# Genre Differences in User Movie Ratings

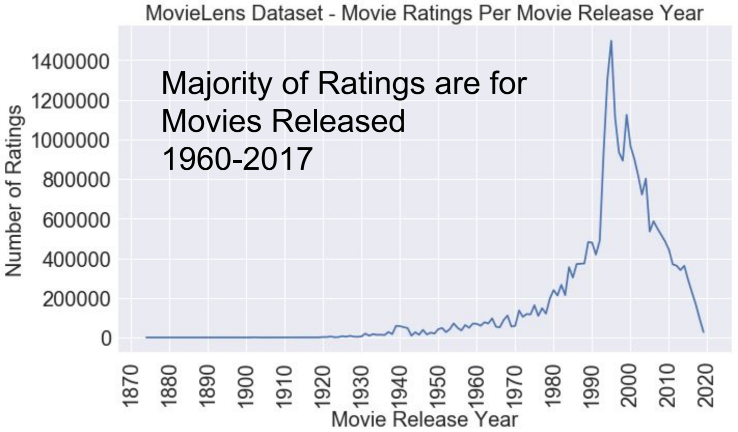
Paul Coster

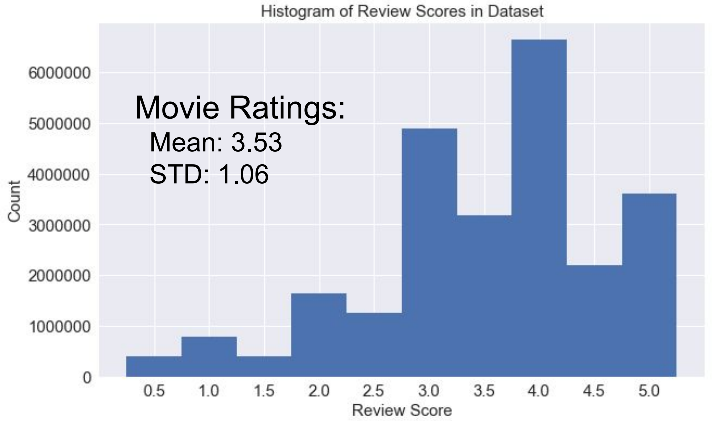
# Movielens Dataset:

This dataset is a selection of movie ratings from the 'MovieLens' movie recommendation service. The dataset contains:

- **160K Users**
- **25MM Ratings**
- **62K Movies, with 19 Genre Tags**
- **1MM 'Sentiment Tags'**
- **Ratings created Jan 1995 to Nov 2019**

For the dataset and more information about it:
https://grouplens.org/datasets/movielens/



# Motivation

In this analysis, I was looking to determine the differences in movie ratings between genres. Some genres may, on average, be more highly or poorly reviewed than others, which could be for a variety of reasons e.g.:

- Highly-rated genres may be watched / rated primarily by fans of the genre, who will be more likely to give the movie a positive review.
- Poorly-rated genres might be watched by a wider audience, with a resulting broader distribution of ratings, resulting in a lower average score. Alternatively, poor ratings may be caused by a genre containing a large number of poor quality films.

Understanding the differences in ratings of different movie genres may give insights such as:

- Better movie recommendations for users of a movie service
- Targeted movie advertising to increase review scores
- Aid a producer in when deciding on movie projects to proceed with, desiring a critical success

# Research Question(s)

1. **Do certain movie genres tend to get better critical reviews, on average?**
   - **Are these genre's mean review scores consistently above/below average, every year?**
   - **Do these genres get greater or fewer reviews per film on average?**

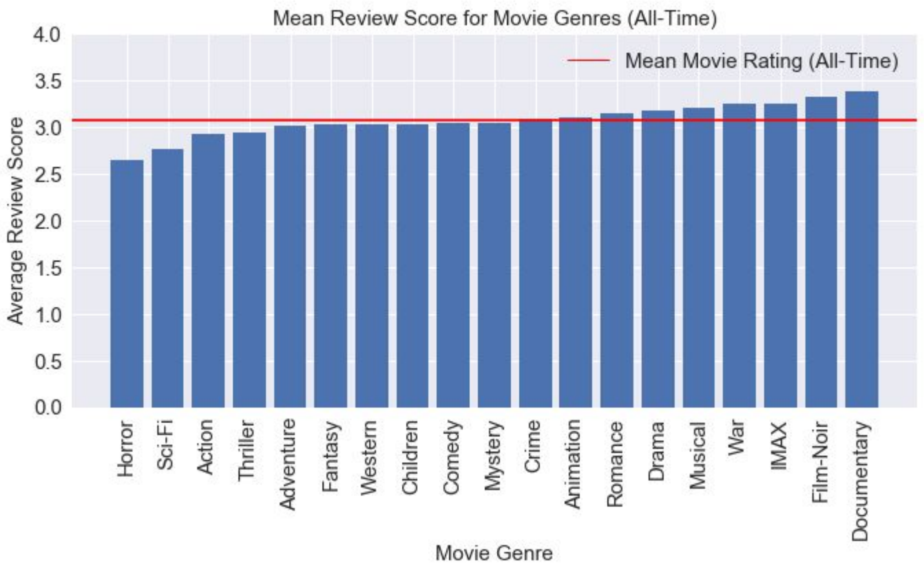
2. **For genres that are highly or poorly rated on average, does their distribution of mean title review scores look significantly different?**

3. **Are movies tagged with more genres on average rated better or worse on average than movies with fewer genres?**

# 1.a. Best/Worst Rated Genres:

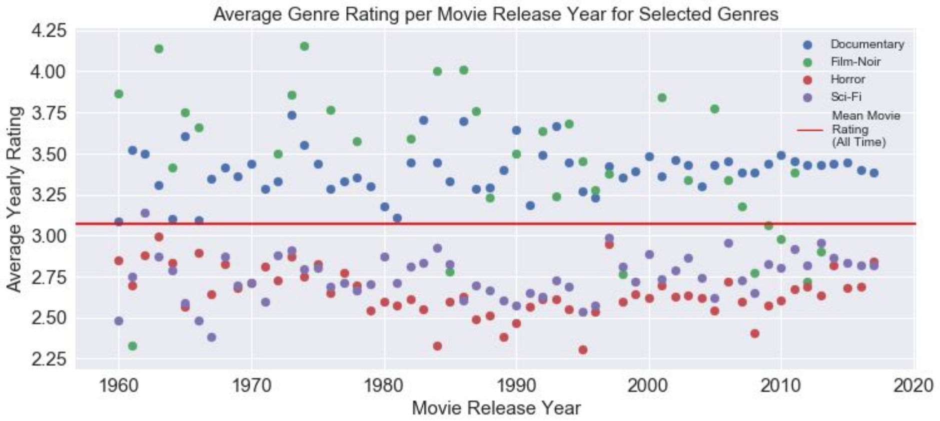
**All-Time Genre Data:**

- **Horror** and **Sci-Fi** have the worst average genre ratings, **2.65** and **2.76** respectively.
- **Film-Noir** and **Documentary** have the best average genre ratings, **3.32** and **3.38** respectively.
- Mean Movie Rating is **3.07**.




**Are these genres consistently above/below average over all release years?**

- **Documentary** genre consistently scores above the average.
- **Film-Noir** generally above average
- **Horror and Sci-Fi** genres all below average, with one exception (**SciFi** 1962)



# 1.b. Reviews per Title, by Genre:

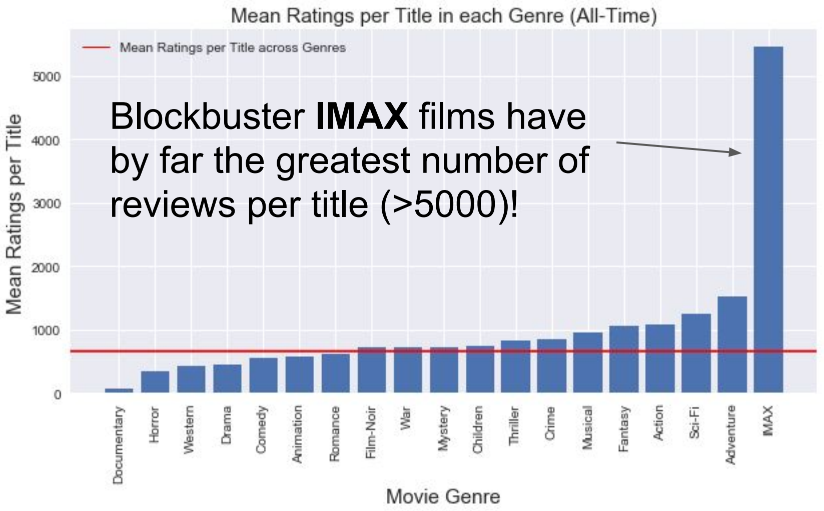
**Do these genres get greater or fewer reviews per film on average?**

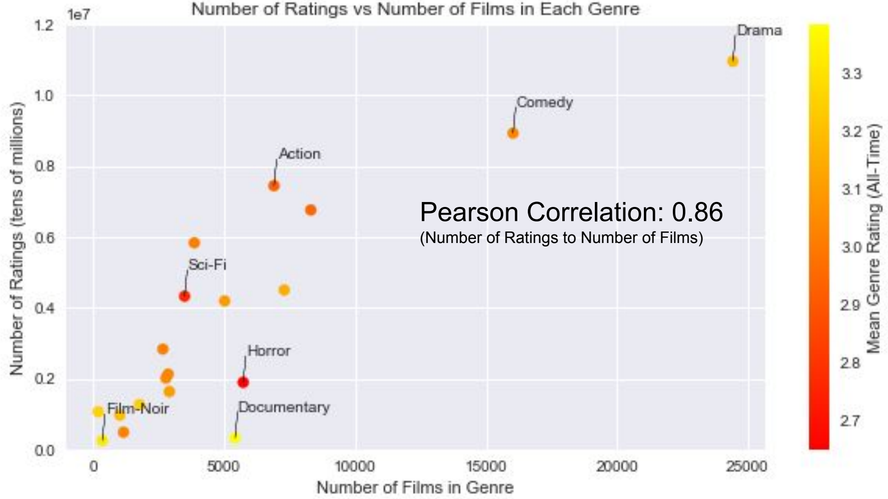
- **Documentary** films reviewed by very niche audiences - fewer than **60** reviews per title.
- However **Horror** films have the next fewest reviews per title ~**330**.
- **Film Noir** has an average number of reviews per title (~**700**), while **Sci-Fi** is above the mean (~**1200**).

**Strong correlation (0.86) between number of ratings and number of titles in each genre.**

**No correlation (0.14) between the mean ratings per title in each genre and the mean rating of each genre.**

**Perhaps Sci-Fi and Horror rate poorly on average due to more split reviewer opinions, or perhaps there are a larger number of very poorly reviewed films in these genres?**



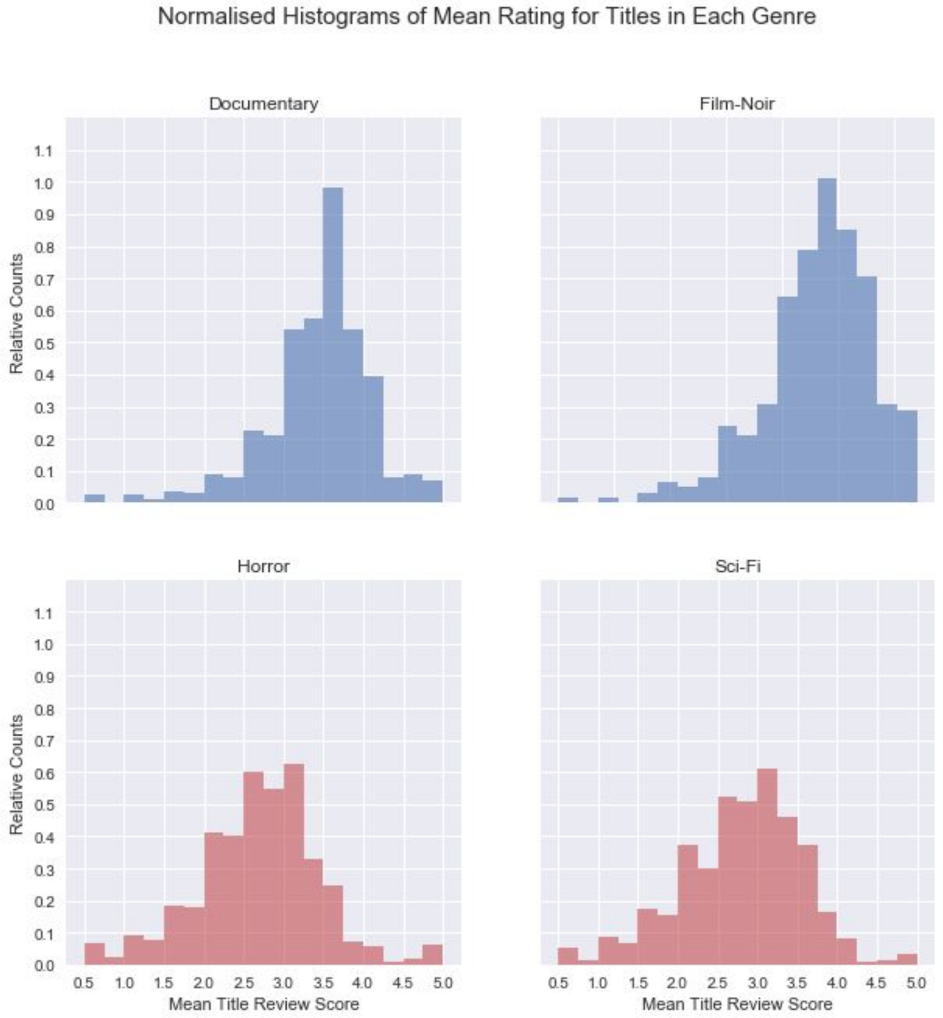



# 2. Ratings Distribution in Genres:

**How do the mean title review scores differ between highly and poorly rated genres?**

- High proportion of **Documentary** and **Film-Noir** genre titles with a **mean rating above 3.5.**
- Both these highly-scoring genres have relatively few titles with a mean rating below 3.0.
  - Evidence of higher average movie quality, or perhaps that these more niche genres tend to be watched by fans of the genre, documentary topic, resulting in high ratings?

- **Horror** and **Sci-Fi**, have much broader distributions of mean title ratings.
- Both these genres have relatively few titles with scores above 3.5, and far more titles with scores of 1.0 to 2.5.
  - Evidence that these genres contain a higher-proportion of low-quality titles?



# 3. Listed Genres per Title:

**Number of Listed Genres per Title:**

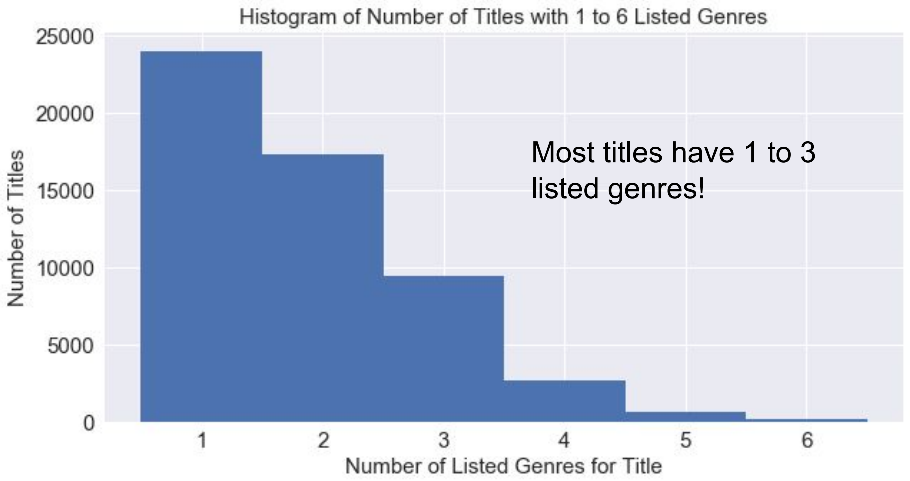
- The **vast majority (>93%)** of titles in the dataset have **1 to 3 listed genres**.

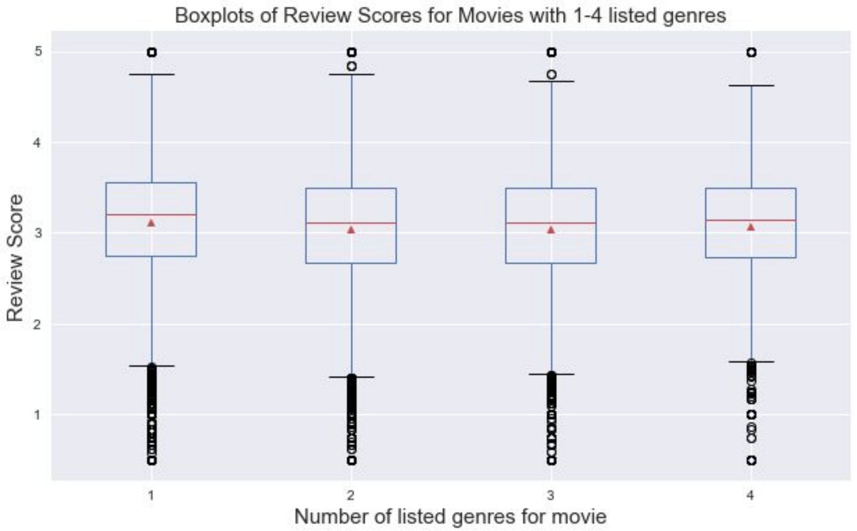
- <2% of titles have 5 or more listed genres.

**Are titles with fewer genres rated better or worse than those with more genres?**

- Separate boxplots for titles with 1-4 listed genres are very similar!
- Mean review scores are also very similar (red triangles on box plots).

**Number of genres does not seem to affect review scores! Clearly both good and bad films can be made with few or many genres!**



# Summary

The **MovieLens** dataset was analysed to determine the differences in review scores for titles with different listed genres.

- Movies with **Documentary** and **Film-Noir** genres listed have the highest mean review score.
- Movies with **Horror** and **Sci-Fi** genres listed have the lowest mean review score.

- The mean title score distributions for the two highest scoring genres are very different to those of the two lowest scoring genres.
  - **Documentary** and **Film-Noir** are likely more niche genres, more likely to be watched and reviewed by fans of the genre or subject matter, their distributions skew towards high scores.
  - **Horror** and **Sci-Fi** genres possibly contain more examples of poor-quality films, resulting in a broader distribution of lower scores. They also might be watched and reviewed by a more general audience, such that not all review scores are likely to be favourable, bringing average review scores down.

- Movies with a single genre listed have a similar review score distribution to those with 2, 3 or 4 listed genres.

# Acknowledgements

Thanks to my wife Meg for listening to me talk through my ideas for this project!

# References

Matplotlib Documentation

Pandas Documentation

Stack Overflow(!)

# Week 6 Mini-Project

June 26, 2020

## 1 Week 6 Mini-Project: MovieLens Dataset

The MovieLens dataset (ml-25m) consists of around 25 million ratings (including 1 million 'tag' applications) for 62423 movies. The ratings were created by around 160 thousand users between Jan 1995 and November 2019.

Users were selected at random for the dataset from users on the MovieLens movie recommendation service, from users who had rated at least 20 movies. Each user is represented by an anonymous ID, with no demographic information included.

The dataset is available from https://grouplens.org/datasets/movielens/

When downloading the dataset, there is an MD5 checksum available to verify the dataset contents, e.g. on linux:

$md5sum ml-25m.zip; cat ml-25m.zip.md5

## 2 Exploring the Dataset:

The movielens dataset primarily consists of 3 csv files:

- ratings.csv - each row is a timestamped movie rating by a single user
  - columns - userId, movieId, rating, timestamp
- movies.csv - each row is a movie and its genre information as a pipe-separated list
  - columns - movieId, title, genres
- tags.csv - each row is a tag given by a user for a movie as part of a review, with a timestamp
  - columns - userId, movieId, tag, timestamp

[1]:
```
#import required libraries and load in the datasets
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
import seaborn as sns
%matplotlib inline
plt.style.use('seaborn')

# Load in the datasets from csv files:
ratings = pd.read_csv('movielens/ratings.csv', sep=',')
```

```
movies = pd.read_csv('movielens/movies.csv', sep=',')
tags = pd.read_csv('movielens/tags.csv', sep=',')
```

[2]:
```
print(ratings.shape)
ratings.head()
```

```
(25000095, 4)
```

[2]:
```
   userId  movieId  rating   timestamp
0       1      296     5.0  1147880044
1       1      306     3.5  1147868817
2       1      307     5.0  1147868828
3       1      665     5.0  1147878820
4       1      899     3.5  1147868510
```

[3]:
```
print(movies.shape)
movies.head()
```

```
(62423, 3)
```

[3]:
```
   movieId                               title  \
0        1                    Toy Story (1995)
1        2                      Jumanji (1995)
2        3             Grumpier Old Men (1995)
3        4            Waiting to Exhale (1995)
4        5  Father of the Bride Part II (1995)

                                        genres
0  Adventure|Animation|Children|Comedy|Fantasy
1                   Adventure|Children|Fantasy
2                               Comedy|Romance
3                         Comedy|Drama|Romance
4                                       Comedy
```

[4]:
```
print(tags.shape)
tags.head()
```

```
(1093360, 4)
```

[4]:
```
   userId  movieId               tag   timestamp
0       3      260           classic  1439472355
1       3      260            sci-fi  1439472256
2       4     1732       dark comedy  1573943598
3       4     1732    great dialogue  1573943604
4       4     7569  so bad it's good  1573943455
```

We can see from the above that the datasets are as described above, around 25 million reviews of 62 thousand movies with 1 million tags in reviews.

We can quickly get an idea of the average review score in the dataset using the describe() pandas method, and get an indication of the allowed review scores using the value_counts() method:

```
[5]: # Descriptive statistics:
     print(ratings['rating'].describe())
     ratings['rating'].value_counts()
```

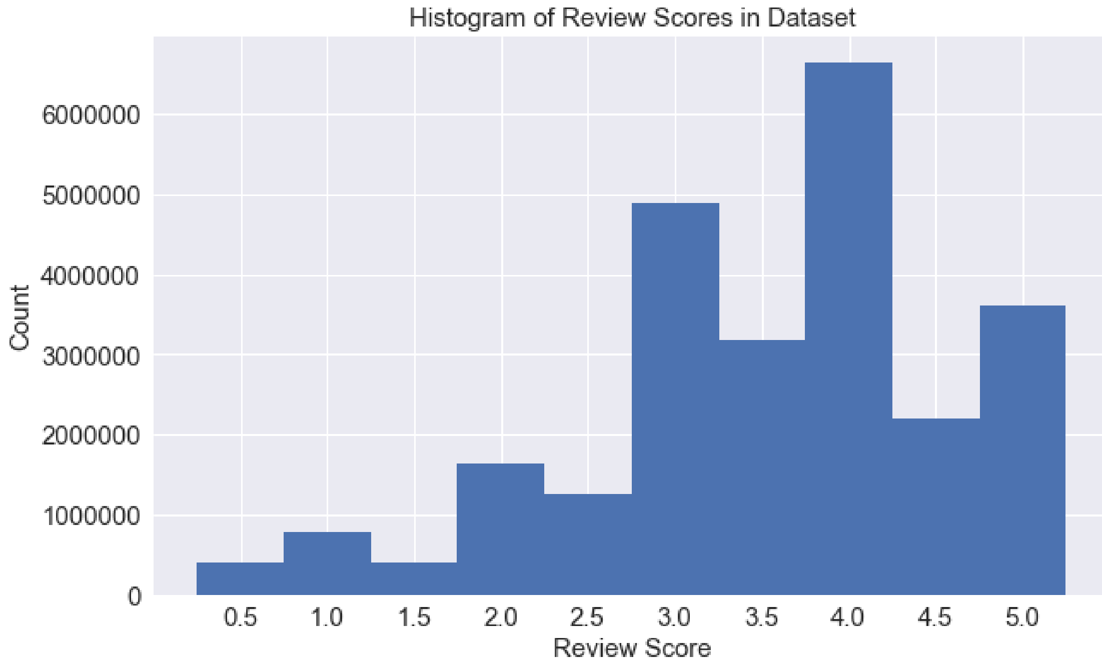
```
count    2.500010e+07
mean     3.533854e+00
std      1.060744e+00
min      5.000000e-01
25%      3.000000e+00
50%      3.500000e+00
75%      4.000000e+00
max      5.000000e+00
Name: rating, dtype: float64
```

```
[5]: 4.0    6639798
     3.0    4896928
     5.0    3612474
     3.5    3177318
     4.5    2200539
     2.0    1640868
     2.5    1262797
     1.0     776815
     1.5     399490
     0.5     393068
     Name: rating, dtype: int64
```

From the above we can see that the reviews score films from 0.5 to 5 stars in 0.5 increments. The mean review score is just over 3.5 stars, with the median also at 3.5 stars. However the most common review score (mode) is 4.0. This also shows that no ratings are outside this range (i.e. negative or greater than 5), which is a good starting point!

We can visualise the review distribution easily using a boxplot or histogram, here we will use a histogram:

```
[6]: # Histogram of Ratings column in Ratings dataframe
     plt.figure(figsize=[10,6])
     plt.hist(ratings['rating'], bins=np.arange(0.5,6.0, step=0.5)-0.25)
     plt.title('Histogram of Review Scores in Dataset', fontsize=15)
     plt.ylabel('Count', fontsize=15)
     plt.xlabel('Review Score', fontsize=15)
     plt.xticks(np.arange(0.5, 5.5, 0.5), fontsize=15)
     plt.yticks(fontsize=15)
     plt.show()
```




It is clear from above plot that the data has a left skew to it - a large number of higher ratings with a tail of lower ratings. This data in itself could raise other research questions e.g. are users more likely to submit a positive review for a negative review, which starts to enter into more of a psychological question!

In any case lets continue with the exploration - how many different unique review tags are in the tags dataset?

[7]:
```
# Value counts and number of unique entries for tags:
print(tags['tag'].value_counts().head(10))
tags['tag'].unique().shape
```

```
sci-fi               8330
atmospheric          6516
action               5907
comedy               5702
surreal              5326
based on a book      5079
twist ending         4820
funny                4738
visually appealing   4526
dystopia             4257
Name: tag, dtype: int64
```

[7]: (73051,)

From the information above we can see that there are around 73 thousand unique tags in the dataset. It is also clear that some of the tags are likely related to the genre of the film being reviewed, while others refer to aspects or themes of the film.

## 3 Research Question(s):

Here we will look at trends in Review Scores across Genres, where we can ask several questions such as:

1. Do certain movie genres tend to get better critical reviews, on average?
   - Follow Up: do these trends in popularity change over time?
2. For genres that are highly or poorly rated on average, does their distribution of review scores look significantly different?
   - I.e. do highly rated genres have a left skew with mostly high scoring reviews, do pooring scoring genres have a more central distribution, or perhaps a split distribution with peaks of low and high ratings, averaging to a lower score?
3. Do broader films covering more genres (>=3) on average score better or worse than films with fewer genres (<3)?

## 4 1. Do certain movie genres tend to get better critical reviews, on average?

Lets start by looking at this question. Firstly we will have to check for any issues with the dataset that we may have to correct before carrying out the analysis. This question will involve looking at the ratings and movies datasets:

[8]:
```
# Check for NULL in any rows of movies:
movies.isnull().any()
```

[8]:
```
movieId    False
title      False
genres     False
dtype: bool
```

[9]:
```
#Check for NULL in any rows of ratings:
ratings.isnull().any()
```

[9]:
```
userId       False
movieId      False
rating       False
timestamp    False
dtype: bool
```

No Null values in either dataset, which is a good start.

Now in order to look at review score vs. genre we are going to have to split up the 'genre' entry in the movies dataset,

[10]:
```
# Make copy of the movies table and split genres into a list of genres
movie_genres = movies.copy()
movie_genres['genres'] = movie_genres['genres'].str.split('|')
movie_genres.head()
```

[10]:
```
   movieId                               title  \
0        1                    Toy Story (1995)
1        2                      Jumanji (1995)
2        3             Grumpier Old Men (1995)
3        4            Waiting to Exhale (1995)
4        5  Father of the Bride Part II (1995)

                                              genres
0  [Adventure, Animation, Children, Comedy, Fantasy]
1                     [Adventure, Children, Fantasy]
2                                  [Comedy, Romance]
3                           [Comedy, Drama, Romance]
4                                           [Comedy]
```

[11]:
```
movie_genres.dtypes
```

[11]:
```
movieId     int64
title      object
genres     object
dtype: object
```

Lets make a dataframe linking individual genres to each movie, using each movies movieId.

Note that the below step takes quite a long time to run!

[12]:
```
# Lets make a dataframe that links individual genres to each movie via movie Ids:
movie_to_genres = pd.DataFrame(columns=['movieId', 'genre'])
for index, row in movie_genres.iterrows():
    movie_to_genres = movie_to_genres.append([pd.DataFrame([[row['movieId'],␣
 ↪genre]], columns=['movieId', 'genre']) for genre in row['genres']],␣
 ↪ignore_index=True)

movie_to_genres.head()
```

[12]:
```
  movieId      genre
0       1  Adventure
1       1  Animation
2       1   Children
3       1     Comedy
4       1    Fantasy
```

[13]:
```
# Check that toy story has its five genres listed - great!
movie_to_genres[movie_to_genres['movieId']==1]
```

[13]:
```
   movieId      genre
0        1  Adventure
1        1  Animation
2        1   Children
3        1     Comedy
4        1    Fantasy
```

Since creating this linking dataframe between genres and movies takes some time, lets save it so we can reuse it later:

[14]:
```
movie_to_genres.to_csv("movie_to_genres")
```

Lets check that the number of rows we have here make sense in the movie_to_genres dataframe, first lets add a column to movie_genres that counts the number of genres each film has, then we can sum that column to get the number of rows we should have:

[15]:
```
movie_genres['num_genres'] = movie_genres['genres'].apply(lambda x: len(x))
print(movie_to_genres.shape)
movie_genres['num_genres'].sum()
```

```
(112307, 2)
```

[15]:
```
112307
```

We have the expected number of rows so everything looks ok. Now we can use inner joins to add the average ratings and every genre for every film in the dataset. Lets also remove any films that have no genre listed:

[16]:
```
# Create table with average review score for every film
avg_ratings = ratings[['movieId', 'rating']].groupby('movieId', as_index=False).
 ↪mean()
# Merge average ratings table with movies table:
movie_avg_ratings = movies.merge(avg_ratings, on='movieId', how='inner')

# Remove movies with no genre listed
movie_avg_ratings = movie_avg_ratings[~movie_avg_ratings['genres'].str.
 ↪contains('no genres listed')]

# Remove the 'genres column from the dataframe'
del movie_avg_ratings['genres']

print(movie_avg_ratings.shape)
print(movie_avg_ratings['rating'].mean())
movie_avg_ratings.head()
```

```
(54479, 3)
3.073282369618579
```

[16]:

| | movieId | title | rating |
|---|---|---|---|
| 0 | 1 | Toy Story (1995) | 3.893708 |
| 1 | 2 | Jumanji (1995) | 3.251527 |
| 2 | 3 | Grumpier Old Men (1995) | 3.142028 |
| 3 | 4 | Waiting to Exhale (1995) | 2.853547 |
| 4 | 5 | Father of the Bride Part II (1995) | 3.058434 |

It looks like there are reviews for around 54 thousand unique films with at least one listed genre in the dataset.

Lets also extract the film year from the title to create a separate 'year' column for each film. A handful of the films do not have years listed next to their titles - we will remove these films from our analysis.

[17]:
```
movie_avg_ratings['year'] = movie_avg_ratings['title'].str.extract('.*\((.*)\).
↪*', expand=True)
movie_avg_ratings.dropna(inplace=True)
movie_avg_ratings.loc[18754, 'year'] = '1983'
movie_avg_ratings.loc[43293, 'year'] = '2006'
movie_avg_ratings = movie_avg_ratings[movie_avg_ratings['year'].str.len() == 4]
movie_avg_ratings.shape
```

[17]: (54351, 4)

We have dropped around 128 films from the dataset that did not have their year included in their title. Now with another inner join to movie_to_genres we can create a dataframe with a separate row for each genre of a film, ready to perform some groupby operations for plotting:

[18]:
```
movie_avg_rat_genres = movie_avg_ratings.merge(movie_to_genres, on='movieId',␣
↪how='inner')
# Convert year column to integer values:
movie_avg_rat_genres['year'] = movie_avg_rat_genres['year'].astype(str).
↪astype(int)
print(movie_avg_rat_genres.shape)
movie_avg_rat_genres.head(10)
```

(102167, 5)

[18]:

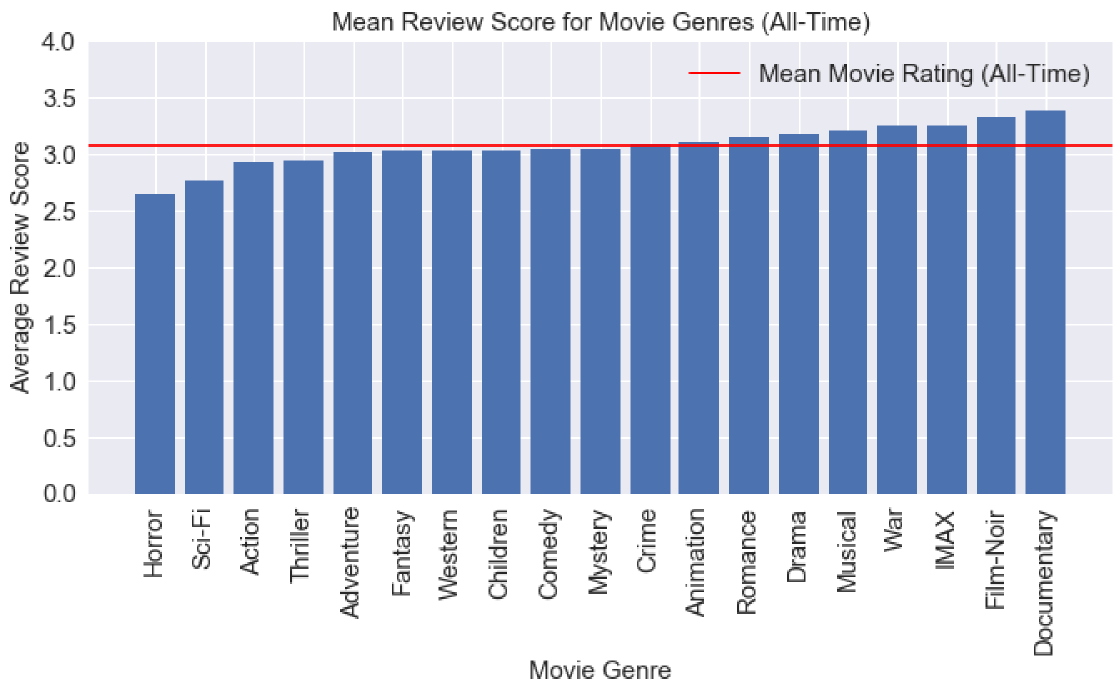
| | movieId | title | rating | year | genre |
|---|---|---|---|---|---|
| 0 | 1 | Toy Story (1995) | 3.893708 | 1995 | Adventure |
| 1 | 1 | Toy Story (1995) | 3.893708 | 1995 | Animation |
| 2 | 1 | Toy Story (1995) | 3.893708 | 1995 | Children |
| 3 | 1 | Toy Story (1995) | 3.893708 | 1995 | Comedy |
| 4 | 1 | Toy Story (1995) | 3.893708 | 1995 | Fantasy |
| 5 | 2 | Jumanji (1995) | 3.251527 | 1995 | Adventure |
| 6 | 2 | Jumanji (1995) | 3.251527 | 1995 | Children |
| 7 | 2 | Jumanji (1995) | 3.251527 | 1995 | Fantasy |
| 8 | 3 | Grumpier Old Men (1995) | 3.142028 | 1995 | Comedy |

```
9       3  Grumpier Old Men (1995)  3.142028  1995    Romance
```

With the datasets cleaned and merged we can now do some plotting, first lets see the average ratings for each genre of film across all years, using a bar plot

[19]:
```
all_time_genre_ratings = movie_avg_rat_genres.groupby('genre', as_index=False).
    ↪mean()
all_time_genre_ratings.sort_values('rating', inplace=True)
print(all_time_genre_ratings.head(19))
print('Mean rating across all genres: ', all_time_genre_ratings['rating'].mean())

plt.figure(figsize=[11, 5])
plt.bar(all_time_genre_ratings['genre'].values, all_time_genre_ratings['rating'])
plt.xticks(rotation=90)
plt.title('Mean Review Score for Movie Genres (All-Time)', fontsize=15)
plt.xlabel('Movie Genre', fontsize=15)
plt.ylabel('Average Review Score', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylim(0, 4)
# Plot horizontal line at average score of all genres
plt.axhline(color='r', y=all_time_genre_ratings['rating'].mean())
# Add legend for horizontal line
line = Line2D([0], [0], color='r', linewidth=1)
plt.legend([line], ['Mean Movie Rating (All-Time)'], fontsize=15)
plt.show()
```

```
          genre    rating         year
10       Horror  2.650763  1996.456529
15       Sci-Fi  2.761345  1995.777364
0        Action  2.925590  1996.359554
16     Thriller  2.945069  1998.892767
1     Adventure  3.010614  1990.547927
8       Fantasy  3.022389  1992.709774
18      Western  3.025755  1970.184256
3      Children  3.032964  1995.672956
4        Comedy  3.042288  1992.859870
13      Mystery  3.045802  1990.879410
5         Crime  3.092559  1989.542937
2     Animation  3.100958  1989.758680
14      Romance  3.147334  1989.127622
7         Drama  3.178987  1992.813994
12      Musical  3.199744  1975.496063
17          War  3.246515  1982.052542
11         IMAX  3.252121  2008.466667
9     Film-Noir  3.318142  1955.785100
6   Documentary  3.383110  2004.299852
Mean rating across all genres:  3.07273932628101
```




In the plot above we can see that most movie genres average ratings are fairly close to the average across all genres. Horror and Sci-Fi genre films in particular appear to suffer from lower than average scores, with ratings of 2.65 and 2.76 overall. On the other hand, Documentary and Film-Noir films tend to review better than average, with ratings of 3.38 and 3.32 overall. The mean rating across all genres is 3.07.

Lets analyse whether the genres that review well over all time have consistently high reviews over the years, or whether perhaps they have a few yeras of very high reviews bringing them above the average. Similarly we can look at whether the poorly reviewde genres have consistently poor reviews over the years, or just a few very bad years that have brought their average down.

Before we start looking at each genre over time, lets look at how many reviews there are for films that were made in each year in the dataset. For example how many reviews were for films made in 1936, 1950, 2003 etc.

[20]:
```
# Get list of unique movie titles from our cleaned dataset:
unique_movies = movie_avg_rat_genres[['movieId', 'title', 'year']].
 ↪drop_duplicates('movieId')
# Inner join with ratings
unique_movie_ratings = unique_movies.merge(ratings, on='movieId', how='inner')

# Groupby Year and count number of reviews each year
ratings_per_year = unique_movie_ratings.groupby('year', as_index=False).count()
ratings_per_year = ratings_per_year[['year', 'rating']]
ratings_per_year.columns = ['year', 'num_ratings']
ratings_per_year.sort_values('num_ratings', ascending=False).head()
```

```
ratings_per_year.sort_values('num_ratings', ascending=False).tail()
```

[20]:

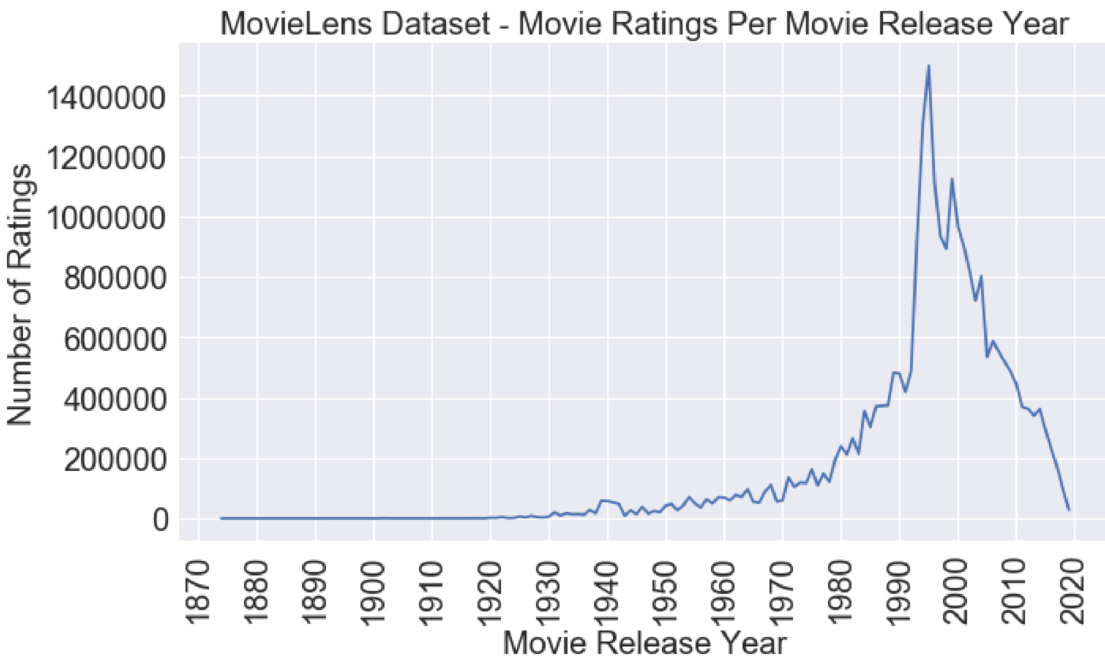
| | year | num_ratings |
|---|---|---|
| 11 | 1899 | 34 |
| 0 | 1874 | 18 |
| 3 | 1890 | 15 |
| 5 | 1892 | 13 |
| 1 | 1880 | 3 |

It looks like there are most reviews for movies released in 1995, with 1994 and 1999 following behind. There are far fewer reviews for movies released a long time ago (pre 1900s). This appears to show (as might be expected) that more people watch and review more current movies than very old ones!

Lets plot the ratings per year on a line graph to see how the number of ratings varies by year of film release:

[21]:
```
print(ratings_per_year[ratings_per_year['num_ratings'] ==␣
 ↪ratings_per_year['num_ratings'].max()])
plt.figure(figsize=[10,6])
plt.plot('year', 'num_ratings', data=ratings_per_year)
plt.xticks(np.arange(1870, 2021, 10), fontsize=20, rotation=90)
plt.yticks(fontsize=20)
plt.title('MovieLens Dataset - Movie Ratings Per Movie Release Year',␣
 ↪fontsize=20)
plt.xlabel('Movie Release Year', fontsize=20)
plt.ylabel('Number of Ratings', fontsize=20)
plt.tight_layout()
plt.show()
```

```
     year  num_ratings
107  1995      1497293
```




Lets also look at the number of movie ratings per genre over all time to see which movie genres have the most reviews.

[22]:
```
# Join movie_avg_rating_genres with ratings
ratings_per_genre = movie_avg_rat_genres.merge(ratings, on='movieId',␣
 ↪how='inner')

# Groupby genre and remove unnecessary columns, rename for clarity
ratings_per_genre = ratings_per_genre.groupby('genre', as_index=False).count()
ratings_per_genre = ratings_per_genre[['genre', 'movieId']]
ratings_per_genre.columns = ['genre', 'num_ratings']
ratings_per_genre.head()
```

[22]:

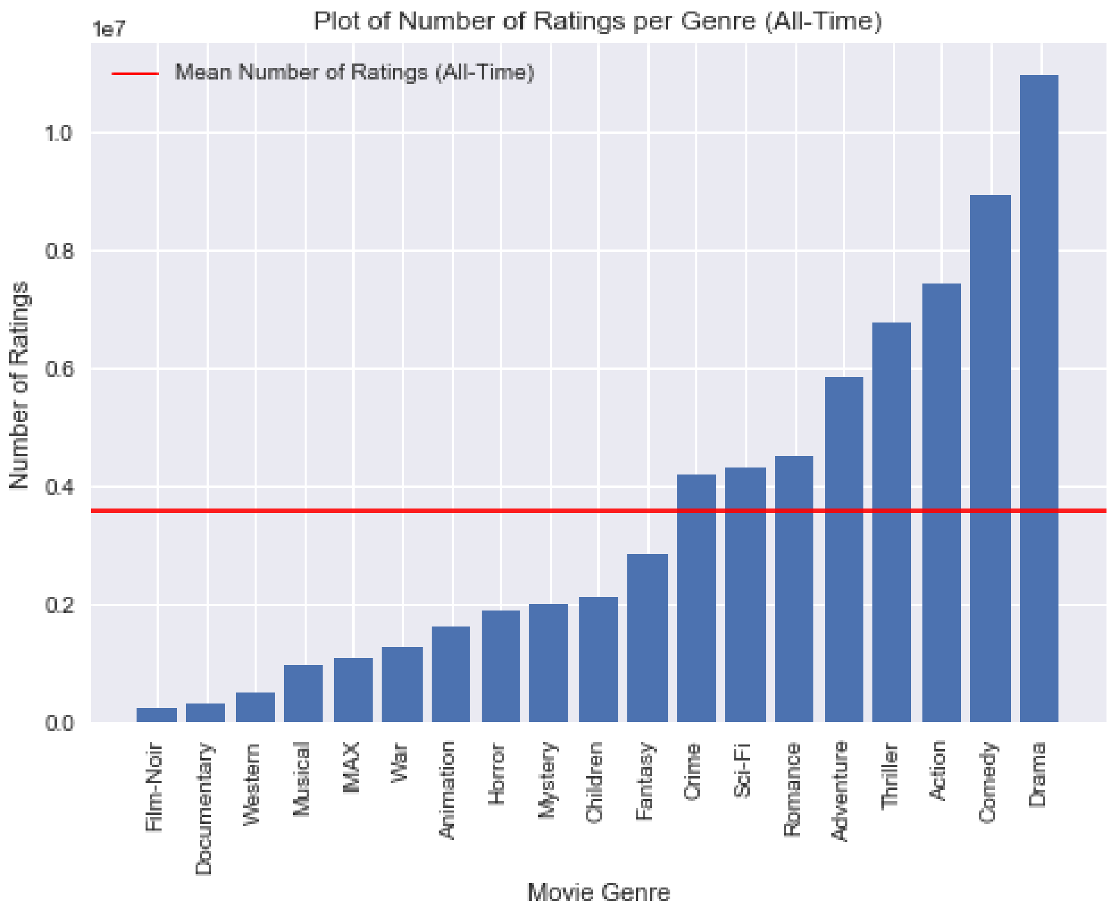
| | genre | num_ratings |
|---|---|---|
| 0 | Action | 7444344 |
| 1 | Adventure | 5832398 |
| 2 | Animation | 1630897 |
| 3 | Children | 2124214 |
| 4 | Comedy | 8926124 |

Lets create a bar plot of number of reviews for each genre in the database:

[23]:
```
# Sort values in ascending order
ratings_per_genre.sort_values('num_ratings', ascending=True, inplace=True)
print(ratings_per_genre.head(19))
print('Total number of ratings: ',ratings_per_genre['num_ratings'].sum())
```

```python
# Create Bar Plot:
plt.bar(ratings_per_genre['genre'].values, ratings_per_genre['num_ratings'])
plt.xticks(rotation=90)
plt.title('Plot of Number of Ratings per Genre (All-Time)')
plt.xlabel('Movie Genre')
plt.ylabel('Number of Ratings')
#plt.ylim(0, 4)
# Plot horizontal line at average score of all genres
plt.axhline(color='r', y=ratings_per_genre['num_ratings'].mean())
# Add legend for horizontal line
line = Line2D([0], [0], color='r', linewidth=1)
plt.legend([line], ['Mean Number of Ratings (All-Time)'])
plt.show()
```

```
          genre  num_ratings
9     Film-Noir       247227
6   Documentary       322359
18      Western       483731
12      Musical       964250
11         IMAX      1063279
17          War      1267346
2     Animation      1630897
10       Horror      1892070
13      Mystery      2010961
3      Children      2124214
8       Fantasy      2831544
5         Crime      4190215
15       Sci-Fi      4323063
14      Romance      4497170
1     Adventure      5832398
16     Thriller      6758772
0        Action      7444344
4        Comedy      8926124
7         Drama     10957241
Total number of ratings:  67767205
```




From the plot above it is clear that the two most highly reviewed genres (Film-Noir and Documentary) have the lowest number of total ratings. However while one of the most poorly reviewed genres, Sci-Fi, has an above average number of total ratings, Horror, which is the worst reviewed genre overall has a well below average number of reviews.

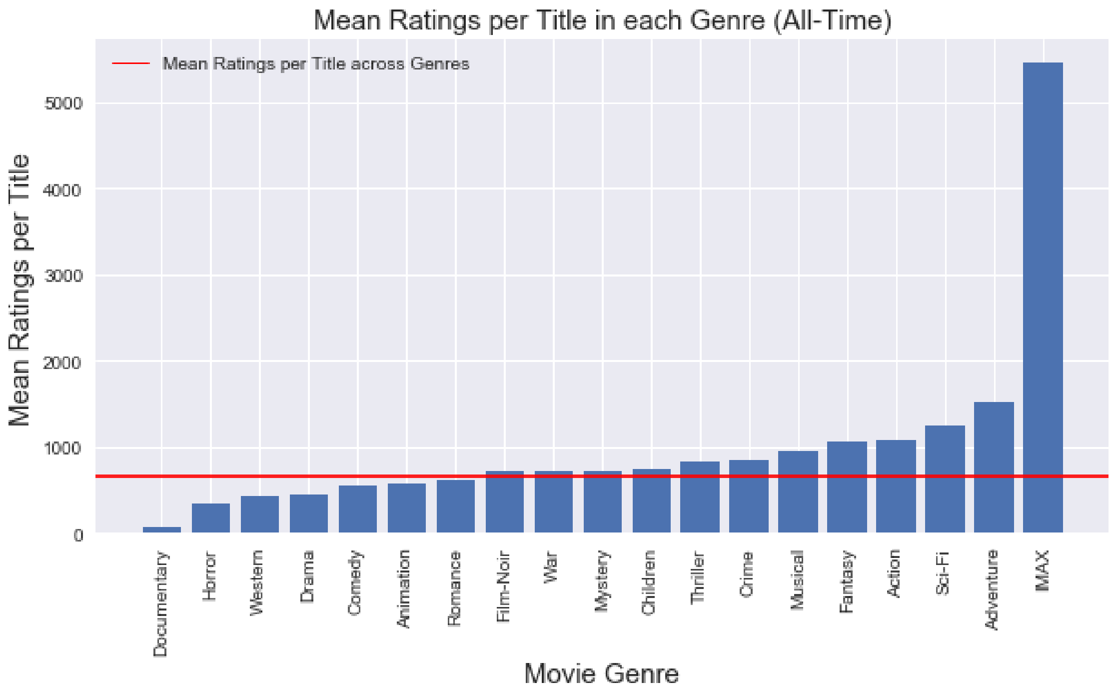
Since some genres may have many more titles of that category, it would make sense to create another plot that looks at the number of ratings per title for each genre:

[24]:
```
ratings_per_genre.head(19)
num_titles = []

for genre in list(ratings_per_genre['genre']):
    num_titles.append(movie_avg_rat_genres[movie_avg_rat_genres['genre'] ==␣
 ↪genre]['title'].count())

#print(num_titles, sum(num_titles)) # Agrees with number of titles * genres

ratings_per_genre['num_titles'] = num_titles
```

```python
ratings_per_genre['ratings_per_title'] = ratings_per_genre['num_ratings'] /
 ↪ratings_per_genre['num_titles']
print(ratings_per_genre.head(19))

# Create bar plot of ratings per title:

# Sort values in ascending order
ratings_per_genre.sort_values('ratings_per_title', ascending=True, inplace=True)

# Create Bar Plot:
plt.figure(figsize=[10,5])
plt.bar(ratings_per_genre['genre'].values,
 ↪ratings_per_genre['ratings_per_title'])
plt.xticks(rotation=90)
plt.title('Mean Ratings per Title in each Genre (All-Time)', fontsize=15)
plt.xlabel('Movie Genre', fontsize=15)
plt.ylabel('Mean Ratings per Title', fontsize=15)
#plt.ylim(0, 4)
# Plot horizontal line at average number of ratings per title
plt.axhline(color='r', y=ratings_per_genre['num_ratings'].sum() /
 ↪ratings_per_genre['num_titles'].sum())
# Add legend for horizontal line
line = Line2D([0], [0], color='r', linewidth=1)
plt.legend([line], ['Mean Ratings per Title across Genres'])
print(ratings_per_genre['num_ratings'].sum() / ratings_per_genre['num_titles'].
 ↪sum())
plt.show()
```

```
          genre  num_ratings  num_titles  ratings_per_title
9     Film-Noir       247227         349         708.386819
6   Documentary       322359        5416          59.519756
18      Western       483731        1156         418.452422
12      Musical       964250        1016         949.064961
11         IMAX      1063279         195        5452.712821
17          War      1267346        1770         716.014689
2     Animation      1630897        2909         560.638364
10       Horror      1892070        5728         330.319483
13      Mystery      2010961        2778         723.888049
3      Children      2124214        2862         742.213138
8       Fantasy      2831544        2660        1064.490226
5         Crime      4190215        5019         834.870492
15       Sci-Fi      4323063        3490        1238.700000
14      Romance      4497170        7295         616.472927
1     Adventure      5832398        3860        1510.983938
16     Thriller      6758772        8309         813.427849
```

```
0         Action      7444344         6903        1078.421556
4         Comedy      8926124        16028         556.908161
7          Drama     10957241        24424         448.625983
663.2983742304267
```




An interesting analysis would be to plot number of reviews per genre vs. number of films in the genre, to see if each genre receives a proportional number of reviews, or whether certain genres are reviewed by far moe people per film than others:

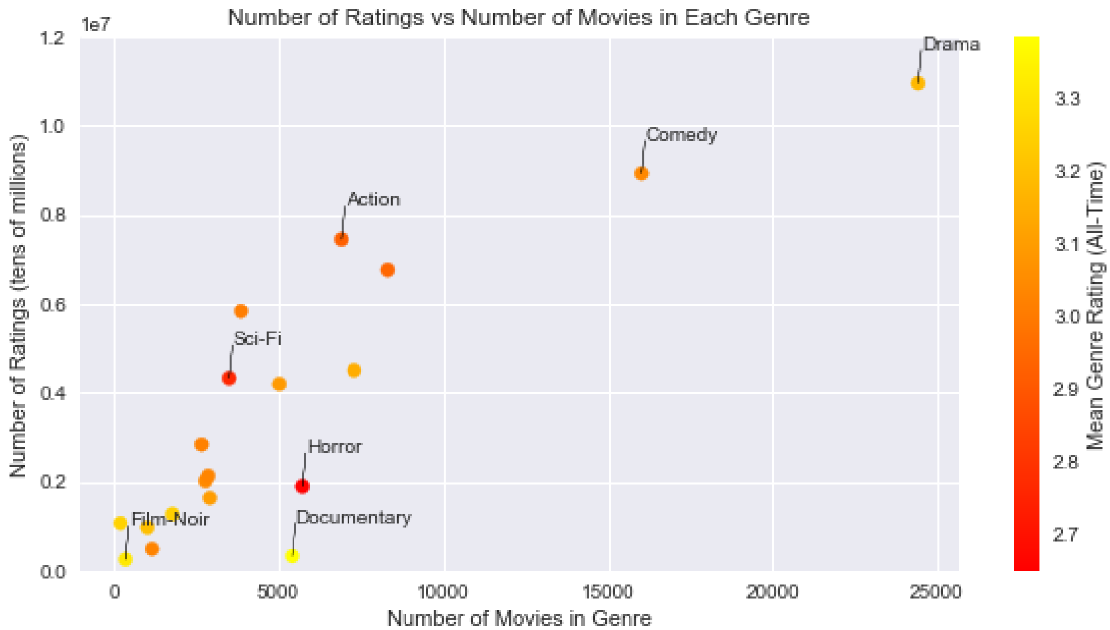
[25]:
```
# Get number of films in each genre
films_per_genre = movie_avg_rat_genres.groupby('genre', as_index=False).count()
films_per_genre = films_per_genre[['genre', 'movieId']]
films_per_genre.columns = ['genre', 'num_films']
print(films_per_genre.head(19))

# Join number of films per genre with ratings per genre, mean rating per genre:
genre_comparison = ratings_per_genre.merge(films_per_genre, on='genre',␣
↪how='inner')
genre_comparison = genre_comparison.merge(all_time_genre_ratings, on='genre',␣
↪how='inner')
genre_comparison = genre_comparison[['genre', 'num_ratings', 'rating',␣
↪'num_films', 'ratings_per_title']]
print(genre_comparison.head(19))

# Create figure
```

```
plt.figure(figsize=[10,5])
plt.scatter('num_films', 'num_ratings', data=genre_comparison, c='rating',␣
 ↪cmap=plt.cm.autumn)
cbar = plt.colorbar()
cbar.set_label('Mean Genre Rating (All-Time)')
plt.title('Number of Ratings vs Number of Movies in Each Genre')
plt.xlabel('Number of Movies in Genre')
plt.ylabel('Number of Ratings (tens of millions)')
plt.ylim([0, 1.2e7])

# First Degree Polynomial Fit to Data:
#linear_fit = np.poly1d(np.polyfit(x=genre_comparison['num_films'],␣
 ↪y=genre_comparison['num_ratings'], deg=1))
#m, b = np.polyfit(x=genre_comparison['num_films'],␣
 ↪y=genre_comparison['num_ratings'], deg=1)
#print(m, b)
#plt.plot(genre_comparison['num_films'],␣
 ↪linear_fit(genre_comparison['num_films']))
#line = Line2D([0], [0], color='#4c72b0', linewidth=1)
#plt.legend([line], ['Linear Fit'])

# Data Point Labels:
selected_genres = ['Documentary', 'Film-Noir', 'Horror', 'Sci-Fi', 'Action',␣
 ↪'Comedy', 'Drama']
x = genre_comparison[genre_comparison['genre'].
 ↪isin(selected_genres)]['num_films'].values
y = genre_comparison[genre_comparison['genre'].
 ↪isin(selected_genres)]['num_ratings'].values
names = genre_comparison[genre_comparison['genre'].
 ↪isin(selected_genres)]['genre'].values

for i, name in enumerate(names):
    x_offset = 100
    y_offset = 750000
    plt.annotate(name, (x[i]+x_offset, y[i]+y_offset))
    plt.arrow(x[i]+x_offset, y[i]+y_offset, -x_offset, -y_offset)

plt.show()

print(genre_comparison.corr())
```

```
         genre  num_films
0       Action       6903
1    Adventure       3860
2    Animation       2909
3     Children       2862
4       Comedy      16028
```

| | | |
|---|---|---|
| 5 | Crime | 5019 |
| 6 | Documentary | 5416 |
| 7 | Drama | 24424 |
| 8 | Fantasy | 2660 |
| 9 | Film-Noir | 349 |
| 10 | Horror | 5728 |
| 11 | IMAX | 195 |
| 12 | Musical | 1016 |
| 13 | Mystery | 2778 |
| 14 | Romance | 7295 |
| 15 | Sci-Fi | 3490 |
| 16 | Thriller | 8309 |
| 17 | War | 1770 |
| 18 | Western | 1156 |

| | genre | num_ratings | rating | num_films | ratings_per_title |
|---|---|---|---|---|---|
| 0 | Documentary | 322359 | 3.383110 | 5416 | 59.519756 |
| 1 | Horror | 1892070 | 2.650763 | 5728 | 330.319483 |
| 2 | Western | 483731 | 3.025755 | 1156 | 418.452422 |
| 3 | Drama | 10957241 | 3.178987 | 24424 | 448.625983 |
| 4 | Comedy | 8926124 | 3.042288 | 16028 | 556.908161 |
| 5 | Animation | 1630897 | 3.100958 | 2909 | 560.638364 |
| 6 | Romance | 4497170 | 3.147334 | 7295 | 616.472927 |
| 7 | Film-Noir | 247227 | 3.318142 | 349 | 708.386819 |
| 8 | War | 1267346 | 3.246515 | 1770 | 716.014689 |
| 9 | Mystery | 2010961 | 3.045802 | 2778 | 723.888049 |
| 10 | Children | 2124214 | 3.032964 | 2862 | 742.213138 |
| 11 | Thriller | 6758772 | 2.945069 | 8309 | 813.427849 |
| 12 | Crime | 4190215 | 3.092559 | 5019 | 834.870492 |
| 13 | Musical | 964250 | 3.199744 | 1016 | 949.064961 |
| 14 | Fantasy | 2831544 | 3.022389 | 2660 | 1064.490226 |
| 15 | Action | 7444344 | 2.925590 | 6903 | 1078.421556 |
| 16 | Sci-Fi | 4323063 | 2.761345 | 3490 | 1238.700000 |
| 17 | Adventure | 5832398 | 3.010614 | 3860 | 1510.983938 |
| 18 | IMAX | 1063279 | 3.252121 | 195 | 5452.712821 |




| | num_ratings | rating | num_films | ratings_per_title |
|---|---|---|---|---|
| num_ratings | 1.000000 | -0.269135 | 0.860396 | -0.120534 |
| rating | -0.269135 | 1.000000 | -0.059239 | 0.142688 |
| num_films | 0.860396 | -0.059239 | 1.000000 | -0.278687 |
| ratings_per_title | -0.120534 | 0.142688 | -0.278687 | 1.000000 |

[ ]:

It can clearly be seen from the graph and the correlation matrix that there is a strong correlation between the number of film ratings and the number of reviews. The Pearson Correlation Coefficient is 0.86 - very high.

It can also be seen that the Film-Noir and Documentary genres fall well below the trendline - they have a lower than average number of reviews for the number of films in the genres. It could be that films in these categories appeal very specifically to small subsections of fans of the genre. For example, a subsection of documentary fans are interested in sports documentaries, another subsection in war documentaries etc. As such each film will only be watched (and potentially reviewed) by a far smalller number of people.

By comparison, film genres such as Action and Sci-Fi, are placed above the trendline, with a higher than average number of reviews for the number of films in the genres. These genres clearly attract many reviews per film, perhaps because these films appeal to a much broader audience?

With the movie_avg_rat_genres dataframe we can also plot line graphs of the average review score for each genre per year, to determine whether there have been trends in the popularity of film genres over time.

Since from the plot of reviews per year above we have seen there are far fewer reviews for older films, and also there are as many reviews for films after 2017 lets look at reviews in the years 1980

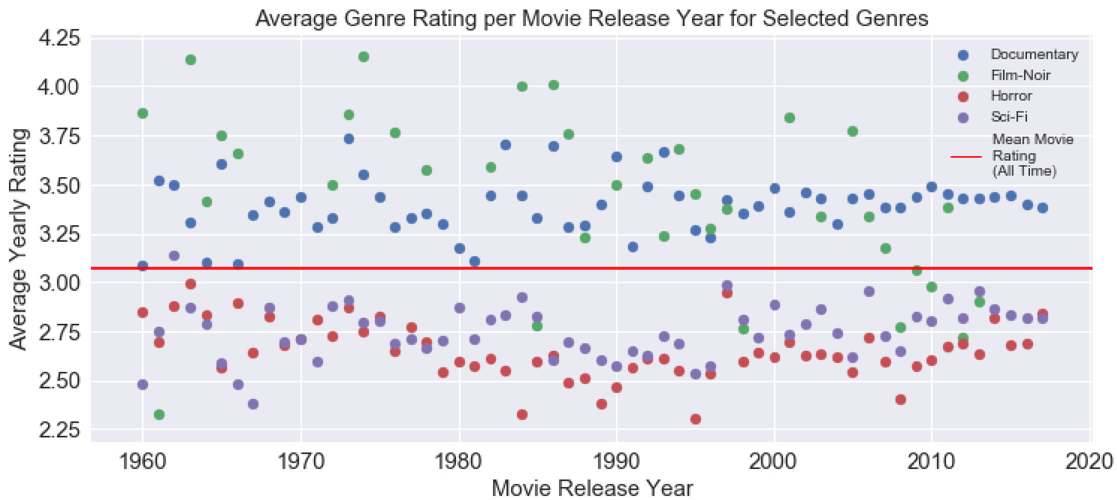
- 2017.

[26]:
```python
# Group Film Ratings by Year and then By Genre:
per_year_genre_ratings = movie_avg_rat_genres.groupby(['year', 'genre'],␣
↪as_index=False).mean()
per_year_genre_ratings.tail(10)
```

[26]:
```
      year        genre    rating
1936  2019  Documentary  2.958223
1937  2019        Drama  2.759569
1938  2019      Fantasy  2.803967
1939  2019       Horror  2.232229
1940  2019      Mystery  2.852543
1941  2019      Romance  2.829436
1942  2019       Sci-Fi  2.497187
1943  2019     Thriller  2.540989
1944  2019          War  2.612769
1945  2019      Western  2.063752
```

[27]:
```python
years = np.arange(1960, 2018)
selected_genres = ['Documentary', 'Film-Noir', 'Horror', 'Sci-Fi']
# Select 1960 to 2019 and genres
recent_per_year_ratings = per_year_genre_ratings[per_year_genre_ratings['year'].
↪isin(years)]
selected_per_year_ratings =␣
↪recent_per_year_ratings[recent_per_year_ratings['genre'].isin(selected_genres)]

plt.figure(figsize=[12,5])

# Plot the data using a line graph:
for genre in selected_genres:
    data = selected_per_year_ratings[selected_per_year_ratings['genre'] == genre]
    plt.scatter('year', 'rating', data=data)

plt.title('Average Genre Rating per Movie Release Year for Selected Genres',␣
↪fontsize=15)
plt.xlabel('Movie Release Year', fontsize=15)
plt.ylabel('Average Yearly Rating', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.gca().add_artist(plt.legend(selected_genres))

# Plot horizontal line at average score of all genres
plt.axhline(color='r', y=all_time_genre_ratings['rating'].mean())

# Add legend for horizontal line
line = Line2D([0], [0], color='r', linewidth=1)
```

```
plt.legend([line], ['Mean Movie \nRating \n(All Time)'], bbox_to_anchor=(0.845,␣
↪0.79))


plt.show()
```




From the above plot it is clear that films in the Documentary genre rate well above the mean rating across all genres, year on year. Film-Noir is similar although movies in this genre do appear to have suffered from some lower than average review scores in more recent years (post 2008).

Similarly, Horror and Sci-Fi films score below the mean genre rating, year on year (with one exception in the 60s for the Sci-Fi genre). These genres are rated worse than average for nearly all years from 1970 - 2019. One interesting analysis would be to plot the distribution of rating scores for each of the genres, to see if there is a noticeable difference in the shape of the distributions for highly-rated and poorly-rated genres.

First we need to get each individual rating for each movie / genre by joining the ratings dataframe to the 'movie_avg_rat_genres' dataframe:

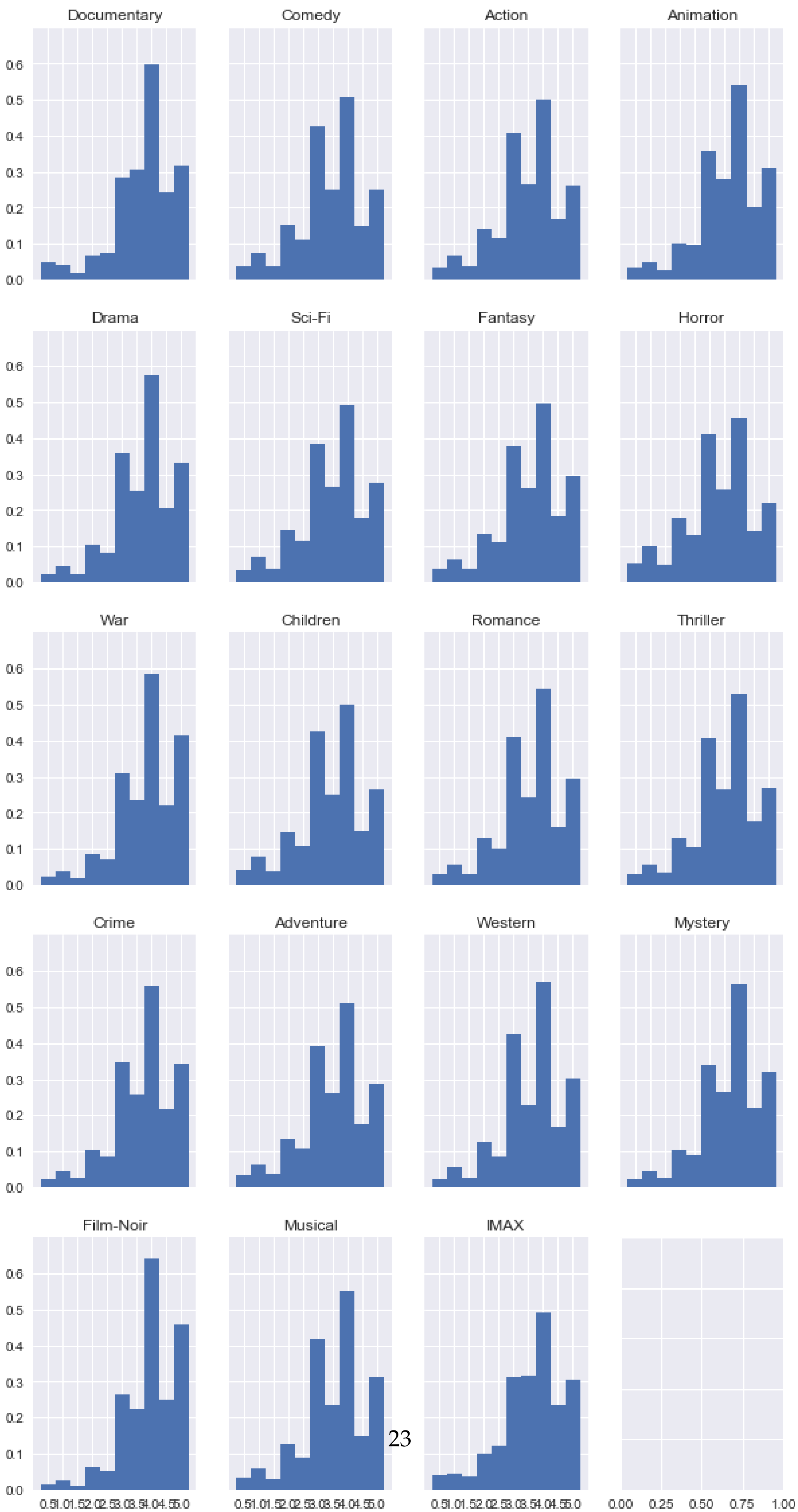
[28]:
```
# Inner join to ratings dataframe:
movie_genre_ratings = movie_avg_rat_genres[['movieId', 'title', 'year',␣
↪'genre']].merge(ratings[['movieId', 'rating']], on='movieId', how='inner')
movie_genre_ratings.head()
```

[28]:

| | movieId | title | year | genre | rating |
|---|---|---|---|---|---|
| 0 | 1 | Toy Story (1995) | 1995 | Adventure | 3.5 |
| 1 | 1 | Toy Story (1995) | 1995 | Adventure | 4.0 |
| 2 | 1 | Toy Story (1995) | 1995 | Adventure | 3.0 |
| 3 | 1 | Toy Story (1995) | 1995 | Adventure | 4.0 |
| 4 | 1 | Toy Story (1995) | 1995 | Adventure | 4.0 |

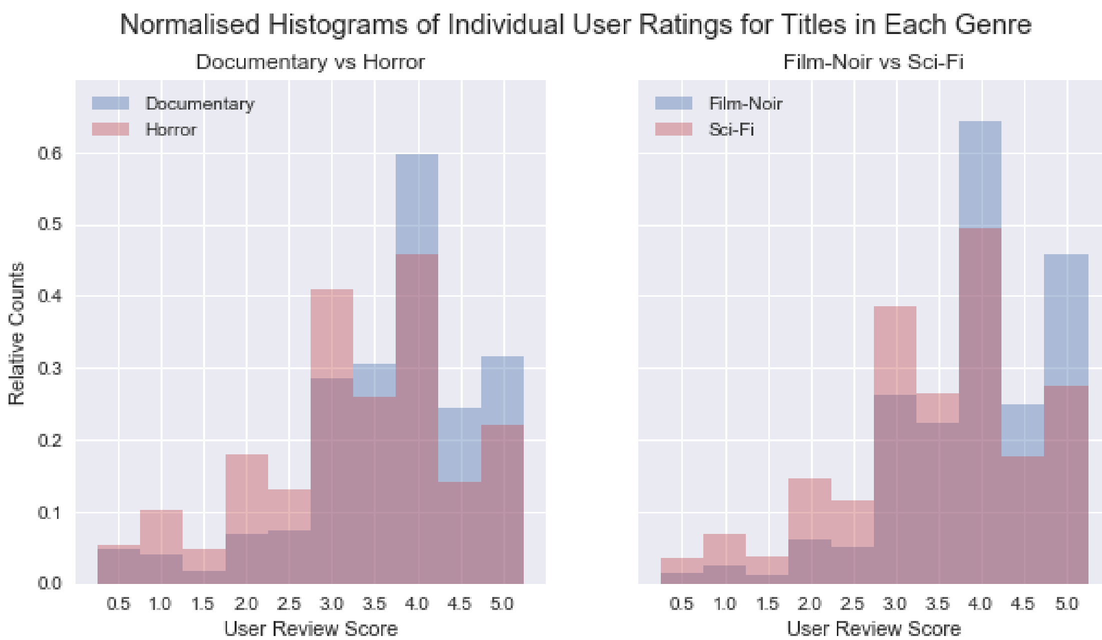
Now we can plot histograms for the ratings for each genre:

[29]:
```
# List of all Genres
genre_list = list(per_year_genre_ratings['genre'].unique())
genre_tuples = []

g = 0
for i in range(5):
    for j in range(4):
        genre_tuples.append((i, j, genre_list[g]))
        g += 1
        if g == 19:
            break

# Create Subplots
fig, axs = plt.subplots(5, 4)
fig.set_figheight(20)
fig.set_figwidth(10)

# Plot historgram for each genre
for entry in genre_tuples:
    plot = axs[entry[0], entry[1]]
    data = movie_genre_ratings[movie_genre_ratings['genre'] == entry[2]]
    plot.hist(data['rating'], bins=np.arange(0.5,6.0, step=0.5)-0.25,␣
↪density=True)
    plot.set_title(entry[2])
    plot.set_xticks(np.arange(0.5, 5.5, 0.5))
    plot.set_yticks(np.arange(0, 0.7, 0.1))
    plot.set_ylim(0, 0.7)

for ax in fig.get_axes():
    ax.label_outer()

# Rotate x axis labels 90 degrees
#plt.setp(axs.xaxis.get_majorticklabels(), rotation=45)

plt.show()
```


Documentary
Comedy
Action
Animation
Drama
Sci-Fi
Fantasy
Horror
War
Children
Romance
Thriller
Crime
Adventure
Western
Mystery
Film-Noir
Musical
IMAX
0.0 0.1 0.2 0.3 0.4 0.5 0.6
0.51.01.52.02.53.03.54.04.55.0
0.00 0.25 0.50 0.75 1.00

Quite a large plot! Lets narrow it down to the two highest and two lowest rated genres for comparison:

[40]:
```
selected_genres = [('Documentary', 'Horror') , ('Film-Noir',  'Sci-Fi')]
colors = ['#4c72b0', '#c44e52']

# Create Subplots
fig, axs = plt.subplots(1, 2)
fig.set_figheight(5)
fig.set_figwidth(10)
fig.suptitle('Normalised Histograms of Individual User Ratings for Titles in␣
↪Each Genre', fontsize=15)

# Plot historgram for each genre
for i in range(len(selected_genres)):
    plot = axs[i]
    plot.set_title(f'{selected_genres[i][0]} vs {selected_genres[i][1]}')
    plot.set_xticks(np.arange(0.5, 5.5, 0.5))
    plot.set_yticks(np.arange(0, 0.7, 0.1))
    plot.set_ylim(0, 0.7)
    plot.set_xlabel('User Review Score')
    plot.set_ylabel('Relative Counts')

    for j in range(2):
        data = movie_genre_ratings[movie_genre_ratings['genre'] ==␣
↪selected_genres[i][j]]
        plot.hist(data['rating'], bins=np.arange(0.5,6.0, step=0.5)-0.25,␣
↪density=True, alpha=0.4, label= selected_genres[i][j], color=colors[j])
        plot.legend(loc='upper left')


# Only show outer axis labels
for ax in fig.get_axes():
        ax.label_outer()

plt.show()
```




From the overlaid normalised histograms for the four genres shown above, it can clearly be seen that for both the Horror and Sci-Fi movie genres, the distribution of review scores is not as left-skewed than for Documentary and Film Noir. A much higher proportion of ratings for films in the Horror and Sci-Fi genres are in the 1-3 star range, than for Documentary and Film Noir.

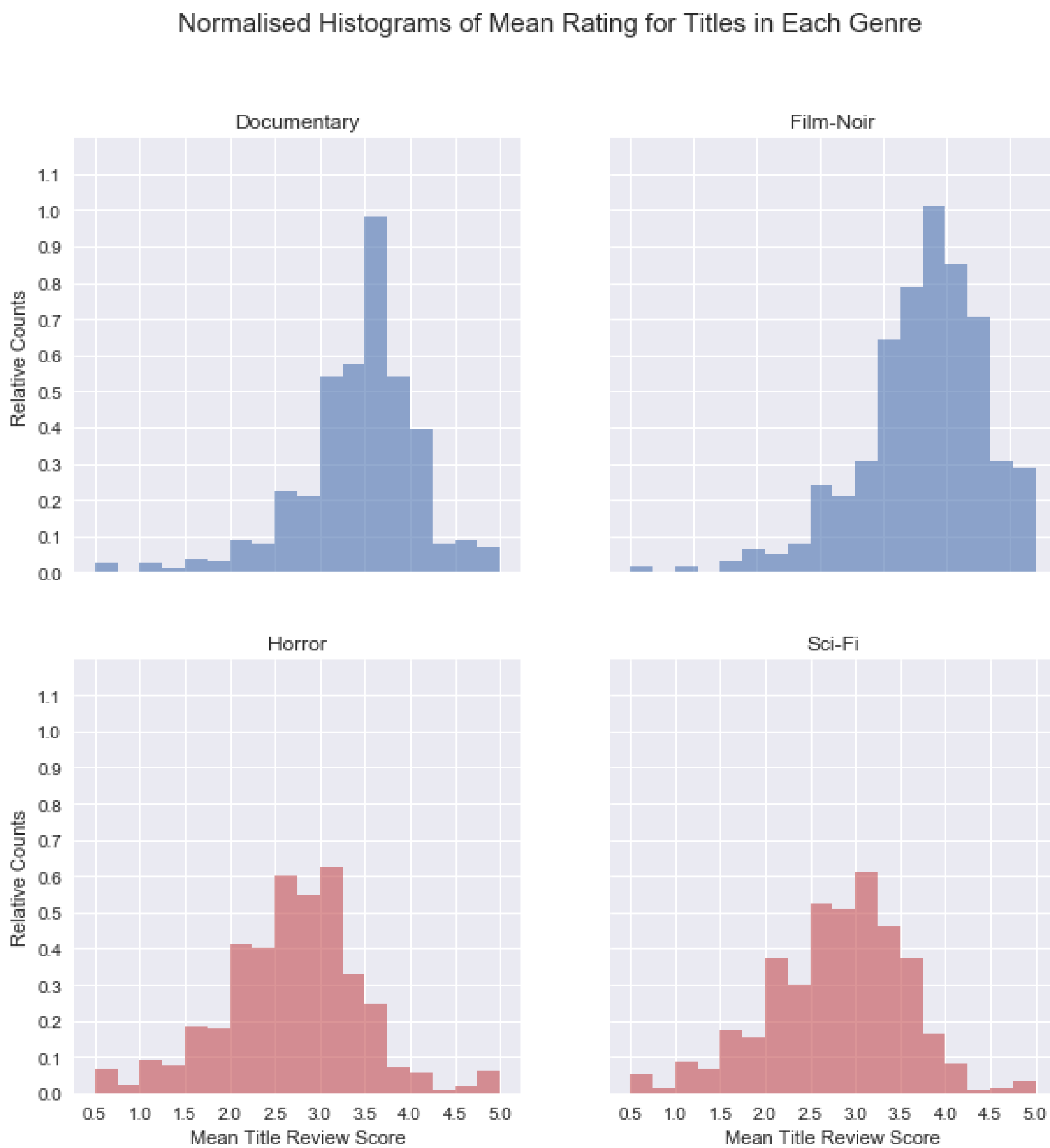
The distribution of ratings for Horror and Sci-Fi films could be described as 'broader' than that of Documentary and Film-Noir, which have a narrower peak centered on review scores of ~ 4.0.

Lets also look at whether the distributions look different when you take the mean rating for each film in the selected genres:

[109]:
```
# Group by movieId and take mean rating, keeping the genres for each movie listed
avg_movie_genre_ratings = movie_genre_ratings.groupby(['movieId', 'genre'],␣
 ↪as_index=False).mean()

selected_genres = ['Documentary', 'Film-Noir', 'Horror', 'Sci-Fi']
colors = ['#4c72b0', '#4c72b0', '#c44e52', '#c44e52']

# Plot histograms for the selected genres against each other:
# Create Subplots
fig, axs = plt.subplots(2, 2)
fig.set_figheight(10)
fig.set_figwidth(10)
fig.suptitle('Normalised Histograms of Mean Rating for Titles in Each Genre',␣
 ↪fontsize=15)

# Plot historgram for each genre
```

```
p = 0
for i in range(2):
    for j in range(2):
        plot = axs[i][j]
        plot.set_title(f'{selected_genres[p]}')
        plot.set_xticks(np.arange(0.5, 5.5, 0.5))
        plot.set_yticks(np.arange(0, 1.2, 0.1))
        plot.set_ylim(0, 1.2)
        plot.set_xlabel('Mean Title Review Score')
        plot.set_ylabel('Relative Counts')

        data = avg_movie_genre_ratings[avg_movie_genre_ratings['genre'] ==␣
 ↪selected_genres[p]]
        plot.hist(data['rating'], density=True, bins=18, alpha=0.6,␣
 ↪color=colors[p])
        plot.legend(loc='upper left')

        p += 1


# Only show outer axis labels
for ax in fig.get_axes():
        ax.label_outer()

plt.show()
```

```
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
No handles with labels found to put in legend.
```




It is easy to see from the above histogram the quite stark differences in the distribution of review scores across the high-scoring and low-scoring genres. Documentary and Film-Noir Movies have relatively few titles scoring below 3.0, while Horror and Sci-Fi have many films with scores in the 1.0-2.5 range. A possible cause of this could be that Documentary and Film-Noir films are more niche genres and watched by a smaller audience of fans of genre / documentary subject. This would likely result in reviews written mostly by fans of the genre and so increase the chance of a highly rated film. For Horror and SciFi, their broader distribution could perhaps be attributed to many more poorer quality titles being created for the genres over the years - low quality horror and Sci-Fi films being relatively common.

Finally lets look at whether movies that are labelled with more genres do better or worse than movies labelled with fewer genres.

[144]:
```
# Start by looking at this dataframe
movie_avg_ratings.head()
```

[144]:
```
   movieId                               title    rating  year
0      1.0                    Toy Story (1995)  3.893708  1995
1      2.0                      Jumanji (1995)  3.251527  1995
2      3.0             Grumpier Old Men (1995)  3.142028  1995
3      4.0            Waiting to Exhale (1995)  2.853547  1995
4      5.0  Father of the Bride Part II (1995)  3.058434  1995
```

[145]:
```
# Add back in the list of genres for each film
genres_per_title = movie_avg_ratings.merge(movie_genres[['movieId', 'genres']],␣
 ↪on='movieId', how='inner')

# Get number of genres for each title:
genres_per_title['num_genres'] = genres_per_title['genres'].str.len()
genres_per_title.head()
```

[145]:
```
   movieId                               title    rating  year  \
0      1.0                    Toy Story (1995)  3.893708  1995
1      2.0                      Jumanji (1995)  3.251527  1995
2      3.0             Grumpier Old Men (1995)  3.142028  1995
3      4.0            Waiting to Exhale (1995)  2.853547  1995
4      5.0  Father of the Bride Part II (1995)  3.058434  1995

                                          genres  num_genres
0  [Adventure, Animation, Children, Comedy, Fantasy]           5
1                   [Adventure, Children, Fantasy]           3
2                                [Comedy, Romance]           2
3                         [Comedy, Drama, Romance]           3
4                                         [Comedy]           1
```

Great, now we can plot the average ratings of films vs the number of genres:

[155]:
```
# Get average rating for each set of movies with X number genres
genres_per_title_avg_rating = genres_per_title[['rating','num_genres']].
 ↪groupby('num_genres', as_index=False).mean()

# Get number of titles with X many genres as well
genres_per_title_count = genres_per_title[['rating','num_genres']].
 ↪groupby('num_genres', as_index=False).count()
genres_per_title_count.columns = ['num_genres', 'num_titles']

genres_per_title_avg_rating = genres_per_title_avg_rating.
 ↪merge(genres_per_title_count, on='num_genres', how='inner')
```

```
genres_per_title_avg_rating['num_titles_perc'] = 100 *␣
 ↪genres_per_title_avg_rating['num_titles'] /␣
 ↪genres_per_title_avg_rating['num_titles'].sum()
genres_per_title_avg_rating.columns = ['num_genres', 'avg_rating', 'num_titles',␣
 ↪'num_titles_perc']
genres_per_title_std_rating = genres_per_title[['rating', 'num_genres']].
 ↪groupby('num_genres', as_index=True).std()
genres_per_title_std_rating = genres_per_title_std_rating.reset_index()
genres_per_title_std_rating.columns = ['num_genres', 'std_rating']
genres_per_title_std_rating.head()

genres_per_title_avg_rating = genres_per_title_avg_rating.
 ↪merge(genres_per_title_std_rating, on='num_genres', how='inner')
genres_per_title_avg_rating.head(10)
```

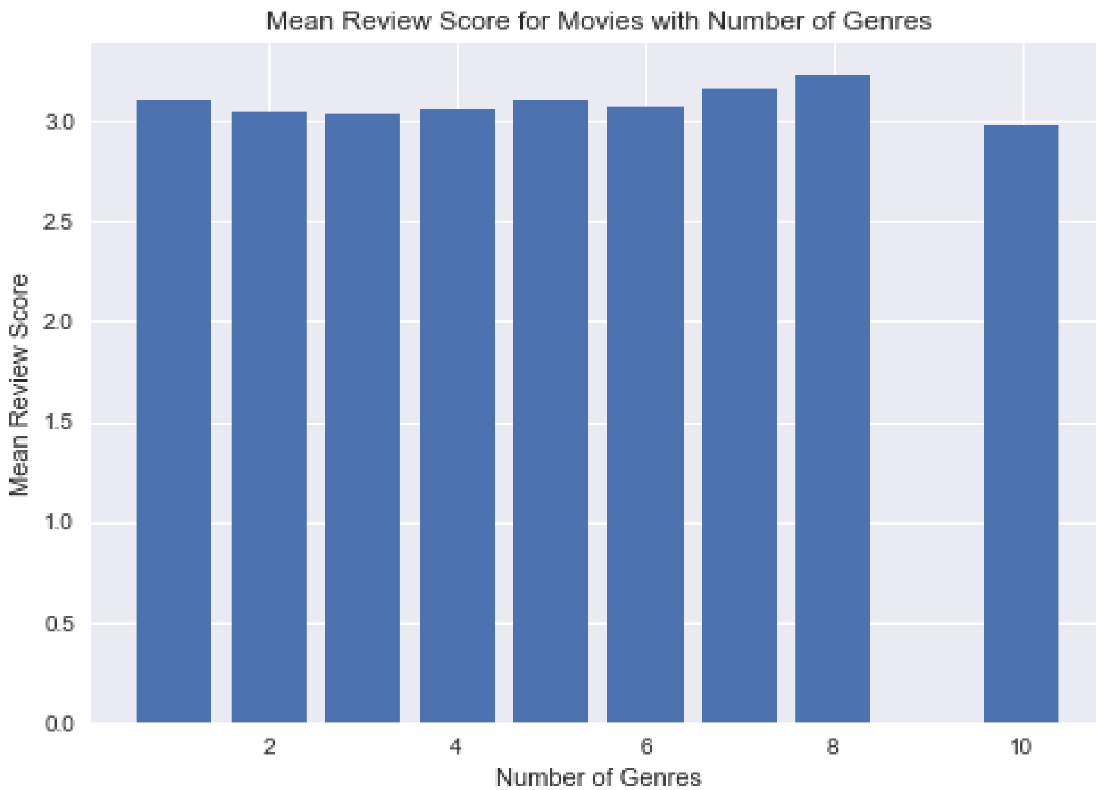
[155]:

| | num_genres | avg_rating | num_titles | num_titles_perc | std_rating |
|---|---|---|---|---|---|
| 0 | 1 | 3.109229 | 24027 | 44.207912 | 0.739251 |
| 1 | 2 | 3.043785 | 17335 | 31.895124 | 0.714576 |
| 2 | 3 | 3.037269 | 9476 | 17.435143 | 0.703800 |
| 3 | 4 | 3.063298 | 2700 | 4.967801 | 0.653624 |
| 4 | 5 | 3.101604 | 662 | 1.218031 | 0.647633 |
| 5 | 6 | 3.072314 | 123 | 0.226311 | 0.731424 |
| 6 | 7 | 3.158067 | 24 | 0.044158 | 0.530741 |
| 7 | 8 | 3.228788 | 2 | 0.003680 | 0.194990 |
| 8 | 10 | 2.978520 | 1 | 0.001840 | NaN |

Looks like there are many titles with 1, 2 or 3 listed genres, then the number of titles with 4 or more genres drops off quite steeply. Also we can see from above that although films with only 1 listed genre rate slightly higher than those with 2, 3 or 4 listed genres, these values are all well within the standard deviation of around 0.6-0.7 for each number of genres.

Lets plot a quick bar graph to see what these look like:

[85]:
```
plt.bar('num_genres', 'rating', data=genres_per_title_avg_rating)
plt.title('Mean Review Score for Movies with Number of Genres')
plt.xlabel('Number of Genres')
plt.ylabel('Mean Review Score')
plt.show()
print(genres_per_title['rating'].mean())
```




```
3.073374788023547
```

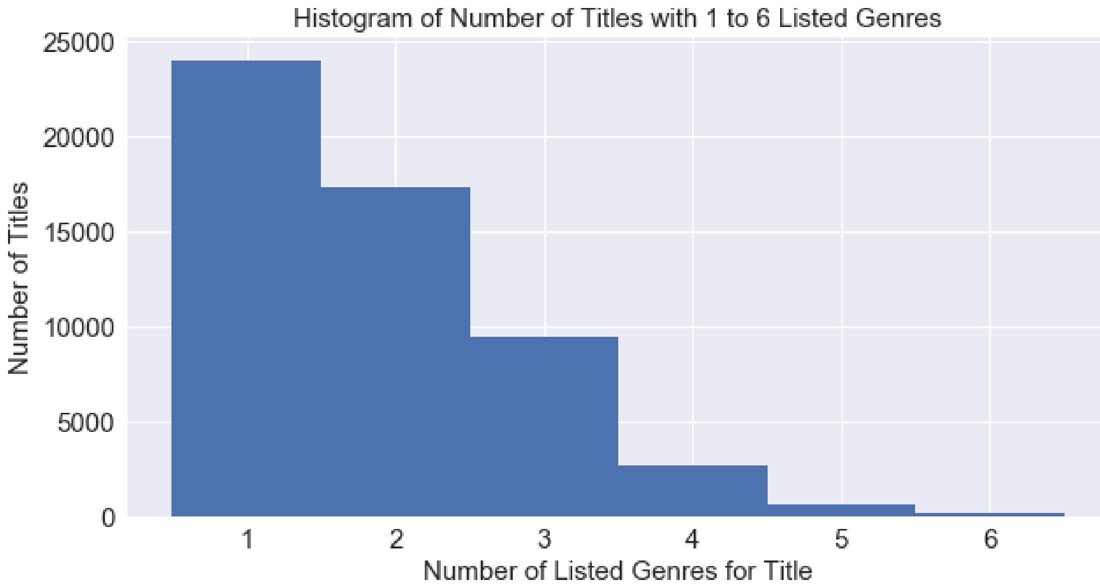
Lets see if anything interesting can be seen by plotting boxplots for the different number of genres titles have, sticking to the range 1-4 genres where there are more titles:

[156]:
```
boxplot = genres_per_title[genres_per_title['num_genres'] <= 4].
    ↪boxplot(column=['rating'], by='num_genres', figsize=(10,6), showmeans=True)
boxplot.set_title('Boxplots of Review Scores for Movies with 1-4 listed genres',␣
    ↪fontsize=15)
boxplot.set_xlabel('Number of listed genres for movie', fontsize=15)
boxplot.set_ylabel('Review Score', fontsize=15)
boxplot.get_figure().suptitle('')
plt.show()
```


The box plot is a bit easier to understand the details of than the bar plot. There does not seem to be much difference between the review score distributions for movies with 1-4 genres! Clearly both good and bad movies can be made with any number of genres - having a more focused title with fewer genres or a more accessible title spanning more genres does not appear to confer any advantage when it comes to movie ratings.

Lets also create a histogram of the number of genres each title in the dataset has:

[125]:
```
plt.figure(figsize=(10,5))
plt.hist('num_genres', data=genres_per_title[genres_per_title['num_genres']
    <=6], bins=np.arange(1,8)-0.5)
plt.title('Histogram of Number of Titles with 1 to 6 Listed Genres', fontsize=15)
plt.xlabel('Number of Listed Genres for Title', fontsize=15)
plt.ylabel('Number of Titles', fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```




I think that has answered the initial questions, in summary:

The MovieLens dataset was analysed to determine the differences in review scores for titles with different listed genres.

Movies with Documentary and Film-Noir genres listed have the highest mean review score. Movies with Horror and Sci-Fi genres listed have the lowest mean review score.

The mean title score distributions for the two highest scoring genres are very different to those of the two lowest scoring genres. Documentary and Film-Noir are likely more niche genres, more likely to be watched and reviewed by fans of the genre or subject matter, their distributions skew towards high scores. Horror and Sci-Fi genres possibly contain more examples of poor-quality films, resulting in a broader distribution of lower scores. They also might be watched and reviewed by a more general audience, such that not all review scores are likely to be favourable, bringing average review scores down.

Movies with a single genre listed have a similar review score distribution to those with 2, 3 or 4 listed genres.

```
[ ]:
```

```
[ ]:
```