lightening_talk_PANDAS_tong

April 28, 2018

Pandas

What is Pandas

A Python library providing data structures and data analysis tools. In particular, it offers data structures and operations for manipulating numerical tables and time series. The name comes from "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

Benefits

- Efficient storage and processing of data.
- Includes many built in functions for data transformation, aggregations, and plotting.
- Great for exploratory work.

Pandas is built on Numpy

Numpy is one of the fundamental packages for scientific computing in Python.

- They are like lists in Python however they allow faster computation
  - They are stored as one contiguous block of memory, rather than being spread out across multiple locations like a list.
  - Each item in a numpy array is of the same data type (i.e. all integers, all floats, etc.), rather than a conglomerate of any number of data types (as a list is).

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        # By convention
        from numpy.random import randn
        #return samples from the standard normal distribution
```

```
In [2]: np.random.randn(3,4)
```

```
Out[2]: array([[ 0.15705567, -0.53229413, -0.56556602, -0.90363394],
               [ 0.02396642, -0.61114129,  0.85944851,  0.75094198],
               [ 0.48659273, -0.17833593,  1.01057687,  0.00202565]])
```

Pandas Series

Series is a one-dimensional labeled array capable of holding any data type (integers, strings, floating point numbers, Python objects, etc.).

```
In [3]: dt_index = pd.date_range('2015-1-1',
                                 '2015-11-1',
                                 freq='m')
        dt_series = pd.Series(randn(10),
                              index = dt_index)
        dt_series
```

```
Out[3]: 2015-01-31   -0.610741
        2015-02-28    2.135324
        2015-03-31   -0.724126
        2015-04-30   -0.056732
        2015-05-31   -0.163344
        2015-06-30   -1.809111
        2015-07-31    0.748215
        2015-08-31   -0.510318
        2015-09-30    1.336520
        2015-10-31   -0.756852
        Freq: M, dtype: float64
```

```
In [4]: type(dt_series)
```

```
Out[4]: pandas.core.series.Series
```

```
In [5]: dt_series.mean()
```

```
Out[5]: -0.04111656845158672
```

Pandas DataFrames
A set of Pandas Series that share the same index

```
In [6]: df = pd.DataFrame(randn(10, 5), index=dt_index, columns=[x for x in 'abcde'])
        df
```

```
Out[6]:                    a         b         c         d         e
        2015-01-31 -0.606261 -0.111042 -1.097014  0.225621  1.316132
        2015-02-28  0.159944  1.265508 -0.689348  0.799481 -1.003818
        2015-03-31 -0.641932  1.475988  1.616184  0.960701 -1.162895
        2015-04-30 -0.422092  0.268672  0.315747 -0.311609  1.079811
        2015-05-31  0.762332  0.529420  2.218527 -0.707606  1.466735
        2015-06-30  0.413518 -0.802546 -0.451326 -1.814800 -0.821017
        2015-07-31 -1.038346  1.111153  0.356954 -1.678583  0.729438
        2015-08-31  0.787961 -1.418694 -0.853641 -0.985456 -0.544357
        2015-09-30 -1.915163  0.241606 -0.779002  0.642172  1.250826
        2015-10-31  4.043589 -0.012271  0.798781  1.324500 -0.178649
```

```
In [7]: type(df)
```

```
Out[7]: pandas.core.frame.DataFrame
```

```
In [8]: df['a']
        #Select a column
```

```
Out[8]: 2015-01-31   -0.606261
        2015-02-28    0.159944
        2015-03-31   -0.641932
        2015-04-30   -0.422092
        2015-05-31    0.762332
        2015-06-30    0.413518
        2015-07-31   -1.038346
        2015-08-31    0.787961
        2015-09-30   -1.915163
        2015-10-31    4.043589
        Freq: M, Name: a, dtype: float64
```

```
In [9]: df.a
        #This may fail when your column has the same name as a
        # dataframe method
```

```
Out[9]: 2015-01-31   -0.606261
        2015-02-28    0.159944
        2015-03-31   -0.641932
        2015-04-30   -0.422092
        2015-05-31    0.762332
        2015-06-30    0.413518
        2015-07-31   -1.038346
        2015-08-31    0.787961
        2015-09-30   -1.915163
        2015-10-31    4.043589
        Freq: M, Name: a, dtype: float64
```

```
In [10]: df.loc['2015-10-31']
         #select one row
```

```
Out[10]: a    4.043589
         b   -0.012271
         c    0.798781
         d    1.324500
         e   -0.178649
         Name: 2015-10-31 00:00:00, dtype: float64
```

```
In [11]: col = df['a']
         col.index
```

```
Out[11]: DatetimeIndex(['2015-01-31', '2015-02-28', '2015-03-31', '2015-04-30',
                        '2015-05-31', '2015-06-30', '2015-07-31', '2015-08-31',
                        '2015-09-30', '2015-10-31'],
                       dtype='datetime64[ns]', freq='M')
```

```
In [12]: df[['a','b']]
         #Select multiple columns
         #Interior brackets are for list,
         #outside brackets are indexing operator.
         #Must use double brackets if to select tow or more columns.
```

```
Out[12]:                    a         b
         2015-01-31 -0.606261 -0.111042
         2015-02-28  0.159944  1.265508
         2015-03-31 -0.641932  1.475988
         2015-04-30 -0.422092  0.268672
         2015-05-31  0.762332  0.529420
         2015-06-30  0.413518 -0.802546
         2015-07-31 -1.038346  1.111153
         2015-08-31  0.787961 -1.418694
         2015-09-30 -1.915163  0.241606
         2015-10-31  4.043589 -0.012271
```

```
In [13]: df['a'] + df['b']
         #Do math within columns
```

```
Out[13]: 2015-01-31   -0.717303
         2015-02-28    1.425452
         2015-03-31    0.834056
         2015-04-30   -0.153420
         2015-05-31    1.291752
         2015-06-30   -0.389029
         2015-07-31    0.072807
         2015-08-31   -0.630733
         2015-09-30   -1.673558
         2015-10-31    4.031319
         Freq: M, dtype: float64
```

```
In [14]: df['h'] = df['a'] + df['b']
         #Adding a new column
```

```
In [15]: df.head()
```

```
Out[15]:                    a         b         c         d         e         h
         2015-01-31 -0.606261 -0.111042 -1.097014  0.225621  1.316132 -0.717303
         2015-02-28  0.159944  1.265508 -0.689348  0.799481 -1.003818  1.425452
         2015-03-31 -0.641932  1.475988  1.616184  0.960701 -1.162895  0.834056
         2015-04-30 -0.422092  0.268672  0.315747 -0.311609  1.079811 -0.153420
         2015-05-31  0.762332  0.529420  2.218527 -0.707606  1.466735  1.291752
```

```
In [16]: df.loc['2015-05-31':'2015-08-31', 'c':'e']
         #Select by index and column label
```

```
Out[16]:                    c         d         e
         2015-05-31  2.218527 -0.707606  1.466735
         2015-06-30 -0.451326 -1.814800 -0.821017
         2015-07-31  0.356954 -1.678583  0.729438
         2015-08-31 -0.853641 -0.985456 -0.544357
```

```
In [17]: df.iloc[2:4,2:5]
         #Slect by slicing
```

```
Out[17]:                    c         d         e
         2015-03-31  1.616184  0.960701 -1.162895
         2015-04-30  0.315747 -0.311609  1.079811
```

How Pandas works with imported data

```
In [18]: df = pd.read_csv('IMDB-Movie-Data.csv')
         #https://www.kaggle.com/PromptCloudHQ/imdb-data
         df.head()
```

```
Out[18]:    Rank                    Title                     Genre  \
         0     1  Guardians of the Galaxy   Action,Adventure,Sci-Fi   
         1     2               Prometheus  Adventure,Mystery,Sci-Fi   
         2     3                    Split           Horror,Thriller   
         3     4                     Sing   Animation,Comedy,Family   
         4     5            Suicide Squad  Action,Adventure,Fantasy   

                                                  Description             Director  \
         0  A group of intergalactic criminals are forced ...           James Gunn   
         1  Following clues to the origin of mankind, a te...         Ridley Scott   
         2  Three girls are kidnapped by a man with a diag...   M. Night Shyamalan   
         3  In a city of humanoid animals, a hustling thea...  Christophe Lourdelet   
         4  A secret government agency recruits some of th...           David Ayer   

                                                       Actors  Year  Runtime (Minutes)  \
         0  Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...  2014                121   
         1  Noomi Rapace, Logan Marshall-Green, Michael Fa...  2012                124   
         2  James McAvoy, Anya Taylor-Joy, Haley Lu Richar...  2016                117   
         3  Matthew McConaughey,Reese Witherspoon, Seth Ma...  2016                108   
         4  Will Smith, Jared Leto, Margot Robbie, Viola D...  2016                123   

            Rating   Votes  Revenue (Millions)  Metascore  
         0     8.1  757074              333.13       76.0  
         1     7.0  485820              126.46       65.0  
         2     7.3  157606              138.12       62.0  
         3     7.2   60545              270.32       59.0  
         4     6.2  393727              325.02       40.0  
```

```
In [19]: df.shape
```

```
Out[19]: (1000, 12)
```

```
In [20]: df.columns
```

```
Out[20]: Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
                'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
                'Metascore'],
               dtype='object')
```

```
In [21]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
Rank                    1000 non-null int64
Title                   1000 non-null object
Genre                   1000 non-null object
Description             1000 non-null object
Director                1000 non-null object
Actors                  1000 non-null object
Year                    1000 non-null int64
Runtime (Minutes)       1000 non-null int64
Rating                  1000 non-null float64
Votes                   1000 non-null int64
Revenue (Millions)      872 non-null float64
Metascore               936 non-null float64
dtypes: float64(3), int64(4), object(5)
memory usage: 93.8+ KB
```

```
In [22]: df['Rating']
```

```
Out[22]: 0      8.1
         1      7.0
         2      7.3
         3      7.2
         4      6.2
         5      6.1
         6      8.3
         7      6.4
         8      7.1
         9      7.0
         10     7.5
         11     7.8
         12     7.9
         13     7.7
         14     6.4
         15     6.6
         16     8.2
         17     6.7
         18     8.1
         19     8.0
         20     6.7
         21     7.9
         22     6.7
         23     6.5
         24     5.3
         25     6.8
         26     8.3
         27     4.7
         28     6.2
         29     5.9
               ...
         970    4.8
         971    6.9
         972    5.9
         973    6.2
         974    7.4
         975    6.0
         976    6.2
         977    5.0
         978    7.0
         979    7.6
         980    7.0
         981    5.3
         982    7.4
         983    6.5
         984    6.8
         985    5.6
         986    5.9
         987    6.3
         988    7.1
         989    7.5
         990    6.6
         991    8.5
         992    6.3
         993    5.9
         994    6.7
         995    6.2
         996    5.5
         997    6.2
         998    5.6
         999    5.3
         Name: Rating, Length: 1000, dtype: float64
```

```
In [23]: df['Rating'] <= 7
```

```
Out[23]: 0      False
         1       True
         2      False
         3      False
         4       True
         5       True
         6      False
         7       True
         8      False
         9       True
```

```
10     False
11     False
12     False
13     False
14      True
15      True
16     False
17      True
18     False
19     False
20      True
21     False
22      True
23      True
24      True
25      True
26     False
27      True
28      True
29      True
       ...
970     True
971     True
972     True
973     True
974    False
975     True
976     True
977     True
978     True
979    False
980     True
981     True
982    False
983     True
984     True
985     True
986     True
987     True
988    False
989    False
990     True
991    False
992     True
993     True
994     True
995     True
996     True
```

```
         997     True
         998     True
         999     True
         Name: Rating, Length: 1000, dtype: bool

In [24]: mask = df['Rating'] <= 7

In [25]: df[mask].head()
         #Use boolean series to mask a datafram
         #returning only those rows where the mask is True

Out[25]:    Rank            Title                       Genre  \
         1     2      Prometheus  Adventure,Mystery,Sci-Fi
         4     5   Suicide Squad  Action,Adventure,Fantasy
         5     6  The Great Wall  Action,Adventure,Fantasy
         7     8         Mindhorn                    Comedy
         9    10       Passengers   Adventure,Drama,Romance

                                                  Description       Director  \
         1  Following clues to the origin of mankind, a te...   Ridley Scott
         4  A secret government agency recruits some of th...     David Ayer
         5  European mercenaries searching for black powde...    Yimou Zhang
         7  A has-been actor best known for playing the ti...     Sean Foley
         9  A spacecraft traveling to a distant colony pla...  Morten Tyldum

                                                       Actors  Year  Runtime (Minutes)  \
         1  Noomi Rapace, Logan Marshall-Green, Michael Fa...  2012                124
         4  Will Smith, Jared Leto, Margot Robbie, Viola D...  2016                123
         5      Matt Damon, Tian Jing, Willem Dafoe, Andy Lau  2016                103
         7  Essie Davis, Andrea Riseborough, Julian Barrat...  2016                 89
         9  Jennifer Lawrence, Chris Pratt, Michael Sheen,...  2016                116

            Rating   Votes  Revenue (Millions)  Metascore
         1     7.0  485820              126.46       65.0
         4     6.2  393727              325.02       40.0
         5     6.1   56036               45.13       42.0
         7     6.4    2490                 NaN       71.0
         9     7.0  192177              100.01       41.0

In [26]: g = df.groupby('Year').count()['Title']
         g

Out[26]: Year
         2006     44
         2007     53
         2008     52
         2009     51
         2010     60
         2011     63
```

```
2012     64
2013     91
2014     98
2015    127
2016    297
Name: Title, dtype: int64
```

```
In [27]: df.drop('Votes', axis=1)
         #drop a column
```

```
Out[27]:
     Rank                                    Title  \
0       1                  Guardians of the Galaxy
1       2                               Prometheus
2       3                                    Split
3       4                                     Sing
4       5                            Suicide Squad
5       6                           The Great Wall
6       7                               La La Land
7       8                                 Mindhorn
8       9                       The Lost City of Z
9      10                               Passengers
10     11  Fantastic Beasts and Where to Find Them
11     12                           Hidden Figures
12     13                                Rogue One
13     14                                    Moana
14     15                                 Colossal
15     16                  The Secret Life of Pets
16     17                            Hacksaw Ridge
17     18                             Jason Bourne
18     19                                     Lion
19     20                                  Arrival
20     21                                     Gold
21     22                    Manchester by the Sea
22     23                           Hounds of Love
23     24                                   Trolls
24     25             Independence Day: Resurgence
25     26                          Paris pieds nus
26     27                  Bahubali: The Beginning
27     28                               Dead Awake
28     29                                 Bad Moms
29     30                         Assassin's Creed
..    ...                                      ...
970   971                        Texas Chainsaw 3D
971   972                                Disturbia
972   973                             Rock of Ages
973   974                                 Scream 4
974   975                           Queen of Katwe
975   976               My Big Fat Greek Wedding 2
```

```
976   977                        Dark Places
977   978                      Amateur Night
978   979        It's Only the End of the World
979   980                   The Skin I Live In
980   981                 Miracles from Heaven
981   982                               Annie
982   983                Across the Universe
983   984                      Let's Be Cops
984   985                                 Max
985   986                       Your Highness
986   987                Final Destination 5
987   988                        Endless Love
988   989                             Martyrs
989   990                               Selma
990   991        Underworld: Rise of the Lycans
991   992                    Taare Zameen Par
992   993                 Take Me Home Tonight
993   994             Resident Evil: Afterlife
994   995                           Project X
995   996                 Secret in Their Eyes
996   997                    Hostel: Part II
997   998               Step Up 2: The Streets
998   999                        Search Party
999  1000                          Nine Lives

                          Genre  \
0       Action,Adventure,Sci-Fi
1      Adventure,Mystery,Sci-Fi
2               Horror,Thriller
3       Animation,Comedy,Family
4      Action,Adventure,Fantasy
5      Action,Adventure,Fantasy
6            Comedy,Drama,Music
7                        Comedy
8    Action,Adventure,Biography
9       Adventure,Drama,Romance
10     Adventure,Family,Fantasy
11      Biography,Drama,History
12      Action,Adventure,Sci-Fi
13   Animation,Adventure,Comedy
14          Action,Comedy,Drama
15   Animation,Adventure,Comedy
16      Biography,Drama,History
17              Action,Thriller
18              Biography,Drama
19         Drama,Mystery,Sci-Fi
20     Adventure,Drama,Thriller
21                        Drama
```

```
22           Crime,Drama,Horror
23   Animation,Adventure,Comedy
24      Action,Adventure,Sci-Fi
25                       Comedy
26       Action,Adventure,Drama
27              Horror,Thriller
28                       Comedy
29       Action,Adventure,Drama
..                          ...
970             Horror,Thriller
971      Drama,Mystery,Thriller
972        Comedy,Drama,Musical
973              Horror,Mystery
974       Biography,Drama,Sport
975       Comedy,Family,Romance
976      Drama,Mystery,Thriller
977                      Comedy
978                       Drama
979              Drama,Thriller
980      Biography,Drama,Family
981         Comedy,Drama,Family
982       Drama,Fantasy,Musical
983                      Comedy
984            Adventure,Family
985    Adventure,Comedy,Fantasy
986             Horror,Thriller
987               Drama,Romance
988                      Horror
989     Biography,Drama,History
990    Action,Adventure,Fantasy
991          Drama,Family,Music
992        Comedy,Drama,Romance
993     Action,Adventure,Horror
994                      Comedy
995         Crime,Drama,Mystery
996                      Horror
997         Drama,Music,Romance
998            Adventure,Comedy
999       Comedy,Family,Fantasy

                                           Description                Director  \
0    A group of intergalactic criminals are forced ...              James Gunn
1    Following clues to the origin of mankind, a te...            Ridley Scott
2    Three girls are kidnapped by a man with a diag...      M. Night Shyamalan
3    In a city of humanoid animals, a hustling thea...    Christophe Lourdelet
4    A secret government agency recruits some of th...              David Ayer
5    European mercenaries searching for black powde...             Yimou Zhang
6    A jazz pianist falls for an aspiring actress i...         Damien Chazelle
```

```
7    A has-been actor best known for playing the ti...             Sean Foley
8    A true-life drama, centering on British explor...             James Gray
9    A spacecraft traveling to a distant colony pla...          Morten Tyldum
10   The adventures of writer Newt Scamander in New...            David Yates
11   The story of a team of female African-American...         Theodore Melfi
12   The Rebel Alliance makes a risky move to steal...         Gareth Edwards
13   In Ancient Polynesia, when a terrible curse in...           Ron Clements
14   Gloria is an out-of-work party girl forced to ...        Nacho Vigalondo
15   The quiet life of a terrier named Max is upend...           Chris Renaud
16   WWII American Army Medic Desmond T. Doss, who ...             Mel Gibson
17   The CIA's most dangerous former operative is d...        Paul Greengrass
18   A five-year-old Indian boy gets lost on the st...            Garth Davis
19   When twelve mysterious spacecraft appear aroun...       Denis Villeneuve
20   Kenny Wells, a prospector desperate for a luck...         Stephen Gaghan
21   A depressed uncle is asked to take care of his...       Kenneth Lonergan
22   A cold-blooded predatory couple while cruising...              Ben Young
23   After the Bergens invade Troll Village, Poppy,...             Walt Dohrn
24   Two decades after the first Independence Day i...        Roland Emmerich
25   Fiona visits Paris for the first time to assis...         Dominique Abel
26   In ancient India, an adventurous and daring ma...         S.S. Rajamouli
27   A young woman must save herself and her friend...         Phillip Guzman
28   When three overworked and under-appreciated mo...              Jon Lucas
29   When Callum Lynch explores the memories of his...          Justin Kurzel
..                                                 ...                    ...
970  A young woman travels to Texas to collect an i...        John Luessenhop
971  A teen living under house arrest becomes convi...            D.J. Caruso
972  A small town girl and a city boy meet on the S...          Adam Shankman
973  Ten years have passed, and Sidney Prescott, wh...             Wes Craven
974  A Ugandan girl sees her world rapidly change a...              Mira Nair
975  A Portokalos family secret brings the beloved ...             Kirk Jones
976  Libby Day was only eight years old when her fa...  Gilles Paquet-Brenner
977  Guy Carter is an award-winning graduate studen...           Lisa Addario
978  Louis (Gaspard Ulliel), a terminally ill write...           Xavier Dolan
979  A brilliant plastic surgeon, haunted by past t...        Pedro Almodóvar
980  A young girl suffering from a rare digestive d...        Patricia Riggen
981  A foster kid, who lives with her mean foster m...             Will Gluck
982  The music of the Beatles and the Vietnam War f...           Julie Taymor
983  Two struggling pals dress as police officers f...        Luke Greenfield
984  A Malinois dog that helped American Marines in...             Boaz Yakin
985  When Prince Fabious's bride is kidnapped, he g...     David Gordon Green
986  Survivors of a suspension-bridge collapse lear...           Steven Quale
987  The story of a privileged girl and a charismat...            Shana Feste
988  A young woman's quest for revenge against the ...         Pascal Laugier
989  A chronicle of Martin Luther King's campaign t...           Ava DuVernay
990  An origins story centered on the centuries-old...     Patrick Tatopoulos
991  An eight-year-old boy is thought to be a lazy ...             Aamir Khan
992  Four years after graduation, an awkward high s...          Michael Dowse
993  While still out to destroy the evil Umbrella C...     Paul W.S. Anderson
```

```
994  3 high school seniors throw a birthday party t...        Nima Nourizadeh
995  A tight-knit team of rising investigators, alo...              Billy Ray
996  Three American college students studying abroa...               Eli Roth
997  Romantic sparks occur between two dance studen...             Jon M. Chu
998  A pair of friends embark on a mission to reuni...         Scot Armstrong
999  A stuffy businessman finds himself trapped ins...       Barry Sonnenfeld

                                                Actors  Year  \
0    Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...  2014
1    Noomi Rapace, Logan Marshall-Green, Michael Fa...  2012
2    James McAvoy, Anya Taylor-Joy, Haley Lu Richar...  2016
3    Matthew McConaughey,Reese Witherspoon, Seth Ma...  2016
4    Will Smith, Jared Leto, Margot Robbie, Viola D...  2016
5        Matt Damon, Tian Jing, Willem Dafoe, Andy Lau  2016
6    Ryan Gosling, Emma Stone, Rosemarie DeWitt, J...  2016
7    Essie Davis, Andrea Riseborough, Julian Barrat...  2016
8    Charlie Hunnam, Robert Pattinson, Sienna Mille...  2016
9    Jennifer Lawrence, Chris Pratt, Michael Sheen,...  2016
10   Eddie Redmayne, Katherine Waterston, Alison Su...  2016
11   Taraji P. Henson, Octavia Spencer, Janelle Mon...  2016
12   Felicity Jones, Diego Luna, Alan Tudyk, Donnie...  2016
13   Auli'i Cravalho, Dwayne Johnson, Rachel House,...  2016
14   Anne Hathaway, Jason Sudeikis, Austin Stowell,...  2016
15   Louis C.K., Eric Stonestreet, Kevin Hart, Lake...  2016
16   Andrew Garfield, Sam Worthington, Luke Bracey,...  2016
17   Matt Damon, Tommy Lee Jones, Alicia Vikander,V...  2016
18   Dev Patel, Nicole Kidman, Rooney Mara, Sunny P...  2016
19   Amy Adams, Jeremy Renner, Forest Whitaker,Mich...  2016
20   Matthew McConaughey, Edgar Ramírez, Bryce Dall...  2016
21   Casey Affleck, Michelle Williams, Kyle Chandle...  2016
22   Emma Booth, Ashleigh Cummings, Stephen Curry,S...  2016
23   Anna Kendrick, Justin Timberlake,Zooey Deschan...  2016
24   Liam Hemsworth, Jeff Goldblum, Bill Pullman,Ma...  2016
25   Fiona Gordon, Dominique Abel,Emmanuelle Riva, ...  2016
26   Prabhas, Rana Daggubati, Anushka Shetty,Tamann...  2015
27   Jocelin Donahue, Jesse Bradford, Jesse Borrego...  2016
28   Mila Kunis, Kathryn Hahn, Kristen Bell,Christi...  2016
29   Michael Fassbender, Marion Cotillard, Jeremy I...  2016
..                                                 ...   ...
970  Alexandra Daddario, Tania Raymonde, Scott East...  2013
971  Shia LaBeouf, David Morse, Carrie-Anne Moss, S...  2007
972  Julianne Hough, Diego Boneta, Tom Cruise, Alec...  2012
973  Neve Campbell, Courteney Cox, David Arquette, ...  2011
974  Madina Nalwanga, David Oyelowo, Lupita Nyong'o...  2016
975  Nia Vardalos, John Corbett, Michael Constantin...  2016
976  Charlize Theron, Nicholas Hoult, Christina Hen...  2015
977  Jason Biggs, Janet Montgomery,Ashley Tisdale, ...  2016
978  Nathalie Baye, Vincent Cassel, Marion Cotillar...  2016
```

```
979  Antonio Banderas, Elena Anaya, Jan Cornet,Mari...  2011
980  Jennifer Garner, Kylie Rogers, Martin Henderso...  2016
981  Quvenzhané Wallis, Cameron Diaz, Jamie Foxx, R...  2014
982  Evan Rachel Wood, Jim Sturgess, Joe Anderson, ...  2007
983  Jake Johnson, Damon Wayans Jr., Rob Riggle, Ni...  2014
984  Thomas Haden Church, Josh Wiggins, Luke Kleint...  2015
985  Danny McBride, Natalie Portman, James Franco, ...  2011
986  Nicholas D'Agosto, Emma Bell, Arlen Escarpeta,...  2011
987  Gabriella Wilde, Alex Pettyfer, Bruce Greenwoo...  2014
988  Morjana Alaoui, Mylène Jampanoï, Catherine Bég...  2008
989  David Oyelowo, Carmen Ejogo, Tim Roth, Lorrain...  2014
990  Rhona Mitra, Michael Sheen, Bill Nighy, Steven...  2009
991  Darsheel Safary, Aamir Khan, Tanay Chheda, Sac...  2007
992  Topher Grace, Anna Faris, Dan Fogler, Teresa P...  2011
993  Milla Jovovich, Ali Larter, Wentworth Miller,K...  2010
994  Thomas Mann, Oliver Cooper, Jonathan Daniel Br...  2012
995  Chiwetel Ejiofor, Nicole Kidman, Julia Roberts...  2015
996  Lauren German, Heather Matarazzo, Bijou Philli...  2007
997  Robert Hoffman, Briana Evigan, Cassie Ventura,...  2008
998  Adam Pally, T.J. Miller, Thomas Middleditch,Sh...  2014
999  Kevin Spacey, Jennifer Garner, Robbie Amell,Ch...  2016

     Runtime (Minutes)  Rating  Revenue (Millions)  Metascore
0                  121     8.1              333.13       76.0
1                  124     7.0              126.46       65.0
2                  117     7.3              138.12       62.0
3                  108     7.2              270.32       59.0
4                  123     6.2              325.02       40.0
5                  103     6.1               45.13       42.0
6                  128     8.3              151.06       93.0
7                   89     6.4                 NaN       71.0
8                  141     7.1                8.01       78.0
9                  116     7.0              100.01       41.0
10                 133     7.5              234.02       66.0
11                 127     7.8              169.27       74.0
12                 133     7.9              532.17       65.0
13                 107     7.7              248.75       81.0
14                 109     6.4                2.87       70.0
15                  87     6.6              368.31       61.0
16                 139     8.2               67.12       71.0
17                 123     6.7              162.16       58.0
18                 118     8.1               51.69       69.0
19                 116     8.0              100.50       81.0
20                 120     6.7                7.22       49.0
21                 137     7.9               47.70       96.0
22                 108     6.7                 NaN       72.0
23                  92     6.5              153.69       56.0
24                 120     5.3              103.14       32.0
```

```
25                  83      6.8                 NaN         NaN
26                 159      8.3                6.50         NaN
27                  99      4.7                0.01         NaN
28                 100      6.2              113.08        60.0
29                 115      5.9               54.65        36.0
..                 ...      ...                 ...         ...
970                 92      4.8               34.33        62.0
971                105      6.9               80.05         NaN
972                123      5.9               38.51        47.0
973                111      6.2               38.18        52.0
974                124      7.4                8.81        73.0
975                 94      6.0               59.57        37.0
976                113      6.2                 NaN        39.0
977                 92      5.0                 NaN        38.0
978                 97      7.0                 NaN        48.0
979                120      7.6                3.19        70.0
980                109      7.0               61.69        44.0
981                118      5.3               85.91        33.0
982                133      7.4               24.34        56.0
983                104      6.5               82.39        30.0
984                111      6.8               42.65        47.0
985                102      5.6               21.56        31.0
986                 92      5.9               42.58        50.0
987                104      6.3               23.39        30.0
988                 99      7.1                 NaN        89.0
989                128      7.5               52.07         NaN
990                 92      6.6               45.80        44.0
991                165      8.5                1.20        42.0
992                 97      6.3                6.92         NaN
993                 97      5.9               60.13        37.0
994                 88      6.7               54.72        48.0
995                111      6.2                 NaN        45.0
996                 94      5.5               17.54        46.0
997                 98      6.2               58.01        50.0
998                 93      5.6                 NaN        22.0
999                 87      5.3               19.64        11.0

[1000 rows x 11 columns]
```

```
In [28]: df.columns
```

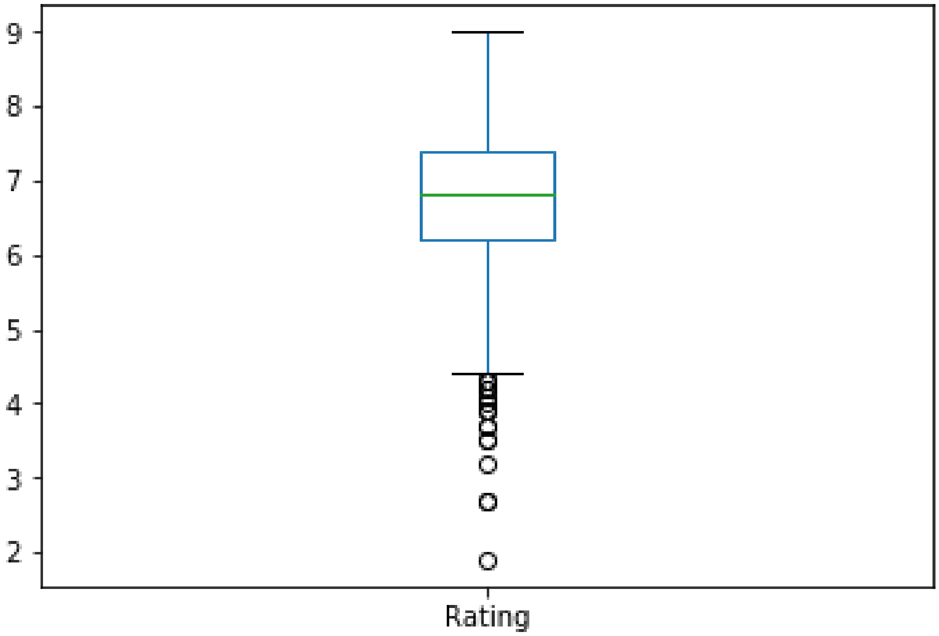
```
Out[28]: Index(['Rank', 'Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
                'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
                'Metascore'],
               dtype='object')
```

```
In [29]: df['Genre'].value_counts()
         #For categorical data, this gets you the frequencies
```

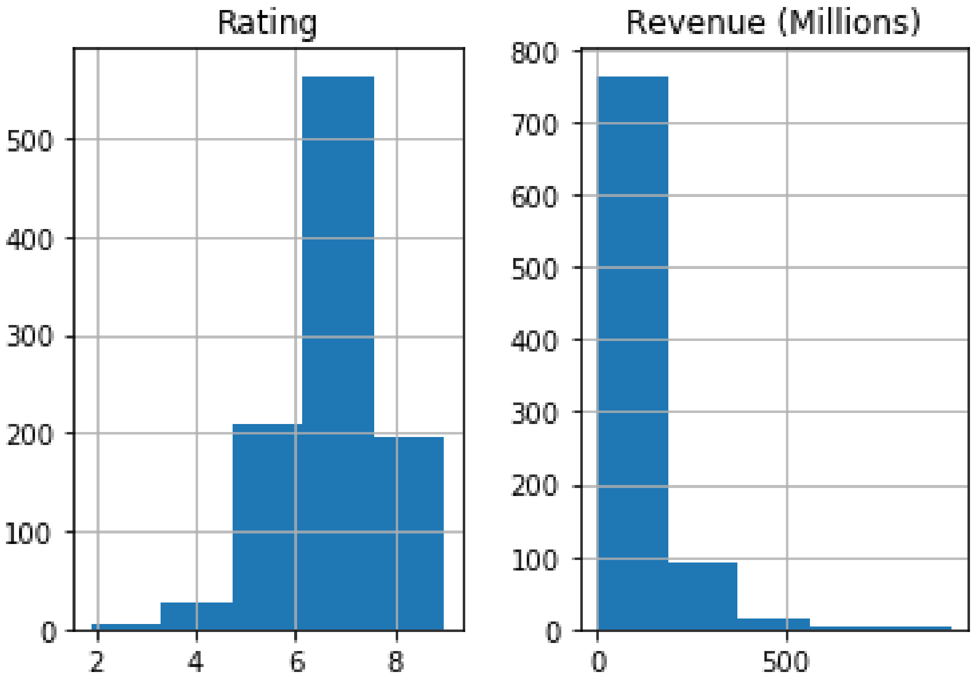
```
Out[29]: Action,Adventure,Sci-Fi       50
         Drama                         48
         Comedy,Drama,Romance          35
         Comedy                        32
         Drama,Romance                 31
         Animation,Adventure,Comedy    27
         Action,Adventure,Fantasy      27
         Comedy,Drama                  27
         Comedy,Romance                26
         Crime,Drama,Thriller          24
         Crime,Drama,Mystery           23
         Action,Adventure,Drama        18
         Action,Crime,Drama            17
         Horror,Thriller               16
         Drama,Thriller                15
         Adventure,Family,Fantasy      14
         Action,Adventure,Comedy       14
         Biography,Drama               14
         Biography,Drama,History       14
         Action,Crime,Thriller         12
         Action,Comedy,Crime           12
         Action,Adventure,Thriller     11
         Horror                        11
         Crime,Drama                   10
         Animation,Action,Adventure     9
         Biography,Crime,Drama          9
         Thriller                       9
         Horror,Mystery,Thriller        9
         Action,Thriller                9
         Drama,Mystery,Thriller         8
                                       ..
         Action,Fantasy,War             1
         Drama,Fantasy,Music            1
         Sci-Fi,Thriller                1
         Comedy,Western                 1
         Fantasy,Horror,Thriller        1
         Comedy,Family,Romance          1
         Biography,History,Thriller     1
         Comedy,Horror,Romance          1
         Crime,Thriller                 1
         Comedy,Drama,Thriller          1
         Action,Fantasy                 1
         Mystery,Romance,Thriller       1
         Comedy,Romance,Sport           1
         Drama,Fantasy,Thriller         1
         Drama,Fantasy,Mystery          1
         Biography,Drama,Mystery        1
         Fantasy,Mystery,Thriller       1
```

```
Drama,Romance,War              1
Thriller,War                   1
Adventure,Drama,War            1
Adventure,Fantasy,Mystery      1
Action,Comedy,Sci-Fi           1
Adventure,Comedy,Fantasy       1
Action,Horror,Romance          1
Romance,Sci-Fi                 1
Drama,Thriller,War             1
Animation,Comedy,Drama         1
Animation,Fantasy              1
Drama,Family                   1
Animation,Drama,Fantasy        1
Name: Genre, Length: 207, dtype: int64
```

```
In [37]: df[['Rating']].plot(kind='box');
         plt.show()
```



```
In [36]: df.hist(['Rating','Revenue (Millions)'],bins=5);
         plt.show()
```



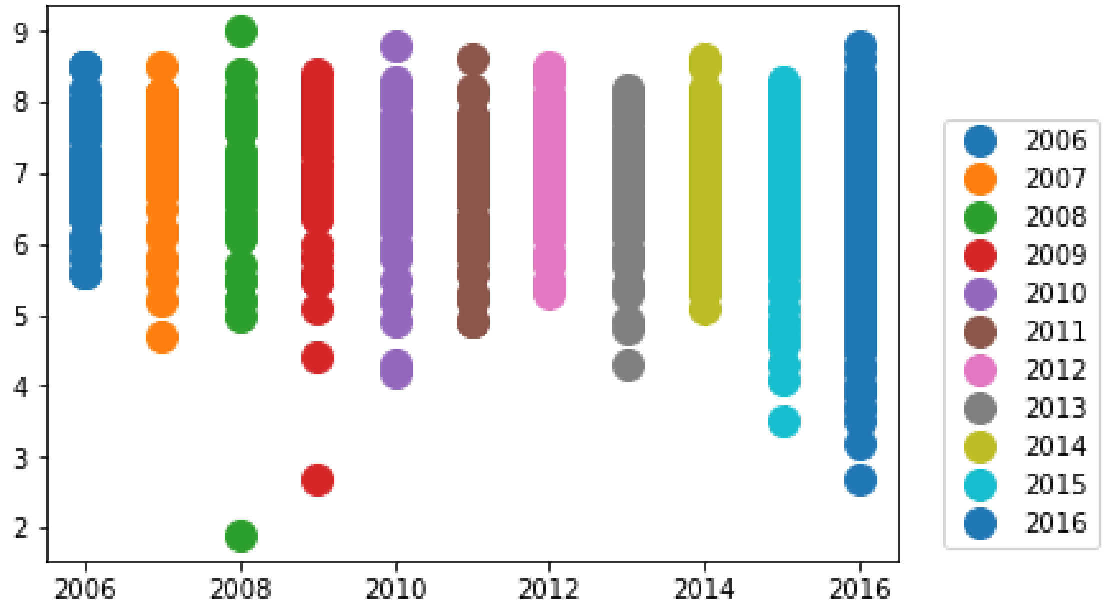
```
In [43]: groups=df.groupby('Year')
         for name, group in groups:
             print(name)
```

```
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
```

```
In [56]: fig, ax = plt.subplots()

         ax.margins(0.05)
         for name, group in groups:
             ax.plot(group.Year, group.Rating, marker='o', linestyle='', ms=12, label=name)
         ax.legend(numpoints=1, loc='lower right', bbox_to_anchor=(1.25, 0.0))

         plt.show()
```



Reference

- https://en.wikipedia.org/wiki/Pandas_(software)
- http://pandas.pydata.org/pandas-docs/stable/index.html
- http://pandas.pydata.org/pandas-docs/stable/comparison_with_sql.html