CSCI.620.01/03 Homework6 Suraj Sureshkumar

Task 1:

The information stored in the file contains:

- The movie id which is stored in the IMDb_ID giving us a unique id for each movie.
- The type of the movie id whether it is string, int, float etc,. In this case it is a literal.
- The rating for each movie.
- The box office revenue for each movie and its representation in its respective country currency.
- The cost for each movie and its associated data type.
- Distributors for each movie and the data type associated with it.
- The title of each movie is represented by the titleLabel.
- Each field has its own data type and value associated with it and is of JSON format.
- The file is in JSON document format where each document contains the information about a single movie.

Task 2:

First I uncompressed the gz file and then created a dummy collection and uploaded the data. After that I exported the data as csv to perform the cleaning on the file and pre-processed it using python. Where the currency label with only United States Dollar is taken into account. From IMDb_ID the starting prefix of ‘tt’ was removed and the duplicates were dropped. Making the data more refined under the constraint of USD.

Below is the query for the update on update on ID(movie id)

```
db.Movies.aggregate([
   {
      $lookup: {
         from : "extra_data",
         localField : "id",
         foreignField : "id",
         as : "movie_box_office"
         }
   },
   {$out: "Movies"}
])
```

The below query counts the documents which have been updated

```
db.Movies.find({
  $expr: {
    $in: [
      "$id",
      "$movie_box_office.id"
    ]
  }
}).count()
```

The documents updated were **1390 documents**

Task 3:

The matching process can be performed with the title as that is the only common field that can be updated.
In cases where titles are the same there will be a match and also where there are more than one movies with the same titles, there will be matching with those titles from extra data provided.

Below is the query for the update on update on title(movie title)

```
db.Movies.aggregate([
   {
      $lookup: {
         from : "extra_data",
         localField : "title",
         foreignField : "titleLabel.value",
         as : "movie_box_office1"
         }
   },
   {$out: "Movies1"}
])
```

The below query counts the documents which have been updated

```
db.Movies1.find({
  $expr: {
    $in: [
      "$title",
      "$titleLabel.value"
    ]
  }
}).count()
```

The documents updated were **20472 documents**

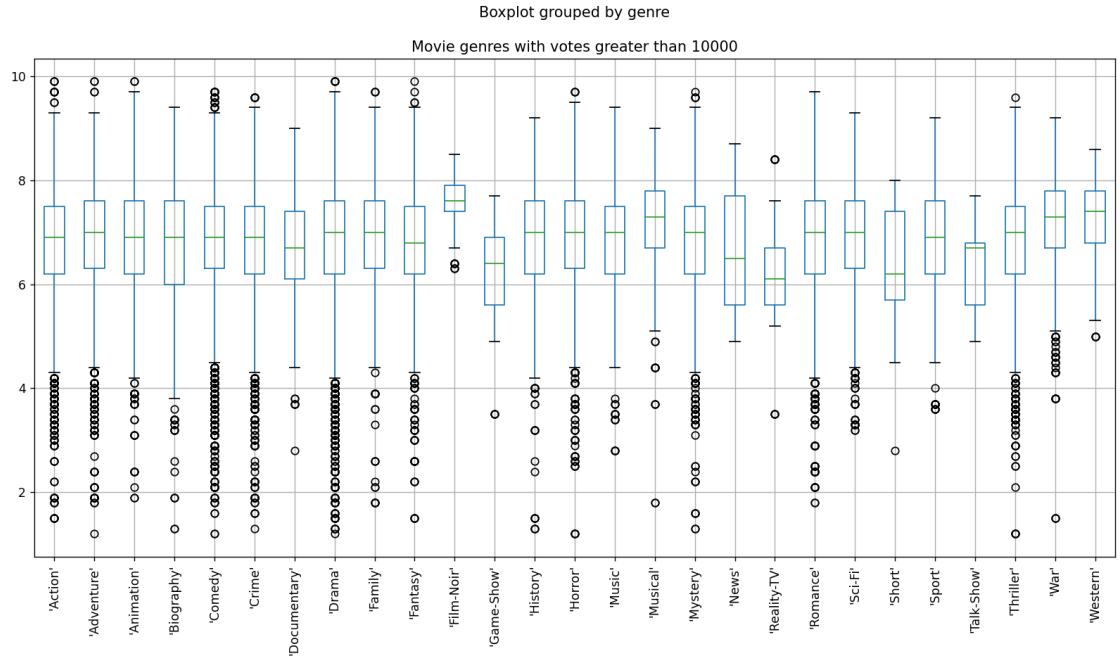
Task 4:

Using the Python package Matplotlib to plot the bar, whisker, and bar chart and the time series plot. All the source code is in a python file which is available in the source code zip file.

4.1:

average ratings of movies with more than 10K vote

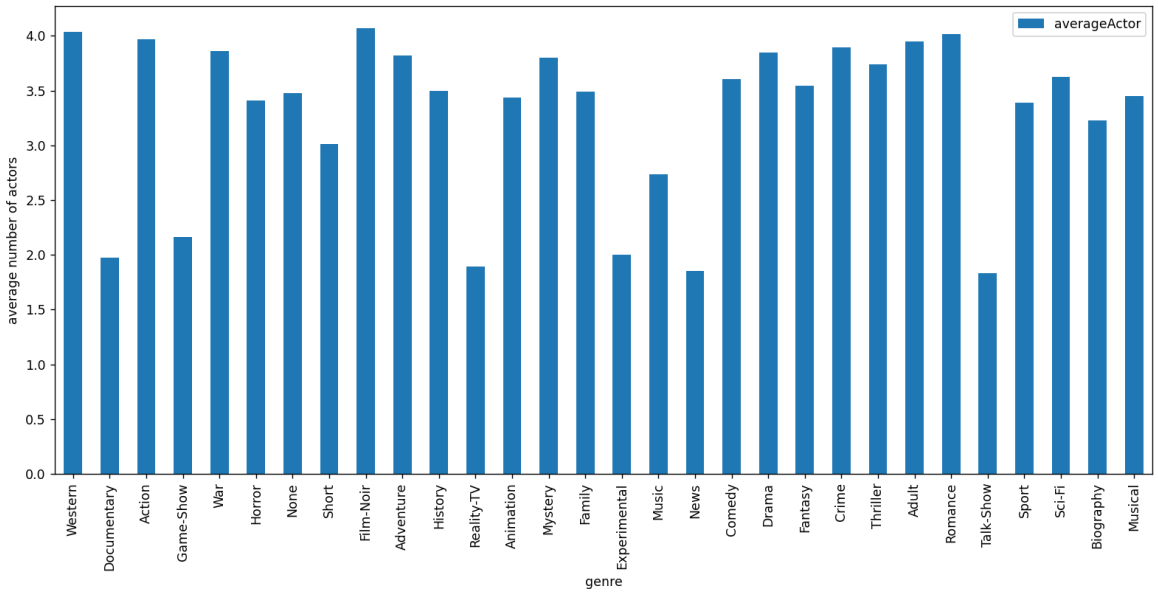
The below output displays the box and whisker plot where outliers are also displayed.



4.2

Average number of actors per movie by genre

The below output gives us a bar graph for the average number of actors per movie by genre and the trends in the data. x-axis represents the genres where y-axis represents the average number of actors per movie by genre.



4.3

Number of movies produced each year (startYear)

The output of the time series plot is as shown below where the x-axis is the year and the y-axis is the number of movies in the years corresponding to x-axis.