Jennie Tannenbaum and Maya Horowitz SI 206 Final Project
Link to Repository: https://github.com/mayahoro/FInal-Project/tree/main

Goals

The main goal for our project was to examine the data from the IMDb API and ultimatemovierankings.com to see the top 100 movies. From this, we wanted to see which Directors had produced the most movies in the top rankings and which years the top movies were made in. Also, we wanted to see the top 100 movies that had the highest box office.

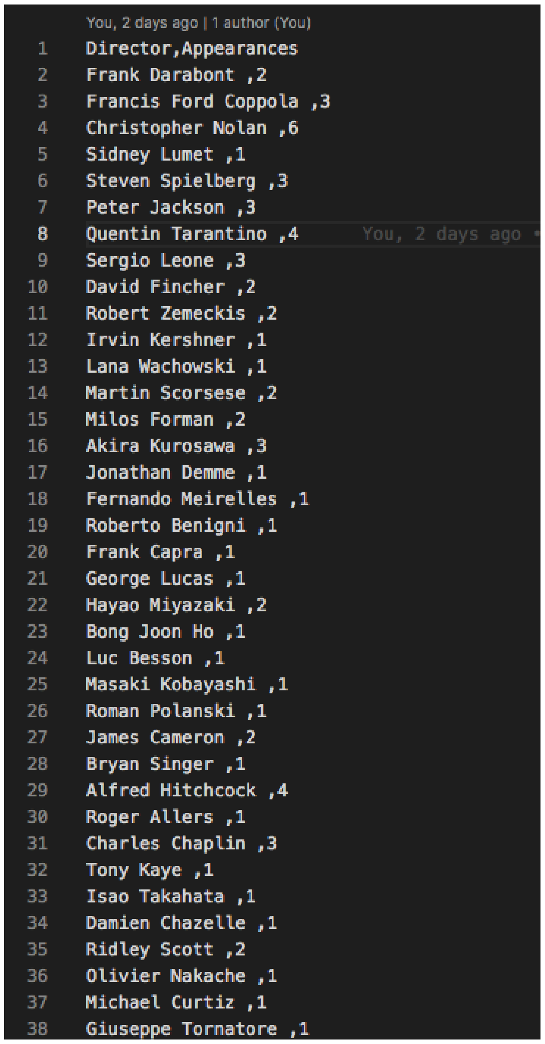
Achieved Goals

We gathered data from the top 100 movies and made a table in the database with each of these movies' year, rank, IMDb rating and director. We calculated the number of appearances each director had in the top 100. We made a table and CSV file that has the Director and number of appearances. Also, we calculated the average IMDb rating (8.479/10). We made a dictionary of which years were in the top 100 movies where the key was the year and value was how many movies were made in this year. We saw that 1994 was the most common year to have a movie in the top 100 movies. We then gathered data from the top 100 movies with the highest box office. We calculated the average box office from these 100 movies which was $429 million. We then made intervals based on $100 million to see where most of the top 100 movies with the highest box office would fall into. Based on our bar graph, most of these movies had a box office between $200-$300 million.

Problems Faced

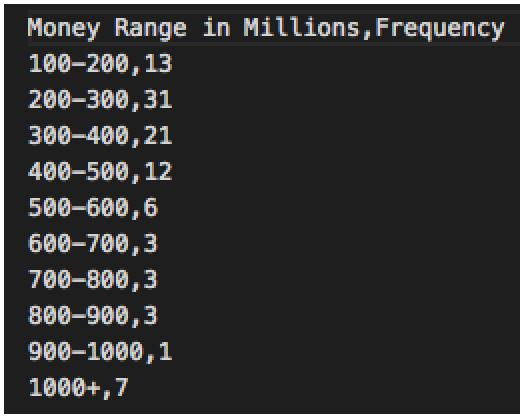
Upon completing the imdb api, we attempted to run the code on both computers and were surprised when it did not work. We were repeatedly getting an error saying “150 tries out of 100”. At first we did not know what this error meant, however we realized that the api key could only be used 100 times per day. We both had to make a free account on the imdb api so we could both get our own key. Another problem we faced was that it was difficult working on two separate computers so we did most of the work on Jennie’s. Maya and Jennie met up in person to do the project.

File with Data Calculation



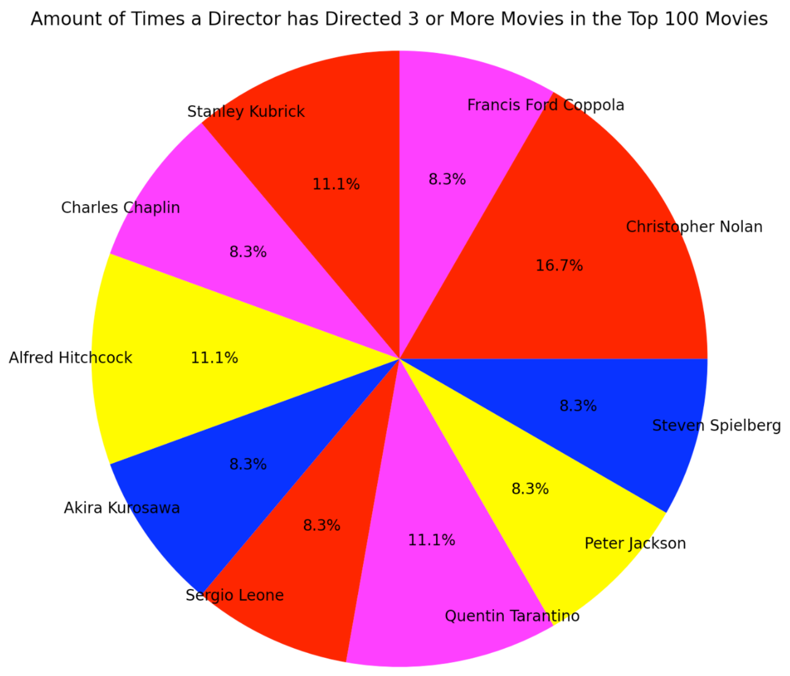
```
You, 2 days ago | 1 author (You)
Director,Appearances
Frank Darabont ,2
Francis Ford Coppola ,3
Christopher Nolan ,6
Sidney Lumet ,1
Steven Spielberg ,3
Peter Jackson ,3
Quentin Tarantino ,4      You, 2 days ago •
Sergio Leone ,3
David Fincher ,2
Robert Zemeckis ,2
Irvin Kershner ,1
Lana Wachowski ,1
Martin Scorsese ,2
Milos Forman ,2
Akira Kurosawa ,3
Jonathan Demme ,1
Fernando Meirelles ,1
Roberto Benigni ,1
Frank Capra ,1
George Lucas ,1
Hayao Miyazaki ,2
Bong Joon Ho ,1
Luc Besson ,1
Masaki Kobayashi ,1
Roman Polanski ,1
James Cameron ,2
Bryan Singer ,1
Alfred Hitchcock ,4
Roger Allers ,1
Charles Chaplin ,3
Tony Kaye ,1
Isao Takahata ,1
Damien Chazelle ,1
Ridley Scott ,2
Olivier Nakache ,1
Michael Curtiz ,1
Giuseppe Tornatore ,1
```

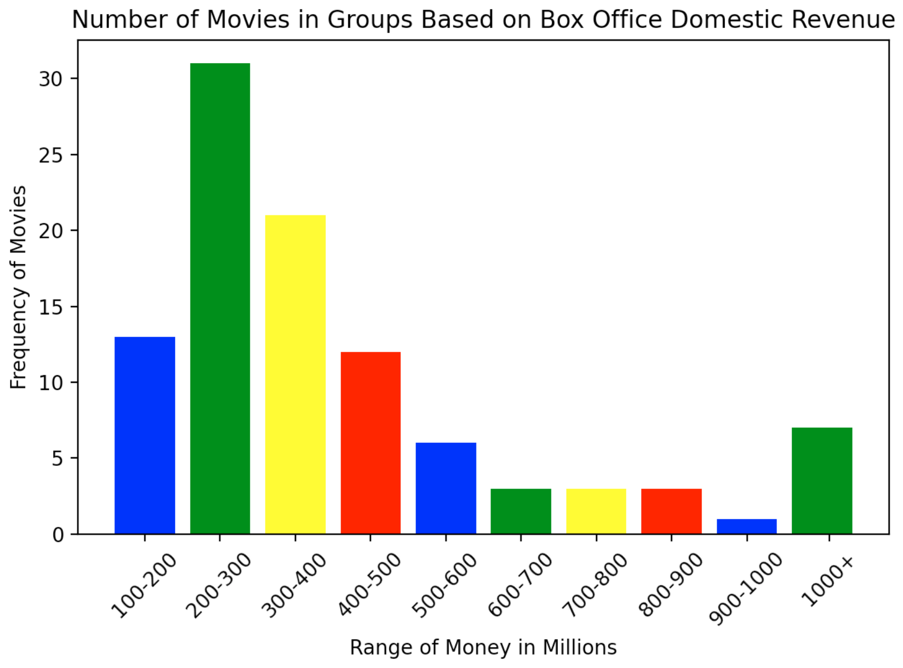
From the data in the database, we made calculations to see how many times each Director appeared in the top 100 movies from IMDB. Not all of the Directors could fit in this screenshot. On the left column you have the name of each director and on the right column you have the amount of appearances. We wanted to see which Directors were the most common.

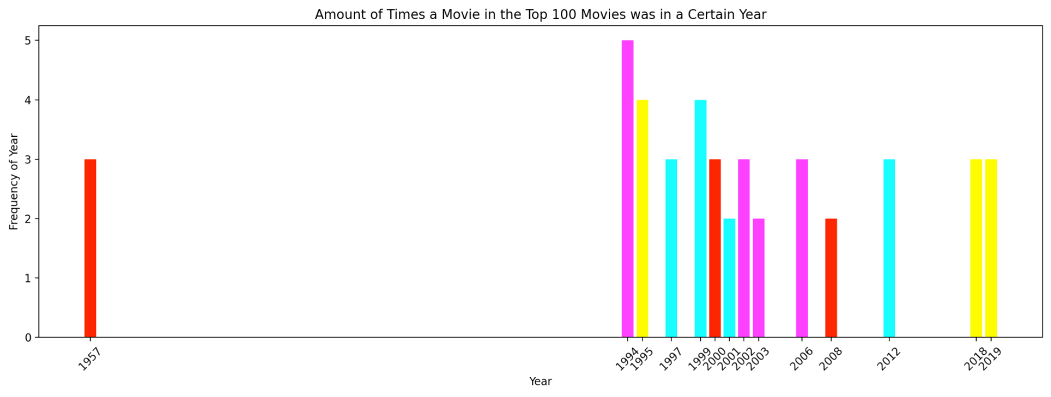

```
Money Range in Millions,Frequency
100-200,13
200-300,31
300-400,21
400-500,12
500-600,6
600-700,3
700-800,3
800-900,3
900-1000,1
1000+,7
```

From the data in the database, we made calculations to see how many millions of dollars the top 100 movies with the highest box office typically fell. On the left column, you have the different ranges of box offices (in millions of dollars). On the right column, you have the amount of movies that were part of this range from their box office. We wanted to see where most of the movies fell under. After calculating this, we found that the majority of the top 100 movies with the highest database had a box office between $200-$300 million.

Visualizations







Instructions for Running the Code

1. First, you have to run the imdbApi.py file four times so there are 100 rows in the Movies database. Then, run ultimateMovieRankings.py. Both python files will also produce visualizations, txt files or csv files.

Input and output for Each Function in the Code

IMDb Api Functions

1. Top100(key)
    a. Inputs an api key generated by the IMDb website and returns the first 100 items from the data.

2. getDirectors(key)

    a. Uses the key and calls upon the Top100 method and returns the name of the director for each movie

3. countDirectors(directors)
    a. Uses the director dictionary from getDirectors to find how many time the specific director has directed a movie in the top 100

4. director_pie(director_frequency)
    a. Uses the director frequency as an input generated from the countDirectors method to create a pie chart of the top directors and the amount of movies they directed in the top 100

5. setUpDatabase(db_name)
    a. Takes in the name of the database as a parameter and sets up the database in db sq3lite

6. setUpMoviesTable(data, cur, conn)
    a. Creates the movies database with the data returned from top100. It finds each variable we are searching for and adds the information to the database.

7. setUpDirectorsTable(director_dict, cur, conn)
    a. Uses the director_dict which is received by calling countDirectors to set up a second database after the movies one.
8. getAvgRating(data, cur, conn)
    a. Uses the SELECT function to find the average of all ratings in the IMDb database
9. getDictOfYears(data, cur, conn)
    a. Uses SELECT to access all the movie data from the database including the year the movie was produced. It then uses a dictionary to match how many times the year appeared in the data as well as which year it was. It returns the sorted data at the end
10. barchart_year_and_frequency(dictionary)
    a. Takes in the dictionary created in getDictOfYears and creates a visual of the year and the amount of times a movie was in the top 100 from that year.
11. title_and_dir(cur, conn)
    a. Uses the JOIN function to show the share in keys between both the directors and movies databases
12. director_freq_csv(dct, cur, conn, filename)
    a. Uses the data from the calculated number of appearances for the directors and puts it into a csv file

Ultimate Movie Rankings Functions

1. getMovies()
    a. Uses the website url and Beautiful soup to return a dictionary with the keys being the movie name and the values being the money they made in millions
2. setUpDatabase(db_name)
    a. Same as setting up database in IMDb api
3. setUpMoneyTable(money_per_movie_dict, cur, conn)
    a. Uses the return value from getMovies to create a table using the dictionary key value pairs.
4. detAvgMoney(data, cur, conn)
    a. Uses SELECT to get the values from the table with money, adds them all up and divides by the total number of elements in the list. It returns a float.
5. dctMoney(dct, cur, conn)
    a. Iterates through the dictionary created in getMovies to see which range the money the movie generated falls into. Returns a dictionary of the range and the number of movies in that range.
6. csvMoney(MoneyDict, filename, cur, conn)
    a. Creates a text file using the dictionary created from csvMoney
7. barchart_frequency_and_money(dct)
    a. Uses the data from dctMoney to create a bar chart

Resources

Date Issue	Description	Location of Resource	Result (did it solve the issue?
12/5/21	We wanted to create a Bar Chart for our UltimateMovieRankings python file. We were stuck on how to go about this.	https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html	We used this link to help us and it definitely worked. It helped us understand the terminology of bar charts and what to fix to make all the data look nice and make sense.
12/7/21	We were having trouble getting our pie chart to show the percentages of each director.	https://matplotlib.org/stable/gallery/pie_and_polar_charts/pie_features.html	We used this link and it definitely helped. We were able to not only label each part of the pie chart with the director name but we were able to put the percentage they took of the chart.

12/7/21	We had trouble changing the colors of the bar graph and its elements	https://www.python-graph-gallery.com/3-control-color-of-barplots	We were able to create an appealing visual after looking through the examples on the website
12/7/21	We forgot how to get rid of the $ in the money without having it be a string	https://stackoverflow.com/questions/265960/best-way-to-strip-punctuation-from-a-string	We were able to get rid of the punctuation and cast the value to a float.
12/8/21	We had trouble finding the values for find_all in the Ultimate Movie Website.	https://tutorme.com/?utm_source=google&utm_medium=cpc&utm_campaign=br_tutorme&utm_term=tutor_me&utm_content=exact&utm_campaign=9211691181&utm_source=google&utm_medium=paid_search&utm_term=e_tutor%20me&adgroupid=92965763933&geoid=9016852&matchtype=e&device=c&gclid=Cj0KCQiAzMGNBhCyARIsANpUkzPEhir2Dq4N5GiaE1ymUAJOfo2ps8PgNmCWZT4JB49-xDXribXVMW0aAkddEALw_wcB	Yes, we were connected with a live tutor who helped us look through the html file to find the places where the title, director, and revenue were located.
12/8/21	We had trouble changing the values that were displayed on the pie chart as well as showing all of them	https://stackoverflow.com/questions/41088236/how-to-have-actual-values-in-matplotlib-pie-chart-displayed	We were able to display the percentages as well as the name of the director. We were also able to change the layout and colors of the chart.
12/8/21	We had trouble sorting the items in the dictionary that we wanted to add to our csv file	https://www.geeksforgeeks.org/how-to-sort-data-by-column-in-a-csv-file-in-python/	We were able to sort the dictionary by director appearances, with the top most being at the top and least being at the bottom.
12/8/21	We had trouble inputting only the Director from the IMDB API into the database and excluding the rest of the cast and crew.	https://pythonspot.com/tag/sql/ And	Instead of having a problem with SQL, we realized we had to split the string at '(dir.)' and use an index of 0 to get

		https://www.w3schools.com/python/ref_string_split.asp	only the Director.