This notebook is in line with the tutorial in https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

```
import pandas as pd
```

```
DatasetBaseFolder = '/content/gdrive/MyDrive/ColabNotebooks/AI club files/'
```

```
data = {
    'apples' : [0, 2, 1, 3],
    'oranges' : [1, 5, 2, 4]
}

purchases = pd.DataFrame(data);
purchases
```

	apples	oranges
0	0	1
1	2	5
2	1	2
3	3	4

```
movies_df = AI club files.read_csv(DatasetBaseFolder+"Copy of IMDB-Movie-Data.csv", index_col
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-16-4e533a8bbe2e> in <module>()
----> 1 movies_df = AIclubfiles.read_csv(DatasetBaseFolder+"Copy of IMDB-Movie-
Data.csv", index_col="Title")

NameError: name 'AIclubfiles' is not defined
```

SEARCH STACK OVERFLOW

```
#Lets see first 5 rows
movies_df.head(5)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-18-ef35342fb93e> in <module>()
      1 #Lets see first 5 rows
----> 2 movies_df.head(5)

NameError: name 'movies_df' is not defined
```

SEARCH STACK OVERFLOW

```
#Lets see last 5 rows
movies_df.tail(5)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-17-9eda529fe65a> in <module>()
      1 #Lets see last 5 rows
----> 2 movies_df.tail(5)

NameError: name 'movies_df' is not defined
```

SEARCH STACK OVERFLOW

```
movies_df.shape
```

```
(1000, 11)
```

```
#To get an overview of the dataset
movies_df.info()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-11-063239a43a12> in <module>()
      1 #To get an overview of the dataset
----> 2 movies_df.info()

NameError: name 'movies_df' is not defined
```

SEARCH STACK OVERFLOW

```
#If you want to remove duplicate instances
movies_df = movies_df.drop_duplicates(keep = 'first') #Drop all instances keep = false inplac
```

```
#If you wish to rename columns
movies_df.columns
```

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

```
movies_df.rename(columns = {'Runtime (Minutes)' : 'Runtime', 'Revenue (Millions)' : 'Revenue_
movies_df.columns
```

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

```
#To count number of null entries in each colum
movies_df.isnull().sum()
```

```
Rank                  0
Genre                 0
Description           0
Director              0
Actors                0
Year                  0
Runtime               0
Rating                0
Votes                 0
Revenue_millions    128
Metascore            64
dtype: int64
```

```
movies_dfTmp = movies_df.dropna(axis=0) #To drop instances with null values
movies_dfTmp.shape
#movies_df.shape
```

```
(838, 11)
```

```
movies_dfTmp = movies_df.dropna(axis=1) #To drop columns containing null values
movies_dfTmp.shape
```

```
(1000, 9)
```

```
movies_df.shape
```

```
(1000, 11)
```

```
#Imputing with Mean
revenue = movies_df['Revenue_millions']
revenue.head(5)
```

```
Title
Guardians of the Galaxy    333.13
Prometheus                 126.46
Split                      138.12
Sing                       270.32
Suicide Squad              325.02
Name: Revenue_millions, dtype: float64
```

```
meanRev = revenue.mean(0)
revenue.fillna(meanRev, inplace=True)
movies_df.isnull().sum() #Note that this get updated
```

```
Rank                 0
Genre                0
Description          0
Director             0
Actors               0
Year                 0
Runtime              0
Rating               0
Votes                0
Revenue_millions     0
Metascore           64
dtype: int64
```

```
#Describ the Dataset
movies_df.describe()
```

	Rank	Year	Runtime	Rating	Votes	Revenue_millions
count	1000.000000	1000.000000	1000.000000	1000.000000	1.000000e+03	1000.000000
mean	500.500000	2012.783000	113.172000	6.723200	1.698083e+05	82.956376
std	288.819436	3.205962	18.810908	0.945429	1.887626e+05	96.412043
min	1.000000	2006.000000	66.000000	1.900000	6.100000e+01	0.000000
25%	250.750000	2010.000000	100.000000	6.200000	3.630900e+04	17.442500
50%	500.500000	2014.000000	111.000000	6.800000	1.107990e+05	60.375000
75%	750.250000	2016.000000	123.000000	7.400000	2.399098e+05	99.177500
max	1000.000000	2016.000000	191.000000	9.000000	1.791916e+06	936.630000

```
Title
Interstellar        8.6
The Dark Knight     9.0
Inception           8.8
Kimi no na wa       8.6
Dangal              8.8
The Intouchables    8.6
Name: Rating, dtype: float64
```

```
#if you want to count
movies_df['Genre'].value_counts()
```

```
Action,Adventure,Sci-Fi     50
Drama                       48
Comedy,Drama,Romance        35
Comedy                      32
Drama,Romance               31
                            ..
Action,Comedy,Sport          1
Adventure,Horror,Mystery     1
Crime,Thriller               1
Drama,Family,Music           1
Drama,Thriller,War           1
Name: Genre, Length: 207, dtype: int64
```

```
#Correlation
movies_df.corr()  #Note the attributes in  S
```

	Rank	Year	Runtime	Rating	Votes	Revenue_millions	M
Rank	1.000000	-0.261605	-0.221739	-0.219555	-0.283876	-0.252996	
Year	-0.261605	1.000000	-0.164900	-0.211219	-0.411904	-0.117562	
Runtime	-0.221739	-0.164900	1.000000	0.392214	0.407062	0.247834	
Rating	-0.219555	-0.211219	0.392214	1.000000	0.511537	0.189527	
Votes	-0.283876	-0.411904	0.407062	0.511537	1.000000	0.607941	
Revenue_millions	-0.252996	-0.117562	0.247834	0.189527	0.607941	1.000000	
Metascore	-0.191869	-0.079305	0.211978	0.631897	0.325684	0.133328	

```
#slicing along columns
subset = movies_df[['Genre', 'Rating']]
type(subset)
```

```
pandas.core.frame.DataFrame
```

```
#Slicing along rows
movies_df.loc['Prometheus'] #using key index
movies_df.iloc[1] #using numerical index
```

```
Rank                                                        2
Genre                                  Adventure,Mystery,Sci-Fi
Description         Following clues to the origin of mankind, a te...
Director                                            Ridley Scott
Actors              Noomi Rapace, Logan Marshall-Green, Michael Fa...
Year                                                     2012
Runtime                                                   124
Rating                                                      7
Votes                                                  485820
Revenue_millions                                       126.46
```

```
Metascore                                                 65
Name: Prometheus, dtype: object
```

```
#few instances 1 through 3
movie_subset = movies_df.iloc[1:4]
movie_subset
```

Title	Rank	Genre	Description	Director	Actors	Ye
Prometheus	2	Adventure,Mystery,Sci-Fi	Following clues to the origin of mankind, a te...	Ridley Scott	Noomi Rapace, Logan Marshall-Green, Michael Fa...	20
Split	3	Horror,Thriller	Three girls are kidnapped by	M. Night Shyamalan	James McAvoy, Anya Taylor-Joy, Haley Lu	20

```
#conditional selection
#Pick movies with rating more than 8.5
rating = movies_df['Rating']
rating[rating.gt(8.5)]
```

```
Title
Interstellar        8.6
The Dark Knight     9.0
Inception           8.8
Kimi no na wa       8.6
Dangal              8.8
The Intouchables    8.6
Name: Rating, dtype: float64
```

```
#Pick movies based on Director
moviesByRidley = movies_df[(movies_df['Director'] == "Ridley Scott") & movies_df['Rating'].gt
moviesByRidley.head(4)
```

Title	Rank	Genre	Description	Director	Actors	Year	Runtime
The Martian	103	Adventure,Drama,Sci-Fi	An astronaut becomes stranded on Mars after hi...	Ridley Scott	Matt Damon, Jessica Chastain, Kristen Wiig, Ka...	2015	144

```
#all movies that were released between 2005 and 2010, have a rating above 8.0, but made below
movies_df[
    ((movies_df['Year'] >= 2005) & (movies_df['Year'] <= 2010))
    & (movies_df['Rating'] > 8.0)
```

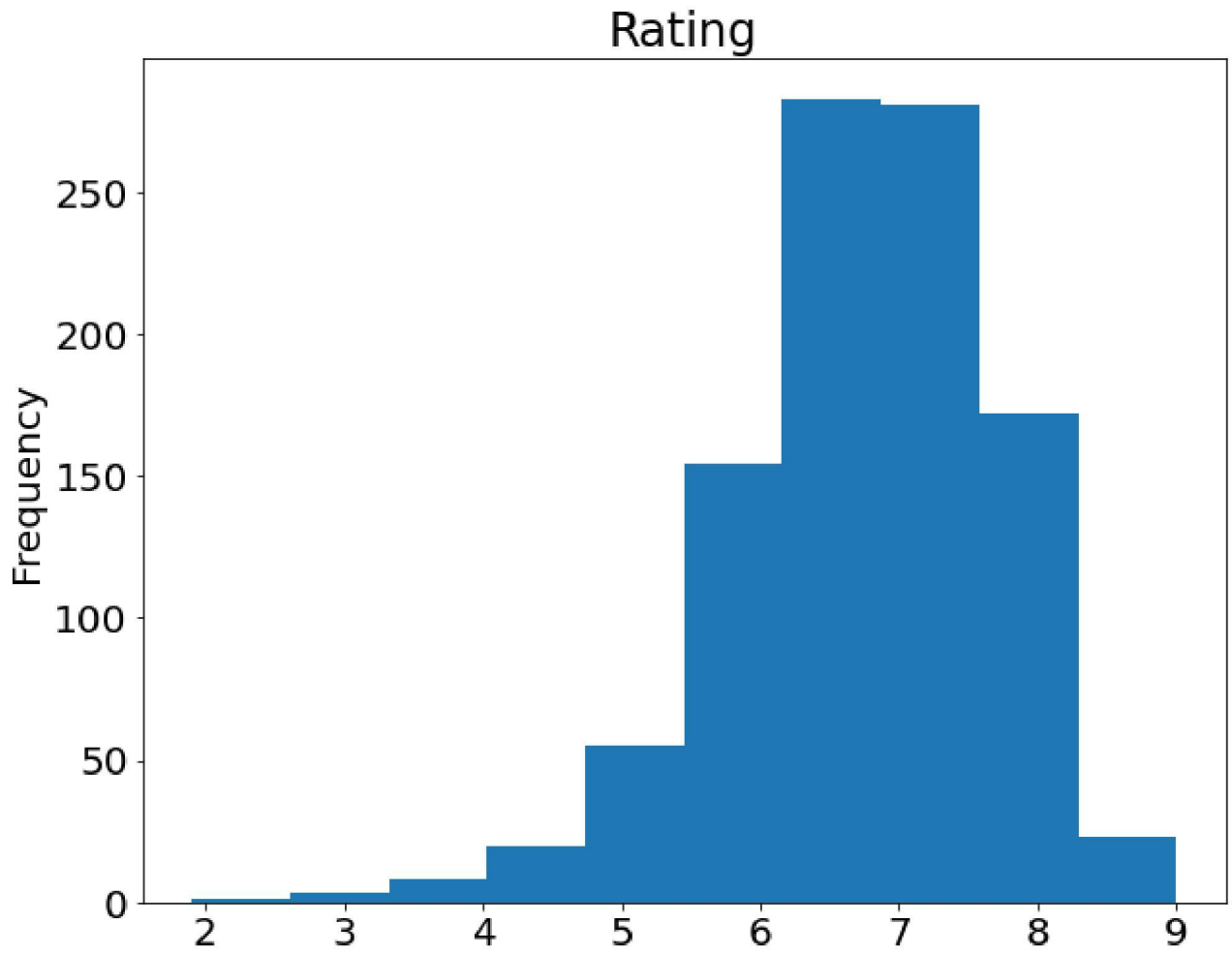
```
    & (movies_df['Revenue_millions'] < movies_df['Revenue_millions'].quantile(0.25))
]
```

Title	Rank	Genre	Description	Director	Actors	Year	Rur
3 Idiots	431	Comedy,Drama	Two friends are searching for their long lost ...	Rajkumar Hirani	Aamir Khan, Madhavan, Mona Singh, Sharman Joshi	2009	
The Lives of Others	477	Drama,Thriller	In 1984 East Berlin, an agent of the secret po...	Florian Henckel von Donnersmarck	Ulrich Mühe, Martina Gedeck,Sebastian Koch, Ul...	2006	
			Twins journey to the		Lubna Azabal,		

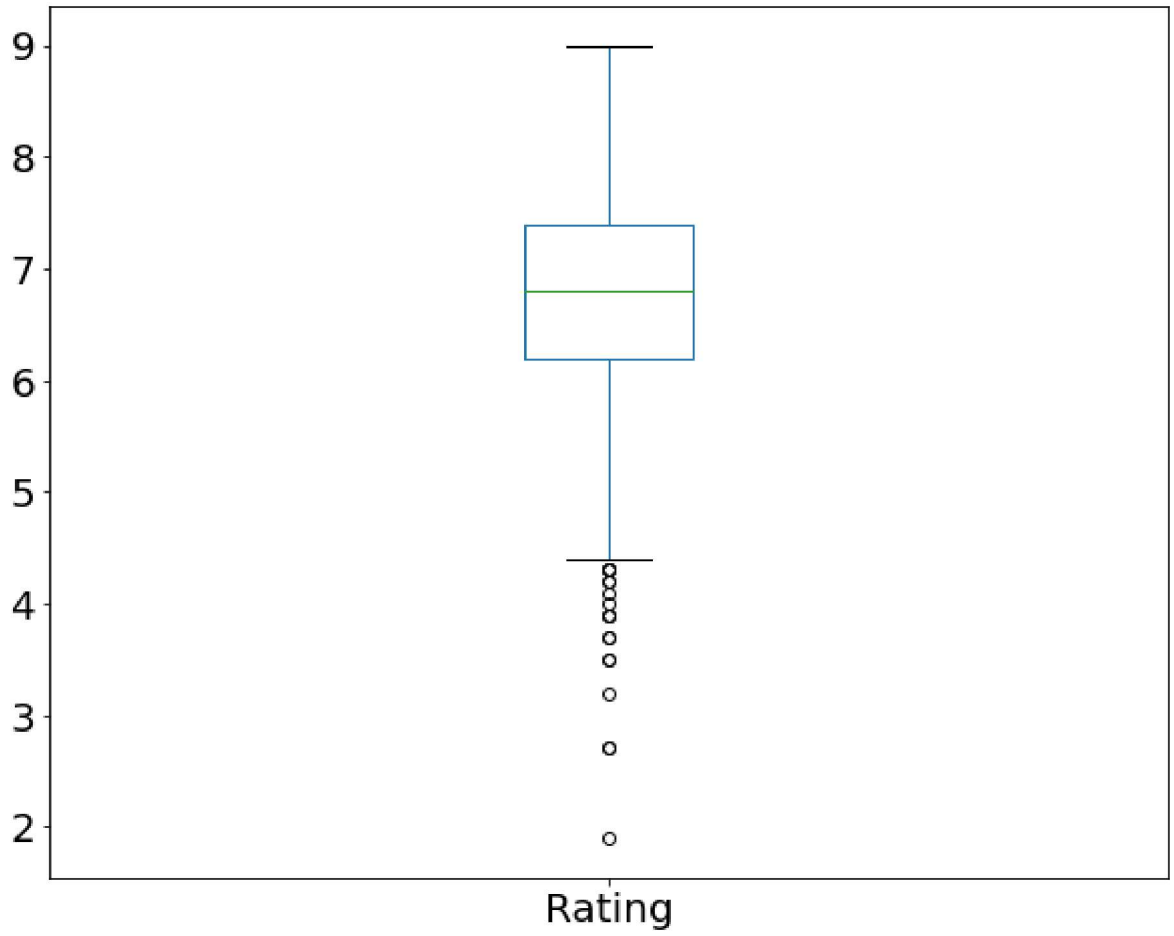
```
import matplotlib.pyplot as plt
plt.rcParams.update({'font.size': 20, 'figure.figsize': (10, 8)})


#For categorical variables utilize Bar Charts* and Boxplots.
#For continuous variables utilize Histograms, Scatterplots, Line graphs, and Boxplots.
movies_df.plot(kind='scatter', x='Rating', y='Revenue_millions', title='Revenue (millions) vs
```

Revenue (millions) vs Rating

```
movies_df['Rating'].plot(kind='hist', title='Rating');
```




```
movies_df['Rating'].plot(kind="box");
```



0s completed at 19:03