Chapter-22 Pandas

Contents

Chapter-22 Pandas

22.1 Pandas

Pandas သည် ဒေတာများကို လေ့လာရန်၊ analysis လုပ်ရန် အထူးပြုလုပ်ထားသည့် Python library တစ်ခု ဖြစ်သည်။ ဒေတာများကို **DataFrame** အဖြစ် ပြုလုပ်ပြီး လေ့လာသည်။ ဒေတာများ တည်ဆောက် ထားပုံကို data structure ဟုခေါ်သည်။ Pandas DataFrame သည် excel spreadsheet မှ ဇယားများနှင့် တူညီသောကြောင့် excel ကို အသုံးပြုတတ်သူအားလုံး အလွယ်တကူ လျှင်မြန်စွာလေ့လာ သင်ယူနိုင်သည်။ Excel spread sheet ၏ ဒေတာတည်ဆောက်ထားပုံ(data structure)သည် ဇယားပုံစံ(table format) ဖြစ်သည်။ ထို့အပြင် SQL မှာကဲ့သို့ ဇယားများကို query လုပ်ခြင်း၊ ဆက်ခြင်း (SQL-like queries and joins of tables) စသည်တို့ ပြုလုပ်နိုင်သည်။

Pandas နှင့် NumPy မတူညီသည့်အချက်မှာ NumPy တွင် ထည့်သွင်းရသည့် အရာအားလုံး(NumPy တွင် ရှိသည့် အရာအားလုံး)သည် ဒေတာအမျိုးအစား တူညီကြသည့် array(array be of the same type)များ ဖြစ်ကြသည်။ Pandas တွင် ဒေတာများကို ဇယားပုံစံ(table format)ဖြင့် သိမ်းဆည်းထားသောကြောင့် ကော်လံ တစ်ခုစီတွင် မတူညီသည် data format များဖြစ်သည့် integer များ ၊ date များ ၊ floating-point number များ ၊ string များ စသည်တို့ ဖြစ်နိုင်သည်။ တစ်နည်းအားဖြင့် NumPy သည် data format တူညီသည့် array များကိုသာ ကိုင်တွယ် စီမံပေးသည်။ Pandas သည် data format မတူညီသည့် ဇယားပုံစံ(table format) ဒေတာများကို ကိုင်တွယ် စီမံပေးနိုင်သည်။

Pandas သည် အမျိုးမျိုးသော ဒေတာဖိုင်များနှင့် data- base များ အားလုံးကို ဖတ်ယူနိုင်စွမ်း ရှိသည်။ ဥပမာ- SQL database ၊ excel ဖိုင်၊ comma-separated values (CSV)ဖိုင် တို့ကို အလွယ်တကူ ဖတ်ယူ နိုင်သည်။ Pandas သည် machine learning ဘာသာရပ်တွင် ဒေတာများ သန့်စင်ခြင်း(cleaning)၊ ရှာဖွေ ဖော်ထုတ်ခြင်း(data exploration)နှင့် transformation လုပ်ခြင်းတို့အတွက် အလွန်အသုံးဝင်သည်။ Data manipulation function များစွာ ပါဝင်သည်။ Indexing လုပ်ရာတွင် အားသာချက်များစွာ ရှိသည်။

- Data များ၏ statistics အချက်အလက်များကို အလွယ်တကူ ရှာဖွေနိုင်သည်။
- Data cleaning လုပ်ရန်အတွက် built in pandas function များစွာ ရှိသည်။
- ဒေတာများ subsetting လုပ်ရန် ၊ filtering လုပ်ရန် ၊ insertion လုပ်ရန် ၊ deletion လုပ်ရန် နှင့် aggregation လုပ်ရန် function များစွာ ရှိသည်။
- Data set များ ပေါင်းခြင်း(merging)၊ ဆက်ခြင်း ပြုလုပ်ရန် function များစွာ ရှိသည်။
- Time-series data များအတွက် timestamp များ ၊ အချိန်နှင့် သက်ဆိုင်သည့် function များစွာ ရှိသည်။

22.1.1 Install Pandas

ပထမဦးစွာ Pandas ကို အသုံးပြုရန်အတွက် PyCharm မှ Terminal တွင် pip install လုပ်ရပါသည်။

22.1.2 Import Libraries

Pandas ကို အသုံးပြုရန်တွက် Pandas library ကို import လုပ်ရပါသည်။

```
import pandas as pd
```

22.1.3 Introduction to Pandas Data Structures

Pandas ၏ အဓိက ဒေတာ တည်ဆောက်ပုံနှစ်မျိုး(two main data structures)မှာ **Series** နှင့် **DataFrames** တို့ ဖြစ်သည်။

22.2 Pandas Series

Pandas series ဆိုသည်မှာ dimension တစ်ခုသာပါသည့် label များ တတ်ထားသည့် array (one-dimensional labeled array)တစ်ခု ဖြစ်သည်။ `pd.Series()` ဖြင့် pandas series တစ်ခု တည်ဆောက်သည်။

```
ser = pd.Series([100, 'foo', 300, 'bar', 500],
                ['tom', 'bob', 'nancy', 'dan', 'eric'])
```

တည်ဆောက်ပြီးသည့် pandas series ကို ပြန်ကြည့်သည်။

```
ser
```

```
tom      100
bob      foo
nancy    300
dan      bar
eric     500
dtype: object
```

`.index` ကို အသုံးပြု၍ Pandas series မှ index များကို ပြန်ကြည့်သည်။

```
ser.index
```

```
Index(['tom', 'bob', 'nancy', 'dan', 'eric'], dtype='object')
```

`.axes` ဖြင့် Pandas series မှ axes ကို ကြည့်သည်။ Pandas series တွင် axis တစ်ခုသာ ရှိသောကြောင့် axis နှင့် index သည် အတူတူပင် ဖြစ်သည်။

```
ser.axes
```

```
[Index(['tom', 'bob', 'nancy', 'dan', 'eric'], dtype='object')]
```

Pandas serie မှ 'nancy' နှင့် 'bob' တို့၏ သက်ဆိုင်ရာ တန်ဖိုးများကို `ser.loc[]` ဖြင့် access လုပ်နိုင်သည်။ Pandas series မှ 'nancy' ၊ 'bob' စသည် နာမည်များကို ထည့်ပေး၍ access လုပ်နိုင်သည်။

```
ser.loc[['nancy','bob']]
```

```
nancy    300
bob      foo
dtype: object
```

Pandas series ၏ index မှ နံပါတ်(row number) ပေး၍ access လုပ်နိုင်သည်။

```
ser[[4, 3, 1]]       # row နံပါတ် 4, 3 နှင့် 1 ကို access လုပ်သည်။
```

```
eric    500
dan     bar
bob     foo
dtype: object
```

Pandas series ၏ index မှ နံပါတ်(row number) ပေး၍ `.iloc[ ]` ဖြင့်လည်း access လုပ်နိုင်သည်။

```
ser.iloc[2]
```

```
300
```

Pandas series ၏ index မှ 'nancy' ၊ 'bob' စသည် နာမည်များကို ပေး၍ series ထဲတွင် ရှိ မရှိ စစ်နိုင်သည်။ ရှိနေလျှင် True သို့မဟုတ် မရှိလျှင် False ဖြင့် ဖော်ပြလိမ့်မည်။

```
'bob' in ser
```

```
True
```

Pandas series ၏ နာမည်ဖြင့် ထို series ထဲတွင် ရှိနေသည့် item များ အားလုံးကို ကြည့်နိုင်သည်။

```
ser
```

```
tom      100
bob      foo
nancy    300
dan      bar
eric     500
dtype: object
```

Pandas series တစ်ခုလုံးကို (၂)နှင့် မြှောက်သည်။ ထိုအခါ series ထဲမှ item သည် တန်ဖိုး(value) ဖြစ်နေလျှင် (၂)ဆတိုးပြီး, string ဖြစ်နေလျှင် (၂)ခုဖြစ်အောင် လုပ်ပေးလိမ့်မည်။

```
ser * 2
```

```
tom         200
bob      foofoo
nancy       600
dan      barbar
eric       1000
dtype: object
```

22.2.1 Pandas Series အတွင်းမှ Item များကို နှစ်ထပ်ကိန်းတင်ခြင်း

```
ser[['nancy', 'eric']] ** 2
```

```
nancy     90000
eric     250000
dtype: object
```

22.2.2 Create DataFrame From Dictionary of Python Series

Pandas DataFrame သည် 2-dimensional labeled data structure တစ်ခု ဖြစ်သည်။ Pandas series နှစ်ခုဖြင့် DataFrame တစ်ခု တည်ဆောက်နိုင်သည်။ DataFrame ဖြစ်သောကြောင့် row index နှင့် column index ဟူ၍ index (၂)ခု ရှိသည်။

```
d = {'one':pd.Series([100., 200., 300.],
      index=['apple', 'ball','clock']),
     'two' : pd.Series([111., 222., 333., 4444.],
     index=['apple', 'ball', 'cerill', 'dancy'])}
```

d သည် DataFrame တစ်ခု မဖြစ်သေးပါ။ 'dict' အဖြစ်သာ ရှိသေးသည်။

```
print(type(d))
```

```
<class 'dict'>
```

```
d
```

```
{'one': apple    100.0
 ball     200.0
 clock    300.0
 dtype: float64,
 'two': apple      111.0
 ball       222.0
 cerill     333.0
 dancy     4444.0
 dtype: float64}
```

`pd.DataFrame(d)`ဖြင့် DataFrame တစ်ခု ဖြစ်အောင် ပြုလုပ်သည်။ `df` သည် Pandas က ပြုလုပ်လိုက်သည့် DataFrame တစ်ခု ဖြစ်သည်။ DataFrame ဖြစ်သည့်အခါ index များကို ဘုံဖြစ်အောင်(common) ပြုလုပ်ပြီး တန်ဖိုး မရှိသည့် နေရာတွင် NaN ဖြင့် အစားထိုးသည်။

```
df = pd.DataFrame(d)
print(df)                    # df ဆိုသည် DataFrame အတွင်းရှိ item များကိုကြည့်ရန်
```

```
          one     two
apple   100.0   111.0
ball    200.0   222.0
```

```
cerill    NaN   333.0
clock   300.0     NaN
dancy     NaN  4444.0
```

DataFrame တွင် row နှင့် column ဟူ၍ axis နှစ်ခုသည် ရှိသည်။ Index သည် row index ဖြစ်သည်။ Column index ကို column ဟူ၍သာ ရေးပေးရန် လိုသည်။ တစ်နည်းအားဖြင့် DataFrame တွင် index ဟု ပြောလျှင် row ကို ဆိုလိုသည်။

```
df.index
```

```
Index(['apple', 'ball', 'cerill', 'clock', 'dancy'], dtype='object')
```

```
df.columns
```

```
Index(['one', 'two'], dtype='object')
```

အထက်တွင် တည်ဆောက်ခဲ့သည် d ဆိုသည့် dict မှ မိမိအလိုရှိသည့် item များကို သာယူ၍ DataFrame တစ်ခုကို အောက်ပါအတိုင်း တည်ဆောက်နိုင်သည်။

`pd.DataFrame(d, index=['dancy', 'ball', 'apple'])` ၏ အဓိပ္ပာယ်မှာ `d` ဆိုသည့် dict ထဲမှ 'dancy', 'ball' နှင့် 'apple' စသည့်တို့ သက်ဆိုင်သည့် တန်ဖိုးများကို ယူ၍ row (၃)ခု အဖြစ်ထားပြီး DataFrame တစ်ခု တည်ဆောက်သည်။

```
pd.DataFrame(d, index=['dancy', 'ball', 'apple'])
```

	one	two
dancy	NaN	4444.0
ball	200.0	222.0
apple	100.0	111.0

အထက်တွင် တည်ဆောက်ခဲ့သည် d ဆိုသည့် dict မှ မိမိအလိုရှိသည့် item များကို သာယူ၍ DataFrame တစ်ခုကို အောက်ပါအတိုင်း တည်ဆောက်နိုင်သလို ကော်လံများကိုလည်း မိမိကြိုက်သည့် နာမည်ပေးနိုင်သည်။ သက်ဆိုင်သည်တန်ဖိုးများ ရှိမနေပါက NaN ဖြင့် အစားထိုးပေး လိမ့်မည်။

- `pd.DataFrame()` ဖြင့် DataFrame တစ်ခု တည်ဆောက်သည်။
- `index=['dancy', 'ball', 'apple']` သည် မိမိ ကြိုက်နှစ်သက်သည့် row နာမည်များကို ပေးခြင်း ဖြစ်သည်။ DataFrame တွင် index သည် row ကို ဆိုလိုသည်။
- `columns=['two', 'five']` သည် မိမိ ကြိုက်နှစ်သက်သည့် column နာမည်များကို ပေးခြင်း ဖြစ်သည်။

```
pd.DataFrame(d, index=['dancy', 'ball', 'apple'],
                columns=['two', 'five'])
```

	two	five
dancy	4444.0	NaN
ball	222.0	NaN
apple	111.0	NaN

22.3 Python Dictionary မှ DataFrame တစ်ခု တည်ဆောက်ခြင်း

`data` သည် dictionary တစ်ခု ဖြစ်သည်။ ထို dictionary ကို `pd.DataFrame()` ဖြင့် pandas dataframe တစ်ခုဖြစ်အောင် တည်ဆောက်သည်။

```
data = [{'alex': 1, 'joe': 2}, {'ema': 5, 'dora': 10, 'alice': 20}]
pd.DataFrame(data)
```

	alex	joe	ema	dora	alice
0	1.0	2.0	NaN	NaN	NaN
1	NaN	NaN	5.0	10.0	20.0

22.3.1 Row နာမည် ပြောင်းခြင်း

အပေါ်က ဆောက်ပြီးခဲ့သည့် pandas dataframe ၏ row နှစ်ခု၏ နာမည်ကို 'orange' နှင့် 'red' ဟု ပြောင်းသည်။ `index=['orange', 'red']` တွင် index သည် row ဖြစ်သည်။

```
pd.DataFrame(data, index=['orange', 'red'])
```

	alex	joe	ema	dora	alice
orange	1.0	2.0	NaN	NaN	NaN
red	NaN	NaN	5.0	10.0	20.0

Pandas တွင် မိမိ ကြည်လို့သည့် ကော်လံများကိုသာ ကြည့်ရှုနိုင်သည်။ 'joe', 'dora','alice' ဆိုသည့် ကော်လံ (၃)ခုကိုသာ ဖော်ပြခိုင်းသည်။

```
pd.DataFrame(data, columns=['joe', 'dora','alice'])
```

	joe	dora	alice
0	2.0	NaN	NaN
1	NaN	10.0	20.0

22.3.2 Basic DataFrame Operations

အောက်တွင် df ဆိုသည့် DataFrame တစ်ခုကို နမူနာ အဖြစ် တည်ဆောက်ထားပြီး ဖြစ်သည်။

```
df      # df ဆိုသည့် DataFrame ကို ကြည့်ရန်
```

	one	two
apple	100.0	111.0
ball	200.0	222.0
cerill	NaN	333.0
clock	300.0	NaN
dancy	NaN	4444.0

ကော်လံနာမည်ပေး၍ DataFrame တစ်ခုမှ ထိုကော်လံအတွင်းရှိ item များကို access လုပ်နိုင်သည်။

```
df['one']          # df ဆိုသည့် DataFrame မှ ကော်လံ နာမည် 'one' ကို ကြည့်ရန်
```

```
apple     100.0
ball      200.0
cerill      NaN
clock     300.0
```

```
dancy       NaN
Name: one, dtype: float64
```

ကော်လံ 'one' နှင့် ကော်လံ 'two' ကို မြှောက်၍ ကော်လံ 'three' နာမည်ဖြင့် ကော်လံ အသစ်တစ်ခု ထပ်ထည့်သည်။

```
df['three'] = df['one'] * df['two']
df
```

	one	two	three
apple	100.0	111.0	11100.0
ball	200.0	222.0	44400.0
cerill	NaN	333.0	NaN
clock	300.0	NaN	NaN
dancy	NaN	4444.0	NaN

ကော်လံ 'one' အတွင်းမှ item များသည် 250 ထက် ပိုများလျှင် True သို့မဟုတ် ပိုနည်းလျှင် False ဆိုသည့် condition ဖြင့် 'flag' ကော်လံတစ်ခု ထပ်ထည့်သည်။

```
df['flag'] = df['one'] > 250
df
```

	one	two	three	flag
apple	100.0	111.0	11100.0	False
ball	200.0	222.0	44400.0	False
cerill	NaN	333.0	NaN	False
clock	300.0	NaN	NaN	True
dancy	NaN	4444.0	NaN	False

df ဆိုသည့် DataFrame ထဲမှာ ကော်လံ three အား **.pop( )** ဖြင့် ဖယ်ထုတ်သည်။

`three_pop = df.pop('three')` တွင် three သည် ဖယ်ထုတ်လိုက်မည့် ကော်လံမှ ဒေတာများ ဖြစ်သည်။

```
three_pop = df.pop('three')     # ကော်လံ three အား .pop()ဖြင့် ဖယ်ထုတ်သည်။
three_pop
```

```
apple     11100.0
ball      44400.0
cerill        NaN
clock         NaN
dancy         NaN
Name: three, dtype: float64
```

ကော်လံ three အား ဖယ်ထုတ်ပြီးနောက် df ကို ပြန်ကြည့်သည်။

```
df
```

	one	two	flag
apple	100.0	111.0	False
ball	200.0	222.0	False
cerill	NaN	333.0	False
clock	300.0	NaN	True
dancy	NaN	4444.0	False

`del df['two']` သည် ကော်လံ 'two' ကို ဖျက်ပစ်သည့် syntax ဖြစ်သည်။ `del` သည် delete command ဖြစ်သည်။ `df[]` သည် DataFrame ဖြစ်သည်။ `'two'` သည် ကော်လံ နာမည်ဖြစ်သည်။ `df['two']` သည် df ဆိုသည့် DataFrame မှ ကော်လံ နာမည် 'two' ကို ဆိုလိုသည်။

```
del df['two']          # ကော်လံ 'two' ကို ဖျက်ပစ်သည်။
df
```

	one	flag
apple	100.0	False
ball	200.0	False
cerill	NaN	False
clock	300.0	True
dancy	NaN	False

22.3.3 ကော်လံ တစ်ခု ထပ်ထည့်ခြင်း

`.insert()` ဖြင့် ကော်လံများ ထည့်နိုင်သည်။ `insert('loc', 'column', and 'value')` `.insert()` လုပ်ရန် ထည့်မည့်နေရာ(loc)၊ ကော်လံနာမည် နှင့် value တို့ကို ထည့်ပေးရသည်။

```
df.insert(2, 'copy_of_one', df['one'])
df
```

	one	flag	copy_of_one
apple	100.0	False	100.0
ball	200.0	False	200.0
cerill	NaN	False	NaN
clock	300.0	True	300.0
dancy	NaN	False	NaN

```
df['one_upper_half'] = df['one'][:2]   # index နံပါတ် 0 မှ 1 အထိသာဖော်ပြရန်
df
```

	one	flag	copy_of_one	one_upper_half
apple	100.0	False	100.0	100.0
ball	200.0	False	200.0	200.0
cerill	NaN	False	NaN	NaN
clock	300.0	True	300.0	NaN
dancy	NaN	False	NaN	NaN

22.4 Case Study: Movies Data Analysis

Movies dataset တစ်ခုကို အသုံးပြု၍ ဥပမာများဖြင့် ဖော်ပြမည်။

Data Source: MovieLens web site (filename: ml-20m.zip)

Location: https://grouplens.org/datasets/movielens/

22.4.1 Use Pandas to Read the Dataset

Movies dataset တွင် CSV file (၃)ခု ပါဝင်သည်။

ratings.csv ဖိုင်တွင် `userId, movieId, rating, timestamp` စသည်တို့ ပါဝင်သည်။

tags.csv ဖိုင်တွင် `userId,movieId, tag, timestamp` စသည်တို့ ပါဝင်သည်။

movies.csv ဖိုင်တွင် `movieId, title, genres` စသည်တို့ ပါဝင်သည်။

`read_csv` function ကို သုံး၍ csv ဖိုင်များကို ဖတ်သည်။

movies.csv မှ ဒေတာများကို pandas မှ `pd.read_csv()` function ကို အသုံးပြု ဖတ်ယူသည်။ လက်သည်းကွင်း() ထဲတွင် ဖိုင်နာမည် ထည့်ပေးရသည်။ `movies` သည် dataframe ၏ နာမည် ဖြစ်သည်။ `movies.head(7)` သည် movies dataframe ပထမဆုံး row (၇)ခုကို ဖော်ပြပေးရန် ဖြစ်သည်။

```
movies = pd.read_csv('movies.csv')
print(type(movies))
movies.head(7)
```

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

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children| Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action|Crime|Thriller
6	7	Sabrina (1995)	Comedy|Romance

tags.csv ကို ဖတ်၍ ဒေတာများကို ယူသည်။ head() ၏ လက်သည်းကွင်း()ထဲတွင် ဘာမျှ ထည့်ရေးလျှင် ပထမဆုံး row (၅)ခုကို ဖော်ပြပေးလိမ့်မည်။

```
# Timestamps represent seconds since midnight
# Coordinated Universal Time (UTC) of January 1, 1970
tags = pd.read_csv('tags.csv')
tags.head()
```

	userId	movieId	tag	timestamp
0	3	260	classic	1439472355
1	3	260	sci-fi	1439472256
2	4	1732	dark comedy	1573943598
3	4	1732	great dialogue	1573943604
4	4	7569	so bad it's good	1573943455

ratings.csv ကို ဖတ်၍ ဒေတာများကို ယူသည်။

```
ratings = pd.read_csv('ratings.csv')
ratings.head()
```

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185
4	1	1172	4.0	1260759205

`ratings` DataFrame မှ 'timestamp' ကော်လံကို ဖယ်ရှားသည်။ (ratings.csv မှ ဒေတာများကို တကယ် ဖျက်ပစ်ခြင်း မဟုတ်ပါ။) ။ tags DataFrame မှ 'timestamp' ကော်လံကိုသာ ဖယ်ရှားခြင်း ဖြစ်သည်။

```
# remove timestamp
del ratings['timestamp']    # ratings DataFrame မှ 'timestamp' ကော်လံကို ဖယ်ရှားသည်။
del tags['timestamp']
```

22.5 Data Structures

22.5.1 Series

`tags` DataFrame မှ ပထမဆုံး row (row နံပါတ် 0) ကို series တစ်ခုအဖြစ် ပြုလုပ်သည်။ DataFrame ထဲမှ row နံပါတ် 0 တစ်ခုတည်းကိုသာ series တစ်ခုအဖြစ် ဆွဲထုတ်ခြင်း ဖြစ်သည်။

```
# Extract 0th row: notice that it is infact a Series
row_0 = tags.iloc[0]       # row_0 ဟုနာမည် အသစ်ပေးသည်။
type(row_0)
```

```
pandas.core.series.Series
```

```
print(row_0)
```

```
userId           3
movieId        260
tag        classic
Name: 0, dtype: object
```

row_0 ၏ index များကို access လုပ်သည်။

```
row_0.index
```

```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

```
row_0['userId']
```

```
3
```

```
'rating' in row_0      # row_0 တွင် 'rating' ရှိ မရှိ စစ်သည်။
```

```
False
```

```
row_0.name             # row_0 ၏ နာမည်ကို ဖတ်သည်။
```

```
0
```

`row_0` ၏ နာမည်ကို 'first_row' ဟု နာမည်ပြောင်းသည်။

```
row_0 = row_0.rename('first_row')
row_0.name
```

```
'first_row'
```

22.5.2 DataFrames

`tags` DataFrame ၏ ပထမဆုံး row (၅)ခုကို ကြည့်ရန်

```
tags.head()
```

	userId	movieId	tag
0	3	260	classic
1	3	260	sci-fi
2	4	1732	dark comedy
3	4	1732	great dialogue
4	4	7569	so bad it's good

`tags` DataFrame ၏ row index များကို ကြည့်ရန်

```
tags.index
```

```
RangeIndex(start=0, stop=1093360, step=1)
```

`tags` DataFrame ၏ ကော်လံ index များကို `.columns` ဖြင့် ကြည့်သည်။

```
tags.columns   # tags DataFrame ၏ ကော်လံနာမည်များကို ဖော်ပြရန်
```

```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

row index နံပါတ် 0 ၊ 11 နှင့် 2000 တို့၏ ဒေတာများကို ကြည့်ရန်

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

	userId	movieId	tag
0	3	260	classic
11	4	164909	cliche
2000	647	164179	twist ending

22.6 Descriptive Statistics

`ratings` dataframe နှင့် သက်ဆိုင်သည့် အချက်အလက်များ(informaiton)ကို ဖတ်ရန်

```
ratings.info()
```

`ratings['rating']` သည် `ratings` dataframe မှ rating ကော်လံ တစ်ခုတည်း၏ statistical အချက်အလက်များကို ဖော်ပြပေးရန် ဖြစ်သည်။ `.describe()` ကို အသုံးပြုသည်။

```
ratings['rating'].describe()
```

```
count    100004.000000
mean          3.543608
std           1.058064
min           0.500000
25%           3.000000
50%           4.000000
75%           4.000000
max           5.000000
Name: rating, dtype: float64
```

`ratings.describe()` သည် `ratings` dataframe မှ userId ၊ movieId ၊ rating စသည့် ကော်လံများ အားလုံး၏ statistical အချက်အလက်များကို ဖော်ပြပေးရန် ဖြစ်သည်။

```
ratings.describe()
```

	userId	movieId	rating
count	100004.000000	100004.000000	100004.000000
mean	347.011310	12548.664363	3.543608
std	195.163838	26369.198969	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

`ratings['rating'].mean()` သည် `ratings` dataframe မှ rating ကော်လံ၏ mean တန်ဖိုး တစ်ခုတည်းကိုသာ ဖော်ပြပေးရန် ဖြစ်သည်။

```
ratings['rating'].mean()
```

```
3.543608255669773
```

`ratings.mean()` သည် `ratings` dataframe မှ ကော်လံများအားလုံး၏ mean တန်ဖိုးကို ဖော်ပြပေးရန် ဖြစ်သည်။

```
ratings.mean()     # ကော်လံများအားလုံး၏ mean တန်ဖိုးကို ဖော်ပြရန်
```

```
userId       347.011310
```

```
movieId    12548.664363
rating         3.543608
dtype: float64
```

```
ratings['rating'].min()        # အနည်းဆုံး တန်ဖိုး(minimum  value)ကို ဖော်ပြရန်
```

```
0.5
```

```
ratings['rating'].max()        # အများဆုံး တန်ဖိုး(maximum value)ကို ဖော်ပြရန်
```

```
5.0
```

`.std()` သည် မိမိသိလိုသည့် axis (row သို့မဟုတ် ကော်လံ)၏ standard deviation ကို ဖော်ပြ ပေးသည်။

```
ratings['rating'].std()      # standard deviation ကို ဖော်ပြရန်
```

```
1.0580641091073735
```

`.corr()` ဖြင့် correlation matrix ကို ဖော်ပြနိုင်သည်။ Column များ၏ pairwise correlation ကို တွက်ပေးသည်။ တန်ဖိုးမဲ့သည့် ဒေတာ(missing value)များကို ထည့်မတွက်ပါ။ (excluding NA/null values.)

```
ratings.corr()
```

	userId	movieId	rating
userId	1.000000	0.007126	0.010467
movieId	0.007126	1.000000	-0.028894
rating	0.010467	-0.028894	1.000000

ကော်လံ 'rating' အတွင်းမှ item များသည် 5 ထက်ပိုများလျှင် True သို့မဟုတ် ပိုနည်းလျှင် False ဖော်ပြရန်

```
filter_1 = ratings['rating'] > 5
print(filter_1)
filter_1.any()
```

```
0         False
1         False
2         False
          ...
100001    False
100002    False
100003    False
Name: rating, Length: 100004, dtype: bool
False
```

```
filter_2 = ratings['rating'] > 0
filter_2.all()
```

```
True
```

22.7 Data Cleaning: Handling Missing Data

`movies.shape` သည် `movies` DataFrame ၏ row အရေအတွက် နှင့် ကော်လံ အရေအတွက်ကို ဖော်ပြပေးရန် ဖြစ်သည်။ `.shape` ကို အသုံးပြုသည်။

```
movies.shape        # row အရေအတွက် နှင့် ကော်လံအရေအတွက်ကို ဖော်ပြရန်
```

```
(9125, 3)
```

`movies` DataFrame တွင် missing value များ ရှိ, မရှိ စစ်ခြင်း ဖြစ်သည်။ **isnull()** သည် missing value များ ရှိ မရှိ စစ်ခြင်း ဖြစ်သည်။ **.any()** သည် ရှိ, မရှိကို ကော်လံအလိုက် ဖော်ပြပေးသည်။ **isnull()** ဖြင့် စစ်လျှင် DataFrame တစ်ခုလုံးကို ဖော်ပြသည်။

```
#is any row NULL ?
movies.isnull().any()
```

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

`ratings.shape` သည် `ratings` DataFrame ၏ row အရေအတွက် နှင့် ကော်လံ အရေအတွက်ကို ဖော်ပြပေးရန် ဖြစ်သည်။

```
ratings.shape
```

```
(100004, 3)
```

`ratings` DataFrame တွင် missing value များ ရှိ မရှိ စစ်ခြင်း ဖြစ်သည်။ **isnull()** သည် missing value များ ရှိ, မရှိစစ်ခြင်း ဖြစ်သည်။ Pandas သည် missing value ကို ကောင်းစွာ စီမံနိုင်အောင် ဒီဇိုင်း လုပ်ထားသည်။ **.any()** သည် ကော်လံအလိုက် ဖော်ပြပေးသည်။ **isnull()** ဖြင့် စစ်လျှင် DataFrame တစ်ခုလုံးကို ဖော်ပြသည်။

```
#is any row NULL ?
ratings.isnull().any()
```

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

NULL value တစ်ခုမှ မရှိပါ။

```
tags.shape
```

```
(1093360, 3)
```

```
#is any row NULL ?
tags.isnull().any()
```

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

`tags` DataFrame တွင် missing value များ ရှိသည်။ ထို missing value များကို ဖယ်ထုတ်ရန် `.dropna()` ကို အသုံးပြုသည်။ (drop Na)

```
tags = tags.dropna()
#Check again: is any row NULL ?
tags.isnull().any()
```

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

```
tags.shape          # row အရေအတွက် နှင့် ကော်လံ အရေ အတွက်ကို ပြန်ကြည့်သည်။
```

```
(1093344, 3)
```

ထို missing value များကို ဖယ်ထုတ်လိုက်သောကြောင့် row အရေအတွက် လျော့နည်းသွားသည်။ (No NULL values ! Notice the number of lines have reduced.)

22.8 Slicing Out Columns

`.head()` ဖြင့် `tags` DataFrame မှ 'tag' ကော်လံ၏ ပထမဆုံး (၅)ခုကို ဖော်ပြသည်။

```
tags['tag'].head()
```

```
0             classic
1              sci-fi
2         dark comedy
3      great dialogue
4    so bad it's good
Name: tag, dtype: object
```

`.head()` ဖြင့် movies DataFrame မှ `'title'` နှင့် `'genres'` ကော်လံတို့၏ ပထမဆုံး (၅)ခုကို ကြည့်သည်။

```
movies[['title','genres']].head()
```

	title	genres

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

`ratings` DataFrame မှ နောက်ဆုံး (၅)ခုကို ကြည့်သည်။

```
ratings[-5:]
```

	userId	movieId	rating
99999	671	6268	2.5
100000	671	6269	4.0
100001	671	6365	4.0
100002	671	6385	2.5
100003	671	6565	3.5

`ratings` DataFrame မှ နောက်ဆုံး (၅)ခုကို ဖော်ပြရန် `.tail()` ကို အသုံးပြုသည်။

```
ratings.tail()                # နောက်ဆုံး (၅) ခုကို ဖော်ပြရန်
```

	userId	movieId	rating
99999	671	6268	2.5
100000	671	6269	4.0
100001	671	6365	4.0
100002	671	6385	2.5
100003	671	6565	3.5

`tags` DataFrame မှ 'tag' ကော်လံမှ ရုပ်ရှင်အမျိုးအစားများကို ရေတွက်ရန် `.value_counts()`ကို အသုံးပြုသည်။ ပထမဆုံး(၅)ခုကို ဖော်ပြသည်။

```
tag_counts = tags['tag'].value_counts()
tag_counts[:5]
```

```
sci-fi          8330
atmospheric     6516
action          5907
comedy          5702
surreal         5326
Name: tag, dtype: int64
```

`tags` DataFrame မှ 'tag' ကော်လံမှ ရုပ်ရှင်အမျိုးအစားများကို ရေတွက်ပြီး နောက်ဆုံး(၅)ခုကို ဖော်ပြရန်

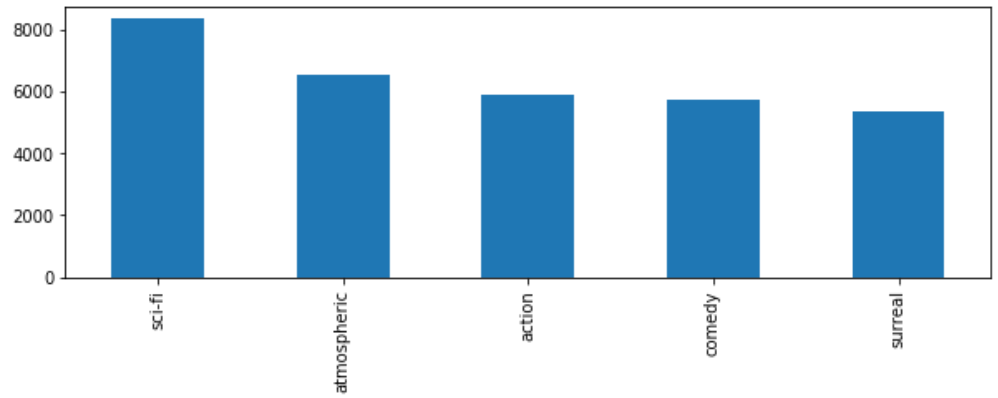
```
tag_counts = tags['tag'].value_counts()
tag_counts[-5:]
```

```
Qatar                    1
ver_com_a_Tayma          1
firearm malfunction      1
coping with lose         1
road runner              1
Name: tag, dtype: int64
```

`tags` DataFrame မှ 'tag' ကော်လံမှ ရုပ်ရှင်အမျိုးအစားများကို ရေတွက်ပြီး အများဆုံး (၅)ခုကိုဘား ဂရပ်ဖြင့် ဖော်ပြရန်

```
tag_counts[:5].plot(kind='bar', figsize=(10,3))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e9963f2248>
```



22.9 Filters for Selecting Rows

`ratings` DataFrame မှ 'rating' ကော်လံရှိတန်ဖိုးများ 4.0 သို့မဟုတ် 4.0 များလျှင် rating ကောင်းသည် (highly_rated)ဟု သတ်မှတ်သည်။ ထို rating ကောင်းသည်(highly_rated) ရုပ်ရှင်များထဲမှ ပထမဆုံး (၅)ကိုသာ ဖော်ပြရန်

```
is_highly_rated = ratings['rating'] >= 4.0
ratings[is_highly_rated][:5]
```

	userId	movieId	rating
4	1	1172	4.0
12	1	1953	4.0
13	1	2105	4.0
20	2	10	4.0
21	2	17	5.0

`movies` DataFrame မှ 'genres' ကော်လံတွင် 'Animation' ဟု ဖော်ထားလျှင် animation ပါသည့် ရုပ်ရှင် ဖြစ်သည်။ ထို Animation ရုပ်ရှင်ကားများ ဟုတ်, မဟုတ်စစ်ဆေးပြီး ရုပ်ရှင်များထဲမှ ပထမဆုံး (၅)ကိုသာ ကြည့်သည်။ ဖော်ပြခိုင်းသည်။

```
is_animation = movies['genres'].str.contains('Animation')
movies[is_animation].head(5)
```

	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

22.10 Group By and Aggregate

```
ratings[['movieId','rating']].head(5)
```

	movieId	rating
0	31	2.5
1	1029	3.0
2	1061	3.0
3	1129	2.0
4	1172	4.0

`'rating'` ကော်လံမှ တန်ဖိုးများ တူရာစုပြီး ရေတွက်ပါ။ `.groupby('rating')` သည် 'rating' ကော်လံမှ တန်ဖိုးများ တူရာစုခြင်း ဖြစ်သည်။ ဥပမာ rating = 3.0 ဖြစ်သည့် ရုပ်ရှင်ကားများ အားလုံးကို စုပေါင်း၍ ဘယ်နှစ်ခု ရှိသည်ကို ရေတွက်သည်။ `.groupby()` ဖြင့်တူရာစုသည် `.count()` ဖြင့် မည်မျှ ရှိသည်ကို ရေတွက်ပေးသည်။

```
ratings_count = ratings[['movieId','rating']].groupby('rating').count()
ratings_count
```

rating	movieId
0.5	1101
1.0	3326
1.5	1687
2.0	7271

2.5	4449
3.0	20064
3.5	10538
4.0	28750
4.5	7723
5.0	15095

`'movieId'` ကော်လံမှ တန်ဖိုးများကို `.groupby('movieId')` ဖြင့် တူရာစုပြီး ပျမ်းမျှ တန်ဖိုး (mean) ရှာသည်။

```
average_rating =ratings[['movieId','rating']].groupby('movieId').mean()
average_rating.head()              # ပထမ (၅)ခုကို ဖော်ပြပေးသည်။
```

movieId	rating
1	3.872470
2	3.401869
3	3.161017
4	2.384615
5	3.267857

'movieId' ကော်လံမှ တန်ဖိုးများ တူရာစုပြီး ရေတွက်သည်။

```
movie_count = ratings[['movieId','rating']].groupby('movieId').count()
movie_count.head()
```

movieId	rating
1	247
2	107
3	59
4	13
5	56

```
movie_count = ratings[['movieId','rating']].groupby('movieId').count()
movie_count.tail()
```

movieId	rating
161944	1
162376	1
162542	1
162672	1
163949	1

22.11 DataFrame များ ပေါင်းစည်းခြင်း(Merge DataFrames)

`movies` DataFrame နှင့် tags DataFrame ကို ပေါင်းမည်။

```
tags.head()
```

	userId	movieId	tag
0	3	260	classic
1	3	260	sci-fi
2	4	1732	dark comedy
3	4	1732	great dialogue
4	4	7569	so bad it's good

```
movies.head()
```

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

```
tags.head(2)
```

	userId	movieId	tag
0	3	260	classic
1	3	260	sci-fi

```
movies.head(2)
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy

`movies` DataFrame နှင့် `tags` DataFrame ကို ပေါင်းမည်။ မည်သည့်နည်းဖြင့် ပေါင်းထည့်မည်ဆိုသည်ကို ပြောပေးရန် လိုသည်။ `how='inner'` ဆိုသည်မှာ DataFrame နှစ်ခု စလုံးမှ key များကို intersection လုပ်သည့်နည်းဖြင့် ပေါင်းခြင်း ဖြစ်သည်။ SQL မှ inner join နှင့် တူသည်။

```
t = movies.merge(tags, on='movieId', how='inner')
t.head()
```

	movieId	title	genres	userId	tag
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	791	Owned
1	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1048	imdb top 250
2	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	1361	Pixar
3	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	3164	Pixar
4	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	3164	time travel

More examples: http://pandas.pydata.org/pandas-docs/stable/merging.html

22.12 Combine Aggreagation, Merging, and Filters to Get Useful Analytics

```
avg_ratings = ratings.groupby('movieId', as_index=False).mean()
avg_ratings.head()
```

	movieId	userId	rating
0	1	338.558704	3.872470
1	2	318.906542	3.401869
2	3	374.423729	3.161017
3	4	355.538462	2.384615
4	5	320.785714	3.267857

`'userId'` ကော်လံကို ဖယ်ထုတ်သည်။

```
avg_ratings = ratings.groupby('movieId', as_index=False).mean()
del avg_ratings['userId']
avg_ratings.head()
```

	movieId	rating
0	1	3.872470
1	2	3.401869
2	3	3.161017
3	4	2.384615
4	5	3.267857

`movies` DataFrame နှင့် အထက်က `avg_ratings` DataFrame ကို ပေါင်းသည်။ ပေါင်း(merge လုပ်)ပြီး box_office ဆိုသည့် DataFrame တစ်ခုတည်ဆောက်သည်။

```
box_office = movies.merge(avg_ratings, on='movieId', how='inner')
box_office.tail()
```

	movieId	title	genres	rating
9061	161944	The Last Brickmaker in America (2001)	Drama	5.0
9062	162376	Stranger Things	Drama	4.5
9063	162542	Rustom (2016)	Romance|Thriller	5.0
9064	162672	Mohenjo Daro (2016)	Adventure|Drama|Romance	3.0
9065	163949	The Beatles: Eight Days a Week - The Touring Y...	Documentary	5.0

`rating` မြင့်သည့် ရုပ်ရှင်ကားများကို စစ်ထုတ်သည်။ `box_office['rating'] >= 4.0` ဖြင့် rating 4.0 နှင့် အထက် ရသည့် ရုပ်ရှင်ကားများကို ရွေးထုတ်သည်။

```
is_highly_rated = box_office['rating'] >= 4.0

box_office[is_highly_rated][-5:]
```

	movieId	title	genres	rating
9055	160718	Piper (2016)	Animation	4.0
9061	161944	The Last Brickmaker in America (2001)	Drama	5.0
9062	162376	Stranger Things	Drama	4.5
9063	162542	Rustom (2016)	Romance|Thriller	5.0
9065	163949	The Beatles: Eight Days a Week - The Touring Y...	Documentary	5.0

`.str.contains('Comedy')` ဖြင့် ဟာသ ရုပ်ရှင်ကား(comedy)များကို ရွေးထုတ်သည်။

```
is_comedy = box_office['genres'].str.contains('Comedy')
box_office[is_comedy][:5]
```

	movieId	title	genres	rating
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy	3.872470
2	3	Grumpier Old Men (1995)	Comedy|Romance	3.161017
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance	2.384615

4	5	Father of the Bride Part II (1995)	Comedy	3.267857
6	7	Sabrina (1995)	Comedy|Romance	3.283019

`is_comedy` ဟာသ ရုပ်ရှင်ကားသာ ပါသည့် ဒေတာများ ဖြစ်သည်။ `is_highly_rated` သည် rating မြင့်သည့် ကားများ ဖြစ်သည်။ **&** operator ကို သုံး၍ ကောင်းသည့်(rating မြင့်သည့်) ဟာသ ရုပ်ရှင်ကား (comedy)များကိုသာ ရွေးထုတ်သည်။

```
box_office[is_comedy & is_highly_rated][-5:]
```

	movieId	title	genres	rating
9019	152081	Zootopia (2016)	Action|Adventure|Animation|Children|Comedy	4.0
9023	153584	The Last Days of Emma Blank (2009)	Comedy	5.0
9027	156025	Ice Age: The Great Egg-Scapade (2016)	Adventure|Animation|Children|Comedy	5.0
9037	158314	Daniel Tosh: Completely Serious (2007)	Comedy	4.5
9052	160567	Mike & Dave Need Wedding Dates (2016)	Comedy	4.0

22.13 Vectorized String Operations

```
movies.head()
```

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

22.14 Split 'genres' Into Multiple Columns

`.str.split()` ဖြင့် `'genres'` ကော်လံမှ စာသားများကို ခွဲထုတ် ပစ်သည်။

```
movie_genres = movies['genres'].str.split('|', expand=True)
movie_genres[:10]
```

	0	1	2	3	4	5	6	7	8	9

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
6	Comedy	Romance	None	None	None	None	None	None	None	None
7	Adventure	Children	None	None	None	None	None	None	None	None
8	Action	None	None	None	None	None	None	None	None	None
9	Action	Adventure	Thriller	None	None	None	None	None	None	None

22.15 Add a new column for comedy genre flag

ဟာသ ရုပ်ရှင်ကား(comedy)များအတွက် ဖော်ပြသည့် ကော်လံတစ်ခု ထပ်ထည့်ရန်

```
movie_genres['isComedy'] = movies['genres'].str.contains('Comedy')
movie_genres[:10]
```

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

22.16 Extract Year From Title e.g. (1995)

`'title'` ကော်လံ မှ ရုပ်ရှင်ကားနာမည်ကို ဖတ်ပြီး ခုနှစ်များကို ယူ၍ Dataframe တွင် year ကော်လံ တစ်ခုဖြင့် ထည့်၍ ဖော်ပြရန်

```
movies['year']=movies['title'].str.extract('.*\((.*)\).*', expand=True)
movies.tail()
```

	movieId	title	genres	year
9120	162672	Mohenjo Daro (2016)	Adventure|Drama|Romance	2016
9121	163056	Shin Godzilla (2016)	Action|Adventure|Fantasy|Sci-Fi	2016

9122	163949	The Beatles: Eight Days a Week - The Touring Y...	Documentary	2016
9123	164977	The Gay Desperado (1936)	Comedy	1936
9124	164979	Women of '69, Unboxed	Documentary	NaN

More here: http://pandas.pydata.org/pandas-docs/stable/text.html#text-string-methods

End-

Additional Recommended Resources:

- *pandas* Documentation: http://pandas.pydata.org/pandas-docs/stable/
- *Python for Data Analysis* by Wes McKinney
- *Python Data Science Handbook* by Jake VanderPlas