<How are movies from different times appreciated by todays audience?>

<Max Muhle>

Dataset(s)

- IMDB Movie Dataset
  - Movies.csv
  - Ratings.csv

Motivation

I want to analyze how movies from different times (data is roughly from 1900 to 2014) are rated by todays viewers in an online database. This will provide valuable insights into the audiences taste and preferences, allowing for a more targeted making of modern films. In this way, there will be better movies for everyone.

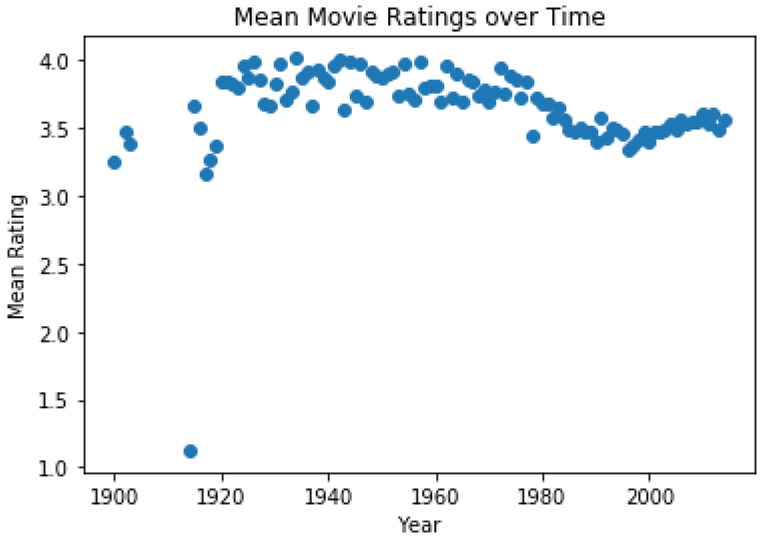
Research Question(s)

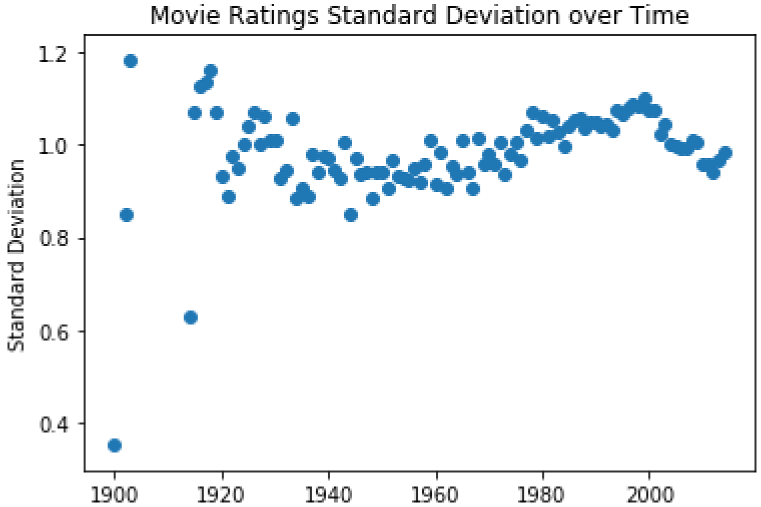
Are older movies (on average) higher or lower rated than modern times movies?

Is there a general trend?

Findings

- Movies before ca. 1960 are higher rated than more modern movies
- There is a greater distribution in movie ratings at the beginning of 20th century
    - This distribution declines with times, movie ratings becoming more similar
- There was a low point in movie ratings right at the end of the 20th century
- Standard deviation of ratings has been rapidly declining in the last years, indicating more similar movies


Mean Movie Ratings over Time


Movie Ratings Standard Deviation over Time

Acknowledgements

I have no feedback to share.

References

No references

In [1]:

```
# Import all necessary libraries
import pandas as pd
import numpy as np
import matplotlib as mp
%matplotlib inline
```

In [2]:

```
# Research question: Are older movies higher rated than more current ones?
```

In [3]:

```
# Strip movies to relevant info and add new column
movies = pd.read_csv('./movielens/movies.csv', sep=',')
movies = movies.drop(['genres'], axis = 1)

#extracting the movie year from the title
movies['title'] = movies['title'].str.replace('"','')
movies['title'] = movies['title'].str.replace(' ','')
movies['title'] = movies['title'].str.replace('-','')

movies['year'] = movies['title'].str[-5:-1]
movies['numeric']  =  movies['year'].str.isnumeric()

movies = movies.drop(['title'], axis = 1)

movies = movies[movies.numeric == True]
movies.head()
```

Out[3]:

```
   movieId  year  numeric
0        1  1995     True
1        2  1995     True
2        3  1995     True
3        4  1995     True
4        5  1995     True
```

In [4]:

```
# Strip ratings to relevant info
ratings = pd.read_csv('./movielens/ratings.csv', sep=',', usecols=[1, 2])
#limiting rows here, due to memory problems
ratings = ratings.loc[0:2000000, :]
ratings.head()
```

Out[4]:

```
   movieId  rating
0        2     3.5
1       29     3.5
2       32     3.5
3       47     3.5
4       50     3.5
```

In [5]:

```
# Prepare variables for FOR loop
current_ratings = pd.DataFrame()
results = pd.DataFrame()
current_year = int(movies['year'].min())

# Loop over the number of years. get the year and the mean rating and ratings stdde
v for each year

for i in range (0,(int(movies['year'].max())-int(movies['year'].min()))):

    # get current year
    results.loc[i,'year'] = current_year

    # get ids of movies of the current year
    current_IDs = movies[movies.year == str(current_year)]
    current_IDs = current_IDs.drop(['numeric'], axis = 1)
    current_IDs = current_IDs.drop(['year'], axis = 1)

    # get ratings of movies of current year
    temp = ratings.loc[ratings['movieId'].isin(current_IDs.movieId)]
    temp = temp.drop(['movieId'], axis = 1)

    # save the results
    results.loc[i,'mean'] = temp.rating.mean()
    results.loc[i,'STD']  = temp.rating.std()

    #increment the year
    current_year = current_year + 1
```

In [6]:

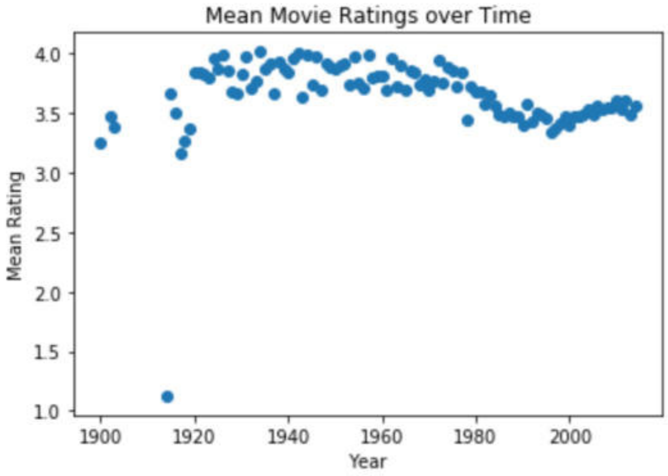
```
#Clean results for years that have no data
results = results.dropna(axis=0, how='any')
```

In [7]:

```
#Plot results
mp.pyplot.scatter(results['year'],results['mean'])
mp.pyplot.title('Mean Movie Ratings over Time')
mp.pyplot.xlabel('Year')
mp.pyplot.ylabel('Mean Rating')
```

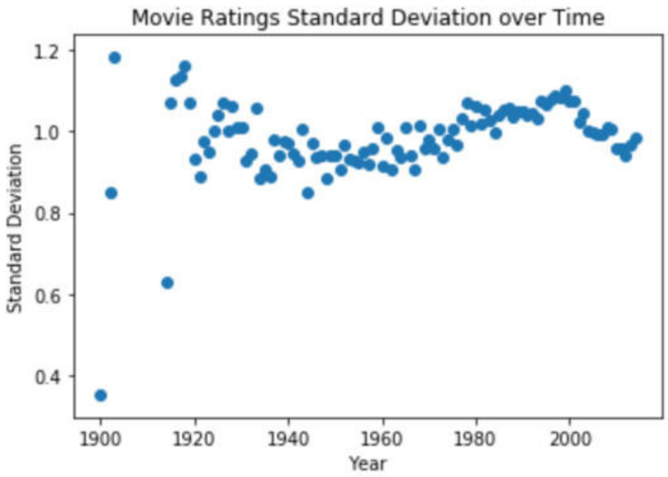
Out[7]:

```
Text(0, 0.5, 'Mean Rating')
```



```
In [8]:  mp.pyplot.scatter(results['year'],results['STD'])
         mp.pyplot.title('Movie Ratings Standard Deviation over Time')
         mp.pyplot.xlabel('Year')
         mp.pyplot.ylabel('Standard Deviation')

Out[8]:  Text(0, 0.5, 'Standard Deviation')
```



```
In [ ]:
```