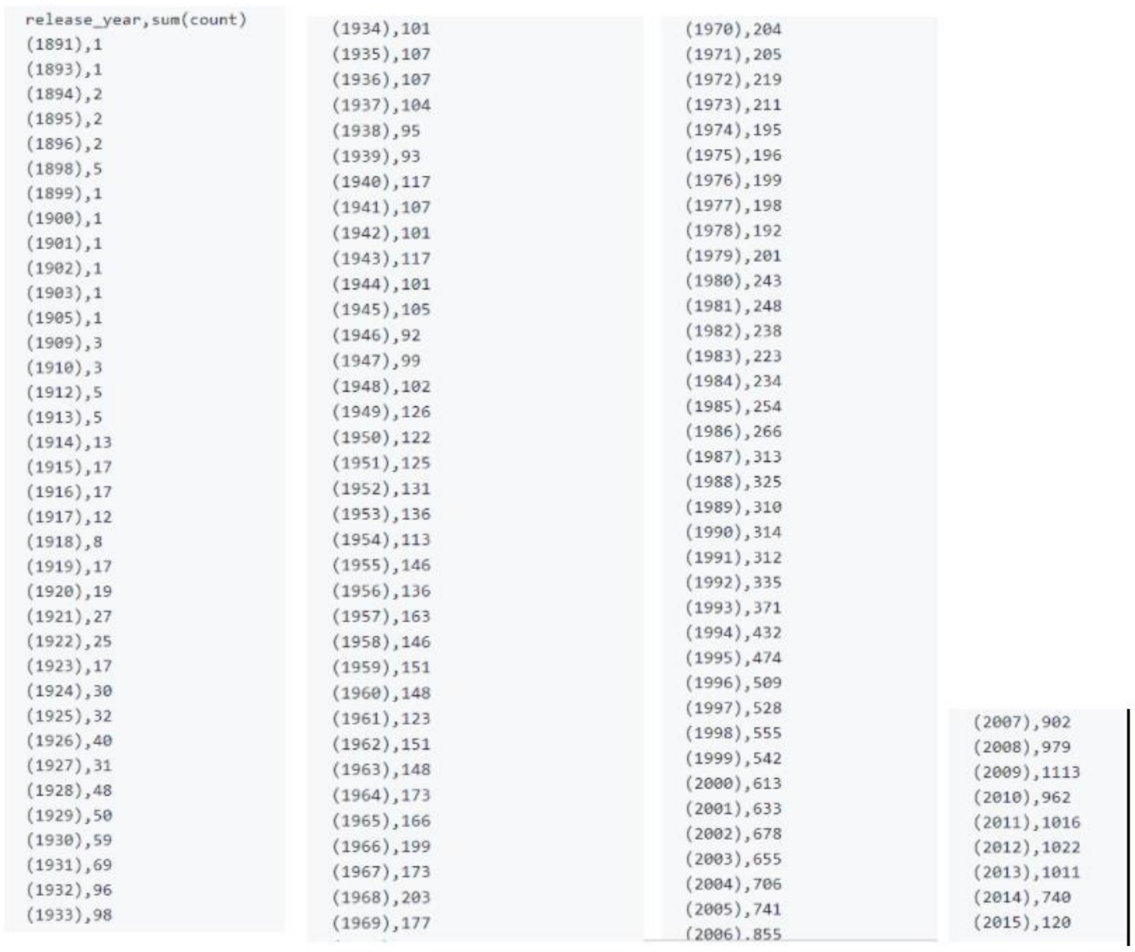
CS555 HW3

1) load up the movie.csv data frame and filter out any that don't have a year in the title, create a new column that pulls out the year that the movie was released with a count of 1, Select just release year and count to reduce dataset, group by release year and sum up the count and organize by year.



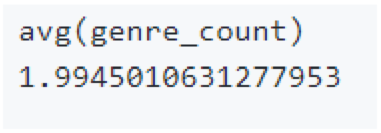
```
release_year,sum(count)
(1891),1
(1893),1
(1894),2
(1895),2
(1896),2
(1898),5
(1899),1
(1900),1
(1901),1
(1902),1
(1903),1
(1905),1
(1909),3
(1910),3
(1912),5
(1913),5
(1914),13
(1915),17
(1916),17
(1917),12
(1918),8
(1919),17
(1920),19
(1921),27
(1922),25
(1923),17
(1924),30
(1925),32
(1926),40
(1927),31
(1928),48
(1929),50
(1930),59
(1931),69
(1932),96
(1933),98
```

```
(1934),101
(1935),107
(1936),107
(1937),104
(1938),95
(1939),93
(1940),117
(1941),107
(1942),101
(1943),117
(1944),101
(1945),105
(1946),92
(1947),99
(1948),102
(1949),126
(1950),122
(1951),125
(1952),131
(1953),136
(1954),113
(1955),146
(1956),136
(1957),163
(1958),146
(1959),151
(1960),148
(1961),123
(1962),151
(1963),148
(1964),173
(1965),166
(1966),199
(1967),173
(1968),203
(1969),177
```

```
(1970),204
(1971),205
(1972),219
(1973),211
(1974),195
(1975),196
(1976),199
(1977),198
(1978),192
(1979),201
(1980),243
(1981),248
(1982),238
(1983),223
(1984),234
(1985),254
(1986),266
(1987),313
(1988),325
(1989),310
(1990),314
(1991),312
(1992),335
(1993),371
(1994),432
(1995),474
(1996),509
(1997),528
(1998),555
(1999),542
(2000),613
(2001),633
(2002),678
(2003),655
(2004),706
(2005),741
(2006),855
```

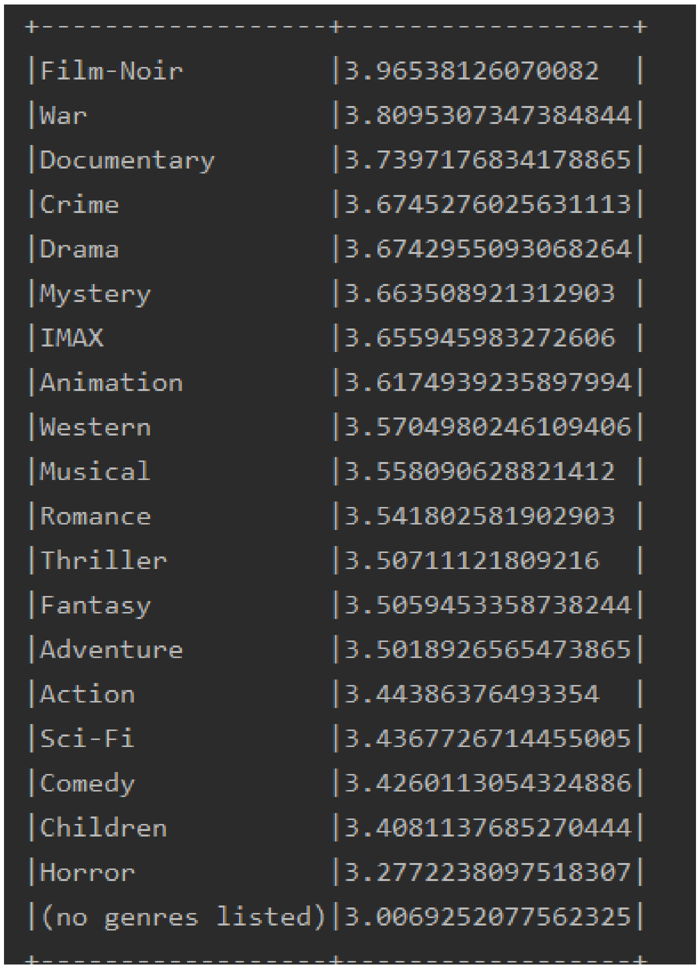
```
(2007),902
(2008),979
(2009),1113
(2010),962
(2011),1016
(2012),1022
(2013),1011
(2014),740
(2015),120
```

2) Load the movies.csv, create a new column that contains the genre count that gets the size of the array created from split by "|". Then take the average of the genre count along the whole column and select it to reduce unwanted data.



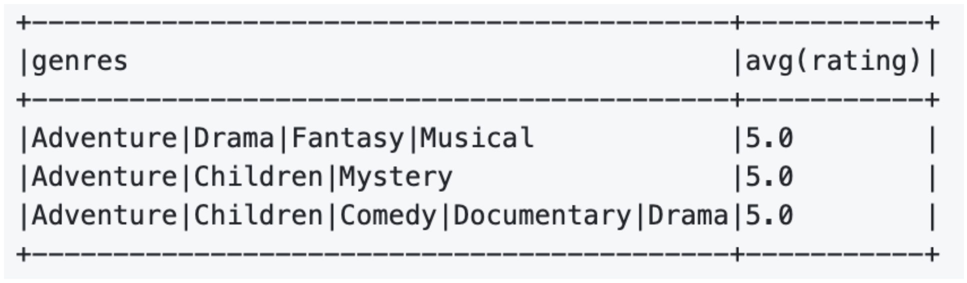
```
avg(genre_count)
1.9945010631277953
```

3) First get the unique genres by loading movies.csv. Split the genres column by the "|". Then I exploded the genre list to create rows for each genre on every movie. Then I went through and dropped all duplicates on the genre type column. This left just the unique genres. Then for each unique genre, I joined the movies.csv data frame with the ratings.csv data frame after removing the unwanted columns. Then did a left join of the ratings with the movies data frame and filtered out all movies that were not for the specified genre. Then compressed all the genre's ratings by taking their average rating for the specified genre.



```
+------------------+------------------+
|Film-Noir         |3.96538126070082  |
|War               |3.8095307347384844|
|Documentary       |3.7397176834178865|
|Crime             |3.6745276025631113|
|Drama             |3.6742955093068264|
|Mystery           |3.663508921312903 |
|IMAX              |3.655945983272606 |
|Animation         |3.6174939235897994|
|Western           |3.5704980246109406|
|Musical           |3.558090628821412 |
|Romance           |3.541802581902903 |
|Thriller          |3.50711121809216  |
|Fantasy           |3.5059453358738244|
|Adventure         |3.5018926565473865|
|Action            |3.44386376493354  |
|Sci-Fi            |3.4367726714455005|
|Comedy            |3.4260113054324886|
|Children          |3.4081137685270444|
|Horror            |3.2772238097518307|
|(no genres listed)|3.0069252077562325|
+------------------+------------------+
```

4) First selected only “movieId” and “rating” columns from ratings.csv, and “movieId” and “genres” from movies.csv as two dataframes. Then did an inner-join on “movieId”, dropping the column once it was finished. Finally, grouped by “genres”, taking the average as the aggregation function, sorted by descending, and took the top 3.



```
+------------------------------------------+-----------+
|genres                                    |avg(rating)|
+------------------------------------------+-----------+
|Adventure|Drama|Fantasy|Musical           |5.0        |
|Adventure|Children|Mystery                |5.0        |
|Adventure|Children|Comedy|Documentary|Drama|5.0        |
+------------------------------------------+-----------+
```

5) First loaded up the tags.csv data frame and filtered all the ones that had case insensitive comedy inside of the tag description. Then filtered down to just tags and movie id. Then created a new column of ones as a counter column. Then do a sum of count for all that is left in the data frame.

```
sum(isComedy)
1375
```

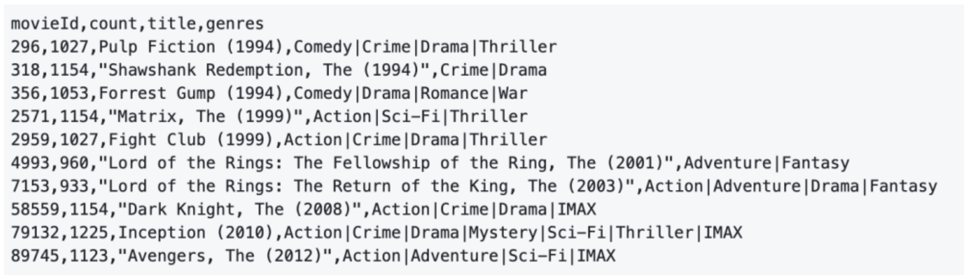
6) First selected the “genres” column from movies.csv, dropping the others. Then created an array column by splitting on the “|” delimiter, and exploded into separate rows. Then grouped by genres, taking the count as the aggregation. Finally, dropped any rows which had no genres, which were represented as “(no genres listed)”.

```
genre,count
Western,676
War,1194
Thriller,4178
Sci-Fi,1743
Romance,4127
Mystery,1514
Musical,1036
IMAX,196
Horror,2611
Film-Noir,330
Fantasy,1412
Drama,13344
Documentary,2471
Crime,2939
Comedy,8374
Children,1139
Animation,1027
Adventure,2329
Action,3520
```

7) Our question here was “Given a time period, what is the most popular movie?”
The inputs to the function are the time bounds and N for the N most popular movies.

Example: begin = “03/02/2012”, end = “08/27/2012”, N = 10:

We calculated this by filtering by ratings which fall between those time bounds, then grouping by movieId, taking the count as the aggregation function on the movieId. This gives us how many times a movie was reviewed within a given time window. Then, we sort by the count descending, limiting to N, and finally join with the movieInfo dataframe to get the movie's genres and title.



```
movieId,count,title,genres
296,1027,Pulp Fiction (1994),Comedy|Crime|Drama|Thriller
318,1154,"Shawshank Redemption, The (1994)",Crime|Drama
356,1053,Forrest Gump (1994),Comedy|Drama|Romance|War
2571,1154,"Matrix, The (1999)",Action|Sci-Fi|Thriller
2959,1027,Fight Club (1999),Action|Crime|Drama|Thriller
4993,960,"Lord of the Rings: The Fellowship of the Ring, The (2001)",Adventure|Fantasy
7153,933,"Lord of the Rings: The Return of the King, The (2003)",Action|Adventure|Drama|Fantasy
58559,1154,"Dark Knight, The (2008)",Action|Crime|Drama|IMAX
79132,1225,Inception (2010),Action|Crime|Drama|Mystery|Sci-Fi|Thriller|IMAX
89745,1123,"Avengers, The (2012)",Action|Adventure|Sci-Fi|IMAX
```