Custom Learnings

Day 15

Python Pandas

Pandas: Python library.

Dataframe: Holds the data in rows and columns form. Combination of more than one series(one column is called series).

In [1]:
```
data = {'orange':["kashmir","ooty","bglr"], 'apples':["chennai","delhi","Kodai"]}
```

In [2]:
```
import pandas as pd
```

In [3]:
```
print(data)
```

```
{'orange': ['kashmir', 'ooty', 'bglr'], 'apples': ['chennai', 'delhi', 'Kodai']}
```

In [4]:
```
fruit_df = pd.DataFrame(data)
```

In [5]:
```
fruit_df.head()
```

Out[5]:

	orange	apples
0	kashmir	chennai
1	ooty	delhi
2	bglr	Kodai

In [6]:
```
fruit_df = pd.DataFrame(data, index = ['jan', 'feb', 'mar'])
fruit_df.head()
```

Out[6]:

	orange	apples
jan	kashmir	chennai
feb	ooty	delhi
mar	bglr	Kodai

In [7]:
```
test = fruit_df.loc['feb']
test.head()
```

Out[7]:
```
orange     ooty
apples    delhi
Name: feb, dtype: object
```

In [7]:
```
test = fruit_df.loc['feb']
test.head()
```

Out[7]:
```
orange     ooty
apples    delhi
Name: feb, dtype: object
```

In [9]:
```
movieDf = pd.read_csv('/home/labuser/Documents/Pandas_datasets/IMDB-Movie-Data.csv', index_col = 0)
movieDf.head()
```

Out[9]:

Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13	76.0
2	Prometheus	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.46	65.0
3	Split	Horror,Thriller	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	7.3	157606	138.12	62.0
4	Sing	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	270.32	59.0
5	Suicide Squad	Action,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.02	40.0

In [10]:
```
movieDf.tail(10)
```

Out[10]:

Rank	Title	Genre	Description	Director	Actors	Year	Runtime (Minutes)	Rating	Votes	Revenue (Millions)	Metascore
991	Underworld: Rise of the Lycans	Action,Adventure,Fantasy	An origins story centered on the centuries-old...	Patrick Tatopoulos	Rhona Mitra, Michael Sheen, Bill Nighy, Steven...	2009	92	6.6	129708	45.80	44.0
992	Taare Zameen Par	Drama,Family,Music	An eight-year-old boy is thought to be a lazy ...	Aamir Khan	Darsheel Safary, Aamir Khan, Tanay Chheda, Sac...	2007	165	8.5	102697	1.20	42.0
993	Take Me Home Tonight	Comedy,Drama,Romance	Four years after graduation, an awkward high s...	Michael Dowse	Topher Grace, Anna Faris, Dan Fogler, Teresa P...	2011	97	6.3	45419	6.92	NaN
994	Resident Evil: Afterlife	Action,Adventure,Horror	While still out to destroy the evil Umbrella C...	Paul W.S. Anderson	Milla Jovovich, Ali Larter, Wentworth Miller,K...	2010	97	5.9	140900	60.13	37.0
995	Project X	Comedy	3 high school seniors throw a birthday party t...	Nima Nourizadeh	Thomas Mann, Oliver Cooper, Jonathan Daniel Br...	2012	88	6.7	164088	54.72	48.0
996	Secret in Their Eyes	Crime,Drama,Mystery	A tight-knit team of rising investigators, alo...	Billy Ray	Chiwetel Ejiofor, Nicole Kidman, Julia Roberts...	2015	111	6.2	27585	NaN	45.0
997	Hostel: Part II	Horror	Three American college students studying abroa...	Eli Roth	Lauren German, Heather Matarazzo, Bijou Philli...	2007	94	5.5	73152	17.54	46.0
998	Step Up 2: The Streets	Drama,Music,Romance	Romantic sparks occur between two dance studen...	Jon M. Chu	Robert Hoffman, Briana Evigan, Cassie Ventura,...	2008	98	6.2	70699	58.01	50.0
999	Search Party	Adventure,Comedy	A pair of friends embark on a mission to reuni...	Scot Armstrong	Adam Pally, T.J. Miller, Thomas Middleditch,Sh...	2014	93	5.6	4881	NaN	22.0
1000	Nine Lives	Comedy,Family,Fantasy	A stuffy businessman finds himself trapped ins...	Barry Sonnenfeld	Kevin Spacey, Jennifer Garner, Robbie Amell,Ch...	2016	87	5.3	12435	19.64	11.0

Drop_duplicates : Removes duplicate rows from the dataframe.

Shape: Shows the number of rows and columns of the dataframe.

Append(): Append two dataframes.

Info(): Gives the information about the dataframe just like describe command in sql.

Columns: Shows the column names of the dataframe.

Inplace Attribute: If we want to do the transformation in the same dataframe.

Rename(): Rename the column names of the dataframe.

By applying the list comprehension method we can convert the column names to upper/lower case:

```
In [11]: movieDf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 1 to 1000
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Title               1000 non-null   object 
 1   Genre               1000 non-null   object 
 2   Description         1000 non-null   object 
 3   Director            1000 non-null   object 
 4   Actors              1000 non-null   object 
 5   Year                1000 non-null   int64  
 6   Runtime (Minutes)   1000 non-null   int64  
 7   Rating              1000 non-null   float64
 8   Votes               1000 non-null   int64  
 9   Revenue (Millions)  872 non-null    float64
 10  Metascore           936 non-null    float64
dtypes: float64(3), int64(3), object(5)
memory usage: 93.8+ KB
```

```
In [12]: movieDf.shape
```

```
Out[12]: (1000, 11)
```

```
In [13]: tempDf = movieDf.append(movieDf)
         tempDf.shape
```

```
/tmp/ipykernel_2442/559746854.py:1: FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  tempDf = movieDf.append(movieDf)
```

```
Out[13]: (2000, 11)
```

```
In [14]: finalDf = tempDf.drop_duplicates()
         finalDf.shape
```

```
Out[14]: (1000, 11)
```

```
In [16]: finalDf.columns
```

```
Out[16]: Index(['Title', 'Genre', 'Description', 'Director', 'Actors', 'Year',
       'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)',
       'Metascore'],
      dtype='object')
```

```
In [19]: finalDf.rename(columns = {'Runtime (Minutes)': "Runtime", 'Revenue (Millions)': "Revenue_millions"}, inplace = True)
```

```
/tmp/ipykernel_2442/3374654768.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  finalDf.rename(columns = {'Runtime (Minutes)': "Runtime", 'Revenue (Millions)': "Revenue_millions"}, inplace = True)
```

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

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

```
In [21]: finalDf.columns = [i.lower() for i in finalDf.columns]
         finalDf.columns
```

```
Out[21]: Index(['title', 'genre', 'description', 'director', 'actors', 'year',
       'runtime', 'rating', 'votes', 'revenue_millions', 'metascore'],
      dtype='object')
```

```
In [22]: finalDf.columns = [i.upper() for i in finalDf.columns]
         finalDf.columns
```

```
Out[22]: Index(['TITLE', 'GENRE', 'DESCRIPTION', 'DIRECTOR', 'ACTORS', 'YEAR',
       'RUNTIME', 'RATING', 'VOTES', 'REVENUE_MILLIONS', 'METASCORE'],
      dtype='object')
```

When you convert one column as index column then that column will not be listed in the columns list.

Dropna(): To drop null rows.

Axis attribute: Used with dropna to drop the columns having null values.

Isnull(): To show where in the rows null values are present.

Sum(): Used with isnull() method to show the count of null values in each column.

Imputation: Do any activity to overcome the null values issue.

Fillna(): To replace null values with some default value.

Mean(): To calculate the mean.

Describe(): To get more information of the table like mean, count, etc.

Value_counts(): To show the records in groups by unique values of the column.

```
In [23]: finalDf.isnull()
```

Out[23]:

Rank	TITLE	GENRE	DESCRIPTION	DIRECTOR	ACTORS	YEAR	RUNTIME	RATING	VOTES	REVENUE_MILLIONS	METASCORE
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
996	False	False	False	False	False	False	False	False	False	True	False
997	False	False	False	False	False	False	False	False	False	False	False
998	False	False	False	False	False	False	False	False	False	False	False
999	False	False	False	False	False	False	False	False	False	True	False
1000	False	False	False	False	False	False	False	False	False	False	False

1000 rows × 11 columns

```
In [24]: finalDf.isnull().sum()
```

Out[24]:

```
TITLE                 0
GENRE                 0
DESCRIPTION           0
DIRECTOR              0
ACTORS                0
YEAR                  0
RUNTIME               0
RATING                0
VOTES                 0
REVENUE_MILLIONS    128
METASCORE            64
dtype: int64
```

In [25]:
```
test = finalDf.dropna()
test.isnull().sum()
```

Out[25]:
```
TITLE               0
GENRE               0
DESCRIPTION         0
DIRECTOR            0
ACTORS              0
YEAR                0
RUNTIME             0
RATING              0
VOTES               0
REVENUE_MILLIONS    0
METASCORE           0
dtype: int64
```

In [26]:
```
test = finalDf.dropna(axis=1)
test.isnull().sum()
```

Out[26]:
```
TITLE          0
GENRE          0
DESCRIPTION    0
DIRECTOR       0
ACTORS         0
YEAR           0
RUNTIME        0
RATING         0
VOTES          0
dtype: int64
```

In [27]:
```
finalDf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 1 to 1000
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   TITLE             1000 non-null   object 
 1   GENRE             1000 non-null   object 
 2   DESCRIPTION       1000 non-null   object 
 3   DIRECTOR          1000 non-null   object 
 4   ACTORS            1000 non-null   object 
 5   YEAR              1000 non-null   int64  
 6   RUNTIME           1000 non-null   int64  
 7   RATING            1000 non-null   float64
 8   VOTES             1000 non-null   int64  
 9   REVENUE_MILLIONS  872 non-null    float64
 10  METASCORE         936 non-null    float64
dtypes: float64(3), int64(3), object(5)
memory usage: 93.8+ KB
```

In [28]:
```
revenue = finalDf['REVENUE_MILLIONS']
```

In [29]:
```
type(finalDf)
```

Out[29]:
```
pandas.core.frame.DataFrame
```

In [30]:
```
type(revenue)
```

Out[30]:
```
pandas.core.series.Series
```

In [31]:
```
revenue.head()
```

Out[31]:
```
Rank
1    333.13
2    126.46
3    138.12
4    270.32
5    325.02
Name: REVENUE_MILLIONS, dtype: float64
```

In [32]:
```
rev_mean = revenue.mean()
rev_mean
```

Out[32]:
```
82.95637614678898
```

In [33]:
```
revenue.fillna(rev_mean, inplace=True)
```

```
/tmp/ipykernel_2442/1605635811.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  revenue.fillna(rev_mean, inplace=True)
```

In [34]:
```
finalDf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 1 to 1000
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   TITLE              1000 non-null   object 
 1   GENRE              1000 non-null   object 
 2   DESCRIPTION        1000 non-null   object 
 3   DIRECTOR           1000 non-null   object 
 4   ACTORS             1000 non-null   object 
 5   YEAR               1000 non-null   int64  
 6   RUNTIME            1000 non-null   int64  
 7   RATING             1000 non-null   float64
 8   VOTES              1000 non-null   int64  
 9   REVENUE_MILLIONS   1000 non-null   float64
 10  METASCORE          936 non-null    float64
dtypes: float64(3), int64(3), object(5)
memory usage: 93.8+ KB
```

In [35]:
```
metascore = finalDf['METASCORE']
```

In [36]:
```
metascore.head()
```

Out[36]:
```
Rank
1    76.0
2    65.0
3    62.0
4    59.0
5    40.0
Name: METASCORE, dtype: float64
```

In [37]:
```
meta_mean = metascore.mean()
meta_mean
```

Out[37]:
```
58.98504273504273
```

In [38]:
```
metascore.fillna(meta_mean, inplace=True)
```

```
/tmp/ipykernel_2442/1236312386.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  metascore.fillna(meta_mean, inplace=True)
```

In [39]:
```
finalDf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 1 to 1000
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   TITLE             1000 non-null   object 
 1   GENRE             1000 non-null   object 
 2   DESCRIPTION       1000 non-null   object 
 3   DIRECTOR          1000 non-null   object 
 4   ACTORS            1000 non-null   object 
 5   YEAR              1000 non-null   int64  
 6   RUNTIME           1000 non-null   int64  
 7   RATING            1000 non-null   float64
 8   VOTES             1000 non-null   int64  
 9   REVENUE_MILLIONS  1000 non-null   float64
 10  METASCORE         1000 non-null   float64
dtypes: float64(3), int64(3), object(5)
memory usage: 93.8+ KB
```

In [40]:
```
finalDf.describe()
```

Out[40]:

	YEAR	RUNTIME	RATING	VOTES	REVENUE_MILLIONS	METASCORE
count	1000.000000	1000.000000	1000.000000	1.000000e+03	1000.000000	1000.000000
mean	2012.783000	113.172000	6.723200	1.698083e+05	82.956376	58.985043
std	3.205962	18.810908	0.945429	1.887626e+05	96.412043	16.634858
min	2006.000000	66.000000	1.900000	6.100000e+01	0.000000	11.000000
25%	2010.000000	100.000000	6.200000	3.630900e+04	17.442500	47.750000
50%	2014.000000	111.000000	6.800000	1.107990e+05	60.375000	58.985043
75%	2016.000000	123.000000	7.400000	2.399098e+05	99.177500	71.000000
max	2016.000000	191.000000	9.000000	1.791916e+06	936.630000	100.000000

In [41]:
```
finalDf["REVENUE_MILLIONS"].describe()
```

Out[41]:
```
count    1000.000000
mean       82.956376
std        96.412043
min         0.000000
25%        17.442500
50%        60.375000
75%        99.177500
max       936.630000
Name: REVENUE_MILLIONS, dtype: float64
```

In [43]:
```
finalDf["REVENUE_MILLIONS"].value_counts().head()
```

Out[43]:
```
82.956376    128
0.030000       7
0.010000       5
0.020000       4
0.040000       4
Name: REVENUE_MILLIONS, dtype: int64
```

In [40]:
```
series1 = finalDf['TITLE']
```

In [41]:
```
df = finalDf[["TITLE", "REVENUE_MILLIONS"]]
```

In [42]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 1 to 1000
Data columns (total 2 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   TITLE             1000 non-null   object 
 1   REVENUE_MILLIONS  1000 non-null   float64
dtypes: float64(1), object(1)
memory usage: 23.4+ KB
```

In [43]:
```
df.describe()
```

Out[43]:

	REVENUE_MILLIONS
count	1000.000000
mean	82.956376
std	96.412043
min	0.000000
25%	17.442500
50%	60.375000
75%	99.177500
max	936.630000

In [44]:
```
df.loc[10]
```

Out[44]:
```
TITLE               Passengers
REVENUE_MILLIONS        100.01
Name: 10, dtype: object
```

In [45]:
```
df.iloc[5]
```

Out[45]:
```
TITLE               The Great Wall
REVENUE_MILLIONS             45.13
Name: 6, dtype: object
```

In [46]:
```
df.iloc[1:4]
```

Out[46]:

Rank	TITLE	REVENUE_MILLIONS
2	Prometheus	126.46
3	Split	138.12
4	Sing	270.32

In [47]:
```
finalDf[finalDf['REVENUE_MILLIONS'] >= 270.32]
```

Out[47]:

Rank	TITLE	GENRE	DESCRIPTION	DIRECTOR	ACTORS	YEAR	RUNTIME	RATING	VOTES	REVENUE_MI
1	Guardians of the Galaxy	Action,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	
4	Sing	Animation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	
5	Suicide Squad	Action,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	
13	Rogue One	Action,Adventure,Sci-Fi	The Rebel Alliance makes a risky move to steal...	Gareth Edwards	Felicity Jones, Diego Luna, Alan Tudyk, Donnie...	2016	133	7.9	323118	
	The Secret		The quiet life of a terrier	Chris	Louis C.K., Eric					

In [49]:
```
finalDf[(finalDf['DIRECTOR'] == 'Christopher Nolan') | (finalDf['GENRE'] == 'Adventure,Drama,Sci-Fi')]
```

Out[49]:

Rank	TITLE	GENRE	DESCRIPTION	DIRECTOR	ACTORS	YEAR	RUNTIME	RATING	VOTES	REVENUE_MILLIONS	METASCORE
37	Interstellar	Adventure,Drama,Sci-Fi	A team of explorers travel through a wormhole ...	Christopher Nolan	Matthew McConaughey, Anne Hathaway, Jessica Ch...	2014	169	8.6	1047747	187.99	74.0
55	The Dark Knight	Action,Crime,Drama	When the menace known as the Joker wreaks havo...	Christopher Nolan	Christian Bale, Heath Ledger, Aaron Eckhart,Mi...	2008	152	9.0	1791916	533.32	82.0
65	The Prestige	Drama,Mystery,Sci-Fi	Two stage magicians engage in competitive one-...	Christopher Nolan	Christian Bale, Hugh Jackman, Scarlett Johanss...	2006	130	8.5	913152	53.08	66.0
81	Inception	Action,Adventure,Sci-Fi	A thief, who steals corporate secrets through ...	Christopher Nolan	Leonardo DiCaprio, Joseph Gordon-Levitt, Ellen...	2010	148	8.8	1583625	292.57	74.0
103	The Martian	Adventure,Drama,Sci-Fi	An astronaut becomes stranded on Mars after hi...	Ridley Scott	Matt Damon, Jessica Chastain, Kristen Wiig, Ka...	2015	144	8.0	556097	228.43	80.0
125	The Dark Knight Rises	Action,Thriller	Eight years after the Joker's reign of anarchy...	Christopher Nolan	Christian Bale, Tom Hardy, Anne Hathaway,Gary ...	2012	164	8.5	1222645	448.13	78.0

In [50]:
```
def rating_function(rating):
    if rating >= 5.0:
        return "Good"
    else:
        return "Bad"
```

In [51]:
```
finalDf["movie_rating_value"] = finalDf["RATING"].apply(rating_function)
```

```
/tmp/ipykernel_2287/4052055073.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  finalDf["movie_rating_value"] = finalDf["RATING"].apply(rating_function)
```

In [52]:
```
finalDf.head()
```

Out[52]:

	GENRE	DESCRIPTION	DIRECTOR	ACTORS	YEAR	RUNTIME	RATING	VOTES	REVENUE_MILLIONS	METASCORE	movie_rating_value
	ion,Adventure,Sci-Fi	A group of intergalactic criminals are forced ...	James Gunn	Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S...	2014	121	8.1	757074	333.13	76.0	Good
	enture,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	2012	124	7.0	485820	126.46	65.0	Good
	Horror,Thriller	Three girls are kidnapped by a man with a diag...	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu Richar...	2016	117	7.3	157606	138.12	62.0	Good
	ation,Comedy,Family	In a city of humanoid animals, a hustling thea...	Christophe Lourdelet	Matthew McConaughey,Reese Witherspoon, Seth Ma...	2016	108	7.2	60545	270.32	59.0	Good
	n,Adventure,Fantasy	A secret government agency recruits some of th...	David Ayer	Will Smith, Jared Leto, Margot Robbie, Viola D...	2016	123	6.2	393727	325.02	40.0	Good

Matplotlib: used to plot graphs.

In [53]:
```
import matplotlib.pyplot as plt
```

```
Matplotlib is building the font cache; this may take a moment.
```

In [56]:
```
finalDf.plot(kind='bar', x='RATING', y='REVENUE_MILLIONS', title='Revenue vs Rating')
```

Out[56]:
```
<Axes: title={'center': 'Revenue vs Rating'}, xlabel='RATING'>
```