Problem 1

In [2]:

```
# Reading the dataset using pandas.

import pandas as pd
ratings = pd.read_csv('ratings.csv')
tags = pd.read_csv('tags.csv')
movies = pd.read_csv('movies.csv')
```

Problem 2

In [4]:

```
# Extract the first row from tags and print its type
tags_first_row = tags.head(1)
print(tags_first_row)
type(tags_first_row)
```

```
   userId  movieId          tag   timestamp
0      18     4141  Mark Waters  1240597180
```

Out[4]:

```
pandas.core.frame.DataFrame
```

Problem 3

In [5]:

```
# Extract row 0, 11, 2000 from tags DataFrame
tags.iloc[[1, 11, 2000], :]
```

Out[5]:

	userId	movieId	tag	timestamp
1	65	208	dark hero	1368150078
11	65	1783	noir thriller	1368149983
2000	910	68554	conspiracy theory	1368043943

Problem 4

In [12]:

```
# Print index, columns of the DataFrame.
print("Index, columns of the ratings dataframe:::\n")
ratings.info()
print("\n Index, columns of the tags dataframe::: \n")
tags.info()
print("\n index, columns of the movies dataframe::: \n")
movies.info()
```

```
Index, columns of the ratings dataframe:::

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000263 entries, 0 to 20000262
Data columns (total 4 columns):
userId       int64
movieId      int64
rating       float64
timestamp    int64
dtypes: float64(1), int64(3)
memory usage: 610.4 MB

 Index, columns of the tags dataframe:::

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 465564 entries, 0 to 465563
Data columns (total 4 columns):
userId       465564 non-null int64
movieId      465564 non-null int64
tag          465548 non-null object
timestamp    465564 non-null int64
dtypes: int64(3), object(1)
memory usage: 14.2+ MB

 index, columns of the movies dataframe:::

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27278 entries, 0 to 27277
Data columns (total 3 columns):
movieId    27278 non-null int64
title      27278 non-null object
genres     27278 non-null object
dtypes: int64(1), object(2)
memory usage: 639.4+ KB
```

Problem 5

In [15]:

```
# Calculate descriptive statistics for the 'ratings' column of the ratings DataF
rame. Verify using describe().

print("Descriptive statistics of the rating dataset using predefined methods")
print("countc:",ratings.rating.count())
print("Mean:",ratings.rating.mean())
print("Std:",ratings.rating.std())
print("Min:",ratings.rating.min())
print("Max:",ratings.rating.max())
print("Descriptive statistics of the rating dataset using describe")
ratings.rating.describe()
```

```
Descriptive statistics of the rating dataset using predefined method
s
countc: 20000263
Mean: 3.5255285642993797
Std: 1.051988919275684
Min: 0.5
Max: 5.0
Descriptive statistics of the rating dataset using describe
```

Out[15]:

```
count    2.000026e+07
mean     3.525529e+00
std      1.051989e+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
```

Problem 6

In [22]:

```
# Filter out ratings with rating > 5
res = ratings[ratings["rating"] > 5]
print(res)
print("Number of records with rating > 5 is: {}".format(res.size))
```

```
Empty DataFrame
Columns: [userId, movieId, rating, timestamp]
Index: []
Number of records with rating > 5 is: 0
```

Problem 7

In [25]:

```
# Find how many null values, missing values are present. Deal with them. Print o
ut how many rows have been modified.

print("No of NaN, missing values in each column of a ratings DataFrame")
print(ratings.isnull().sum(), sep='\n')
print("No of NaN, missing values in each column of a tags DataFrame")
print(tags.isnull().sum(), sep='\n')
print("No of NaN, missing values in each column of a movies DataFrame")
print(movies.isnull().sum(), sep='\n')

print("Filling Null or missing values with ASSIGNMENT")
Dataset=tags.fillna("ASSIGNMENT")
print("Modified row count is\n",Dataset[Dataset["tag"]=="ASSIGNMENT"].count())
```

```
No of NaN, missing values in each column of a ratings DataFrame
userId       0
movieId      0
rating       0
timestamp    0
dtype: int64
No of NaN, missing values in each column of a tags DataFrame
userId        0
movieId       0
tag          16
timestamp     0
dtype: int64
No of NaN, missing values in each column of a movies DataFrame
movieId    0
title      0
genres     0
dtype: int64
Filling Null or missing values with ASSIGNMENT
Modified row count is
 userId       16
movieId      16
tag          16
timestamp    16
dtype: int64
```

Problem 8

In [60]:

```
# Filter out movies from the movies DataFrame that are of type 'Animation'.

def filterGener(geners, gener_anmimation):
    res_lst = []
    for gen in geners:
        if(gen.find(gener_anmimation) != -1):
            res_lst.append(True)
        else:
            res_lst.append(False)
    return res_lst

movies[filterGener(movies.genres, 'Animation')]
```

Out[60]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
12	13	Balto (1995)	Adventure|Animation|Children
47	48	Pocahontas (1995)	Animation|Children|Drama|Musical|Romance
236	239	Goofy Movie, A (1995)	Animation|Children|Comedy|Romance
241	244	Gumby: The Movie (1995)	Animation|Children
310	313	Swan Princess, The (1994)	Animation|Children
360	364	Lion King, The (1994)	Adventure|Animation|Children|Drama|Musical|IMAX
388	392	Secret Adventures of Tom Thumb, The (1993)	Adventure|Animation
547	551	Nightmare Before Christmas, The (1993)	Animation|Children|Fantasy|Musical
553	558	Pagemaster, The (1994)	Action|Adventure|Animation|Children|Fantasy
582	588	Aladdin (1992)	Adventure|Animation|Children|Comedy|Musical
588	594	Snow White and the Seven Dwarfs (1937)	Animation|Children|Drama|Fantasy|Musical
589	595	Beauty and the Beast (1991)	Animation|Children|Fantasy|Musical|Romance|IMAX
590	596	Pinocchio (1940)	Animation|Children|Fantasy|Musical
604	610	Heavy Metal (1981)	Action|Adventure|Animation|Horror|Sci-Fi
610	616	Aristocats, The (1970)	Animation|Children
624	631	All Dogs Go to Heaven 2 (1996)	Adventure|Animation|Children|Fantasy|Musical|R...
653	661	James and the Giant Peach (1996)	Adventure|Animation|Children|Fantasy|Musical
664	673	Space Jam (1996)	Adventure|Animation|Children|Comedy|Fantasy|Sc...
697	709	Oliver & Company (1988)	Adventure|Animation|Children|Comedy|Musical
708	720	Wallace & Gromit: The Best of Aardman Animatio...	Adventure|Animation|Comedy
728	741	Ghost in the Shell (Kôkaku kidôtai) (1995)	Animation|Sci-Fi
732	745	Wallace & Gromit: A Close Shave (1995)	Animation|Children|Comedy
770	783	Hunchback of Notre Dame, The (1996)	Animation|Children|Drama|Musical|Romance
871	888	Land Before Time III: The Time of the Great Gi...	Adventure|Animation|Children|Musical
1003	1022	Cinderella (1950)	Animation|Children|Fantasy|Musical|Romance
1004	1023	Winnie the Pooh and the Blustery Day (1968)	Animation|Children|Musical
1005	1024	Three Caballeros, The (1945)	Animation|Children|Musical
1006	1025	Sword in the Stone, The (1963)	Animation|Children|Fantasy|Musical
1010	1029	Dumbo (1941)	Animation|Children|Drama|Musical
...	...	...	...

	movieId	title	genres
27007	129830	Animals United (2011)	Animation|Children|Comedy
27008	129834	Tom and Jerry: The Lost Dragon (2014)	Animation|Children|Comedy
27018	129859	Mardock Scramble: The Second Combustion (2011)	Animation|Sci-Fi
27054	130034	Stand by Me Doraemon (2014)	Animation|Children|Drama|Fantasy
27070	130075	Frozen Fever (2015)	Adventure|Animation
27103	130394	The Mascot (1934)	Animation
27107	130402	Cardcaptor Sakura: The Sealed Card (2000)	Adventure|Animation|Comedy|Fantasy|Romance
27129	130506	Berserk: The Golden Age Arc 2 - The Battle for...	Action|Animation|Fantasy
27130	130508	Berserk: The Golden Age Arc - The Egg of the K...	Action|Adventure|Animation|Fantasy|Horror
27131	130510	Berserk: The Golden Age Arc 3 - Descent (2013)	Action|Animation|Fantasy
27135	130518	The Amazing Screw-On Head (2006)	Action|Adventure|Animation|Comedy|Sci-Fi
27136	130520	Home (2015)	Animation|Children|Comedy|Fantasy|Sci-Fi
27137	130522	The Brave Little Toaster Goes to Mars (1998)	Animation|Children
27155	130644	The Garden of Sinners - Chapter 5: Paradox Par...	Animation
27205	131058	Santa's Apprentice (2010)	Animation|Children
27208	131066	Ronal the Barbarian (2011)	Adventure|Animation|Fantasy
27215	131080	Cinderella III: A Twist in Time (2007)	Animation|Children|Fantasy|Romance
27218	131086	The Little Polar Bear: Lars and the Little Tig...	Animation|Children
27221	131092	Mickey, Donald, Goofy: The Three Musketeers (2...	Adventure|Animation|Children|Comedy
27222	131094	Rudolph the Red-Nosed Reindeer: The Movie (1998)	Animation|Children
27224	131098	Saving Santa (2013)	Animation|Children|Comedy
27226	131102	Lucky Luke: The Ballad of the Daltons (1978)	Animation|Children|Comedy|Western
27228	131106	Casper's Haunted Christmas (2000)	Animation|Children
27235	131120	Superstar Goofy (1991)	Animation|Children|Comedy
27240	131130	Tom and Jerry: A Nutcracker Tale (2007)	Animation|Comedy
27241	131132	Kleines Arschloch - Der Film (1997)	Animation|Comedy
27247	131144	Werner - Das muss kesseln!!! (1996)	Animation|Comedy
27248	131146	Werner - Volles Rooäää (1999)	Animation|Comedy

	movieId	title	genres
27269	131243	Werner - Gekotzt wird später (2003)	Animation|Comedy
27270	131248	Brother Bear 2 (2006)	Adventure|Animation|Children|Comedy|Fantasy

1027 rows × 3 columns

Problem 9

In [62]:

```
# Find the average rating of movies.
print("Average rating of the movies:",ratings.rating.mean())
```

```
Average rating of the movies: 3.5255285642993797
```

Problem 10

In [99]:

```
# Perform an inner join of movies and tags based on movieId.
movies.join(tags,on='movieId',how='inner', lsuffix='Mov')
```

Out[99]:

	movieIdMov	title	genres	userId	movieId
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	65	208
1	2	Jumanji (1995)	Adventure|Children|Fantasy	65	353
2	3	Grumpier Old Men (1995)	Comedy|Romance	65	521
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	65	592
4	5	Father of the Bride Part II (1995)	Comedy	65	668
5	6	Heat (1995)	Action|Crime|Thriller	65	898
6	7	Sabrina (1995)	Comedy|Romance	65	1248
7	8	Tom and Huck (1995)	Adventure|Children	65	1391
8	9	Sudden Death (1995)	Action	65	1617
9	10	GoldenEye (1995)	Action|Adventure|Thriller	65	1694
10	11	American President, The (1995)	Comedy|Drama|Romance	65	1783
11	12	Dracula: Dead and Loving It (1995)	Comedy|Horror	65	2022
12	13	Balto (1995)	Adventure|Animation|Children	65	2193
13	14	Nixon (1995)	Drama	65	2353
14	15	Cutthroat Island (1995)	Action|Adventure|Romance	65	2662
15	16	Casino (1995)	Crime|Drama	65	2726
16	17	Sense and Sensibility (1995)	Drama|Romance	65	2840
17	18	Four Rooms (1995)	Comedy	65	3052
18	19	Ace Ventura: When Nature Calls (1995)	Comedy	65	5135
19	20	Money Train (1995)	Action|Comedy|Crime|Drama|Thriller	65	6539

	movieIdMov	title	genres	userId	movieId
20	21	Get Shorty (1995)	Comedy|Crime|Thriller	65	6874
21	22	Copycat (1995)	Crime|Drama|Horror|Mystery|Thriller	65	7013
22	23	Assassins (1995)	Action|Crime|Thriller	65	7318
23	24	Powder (1995)	Drama|Sci-Fi	65	8529
24	25	Leaving Las Vegas (1995)	Drama|Romance	65	8622
25	26	Othello (1995)	Drama	65	27803
26	27	Now and Then (1995)	Children|Drama	65	27866
27	28	Persuasion (1995)	Drama|Romance	65	48082
28	29	City of Lost Children, The (Cité des enfants p...	Adventure|Drama|Fantasy|Mystery|Sci-Fi	65	48082
29	30	Shanghai Triad (Yao a yao yao dao waipo qiao) ...	Crime|Drama	65	51884
...	...	...	...	...	...
27248	131146	Werner - Volles Rooäää (1999)	Animation|Comedy	34494	53956
27249	131148	What A Man (2011)	Comedy|Romance	34494	53956
27250	131150	7 Dwarves: The Forest Is Not Enough (2006)	Comedy	34494	56719
27251	131152	The Fat Spy (1966)	Comedy	34494	58998
27252	131154	Die Bademeister – Weiber, saufen, Leben retten...	Comedy	34494	58998
27253	131156	Ants in the Pants 2 (2002)	Comedy	34494	58998
27254	131158	Manta, Manta (1991)	Comedy	34494	58998

	movieIdMov	title	genres	userId	movieId
27255	131160	Oscar and the Lady in Pink (2009)	Drama	34494	58998
27256	131162	Por un puñado de besos (2014)	Drama|Romance	34494	58998
27257	131164	Vietnam in HD (2011)	War	34494	60069
27258	131166	WWII IN HD (2009)	(no genres listed)	34494	60069
27259	131168	Phoenix (2014)	Drama	34494	60408
27260	131170	Parallels (2015)	Sci-Fi	34494	61323
27261	131172	Closed Curtain (2013)	(no genres listed)	34494	61323
27262	131174	Gentlemen (2014)	Drama|Romance|Thriller	34494	61323
27263	131176	A Second Chance (2014)	Drama	34494	61323
27264	131180	Dead Rising: Watchtower (2015)	Action|Horror|Thriller	34494	64614
27265	131231	Standby (2014)	Comedy|Romance	34494	73587
27266	131237	What Men Talk About (2010)	Comedy	34494	76093
27267	131239	Three Quarter Moon (2011)	Comedy|Drama	34494	76093
27268	131241	Ants in the Pants (2000)	Comedy|Romance	34494	77191
27269	131243	Werner - Gekotzt wird später (2003)	Animation|Comedy	34494	77191
27270	131248	Brother Bear 2 (2006)	Adventure|Animation|Children|Comedy|Fantasy	34494	78321
27271	131250	No More School (2000)	Comedy	34494	79132
27272	131252	Forklift Driver Klaus: The First Day on the Jo...	Comedy|Horror	34494	79132
27273	131254	Kein Bund für's Leben (2007)	Comedy	34494	79132

	movieIdMov	title	genres	userId	movieId
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy	34494	79132
27275	131258	The Pirates (2014)	Adventure	34494	79132
27276	131260	Rentun Ruusu (2001)	(no genres listed)	34494	79132
27277	131262	Innocence (2014)	Adventure|Fantasy|Horror	34494	79590

27278 rows × 7 columns

Problem 11

In [100]:

```
def filterGener(geners, gener_anmimation):
    res_lst = []
    for gen in geners:
        if(gen.find(gener_anmimation) != -1):
            res_lst.append(True)
        else:
            res_lst.append(False)
    return res_lst

comedy_movies = movies[filterGener(movies.genres, 'Comedy')]

# Step 1 joining the comedy_movies and ratings data frames.
comedy_movies_rating = comedy_movies.join(ratings,on='movieId',how='inner',lsuff
ix='Mov')
#step 2 Displaying  5 movies which are Comedy in genre and have rating greater t
han 4
comedy_movies_rating_morethan_4 = comedy_movies_rating[comedy_movies_rating.rati
ng > 4.0]
#Step 3 Printing first 5 comedy movies having rating > 4
comedy_movies_rating_morethan_4.head(5)
```

Out[100]:

	movieIdMov	title	genres	userId	movieId	rating	timestamp
169	171	Jeffrey (1995)	Comedy|Drama	1	8636	4.5	1970-01-01
174	176	Living in Oblivion (1995)	Comedy	2	62	5.0	1970-01-01
178	180	Mallrats (1995)	Comedy|Romance	2	260	5.0	1970-01-01
181	183	Mute Witness (1994)	Comedy|Horror|Thriller	2	480	5.0	1970-01-01
192	194	Smoke (1995)	Comedy|Drama	2	1259	5.0	1970-01-01

Problem 12

In [77]:

```
# Split 'genres' into multiple columns.
df = movies['genres'].str.split('|').tolist()

generes_columns=pd.DataFrame(df,columns=['generes1', 'generes2','generes3','gene
res4','generes5','generes6','generes7','generes8','generes9','generes10'])
generes_columns
```

Out[77]:

	generes1	generes2	generes3	generes4	generes5	generes6	generes7	generes8	g
0	Adventure	Animation	Children	Comedy	Fantasy	None	None	None	
1	Adventure	Children	Fantasy	None	None	None	None	None	
2	Comedy	Romance	None	None	None	None	None	None	
3	Comedy	Drama	Romance	None	None	None	None	None	
4	Comedy	None	None	None	None	None	None	None	
5	Action	Crime	Thriller	None	None	None	None	None	
6	Comedy	Romance	None	None	None	None	None	None	
7	Adventure	Children	None	None	None	None	None	None	
8	Action	None	None	None	None	None	None	None	
9	Action	Adventure	Thriller	None	None	None	None	None	
10	Comedy	Drama	Romance	None	None	None	None	None	
11	Comedy	Horror	None	None	None	None	None	None	
12	Adventure	Animation	Children	None	None	None	None	None	
13	Drama	None	None	None	None	None	None	None	
14	Action	Adventure	Romance	None	None	None	None	None	
15	Crime	Drama	None	None	None	None	None	None	
16	Drama	Romance	None	None	None	None	None	None	
17	Comedy	None	None	None	None	None	None	None	
18	Comedy	None	None	None	None	None	None	None	
19	Action	Comedy	Crime	Drama	Thriller	None	None	None	
20	Comedy	Crime	Thriller	None	None	None	None	None	
21	Crime	Drama	Horror	Mystery	Thriller	None	None	None	
22	Action	Crime	Thriller	None	None	None	None	None	
23	Drama	Sci-Fi	None	None	None	None	None	None	
24	Drama	Romance	None	None	None	None	None	None	
25	Drama	None	None	None	None	None	None	None	
26	Children	Drama	None	None	None	None	None	None	
27	Drama	Romance	None	None	None	None	None	None	
28	Adventure	Drama	Fantasy	Mystery	Sci-Fi	None	None	None	
29	Crime	Drama	None	None	None	None	None	None	
...	...	...	...	...	...	...	...	...	
27248	Animation	Comedy	None	None	None	None	None	None	
27249	Comedy	Romance	None	None	None	None	None	None	
27250	Comedy	None	None	None	None	None	None	None	
27251	Comedy	None	None	None	None	None	None	None	
27252	Comedy	None	None	None	None	None	None	None	

	generes1	generes2	generes3	generes4	generes5	generes6	generes7	generes8	g
27253	Comedy	None	None	None	None	None	None	None	
27254	Comedy	None	None	None	None	None	None	None	
27255	Drama	None	None	None	None	None	None	None	
27256	Drama	Romance	None	None	None	None	None	None	
27257	War	None	None	None	None	None	None	None	
27258	(no genres listed)	None	None	None	None	None	None	None	
27259	Drama	None	None	None	None	None	None	None	
27260	Sci-Fi	None	None	None	None	None	None	None	
27261	(no genres listed)	None	None	None	None	None	None	None	
27262	Drama	Romance	Thriller	None	None	None	None	None	
27263	Drama	None	None	None	None	None	None	None	
27264	Action	Horror	Thriller	None	None	None	None	None	
27265	Comedy	Romance	None	None	None	None	None	None	
27266	Comedy	None	None	None	None	None	None	None	
27267	Comedy	Drama	None	None	None	None	None	None	
27268	Comedy	Romance	None	None	None	None	None	None	
27269	Animation	Comedy	None	None	None	None	None	None	
27270	Adventure	Animation	Children	Comedy	Fantasy	None	None	None	
27271	Comedy	None	None	None	None	None	None	None	
27272	Comedy	Horror	None	None	None	None	None	None	
27273	Comedy	None	None	None	None	None	None	None	
27274	Comedy	None	None	None	None	None	None	None	
27275	Adventure	None	None	None	None	None	None	None	
27276	(no genres listed)	None	None	None	None	None	None	None	
27277	Adventure	Fantasy	Horror	None	None	None	None	None	

27278 rows × 10 columns

Problem 13

In [101]:

```
# Extract year from title e.g. (1995).
df=movies['title'].str[-5:-1]
df
```

Out[101]:

```
0        1995
1        1995
2        1995
3        1995
4        1995
5        1995
6        1995
7        1995
8        1995
9        1995
10       1995
11       1995
12       1995
13       1995
14       1995
15       1995
16       1995
17       1995
18       1995
19       1995
20       1995
21       1995
22       1995
23       1995
24       1995
25       1995
26       1995
27       1995
28       1995
29       1995
         ...
27248    1999
27249    2011
27250    2006
27251    1966
27252    1999
27253    2002
27254    1991
27255    2009
27256    2014
27257    2011
27258    2009
27259    2014
27260    2015
27261    2013
27262    2014
27263    2014
27264    2015
27265    2014
27266    2010
27267    2011
27268    2000
27269    2003
27270    2006
27271    2000
27272    2001
27273    2007
27274    2002
27275    2014
```

```
27276    2001
27277    2014
Name: title, Length: 27278, dtype: object
```

Problem 14

In [102]:

```
# Select rows based on timestamps later than 2015-02-01.
import datetime

df=ratings
ratings['timestamp'] = pd.to_datetime(ratings['timestamp']).dt.date
given_date=datetime.date(2015,2,1)
res=ratings[ratings['timestamp'] > given_date]
res
```

Out[102]:

userId	movieId	rating	timestamp

Problem 15

In [103]:

```
# Sort the tags DataFrame based on timestamp.
tags_sorted = tags.sort_values(by='timestamp',ascending=True)
tags_sorted
```

Out[103]:

	userId	movieId	tag	timestamp
333932	100371	2788	monty python	1135429210
333927	100371	1732	coen brothers	1135429236
333924	100371	1206	stanley kubrick	1135429248
333923	100371	1193	jack nicholson	1135429371
333939	100371	5004	peter sellers	1135429399
333922	100371	47	morgan freeman	1135429412
333921	100371	47	brad pitt	1135429412
333936	100371	4011	brad pitt	1135429431
333937	100371	4011	guy ritchie	1135429431
333920	100371	32	bruce willis	1135429442
333933	100371	2858	kevin spacey	1135429466
333931	100371	2329	fuckoff nazi	1135429508
333930	100371	2329	edward norton	1135429508
333944	100371	6016	brazil	1135429520
15903	3797	110	overrated	1135589075
256663	75141	4642	Terrible	1135688095
15926	3797	8573	Nonlinear Surrealism	1135936223
68928	16050	38886	twee	1136036612
68924	16050	38886	eighties	1136036612
68925	16050	38886	family	1136036612
68926	16050	38886	new york	1136036612
68927	16050	38886	overrated	1136036612
68919	16050	7060	christian	1136039301
68921	16050	7060	musical	1136039301
68920	16050	7060	kitsch	1136039301
68922	16050	7060	religious	1136039301
68918	16050	6898	harsh	1136039368
68916	16050	6898	british	1136039368
68917	16050	6898	drugs	1136039368
81444	20513	36529	truth	1136242623
...	...	...	...	...
119997	28906	115149	violent	1427746031
119991	28906	115149	lack of story	1427746053
119993	28906	115149	over the top	1427746060
119989	28906	115149	cheesy	1427746074
405155	123297	128642	CLV	1427746337

	userId	movieId	tag	timestamp
399877	123297	4211	CLV	1427746709
400993	123297	6419	CLV	1427746864
401017	123297	6481	CLV	1427747220
405123	123297	118722	CLV	1427747321
400106	123297	4573	CLV	1427747462
150873	42640	115122	mockumentary	1427748778
150871	42640	115122	comedy	1427748797
150872	42640	115122	dark comedy	1427748808
392905	122523	5004	USA	1427752289
392904	122523	5004	party	1427752292
290536	88044	106782	unsatisfying ending	1427753000
290534	88044	106782	Orgies	1427753111
290529	88044	106782	ENFP	1427753147
290532	88044	106782	Left-Wing Propaganda	1427753266
290533	88044	106782	Left-Wing Writers	1427753621
290528	88044	106782	Economically Illiterate Writers	1427753739
290530	88044	106782	inaccurate	1427753806
290531	88044	106782	Jonah Hill	1427753849
290526	88044	106782	Amoral	1427753913
290527	88044	106782	crime	1427753921
290535	88044	106782	profanity	1427754096
288375	87797	215	Vienna	1427755801
158763	46072	3409	premonition	1427760726
158780	46072	6058	premonition	1427760764
339178	102853	115149	russian mafia	1427771352

465564 rows × 4 columns

In [ ]:

```

```