

▼ Chapter 5 - Exercise 1: GroupBy

- ▼ drinks.csv là tập tin cung cấp dữ liệu về tình hình tiêu thụ rượu bia ở các quốc gia theo từng châu lục

country: tên quốc gia tiêu thụ

beer_servings: số lượng bia tiêu thụ

spirit_servings: số lượng rượu mạnh ($\geq 40^0$)

wine_servings: số lượng rượu vang

total_litres_of_pure_alcohol: tổng số lít rượu nguyên chất

continent: châu lục

```
import pandas as pd
```

Câu 1: Đọc dữ liệu từ tập tin drinks.csv với index_col là cột đầu tiên của dữ liệu, và 1

```
drinks = pd.read_csv('drinks.csv', index_col=0)
```

Cho biết kiểu dữ liệu (type) của drink

```
print(type(drinks))
```

Cho biết ra kích thước (shape) của drink

```
print(drinks.shape)
```

Hiển thị tên các cột (columns) của drink

```
print(drinks.columns)
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
(193, 6)
```

```
Index(['country', 'beer_servings', 'spirit_servings', 'wine_servings',  
      'total_litres_of_pure_alcohol', 'continent'],  
      dtype='object')
```

Xem 5 dòng dữ liệu đầu tiên (head) của drink

```
drinks.head()
```

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_a:
0	Afghanistan	0	0	0	
1	Albania	89	132	54	
2	Algeria	25	0	14	
3	Andorra	245	138	312	
4	Angola	217	57	45	

Xem 5 dòng dữ liệu cuối cùng (tail) của drink

```
drinks.tail()
```

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_i
188	Venezuela	333	100	3	
189	Vietnam	111	2	1	
190	Yemen	6	0	0	
191	Zambia	32	19	4	
192	Zimbabwe	64	18	4	

Câu 2: Cho biết số lượng bia tiêu thụ trung bình ở mỗi châu lục
drinks.groupby('continent').beer_servings.mean()

```
continent
AF      61.471698
AS      37.045455
EU     193.777778
OC      89.687500
SA     175.083333
Name: beer_servings, dtype: float64
```

Câu 3: Cho biết thông tin thống kê tổng quát (describe) số lượng rượu vang được tiêu thụ
drinks.groupby('continent').wine_servings.describe()

	count	mean	std	min	25%	50%	75%	max
continent								
AF	53.0	16.264151	38.846419	0.0	1.0	2.0	13.00	233.0
AS	44.0	9.068182	21.667034	0.0	0.0	1.0	8.00	123.0
EU	45.0	142.222222	97.421738	0.0	59.0	128.0	195.00	370.0
OC	16.0	35.625000	64.555790	0.0	1.0	8.5	23.25	212.0
SA	12.0	62.416667	88.620189	1.0	3.0	12.0	98.50	221.0

Câu 4: Cho biết số lượng các loại bia và rượu tiêu thụ trung bình (mean) ở mỗi châu lục
drinks.groupby('continent').mean()

	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcoh
continent				
AF	61.471698	16.339623	16.264151	3.0075
AS	37.045455	60.840909	9.068182	2.1704
EU	193.777778	132.555556	142.222222	8.6177
OC	89.687500	58.437500	35.625000	3.3812
SA	175.083333	114.750000	62.416667	6.3083

Câu 5: Cho biết giá trị trung vị (median) của các loại bia và rượu tiêu thụ ở mỗi châu lục

```
drinks.groupby('continent').median()
```

	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
continent				
AF	32.0	3.0	2.0	2.0
AS	17.5	16.0	1.0	1.0
EU	219.0	122.0	128.0	10.0
OC	52.5	37.0	8.5	1.0
SA	162.5	108.5	12.0	6.0

```
# Câu 6: Cho biết số lượng rượu mạnh (spirit_servings) tiêu thụ trung bình, lớn nhất và nhỏ nhất
# Gợi ý: Dùng spirit_servings.agg(['mean', 'min', 'max'])
drinks.groupby('continent').spirit_servings.agg(['mean', 'min', 'max'])
```

	mean	min	max
continent			
AF	16.339623	0	152
AS	60.840909	0	326
EU	132.555556	0	373
OC	58.437500	0	254
SA	114.750000	25	302

```
# Câu 7: Sắp xếp dữ liệu tăng dần (sort_values) theo số lượng bia tiêu thụ
drinks_des = drinks.sort_values(by='beer_servings', ascending=True)
drinks_des
```

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_a
--	---------	---------------	-----------------	---------------	------------------------

```
# Cho biết 5 quốc gia có lượng tiêu thụ bia lớn nhất
drinks_des.tail()
```

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_a
135	Poland	343	215	56	
65	Germany	346	117	175	
62	Gabon	347	98	59	
45	Czech Republic	361	170	134	
117	Namibia	376	3	1	
62	Gabon	347	98	59	

```
# Cho biết 5 quốc gia có lượng tiêu thụ bia nhỏ nhất
drinks_des.head()
```

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_a
0	Afghanistan	0	0	0	
40	Cook Islands	0	254	74	
79	Iran	0	0	0	
90	Kuwait	0	0	0	
97	Libya	0	0	0	

▼ Chapter 5 - Exercise 2: Giao dịch chứng khoán

▼ Cho 3 file .csv sau:

- **stocks1.csv** : *date, symbol, open, high, low, close, volume* : chứa thông tin giao dịch chứng khoán các công ty khác nhau
- **stocks2.csv** : *date, symbol, open, high, low, close, volume* : chứa thông tin giao dịch chứng khoán các công ty khác nhau
- **companies.csv** : *name, employees, headquarters_city, headquarters_state* : chứa thông tin về trụ sở và số lượng nhân viên cho một công ty cụ thể

```
import pandas as pd
import numpy as np
```

```
# Câu 1a: Đọc file stocks1.csv => đưa dữ liệu vào stocks1
stocks1 = pd.read_csv('./stock_trading_data/stocks1.csv')
# Hiển thị 5 dòng dữ liệu đầu của stocks1
stocks1.head()
```

	date	symbol	open	high	low	close	volume
0	01-03-19	AMZN	1655.13	1674.26	1651.00	1671.73	4974877
1	04-03-19	AMZN	1685.00	1709.43	1674.36	1696.17	6167358
2	05-03-19	AMZN	1702.95	1707.80	1689.01	1692.43	3681522
3	06-03-19	AMZN	1695.97	NaN	NaN	1668.95	3996001
4	07-03-19	AMZN	1667.37	1669.75	1620.51	1625.95	4957017

```
# Hiển thị 5 dòng dữ liệu cuối của stocks1
stocks1.tail()
```

	date	symbol	open	high	low	close	volume
10	01-03-19	GOOG	1124.90	1142.97	1124.75	1140.99	1450316
11	04-03-19	GOOG	1146.99	1158.28	1130.69	1147.80	1446047
12	05-03-19	GOOG	1150.06	NaN	NaN	1162.03	1443174
13	06-03-19	GOOG	1162.49	1167.57	1155.49	1157.86	1099289
14	07-03-19	GOOG	1155.72	1156.76	1134.91	1143.30	1166559

```
# Cho biết kiểu dữ liệu (dtype) của các cột của stocks1
stocks1.dtypes
```

```
date      object
symbol    object
open      float64
high      float64
low       float64
close     float64
volume    int64
dtype: object
```

```
# Xem thông tin (info) của stocks1
stocks1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 7 columns):
date      15 non-null object
symbol    15 non-null object
open      15 non-null float64
high      13 non-null float64
low       13 non-null float64
close     15 non-null float64
volume    15 non-null int64
dtypes: float64(4), int64(1), object(2)
memory usage: 968.0+ bytes
```

```
# Câu 1b: Đọc file stocks2.csv => đưa dữ liệu vào stocks2
stocks2 = pd.read_csv('./stock_trading_data/stocks2.csv')
# Hiển thị 5 dòng dữ liệu đầu của stocks2
stocks2.head()
```

	date	symbol	open	high	low	close	volume
0	01-03-19	FB	162.60	163.132	161.69	162.28	11097770
1	04-03-19	FB	163.90	167.500	163.83	167.37	18894689
2	05-03-19	FB	167.37	171.880	166.55	171.26	28187890
3	06-03-19	FB	172.90	173.570	171.27	172.51	21531723
4	07-03-19	FB	171.50	171.740	167.61	169.13	18306504

```
# Hiển thị 5 dòng dữ liệu cuối của stocks2
stocks2.tail()
```

	date	symbol	open	high	low	close	volume
5	01-03-19	TSLA	306.94	307.1300	291.90	294.79	22911375
6	04-03-19	TSLA	298.12	299.0000	282.78	285.36	17096818
7	05-03-19	TSLA	282.00	284.0000	270.10	276.54	18764740
8	06-03-19	TSLA	276.48	281.5058	274.39	276.24	10335485
9	07-03-19	TSLA	278.84	284.7000	274.25	276.59	9442483

```
# Cho biết kiểu dữ liệu (dtype) của các cột của stocks2
stocks2.dtypes
```

```
date      object
symbol    object
open      float64
high      float64
low       float64
close     float64
volume    int64
dtype: object
```

```
# Xem thông tin (info) của stocks2
stocks2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 7 columns):
date      10 non-null object
symbol    10 non-null object
open      10 non-null float64
high      10 non-null float64
low       10 non-null float64
close     10 non-null float64
volume    10 non-null int64
dtypes: float64(4), int64(1), object(2)
memory usage: 688.0+ bytes
```

```
# Câu 1c: Đọc file companies.csv => đưa dữ liệu vào companies
companies = pd.read_csv('./stock_trading_data/companies.csv')
# Xem dữ liệu của companies
companies
```

	name	employees	headquarters_city	headquarters_state
0	AMZN	613300	Seattle	WA
1	GOOG	98771	Mountain View	CA
2	AAPL	132000	Cupertino	CA
3	FB	48268	Menlo Park	CA
4	TSLA	48016	Palo Alto	CA

```
# Cho biết kiểu dữ liệu (dtype) của các cột của companies
companies.dtypes
```

```
name      object
employees int64
headquarters_city object
headquarters_state object
dtype: object
```

```
# Xem thông tin (info) của companies
```

```
companies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
name                5 non-null object
employees           5 non-null int64
headquarters_city   5 non-null object
headquarters_state  5 non-null object
dtypes: int64(1), object(3)
memory usage: 288.0+ bytes
```

Câu 2: Cho biết trong stocks1 có dữ liệu Null hay không?

```
stocks1.isnull().any()
```

```
date      False
symbol     False
open       False
high       True
low        True
close      False
volume     False
dtype: bool
```

```
stocks1.groupby(['symbol'])['high'].transform(max)
```

```
0      1709.43
1      1709.43
2      1709.43
3      1709.43
4      1709.43
5       177.75
6       177.75
7       177.75
8       177.75
9       177.75
10     1167.57
11     1167.57
12     1167.57
13     1167.57
14     1167.57
Name: high, dtype: float64
```

Nếu có, hãy thay thế với quy tắc sau:

Nếu Null cột 'high' thì thay bằng giá trị max trên cột 'high' của mã chứng khoán đó

Nếu Null cột 'low' thì thay bằng giá trị min trên cột 'low' của mã chứng khoán đó

```
stocks1['high'] = stocks1['high'].fillna(stocks1.groupby('symbol')['high'].transform(max))
```

```
stocks1['low'] = stocks1['low'].fillna(stocks1.groupby('symbol')['low'].transform(min))
```

```
stocks1
```


	date	symbol	open	high	low	close	volume
0	01-03-19	AMZN	1655.13	1674.26	1651.00	1671.73	4974877
1	04-03-19	AMZN	1685.00	1709.43	1674.36	1696.17	6167358
2	05-03-19	AMZN	1702.95	1707.80	1689.01	1692.43	3681522
3	06-03-19	AMZN	1695.97	1709.43	1620.51	1668.95	3996001
4	07-03-19	AMZN	1667.37	1669.75	1620.51	1625.95	4957017
5	01-03-19	AAPL	174.28	175.15	172.89	174.97	25886167
6	04-03-19	AAPL	175.69	177.75	173.97	175.85	27436203
7	05-03-19	AAPL	175.94	176.00	174.54	175.53	19737419
8	06-03-19	AAPL	174.67	175.49	173.94	174.52	20810384
9	07-03-19	AAPL	173.87	174.44	172.02	172.50	24796374
10	01-03-19	GOOG	1124.90	1142.97	1124.75	1140.99	1450316
11	04-03-19	GOOG	1146.99	1158.28	1130.69	1147.80	1446047
12	05-03-19	GOOG	1150.06	1167.57	1124.75	1162.03	1443174

Câu 3: Tạo dataframe stocks bằng cách gộp stocks1 và stocks2 theo dòng
stocks = pd.concat([stocks1, stocks2], ignore_index=True)
Xem 15 dòng dữ liệu cuối của stocks
stocks.tail(15)

	date	symbol	open	high	low	close	volume
10	01-03-19	GOOG	1124.90	1142.9700	1124.75	1140.99	1450316
11	04-03-19	GOOG	1146.99	1158.2800	1130.69	1147.80	1446047
12	05-03-19	GOOG	1150.06	1167.5700	1124.75	1162.03	1443174
13	06-03-19	GOOG	1162.49	1167.5700	1155.49	1157.86	1099289
14	07-03-19	GOOG	1155.72	1156.7600	1134.91	1143.30	1166559
15	01-03-19	FB	162.60	163.1320	161.69	162.28	11097770
16	04-03-19	FB	163.90	167.5000	163.83	167.37	18894689
17	05-03-19	FB	167.37	171.8800	166.55	171.26	28187890
18	06-03-19	FB	172.90	173.5700	171.27	172.51	21531723
19	07-03-19	FB	171.50	171.7400	167.61	169.13	18306504
20	01-03-19	TSLA	306.94	307.1300	291.90	294.79	22911375
21	04-03-19	TSLA	298.12	299.0000	282.78	285.36	17096818
22	05-03-19	TSLA	282.00	284.0000	270.10	276.54	18764740
23	06-03-19	TSLA	276.48	281.5058	274.39	276.24	10335485
24	07-03-19	TSLA	278.84	284.7000	274.25	276.59	9442483

```
# Câu 4: Tạo dataframe stocks_companies bằng cách gộp stocks và companies
stocks_companies = stocks.merge(companies,left_on='symbol', right_on='name', how='inner')
# Xem 5 dòng dữ liệu đầu của stocks_companies
stocks_companies.head()
```

	date	symbol	open	high	low	close	volume	name	employees	headqu
0	01-03-19	AMZN	1655.13	1674.26	1651.00	1671.73	4974877	AMZN	613300	
1	04-03-19	AMZN	1685.00	1709.43	1674.36	1696.17	6167358	AMZN	613300	
2	05-03-19	AMZN	1702.05	1707.80	1680.01	1692.43	3681522	AMZN	613300	

```
# Câu 5: Cho biết giá (open, high, low, close) trung bình và volume trung bình của mỗi công ty
cols = list(['symbol','open','high','low','close','volume'])
stocks_companies[cols].groupby('symbol').mean()
```

	open	high	low	close	volume
symbol					
AAPL	174.890	175.76600	173.472	174.674	23733309.4
AMZN	1681.284	1694.13400	1651.078	1671.046	4755355.0
FB	167.654	169.56440	166.190	168.510	19603715.2
GOOG	1148.032	1158.63000	1134.118	1150.396	1321077.0
TSLA	288.476	291.26716	278.684	281.904	15710180.2

```
# Câu 6: Cho biết giá đóng cửa (close) trung bình, lớn nhất và nhỏ nhất ở mỗi công ty
stocks_companies.groupby('symbol').close.agg(['mean', 'min', 'max'])
```

	mean	min	max
symbol			
AAPL	174.674	172.50	175.85
AMZN	1671.046	1625.95	1696.17
FB	168.510	162.28	172.51
GOOG	1150.396	1140.99	1162.03
TSLA	281.904	276.24	294.79

```
# Câu 7: Tạo cột parsed_time trong stocks_companies bằng cách đổi thời gian sang định dạng datetime
stocks_companies['parsed_time'] = pd.to_datetime(stocks_companies['date'])
# Cho biết kiểu dữ liệu của cột parsed_time
stocks_companies['parsed_time'].dtype
```

```
# Hiển thị 5 dòng dữ liệu đầu của stocks_companies
stocks_companies.head()
```

	date	symbol	open	high	low	close	volume	name	employees	headqu
0	01-03-19	AMZN	1655.13	1674.26	1651.00	1671.73	4974877	AMZN	613300	
1	04-03-19	AMZN	1685.00	1709.43	1674.36	1696.17	6167358	AMZN	613300	
2	05-03-19	AMZN	1702.95	1707.80	1689.01	1692.43	3681522	AMZN	613300	

```
# Câu 8: Thêm cột result, nếu giá 'close' > 'open' thì cột result có giá trị 'up', ngược
stocks_companies.loc[stocks_companies['close'] > stocks_companies['open'], 'result'] = 'up'
stocks_companies.loc[stocks_companies['close'] < stocks_companies['open'], 'result'] = 'do'
stocks_companies.head()
```

	date	symbol	open	high	low	close	volume	name	employees	headqu
0	01-03-19	AMZN	1655.13	1674.26	1651.00	1671.73	4974877	AMZN	613300	
1	04-03-19	AMZN	1685.00	1709.43	1674.36	1696.17	6167358	AMZN	613300	
2	05-03-19	AMZN	1702.95	1707.80	1689.01	1692.43	3681522	AMZN	613300	

▼ Chapter 5 - Exercise 3: Phân tích dữ liệu Movies

Dữ liệu được lấy từ MovieLens website.

Download the Dataset

Theo link:

- **Data Source:** MovieLens web site (filename: ml-latest-small.zip)
- **Location:** <https://grouplens.org/datasets/movielens/latest/>

▼ Part 1: Đọc dữ liệu & Data Structures

▼ Trong ml-latest-small.zip bao gồm 3 file CSV sau:

- **ratings.csv** : *userId, movieId, rating, timestamp* : Chứa dữ liệu về các xếp hạng của các bộ phim, mỗi dòng biểu thị một xếp hạng của một phim bởi một người dùng.
- **tags.csv** : *userId, movieId, tag, timestamp* : chứa thông tin về các Tag mà người dùng gắn vào cho phim, mỗi dòng biểu thị cho 1 tag của một người dùng cho một phim
- **movies.csv** : *movieId, title, genres* : chứa thông tin về các bộ phim, mỗi dòng biểu thị thông tin của 1 bộ phim

Sử dụng `pd.read_csv()` để đọc dữ liệu

```
import pandas as pd
import numpy as np
```

```
# Câu 1: Đọc file movies.csv => đưa dữ liệu vào movies
movies = pd.read_csv('./movies_data/movies.csv', sep=',')
# Cho biết kiểu dữ liệu (type) của movies
print(type(movies))
# Cho biết kích thước (shape) của movies
print(movies.shape)
# Hiển thị 5 dòng dữ liệu đầu tiên (head) của movies
movies.head()
```

```
<class 'pandas.core.frame.DataFrame'>
(9125, 3)
```

```
# Hiển thị 5 dòng dữ liệu cuối (tail) của movies
movies.tail()
```

	movieId	title	genres
9120	162672	Mohenjo Daro (2016)	Adventure Drama Romance
9121	163056	Shin Godzilla (2016)	Action Adventure Fantasy Sci-Fi
9122	163949	The Beatles: Eight Days a Week - The Touring Y...	Documentary
9123	164977	The Gay Desperado (1936)	Comedy

```
# Cho biết kiểu dữ liệu (dtype) của các cột của movies
movies.dtypes
```

```
movieId    int64
title      object
genres      object
dtype: object
```

```
# Xem thông tin (info) của movies
movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9125 entries, 0 to 9124
Data columns (total 3 columns):
movieId    9125 non-null int64
title      9125 non-null object
genres     9124 non-null object
dtypes: int64(1), object(2)
memory usage: 214.0+ KB
```

```
# Câu 2: Đọc file tags.csv => đưa dữ liệu vào tags
tags = pd.read_csv('./movies_data/tags.csv', sep=',', parse_dates=['timestamp'])
# Cho biết kiểu dữ liệu (type) của tags
print(type(tags))
# Cho biết kích thước (shape) của tags
print(tags.shape)
# Hiển thị 5 dòng đầu tiên (head) của tags
tags.head()
```

```
<class 'pandas.core.frame.DataFrame'>
(1296, 4)
```

	userId	movieId	tag	timestamp
--	--------	---------	-----	-----------

```
# Hiển thị 5 dòng dữ liệu cuối (tail) của tags
tags.tail()
```

	userId	movieId	tag	timestamp
1291	660	135518	meaning of life	1436680885
1292	660	135518	philosophical	1436680885
1293	660	135518	sci-fi	1436680885
1294	663	260	action	1438398078
1295	663	260	Syfy	1438398050

```
# Cho biết kiểu dữ liệu (dtype) của các cột của tags
tags.dtypes
```

```
userId      int64
movieId     int64
tag         object
timestamp   object
dtype: object
```

```
# Xem thông tin (info) của tags
tags.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1296 entries, 0 to 1295
Data columns (total 4 columns):
userId      1296 non-null int64
movieId     1296 non-null int64
tag         1296 non-null object
timestamp   1295 non-null object
dtypes: int64(2), object(2)
memory usage: 40.6+ KB
```

```
# Câu 3: Đọc file ratings.csv => đưa dữ liệu vào ratings
ratings = pd.read_csv('./movies_data/ratings.csv', sep=',', parse_dates=['timestamp'])
# Cho biết kiểu dữ liệu (type) của ratings
print(type(ratings))
# Cho biết kích thước của ratings
print(ratings.shape)
# Hiển thị 5 dòng đầu tiên (head) của ratings
ratings.head()
```

```
<class 'pandas.core.frame.DataFrame'>
(100004, 4)

   userId  movieId  rating  timestamp
0        1         31     2.5  1260759144
1        1      1029     3.0  1260759179
```

Hiển thị 5 dòng dữ liệu cuối (tail) của ratings
ratings.tail()

```
   userId  movieId  rating  timestamp
99999      671     6268     2.5  1065579370
100000     671     6269     4.0  1065149201
100001     671     6365     4.0  1070940363
100002     671     6385     2.5  1070979663
100003     671     6565     3.5  1074784724
```

Cho biết kiểu dữ liệu (dtype) của các cột của ratings
ratings.dtypes

```
userId      int64
movieId     int64
rating      float64
timestamp   object
dtype: object
```

Xem thông tin (info) của ratings
ratings.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100004 entries, 0 to 100003
Data columns (total 4 columns):
userId      100004 non-null int64
movieId     100004 non-null int64
rating      100004 non-null float64
timestamp   100004 non-null object
dtypes: float64(1), int64(2), object(1)
memory usage: 3.1+ MB
```

▼ Part 2: Xử lý dữ liệu bị thiếu/ không hợp lệ

Câu 1: Cho biết trong movies có dữ liệu null hay không? Nếu có loại bỏ dòng có dữ liệu n
movies.isnull().any()

```
movieId     False
title       False
genres      True
dtype: bool
```

```
# Loại bỏ dòng có dữ liệu null
movies.dropna(subset=['genres'], axis=0, inplace=True)
```

```
# Câu 2: Cho biết trong ratings có dữ liệu null hay không? Nếu có loại bỏ dòng có dữ liệu null
ratings.isnull().any()
```

```
userId      False
movieId      False
rating       False
timestamp    False
dtype: bool
```

Trong ratings không có giá trị NULL.

```
# Câu 3: Cho biết trong tags có dữ liệu null hay không? Nếu có loại bỏ dòng có dữ liệu null
tags.isnull().any()
```

```
userId      False
movieId      False
tag          False
timestamp    True
dtype: bool
```

```
# Loại bỏ dòng có dữ liệu null
tags.dropna(subset=['timestamp'], axis=0, inplace=True)
```

```
# Câu 4: Kiểm tra xem có dữ liệu rating nào không hợp lệ hay không ('rating' > 5 hoặc 'rating' < 0)
# Nếu có, hãy thay bằng giá trị xuất hiện nhiều nhất
```

```
filter_rating = np.logical_or(ratings['rating'] > 5, ratings['rating'] < 0)
filter_rating.any()
```

```
True
```

```
# Thay bằng giá trị xuất hiện nhiều nhất
ratings.loc[filter_rating] = ratings['rating'].mode()[0]
```

▼ Part 3: Gộp DataFrame

```
# Câu 1: Tạo movies_tags bằng cách gộp dữ liệu của movies và tags theo cột chung là 'movieId'
movies_tags = movies.merge(tags, on='movieId', how='inner')
# Hiển thị 5 dòng đầu của movies_tags
movies_tags.head()
```


	movieId	title	genres	userId	tag	timestamp
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	501	Pixar	129295634

```
# Câu 2: Tạo movies_ratings bằng cách gộp dữ liệu của movies và ratings theo cột chung là movieId
movies_ratings = movies.merge(ratings, on='movieId', how='inner')
# Hiển thị 5 dòng đầu của movies_ratings
movies_ratings.head()
```

	movieId	title	genres	userId	rating	timestamp
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	7.0	3.0	85186670
1	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	9.0	4.0	93862917
2	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	13.0	5.0	133138005

▼ Part 4: Lọc dữ liệu theo yêu cầu

```
# Câu 1: Tạo dataframe tag_counts cho biết với mỗi tag là có bao nhiêu film chứa giá trị '
# gợi ý: dùng tags['tag'].value_counts()
tag_counts = tags['tag'].value_counts().to_frame()
# Hiển thị 10 dòng đầu của tag_counts
tag_counts.head(10)
```

	tag
getdvd	33
Ei muista	29
tivo	26
toplist07	26
toplist11	20
toplist12	20
tcm	20
toplist08	19
toplist15	19
toplist09	18

```
# Câu 2: Tạo is_highlyRated theo điều kiện: có 'rating' >= 4.0 của dataframe ratings
is_highlyRated = ratings['rating'] >= 4
```

```
# Hiển thị 5 dòng dữ liệu đầu của is_highly Rated
is_highly Rated.head()
```

```
0    False
1    False
2    False
3    False
4     True
Name: rating, dtype: bool
```

```
# Liệt kê các phim thỏa is_highly Rated
ratings[is_highly Rated][['movieId','rating']]
```

	movieId	rating
4	1172.0	4.0
12	1953.0	4.0
13	2105.0	4.0
20	10.0	4.0
21	17.0	5.0
...
99995	5989.0	4.0
99996	5991.0	4.5
99997	5995.0	4.0
100000	6269.0	4.0
100001	6365.0	4.0

51568 rows × 2 columns

```
# Câu 3: Tạo is_animation theo điều kiện trong cột genres của movies có chứa chuỗi 'Animat
is_animation = movies['genres'].str.contains('Animation')
```

```
# Hiển thị 5 dòng dữ liệu đầu của is_animation
is_animation.head()
```

```
0     True
1    False
2    False
3    False
4    False
Name: genres, dtype: bool
```

```
# Liệt kê các phim thỏa is_animation
movies[is_animation]
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
12	13	Balto (1995)	Adventure Animation Children
46	48	Pocahontas (1995)	Animation Children Drama Musical Romance
211	239	Goofy Movie, A (1995)	Animation Children Comedy Romance
216	244	Gumby: The Movie (1995)	Animation Children
...
9070	152081	Zootopia (2016)	Action Adventure Animation Children Comedy
9079	156025	Ice Age: The Great Egg-Scapade (2016)	Adventure Animation Children Comedy

```
# Câu 4: Tạo movie_genres từ cột 'genres' bằng cách tách cột 'genres' dựa vào ký tự '|'
movie_genres = movies['genres'].str.split('|', expand=True)
# Hiển thị 10 dòng cuối của movie_genres
movie_genres.tail(10)
```

	0	1	2	3	4	5	6	7	8	9
9115	Drama	Horror	Thriller	None	None	None	None	None	None	None
9116	Action	Adventure	Horror	Sci-Fi	None	None	None	None	None	None
9117	Drama	None	None	None	None	None	None	None	None	None
9118	Drama	None	None	None	None	None	None	None	None	None
9119	Romance	Thriller	None	None	None	None	None	None	None	None
9120	Adventure	Drama	Romance	None	None	None	None	None	None	None
9121	Action	Adventure	Fantasy	Sci-Fi	None	None	None	None	None	None
9122	Documentary	None	None	None	None	None	None	None	None	None
9123	Comedy	None	None	None	None	None	None	None	None	None
9124	Documentary	None	None	None	None	None	None	None	None	None

```
# Câu 5: Thêm cột mới cho movie_genres có tên là 'isComedy',
# giá trị là True nếu trong movies['genres'] có chứa chuỗi 'Comedy', ngược lại là False
movie_genres['isComedy'] = movies['genres'].str.contains('Comedy')
# Hiển thị 10 dòng đầu của movie_genres
movie_genres.head(10)
```

	0	1	2	3	4	5	6	7	8	9	isC
0	Adventure	Animation	Children	Comedy	Fantasy	None	None	None	None	None	
1	Adventure	Children	Fantasy	None	None	None	None	None	None	None	
2	Comedy	Romance	None	None	None	None	None	None	None	None	
3	Comedy	Drama	Romance	None	None	None	None	None	None	None	
4	Comedy	None	None	None	None	None	None	None	None	None	
5	Action	Crime	Thriller	None	None	None	None	None	None	None	

```
# Câu 6: Thêm cột mới cho movies có tên là 'year' với year được lấy ra từ cột 'title'
movies['year'] = movies['title'].str.extract('.*\(((.*)\).*', expand=True)
# Hiển thị 5 dòng dữ liệu đầu của movies
movies.head()
```

	movieId	title	genres	year
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	1995
1	2	Jumanji (1995)	Adventure Children Fantasy	1995
2	3	Grumpier Old Men (1995)	Comedy Romance	1995
3	4	Waiting to Exhale (1995)	Comedy Drama Romance	1995
4	5	Father of the Bride Part II (1995)	Comedy	1995

▼ Part 5: Thống kê dữ liệu

```
# Câu 1: Thực hiện thống kê chung dữ liệu ratings
ratings.describe()
```

	userId	movieId	rating
count	100004.000000	100004.000000	100004.000000
mean	347.011330	12548.660904	3.543608
std	195.163803	26369.200592	1.058064
min	1.000000	1.000000	0.500000
25%	182.000000	1028.000000	3.000000
50%	367.000000	2406.500000	4.000000
75%	520.000000	5418.000000	4.000000
max	671.000000	163949.000000	5.000000

```
# Câu 2: In giá trị trung bình, giá trị lớn nhất, giá trị nhỏ nhất,
# độ lệch chuẩn, giá trị có tần suất xuất hiện nhiều nhất của cột 'rating'
print('Giá trị trung bình của ratings: ', ratings['rating'].mean())
```

```
print('Giá trị nhỏ nhất của ratings: ', ratings['rating'].min())
print('Giá trị lớn nhất của ratings: ', ratings['rating'].max())
print('Độ lệch chuẩn của ratings: ', ratings['rating'].std())
print('Giá trị có tần suất xuất hiện nhiều nhất của ratings: ', ratings['rating'].mode()[0])
```

Giá trị trung bình của ratings: 3.543608255669773
Giá trị nhỏ nhất của ratings: 0.5
Giá trị lớn nhất của ratings: 5.0
Độ lệch chuẩn của ratings: 1.0580641091070389
Giá trị có tần suất xuất hiện nhiều nhất của ratings: 4.0

```
# Câu 3: Thống kê đếm số lượng phim theo 'rating' (Count of films). Xem kết quả.
count_of_films = movies_ratings[['movieId','rating']].groupby(['rating'], as_index = False)
count_of_films.rename(columns={"movieId":"Count of films"}, inplace = True)
# Xem kết quả
count_of_films
```

	rating	Count of films
0	0.5	1101
1	1.0	3326
2	1.5	1687
3	2.0	7268
4	2.5	4449
5	3.0	20058
6	3.5	10535
7	4.0	28743
8	4.5	7723
9	5.0	15094

```
# Câu 4: Đếm số lượng rating (Total ratings) theo phim, và lưu vào biến 'movie_count'
total_ratings = movies_ratings[['movieId','title','rating','genres']].groupby(['movieId','title','genres'], as_index = False)
total_ratings.rename(columns={"rating":"Total ratings"}, inplace=True)
# Hiển thị 5 dòng dữ liệu đầu của total_ratings
total_ratings.head()
```

	movieId	title	genres	Total ratings
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	247
1	2	Jumanji (1995)	Adventure Children Fantasy	107
2	3	Grumpier Old Men (1995)	Comedy Romance	59
3	4	Waiting to Exhale (1995)	Comedy Drama Romance	14
4	5	Father of the Bride Part II	Comedy	56

```
# Câu 5: Tính rating trung bình (Average ratings) theo mỗi phim, và lưu vào biến avg_rating
```

```

avg_ratings = movies_ratings[['movieId','title','rating','genres']].groupby(['movieId','ti
avg_ratings.rename(columns={"rating":"Average ratings"}, inplace=True)
# Hiển thị 5 dòng dữ liệu đầu của avg_ratings
avg_ratings.head()

```

	movieId	title	genres	Average ratings
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	3.872470
1	2	Jumanji (1995)	Adventure Children Fantasy	3.401869
2	3	Grumpier Old Men (1995)	Comedy Romance	3.161017
3	4	Waiting to Exhale (1995)	Comedy Drama Romance	2.500000
4	5	Father of the Bride Part II	Comedy	3.267957

```

# Câu 6: Hiển thị rating trung bình của các phim là 'Comedy',
# chỉ in ra 5 dòng dữ liệu đầu của dataframe kết quả
is_comedy = avg_ratings['genres'].str.contains('Comedy')
avg_ratings[is_comedy][-5:]

```

	movieId	title	genres	Average ratings
9041	159690	Teenage Mutant Ninja Turtles: Out of the Shado...	Action Adventure Comedy	2.0
9042	159755	Popstar: Never Stop Never Stopping (2016)	Comedy	1.0
9045	160080	Ghostbusters (2016)	Action Comedy Horror Sci-Fi	1.0

```

# Câu 7: Hiển thị rating trung bình của các phim là 'Comedy' và có 'rating' >= 4,
# chỉ in ra 5 dòng dữ liệu cuối của dataframe kết quả.
rating4 = avg_ratings['Average ratings'] >= 4.0
avg_ratings[rating4 & is_comedy][-5:]

```

	movieId	title	genres	Average ratings
9018	152081	Zootopia (2016)	Action Adventure Animation Children Comedy	4.0
9022	153584	The Last Days of Emma Blank (2009)	Comedy	5.0
9026	156025	Ice Age: The Great Egg-Scapade (2016)	Adventure Animation Children Comedy	5.0
9030	156026	Daniel Tosh: Completely	Comedy	4.5

```

# Câu 8: Tính trung bình rating theo year, và lưu vào biến yearly_average
joined = movies.merge(ratings, how = 'inner')
yearly_average = joined[['year','rating']].groupby('year', as_index = False).mean()
# Cho biết shape của yearly_average
print(yearly_average.shape)

```

```
# Hiển thị 5 dòng dữ liệu đầu của yearly_average
yearly_average.head()
```

```
(105, 2)
```

	year	rating
0	1902	4.333333
1	1915	3.000000
2	1916	3.500000
3	1917	4.250000
4	1918	4.250000

```
# Câu 9: Sắp xếp tăng dần yearly_average
yearly_average_asc = yearly_average.sort_values(by = 'year', ascending = True)
# Hiển thị 20 dòng dữ liệu đầu của yearly_average_asc
yearly_average_asc.head(20)
```

	year	rating
0	1902	4.333333
1	1915	3.000000
2	1916	3.500000
3	1917	4.250000
4	1918	4.250000
5	1919	3.000000
6	1920	3.700000
7	1921	4.416667
8	1922	3.803571
9	1923	4.166667
10	1924	4.250000
11	1925	3.925000
12	1926	4.025000
13	1927	3.921053
14	1928	4.261905
15	1929	3.361111
16	1930	3.935484
17	1931	4.040816
18	1932	3.500000
19	1933	3.993243

▼ Part 6: Parsing Timestamps

```
# Câu 1: Tạo cột parsed_time trong tags bằng cách đổi thời gian sang định dạng DateTime
tags['parsed_time'] = pd.to_datetime(tags['timestamp'], unit='s')
# Cho biết kiểu dữ liệu của cột parsed_time
tags['parsed_time'].dtype

dtype('<M8[ns]')

# Hiển thị 5 dòng dữ liệu đầu của tags
tags.head()
```

	userId	movieId	tag	timestamp	parsed_time
0	15	339	sandra 'boring' bullock	1138537770	2006-01-29 12:29:30
1	15	1955	dentist	1193435061	2007-10-26 21:44:21
2	15	7478	Cambodia	1170560997	2007-02-04 03:49:57
3	15	32892	Russian	1170626366	2007-02-04 21:59:26
4	15	34162	forgettable	1141391765	2006-03-03 13:16:05

```
# Câu 2: Tạo selected_rows chứa các dòng có tags['parsed_time'] > '2015-02-01'.
t = tags['parsed_time'] > '2015-02-01'
selected_rows = tags[t]
selected_rows.head()
```

	userId	movieId	tag	timestamp	parsed_time
8	15	100365	activist	1425876220	2015-03-09 04:43:40
9	15	100365	documentary	1425876220	2015-03-09 04:43:40
10	15	100365	uganda	1425876220	2015-03-09 04:43:40
15	73	107999	action	1430799184	2015-05-05 04:13:04
16	73	107999	anime	1430799184	2015-05-05 04:13:04

```
# Câu 3: Sắp xếp dữ liệu tags tăng dần theo cột parsed_time
# Hiển thị 10 dòng dữ liệu đầu của tags
tags.sort_values(by='parsed_time', ascending=True)[:10]
```


	userId	movieId		tag	timestamp	parsed_time
338	353	35836		dumb	1137217440	2006-01-14 05:44:00
0	15	339		sandra 'boring' bullock	1138537770	2006-01-29 12:29:30
232	294	36401		fairy tales	1138983064	2006-02-03 16:11:04
338	353	35836		dumb	1137217440	2006-02-03

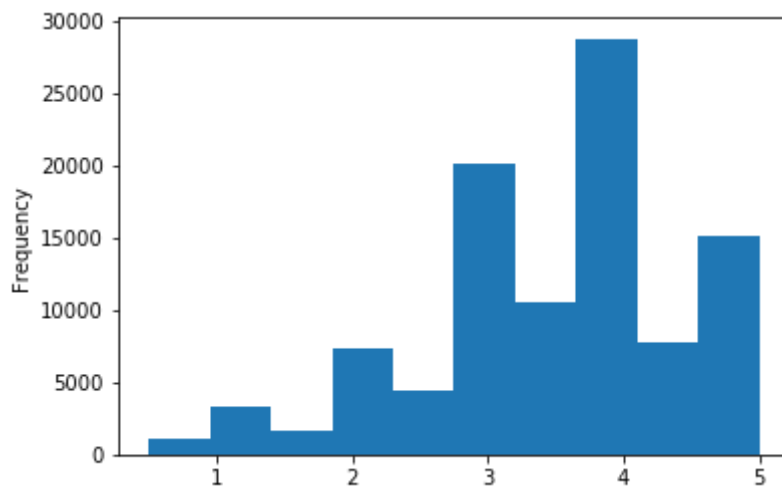
▼ Part 7: Trực quan hóa dữ liệu

VVVVZ

22:44:10

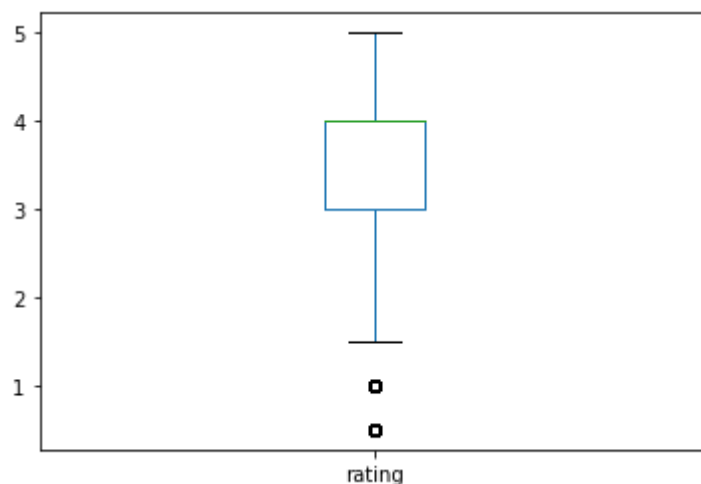
Câu 1: Vẽ biểu đồ histogram cột 'rating' của ratings
`ratings['rating'].plot.hist()`

<matplotlib.axes._subplots.AxesSubplot at 0xaa6b481dc8>



Câu 2: Vẽ biểu đồ bar plot cột 'rating' của ratings
`ratings['rating'].plot.box()`

<matplotlib.axes._subplots.AxesSubplot at 0xaa6b3903c8>



Câu 3: Dựa vào Câu 9 - Part 5, Vẽ biểu đồ plot() cho 20 dòng đầu của yearly_average sau
`yearly_average_asc.head(20).plot(x='year', y='rating')`



<matplotlib.axes._subplots.AxesSubplot at 0xaa6b3dde48>



Quan sát biểu đồ trên, nhận xét:

- Những năm nào có rating đặc biệt cao/thấp?

	country	beer_servi	spirit_servi	wine_servi	total_litres	continent
0	Afghanistan	0	0	0	0	AS
1	Albania	89	132	54	4.9	EU
2	Algeria	25	0	14	0.7	AF
3	Andorra	245	138	312	12.4	EU
4	Angola	217	57	45	5.9	AF
5	Antigua & Barbuda	102	128	45	4.9	
6	Argentina	193	25	221	8.3	SA
7	Armenia	21	179	11	3.8	EU
8	Australia	261	72	212	10.4	OC
9	Austria	279	75	191	9.7	EU
10	Azerbaijan	21	46	5	1.3	EU
11	Bahamas	122	176	51	6.3	
12	Bahrain	42	63	7	2	AS
13	Bangladesh	0	0	0	0	AS
14	Barbados	143	173	36	6.3	
15	Belarus	142	373	42	14.4	EU
16	Belgium	295	84	212	10.5	EU
17	Belize	263	114	8	6.8	
18	Benin	34	4	13	1.1	AF
19	Bhutan	23	0	0	0.4	AS
20	Bolivia	167	41	8	3.8	SA
21	Bosnia-Herzegovina	76	173	8	4.6	EU
22	Botswana	173	35	35	5.4	AF
23	Brazil	245	145	16	7.2	SA
24	Brunei	31	2	1	0.6	AS
25	Bulgaria	231	252	94	10.3	EU
26	Burkina Faso	25	7	7	4.3	AF
27	Burundi	88	0	0	6.3	AF
28	Cote d'Ivoire	37	1	7	4	AF
29	Cabo Verde	144	56	16	4	AF
30	Cambodia	57	65	1	2.2	AS
31	Cameroon	147	1	4	5.8	AF
32	Canada	240	122	100	8.2	
33	Central African Republic	17	2	1	1.8	AF
34	Chad	15	1	1	0.4	AF
35	Chile	130	124	172	7.6	SA
36	China	79	192	8	5	AS
37	Colombia	159	76	3	4.2	SA
38	Comoros	1	3	1	0.1	AF
39	Congo	76	1	9	1.7	AF
40	Cook Islands	0	254	74	5.9	OC
41	Costa Rica	149	87	11	4.4	
42	Croatia	230	87	254	10.2	EU
43	Cuba	93	137	5	4.2	
44	Cyprus	192	154	113	8.2	EU
45	Czech Republic	361	170	134	11.8	EU
46	North Korea	0	0	0	0	AS
47	DR Congo	32	3	1	2.3	AF
48	Denmark	224	81	278	10.4	EU

49 Djibouti	15	44	3	1.1 AF
50 Dominica	52	286	26	6.6
51 Dominican	193	147	9	6.2
52 Ecuador	162	74	3	4.2 SA
53 Egypt	6	4	1	0.2 AF
54 El Salvador	52	69	2	2.2
55 Equatorial	92	0	233	5.8 AF
56 Eritrea	18	0	0	0.5 AF
57 Estonia	224	194	59	9.5 EU
58 Ethiopia	20	3	0	0.7 AF
59 Fiji	77	35	1	2 OC
60 Finland	263	133	97	10 EU
61 France	127	151	370	11.8 EU
62 Gabon	347	98	59	8.9 AF
63 Gambia	8	0	1	2.4 AF
64 Georgia	52	100	149	5.4 EU
65 Germany	346	117	175	11.3 EU
66 Ghana	31	3	10	1.8 AF
67 Greece	133	112	218	8.3 EU
68 Grenada	199	438	28	11.9
69 Guatemala	53	69	2	2.2
70 Guinea	9	0	2	0.2 AF
71 Guinea-Bis	28	31	21	2.5 AF
72 Guyana	93	302	1	7.1 SA
73 Haiti	1	326	1	5.9
74 Honduras	69	98	2	3
75 Hungary	234	215	185	11.3 EU
76 Iceland	233	61	78	6.6 EU
77 India	9	114	0	2.2 AS
78 Indonesia	5	1	0	0.1 AS
79 Iran	0	0	0	0 AS
80 Iraq	9	3	0	0.2 AS
81 Ireland	313	118	165	11.4 EU
82 Israel	63	69	9	2.5 AS
83 Italy	85	42	237	6.5 EU
84 Jamaica	82	97	9	3.4
85 Japan	77	202	16	7 AS
86 Jordan	6	21	1	0.5 AS
87 Kazakhstar	124	246	12	6.8 AS
88 Kenya	58	22	2	1.8 AF
89 Kiribati	21	34	1	1 OC
90 Kuwait	0	0	0	0 AS
91 Kyrgyzstan	31	97	6	2.4 AS
92 Laos	62	0	123	6.2 AS
93 Latvia	281	216	62	10.5 EU
94 Lebanon	20	55	31	1.9 AS
95 Lesotho	82	29	0	2.8 AF
96 Liberia	19	152	2	3.1 AF
97 Libya	0	0	0	0 AF
98 Lithuania	343	244	56	12.9 EU

99 Luxembourg	236	133	271	11.4 EU
100 Madagascar	26	15	4	0.8 AF
101 Malawi	8	11	1	1.5 AF
102 Malaysia	13	4	0	0.3 AS
103 Maldives	0	0	0	0 AS
104 Mali	5	1	1	0.6 AF
105 Malta	149	100	120	6.6 EU
106 Marshall Is	0	0	0	0 OC
107 Mauritania	0	0	0	0 AF
108 Mauritius	98	31	18	2.6 AF
109 Mexico	238	68	5	5.5
110 Micronesia	62	50	18	2.3 OC
111 Monaco	0	0	0	0 EU
112 Mongolia	77	189	8	4.9 AS
113 Montenegro	31	114	128	4.9 EU
114 Morocco	12	6	10	0.5 AF
115 Mozambique	47	18	5	1.3 AF
116 Myanmar	5	1	0	0.1 AS
117 Namibia	376	3	1	6.8 AF
118 Nauru	49	0	8	1 OC
119 Nepal	5	6	0	0.2 AS
120 Netherlands	251	88	190	9.4 EU
121 New Zealand	203	79	175	9.3 OC
122 Nicaragua	78	118	1	3.5
123 Niger	3	2	1	0.1 AF
124 Nigeria	42	5	2	9.1 AF
125 Niue	188	200	7	7 OC
126 Norway	169	71	129	6.7 EU
127 Oman	22	16	1	0.7 AS
128 Pakistan	0	0	0	0 AS
129 Palau	306	63	23	6.9 OC
130 Panama	285	104	18	7.2
131 Papua New Guinea	44	39	1	1.5 OC
132 Paraguay	213	117	74	7.3 SA
133 Peru	163	160	21	6.1 SA
134 Philippines	71	186	1	4.6 AS
135 Poland	343	215	56	10.9 EU
136 Portugal	194	67	339	11 EU
137 Qatar	1	42	7	0.9 AS
138 South Korea	140	16	9	9.8 AS
139 Moldova	109	226	18	6.3 EU
140 Romania	297	122	167	10.4 EU
141 Russian Federation	247	326	73	11.5 AS
142 Rwanda	43	2	0	6.8 AF
143 St. Kitts & Nevis	194	205	32	7.7
144 St. Lucia	171	315	71	10.1
145 St. Vincent & the Grenadines	120	221	11	6.3
146 Samoa	105	18	24	2.6 OC
147 San Marino	0	0	0	0 EU
148 Sao Tome & Principe	56	38	140	4.2 AF

149 Saudi Arab	0	5	0	0.1 AS
150 Senegal	9	1	7	0.3 AF
151 Serbia	283	131	127	9.6 EU
152 Seychelles	157	25	51	4.1 AF
153 Sierra Leor	25	3	2	6.7 AF
154 Singapore	60	12	11	1.5 AS
155 Slovakia	196	293	116	11.4 EU
156 Slovenia	270	51	276	10.6 EU
157 Solomon Is	56	11	1	1.2 OC
158 Somalia	0	0	0	0 AF
159 South Afric	225	76	81	8.2 AF
160 Spain	284	157	112	10 EU
161 Sri Lanka	16	104	0	2.2 AS
162 Sudan	8	13	0	1.7 AF
163 Suriname	128	178	7	5.6 SA
164 Swaziland	90	2	2	4.7 AF
165 Sweden	152	60	186	7.2 EU
166 Switzerland	185	100	280	10.2 EU
167 Syria	5	35	16	1 AS
168 Tajikistan	2	15	0	0.3 AS
169 Thailand	99	258	1	6.4 AS
170 Macedonia	106	27	86	3.9 EU
171 Timor-Lest	1	1	4	0.1 AS
172 Togo	36	2	19	1.3 AF
173 Tonga	36	21	5	1.1 OC
174 Trinidad &	197	156	7	6.4
175 Tunisia	51	3	20	1.3 AF
176 Turkey	51	22	7	1.4 AS
177 Turkmenist	19	71	32	2.2 AS
178 Tuvalu	6	41	9	1 OC
179 Uganda	45	9	0	8.3 AF
180 Ukraine	206	237	45	8.9 EU
181 United Ara	16	135	5	2.8 AS
182 United Kin	219	126	195	10.4 EU
183 Tanzania	36	6	1	5.7 AF
184 USA	249	158	84	8.7
185 Uruguay	115	35	220	6.6 SA
186 Uzbekistan	25	101	8	2.4 AS
187 Vanuatu	21	18	11	0.9 OC
188 Venezuela	333	100	3	7.7 SA
189 Vietnam	111	2	1	2 AS
190 Yemen	6	0	0	0.1 AS
191 Zambia	32	19	4	2.5 AF
192 Zimbabwe	64	18	4	4.7 AF

movielfld	imdbld	tmdbld
1	114709	862
2	113497	8844
3	113228	15602
4	114885	31357
5	113041	11862
6	113277	949
7	114319	11860
8	112302	45325
9	114576	9091
10	113189	710
11	112346	9087
12	112896	12110
13	112453	21032
14	113987	10858
15	112760	1408
16	112641	524
17	114388	4584
18	113101	5
19	112281	9273
20	113845	11517
21	113161	8012
22	112722	1710
23	112401	9691
24	114168	12665
25	113627	451
26	114057	16420
27	114011	9263
28	114117	17015
29	112682	902
30	115012	37557
31	112792	9909
32	114746	63
34	112431	9598
35	112637	47018
36	112818	687
37	112286	139405
38	113442	33689
39	112697	9603
40	112749	34615
41	114279	31174
42	112819	11443
43	114272	35196
44	113855	9312
45	114681	577
46	113347	11861
47	114369	807
48	114148	10530
49	114916	8391
50	114814	629

52	113819	11448
53	110299	49133
54	112499	26441
55	113158	97406
57	113321	9089
58	110877	11010
59	112714	99040
60	113419	11359
61	116260	17182
62	113862	2054
63	116126	10607
64	118002	19760
65	115683	9536
66	116839	11525
68	113149	4482
69	113118	10634
70	116367	755
71	113010	11859
72	113537	28387
73	113828	48750
74	115644	20927
76	114367	9102
77	113973	124626
78	112744	27526
79	116731	9623
80	112445	46785
81	114660	400
82	112379	880
83	114039	146599
84	113612	188588
85	112365	8447
86	118158	10534
87	116151	17414
88	115697	13997
89	113972	2086
92	117002	9095
93	114825	12158
94	115639	9283
95	115759	9208
96	113403	40154
97	113247	406
98	111173	45549
99	113283	63076
100	115907	11062
101	115734	13685
102	117102	47475
103	118040	2045
104	116483	9614
105	112579	688
107	117110	10874

108	112646	89333
110	112573	197
111	75314	103
112	113326	33542
113	115645	43566
114	113774	71754
116	112373	51352
117	115033	16934
118	116606	10324
119	114536	78406
121	106473	32119
122	103859	11066
123	109424	11104
124	114808	37975
125	116324	2074
126	110647	27793
129	114131	110972
130	112364	32622
131	113107	73067
132	113451	11863
135	116130	9101
137	113756	5757
140	118055	9302
141	115685	11000
144	112585	16388
145	112442	9737
146	112342	30765
147	112461	10474
148	112427	22279
149	109093	30157
150	112384	568
151	114287	11780
152	112288	34996
153	112462	414
154	61395	649
155	112495	1873
156	112541	5894
157	109370	1775
158	112642	8839
159	112688	20649
160	112715	10329
161	112740	8963
162	109508	26564
163	112851	8068
164	112857	8512
165	112864	1572
166	112887	13552
167	113044	259209
168	113071	6520
169	113114	9073

170	113243	10428
171	113464	17447
172	113481	9886
173	113492	9482
174	113500	19326
175	113540	9344
176	113677	9071
177	113690	8973
178	107447	15730
179	113729	47608
180	113749	2293
181	113820	9070
183	110604	48787
184	110620	34574
185	113957	1642
186	113986	11472
187	114095	36196
188	114194	11980
189	114241	58372
190	114323	32646
191	114345	10533
193	114436	10802
194	114478	10149
195	114496	18402
196	114508	9348
198	114558	281
199	58450	5967
200	114666	79593
201	114663	47939
202	114702	36834
203	114682	9090
204	114781	3512
205	114798	52856
206	114805	77350
207	114887	9560
208	114898	9804
209	114928	31611
211	109340	49805
213	111579	50797
214	110882	19155
215	112471	76
216	112508	11017
217	112438	37141
218	112571	9382
219	112757	6715
220	112643	18256
222	112679	22625
223	109445	2292
224	112883	1909
225	109635	8984

userId	movieId	rating	timestamp
1	31	2.5	1260759144
1	1029	3	1260759179
1	1061	3	1260759182
1	1129	2	1260759185
1	1172	4	1260759205
1	1263	2	1260759151
1	1287	2	1260759187
1	1293	2	1260759148
1	1339	3.5	1260759125
1	1343	2	1260759131
1	1371	2.5	1260759135
1	1405	1	1260759203
1	1953	4	1260759191
1	2105	4	1260759139
1	2150	3	1260759194
1	2193	2	1260759198
1	2294	2	1260759108
1	2455	2.5	1260759113
1	2968	1	1260759200
1	3671	3	1260759117
2	10	4	835355493
2	17	5	835355681
2	39	5	835355604
2	47	4	835355552
2	50	4	835355586
2	52	3	835356031
2	62	3	835355749
2	110	4	835355532
2	144	3	835356016
2	150	5	835355395
2	153	4	835355441
2	161	3	835355493
2	165	3	835355441
2	168	3	835355710
2	185	3	835355511
2	186	3	835355664
2	208	3	835355511
2	222	5	835355840
2	223	1	835355749
2	225	3	835355552
2	235	3	835355664
2	248	3	835355896
2	253	4	835355511
2	261	4	835355681
2	265	5	835355697
2	266	5	835355586
2	272	3	835355767
2	273	4	835355779
2	292	3	835355492

2	296	4	835355395
2	300	3	835355532
2	314	4	835356044
2	317	2	835355551
2	319	1	835355918
2	339	3	835355492
2	349	4	835355441
2	350	6	835355697
2	356	3	835355628
2	357	3	835355749
2	364	3	835355604
2	367	3	835355619
2	370	2	835355932
2	371	3	835355968
2	372	3	835356094
2	377	3	835355710
2	382	3	835356165
2	405	2	835356246
2	410	3	835355532
2	454	4	835355604
2	457	3	835355511
2	468	4	835355790
2	474	2	835355828
2	480	4	835355643
2	485	3	835355918
2	497	3	835355880
2	500	4	835355731
2	508	4	835355860
2	509	4	835355719
2	515	4	835355817
2	527	4	835355731
2	537	4	835356199
2	539	3	835355767
2	550	3	835356109
2	551	5	835355767
2	552	3	835355860
2	585	5	835355817
2	586	3	835355790
2	587	3	835355779
2	588	3	835355441
2	589	5	835355697
2	590	5	835355395
2	592	5	835355395
2	593	3	835355511
2	616	3	835355932
2	661	4	835356141
2	720	4	835355978
3	60	3	1298861675
3	110	4	1298922049
3	247	3.5	1298861637

3	267	3	1298861761
3	296	4.5	1298862418
3	318	5	1298862121
3	355	2.5	1298861589
3	356	5	1298862167
3	377	2.5	1298923242
3	527	3	1298862528
3	588	3	1298922100
3	592	3	1298923247
3	593	3	1298921840
3	595	2	1298923260
3	736	3.5	1298932787
3	778	4	1298863157
3	866	3	1298861687
3	1197	5	1298932770
3	1210	3	1298921795
3	1235	4	1298861628
3	1271	3	1298861605
3	1378	4	1298861658
3	1580	3.5	1298922089
3	1721	4.5	1298923236
3	1884	4	1298863143
3	2028	4	1298921862
3	2318	4	1298861753
3	2513	3	1298861789
3	2694	3	1298862710
3	2702	3.5	1298861796
3	2716	3	1298924017
3	2762	3.5	1298922057
3	2841	4	1298861733
3	2858	4	1298921825
3	2959	5	1298862874
3	3243	3	1298861968
3	3510	4	1298861633
3	3949	5	1298863174
3	5349	3	1298923266
3	5669	3.5	1298862672
3	6377	3	1298922080
3	7153	2.5	1298921787
3	7361	3	1298922065
3	8622	3.5	1298861650
3	8636	3	1298932766
3	27369	3.5	1298862555
3	44191	3.5	1298932740
3	48783	4.5	1298862361
3	50068	4.5	1298862467
3	58559	3	1298922071
3	84236	4	1298922130
4	10	4	949810645
4	34	5	949919556

4	112	5	949810582
4	141	5	949919681
4	153	4	949811346
4	173	3	949811346
4	185	3	949920047
4	260	5	949779042
4	289	4	949778802
4	296	5	949895708
4	329	3	949810618
4	349	5	949810582
4	356	5	949919763
4	357	5	949919681
4	364	5	949949538
4	367	4	949895887
4	380	3	949810534
4	410	3	949919883
4	431	3	949895772
4	434	4	949810688
4	435	1	949920135
4	440	4	949919802
4	442	4	949920028
4	464	4	949811315
4	480	5	949810582
4	541	5	949779091
4	588	5	949949486
4	589	5	949919938
4	590	3	949810534
4	594	5	949949538
4	596	5	949949638
4	610	4	949982238
4	616	5	949949444
4	858	5	949779022
4	903	5	949919189
4	910	4	949919306
4	913	5	949919247
4	919	5	949949396
4	1011	4	949919454
4	1016	4	949919322
4	1022	5	949949638
4	1028	5	949949638
4	1030	5	949896377
4	1031	5	949896377
4	1032	5	949949538
4	1033	5	949949638
4	1036	5	949896244
4	1073	5	949919372
4	1079	5	949811523
4	1089	5	949895732
4	1097	5	949778771
4	1125	5	949919399

userId	movieId	tag	timestamp
15	339	sandra 'boring' bullock	1138537770
15	1955	dentist	1193435061
15	7478	Cambodia	1170560997
15	32892	Russian	1170626366
15	34162	forgettable	1141391765
15	35957	short	1141391873
15	37729	dull story	1141391806
15	45950	powerpoint	1169616291
15	100365	activist	1425876220
15	100365	documentary	1425876220
15	100365	uganda	1425876220
23	150	Ron Howard	1148672905
68	2174	music	1249808064
68	2174	weird	1249808102
68	8623	Steve Martin	1249808497
73	107999	action	1430799184
73	107999	anime	1430799184
73	107999	kung fu	1430799184
73	111624	drama	1431584497
73	111624	indie	1431584497
73	111624	love	1431584497
73	130682	b movie	1432523704
73	130682	comedy	1432523704
73	130682	horror	1432523704
77	1199	Trilogy of the Imagination	1163220043
77	2968	Gilliam	1163220138
77	2968	Trilogy of the Imagination	1163220039
77	4467	Trilogy of the Imagination	1163220065
77	4911	Gilliam	1163220167
77	5909	Takashi Miike	1163219591
77	47465	Gilliam	1163220186
84	296	intense	1429911417
84	296	r:violence	1429911417
84	296	tarantino	1429911417
91	4388	parody	1448813502
94	1131	emotional	1291781542
94	1131	tragedy	1291781538
94	64957	original plot	1291781246
94	74458	Predictable	1291780920
106	48711	CHRISTIAN	1215923364
132	4189	jesus	1367909949
132	4612	jesus	1367909949
132	6683	bollywood	1367909913
132	6986	jesus	1367909949
132	27255	No progress	1283581045
132	27255	Too slow	1283581045
132	27255	Views	1283581045
138	260	cult classic	1440379022
138	260	Science Fiction	1440379018

138	1258 cult film	1440380361
138	1258 jack nicholson	1440380355
138	1258 psychological	1440380357
138	1258 Stanley Kubrick	1440380352
138	1704 genius	1440380467
138	1704 intellectual	1440380463
138	1704 mathematics	1440380466
138	1704 psychology	1440380470
138	4226 Mindfuck	1440380125
138	4226 nonlinear	1440380113
138	4226 psychology	1440380115
138	4226 twist ending	1440380118
138	4995 genius	1440380440
138	4995 intelligent	1440380446
138	4995 math	1440380438
138	4995 mathematics	1440380436
138	4995 twist ending	1440380448
138	48780 Christopher Nolan	1440380053
138	48780 complicated	1440380062
138	48780 Hugh Jackman	1440380072
138	48780 nonlinear	1440380067
138	48780 psychological	1440380064
138	48780 twist ending	1440380047
138	79132 alternate reality	1440380233
138	79132 Christopher Nolan	1440380255
138	79132 intellectual	1440380251
138	79132 mindfuck	1440380252
138	79132 philosophy	1440380242
138	79132 sci-fi	1440380239
138	79132 twist ending	1440380245
138	109487 artificial intelligence	1440379984
138	109487 Christopher Nolan	1440380001
138	109487 good science	1440379977
138	109487 interesting idea	1440379991
138	109487 philosophical issues	1440379996
138	109487 physics	1440379925
138	109487 relativity	1440380500
138	109487 sci-fi	1440380008
138	109487 sentimental	1440379993
138	109487 space	1440379974
138	109487 time travel	1440379981
138	109487 time-travel	1440379968
149	121231 eerie	1436920551
149	121231 friendship	1436920551
149	121231 teenagers	1436920551
152	52319 World War II	1335900622
164	608 Quirky	1179031045
176	5445 Tom Cruise	1378384753
176	7458 Brad Pitt	1341055445
176	8972 Nicolas Cage	1341055378

176	69640 Johnny Depp	1340916254
176	104841 sexist	1384107184
179	260 nerdy	1436669973
179	260 Science Fiction	1436669968
187	2082 Emilio Estevez	1233517175
200	260 critically acclaimed	1437932756
200	260 Science Fiction	1437932763
200	80551 strong female presence	1438023607
212	1061 emotional	1253930128
212	1061 revenge	1253930131
212	1061 true story	1253930149
212	1732 cult film	1253929373
212	1732 dark comedy	1253929372
212	1732 satirical	1253929376
212	2359 British	1253930034
212	2641 childish plot	1253930390
212	2641 plot holes	1253930395
212	2641 superhero	1253930399
212	3257 Kevin Costner	1253930539
212	3257 snorefest	1253930543
212	3275 vigilantism	1218405594
212	3300 anti-hero	1253929917
212	3300 Cole Hauser	1253929922
212	3300 Vin Diesel	1253929910
212	3301 Amanda Peet	1253929798
212	3301 Matthew Perry	1253929802
212	3301 Rosanna Arquette	1253929813
212	3809 Bill Murray	1253931205
212	3809 quirky	1253931216
212	3882 cheerleading	1253932873
212	3882 Eliza Dushku	1253932876
212	3882 Overrated	1253932878
212	4069 standard romantic comedy	1253933882
212	5219 video game adaptation	1253930587
212	5507 play enough video games and you can become an NSA agent	1253930345
212	5507 ridiculous training sequence	1253930342
212	7373 Guillermo del Toro	1253929290
212	7373 steampunk	1253929296
212	7373 superhero	1253929293
212	8861 Oded Fehr	1253930624
212	8861 post-apocalyptic	1253930636
212	8861 Sienna Guillory	1253930633
212	8861 zombies	1253930644
212	8928 Beautiful Woman	1253926735
212	8928 campy	1253926723
212	8928 irreverent	1253926726
212	8928 quirky	1253926728
212	8928 Roman Polanski	1253926730
212	27904 good animation	1253931241
212	27904 interesting concept - bad execution	1253931234

212	27904 surrealism	1253931244
212	34405 aliens	1253928597
212	34405 black comedy	1253928590
212	34405 Firefly	1253928585
212	37733 disappointing	1253929514
212	37733 overrated	1253929532
212	37733 Viggo Mortensen	1253929518
212	48394 dark	1218405627
212	48394 fairytale	1218405627
212	60684 alternate reality	1253926526
212	60684 diluted version of comic	1253926511
212	60684 dystopia	1253926517
212	60684 stylized	1253926519
212	63992 high school	1253932469
212	63992 Kristen Stewart	1253932484
212	63992 Robert Pattinson	1253932481
212	63992 Teen movie	1253932459
212	63992 Vampire Human Love	1253932463
212	64957 adapted from:book	1253929184
212	64957 Aging Disorder	1253929189
212	64957 Brad Pitt	1253929141
212	64957 cinematography	1253929177
212	64957 drama	1253929173
212	64957 original plot	1253929180
212	64957 slow parts	1253929162
212	64957 touching	1253929169
212	66097 alternate universe	1253926354
212	66097 author:Neil Gaiman	1253926328
212	66097 claymation	1253926330
212	66097 Dakota Fanning	1253926349
212	66097 dark	
212	66097 fairy tale	1253926364
212	66097 Neil Gaiman	1253926360
212	66934 Captain Hammer	1253926165
212	66934 joss whedon	1253926158
212	66934 mad scientist	1253926186
212	66934 musical	1253926178
212	66934 Nathan Fillion	1253926182
212	66934 Neil Patrick Harris	1253926160
212	66934 parody	1253926168
212	68157 Brad Pitt	1253926443
212	68157 ending	1253926451
212	68157 gratuitous violence	1253926429
212	68157 Quentin Tarantino	1253926404
212	68157 satire	1253926435
212	68157 unusual plot structure	1253926417
212	68157 violence	1253926426
212	68157 World War II	1253926408
212	68319 action	1253928934
212	68319 bad plot	1253928950

movieid	title	genres
1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
2	Jumanji (1995)	Adventure Children Fantasy
3	Grumpier Old Men (1995)	Comedy Romance
4	Waiting to Exhale (1995)	Comedy Drama Romance
5	Father of the Bride Part 2 (1995)	Comedy
6	Heat (1995)	Action Crime Thriller
7	Sabrina (1995)	Comedy Romance
8	Tom and Huck (1995)	Adventure Children
9	Sudden Death (1995)	Action
10	GoldenEye (1995)	Action Adventure Thriller
11	American President, The (1995)	Comedy Drama Romance
12	Dracula: Dead and Loving It (1995)	Comedy Horror
13	Balto (1995)	Adventure Animation Children
14	Nixon (1995)	Drama
15	Cutthroat Island (1995)	Action Adventure Romance
16	Casino (1995)	Crime Drama
17	Sense and Sensibility (1995)	Drama Romance
18	Four Rooms (1995)	Comedy
19	Ace Ventura: When Nature Calls (1995)	Comedy
20	Money Train (1995)	Action Comedy Crime Drama Thriller
21	Get Shorty (1995)	Comedy Crime Thriller
22	Copcat (1995)	Crime Drama Horror Mystery Thriller
23	Assassins (1995)	Action Crime Thriller
24	Powder (1995)	Drama Sci-Fi
25	Leaving Las Vegas (1995)	Drama Romance
26	Othello (1995)	Drama
27	Now and Then (1995)	Children Drama
28	Persuasion (1995)	Drama Romance
29	City of Lost Children, The (1995)	Adventure Drama Fantasy Mystery Sci-Fi
30	Shanghai Triad (Yao a Hei) (1995)	Crime Drama
31	Dangerous Minds (1995)	Drama
32	Twelve Monkeys (a.k.a. The Third Man of Shanghai) (1995)	Mystery Sci-Fi Thriller
34	Babe (1995)	Children Drama
35	Carrington (1995)	Drama Romance
36	Dead Man Walking (1995)	Crime Drama
37	Across the Sea of Time (1995)	Documentary IMAX
38	It Takes Two (1995)	Children Comedy
39	Clueless (1995)	Comedy Romance
40	Cry, the Beloved Country (1995)	Drama
41	Richard III (1995)	Drama War
42	Dead Presidents (1995)	Action Crime Drama
43	Restoration (1995)	Drama
44	Mortal Kombat (1995)	Action Adventure Fantasy
45	To Die For (1995)	Comedy Drama Thriller
46	How to Make an American Quilt (1995)	Drama Romance
47	Seven (a.k.a. Se7en) (1995)	Mystery Thriller
48	Pocahontas (1995)	Animation Children Drama Musical Romance
49	When Night Is Falling (1995)	Drama Romance
50	Usual Suspects, The (1995)	Crime Mystery Thriller

52 Mighty Aphrodite (1994) Comedy|Drama|Romance
 53 Lamerica (1994) Adventure|Drama
 54 Big Green, The (1995) Children|Comedy
 55 Georgia (1995) Drama
 57 Home for the Holiday Drama
 58 Postman, The (Postino) Comedy|Drama|Romance
 59 Confessional, The (Confessions) Drama|Mystery
 60 Indian in the Cupboard Adventure|Children|Fantasy
 61 Eye for an Eye (1996) Drama|Thriller
 62 Mr. Holland's Opus (1995) Drama
 63 Don't Be a Menace to Comedy|Crime
 64 Two if by Sea (1996) Comedy|Romance
 65 Bio-Dome (1996) Comedy
 66 Lawnmower Man 2: Beyond the Madness Action|Sci-Fi|Thriller
 68 French Twist (Gazon rouge) Comedy|Romance
 69 Friday (1995) Comedy
 70 From Dusk Till Dawn (1999) Action|Comedy|Horror|Thriller
 71 Fair Game (1995) Action
 72 Kicking and Screaming Comedy|Drama
 73 Misérables, Les (1994) Drama|War
 74 Bed of Roses (1996) Drama|Romance
 76 Screamers (1995) Action|Sci-Fi|Thriller
 77 Nico Icon (1995) Documentary
 78 Crossing Guard, The (1994) Action|Crime|Drama|Thriller
 79 Juror, The (1996) Drama|Thriller
 80 White Balloon, The (The White Balloon) Children|Drama
 81 Things to Do in Denver When You're in Love Crime|Drama|Romance
 82 Antonia's Line (Antonia) Comedy|Drama
 83 Once Upon a Time... in the West Drama|Romance
 84 Last Summer in the Hamptons Comedy|Drama
 85 Angels and Insects (1995) Drama|Romance
 86 White Squall (1996) Action|Adventure|Drama
 87 Dunston Checks In (1995) Children|Comedy
 88 Black Sheep (1996) Comedy
 89 Nick of Time (1995) Action|Thriller
 92 Mary Reilly (1996) Drama|Horror|Thriller
 93 Vampire in Brooklyn (1997) Comedy|Horror|Romance
 94 Beautiful Girls (1996) Comedy|Drama|Romance
 95 Broken Arrow (1996) Action|Adventure|Thriller
 96 In the Bleak Midwinter Comedy|Drama
 97 Hate (Haine, La) (1995) Crime|Drama
 98 Shopping (1994) Action|Thriller
 99 Heidi Fleiss: Hollywood Documentary
 100 City Hall (1996) Drama|Thriller
 101 Bottle Rocket (1996) Adventure|Comedy|Crime|Romance
 102 Mr. Wrong (1996) Comedy
 103 Unforgettable (1996) Mystery|Sci-Fi|Thriller
 104 Happy Gilmore (1996) Comedy
 105 Bridges of Madison County Drama|Romance
 107 Muppet Treasure Island Adventure|Children|Comedy|Musical

108 Catwalk (1996) Documentary
 110 Braveheart (1995) Action|Drama|War
 111 Taxi Driver (1976) Crime|Drama|Thriller
 112 Rumble in the Bronx (Action|Adventure|Comedy|Crime
 113 Before and After (199 Drama|Mystery
 114 Margaret's Museum (Drama
 116 Anne Frank Remembered Documentary
 117 Young Poisoner's Hand Crime|Drama
 118 If Lucy Fell (1996) Comedy|Romance
 119 Steal Big, Steal Little (Comedy
 121 Boys of St. Vincent, The Drama
 122 Boomerang (1992) Comedy|Romance
 123 Chungking Express (C Drama|Mystery|Romance
 124 Star Maker, The (Uom Drama
 125 Flirting With Disaster (Comedy
 126 NeverEnding Story III, Adventure|Children|Fantasy
 129 Pie in the Sky (1996) Comedy|Romance
 130 Angela (1995) Drama
 131 Frankie Starlight (199 Drama|Romance
 132 Jade (1995) Thriller
 135 Down Periscope (199 Comedy
 137 Man of the Year (199 Documentary
 140 Up Close and Personal Drama|Romance
 141 Birdcage, The (1996) Comedy
 144 Brothers McMullen, T Comedy
 145 Bad Boys (1995) Action|Comedy|Crime|Drama|Thriller
 146 Amazing Panda Adventure Adventure|Children
 147 Basketball Diaries, The Drama
 148 Awfully Big Adventure Drama
 149 Amateur (1994) Crime|Drama|Thriller
 150 Apollo 13 (1995) Adventure|Drama|IMAX
 151 Rob Roy (1995) Action|Drama|Romance|War
 152 Addiction, The (1995) Drama|Horror
 153 Batman Forever (199 Action|Adventure|Comedy|Crime
 154 Beauty of the Day (Be Drama
 155 Beyond Rangoon (199 Adventure|Drama|War
 156 Blue in the Face (1995 Comedy|Drama
 157 Canadian Bacon (1995 Comedy|War
 158 Casper (1995) Adventure|Children
 159 Clockers (1995) Crime|Drama|Mystery
 160 Congo (1995) Action|Adventure|Mystery|Sci-Fi
 161 Crimson Tide (1995) Drama|Thriller|War
 162 Crumb (1994) Documentary
 163 Desperado (1995) Action|Romance|Western
 164 Devil in a Blue Dress (Crime|Film-Noir|Mystery|Thriller
 165 Die Hard: With a Vengeance Action|Crime|Thriller
 166 Doom Generation, The Comedy|Crime|Drama
 167 Feast of July (1995) Drama
 168 First Knight (1995) Action|Drama|Romance
 169 Free Willy 2: The Adventure Adventure|Children|Drama

170 Hackers (1995) Action|Adventure|Crime|Thriller
 171 Jeffrey (1995) Comedy|Drama
 172 Johnny Mnemonic (1995) Action|Sci-Fi|Thriller
 173 Judge Dredd (1995) Action|Crime|Sci-Fi
 174 Jury Duty (1995) Comedy
 175 Kids (1995) Drama
 176 Living in Oblivion (1995) Comedy
 177 Lord of Illusions (1995) Horror
 178 Love & Human Remains (1995) Comedy|Drama
 179 Mad Love (1995) Drama|Romance
 180 Mallrats (1995) Comedy|Romance
 181 Mighty Morphin Power Rangers (1995) Action|Children
 183 Mute Witness (1994) Comedy|Horror|Thriller
 184 Nadja (1994) Drama
 185 Net, The (1995) Action|Crime|Thriller
 186 Nine Months (1995) Comedy|Romance
 187 Party Girl (1995) Comedy
 188 Prophecy, The (1995) Fantasy|Horror|Mystery
 189 Reckless (1995) Comedy|Fantasy
 190 Safe (1995) Thriller
 191 Scarlet Letter, The (1995) Drama|Romance
 193 Showgirls (1995) Drama
 194 Smoke (1995) Comedy|Drama
 195 Something to Talk About (1995) Comedy|Drama|Romance
 196 Species (1995) Horror|Sci-Fi
 198 Strange Days (1995) Action|Crime|Drama|Mystery|Sci-Fi|Thriller
 199 Umbrellas of Cherbourg (1964) Drama|Musical|Romance
 200 Tie That Binds, The (1995) Thriller
 201 Three Wishes (1995) Drama|Fantasy
 202 Total Eclipse (1995) Drama|Romance
 203 To Wong Foo, Thanks for the Memories (1995) Comedy
 204 Under Siege 2: Dark Territory (1995) Action
 205 Unstrung Heroes (1995) Comedy|Drama
 206 Unzipped (1995) Documentary
 207 Walk in the Clouds, A (1995) Drama|Romance
 208 Waterworld (1995) Action|Adventure|Sci-Fi
 209 White Man's Burden (1995) Drama
 211 Browning Version, The (1995) Drama
 213 Burnt by the Sun (1995) Drama
 214 Before the Rain (1995) Drama|War
 215 Before Sunrise (1995) Drama|Romance
 216 Billy Madison (1995) Comedy
 217 Babysitter, The (1995) Drama|Thriller
 218 Boys on the Side (1995) Comedy|Drama
 219 Cure, The (1995) Drama
 220 Castle Freak (1995) Horror
 222 Circle of Friends (1995) Drama|Romance
 223 Clerks (1994) Comedy
 224 Don Juan DeMarco (1995) Comedy|Drama|Romance
 225 Disclosure (1994) Drama|Thriller

date	symbol	open	high	low	close	volume
01-03-19	AMZN	1655.13	1674.26	1651	1671.73	4974877
04-03-19	AMZN	1685	1709.43	1674.36	1696.17	6167358
05-03-19	AMZN	1702.95	1707.8	1689.01	1692.43	3681522
06-03-19	AMZN	1695.97			1668.95	3996001
07-03-19	AMZN	1667.37	1669.75	1620.51	1625.95	4957017
01-03-19	AAPL	174.28	175.15	172.89	174.97	25886167
04-03-19	AAPL	175.69	177.75	173.97	175.85	27436203
05-03-19	AAPL	175.94	176	174.54	175.53	19737419
06-03-19	AAPL	174.67	175.49	173.94	174.52	20810384
07-03-19	AAPL	173.87	174.44	172.02	172.5	24796374
01-03-19	GOOG	1124.9	1142.97	1124.75	1140.99	1450316
04-03-19	GOOG	1146.99	1158.28	1130.69	1147.8	1446047
05-03-19	GOOG	1150.06			1162.03	1443174
06-03-19	GOOG	1162.49	1167.57	1155.49	1157.86	1099289
07-03-19	GOOG	1155.72	1156.76	1134.91	1143.3	1166559

date	symbol	open	high	low	close	volume
01-03-19	FB	162.6	163.132	161.69	162.28	11097770
04-03-19	FB	163.9	167.5	163.83	167.37	18894689
05-03-19	FB	167.37	171.88	166.55	171.26	28187890
06-03-19	FB	172.9	173.57	171.27	172.51	21531723
07-03-19	FB	171.5	171.74	167.61	169.13	18306504
01-03-19	TSLA	306.94	307.13	291.9	294.79	22911375
04-03-19	TSLA	298.12	299	282.78	285.36	17096818
05-03-19	TSLA	282	284	270.1	276.54	18764740
06-03-19	TSLA	276.48	281.5058	274.39	276.24	10335485
07-03-19	TSLA	278.84	284.7	274.25	276.59	9442483

name	employees	headquarters	headquarters_state
AMZN	613300	Seattle	WA
GOOG	98771	Mountain View	CA
AAPL	132000	Cupertino	CA
FB	48268	Menlo Park	CA
TSLA	48016	Palo Alto	CA