Hi guys, this is Akshansh (+91 8384891269, Akshanshofficial@gmail.com (mailto:Akshanshofficial@gmail.com)). We have discussed Pandas Basic in lecture notes 17 & 18 and this is continuation of those lectures. I hope y'all are doing great with Python. Let's continue on that-

In [1]:

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

I'll be focused on data input and output using pandas. I'll be using following commands to parse files -

```
read_csv = loads comma seperator file
read_table =loads tab seperator file
read_fwf =loads data in fixed width column format (no seprator)
read_clipboard = version of read_table data from clipboard
read_excel=loads excel files
```

handling dates and other custom types can require extra effort. Let's start with a smaill comma-seperated file

In [2]:

```
pwd #checking current directory
```

Out[2]:

```
'/home/akshansh'
```

In [3]:

```
cd Desktop/ #current directory is Desktop now
```

```
[Errno 2] No such file or directory: 'Desktop/ #current directory is D
esktop now'
/home/akshansh
```

What I am gonna do here is, creating a comma seperated file and then I am going to save in my current directory name Dasktop. I am gonna name it "practice_01.csv" , to work with me, you have to name the file like i did so that you can work along with me.

practice_01.csv

In [4]:

```
frame=pd.DataFrame(np.arange(16).reshape((4,4)),
                  index=['red','blue','yellow','white'],
                  columns=['ball','pen','pencil','paper'])

```

In [5]:

```
frame #i created a frame which is shown below
```

Out[5]:

	ball	pen	pencil	paper
red	0	1	2	3
blue	4	5	6	7
yellow	8	9	10	11
white	12	13	14	15

let's save this to csv format and name it 'practice_01.csv'

In [6]:

```
frame.to_csv('practice_01.csv')
```

let's read it

In [7]:

```
df=pd.read_csv('practice_01.csv')
```

In [8]:

```
df #prints what practice_01.csv has
```

Out[8]:

	Unnamed: 0	ball	pen	pencil	paper
0	red	0	1	2	3
1	blue	4	5	6	7
2	yellow	8	9	10	11
3	white	12	13	14	15

This is how can write .csv files and then save them too. Anyway, I am choosig a common file that can be accessible to all so that we all can perform same actions together on a common file.

Let's grab a file from bit.ly which contains all the movie ratings so far. File has been written in csv format. You can access it like this-

In [9]:

```
df1=pd.read_csv('https://bit.ly/imdbratings')
```

I named this csv file 'df1'.

##AkshanshTips - I showed you that DataFrame can be converted like csv file. So all the operations or methods we have used for the DataFrame in lecture number 18/17 combined. Put this under your pillow, this is going to save your ass in the time of crisis.

1) let's print what our dataframe df1 has

In [10]:

```
df1
```

Out[10]:

	star_rating	title	content_rating	genre	duration	actors_list
0	9.3	The Shawshank Redemption	R	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	9.2	The Godfather	R	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	9.1	The Godfather: Part II	R	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	9.0	The Dark Knight	PG-13	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	8.9	Pulp Fiction	R	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....
...	...	...	...	...	...	...
974	7.4	Tootsie	PG	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	7.4	Back to the Future Part III	PG	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	7.4	Master and Commander: The Far Side of the World	PG-13	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	7.4	Poltergeist	PG	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	7.4	Wall Street	R	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

979 rows × 6 columns

It has 979 rows and 6 columns

2) head()

when you file has too many rows, you can use head() method to see first 5 lines of your dataset. By default it will take 5 lines but you can pass any number of lines.

In [11]:

```
df1.head()  #prints first 5 rows
```

Out[11]:

	star_rating	title	content_rating	genre	duration	actors_list
0	9.3	The Shawshank Redemption	R	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	9.2	The Godfather	R	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	9.1	The Godfather: Part II	R	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	9.0	The Dark Knight	PG-13	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	8.9	Pulp Fiction	R	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....

In [12]:

```
df1.head(3)  #prints first 3 rows
```

Out[12]:

	star_rating	title	content_rating	genre	duration	actors_list
0	9.3	The Shawshank Redemption	R	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	9.2	The Godfather	R	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	9.1	The Godfather: Part II	R	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...

2)tail()

head() gives you the view of data from top then what will tail() give you? A view of data from bottom. It works like head() and by default it is also set on 5.

In [13]:

```
df1.tail()  #prints last 5 rows
```

Out[13]:

	star_rating	title	content_rating	genre	duration	actors_list
974	7.4	Tootsie	PG	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	7.4	Back to the Future Part III	PG	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	7.4	Master and Commander: The Far Side of the World	PG-13	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	7.4	Poltergeist	PG	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	7.4	Wall Street	R	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

In [14]:

```
df1.tail(4)  #prints last 4 rows
```

Out[14]:

	star_rating	title	content_rating	genre	duration	actors_list
975	7.4	Back to the Future Part III	PG	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	7.4	Master and Commander: The Far Side of the World	PG-13	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	7.4	Poltergeist	PG	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	7.4	Wall Street	R	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

3) changing orders of cols

```
look at your DataFrame, cols order are-
star_rating,title,content_rating,genre,duration,actors_list
I want to change them as
title,star_rating,genre,duration,content_ratings,actors_list
```

##AkshanshTips- Treat you file as DataFrame, becuase it is DataFrame not your GF's best male friend. No need to afraid of.

In [15]:

```
df2=pd.DataFrame(df1,columns=['title','star_rating','genre','duration','content
```

In [16]:

```
df2 #see what we got?
```

Out[16]:

	title	star_rating	genre	duration	content_rating	actors_list
0	The Shawshank Redemption	9.3	Crime	142	R	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	The Godfather	9.2	Crime	175	R	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	The Godfather: Part II	9.1	Crime	200	R	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	The Dark Knight	9.0	Action	152	PG-13	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	Pulp Fiction	8.9	Crime	154	R	[u'John Travolta', u'Uma Thurman', u'Samuel L....
...	...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	PG	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	Back to the Future Part III	7.4	Adventure	118	PG	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	Master and Commander: The Far Side of the World	7.4	Action	138	PG-13	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	Poltergeist	7.4	Horror	114	PG	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	Wall Street	7.4	Crime	126	R	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

979 rows × 6 columns

Notice at those cols order, successfully changed. Easy? Definitely!!

4) adding new col

We will do like we did in lecture number 18 while playing with DataFrames. Just pass a new col while treating it as a DataFrame

In [17]:

```
df3=pd.DataFrame(df2,columns=['title','star_rating','genre','duration','actors_
```

In [18]:

```
df3 #let's see if it is added or not
```

Out[18]:

	title	star_rating	genre	duration	actors_list	new
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...	NaN
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']	NaN
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...	NaN
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...	NaN
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....	NaN
...	...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...	NaN
975	Back to the Future Part III	7.4	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...	NaN
976	Master and Commander: The Far Side of the World	7.4	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...	NaN
977	Poltergeist	7.4	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...	NaN
978	Wall Street	7.4	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...	NaN

979 rows × 6 columns

Aww, look at the right corner, you'll find 'new' col right there. Isn't is cute? Boys will not find it cute (i didn't fount that cute) but girls will, and tiktokers too :-P. And one more thing to notice, all the values in col 'new' is NaN (not a number) type. We have talked about this in lecture number 18.

##AkshanshTips - New col can be created by this too -

In [19]:

```
df2['new']=np.nan  #look we stated df2, because df3 alreay has 'new'
```

What will it do, it will go to df2, look for col name 'new', if there is a col with this name, all the values will become np.nan (NaN) otherwise it will create a new col named 'new' and set all the value to NaN type

In [20]:

```
df2 #let's check if everything went right, finger crossed he he
```

Out[20]:

	title	star_rating	genre	duration	content_rating	actors_list	new
0	The Shawshank Redemption	9.3	Crime	142	R	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...	NaN
1	The Godfather	9.2	Crime	175	R	[u'Marlon Brando', u'Al Pacino', u'James Caan']	NaN
2	The Godfather: Part II	9.1	Crime	200	R	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...	NaN
3	The Dark Knight	9.0	Action	152	PG-13	[u'Christian Bale', u'Heath Ledger', u'Aaron E...	NaN
4	Pulp Fiction	8.9	Crime	154	R	[u'John Travolta', u'Uma Thurman', u'Samuel L....	NaN
...	...	...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	PG	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...	NaN
975	Back to the Future Part III	7.4	Adventure	118	PG	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...	NaN
976	Master and Commander: The Far Side of the World	7.4	Action	138	PG-13	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...	NaN
977	Poltergeist	7.4	Horror	114	PG	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...	NaN
978	Wall Street	7.4	Crime	126	R	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...	NaN

979 rows × 7 columns

5) Accessing column/s

our recent DataFrame was df3, I'll be working with that. Let's access what df3 has in specific col

In [21]:

```
df3['title'] #prints what is stored in col title
```

Out[21]:

```
0                             The Shawshank Redemption
1                                        The Godfather
2                               The Godfather: Part II
3                                      The Dark Knight
4                                         Pulp Fiction
                            ...
974                                            Tootsie
975                        Back to the Future Part III
976    Master and Commander: The Far Side of the World
977                                        Poltergeist
978                                        Wall Street
Name: title, Length: 979, dtype: object
```

##AkshanshTips- it can be accessed like df3.title

In [22]:

```
df3.title
```

Out[22]:

```
0                             The Shawshank Redemption
1                                        The Godfather
2                               The Godfather: Part II
3                                      The Dark Knight
4                                         Pulp Fiction
                            ...
974                                            Tootsie
975                        Back to the Future Part III
976    Master and Commander: The Far Side of the World
977                                        Poltergeist
978                                        Wall Street
Name: title, Length: 979, dtype: object
```

Col->Series then we play around. I show you how?

In [23]:

```
Title=df3['title'] #let's save this for later use
```

In [24]:

```
Title #check waht is in Title
```

Out[24]:

```
0                             The Shawshank Redemption
1                                        The Godfather
2                               The Godfather: Part II
3                                      The Dark Knight
4                                         Pulp Fiction
                            ...
974                                            Tootsie
975                        Back to the Future Part III
976    Master and Commander: The Far Side of the World
977                                        Poltergeist
978                                        Wall Street
Name: title, Length: 979, dtype: object
```

In [25]:

```
type(Title) #checking the Datatype
```

Out[25]:

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

In [26]:

```
Title.head(10) #now applying head(10) to know top 10 movies
```

Out[26]:

```
0                         The Shawshank Redemption
1                                    The Godfather
2                           The Godfather: Part II
3                                  The Dark Knight
4                                     Pulp Fiction
5                                     12 Angry Men
6                   The Good, the Bad and the Ugly
7    The Lord of the Rings: The Return of the King
8                                 Schindler's List
9                                       Fight Club
Name: title, dtype: object
```

This is how you can save each col of DataFrame which will definitely be a Series and all the methods those were aplicable to Series and Dataframe can be applied on that. Thank me later, I just gave you top 10 highest rated movies of all time.

What if you want to print more than 1 col

In [27]:

```
df3[['title','star_rating']]  #prints title and star_ratings together
```

Out[27]:

	title	star_rating
0	The Shawshank Redemption	9.3
1	The Godfather	9.2
2	The Godfather: Part II	9.1
3	The Dark Knight	9.0
4	Pulp Fiction	8.9
...	...	...
974	Tootsie	7.4
975	Back to the Future Part III	7.4
976	Master and Commander: The Far Side of the World	7.4
977	Poltergeist	7.4
978	Wall Street	7.4

979 rows × 2 columns

In [28]:

```
df3[['title','actors_list']]  #prints movies and actors in it
```

Out[28]:

	title	actors_list
0	The Shawshank Redemption	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	The Godfather	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	The Godfather: Part II	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	The Dark Knight	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	Pulp Fiction	[u'John Travolta', u'Uma Thurman', u'Samuel L....
...	...	...
974	Tootsie	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	Back to the Future Part III	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	Master and Commander: The Far Side of the World	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	Poltergeist	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	Wall Street	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

979 rows × 2 columns

6) accessing rows

In [29]:

```
df3 #let's see what df3 has
```

Out[29]:

	title	star_rating	genre	duration	actors_list	new
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...	NaN
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']	NaN
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...	NaN
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...	NaN
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....	NaN
...	...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...	NaN
975	Back to the Future Part III	7.4	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...	NaN
976	Master and Commander: The Far Side of the World	7.4	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...	NaN
977	Poltergeist	7.4	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...	NaN
978	Wall Street	7.4	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...	NaN

979 rows × 6 columns

In [30]:

```
df3[0:10]  #prints rows from 1 to 9(10-1), easy, like easily your gf got you.Jus
```

Out[30]:

	title	star_rating	genre	duration	actors_list	new
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...	NaN
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']	NaN
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...	NaN
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...	NaN
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....	NaN
5	12 Angry Men	8.9	Drama	96	[u'Henry Fonda', u'Lee J. Cobb', u'Martin Bals...	NaN
6	The Good, the Bad and the Ugly	8.9	Western	161	[u'Clint Eastwood', u'Eli Wallach', u'Lee Van ...	NaN
7	The Lord of the Rings: The Return of the King	8.9	Adventure	201	[u'Elijah Wood', u'Viggo Mortensen', u'Ian McK...	NaN
8	Schindler's List	8.9	Biography	195	[u'Liam Neeson', u'Ralph Fiennes', u'Ben Kings...	NaN
9	Fight Club	8.9	Drama	139	[u'Brad Pitt', u'Edward Norton', u'Helena Bonh...	NaN

In [31]:

```
df3[1:2]  #it includes 1 but not 2, it includes last-1 value
```

Out[31]:

	title	star_rating	genre	duration	actors_list	new
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']	NaN

##AkshanshTips - use loc for this

In [32]:

```
df3.loc[1]  #it tells you want index 1 (row2) has on each col
```

Out[32]:

```
title                                                The Godfather
star_rating                                                    9.2
genre                                                        Crime
duration                                                       175
actors_list     [u'Marlon Brando', u'Al Pacino', u'James Caan']
new                                                            NaN
Name: 1, dtype: object
```

In [33]:

```
df3.loc[0:10]  #its bit different it includes 0 and 10 too for rows, use this it
```

Out[33]:

	title	star_rating	genre	duration	actors_list	new
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...	NaN
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']	NaN
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...	NaN
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...	NaN
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....	NaN
5	12 Angry Men	8.9	Drama	96	[u'Henry Fonda', u'Lee J. Cobb', u'Martin Bals...	NaN
6	The Good, the Bad and the Ugly	8.9	Western	161	[u'Clint Eastwood', u'Eli Wallach', u'Lee Van ...	NaN
7	The Lord of the Rings: The Return of the King	8.9	Adventure	201	[u'Elijah Wood', u'Viggo Mortensen', u'Ian McK...	NaN
8	Schindler's List	8.9	Biography	195	[u'Liam Neeson', u'Ralph Fiennes', u'Ben Kings...	NaN
9	Fight Club	8.9	Drama	139	[u'Brad Pitt', u'Edward Norton', u'Helena Bonh...	NaN
10	The Lord of the Rings: The Fellowship of the Ring	8.8	Adventure	178	[u'Elijah Wood', u'Ian McKellen', u'Orlando Bl...	NaN

7) filling NaN values

NaN values are like your Facebook girlfriends, they are there but there is no use of them. No practically saying, they can't cook food for you, can't go outside with you, can't make love with you. Don't give me shit that she exchanges GIF with you and send some picture and make you feel heaven. That's not heaven thats stupidity. :-P

It is always good to remove or filter you NaN value. AkshanshTips here again, focus on memes rather than those girls who don't want to talk to you. They'all using you. You are a boy, not a bell of some temple.

In [34]:

```
df3.fillna(0)  #fill NaN values with 0, look at col new
```

Out[34]:

	title	star_rating	genre	duration	actors_list	new
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...	0.0
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']	0.0
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...	0.0
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...	0.0
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....	0.0
...	...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...	0.0
975	Back to the Future Part III	7.4	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...	0.0
976	Master and Commander: The Far Side of the World	7.4	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...	0.0
977	Poltergeist	7.4	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...	0.0
978	Wall Street	7.4	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...	0.0

979 rows × 6 columns

##AkshanshTips- df3['new']=0.0 will do the same

In [35]:

```
df3['new']= 0
```

In [36]:

```
df3 #checks df3, look at col new
```

Out[36]:

	title	star_rating	genre	duration	actors_list	new
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...	0
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']	0
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...	0
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...	0
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....	0
...	...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...	0
975	Back to the Future Part III	7.4	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...	0
976	Master and Commander: The Far Side of the World	7.4	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...	0
977	Poltergeist	7.4	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...	0
978	Wall Street	7.4	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...	0

979 rows × 6 columns

8) Take advantage of Numpy to fill a col

I know there are 979 rows

In [37]:

```
df3['new']=np.arange(1,980)
```

In [38]:

```
df3 #look new col, it has filled with 1 - 979
```

Out[38]:

	title	star_rating	genre	duration	actors_list	new
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...	1
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']	2
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...	3
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...	4
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....	5
...	...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...	975
975	Back to the Future Part III	7.4	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...	976
976	Master and Commander: The Far Side of the World	7.4	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...	977
977	Poltergeist	7.4	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...	978
978	Wall Street	7.4	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...	979

979 rows × 6 columns

##AkshanshTips- you can fil the values at desired place, thanks to indexing and Series

In [39]:

```
val=pd.Series([5,6,7],index=[0,2,4])
```

In [40]:

```
df3['new']=val  #assigns the val to new col
```

In [41]:

```
df3 #let's see what happens to new col
```

Out[41]:

	title	star_rating	genre	duration	actors_list	new
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...	5.0
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']	NaN
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...	6.0
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...	NaN
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....	7.0
...	...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...	NaN
975	Back to the Future Part III	7.4	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...	NaN
976	Master and Commander: The Far Side of the World	7.4	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...	NaN
977	Poltergeist	7.4	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...	NaN
978	Wall Street	7.4	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...	NaN

979 rows × 6 columns

All the values have changed to NaN only 3 survived at index 0,2,4 those we saved and filled with 5,6,7. See How you manipulate the data. This is easy now go and manipulate your gf. You'll probably get caught same happens with data too :-P

9) Deleting a col from DataFrame

Enough playing with col 'new'. Let's remove it. Same thing my ex told me :-P

In [42]:

```
del df3['new']  #deletes 'new' col
```

In [43]:

```
df3
```

Out[43]:

	title	star_rating	genre	duration	actors_list
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....
...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	Back to the Future Part III	7.4	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	Master and Commander: The Far Side of the World	7.4	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	Poltergeist	7.4	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	Wall Street	7.4	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

979 rows × 5 columns

See col 'new' is no more. That was a cute column though. Boys always remove cute things ha ha ha. Girls complain about that.

10) Transpose of data

Rows will become columns and columns will become rows. Kind of tiktok - where boys playing girls and girls trying to be boys

In [44]:

```
df3.T
```

Out[44]:

	0	1	2	3	4	5	6	
title	The Shawshank Redemption	The Godfather	The Godfather: Part II	The Dark Knight	Pulp Fiction	12 Angry Men	The Good, the Bad and the Ugly	The Lo the R The Re of the
star_rating	9.3	9.2	9.1	9	8.9	8.9	8.9	
genre	Crime	Crime	Crime	Action	Crime	Drama	Western	Adve
duration	142	175	200	152	154	96	161	
actors_list	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...	[u'Marlon Brando', u'Al Pacino', u'James Caan']	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...	[u'Christian Bale', u'Heath Ledger', u'Aaron E...	[u'John Travolta', u'Uma Thurman', u'Samuel L....	[u'Henry Fonda', u'Lee J. Cobb', u'Martin Bals...	[u'Clint Eastwood', u'Eli Wallach', u'Lee Van ...	[u'E W u'V Morten u'Ian M

5 rows × 979 columns

See eveything is upside down. Kind of position 69 for DataFrame.

11) dropping something

In [45]:

```
df3 #what's df3
```

Out[45]:

	title	star_rating	genre	duration	actors_list
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....
...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	Back to the Future Part III	7.4	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	Master and Commander: The Far Side of the World	7.4	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	Poltergeist	7.4	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	Wall Street	7.4	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

979 rows × 5 columns

In [46]:

```
df3.drop('genre',axis='columns')  #drops genre from DataFrame
```

Out[46]:

	title	star_rating	duration	actors_list
0	The Shawshank Redemption	9.3	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	The Godfather	9.2	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	The Godfather: Part II	9.1	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	The Dark Knight	9.0	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	Pulp Fiction	8.9	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....
...	...	...	...	...
974	Tootsie	7.4	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	Back to the Future Part III	7.4	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	Master and Commander: The Far Side of the World	7.4	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	Poltergeist	7.4	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	Wall Street	7.4	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

979 rows × 4 columns

In [47]:

```
df3 #prints df3
```

Out[47]:

	title	star_rating	genre	duration	actors_list
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
1	The Godfather	9.2	Crime	175	[u'Marlon Brando', u'Al Pacino', u'James Caan']
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....
...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	Back to the Future Part III	7.4	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	Master and Commander: The Far Side of the World	7.4	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	Poltergeist	7.4	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	Wall Street	7.4	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

979 rows × 5 columns

see in df3 genre is back again.

##AkshanshTips - drop works temporarily. It does not effect original DataFrame. So when you want to see dataframe with some col, use drop

In [48]:

```
df3.drop(['genre','duration','actors_list'],axis='columns')   #works this way too
```

Out[48]:

```
                                            title  star_rating
  0                      The Shawshank Redemption          9.3
  1                                 The Godfather          9.2
  2                        The Godfather: Part II          9.1
  3                               The Dark Knight          9.0
  4                                  Pulp Fiction          8.9
...                                           ...          ...
974                                       Tootsie          7.4
975                      Back to the Future Part III          7.4
976   Master and Commander: The Far Side of the World          7.4
977                                    Poltergeist          7.4
978                                   Wall Street          7.4
```

979 rows × 2 columns

what if you want to drop some row?

In [49]:

```
df3.drop(1,axis='rows')  #row2 (that means index 1) is no more
```

Out[49]:

	title	star_rating	genre	duration	actors_list
0	The Shawshank Redemption	9.3	Crime	142	[u'Tim Robbins', u'Morgan Freeman', u'Bob Gunt...
2	The Godfather: Part II	9.1	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...
3	The Dark Knight	9.0	Action	152	[u'Christian Bale', u'Heath Ledger', u'Aaron E...
4	Pulp Fiction	8.9	Crime	154	[u'John Travolta', u'Uma Thurman', u'Samuel L....
5	12 Angry Men	8.9	Drama	96	[u'Henry Fonda', u'Lee J. Cobb', u'Martin Bals...
...	...	...	...	...	...
974	Tootsie	7.4	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	Back to the Future Part III	7.4	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	Master and Commander: The Far Side of the World	7.4	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	Poltergeist	7.4	Horror	114	[u'JoBeth Williams', u"Heather O'Rourke", u'Cr...
978	Wall Street	7.4	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

978 rows × 5 columns

##AkshanshTips - instead of using axis='rows' or axis='columns' you can use axis=0 for rows, and axis=1 for cols. But, I, personally like using 'rows' and 'columns'

##AkshanshTips - if you want drop to be permanent instead of temporarily, use inplace=True like this

In [50]:

```
df3.drop('actors_list',axis='columns',inplace=True)
```

In [51]:

```
df3#let's check if actors_list is there or not
```

Out[51]:

	title	star_rating	genre	duration
0	The Shawshank Redemption	9.3	Crime	142
1	The Godfather	9.2	Crime	175
2	The Godfather: Part II	9.1	Crime	200
3	The Dark Knight	9.0	Action	152
4	Pulp Fiction	8.9	Crime	154
...	...	...	...	...
974	Tootsie	7.4	Comedy	116
975	Back to the Future Part III	7.4	Adventure	118
976	Master and Commander: The Far Side of the World	7.4	Action	138
977	Poltergeist	7.4	Horror	114
978	Wall Street	7.4	Crime	126

979 rows × 4 columns

see actors_list is no more. I miss those funeral guys who are trending now a days from Ghana. Black suits and coffin, I can even hear the music now :-P

12) filtering

let's check which movies has star_rating more than 9.0

In [52]:

```
df3.star_rating>9.0
```

Out[52]:

```
0       True
1       True
2       True
3      False
4      False
       ...
974    False
975    False
976    False
977    False
978    False
Name: star_rating, Length: 979, dtype: bool
```

that's not what we were looking for

In [53]:

```
df3[df3.star_rating>9.0]  #now we got it
```

Out[53]:

	title	star_rating	genre	duration
0	The Shawshank Redemption	9.3	Crime	142
1	The Godfather	9.2	Crime	175
2	The Godfather: Part II	9.1	Crime	200

In [54]:

```
df3[df3.star_rating>7.5]  #similiarly we can check movies more that 7.5 rating
```

Out[54]:

	title	star_rating	genre	duration
0	The Shawshank Redemption	9.3	Crime	142
1	The Godfather	9.2	Crime	175
2	The Godfather: Part II	9.1	Crime	200
3	The Dark Knight	9.0	Action	152
4	Pulp Fiction	8.9	Crime	154
...	...	...	...	...
817	Saving Mr. Banks	7.6	Biography	125
818	The Hurricane	7.6	Biography	146
819	Jacob's Ladder	7.6	Drama	113
820	The Curse of the Were-Rabbit	7.6	Animation	85
821	Christmas Vacation	7.6	Comedy	97

822 rows × 4 columns

F..... there are lot of good movies more than 7.5 ratings. 822 movies..... this will make my quarantine days more happier

13) selection from iloc()

```
What's the main difference between loc and iloc. It is same as thanks vs
thanks alot. loc works woth lables, and iloc works with integer of index.
That's for what i stands in iloc.
DataFrame.iloc[rows_slicing,cols_slicing]
```

In [55]:

```
df3 #see what df3 has
```

Out[55]:

```
                                          title  star_rating      genre  duration
  0                    The Shawshank Redemption          9.3      Crime       142
  1                               The Godfather          9.2      Crime       175
  2                      The Godfather: Part II          9.1      Crime       200
  3                             The Dark Knight          9.0     Action       152
  4                                Pulp Fiction          8.9      Crime       154
...                                         ...          ...        ...       ...
974                                     Tootsie          7.4     Comedy       116
975                   Back to the Future Part III          7.4  Adventure       118
976  Master and Commander: The Far Side of the World    7.4     Action       138
977                                  Poltergeist          7.4     Horror       114
978                                  Wall Street          7.4      Crime       126
```

979 rows × 4 columns

In [56]:

```
df3.iloc[0:5,0:2]
```

Out[56]:

```
                     title  star_rating
0  The Shawshank Redemption          9.3
1             The Godfather          9.2
2    The Godfather: Part II          9.1
3           The Dark Knight          9.0
4              Pulp Fiction          8.9
```

see above data frame has rows indexed from 0 to 5, and col 0 to 1(2-1)

14) applying functions.

What I am gonna do. I will create a function that will tell us if rating of a movie is greater than 9 then it is excellent, if greater than 8.7 then it is good, if greater than 8.3 then one time watch. Below than eveything is bad.

Let's create a function

In [57]:

```
def my_func(x):
    if x>=9.0:
        return 'Excellent'
    elif x>=8.7:
        return 'Good'
    elif x>=8.3:
        return 'One Time Watch'
    else:
        return 'Bad'

```

let's make a new column in our beloved df3, named Verdict and assigns it's value to pending

In [58]:

```
df3['verdict']='pending'
```

In [59]:

```
df3.head(3)
```

Out[59]:

	title	star_rating	genre	duration	verdict
0	The Shawshank Redemption	9.3	Crime	142	pending
1	The Godfather	9.2	Crime	175	pending
2	The Godfather: Part II	9.1	Crime	200	pending

caolumn Verdict has been introduced.

I want my_func to apply on star_rating

In [60]:

```
x=df3['star_rating'].apply(my_func)
```

now I want my result on col Verdict

In [61]:

```
df3['verdict']=x
```

In [62]:

```
df3.head(50)  #let's see first 50 rows
```

Out[62]:

	title	star_rating	genre	duration	verdict
0	The Shawshank Redemption	9.3	Crime	142	Excellent
1	The Godfather	9.2	Crime	175	Excellent
2	The Godfather: Part II	9.1	Crime	200	Excellent
3	The Dark Knight	9.0	Action	152	Excellent
4	Pulp Fiction	8.9	Crime	154	Good
5	12 Angry Men	8.9	Drama	96	Good
6	The Good, the Bad and the Ugly	8.9	Western	161	Good
7	The Lord of the Rings: The Return of the King	8.9	Adventure	201	Good
8	Schindler's List	8.9	Biography	195	Good
9	Fight Club	8.9	Drama	139	Good
10	The Lord of the Rings: The Fellowship of the Ring	8.8	Adventure	178	Good
11	Inception	8.8	Action	148	Good
12	Star Wars: Episode V - The Empire Strikes Back	8.8	Action	124	Good
13	Forrest Gump	8.8	Drama	142	Good
14	The Lord of the Rings: The Two Towers	8.8	Adventure	179	Good
15	Interstellar	8.7	Adventure	169	Good
16	One Flew Over the Cuckoo's Nest	8.7	Drama	133	Good
17	Seven Samurai	8.7	Drama	207	Good
18	Goodfellas	8.7	Biography	146	Good
19	Star Wars	8.7	Action	121	Good
20	The Matrix	8.7	Action	136	Good
21	City of God	8.7	Crime	130	Good
22	It's a Wonderful Life	8.7	Drama	130	Good
23	The Usual Suspects	8.7	Crime	106	Good
24	Se7en	8.7	Drama	127	Good
25	Life Is Beautiful	8.6	Comedy	116	One Time Watch
26	Once Upon a Time in the West	8.6	Western	175	One Time Watch
27	The Silence of the Lambs	8.6	Drama	118	One Time Watch
28	Leon: The Professional	8.6	Crime	110	One Time Watch
29	City Lights	8.6	Comedy	87	One Time Watch

	title	star_rating	genre	duration	verdict
30	Spirited Away	8.6	Animation	125	One Time Watch
31	The Intouchables	8.6	Biography	112	One Time Watch
32	Casablanca	8.6	Drama	102	One Time Watch
33	Whiplash	8.6	Drama	107	One Time Watch
34	American History X	8.6	Crime	119	One Time Watch
35	Modern Times	8.6	Comedy	87	One Time Watch
36	Saving Private Ryan	8.6	Action	169	One Time Watch
37	Raiders of the Lost Ark	8.6	Action	115	One Time Watch
38	Rear Window	8.6	Mystery	112	One Time Watch
39	Psycho	8.6	Horror	109	One Time Watch
40	The Green Mile	8.5	Crime	189	One Time Watch
41	Sunset Blvd.	8.5	Drama	110	One Time Watch
42	The Pianist	8.5	Biography	150	One Time Watch
43	The Dark Knight Rises	8.5	Action	165	One Time Watch
44	Gladiator	8.5	Action	155	One Time Watch
45	Terminator 2: Judgment Day	8.5	Action	137	One Time Watch
46	Memento	8.5	Mystery	113	One Time Watch
47	Taare Zameen Par	8.5	Drama	165	One Time Watch
48	Dr. Strangelove or: How I Learned to Stop Worr...	8.5	Comedy	95	One Time Watch
49	The Departed	8.5	Crime	151	One Time Watch

Function has been succeddully applied. Go watch it once again how it worked.

15)Sorting

sorting arranges your data in alphabatical order.

In [63]:

```
df3.head(3)
```

Out[63]:

	title	star_rating	genre	duration	verdict
0	The Shawshank Redemption	9.3	Crime	142	Excellent
1	The Godfather	9.2	Crime	175	Excellent
2	The Godfather: Part II	9.1	Crime	200	Excellent

you see row index is sorted started from 0. But cols are not alphabatically sorted. Lets sort this.

In [64]:

```
df3.sort_index(axis='columns')
```

Out[64]:

	duration	genre	star_rating	title	verdict
0	142	Crime	9.3	The Shawshank Redemption	Excellent
1	175	Crime	9.2	The Godfather	Excellent
2	200	Crime	9.1	The Godfather: Part II	Excellent
3	152	Action	9.0	The Dark Knight	Excellent
4	154	Crime	8.9	Pulp Fiction	Good
...	...	...	...	...	...
974	116	Comedy	7.4	Tootsie	Bad
975	118	Adventure	7.4	Back to the Future Part III	Bad
976	138	Action	7.4	Master and Commander: The Far Side of the World	Bad
977	114	Horror	7.4	Poltergeist	Bad
978	126	Crime	7.4	Wall Street	Bad

979 rows × 5 columns

see that cols has alphabatically sorted now.

if you want to sort the values of col

let's sort the values of duration

```
In [65]:
df3.sort_values('duration')
Out[65]:
```

	title	star_rating	genre	duration	verdict
389	Freaks	8.0	Drama	64	Bad
338	Battleship Potemkin	8.0	History	66	Bad
258	The Cabinet of Dr. Caligari	8.1	Crime	67	Bad
293	Duck Soup	8.1	Comedy	68	Bad
88	The Kid	8.4	Comedy	68	One Time Watch
...	...	...	...	...	...
445	The Ten Commandments	7.9	Adventure	220	Bad
142	Lagaan: Once Upon a Time in India	8.3	Adventure	224	One Time Watch
78	Once Upon a Time in America	8.4	Crime	229	One Time Watch
157	Gone with the Wind	8.2	Drama	238	Bad
476	Hamlet	7.8	Drama	242	Bad

979 rows × 5 columns

##AkshanshTips- See duration now, values are in increasing order. If you pass more than one argument, priority will be given to first argument.

16) computing

```
In [67]:
df3.duration.sum()  #calculates the sum of the col duration
Out[67]:
118439
```

```
In [68]:
df3.star_rating.sum()  #you know what it does
Out[68]:
7724.1
```

In [69]:

```
df3.duration.mean()  #finds mean value of duration col
```

Out[69]:

```
120.97957099080695
```

In [70]:

```
df3.star_rating.mean()
```

Out[70]:

```
7.889785495403474
```

##AkshanshTips - why do all the normal computing when you can just - Describe it-

In [71]:

```
df3.describe()
```

Out[71]:

	star_rating	duration
count	979.000000	979.000000
mean	7.889785	120.979571
std	0.336069	26.218010
min	7.400000	64.000000
25%	7.600000	102.000000
50%	7.800000	117.000000
75%	8.100000	134.000000
max	9.300000	242.000000

See that describe provides info of two cols, because describe works on numerical data only.

That's it for today, we will be working on more stuff regarding the Pandas and how it works. This was Akshansh and my time has up now. Keep pythoning. Stay safe, stay at home - more importantly stay smarter than others. Have a great day all.

In [ ]:

```

```