Choosing Dataset:

TMDb movie data

Questions

1. Which genres are most popular from year to year?
    - (What changed between 2015 to 2016)
2. What kinds of properties are associated with movies that have high revenues?
    - or what are the attributes of the movie with high revenues
    - like What properties/attributes does the movies, which have done well at the box office, have?
    - For Instance, Whether the popularity of the movie is dependent on the movie's budget?

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
%pylab inline
```

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

In [2]:
```
data = pd.read_csv("tmdb-movies.csv")
```

In [3]:
```
data.head()
```

Out[3]:

	id	imdb_id	popularity	budget	revenue	original_title	cast
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...

5 rows × 21 columns

In [4]:
```
data.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
```

In [5]:
```
# DATA WRANGLING

# cast, homepage, director, tagline, keywords, genres, production_companies co
lumns have less than 10866 values
# so we will inspect these columns

data['homepage'] = data['homepage'].fillna('Homepage Unavailable')
data['cast'] = data['cast'].fillna('Information not available')
data['director'] = data['director'].fillna('Information not available')
data['tagline'] =data['tagline'].fillna('Will Update soon!')
data['keywords'] = data['keywords'].fillna('')
data['overview'] =data['overview'].fillna('Will Update soon!')
data['genres'] = data['genres'].fillna('NA')
data['production_companies'] = data['production_companies'].fillna('')
data['imdb_id'] = data['imdb_id'].fillna('NA')
# print(data['homepage'])
# data['imdb_id'].isnull().values.any()
# data['imdb_id'].isnull().sum()
```

In [6]:
```
# data.info()
data.describe()

# so we can see that there is no revenue and budget for some of the movies

# count_budget = 0
# count_revenue = 0
# for i in range(len(data)):
#     count_budget = (count_budget + 1) if data.loc[i, 'budget'] == 0 else cou
nt_budget
#     count_revenue = (count_revenue + 1) if data.loc[i, 'revenue'] == 0 else
 count_revenue
# print(count_budget, count_revenue)
```

Out[6]:

	id	popularity	budget	revenue	runtime	vote_
count	10866.000000	10866.000000	1.086600e+04	1.086600e+04	10866.000000	10866.0
mean	66064.177434	0.646441	1.462570e+07	3.982332e+07	102.070863	217.389
std	92130.136561	1.000185	3.091321e+07	1.170035e+08	31.381405	575.619
min	5.000000	0.000065	0.000000e+00	0.000000e+00	0.000000	10.0000
25%	10596.250000	0.207583	0.000000e+00	0.000000e+00	90.000000	17.0000
50%	20669.000000	0.383856	0.000000e+00	0.000000e+00	99.000000	38.0000
75%	75610.000000	0.713817	1.500000e+07	2.400000e+07	111.000000	145.750
max	417859.000000	32.985763	4.250000e+08	2.781506e+09	900.000000	9767.00

Posing 1st Question

Which genres are most popular from year to year?

In [7]:
```
print("From: {} to {}.\n".format(data['release_year'].min(), data['release_yea
r'].max()))
from_1960_to_2015 = range(data['release_year'].min(), data['release_year'].max
() + 1)

genre_column = list(data["genres"])
genre_set = set()
for i in genre_column:
    row = i.split("|")
    for value in row:
        genre_set.add(value)

print(len(genre_set), genre_set)
# total -> 20 genres with one extra 'NA' genre which is substituted in place o
f the genres
# with missing values in the original data
```

```
From: 1960 to 2015.

21 {'Comedy', 'NA', 'Music', 'Action', 'War', 'Thriller', 'Science Fiction',
'Adventure', 'Western', 'Crime', 'Family', 'Documentary', 'History', 'Horro
r', 'TV Movie', 'Mystery', 'Animation', 'Foreign', 'Drama', 'Romance', 'Fanta
sy'}
```

In [8]:
```
# row by column dataframe of years by genres
years_by_genres = pd.DataFrame(index = from_1960_to_2015, columns = genre_set)
years_by_genres = years_by_genres.fillna(0)
years_by_genres.head()

for i in range(len(data)):
    genres_per_year = data.loc[i, 'genres'].split("|")
    year = data.loc[i, 'release_year']
    for val in genres_per_year:
        years_by_genres.loc[year, val] = years_by_genres.loc[year, val] + 1

years_by_genres.describe()
```

Out[8]:

	Comedy	NA	Music	Action	War	Thriller	Science Fiction
count	56.000000	56.000000	56.000000	56.000000	56.000000	56.000000	56.000000
mean	67.732143	0.410714	7.285714	42.589286	4.821429	51.928571	21.964286
std	59.957710	0.826312	7.581591	35.981195	4.204358	51.350793	19.393499
min	5.000000	0.000000	0.000000	4.000000	0.000000	0.000000	2.000000
25%	13.000000	0.000000	2.750000	12.500000	2.000000	15.250000	6.000000
50%	52.500000	0.000000	4.000000	32.500000	4.000000	30.500000	18.000000
75%	101.750000	1.000000	10.250000	63.250000	6.250000	67.750000	28.250000
max	198.000000	4.000000	33.000000	129.000000	23.000000	179.000000	86.000000

8 rows × 21 columns

These statistics clearly show that "Drama" genre is the most watched genre overall until now, followed by "Comedy" genre.

```
- So, the maximum number of movies fall in the category of Drama genre
```

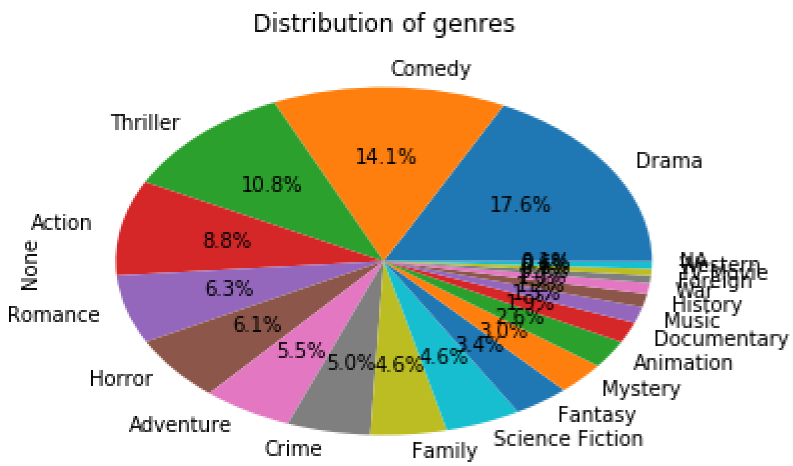
In [386]:

```
# Most popular genres from year to year
year_to_year = pd.Series(np.zeros(len(years_by_genres)), index=from_1960_to_20
15)

# as i could see more than 1 genre having max values so i will individually lo
op through each row
for i in from_1960_to_2015:
    maximum = years_by_genres.max(axis=1).loc[i]
    genre_temp = "| "
    for j in genre_set:
        if maximum == years_by_genres.loc[i, j]:
            genre_temp = genre_temp + j + " | "
    year_to_year.loc[i] = genre_temp

print("Year\t\tPopular Genres\n\n{}".format(year_to_year))

genres_total = pd.Series(np.zeros(len(genre_set)), index = genre_set)
for j in genre_set:
    genres_total.loc[j] = years_by_genres[j].sum()
genres_total.sort_values(ascending=False).plot(kind="pie", autopct = '%1.1f%%'
, title="Distribution of genres")
```

Year	Popular Genres	
1960		Drama
1961		Drama
1962		Drama
1963	Comedy	Drama
1964		Drama
1965		Drama
1966	Comedy	Drama
1967		Comedy
1968		Drama
1969		Drama
1970		Drama
1971		Drama
1972		Drama
1973		Drama
1974		Drama
1975		Drama
1976		Drama
1977		Drama
1978		Drama
1979		Drama
1980		Drama
1981		Drama
1982		Drama
1983		Drama
1984		Drama
1985		Comedy
1986		Drama
1987		Comedy
1988		Comedy
1989		Comedy
1990		Drama
1991		Drama
1992		Drama
1993		Drama
1994		Comedy
1995		Drama
1996		Drama
1997		Drama
1998		Drama
1999		Drama
2000		Drama
2001		Comedy
2002		Drama
2003		Comedy
2004		Drama
2005		Drama
2006		Drama
2007		Drama
2008		Drama
2009		Drama
2010		Drama
2011		Drama
2012		Drama
2013		Drama
2014		Drama

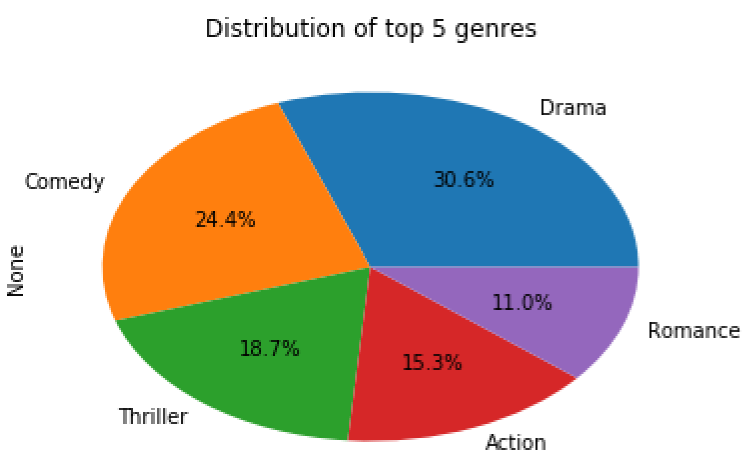
```
2015              | Drama |
dtype: object
```

Out[386]: <matplotlib.axes._subplots.AxesSubplot at 0x5d1dd18860>



In [387]:
```
genres_total.sort_values(ascending=False)[:5].plot(kind="pie", autopct = '%1.1f%%', title="Distribution of top 5 genres")
```

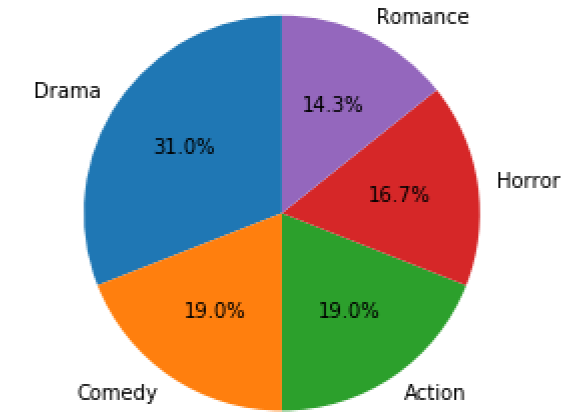
Out[387]: <matplotlib.axes._subplots.AxesSubplot at 0x5d1eed4ba8>



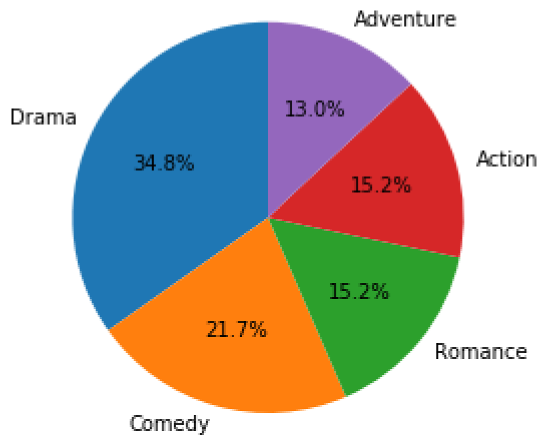
In [375]:

```
# Distribution of Movies per year
for i in from_1960_to_2015:
    fig, axis = plt.subplots()
    # top 5 movies per year shown in pi chart
    n = 5
    distribution = years_by_genres.loc[i, :].sort_values(ascending = False)[0:
n]
    genre_dist = list(distribution.keys())[0:n]
    axis.pie(distribution, labels = genre_dist, autopct = '%1.1f%%', startangl
e = 90)
    axis.axis('equal')
    plt.title("Percentage of movies in each genre in {}".format(i))
    plt.show()
```

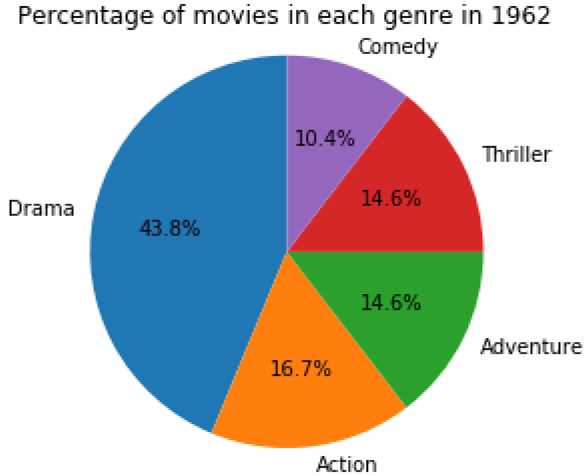
Percentage of movies in each genre in 1960

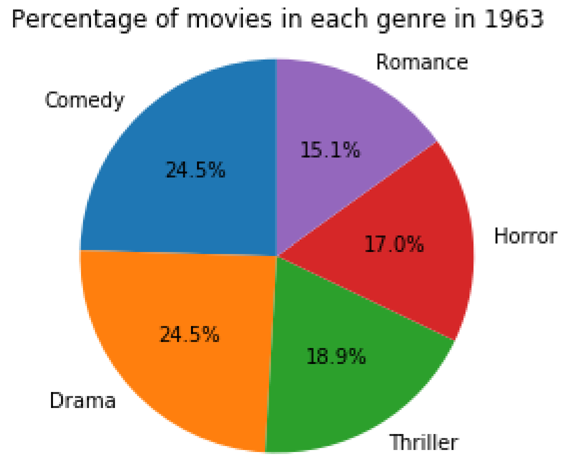


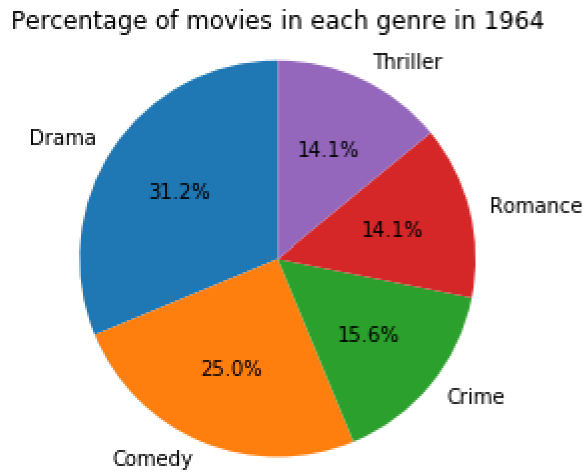
Percentage of movies in each genre in 1961

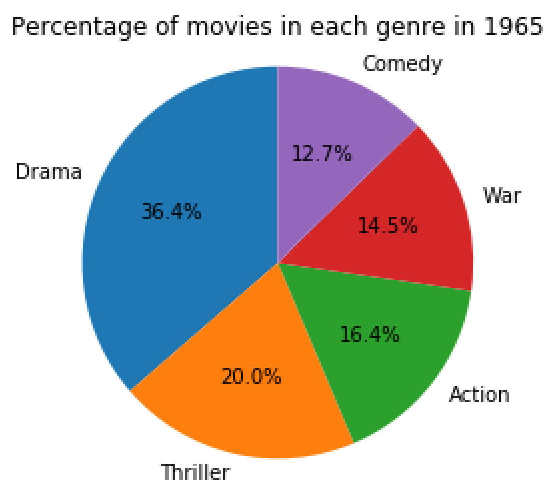


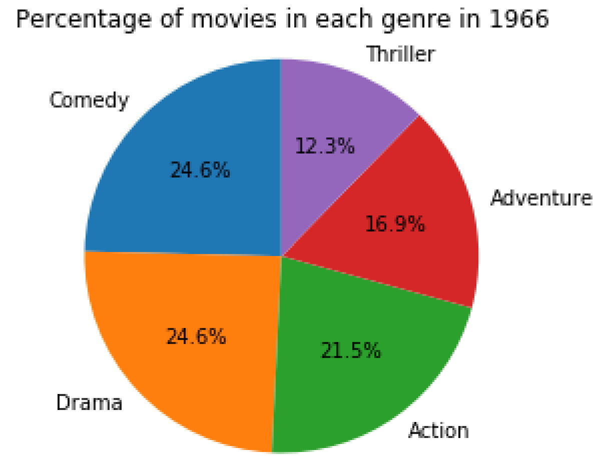
Percentage of movies in each genre in 1962

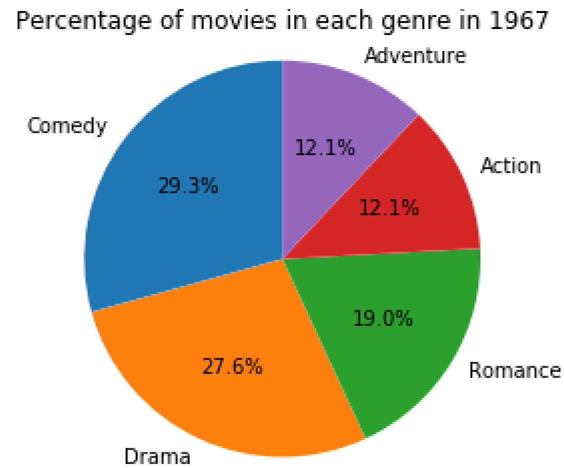


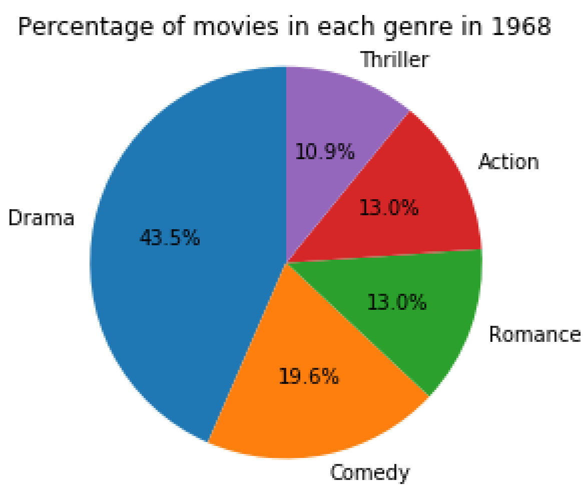

Percentage of movies in each genre in 1963
Romance
Comedy
15.1%
24.5%
17.0%
Horror
24.5%
18.9%
Drama
Thriller


Percentage of movies in each genre in 1964
Thriller
Drama
14.1%
31.2%
Romance
14.1%
15.6%
25.0%
Crime
Comedy

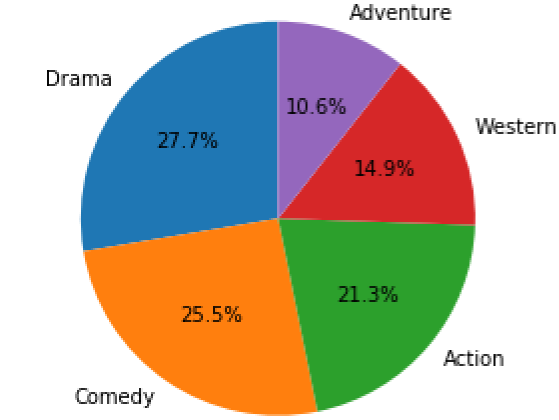

Percentage of movies in each genre in 1965
Comedy
Drama
12.7%
War
36.4%
14.5%
16.4%
20.0%
Action
Thriller


Percentage of movies in each genre in 1966
Thriller
Comedy
12.3%
24.6%
Adventure
16.9%
24.6%
21.5%
Drama
Action

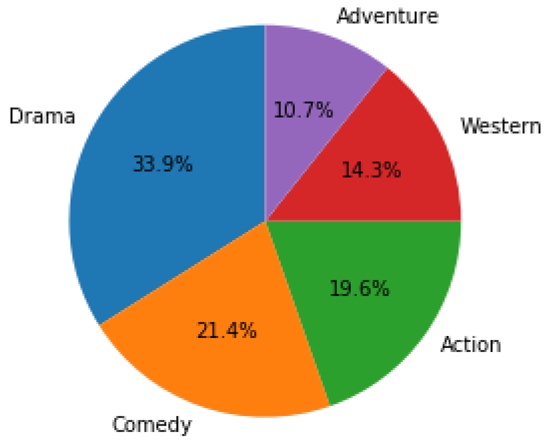

Percentage of movies in each genre in 1967
Adventure
Comedy
12.1%
Action
29.3%
12.1%
19.0%
27.6%
Romance
Drama


Percentage of movies in each genre in 1968
Thriller
10.9%
Action
Drama
13.0%
43.5%
13.0%
Romance
19.6%
Comedy

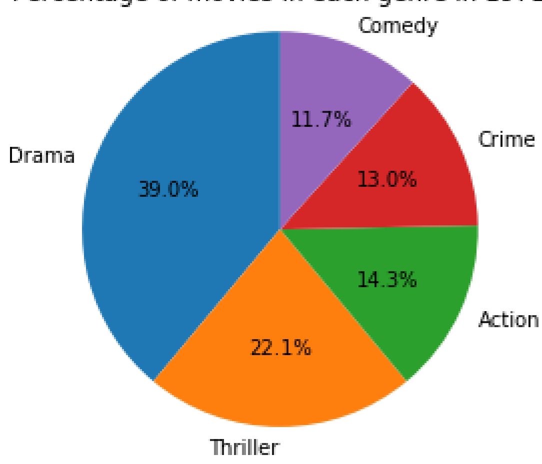
Percentage of movies in each genre in 1969

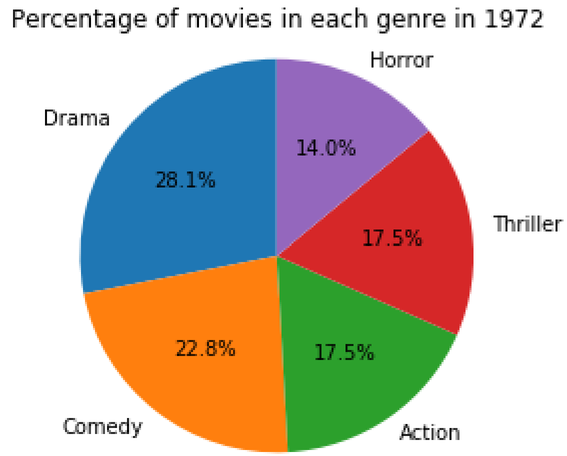


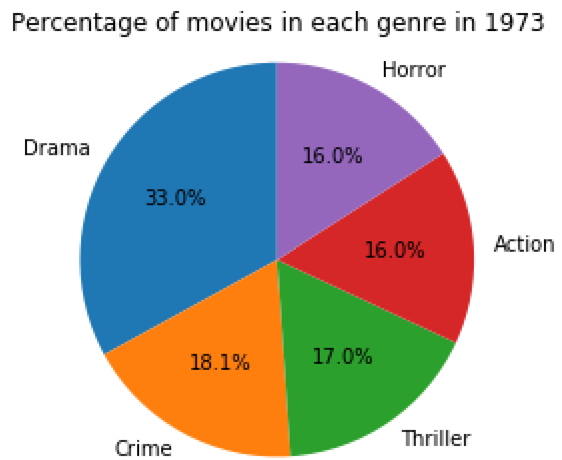
Percentage of movies in each genre in 1970

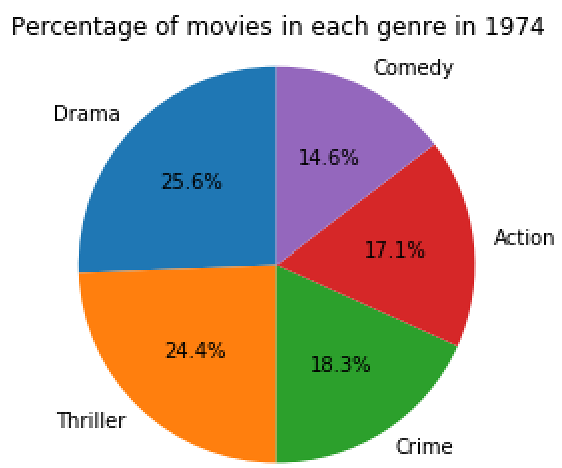


Percentage of movies in each genre in 1971

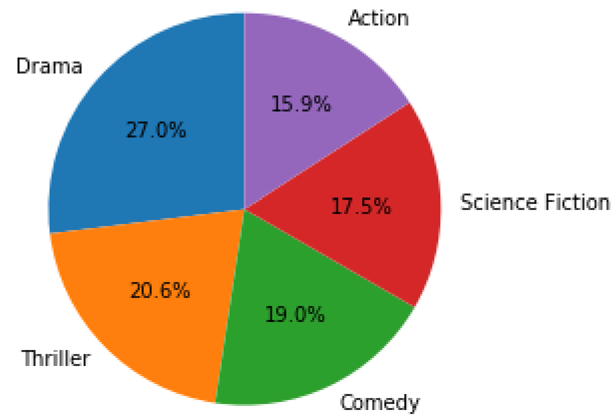



Percentage of movies in each genre in 1972
Horror
14.0%
Drama
28.1%
Thriller
17.5%
22.8%
17.5%
Comedy
Action

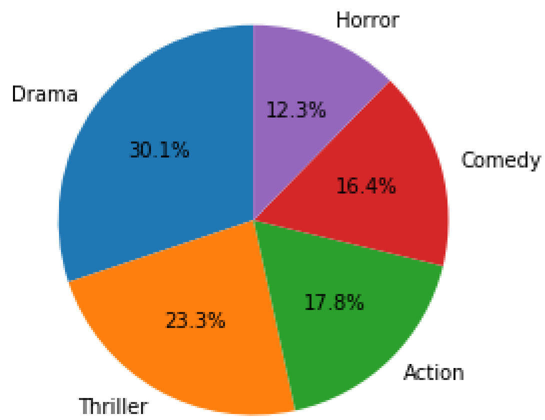

Percentage of movies in each genre in 1973
Horror
Drama
16.0%
33.0%
Action
16.0%
18.1%
17.0%
Thriller
Crime


Percentage of movies in each genre in 1974
Comedy
Drama
14.6%
25.6%
Action
17.1%
24.4%
18.3%
Thriller
Crime

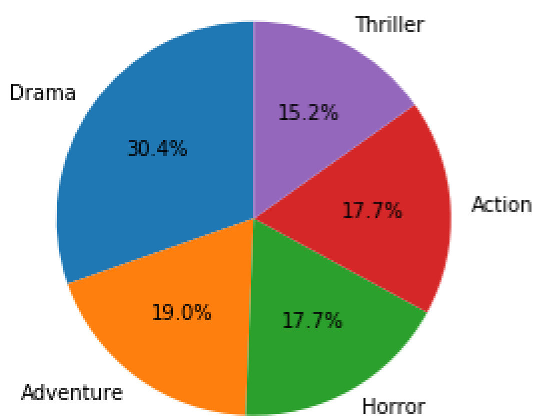
Percentage of movies in each genre in 1975

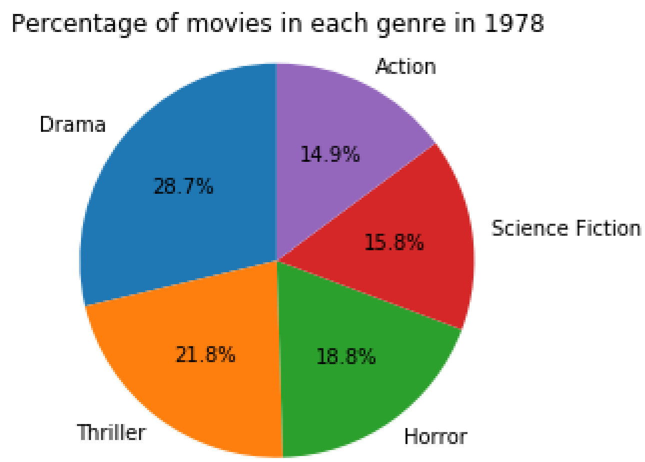


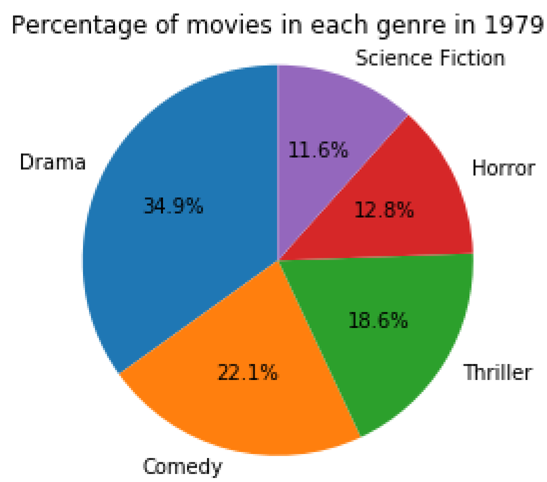
Percentage of movies in each genre in 1976

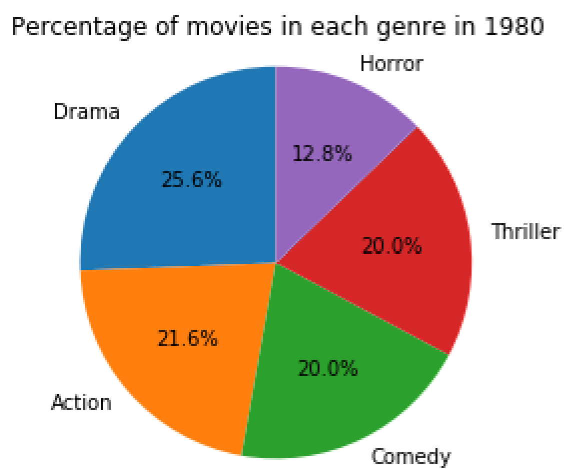


Percentage of movies in each genre in 1977

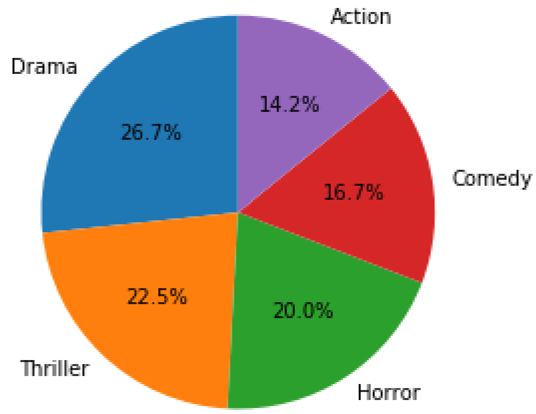



Percentage of movies in each genre in 1978
Action
14.9%
Drama
28.7%
Science Fiction
15.8%
21.8%
18.8%
Thriller
Horror

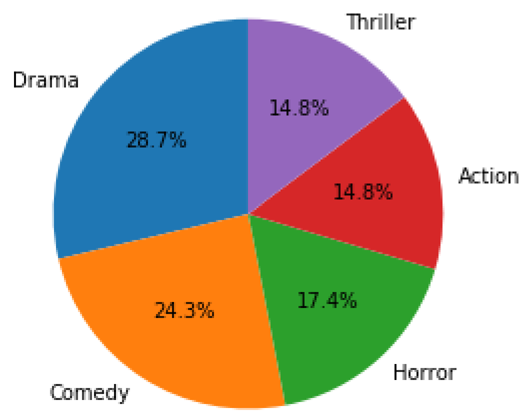

Percentage of movies in each genre in 1979
Science Fiction
11.6%
Drama
Horror
34.9%
12.8%
18.6%
22.1%
Thriller
Comedy


Percentage of movies in each genre in 1980
Horror
Drama
12.8%
25.6%
Thriller
20.0%
21.6%
20.0%
Action
Comedy

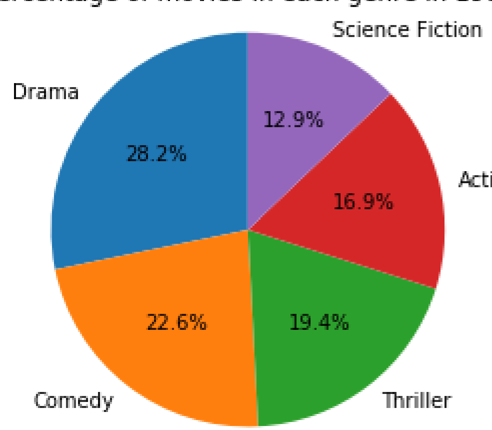
Percentage of movies in each genre in 1981



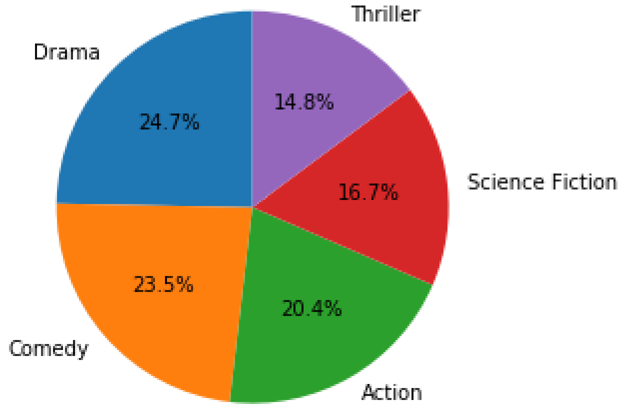
Percentage of movies in each genre in 1982



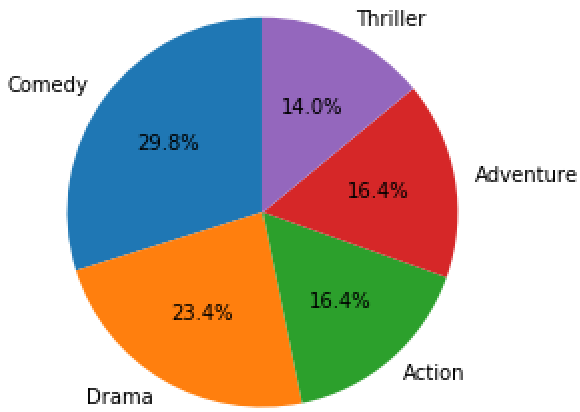
Percentage of movies in each genre in 1983



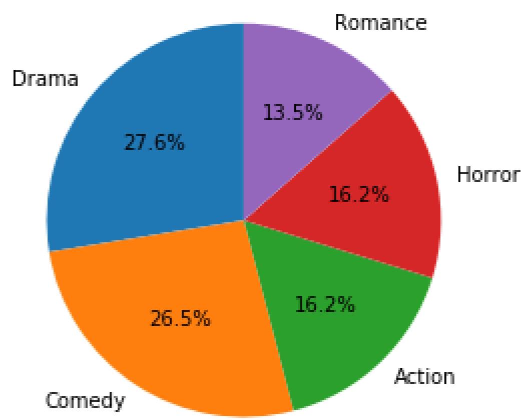
Percentage of movies in each genre in 1984



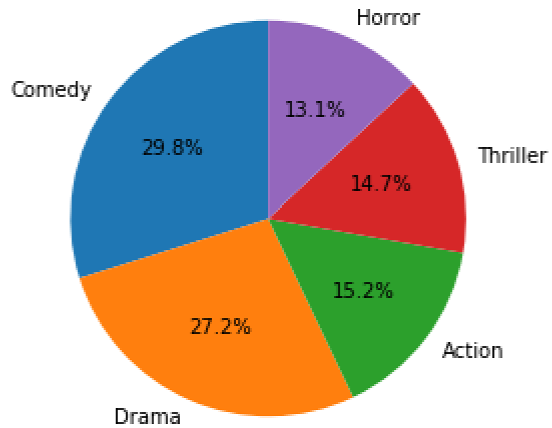
Percentage of movies in each genre in 1985



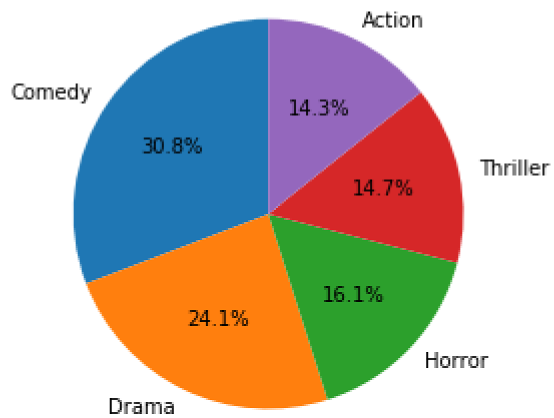
Percentage of movies in each genre in 1986



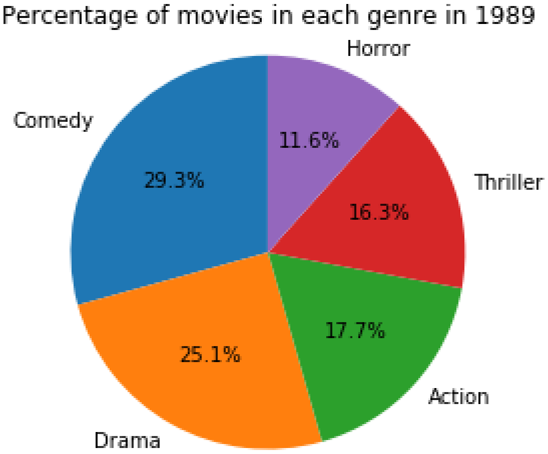
Percentage of movies in each genre in 1987



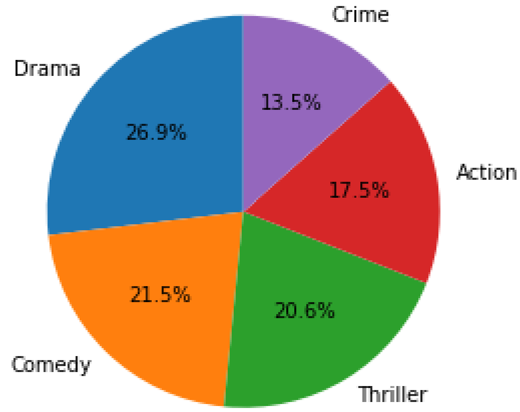
Percentage of movies in each genre in 1988



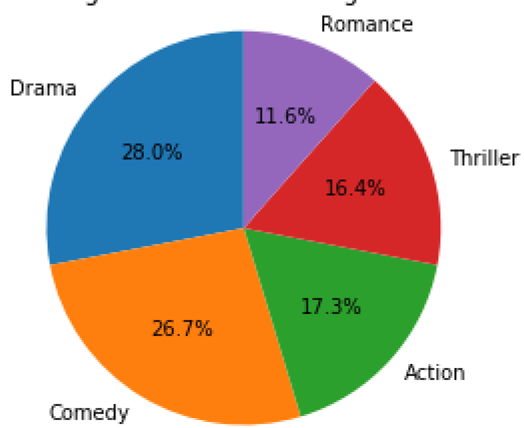
Percentage of movies in each genre in 1989



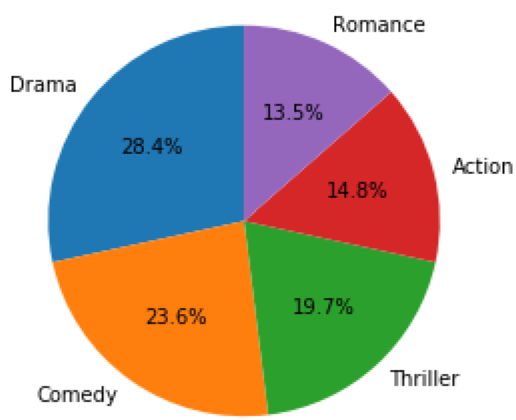
Percentage of movies in each genre in 1990



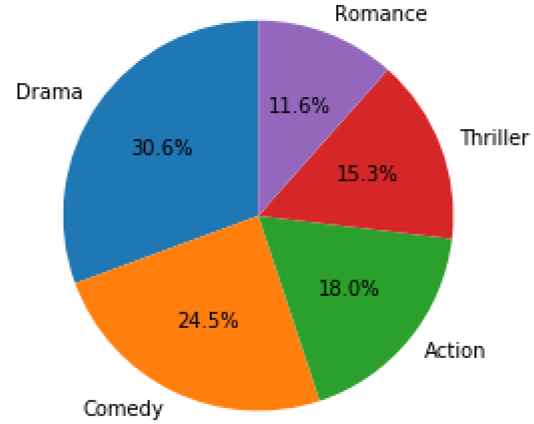
Percentage of movies in each genre in 1991



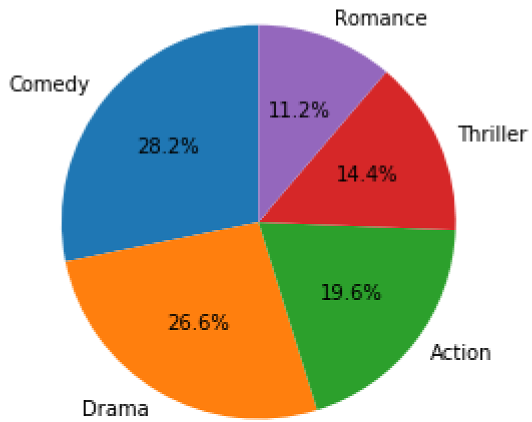
Percentage of movies in each genre in 1992



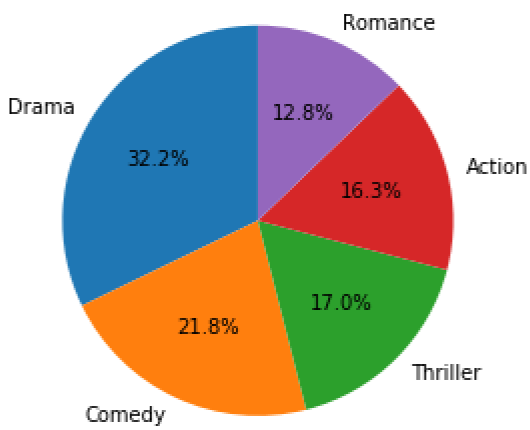
Percentage of movies in each genre in 1993



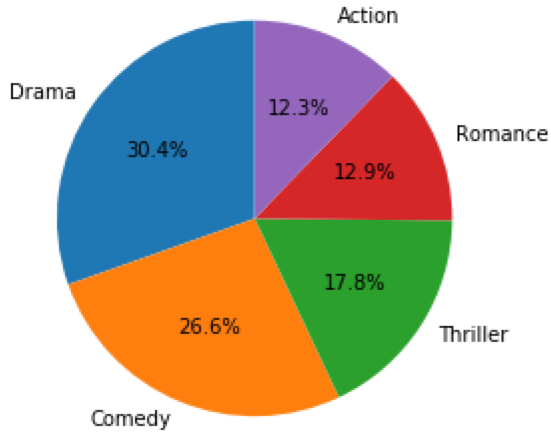
Percentage of movies in each genre in 1994



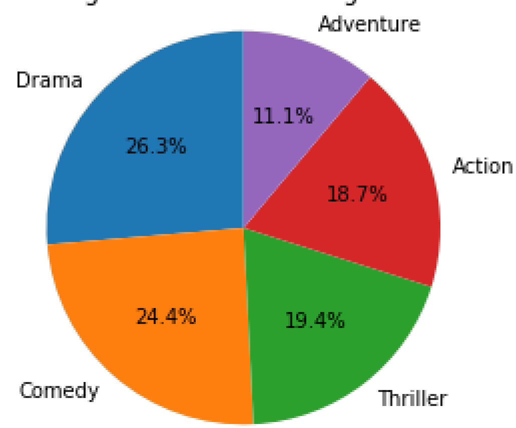
Percentage of movies in each genre in 1995



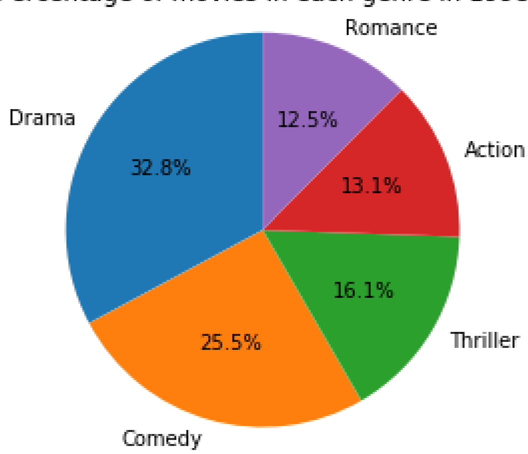
Percentage of movies in each genre in 1996



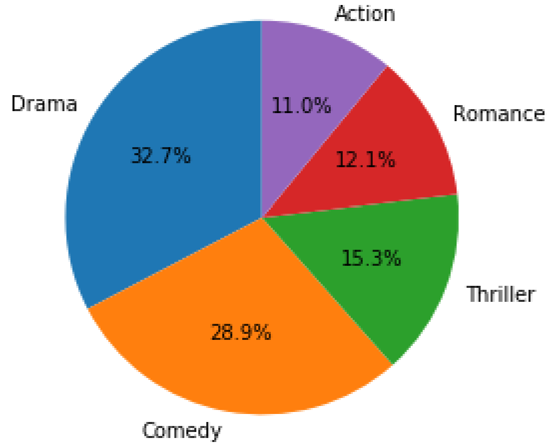
Percentage of movies in each genre in 1997



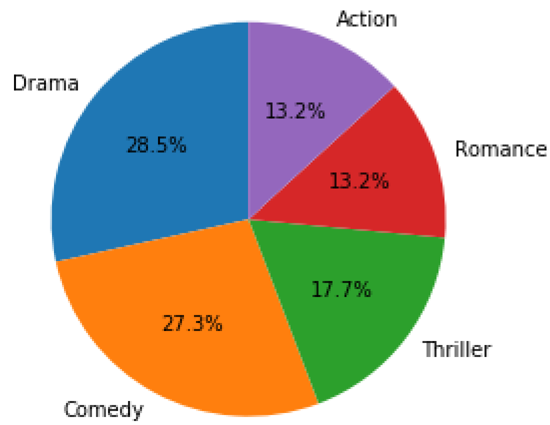
Percentage of movies in each genre in 1998



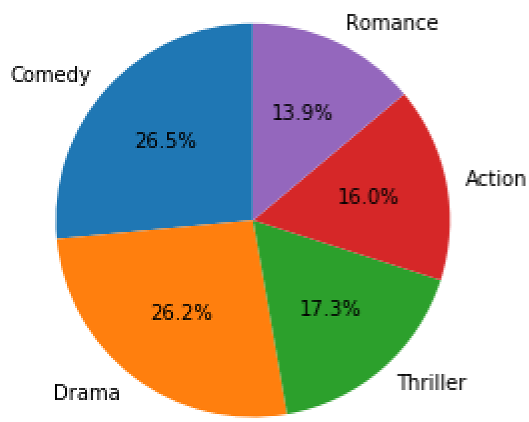
Percentage of movies in each genre in 1999

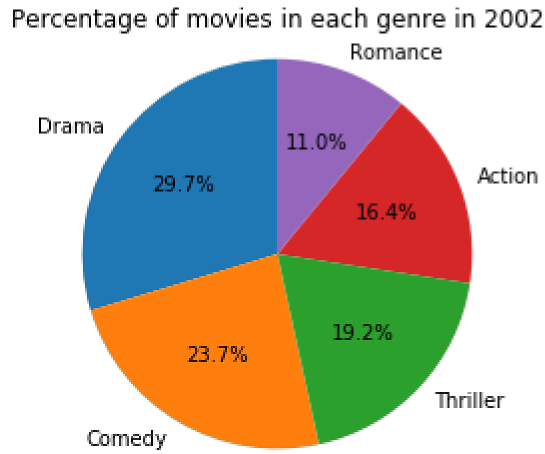


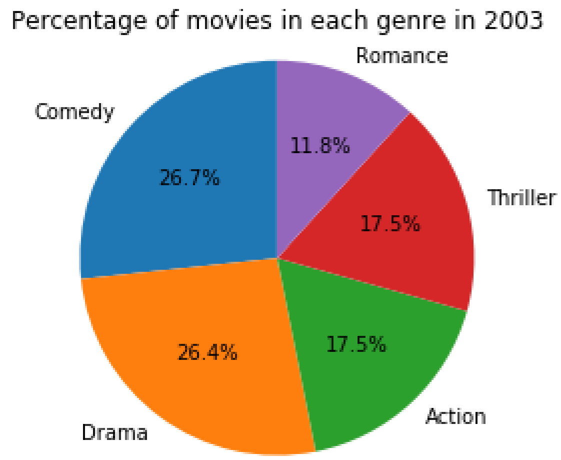
Percentage of movies in each genre in 2000

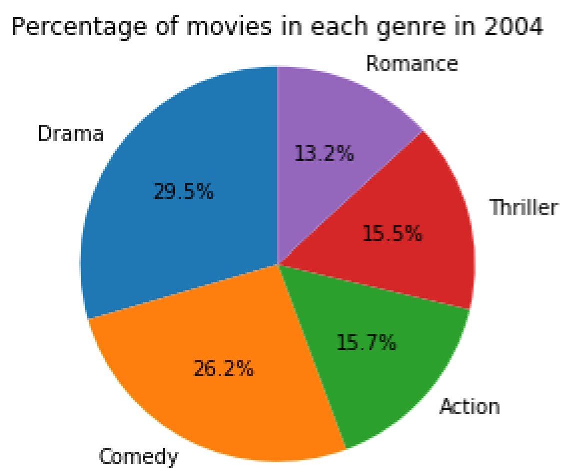


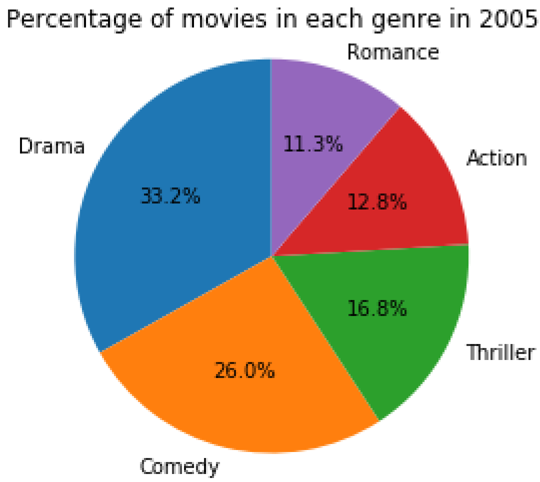
Percentage of movies in each genre in 2001

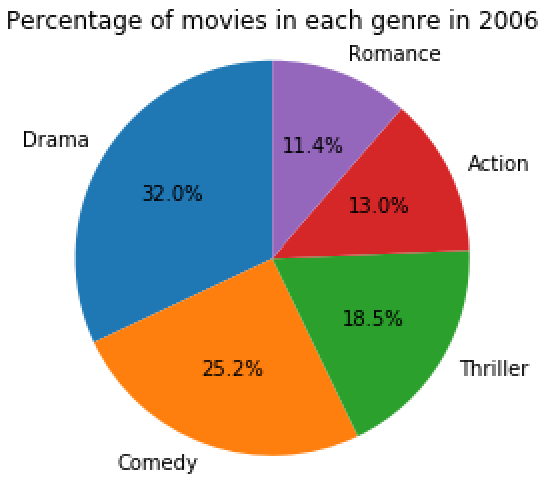


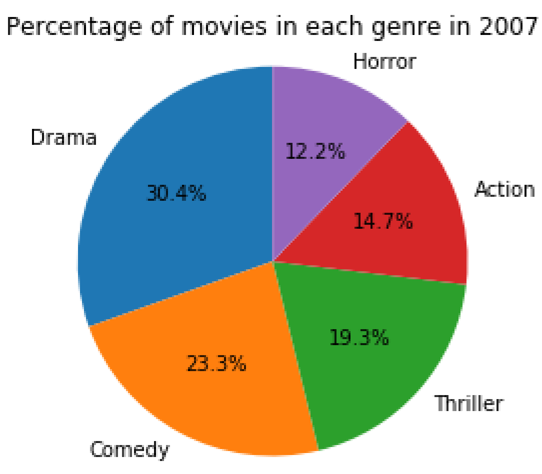

Percentage of movies in each genre in 2002
Romance
11.0%
Action
16.4%
Thriller
19.2%
Comedy
23.7%
Drama
29.7%

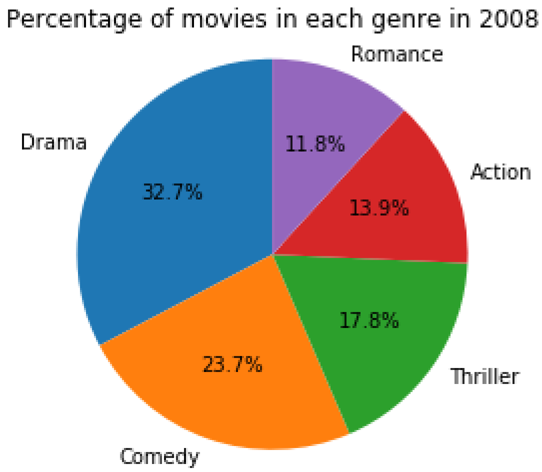

Percentage of movies in each genre in 2003
Romance
11.8%
Thriller
17.5%
Action
17.5%
Drama
26.4%
Comedy
26.7%

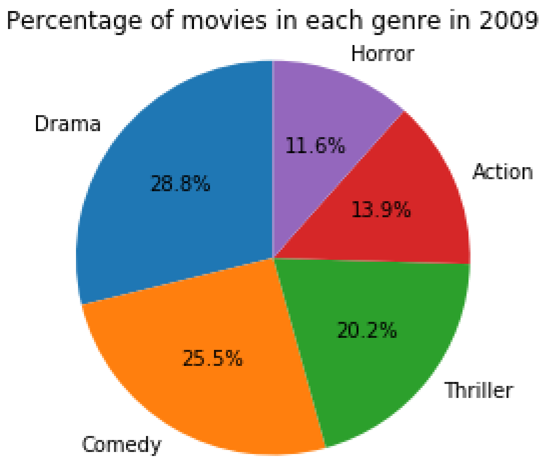

Percentage of movies in each genre in 2004
Romance
13.2%
Thriller
15.5%
Action
15.7%
Comedy
26.2%
Drama
29.5%

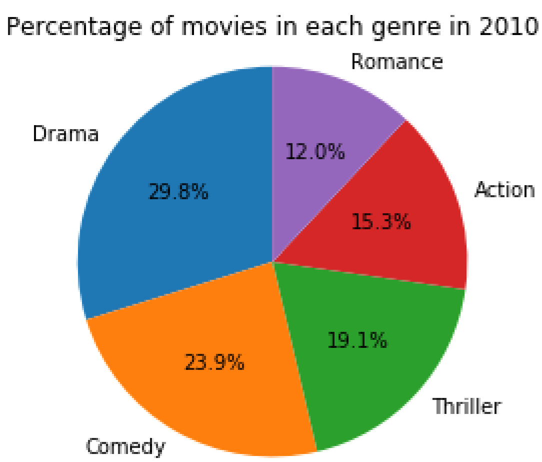

Percentage of movies in each genre in 2005
Romance
Drama
Action
11.3%
33.2%
12.8%
16.8%
Thriller
26.0%
Comedy


Percentage of movies in each genre in 2006
Romance
Drama
Action
11.4%
32.0%
13.0%
18.5%
Thriller
25.2%
Comedy

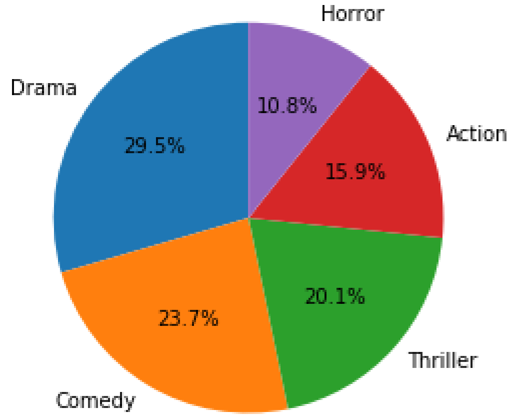

Percentage of movies in each genre in 2007
Horror
Drama
Action
12.2%
30.4%
14.7%
19.3%
Thriller
23.3%
Comedy


Percentage of movies in each genre in 2008
Romance
Drama
11.8%
Action
32.7%
13.9%
17.8%
23.7%
Thriller
Comedy

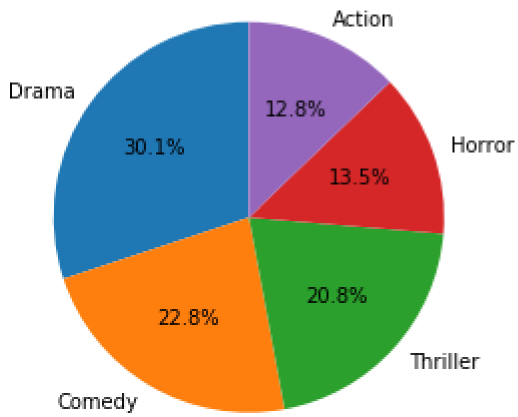

Percentage of movies in each genre in 2009
Horror
Drama
11.6%
Action
28.8%
13.9%
20.2%
25.5%
Thriller
Comedy


Percentage of movies in each genre in 2010
Romance
Drama
12.0%
Action
29.8%
15.3%
19.1%
23.9%
Thriller
Comedy

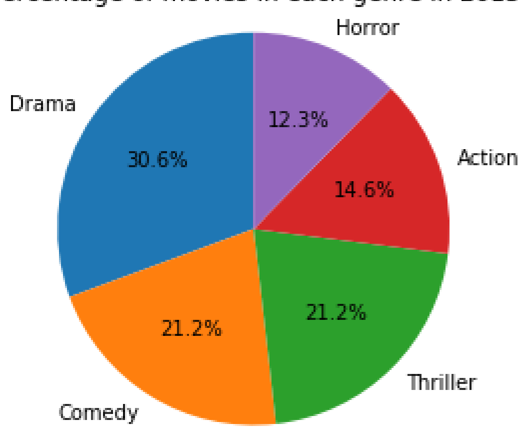
Percentage of movies in each genre in 2011

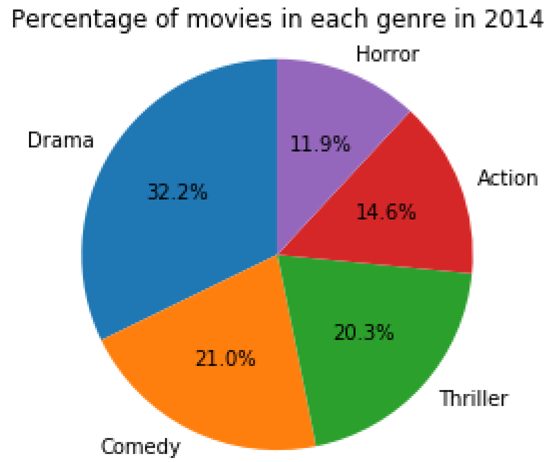


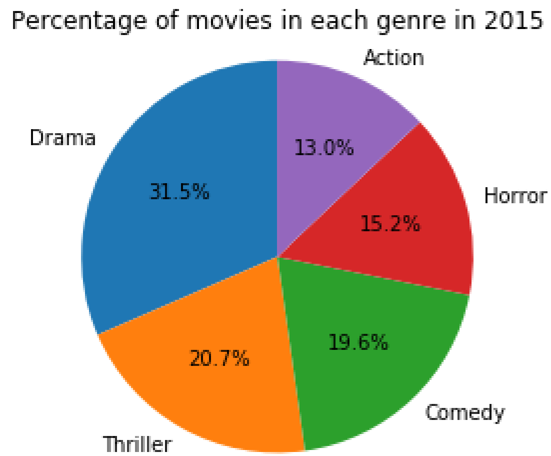
Percentage of movies in each genre in 2012



Percentage of movies in each genre in 2013




Percentage of movies in each genre in 2014
Horror
11.9%
Drama
32.2%
Action
14.6%
20.3%
21.0%
Thriller
Comedy


Percentage of movies in each genre in 2015
Action
Drama
13.0%
31.5%
Horror
15.2%
19.6%
20.7%
Comedy
Thriller

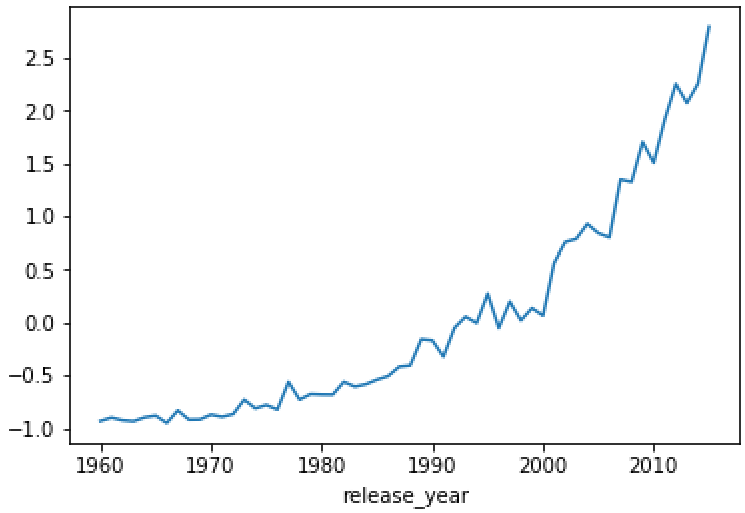
In [277]:
```python
def standardize(x):
    x_standardized = (x - x.mean()) / x.std(ddof = 0)
    return x_standardized

# standardize popularity
std_pop =  standardize(data['popularity'])

# Exploring profit generated by the movies
profit = data['revenue'].sub(data['budget'], axis=0)

profit_temp = {'id': data['id'],
               'revenue': data['revenue'].values,
               'profits_per_movie': profit.values,
               'release_year': data['release_year'].values,
               'popularity': data['popularity'].values,
               'std_popularity': std_pop.values,
               'budget': data['budget'].values
                #'genres': data['genres'].values
              }

profitable_std_movies = pd.DataFrame(profit_temp)
profitable_std_movies.head()
```

Out[277]:

	budget	id	popularity	profits_per_movie	release_year	revenue	std_popu
0	150000000	135397	32.985763	1363528810	2015	1513528810	32.33483
1	150000000	76341	28.419936	228436354	2015	378436354	27.76963
2	110000000	262500	13.112507	185238201	2015	295238201	12.46433
3	200000000	140607	11.173104	1868178225	2015	2068178225	10.52520
4	190000000	168259	9.335014	1316249360	2015	1506249360	8.687366

In [279]:
```python
def grouping_data(x, column_name):
    return x.groupby(column_name)

# group data by release years
grouped_data_by_years = grouping_data(data, 'release_year')
grouped_profits_by_years = grouping_data(profitable_std_movies, 'release_year'
)

# standardize profits of movies per year
standardize(grouped_profits_by_years.sum()['profits_per_movie']).plot()
```

Out[279]: <matplotlib.axes._subplots.AxesSubplot at 0x5d0c1a4b38>

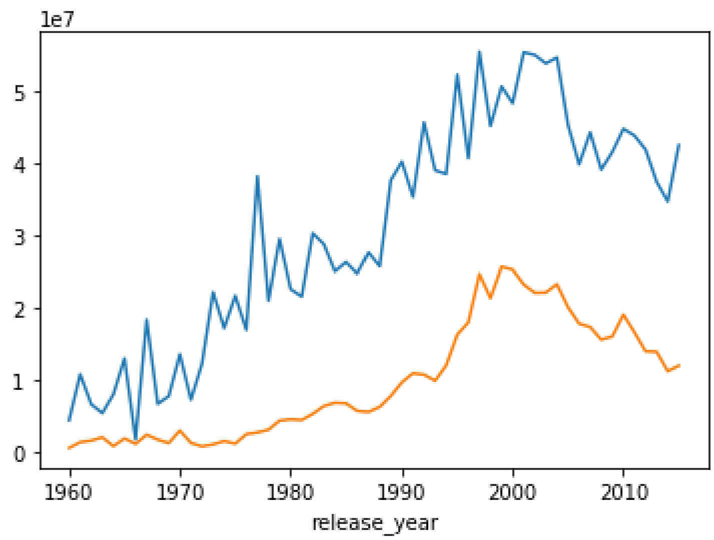


We can visualize from the above plot that the profits made by the movie in the 2010-2015 years time period are lot more than the profits made by the movies earlier.

In [280]:
```
# checking whether the revenue collected for each movie is always higher than
 its budget or not
def mean_of_grouped_data(x, column_name):
    return x.mean()[column_name]

per_year_revenue = mean_of_grouped_data(grouped_data_by_years, 'revenue')
per_year_budget = mean_of_grouped_data(grouped_data_by_years, 'budget')

# finding correlation between revenue and budget per year
relation = (np.corrcoef(per_year_revenue, per_year_budget))[0, 1]
print("\nCorrelation between revenue and budget (of the movies per year): {}".
format(relation))

per_year_revenue.plot()
per_year_budget.plot()
```

```
Correlation between revenue and budget (of the movies per year): 0.9059874665
884495
```

Out[280]: `<matplotlib.axes._subplots.AxesSubplot at 0x5d0caffb38>`



As the coorelation between revenue and budget (per year) is approx. 0.9. It depicts that both the budget and revenue (variables) are strongly correlated. So, it might be possible that the movie with higher budget also has higher movie revenues. But as the data is not cleaned yet so we cannot assure anything.

We can also see from this plot that the revenue has been mostly higher than the budget of the movie but this does not suggest that the movies will almost always be profitable as in some cases, the revenue and the budget have also not been reported (as discussed earlier). So, it may be a possibility that some of theses movies could have faced losses.

In [325]:
```
# sort the table in descending order according to standardized popularity of t
he movies
profitable_std_movies = profitable_std_movies.sort_values(by='std_popularity',
 axis='index')
profitable_std_movies.head()
```

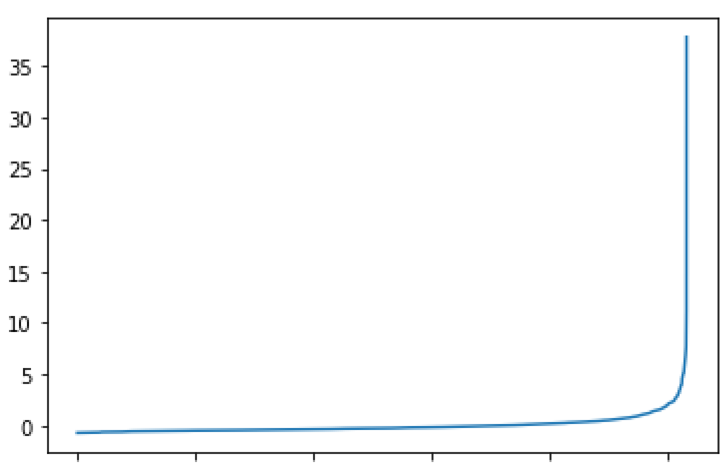
Out[325]:

	budget	id	popularity	profits_per_movie	release_year	revenue	std_popular
6181	0	18729	0.000065	0	1985	0	-0.646286
9977	0	32082	0.000188	0	1971	0	-0.646163
6080	0	174323	0.000620	0	2013	0	-0.645731
6551	0	31329	0.000973	0	2005	0	-0.645378
6961	0	15412	0.001115	0	2006	0	-0.645236

In [368]:
```
def correlation(x, y):
    std_x = standardize(x)
    std_y = standardize(y)
    return (std_x * std_y).mean()

def bool_index(x, column_name):
    return x[column_name].values != 0

index = bool_index(data, 'budget')
popular_vs_budget = correlation(data['budget'].values[index],
                                data['popularity'].values[index])
print(popular_vs_budget)

temp = pd.Series(standardize(profitable_std_movies['popularity'].values[index
]),
                 index=[standardize(profitable_std_movies['budget'].values[ind
ex])])
temp.plot()
```

```
0.479958191675
```

Out[368]: <matplotlib.axes._subplots.AxesSubplot at 0x5d15615588>



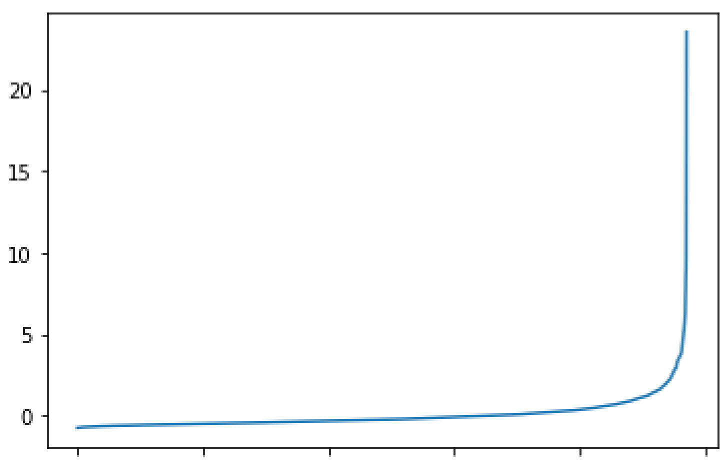
From the above correlation, we can observe that the value 0.48 approx. is positive which depicts a strong relaitonship between budget and popularity of the movie.

Also, the graph shows us the same thing that when the budget (along the x-axis) of the movies increases, the popularity of the also increases. But this graph can be also misleading as the testing values are reduced to half in this case.

In [369]:
```
index = bool_index(profitable_std_movies, 'revenue')

popular_vs_rev = (np.corrcoef((profitable_std_movies['revenue'].values)[index
],
                              (profitable_std_movies['popularity'].values)[ind
ex]))[0, 1]
print(popular_vs_rev)
temp = pd.Series(standardize(profitable_std_movies['popularity'].values[index
]),
                 index=[standardize(profitable_std_movies['revenue'].values[in
dex])])
temp.plot()
```

```
0.629315667894
```

Out[369]: `<matplotlib.axes._subplots.AxesSubplot at 0x5d153e9828>`



So, from the above correlation values 0.63 approx., between variables revenue and std_popularity, we can see that it is close to +1. Hence, we can see that there is a some STRONG relation between revenue and popularity. Because the Correlation value is > 0, which suggests that if one variable increases with certain margin the other will also increase with approx. similar margin as well.

We can say that revenue and popularity are closely and positively correlated.

In [371]:
```
index = bool_index(profitable_std_movies, 'profits_per_movie')

popular_vs_profit = (np.corrcoef((profitable_std_movies['profits_per_movie'].v
alues)[index],
                                 (profitable_std_movies['popularity'].values)[
index])
                    )[0, 1]
print(popular_vs_profit)
```

```
0.615919873113
```

Here, we can see that the variables profits_per_movie and std_popularity are also positively correlated (close to +1) which implies a STRONG relationship between both the variables. So, it may be a possibility that if a movie is more popular it may be more profitable also or vice versa. But not in all cases.

SUMMARY

1. The most popular genre from year to year is DRAMA (with a total of 17.6% drama movies) followed by Comedy (14.1%) as depicted by the plots above.
2. properties/attributes of the movies which are more popular:
   - We could see that some of the values in the budget (column) were 0. So, we did not include those 0 values for finding a similarity in our data. (Data Cleaning)
   - We found the similarity in our data by using Pearson's coefficient.
   - by the correlation value ablove, we can say that the budget of the movies contributes to the popularity of the movie at the box office, but 0.48 value is not a strong evidence to say so. So, it may or may not be the factor in some cases.
   - Similarly, we can also say that the revenue collected and the profit per movie can also be the factors or reasons towards the popularity of the movie as the correlation values for these variables with popularity variable are also positive and above 0.60.
   - We are **tentatively** concluding above points but we don't have any concrete results as the correlation values are positive but not so close to +1.