Change in movie genre development trends since 1970

Analysis based on IMDB data set.

In [94]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Reading the main file

In [95]:
```
data_set = pd.read_csv('IMDB data.tsv', sep='\t',low_memory=False)
data_set.head(5)
```

Out[95]:

	tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtimeM
0	tt0000001	short	Carmencita	Carmencita	0	1894	\N	1
1	tt0000002	short	Le clown et ses chiens	Le clown et ses chiens	0	1892	\N	5
2	tt0000003	short	Pauvre Pierrot	Pauvre Pierrot	0	1892	\N	4
3	tt0000004	short	Un bon bock	Un bon bock	0	1892	\N	\N
4	tt0000005	short	Blacksmith Scene	Blacksmith Scene	0	1893	\N	1

Filtering the data to only contain movie type

In [96]:
```
movies= data_set[data_set.loc[:,'titleType']=='movie']
movies.head()
```

Out[96]:

	tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtime
8	tt0000009	movie	Miss Jerry	Miss Jerry	0	1894	\N	45
145	tt0000147	movie	The Corbett-Fitzsimmons Fight	The Corbett-Fitzsimmons Fight	0	1897	\N	20
332	tt0000335	movie	Soldiers of the Cross	Soldiers of the Cross	0	1900	\N	\N
499	tt0000502	movie	Bohemios	Bohemios	0	1905	\N	100
571	tt0000574	movie	The Story of the Kelly Gang	The Story of the Kelly Gang	0	1906	\N	70

Filtering out the extra columns to have the genre and release date only

In [97]:
```
genre_yr = movies.loc[:,['startYear','genres']]
genre_yr.head()
```

Out[97]:

	startYear	genres
8	1894	Romance
145	1897	Documentary,News,Sport
332	1900	Biography,Drama
499	1905	\N
571	1906	Biography,Crime,Drama

Checking if there are rows with missing data

In [98]:
```
genre_yr.isnull().any()
```

Out[98]:
```
startYear    False
genres       False
dtype: bool
```

Finding the total types of genres avaible

In [99]:
```
temp = genre_yr['genres'].str.split(',',expand = True)
all_genres = pd.unique(temp[:].values.ravel('K'))
all_genres
```

Out[99]:
```
array(['Romance', 'Documentary', 'Biography', '\\N', 'Drama', 'Adventure',
       'Comedy', 'Crime', 'War', 'Sci-Fi', 'History', 'Western',
       'Fantasy', 'Action', 'Horror', 'Thriller', 'Mystery', 'Animation',
       'Music', 'Musical', 'Sport', 'Family', 'Film-Noir', 'Adult',
       'News', 'Game-Show', 'Reality-TV', 'Talk-Show', 'Short', None],
      dtype=object)
```

Deleting rows with genre '\N' and None

In [102]:
```
temp = genre_yr[genre_yr.loc[:,'genres']=='\\N']
genre_yr = genre_yr.drop(temp.index)
temp = genre_yr[genre_yr.loc[:,'genres']==None]
genre_yr = genre_yr.drop(temp.index)
```

Checking the total genres again

In [177]:
```
temp = genre_yr['genres'].str.split(',',expand = True)
all_genres = pd.unique(temp[:].values.ravel('K'))
all_genres
```

Out[177]:
```
array(['Romance', 'Documentary', 'Biography', 'Drama', 'Adventure',
       'Comedy', 'Crime', 'War', 'Sci-Fi', 'History', 'Western',
       'Fantasy', 'Action', 'Horror', 'Thriller', 'Mystery', 'Animation',
       'Music', 'Musical', 'Sport', 'Family', 'Film-Noir', 'Adult',
       'News', 'Game-Show', 'Reality-TV', 'Talk-Show', 'Short', None],
      dtype=object)
```

Counting the total number of movies by year in each genre

In [178]:
```
TMG = pd.DataFrame() #Total Movie Genre by year
for i in range(28):
    temp = genre_yr[genre_yr['genres'].str.contains(all_genres[i])]

    TMG[all_genres[i]] = temp.groupby(['startYear']).genres.count()
```

Checking the value of TMG

In [179]:
```
TMG.head()
```

Out[179]:

	Romance	Documentary	Biography	Drama	Adventure	Comedy	Crime	War
startYear								
1894	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1898	1	8.0	NaN	NaN	NaN	NaN	NaN	NaN
1908	1	5.0	NaN	4.0	1.0	NaN	NaN	NaN
1909	1	16.0	1.0	10.0	1.0	5.0	1.0	3.0
1910	1	24.0	1.0	21.0	2.0	4.0	2.0	1.0

5 rows × 28 columns

Filtering out movies before 1970

In [180]:
```
TMG = TMG.drop(TMG.index[:64])
TMG.head()
```

Out[180]:

	Romance	Documentary	Biography	Drama	Adventure	Comedy	Crime	Wa
startYear								
1970	357	313.0	32.0	1383.0	267.0	650.0	236.0	107
1971	297	292.0	31.0	1288.0	238.0	613.0	254.0	75.0
1972	282	263.0	33.0	1259.0	229.0	619.0	288.0	59.0
1973	240	268.0	45.0	1267.0	194.0	591.0	332.0	70.0
1974	284	281.0	32.0	1244.0	194.0	637.0	284.0	65.0

5 rows × 28 columns

In [181]:
```
TMG.tail()
```

Out[181]:

	Romance	Documentary	Biography	Drama	Adventure	Comedy	Crime	\
startYear								
2017	966	6398.0	751.0	5698.0	673.0	2880.0	762.0	15
2018	848	3983.0	476.0	5290.0	587.0	2649.0	715.0	12
2019	41	495.0	53.0	396.0	86.0	175.0	73.0	13
2020	1	1.0	NaN	7.0	16.0	10.0	2.0	Na
\N	1947	2168.0	1284.0	15400.0	2562.0	7465.0	2451.0	50

5 rows × 28 columns

Filtering out upcoming movie dates, i.e. 2018 onwards

In [182]:
```
TMG = TMG.drop(TMG.index[-4:])
TMG.tail()
```

Out[182]:

	Romance	Documentary	Biography	Drama	Adventure	Comedy	Crime	Wa
startYear								
2013	1019	5596.0	1286.0	5290.0	787.0	2623.0	711.0	117.
2014	1127	5991.0	1402.0	5404.0	795.0	2788.0	725.0	174
2015	1046	6035.0	977.0	5389.0	697.0	2722.0	723.0	169
2016	1040	6150.0	841.0	5509.0	697.0	2814.0	853.0	162
2017	966	6398.0	751.0	5698.0	673.0	2880.0	762.0	157

5 rows × 28 columns

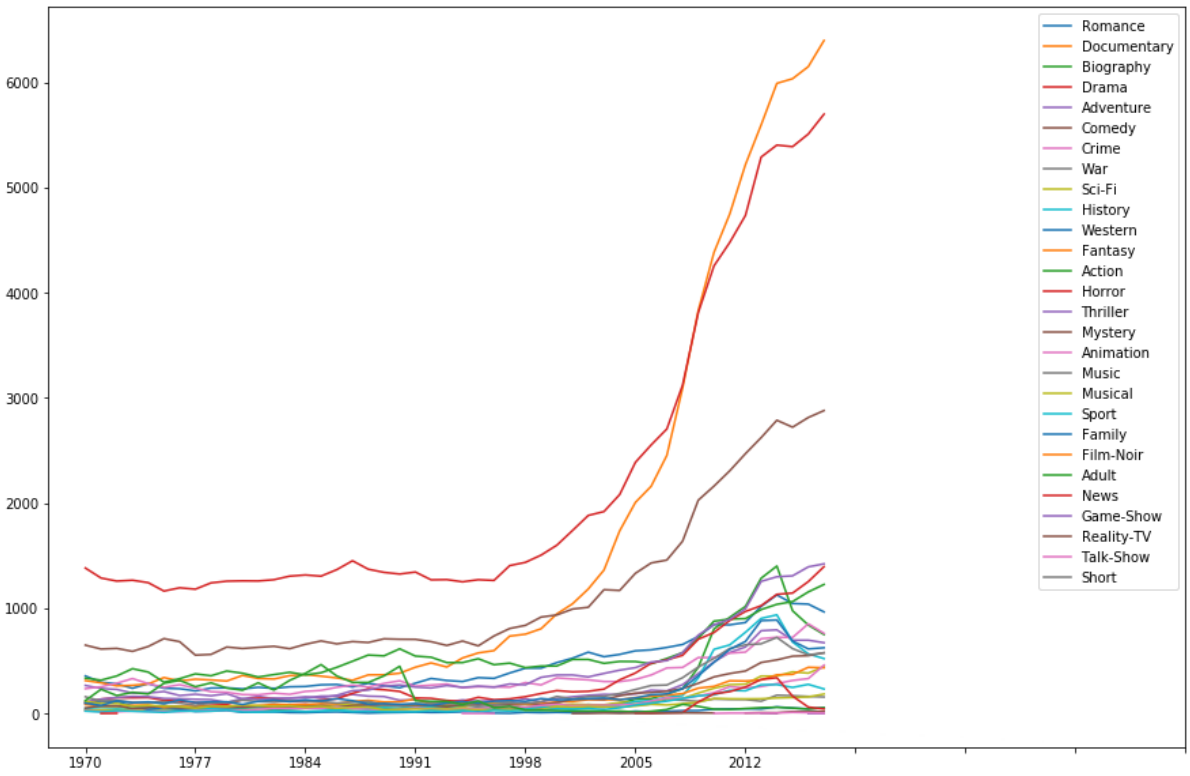
Plotting the graph of a movie genres over years}

In [183]:
```
plt.figure(figsize=(15,10))
plt.plot(TMG)

plt.xticks(np.arange(0,75, step =7))
plt.legend(all_genres)

plt.show
```

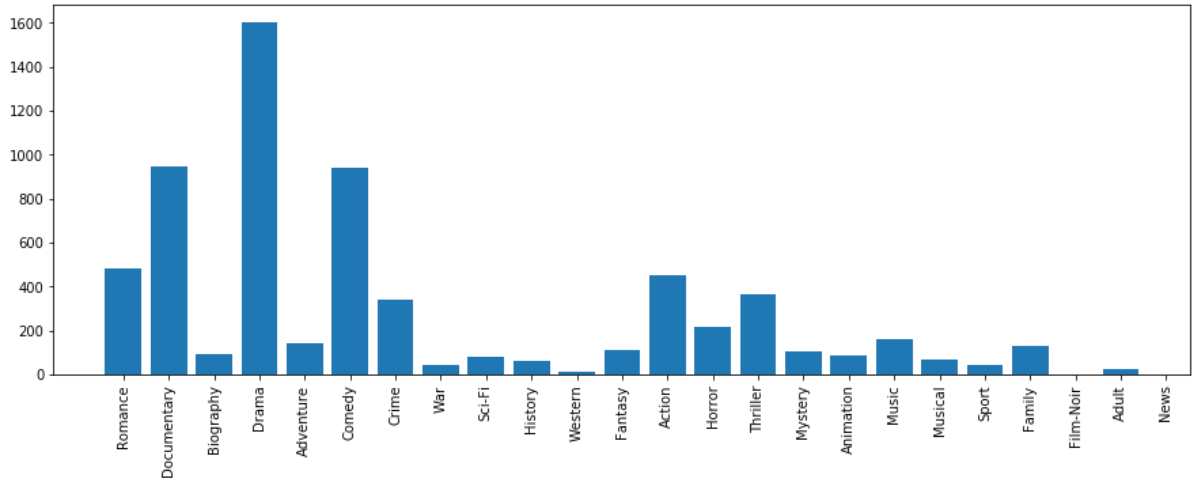
Out[183]: <function matplotlib.pyplot.show(*args, **kw)>



Comparing the number of films per genre in 2000 vs 2017

In [215]:
```
plt.figure(figsize=(15,5))
plt.bar(['Romance', 'Documentary', 'Biography', 'Drama', 'Adventure',
       'Comedy', 'Crime', 'War', 'Sci-Fi', 'History', 'Western',
       'Fantasy', 'Action', 'Horror', 'Thriller', 'Mystery', 'Animation',
       'Music', 'Musical', 'Sport', 'Family', 'Film-Noir', 'Adult',
       'News', 'Game-Show', 'Reality-TV', 'Talk-Show', 'Short'],height = TM
G.iloc[30,:])
plt.xticks(fontsize=10,rotation='vertical')
plt.plot()
```

Out[215]: []

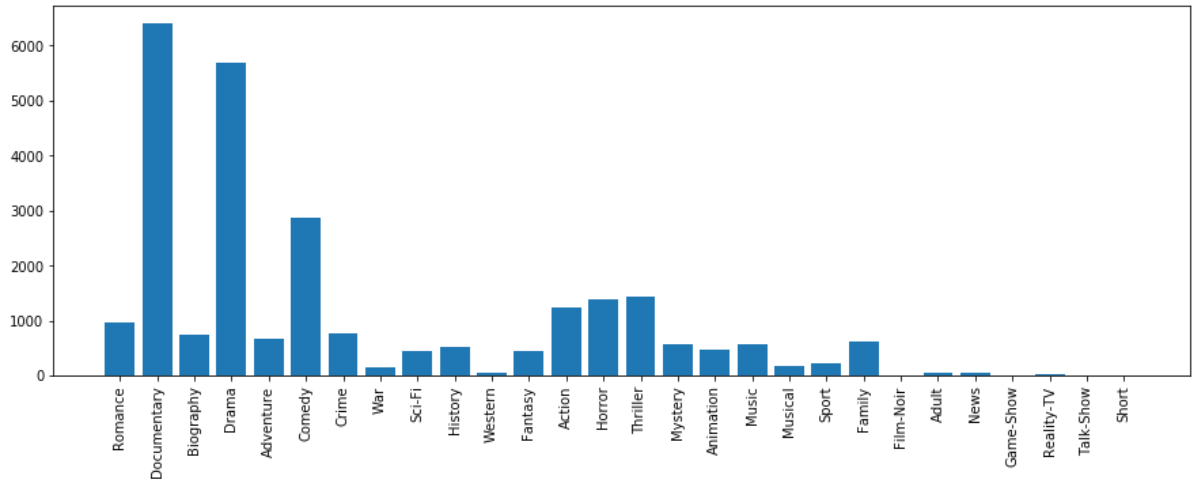


In [216]:
```
plt.figure(figsize=(15,5))
plt.bar(['Romance', 'Documentary', 'Biography', 'Drama', 'Adventure',
       'Comedy', 'Crime', 'War', 'Sci-Fi', 'History', 'Western',
       'Fantasy', 'Action', 'Horror', 'Thriller', 'Mystery', 'Animation',
       'Music', 'Musical', 'Sport', 'Family', 'Film-Noir', 'Adult',
       'News', 'Game-Show', 'Reality-TV', 'Talk-Show', 'Short'],height = TM
G.iloc[-1,:])

plt.xticks(fontsize=10,rotation='vertical')

plt.plot()
```

Out[216]: []



Deleting outliers

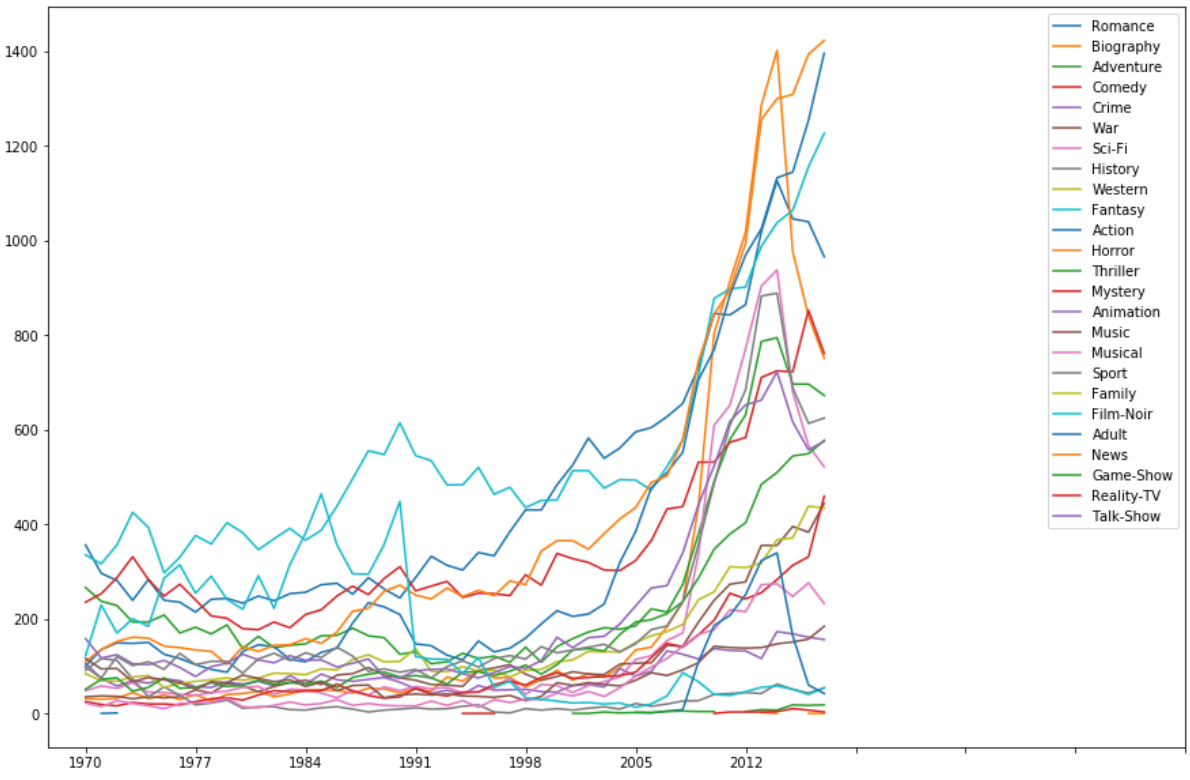
In [ ]:
```
del TMG['Documentary']
del TMG['Drama']
```

In [161]:
```
plt.figure(figsize=(15,10))
plt.plot(TMG)

plt.xticks(np.arange(0,75, step =7))
plt.legend(['Romance','Biography', 'Adventure',
       'Comedy', 'Crime', 'War', 'Sci-Fi', 'History', 'Western',
       'Fantasy', 'Action', 'Horror', 'Thriller', 'Mystery', 'Animation',
       'Music', 'Musical', 'Sport', 'Family', 'Film-Noir', 'Adult',
       'News', 'Game-Show', 'Reality-TV', 'Talk-Show', 'Short'])

plt.show
```

Out[161]:
```
<function matplotlib.pyplot.show(*args, **kw)>
```



Plotting the well known genres.

In [228]:
```
plt.figure(figsize=(15,10))
plt.plot(TMG.loc[:,['Romance','Sci-Fi','Action','History']])

plt.xticks(np.arange(0,47, step =4))
plt.legend(['Romance','Sci-Fi','Action','History'])
plt.grid(True)
plt.show
```

Out[228]:
```
<function matplotlib.pyplot.show(*args, **kw)>
```