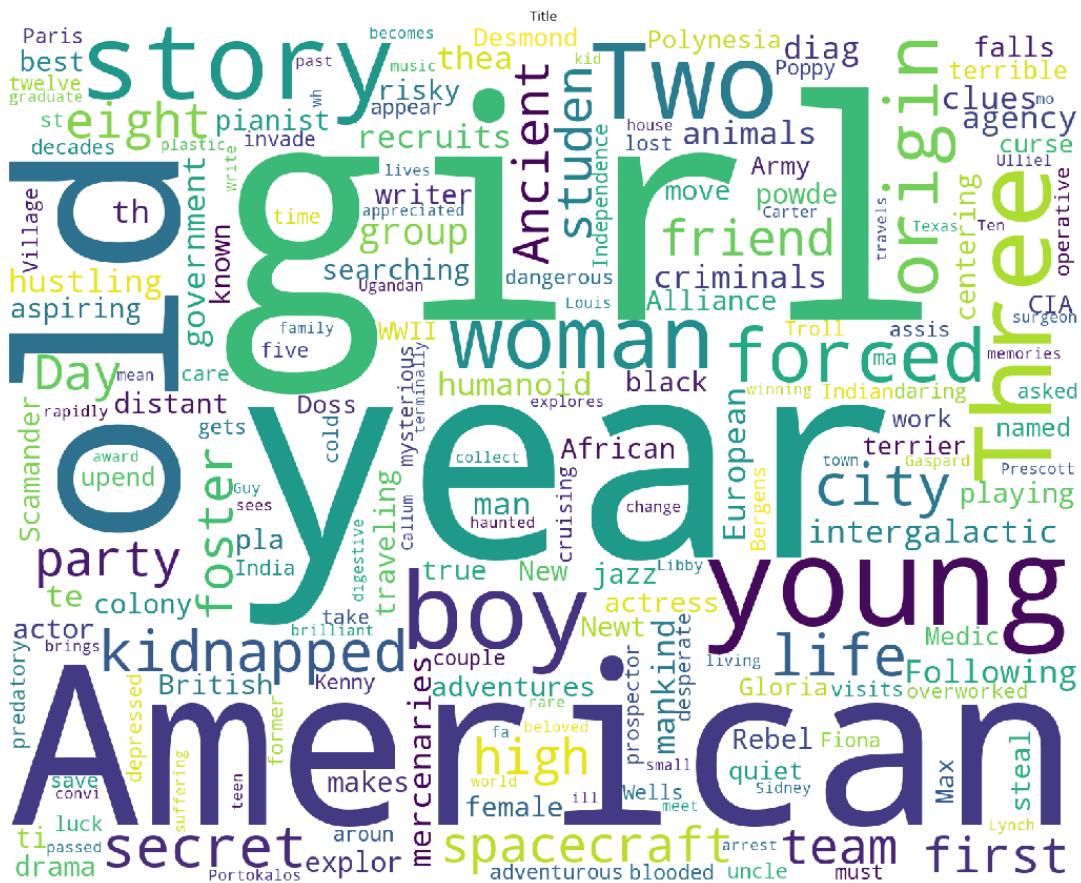
### 3.5.1 Implementation

#### Get the keywords

```
In [4]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
%matplotlib inline
        import seaborn as sns
        from os import path
        import matplotlib.pyplot as plt
        import random
        from wordcloud import WordCloud, STOPWORDS
imdbdata=pd.read_csv('../Dataset/IMDB-Movie-Data.csv')
text = (str(imdbdata['Description']))
plt.subplots(figsize=(20,15))
wordcloud = WordCloud(
```

```
stopwords=STOPWORDS,  
background_color='white',  
width=1500,  
height=1200  
).generate(text)
```

```
plt.imshow(wordcloud)
plt.title('Title')
plt.axis('off')
plt.show()
```



### 3.5.2 Summary

From the graph, we can get these following keywords:

- Year
  - American

- Old
- Young
- Girl

Since most of the 1000 movies are American made, it is not surprising that American will be one of the biggest word in the word cloud. It is interesting that the word "girl" is much bigger than "boy". It shows that people are more interested in female rather than male. As for the word "year", the reason of why it is so big is that when describing a story, the timeline is always important. So the word "year" appeared frequently in the description of movies.

**\*\*\* This is the end of the report. \*\*\***

L<sup>A</sup>T<sub>E</sub>X generated by Jupyter @ Anaconda 5.0.1 @ Python 3.6.3, refined by Junru (Bill) Zhong

Last modification: 23:00, December 19, 2017