Project: Investigate a Dataset (The Movies Database)

Table of Contents

Introduction

> This dataset contains information about 10,000 movies collected from The Movie Database, including user ratings and revenue. In this report, I will be exploring the following questions: (1) which year has the highest patronage of movies. (2) what kinds of properties are associated with movies that have high revenues. (3) which genres are most popular from year to year. (4) do customers have preference for production from a particular company. (5) what factors or variables significantly impact the popularity of the movies. (6) which year did the film industry make the highest profit

Data Set Up

In [1]:
```
import pandas as pd
import numpy as np
% matplotlib inline
```

Data Wrangling

General Properties

> The data is loaded and few lines of the dataset are printed out. By inspecting the dataset, duplicates, missing rows, and datatypes of the features are determined. It is evident that some rows are missing in "popularity", "cast", "homepage", "director", "keywords", "overview", "genres", and "production_companies" columns. The nature of the datatypes of each feature is displayed below.

In [2]:

```
df_movie = pd.read_csv('tmdb_movies.csv')

## Printout of raw data
df_movie.head(-4)
```

Out[2]:

	id	imdb_id	popularity	budget	revenue	original_title	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Br Dallas Howar Khan|Vi...
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|C Theron|Hugh Byrne|Nic...
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|The James|Kate Winslet|Anse
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Forc Hamill|Carrie Fisher|Adam
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Pa Walker|Jason Statham|Mich
5	281957	tt1663202	9.110700	135000000	532950503	The Revenant	Leonardo DiCaprio|Tom Hardy|Will Poulter|Domh
6	87101	tt1340138	8.654359	155000000	440603537	Terminator Genisys	Arnold Schwarzeneg Clarke|Emilia
7	286217	tt3659388	7.667400	108000000	595380321	The Martian	Matt Damon|J Chastain|Kris Wiig|Jeff ...
8	211672	tt2293640	7.404165	74000000	1156730962	Minions	Sandra Bulloc Hamm|Micha Keaton|Alliso

	id	imdb_id	popularity	budget	revenue	original_title	
9	150540	tt2096673	6.326804	175000000	853708609	Inside Out	Amy Poehler|Smith|Richard Ha...
10	206647	tt2379713	6.200282	245000000	880674609	Spectre	Daniel Craig|Waltz|LÃ©a Seydoux|Ral
11	76757	tt1617661	6.189369	176000003	183987723	Jupiter Ascending	Mila Kunis|Ch Tatum|Sean Bean|Eddie R
12	264660	tt0470752	6.118847	15000000	36869414	Ex Machina	Domhnall Gleeson|Alicia Vikander|Osc Isaac|S...
13	257344	tt2120120	5.984995	88000000	243637091	Pixels	Adam Sandle Monaghan|Pe Dinklage|...
14	99861	tt2395427	5.944927	280000000	1405035767	Avengers: Age of Ultron	Robert Down Jr.|Chris Hemsworth|M Ruffalo...
15	273248	tt3460252	5.898400	44000000	155760117	The Hateful Eight	Samuel L. Jackson|Kurt Russell|Jenni ...
16	260346	tt2446042	5.749758	48000000	325771424	Taken 3	Liam Neeson Whitaker|Mag Grace|Famke
17	102899	tt0478970	5.573184	130000000	518602163	Ant-Man	Paul Rudd|Mi Douglas|Evan Lilly|Cor...

	id	imdb_id	popularity	budget	revenue	original_title	
18	150689	tt1661199	5.556818	95000000	542351353	Cinderella	Lily James|Ca Blanchett|Ric Madden|Hele
19	131634	tt1951266	5.476958	160000000	650523427	The Hunger Games: Mockingjay - Part 2	Jennifer Lawrence|Jos Hutcherson|L Hemswor...
20	158852	tt1964418	5.462138	190000000	209035668	Tomorrowland	Britt Robertsc Clooney|Raffe Cassidy|...
21	307081	tt1798684	5.337064	30000000	91709827	Southpaw	Jake Gyllenha McAdams|Fo Whitaker...
22	254128	tt2126355	4.907832	110000000	470490832	San Andreas	Dwayne Johnson|Alex Daddario|Car Gugino...
23	216015	tt2322441	4.710402	40000000	569651467	Fifty Shades of Grey	Dakota Johns Dornan|Jenni Ehle|Eloi...
24	318846	tt1596363	4.648046	28000000	133346506	The Big Short	Christian Bale Carell|Ryan Gosling|Brad
25	177677	tt2381249	4.566713	150000000	682330139	Mission: Impossible - Rogue Nation	Tom Cruise|J Renner|Simo Pegg|Rebecc
26	214756	tt2637276	4.564549	68000000	215863606	Ted 2	Mark Wahlbe MacFarlane|A Seyfried|...

	id	imdb_id	popularity	budget	revenue	original_title	
27	207703	tt2802144	4.503789	81000000	403802136	Kingsman: The Secret Service	Taron Egertor Firth|Samuel Jackson|Mi...
28	314365	tt1895587	4.062293	20000000	88346473	Spotlight	Mark Ruffalo| Keaton|Rach McAdams|Lie
29	294254	tt4046784	3.968891	61000000	311256926	Maze Runner: The Scorch Trials	Dylan O'Brier Scodelario|Tl Brodie-Sa...
...	...	...	...	...	...	...	...
10832	23030	tt0060121	0.358161	4800000	0	Arabesque	Gregory Peck Loren|Alan Badel|Kieron
10833	3001	tt0060522	0.737730	0	0	How to Steal a Million	Audrey Hepb O'Toole|Eli Wallach|Hugh
10834	12639	tt0060897	0.310688	0	0	Return of the Seven	Yul Brynner|F Fuller|JuliÃ¡n Mateos|Warre
10835	5923	tt0060934	0.299911	12000000	20000000	The Sand Pebbles	Steve McQueen|Ric Attenborough Cre...
10836	38720	tt0061170	0.239435	0	0	Walk Don't Run	Cary Grant|S Eggar|Jim Hu Stan...
10837	19728	tt0060177	0.291704	0	0	The Blue Max	George Peppard|Jam Mason|Ursula Andress|Jere

	id	imdb_id	popularity	budget	revenue	original_title	
10838	22383	tt0060862	0.151845	0	0	The Professionals	Burt Lancaste Marvin|Rober Ryan|Woody
10839	13353	tt0060550	0.276133	0	0	It's the Great Pumpkin, Charlie Brown	Christopher S Dryer|Kathy Steinberg|A...
10840	34388	tt0060437	0.102530	0	0	Funeral in Berlin	Michael Cain Hubschmid|C Homolka|Eva
10841	42701	tt0062262	0.264925	75000	0	The Shooting	Will Hutchins| Perkins|Jack Nicholson|Wa
10842	36540	tt0061199	0.253437	0	0	Winnie the Pooh and the Honey Tree	Sterling Holloway|Jun Matthews|Sel Ca...
10843	29710	tt0060588	0.252399	0	0	Khartoum	Charlton Heston|Laure Olivier|Richar
10844	23728	tt0059557	0.236098	0	0	Our Man Flint	James Cobur Cobb|Gila Golan|Edwar
10845	5065	tt0059014	0.230873	0	0	Carry On Cowboy	Sid James|Jir Dale|Angela Douglas|Kenr
10846	17102	tt0059127	0.212716	0	0	Dracula: Prince of Darkness	Christopher Lee|Barbara Shelley|Andre Keir|Fr...

	id	imdb_id	popularity	budget	revenue	original_title	
10847	28763	tt0060548	0.034555	0	0	Island of Terror	Peter Cushin Judd|Carole Gray|Eddie B
10848	2161	tt0060397	0.207257	5115000	12000000	Fantastic Voyage	Stephen Boy Welch|Edmor O'Brien|Dona
10849	28270	tt0060445	0.206537	0	0	Gambit	Michael Cain MacLaine|He Lom|Joh...
10850	26268	tt0060490	0.202473	0	0	Harper	Paul Newmar Bacall|Julie Harris|Arthur
10851	15347	tt0060182	0.342791	0	0	Born Free	Virginia McK Travers|Geof Keen|Pe...
10852	37301	tt0060165	0.227220	0	0	A Big Hand for the Little Lady	Henry Fonda| Woodward|Ja Robards|Pau
10853	15598	tt0060086	0.163592	0	0	Alfie	Michael Cain Winters|Millic Martin...
10854	31602	tt0060232	0.146402	0	0	The Chase	Marlon Brand Fonda|Rober Redford|E.G.
10855	13343	tt0059221	0.141026	700000	0	The Ghost & Mr. Chicken	Don Knotts|J Staley|Liam Redmond|Di

	id	imdb_id	popularity	budget	revenue	original_title	
10856	20277	tt0061135	0.140934	0	0	The Ugly Dachshund	Dean Jones|S Pleshette|Cha Ruggles|K...
10857	5921	tt0060748	0.131378	0	0	Nevada Smith	Steve McQue Malden|Brian Keith|Arthur K
10858	31918	tt0060921	0.317824	0	0	The Russians Are Coming, The Russians Are Coming	Carl Reiner|E Saint|Alan Ar K...
10859	20620	tt0060955	0.089072	0	0	Seconds	Rock Hudson Jens|John Randolph|Wil
10860	5060	tt0060214	0.087034	0	0	Carry On Screaming!	Kenneth Willi Dale|Harry H. Corbett|Joa...
10861	21	tt0060371	0.080598	0	0	The Endless Summer	Michael Hyns August|Lord ' B...

10862 rows × 21 columns

In [3]:
```
## The goal is to check the datatypes of features in the dataset
df_movie.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
id                      10866 non-null int64
imdb_id                 10856 non-null object
popularity              10866 non-null float64
budget                  10866 non-null int64
revenue                 10866 non-null int64
original_title          10866 non-null object
cast                    10790 non-null object
homepage                2936 non-null object
director                10822 non-null object
tagline                 8042 non-null object
keywords                9373 non-null object
overview                10862 non-null object
runtime                 10866 non-null int64
genres                  10843 non-null object
production_companies    9836 non-null object
release_date            10866 non-null object
vote_count              10866 non-null int64
vote_average            10866 non-null float64
release_year            10866 non-null int64
budget_adj              10866 non-null float64
revenue_adj             10866 non-null float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

Data Cleaning

> Looking at the dataset, some rows need to be dropped. Duplicates will be checked and cleaned. No column will be dropped in this analysis, as there is no cause for it; all the columns are relevant. Rows with null values need to be dropped so as to have compact dataset with non-missing values in the columns only; since almost all the columns of the dataset are relevant for this analysis. Duplicates may be due to human error, so duplicate data will negatively impact the data analysis.

In [4]:
```
#drop rows with any null values in the dataset

df_movie.dropna(how='any',inplace=True)
```

In [5]:
```
# check if any columns have null values; it should print false

df_movie.isnull().sum().any()
```

Out[5]: False

In [6]:
```
# check for duplicates in the dataset. If none, it should print out 0

sum(df_movie.duplicated())
```

Out[6]: 0

In [7]:
```
# The structure of the dataset after it has been cleaned
df_movie.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1992 entries, 0 to 10819
Data columns (total 21 columns):
id                      1992 non-null int64
imdb_id                 1992 non-null object
popularity              1992 non-null float64
budget                  1992 non-null int64
revenue                 1992 non-null int64
original_title          1992 non-null object
cast                    1992 non-null object
homepage                1992 non-null object
director                1992 non-null object
tagline                 1992 non-null object
keywords                1992 non-null object
overview                1992 non-null object
runtime                 1992 non-null int64
genres                  1992 non-null object
production_companies    1992 non-null object
release_date            1992 non-null object
vote_count              1992 non-null int64
vote_average            1992 non-null float64
release_year            1992 non-null int64
budget_adj              1992 non-null float64
revenue_adj             1992 non-null float64
dtypes: float64(4), int64(6), object(11)
memory usage: 342.4+ KB
```

Exploratory Data Analysis

Research Question 1: Which year has the highest patronage of movies?

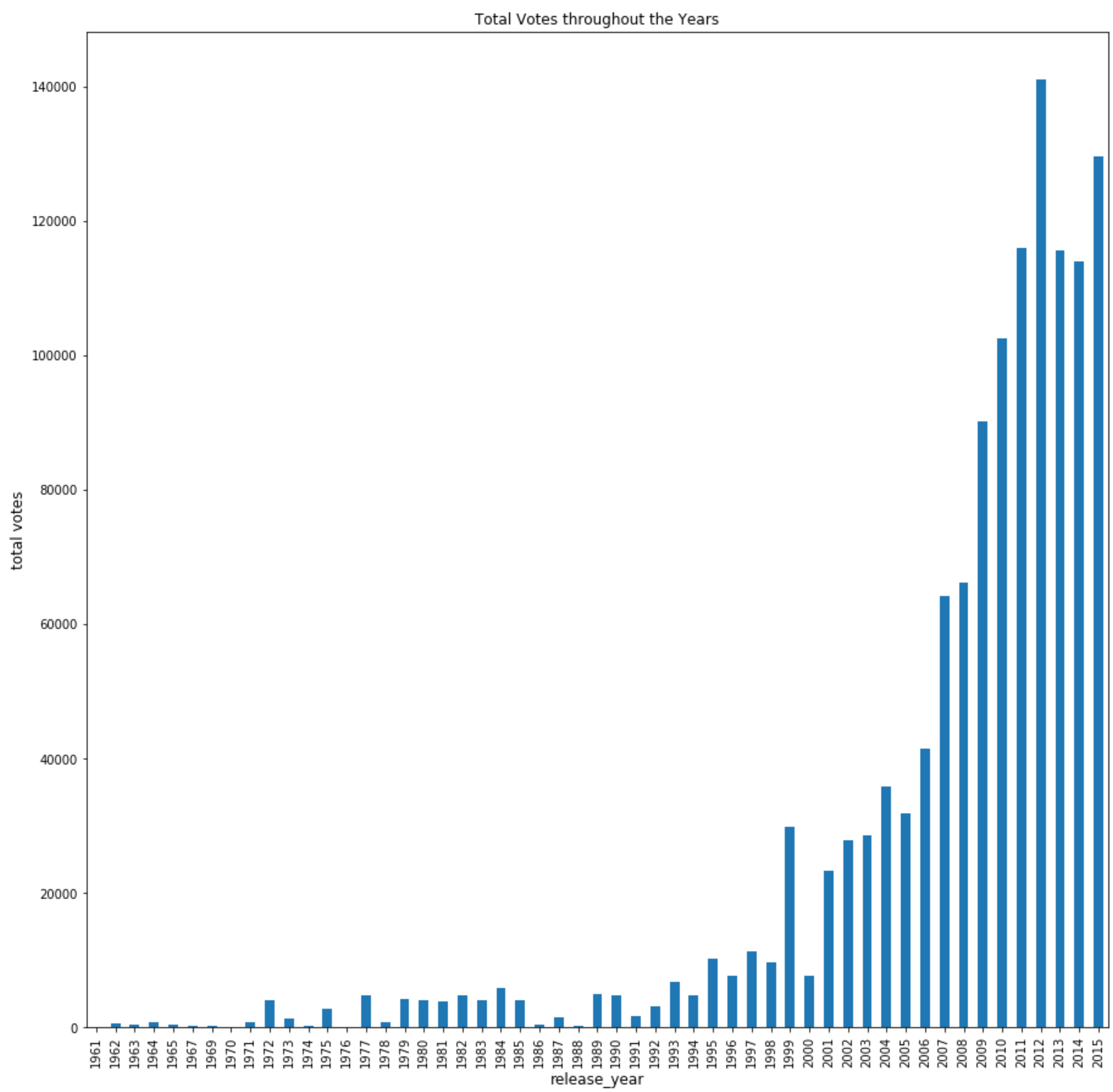
Discussion

> The number of vote count will be used as a measure of patronage in this analysis. From the bar chart plot below it is evident that in 2012 the movie industry has the highest patronage of customers; the highest record in history.

In [8]:
```
import matplotlib.pyplot as plt
% matplotlib inline
```

In [9]:
```
## Plot visualizzation of the relationship between release year and number of
 votes for each genre
df_movie.groupby('release_year')['vote_count'].sum().plot(kind='bar',figsize=(
15,15),title='Total Votes throughout the Years')
plt.xlabel('release_year',fontsize=12)
plt.ylabel('total votes',fontsize=12)
```

Out[9]: Text(0,0.5,'total votes')



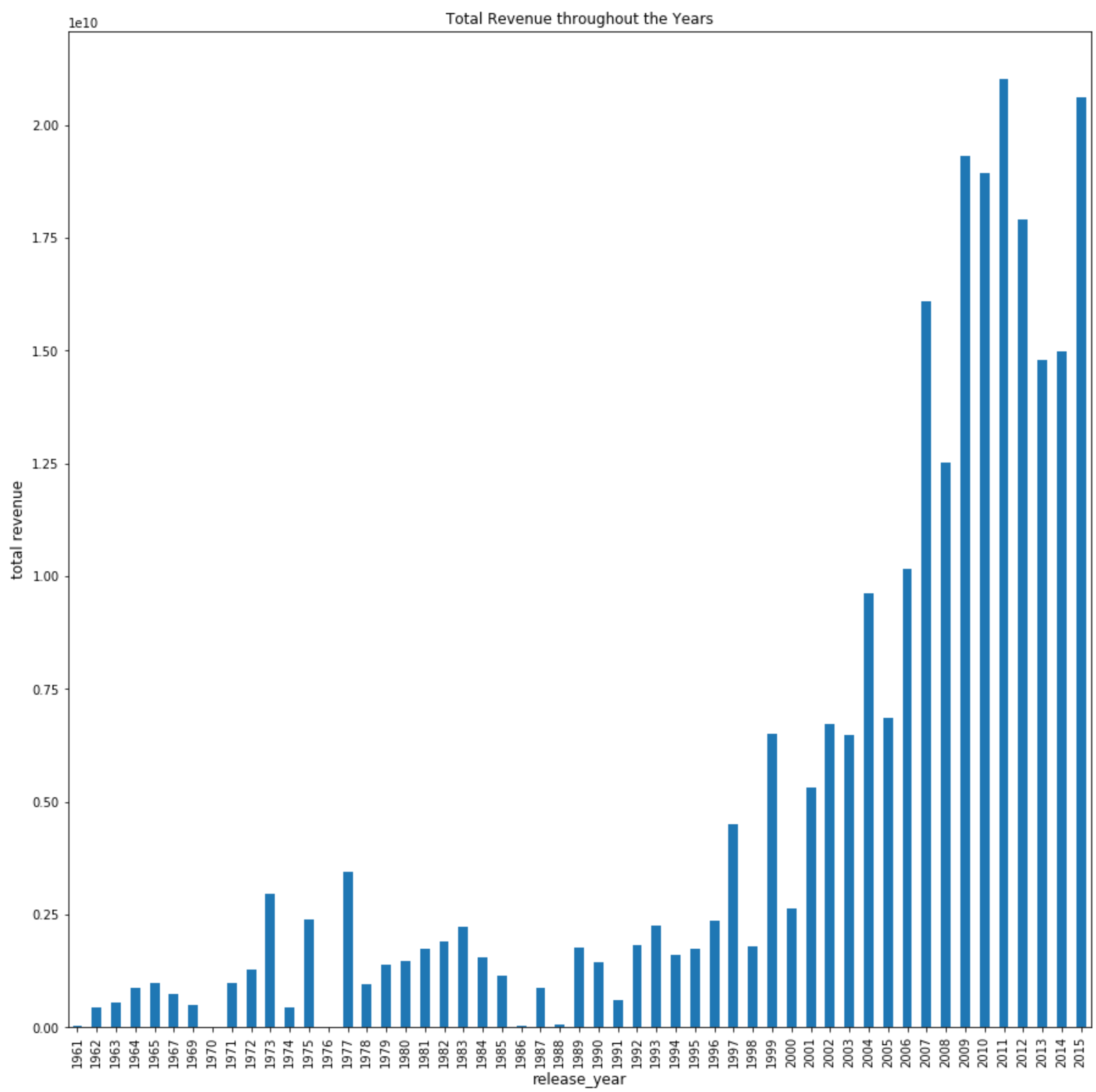
Research Question 2: Which year did the film industry make the highest profit?

Discussion:

> To answer this question, I have plotted the adjusted revenue and profit index for each year. The profit index is the major indicator that determines how profitable the industry is. The index is a measure of the amount made on investment. From the Figure on Profit Index, the film industry made the highest profit in 1978 although the highest revenue was made in 2011. This figure indicates that the financial attractiveness of making a movie is reducing with time; although it has remained relatively stable at 5 for the last decade of this survey.

In [10]:
```
#Plot visualization of adjusted revenue per year
df_movie.groupby('release_year')['revenue_adj'].sum().plot(kind='bar',figsize=
(15,15),title='Total Revenue throughout the Years')
plt.xlabel('release_year',fontsize=12)
plt.ylabel('total revenue',fontsize=12)
```

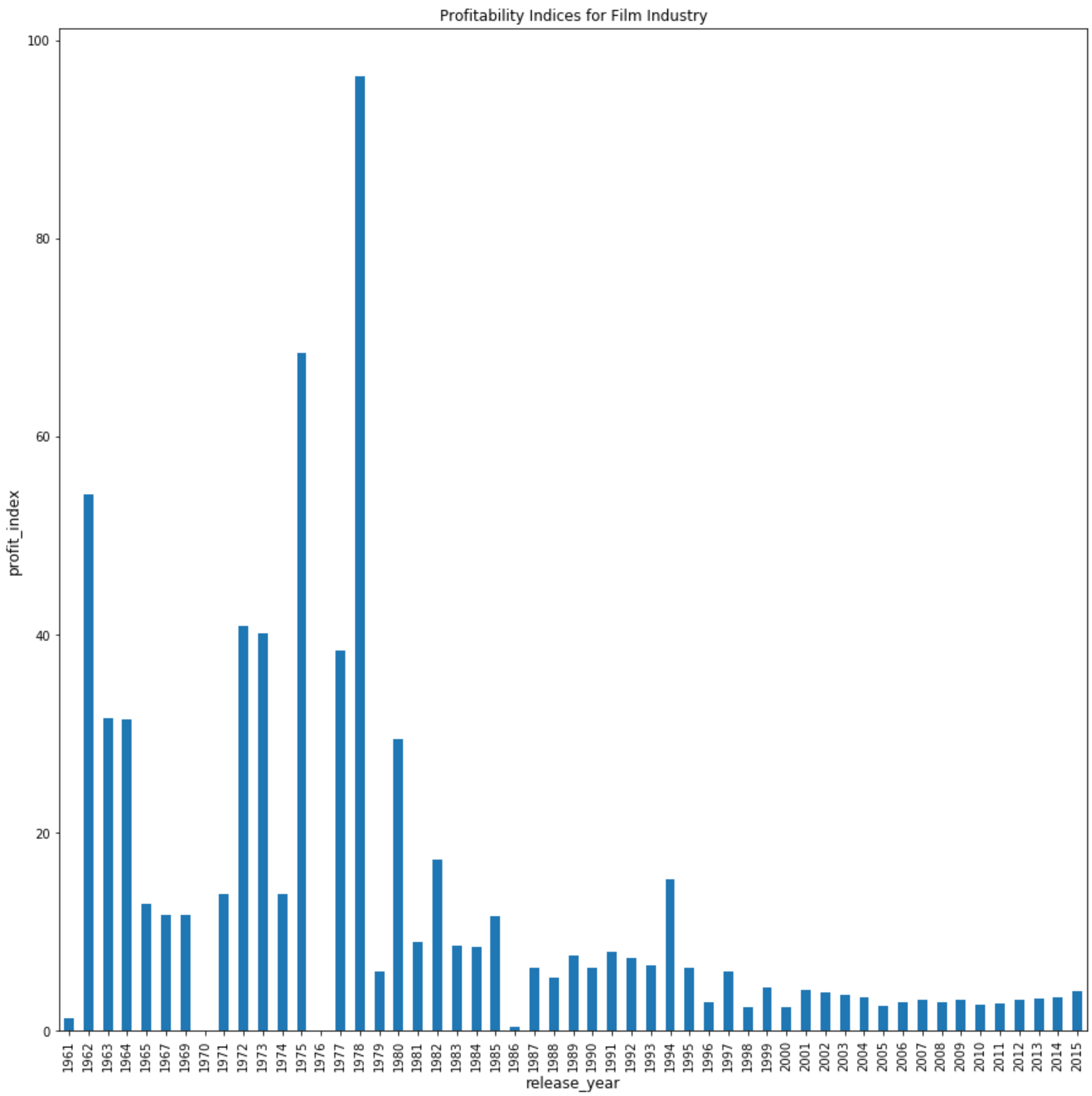
Out[10]: `Text(0,0.5,'total revenue')`



In [11]:

```
#Plot visualization of profitability index per year
revenue = df_movie.groupby('release_year')['revenue_adj'].sum()
expense = df_movie.groupby('release_year')['budget_adj'].sum()
profit_index = (revenue)/expense
profit_index.plot(kind='bar',figsize=(15,15),title='Profitability Indices for
 Film Industry')
plt.xlabel('release_year',fontsize=12)
plt.ylabel('profit_index',fontsize=12)
```

Out[11]: `Text(0,0.5,'profit_index')`



Research Question 3: Which genres are most popular from year to year?

Discussion

> To analyse this problem, three different indicators of the genres popularity are considered:(1) revenue (2) profitability index (3) popularity index. The revenue indicator is used because the popularity of the genre means many customers patronised it. From the plot below Adventure, Animation, Science Fiction, and Western movies have received a lot of popularity over the years.
>
> The profitability index, which is a derivative of the revenue shows a different pattern of popularity measure among the genres. Adventure, Documentary, Horror, and Science Fiction are the most popular.
>
> Now following the popularity index, Action, Adventure, Science Fiction, and Western are the most popular year to year. Thus, it implies that the use revenue and profitability index are not good indicators to measure how popular the genres are.

In [12]:
```
## First, getting all the movies with more than one genres

genre_more = df_movie[df_movie['genres'].str.contains('|')]
```

In [13]:
```
# making a copy

df_movie1 = genre_more.copy()
```

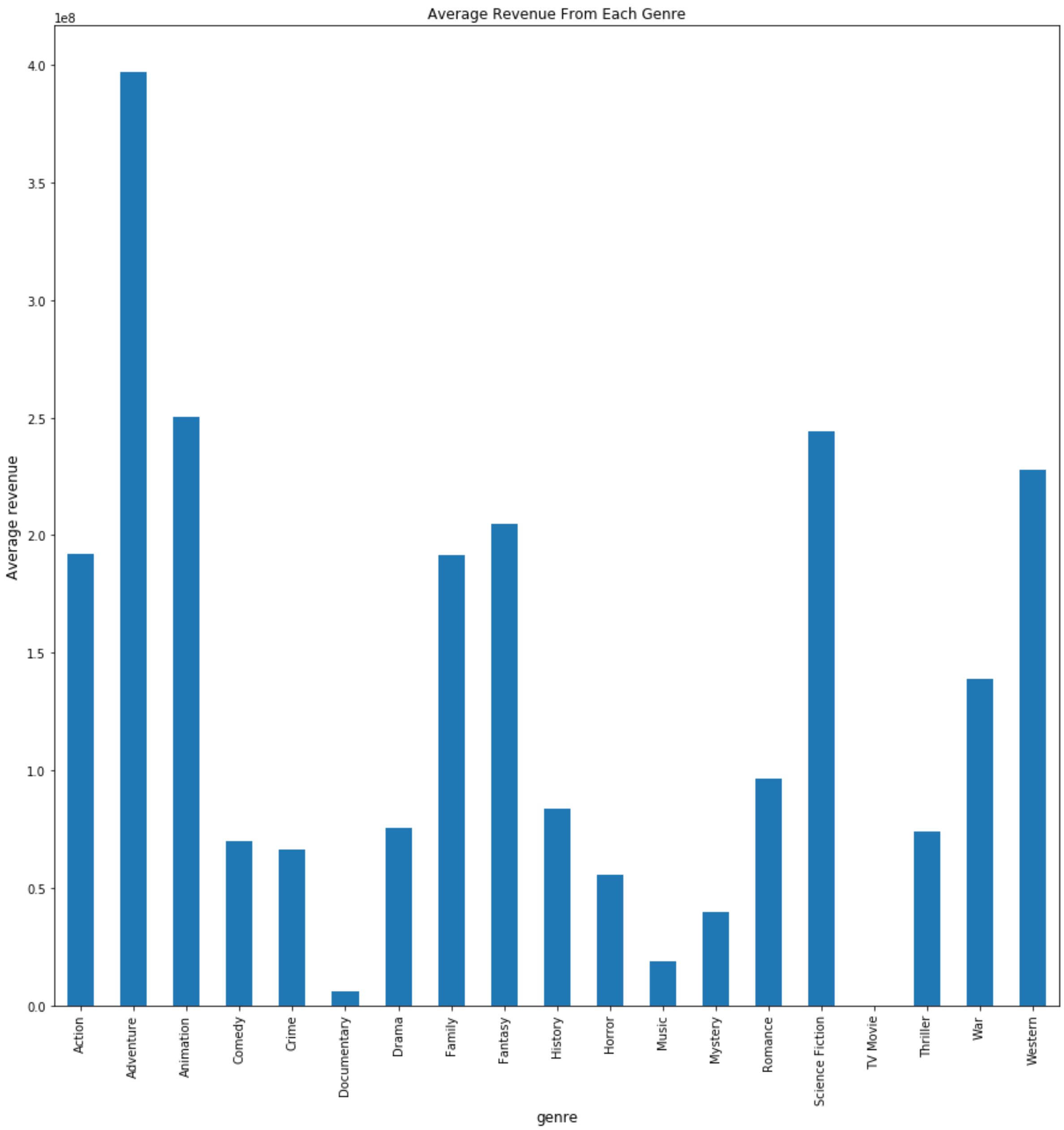
In [14]:
```
# Using Pandas' apply function to split the column
split_columns = ['genres','cast','production_companies']

for c in split_columns:
    df_movie1[c] = df_movie1[c].apply(lambda x: x.split("|")[0])
```

In [15]:
```
# A view of the cleaned dataset
df_movie1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1992 entries, 0 to 10819
Data columns (total 21 columns):
id                      1992 non-null int64
imdb_id                 1992 non-null object
popularity              1992 non-null float64
budget                  1992 non-null int64
revenue                 1992 non-null int64
original_title          1992 non-null object
cast                    1992 non-null object
homepage                1992 non-null object
director                1992 non-null object
tagline                 1992 non-null object
keywords                1992 non-null object
overview                1992 non-null object
runtime                 1992 non-null int64
genres                  1992 non-null object
production_companies    1992 non-null object
release_date            1992 non-null object
vote_count              1992 non-null int64
vote_average            1992 non-null float64
release_year            1992 non-null int64
budget_adj              1992 non-null float64
revenue_adj             1992 non-null float64
dtypes: float64(4), int64(6), object(11)
memory usage: 342.4+ KB
```

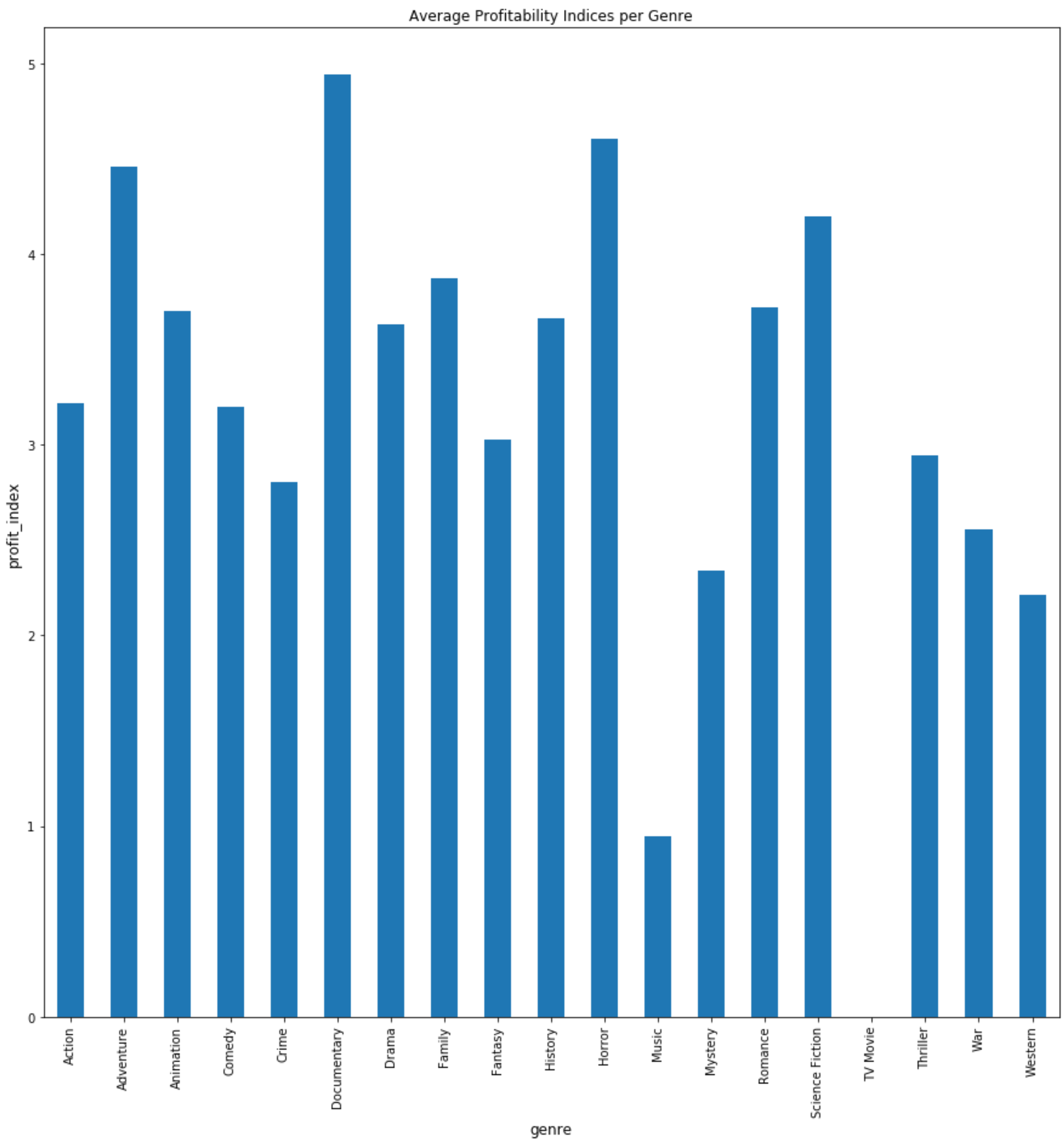
In [16]:
```
# Considering total revenue generated per genre from year to year
df_movie1.groupby('genres')['revenue_adj'].mean().plot(kind='bar',figsize=(15,
15),title='Average Revenue From Each Genre')
plt.xlabel('genre',fontsize=12)
plt.ylabel('Average revenue',fontsize=12)
```

Out[16]: `Text(0,0.5,'Average revenue')`



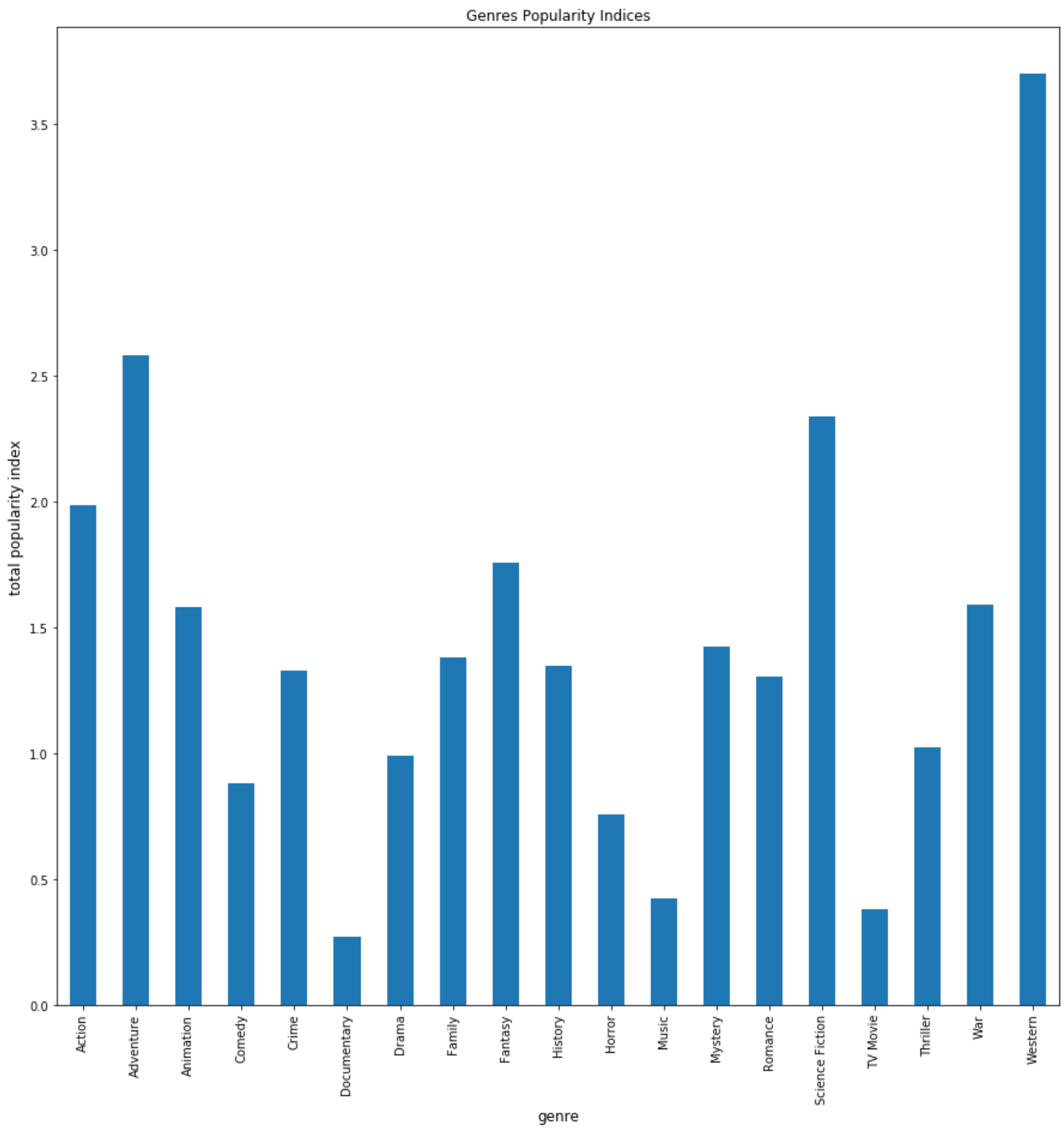
In [17]:
```
#Considering profitability index per genre

revenue_genre = df_movie1.groupby('genres')['revenue_adj'].mean()
expense_genre = df_movie1.groupby('genres')['budget_adj'].mean()
profit_index_genre = (revenue_genre)/expense_genre
profit_index_genre.plot(kind='bar',figsize=(15,15),title='Average Profitabilit
y Indices per Genre')
plt.xlabel('genre',fontsize=12)
plt.ylabel('profit_index',fontsize=12)
```

Out[17]:
```
Text(0,0.5,'profit_index')
```



In [18]:
```
# Using popularity index
df_movie1.groupby('genres')['popularity'].mean().plot(kind='bar',figsize=(15,1
5),title='Genres Popularity Indices')
plt.xlabel('genre',fontsize=12)
plt.ylabel('total popularity index',fontsize=12)
```

Out[18]:
```
Text(0,0.5,'total popularity index')
```



Research Question 4: Which of the movies are the most expensive to make?

Discussion

> To answer this question, the budget per genre for each year are plotted. From this plot, the Western movies are the most expensive to make, follow by Adventure, Animation and Fantasy.
>
> Digging deeper, to know what could have caused these movies to be expensive, I considered if the number of casts could be a major factor. Unfortunately, from the plot below, the number of casts seem not be a major factor, as the most expensive genre to make has the least number of casts on the average. It thus imply that other factors may be responsible for the high cost of making these movies; these may be further investigated.

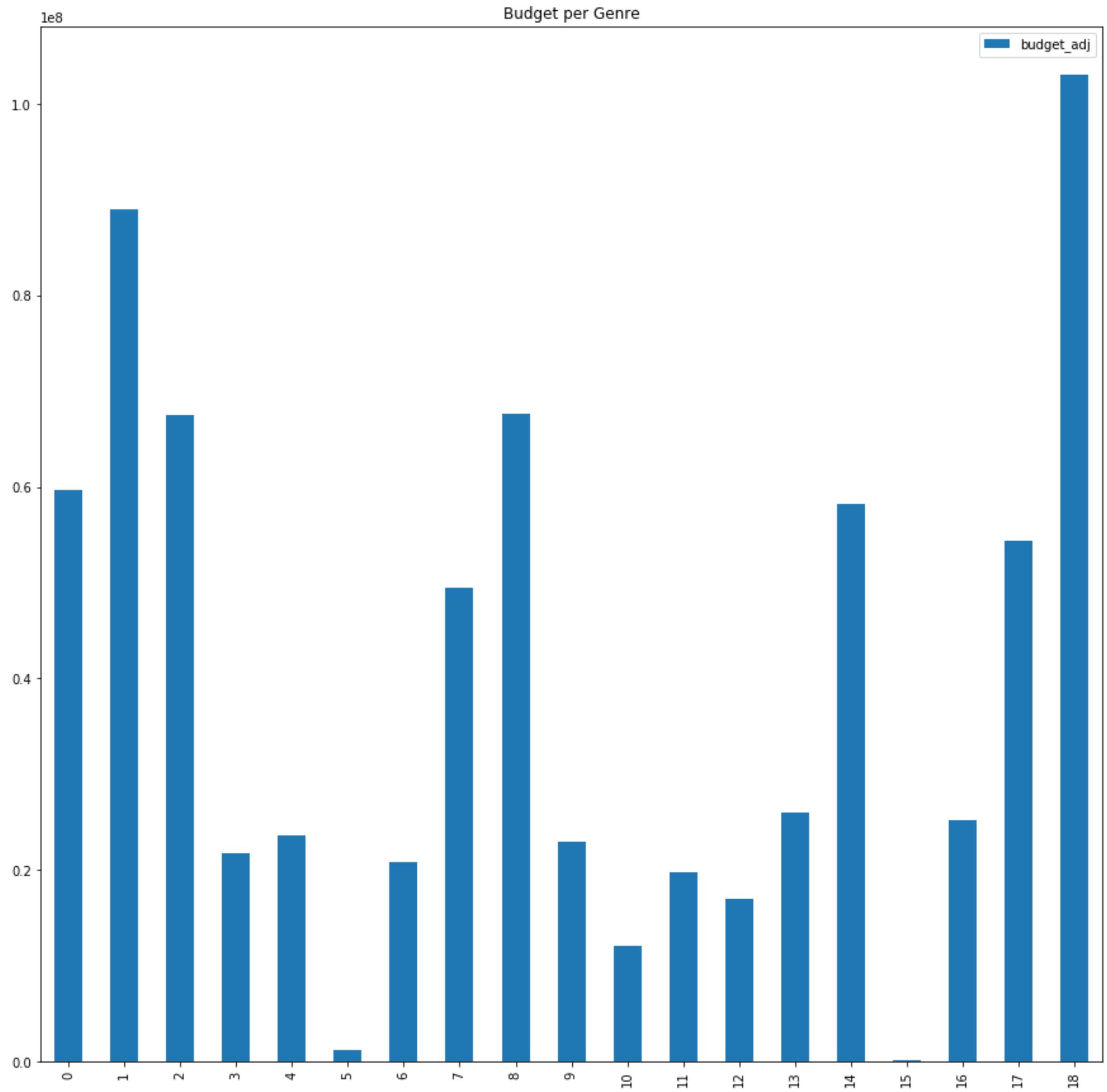
In [19]:
```
## Find the average cost to produce

df_movie_avg = df_movie1.groupby(['genres'],as_index=False)['budget_adj'].mean
()
df_movie_avg
```

Out[19]:

	genres	budget_adj
0	Action	5.968364e+07
1	Adventure	8.899322e+07
2	Animation	6.756489e+07
3	Comedy	2.178743e+07
4	Crime	2.363721e+07
5	Documentary	1.187675e+06
6	Drama	2.076619e+07
7	Family	4.947273e+07
8	Fantasy	6.767308e+07
9	History	2.290955e+07
10	Horror	1.211169e+07
11	Music	1.979859e+07
12	Mystery	1.701784e+07
13	Romance	2.597669e+07
14	Science Fiction	5.821643e+07
15	TV Movie	1.560056e+05
16	Thriller	2.514665e+07
17	War	5.436486e+07
18	Western	1.030102e+08

In [20]:
```
df_movie_avg.plot(kind='bar',figsize=(15,15),title='Budget per Genre')
```

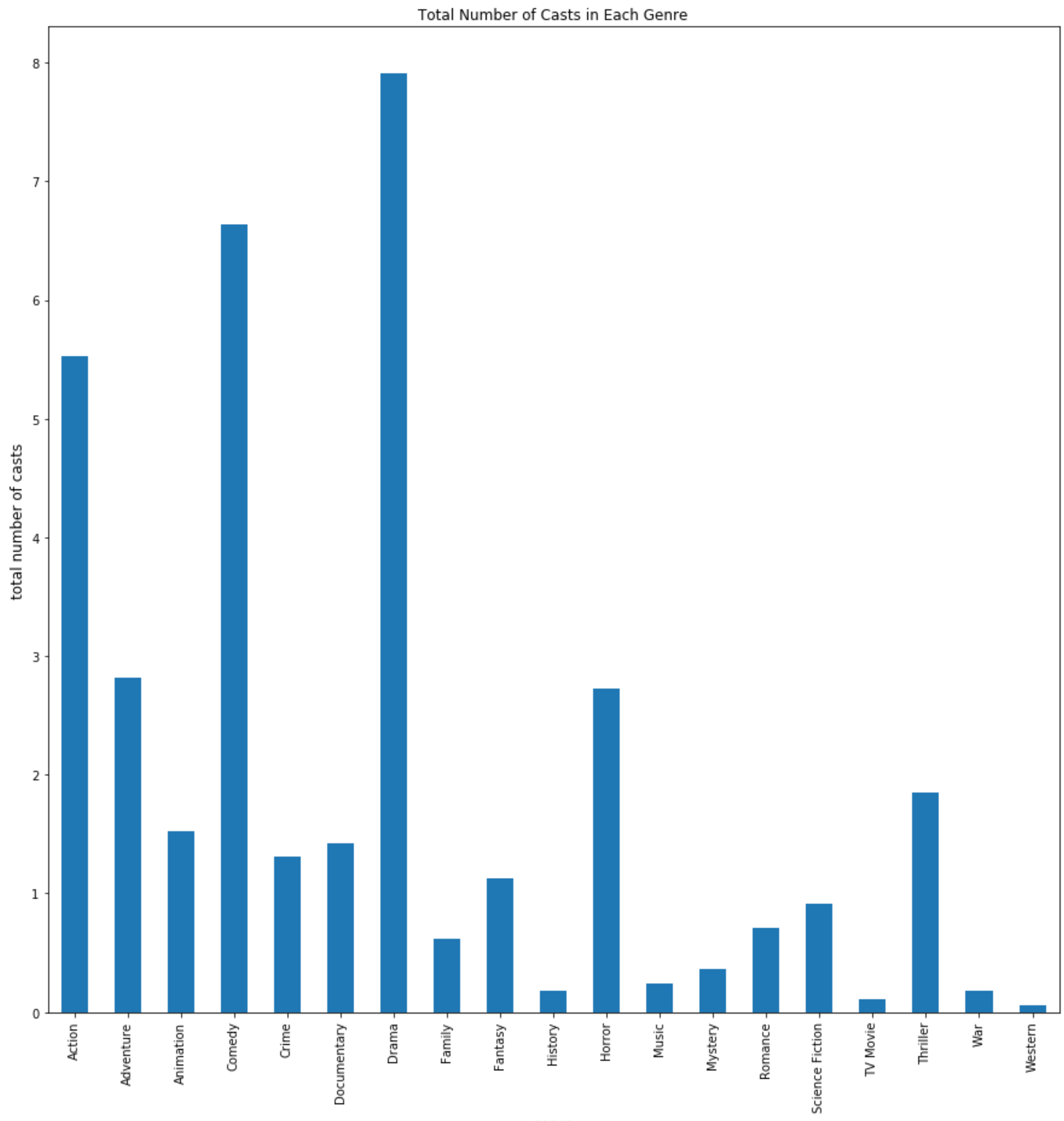
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3dfe09a9b0>



In [21]:

```
# Average cast per genre
df_movie2 = df_movie1.query('release_year == "2015"')
df_movie3=df_movie1.groupby('genres')['cast'].count()/55
df_movie3.plot(kind='bar',figsize=(15,15),title='Total Number of Casts in Each Genre')
plt.xlabel('genre',fontsize=12)
plt.ylabel('total number of casts',fontsize=12)
```

Out[21]: `Text(0,0.5,'total number of casts')`



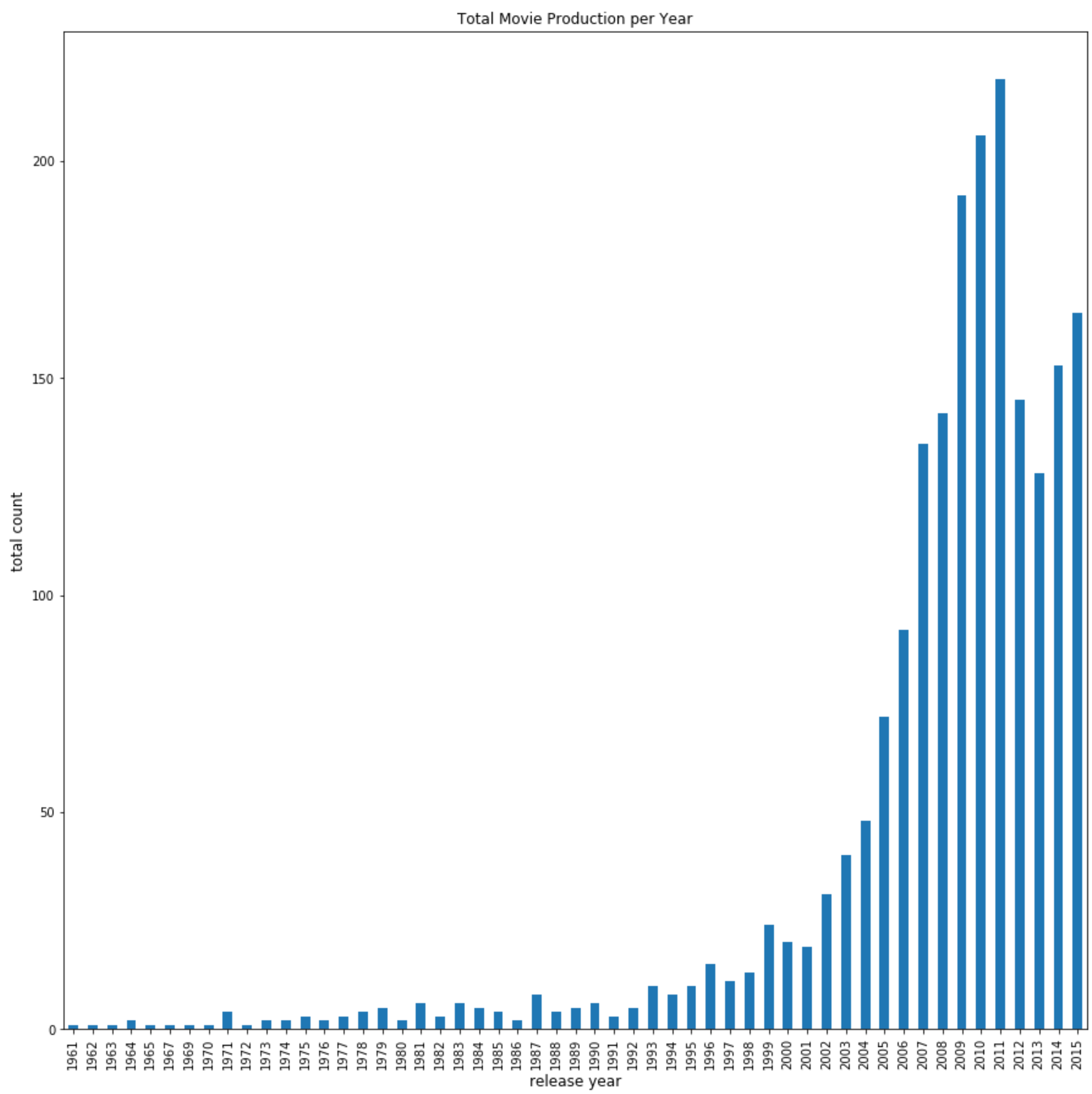
Research Question 5: Which year has the highest number of movies produced?

Discussion

> From the figure below the highest number of movies were produced in 2011, followed by 2010 and 2009 consecutively. Film production has increased exponentially from the early 1960's but with a decline in 2012. The cause for the decline cannot be ascertained from the available data accurately, but from the year versus budget, it is evident that the budget dropped. The drop in budget could have resulted in low investment, hence fewer movies were produced.

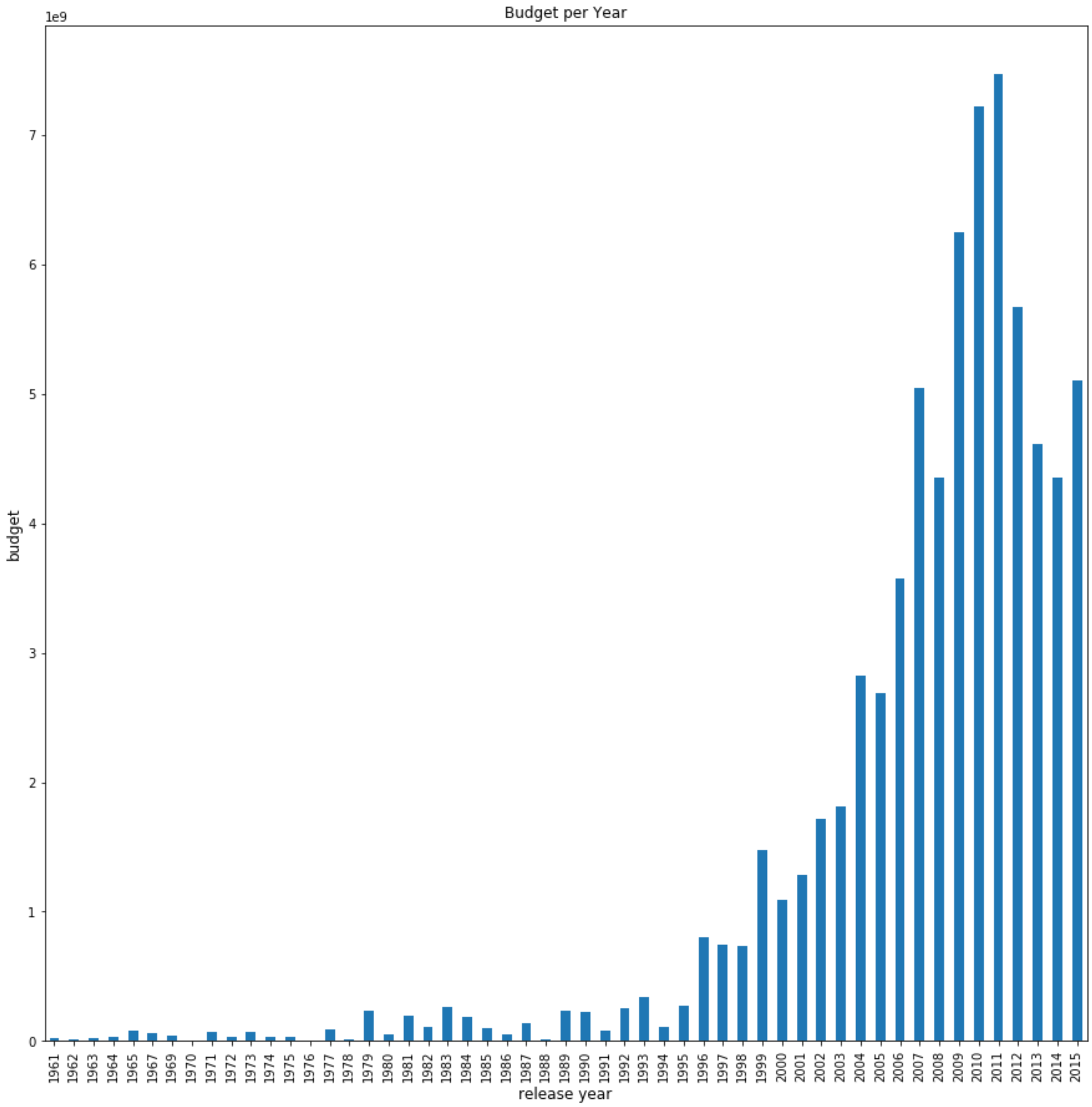
In [29]:
```
df_movie1.groupby('release_year')['genres'].count().plot(kind='bar',figsize=(1
5,15),title='Total Movie Production per Year')
plt.xlabel('release year',fontsize=12)
plt.ylabel('total count',fontsize=12)
```

Out[29]:
```
Text(0,0.5,'total count')
```



In [26]:
```
df_movie1.groupby('release_year')['budget_adj'].sum().plot(kind='bar',figsize=(15,15),title='Budget per Year')
plt.xlabel('release year',fontsize=12)
plt.ylabel('budget',fontsize=12)
```

Out[26]: `Text(0,0.5,'budget')`



Research Question 6: Which year of the three years with highest number of movies produced 2009, 2010, and 2011 has the highest number of movies with different genres?

Discussion:

> The essence of this question is to know the distribution of the genres in each of the years with highest movie production. This knowledge may be an indicator of which genre is most appreciated by the customers. In 2009 seventeen different genres were produced, making it the year in which the highest number of genres were produced; comedy being the highest that year. In each of the years, drama is the most produced on the average. With more features to the dataset, more information can be determined about the demographic, cultural orientation, and socio-economic status of these customers.

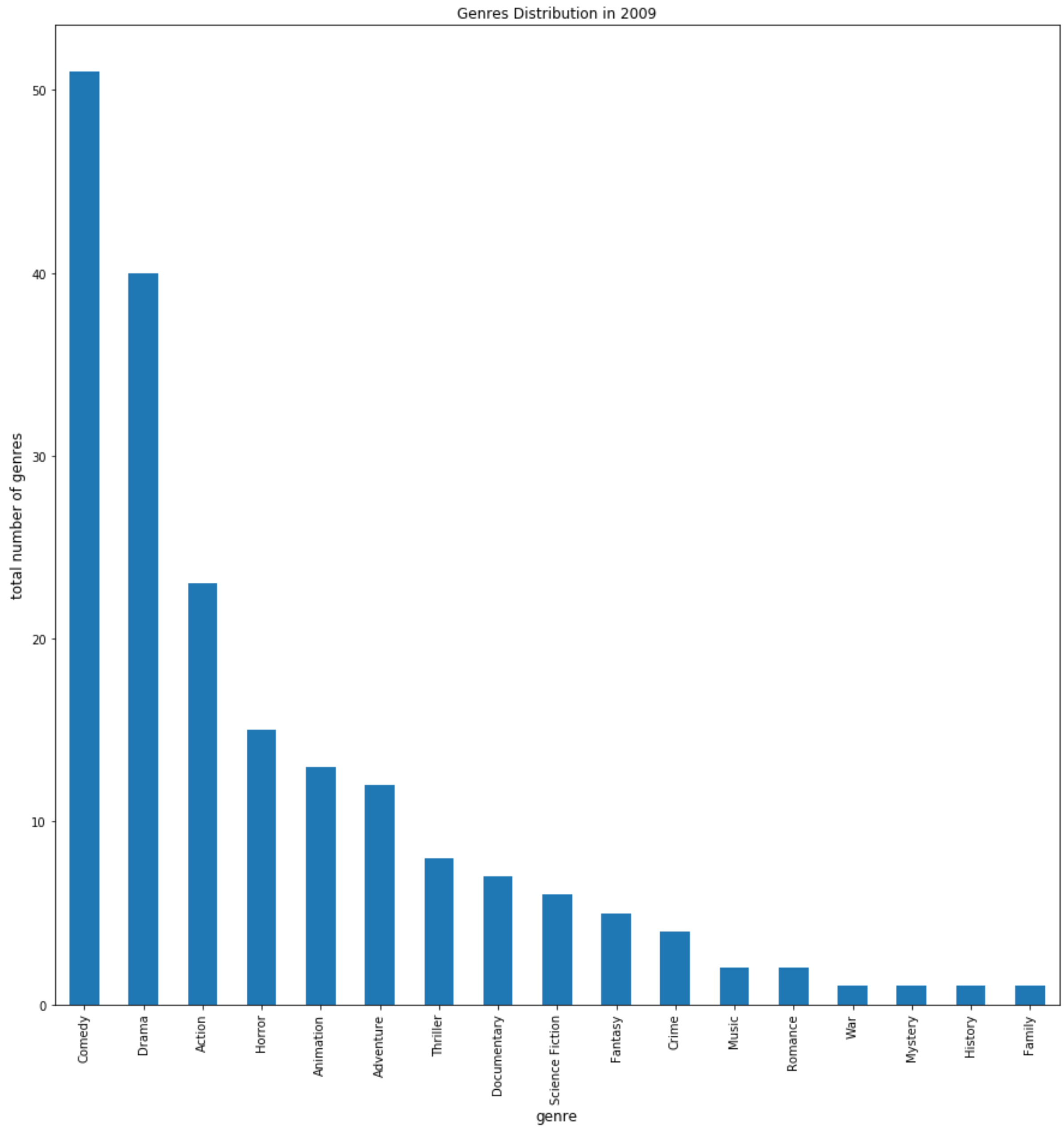
In [30]:
```
# Distribution of genres produced in 2009
df_movie2009a = df_movie1.query('release_year == "2009"')
df_movie2009b=df_movie2009a['genres'].value_counts()
```

In [31]:
```
df_movie2009b
```

Out[31]:
```
Comedy             51
Drama              40
Action             23
Horror             15
Animation          13
Adventure          12
Thriller            8
Documentary         7
Science Fiction     6
Fantasy             5
Crime               4
Music               2
Romance             2
War                 1
Mystery             1
History             1
Family              1
Name: genres, dtype: int64
```

In [32]:
```
df_movie2009b.plot(kind='bar',figsize=(15,15),title='Genres Distribution in 20
09')
plt.xlabel('genre',fontsize=12)
plt.ylabel('total number of genres',fontsize=12)
```

Out[32]: Text(0,0.5,'total number of genres')



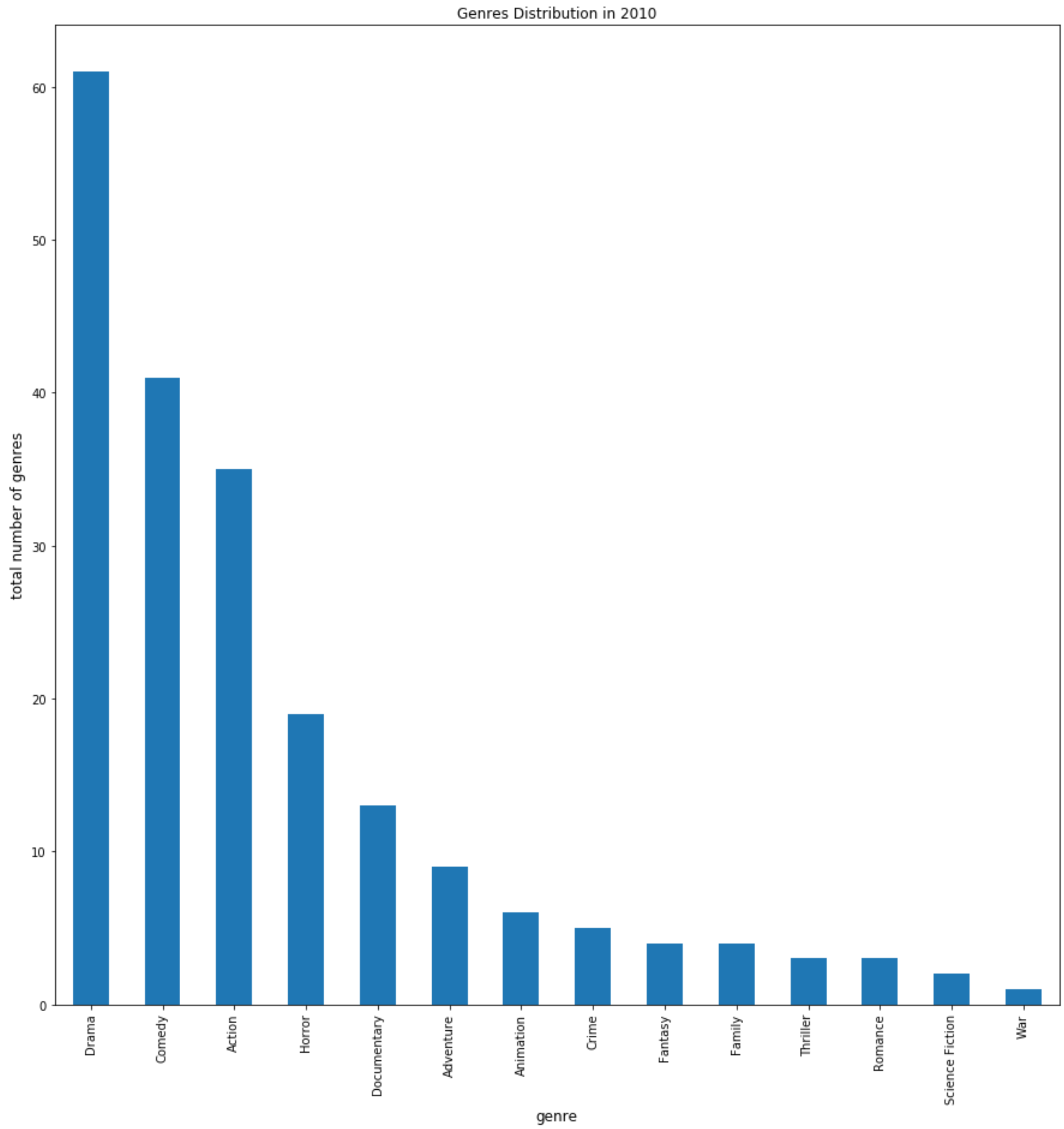
In [33]:
```
# Distribution of genres produced in 2010
df_movie2010a = df_movie1.query('release_year == "2010"')
df_movie2010b=df_movie2010a['genres'].value_counts()
```

In [34]:
```
df_movie2010b
```

Out[34]:
```
Drama              61
Comedy             41
Action             35
Horror             19
Documentary        13
Adventure           9
Animation           6
Crime               5
Fantasy             4
Family              4
Thriller            3
Romance             3
Science Fiction     2
War                 1
Name: genres, dtype: int64
```

In [35]:
```
df_movie2010b.plot(kind='bar',figsize=(15,15),title='Genres Distribution in 20
10')
plt.xlabel('genre',fontsize=12)
plt.ylabel('total number of genres',fontsize=12)
```

Out[35]: Text(0,0.5,'total number of genres')



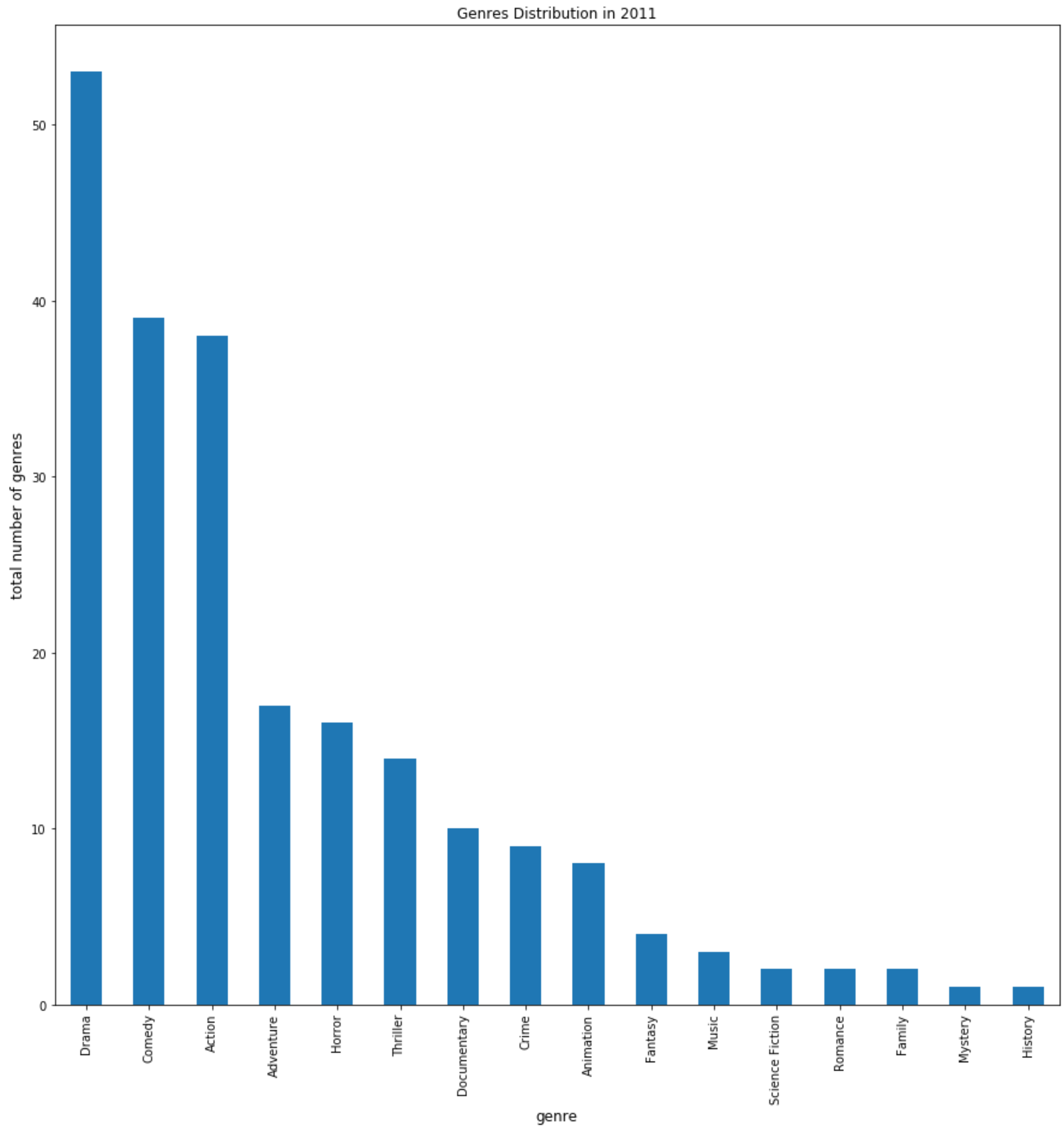
In [36]:
```
# Distribution of genres produced in 2011
df_movie2011a = df_movie1.query('release_year == "2011"')
df_movie2011b=df_movie2011a['genres'].value_counts()
```

In [37]:
```
df_movie2011b
```

Out[37]:
```
Drama              53
Comedy             39
Action             38
Adventure          17
Horror             16
Thriller           14
Documentary        10
Crime               9
Animation           8
Fantasy             4
Music               3
Science Fiction     2
Romance             2
Family              2
Mystery             1
History             1
Name: genres, dtype: int64
```

In [38]:
```
df_movie2011b.plot(kind='bar',figsize=(15,15),title='Genres Distribution in 20
11')
plt.xlabel('genre',fontsize=12)
plt.ylabel('total number of genres',fontsize=12)
```

Out[38]: Text(0,0.5,'total number of genres')



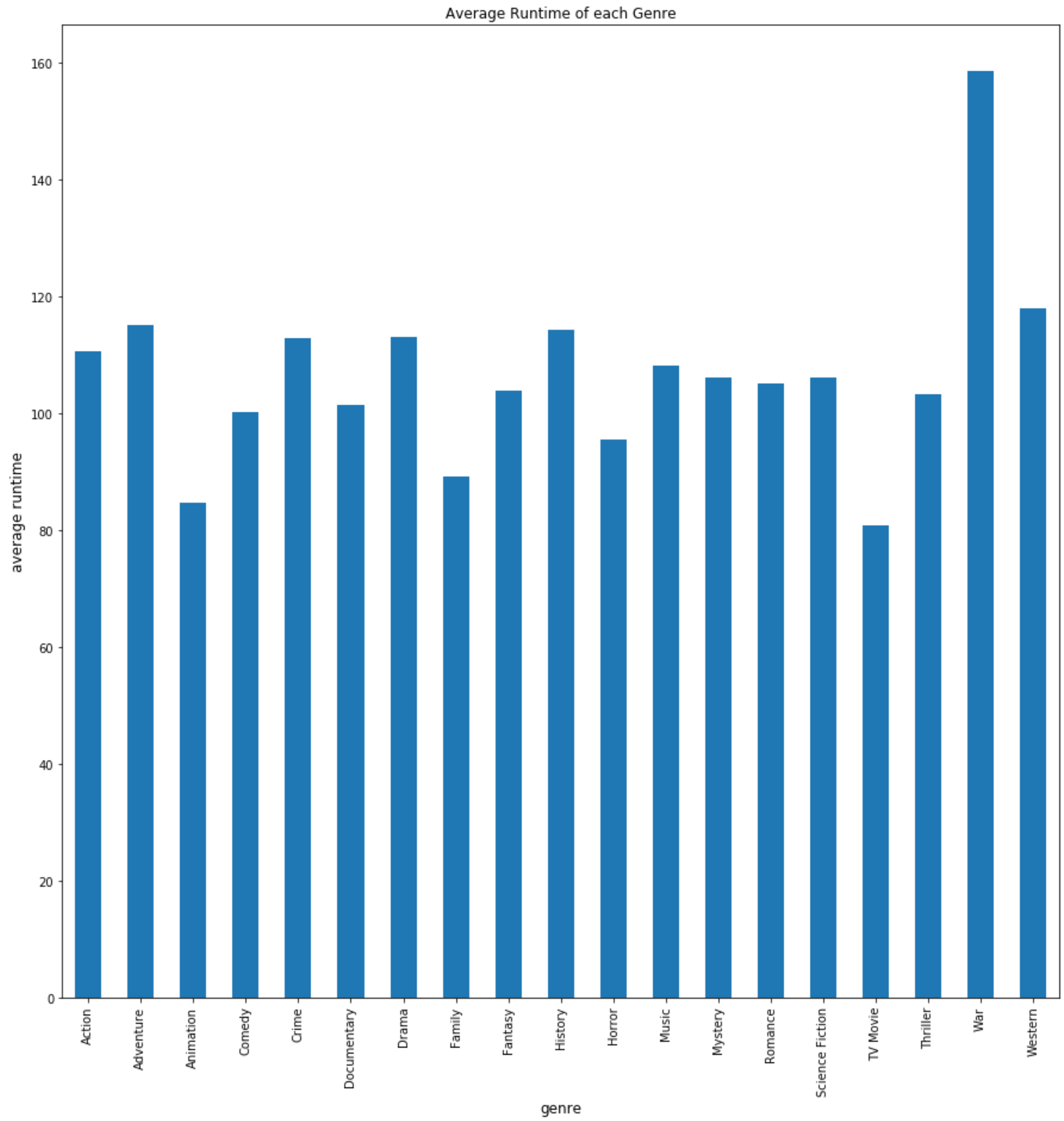
Research Question 7: Which genre has the longest run time on the average?

Discussion:

> The essence of this question is to know the impact of run time on the popularity of the genres. The question aims to determine if a genre with long runtime on the average will be less popular. On the average, the war genre has the longest run time. From Questions 6 above, the genre seems to be less popular. Unfortunately, the dataset is not sufficient to prove a direct relationship between popularity and runtime. More features will be needed to ascertain any relationship.

In [36]:
```
df_movie1.groupby('genres')['runtime'].mean().plot(kind='bar',figsize=(15,15),
title='Average Runtime of each Genre')
plt.xlabel('genre',fontsize=12)
plt.ylabel('average runtime',fontsize=12)
```

Out[36]: Text(0,0.5,'average runtime')



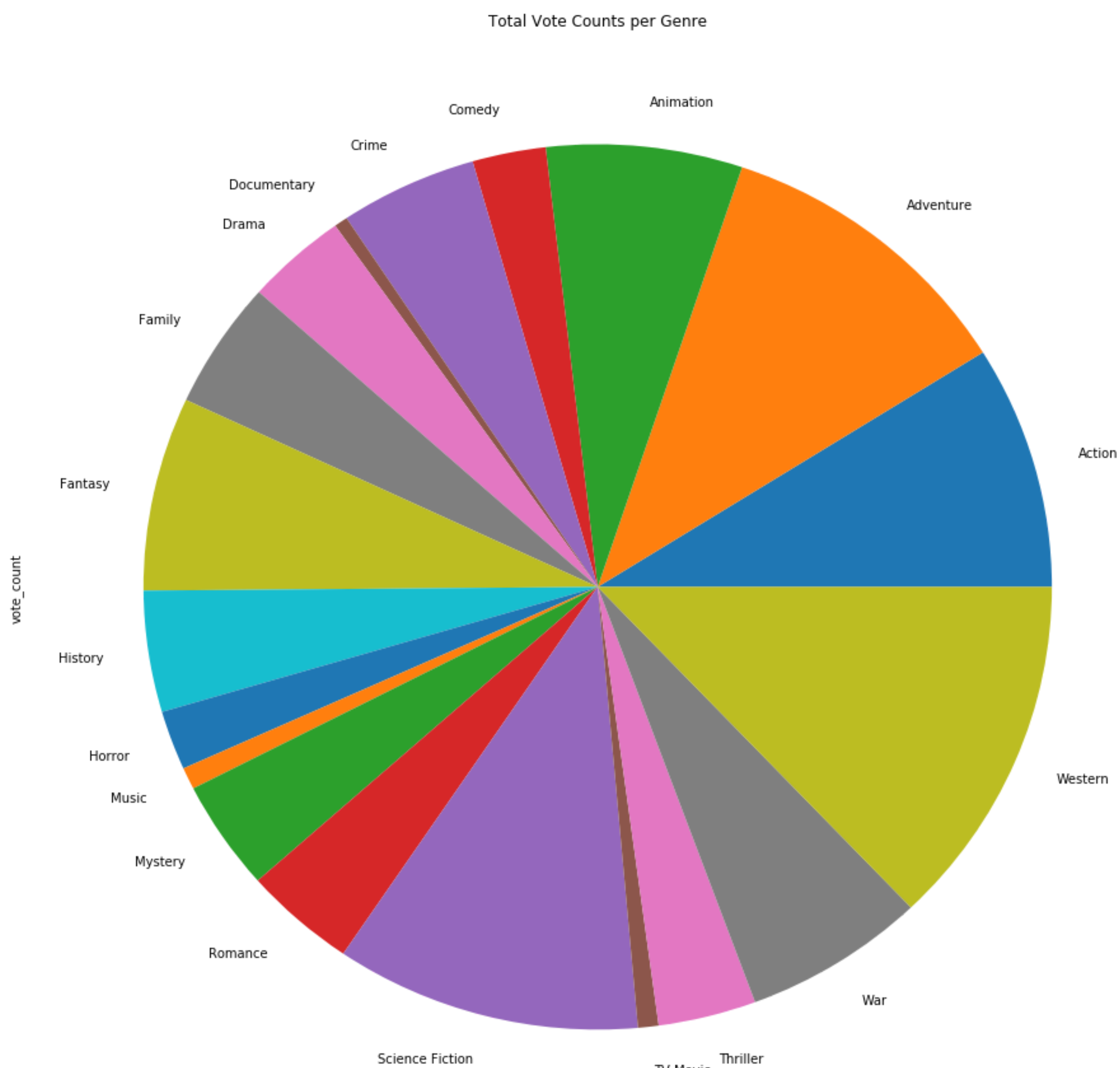
Research Question 8: Over the years, which genres are watched the most?

Discussion:

> This question aims to give the distribution of the cumulative patronage of the genres; the pie chart is plotted to give a sense of proportions of the cumulative patronage of the genres and a bar chart to show the total cumulative patronage. The Western movies are the most watched averagely per year, followed by Action, Science Fiction, and Adventure. On the other hand, cumulatively the Western movies have the least vote count among likely customers.

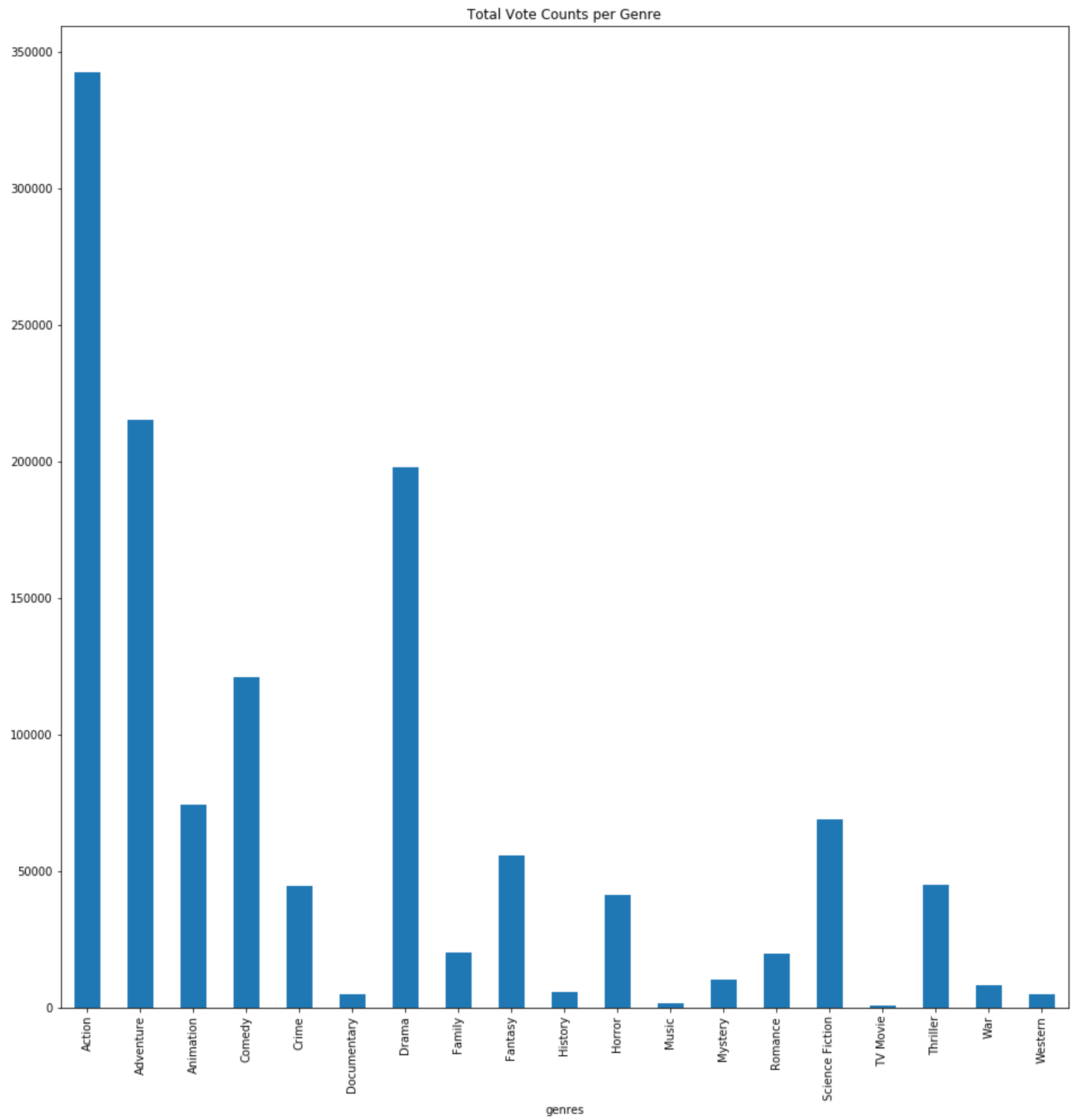
In [25]:
```
df_movie1.groupby('genres')['vote_count'].mean().plot(kind='pie',figsize=(16,16),title='Total Vote Counts per Genre')
```

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x7f83998b0b00>



In [42]:
```
df_movie1.groupby('genres')['vote_count'].sum().plot(kind='bar',figsize=(16,16
),title='Total Vote Counts per Genre')
```

Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3dfc7544a8>



Conclusions

> From the analyses above the following conclusions can be drawn: The highest patronage of movies by customers was in the year 2012, although investment by production companies dipped that year compared with the years 2011, 2010, and 2009. Over the years the patronage of movies has grown exponentially. Despite the high patronage from customers in 2012, the movie industry made highest profit in the preceding year in history.
>
> The reason the industry made the highest profit in 2011 could be inferred that the industry produced the highest number of movies this year; 2010 and 2009 are years with high profit also. In 2009 the highest number of movies with different genres (seventeen) were produced, followed by 2011.
>
> War has the longest run time among all the genres, but one of the least watched; Western is the least watched. The correlation between these two facts could not be ascertained drawn from the data, but I will suggest that may be the customers do not like movies with long run time.
>
> The most watched of the genres are Action, Science Fiction, and Adventure.

Further Works

> More features are needed to ascertain the correlations among the different features in this dataset. Nevertheless, the limited features in this dataset has been helpful in providing some weak connclusions that can further assist in determining what kind of features will be needed in order to make strong conclusions or inferences. For instance, to have more insights into the force behind the patronage of the genres, the demographic distribution of the voters, socio-economic status, cultural orientation, and others are needed. The current features could not help in determining why drama seemed to be the most watched genre on average. Another example is the relationsip between runtime and popularity. Could it be possible that War genre is not highly popular because of the long runtime?