In [1]:
```
#SWETHA JENIFER S_25-1-23
```

Lab3. Pandas Indexing and Selection

Simple Series and DataFrames

In [2]:
```
#Import necessary modules
```

In [3]:
```
import pandas as pd
```

In [4]:
```
#Create a Series to store Temperature values for 1 week
temperature_trichy = pd.Series([40.2, 39.8, 36.3, 39.1, 41.3, 32.9, 36.6])
```

In [5]:
```
#show temperature values
temperature_trichy
```

Out[5]:
```
0    40.2
1    39.8
2    36.3
3    39.1
4    41.3
5    32.9
6    36.6
dtype: float64
```

In [6]:
```
#What is the weather on 2nd day?
temperature_trichy[1]
```

Out[6]:
```
39.8
```

In [7]:
```
#Find all days and temperatures where temperature over 40.0 degree Celsius
temperature_trichy[temperature_trichy>40]
```

Out[7]:
```
0    40.2
4    41.3
dtype: float64
```

In [8]:
```
#Find only day, not temperature where temperature over 40.0 degree Celsius
temperature_trichy[temperature_trichy>=40].index
```

Out[8]:
```
Int64Index([0, 4], dtype='int64')
```

Create a Dataframe for student details from List

In [9]:
```
students = [['DS01', 'Rex', '1msc'], ['DS02', 'peter', '2msc'], ['CS01', 'ann', '
df_stud = pd.DataFrame(students, columns=['rollno', 'name', 'class'])
```

In [10]:
```
#show df_stud dataframe
df_stud
```

Out[10]:

	rollno	name	class
0	DS01	Rex	1msc
1	DS02	peter	2msc
2	CS01	ann	3bsc

In [11]:
```
#show df_stud dataframe
df_stud.columns
```

Out[11]:
```
Index(['rollno', 'name', 'class'], dtype='object')
```

In [12]:
```
#Add a new column "address" with values ['Delhi', 'Bangalore', 'Chennai'] to df_s
df_stud['address']=['Delhi','Bangalore','Chennai']
```

In [13]:
```
df_stud
```

Out[13]:

	rollno	name	class	address
0	DS01	Rex	1msc	Delhi
1	DS02	peter	2msc	Bangalore
2	CS01	ann	3bsc	Chennai

Create a Dataframe for Phone book from Dictionary

In [14]:
```
phonebook = {'rex':[9942002764, 'rex@abc.com'], 'sam':[9932176542, 'sam@xyz.com']
```

In [15]:
```
df_phonebook = pd.DataFrame(phonebook)
```

In [16]:
```
#Display df_phonebook
df_phonebook
```

Out[16]:

	peter	rex	sam
0	9865323645	9942002764	9932176542
1	ann@bhc.com	rex@abc.com	sam@xyz.com

Exploratory Data Analysis on Video Game Review Dataset

In [17]:
```
#Import ign.csv dataset
reviews=pd.read_csv("ign.csv")
```

In [18]:
```
#Show top-5 rows
reviews.head(5)
```

Out[18]:

	Unnamed: 0	score_phrase	title	url	platform	score	genre	edito
0	0	Amazing	LittleBigPlanet PS Vita	/games/littlebigplanet-vita/vita-98907	PlayStation Vita	9.0	Platformer	
1	1	Amazing	LittleBigPlanet PS Vita -- Marvel Super Hero E...	/games/littlebigplanet-ps-vita-marvel-super-he...	PlayStation Vita	9.0	Platformer	
2	2	Great	Splice: Tree of Life	/games/splice/ipad-141070	iPad	8.5	Puzzle	
3	3	Great	NHL 13	/games/nhl-13/xbox-360-128182	Xbox 360	8.5	Sports	
4	4	Great	NHL 13	/games/nhl-13/ps3-128181	PlayStation 3	8.5	Sports	

In [19]:
```
#Show bottom 3 rows
reviews.tail(3)
```

Out[19]:

	Unnamed: 0	score_phrase	title	url	platform	score	genre	editors
18622	18622	Mediocre	Star Ocean: Integrity and Faithlessness	/games/star-ocean-5/ps4-20035681	PlayStation 4	5.8	RPG	
18623	18623	Masterpiece	Inside	/games/inside-playdead/xbox-one-121435	Xbox One	10.0	Adventure	
18624	18624	Masterpiece	Inside	/games/inside-playdead/pc-20055740	PC	10.0	Adventure	

In [20]:
```
#How many rows and columns here?
reviews.shape
```

Out[20]:
```
(18625, 11)
```

In [21]:
```
#What are the datatypes?
reviews.dtypes
```

Out[21]:
```
Unnamed: 0          int64
score_phrase       object
title              object
url                object
platform           object
score             float64
genre              object
editors_choice     object
release_year        int64
release_month       int64
release_day         int64
dtype: object
```

Selecting Columns

In [22]:
```
#Select a single column, say title and print head
reviews['title'].tail()
```

Out[22]:
```
18620                  Tokyo Mirage Sessions #FE
18621          LEGO Star Wars: The Force Awakens
18622    Star Ocean: Integrity and Faithlessness
18623                                     Inside
18624                                     Inside
Name: title, dtype: object
```

In [23]:
```
#Select multiple columns, title and genre and print head
reviews[['title','genre']].head(10)
```

Out[23]:

	title	genre
0	LittleBigPlanet PS Vita	Platformer
1	LittleBigPlanet PS Vita -- Marvel Super Hero E...	Platformer
2	Splice: Tree of Life	Puzzle
3	NHL 13	Sports
4	NHL 13	Sports
5	Total War Battles: Shogun	Strategy
6	Double Dragon: Neon	Fighting
7	Guild Wars 2	RPG
8	Double Dragon: Neon	Fighting
9	Total War Battles: Shogun	Strategy

Selection using Positions

In [24]:
```
#Select top-5 rows and all columns, same as head() using iloc
reviews.iloc[0:5]
```

Out[24]:

	Unnamed: 0	score_phrase	title	url	platform	score	genre	edito
0	0	Amazing	LittleBigPlanet PS Vita	/games/littlebigplanet-vita/vita-98907	PlayStation Vita	9.0	Platformer	
1	1	Amazing	LittleBigPlanet PS Vita -- Marvel Super Hero E...	/games/littlebigplanet-ps-vita-marvel-super-he...	PlayStation Vita	9.0	Platformer	
2	2	Great	Splice: Tree of Life	/games/splice/ipad-141070	iPad	8.5	Puzzle	
3	3	Great	NHL 13	/games/nhl-13/xbox-360-128182	Xbox 360	8.5	Sports	
4	4	Great	NHL 13	/games/nhl-13/ps3-128181	PlayStation 3	8.5	Sports	

In [25]:
```
#Select rows from position 5 onwards, and columns from position 5 onwards.
reviews.iloc[4:,4:]
```

Out[25]:

	platform	score	genre	editors_choice	release_year	release_month	release_day
4	PlayStation 3	8.5	Sports	N	2012	9	11
5	Macintosh	7.0	Strategy	N	2012	9	11
6	Xbox 360	3.0	Fighting	N	2012	9	11
7	PC	9.0	RPG	Y	2012	9	11
8	PlayStation 3	3.0	Fighting	N	2012	9	11
9	PC	7.0	Strategy	N	2012	9	11
10	PlayStation 3	7.5	Fighting	N	2012	9	11
11	Xbox 360	7.5	Fighting	N	2012	9	11
12	iPhone	7.0	NaN	N	2012	9	10
13	Xbox 360	9.0	Action, Adventure	Y	2012	9	7
14	PC	9.0	Action, Adventure	Y	2012	9	7
15	Macintosh	6.5	Adventure	N	2012	9	6
16	PC	6.5	Adventure	N	2012	9	6
17	iPhone	8.0	Action	N	2012	9	5
18	PlayStation 3	5.5	Action, Adventure	N	2012	9	3
19	Xbox 360	7.0	Fighting	N	2012	9	3
20	PlayStation 3	7.0	Fighting	N	2012	9	3
21	Xbox 360	7.5	RPG	N	2012	8	31
22	PlayStation 3	7.5	RPG	N	2012	8	31
23	PC	7.5	RPG	N	2012	8	31
24	PC	9.0	Action, RPG	Y	2012	8	31
25	PC	7.0	Shooter	N	2012	8	30
26	iPad	9.0	Action, RPG	Y	2012	8	30
27	PC	7.5	Shooter	N	2012	8	29
28	PC	8.0	Adventure	N	2012	8	29
29	PlayStation 3	6.5	Action, RPG	N	2012	8	28
30	Macintosh	9.0	Adventure	Y	2012	8	28
31	PC	8.7	RPG	Y	2012	10	4

	platform	score	genre	editors_choice	release_year	release_month	release_day
32	PlayStation 3	4.9	Platformer	N	2012	10	4
33	Nintendo DS	9.6	RPG	Y	2012	10	3
...	...	...	...	...	...	...	...
18595	PC	4.4	Action	N	2016	7	16
18596	PC	6.5	Action, Adventure	N	2016	7	14
18597	Xbox One	4.9	Shooter, Adventure	N	2016	7	13
18598	PC	6.8	Action	N	2016	7	13
18599	Android	7.0	Battle	N	2016	7	13
18600	PC	7.4	Shooter	N	2016	8	19
18601	PlayStation 4	7.4	Shooter	N	2016	8	19
18602	Xbox One	7.4	Shooter	N	2016	8	19
18603	PC	7.8	Platformer	N	2016	8	18
18604	PlayStation 4	8.6	Sports	N	2016	8	17
18605	PlayStation 4	6.0	Adventure	N	2016	8	16
18606	PC	6.4	Strategy	N	2016	8	4
18607	iPhone	7.0	Battle	N	2016	7	13
18608	PlayStation 4	5.4	Racing, Action	N	2016	7	13
18609	Nintendo 3DS	8.0	Action	N	2016	7	12
18610	PlayStation 4	6.0	Adventure	N	2016	7	12
18611	Xbox One	5.8	Shooter	N	2016	7	6
18612	Nintendo 3DS	7.8	Puzzle	N	2016	7	6
18613	PC	8.0	Strategy	N	2016	7	1
18614	Nintendo 3DS	9.2	Adventure	Y	2016	6	29
18615	PlayStation Vita	9.2	Adventure	Y	2016	6	29
18616	PC	7.5	Adventure	N	2016	8	2
18617	PlayStation 4	8.4	Adventure	N	2016	8	2
18618	PC	9.1	Action	Y	2016	7	28
18619	PC	7.9	Puzzle, Action	N	2016	7	28

	platform	score	genre	editors_choice	release_year	release_month	release_day
18620	Wii U	7.6	RPG	N	2016	6	29
18621	PlayStation 4	9.0	Action, Adventure	Y	2016	6	29
18622	PlayStation 4	5.8	RPG	N	2016	6	28
18623	Xbox One	10.0	Adventure	Y	2016	6	28
18624	PC	10.0	Adventure	Y	2016	6	28

18621 rows × 7 columns

In [26]:
```
#Select the first column, and all of the rows for the column
reviews.iloc[:,0]
```

Out[26]:
```
0            0
1            1
2            2
3            3
4            4
5            5
6            6
7            7
8            8
9            9
10          10
11          11
12          12
13          13
14          14
15          15
16          16
17          17
18          18
19          19
20          20
21          21
22          22
23          23
24          24
25          25
26          26
27          27
28          28
29          29
           ...
18595    18595
18596    18596
18597    18597
18598    18598
18599    18599
18600    18600
18601    18601
18602    18602
18603    18603
18604    18604
18605    18605
18606    18606
18607    18607
18608    18608
18609    18609
18610    18610
18611    18611
18612    18612
18613    18613
18614    18614
18615    18615
18616    18616
18617    18617
```

```
18618    18618
18619    18619
18620    18620
18621    18621
18622    18622
18623    18623
18624    18624
Name: Unnamed: 0, Length: 18625, dtype: int64
```

In [27]:
```
#the 10th row, and all of the columns for that row.
reviews.iloc[10,:]
```

Out[27]:
```
Unnamed: 0                                                    10
score_phrase                                                Good
title                                    Tekken Tag Tournament 2
url             /games/tekken-tag-tournament-2/ps3-124584
platform                                          PlayStation 3
score                                                       7.5
genre                                                  Fighting
editors_choice                                                N
release_year                                               2012
release_month                                                 9
release_day                                                  11
Name: 10, dtype: object
```

In [28]:
```
#First column is not useful. So remove it
reviews.drop(reviews.columns[0],axis=1,inplace=True)
```

In [29]:
```
reviews.head()
```

Out[29]:

	score_phrase	title	url	platform	score	genre	editors_choice	r
0	Amazing	LittleBigPlanet PS Vita	/games/littlebigplanet-vita/vita-98907	PlayStation Vita	9.0	Platformer	Y	
1	Amazing	LittleBigPlanet PS Vita -- Marvel Super Hero E...	/games/littlebigplanet-ps-vita-marvel-super-he...	PlayStation Vita	9.0	Platformer	Y	
2	Great	Splice: Tree of Life	/games/splice/ipad-141070	iPad	8.5	Puzzle	N	
3	Great	NHL 13	/games/nhl-13/xbox-360-128182	Xbox 360	8.5	Sports	N	
4	Great	NHL 13	/games/nhl-13/ps3-128181	PlayStation 3	8.5	Sports	N	

Selection using Row and Column Labels

In [30]:
```
students=[['DS01','Rex','1msc'],['Ds02','peter','2msc'],['cs01','ann','3bsc']]
df_stud=pd.DataFrame(students,columns=['rollno','name','class'])
```

In [31]:
```
df_stud
```

Out[31]:

	rollno	name	class
0	DS01	Rex	1msc
1	Ds02	peter	2msc
2	cs01	ann	3bsc

In [32]:
```
#Print all names using loc
df_stud['name'].loc[:]
```

Out[32]:
```
0      Rex
1    peter
2      ann
Name: name, dtype: object
```

In [33]:
```
#Let us come back to our reviews. Display the first five rows of reviews using th
reviews.loc[0:5]
```

Out[33]:

	score_phrase	title	url	platform	score	genre	editors_choice	r
0	Amazing	LittleBigPlanet PS Vita	/games/littlebigplanet-vita/vita-98907	PlayStation Vita	9.0	Platformer	Y	
1	Amazing	LittleBigPlanet PS Vita -- Marvel Super Hero E...	/games/littlebigplanet-ps-vita-marvel-super-he...	PlayStation Vita	9.0	Platformer	Y	
2	Great	Splice: Tree of Life	/games/splice/ipad-141070	iPad	8.5	Puzzle	N	
3	Great	NHL 13	/games/nhl-13/xbox-360-128182	Xbox 360	8.5	Sports	N	
4	Great	NHL 13	/games/nhl-13/ps3-128181	PlayStation 3	8.5	Sports	N	
5	Good	Total War Battles: Shogun	/games/total-war-battles-shogun/mac-142565	Macintosh	7.0	Strategy	N	

In [34]:
```
#Select score_phrase column using loc and print head
reviews['score_phrase'].loc[:4]
```

Out[34]:
```
0    Amazing
1    Amazing
2      Great
3      Great
4      Great
Name: score_phrase, dtype: object
```

In [35]:
```
#Print top 10 values of column label "score_phrase"
reviews['score_phrase'].loc[:9]
```

Out[35]:
```
0    Amazing
1    Amazing
2      Great
3      Great
4      Great
5       Good
6      Awful
7    Amazing
8      Awful
9       Good
Name: score_phrase, dtype: object
```

In [36]:
```
#Select from reviews of rows from 5 to 15
some_reviews=reviews.iloc[5:15,:]
```

In [37]:
```
#print top 5 rows from some_reviews
some_reviews.head()
```

Out[37]:

	score_phrase	title	url	platform	score	genre	editors_choice	release_year
5	Good	Total War Battles: Shogun	/games/total-war-battles-shogun/mac-142565	Macintosh	7.0	Strategy	N	2012
6	Awful	Double Dragon: Neon	/games/double-dragon-neon/xbox-360-131320	Xbox 360	3.0	Fighting	N	2012
7	Amazing	Guild Wars 2	/games/guild-wars-2/pc-896298	PC	9.0	RPG	Y	2012
8	Awful	Double Dragon: Neon	/games/double-dragon-neon/ps3-131321	PlayStation 3	3.0	Fighting	N	2012
9	Good	Total War Battles: Shogun	/games/total-war-battles-shogun/pc-142564	PC	7.0	Strategy	N	2012

In [38]:
```
#Select scores of first 3 rows some_reviews
some_reviews['score'].head(3)
```

Out[38]:
```
5    7.0
6    3.0
7    9.0
Name: score, dtype: float64
```

In [39]:
```
#Select "score", "genre", and "release_year" columns from reviews dataframe and pr
reviews[['score','genre','release_year']].head()
```

Out[39]:

	score	genre	release_year
0	9.0	Platformer	2012
1	9.0	Platformer	2012
2	8.5	Puzzle	2012
3	8.5	Sports	2012
4	8.5	Sports	2012

In [40]:
```
#What is the datatype of "score" column?
type(reviews['score'])
```

Out[40]:
```
pandas.core.series.Series
```

Aggregate Columns

In [41]:
```
#Find average value of score column in reviews dataframe
reviews['score'].mean()
```

Out[41]:
```
6.950459060402666
```

In [42]:
```
#Find average value of all numeric columns
reviews.mean()
```

Out[42]:
```
score              6.950459
release_year    2006.515329
release_month      7.138470
release_day       15.603866
dtype: float64
```

In [43]:
```
#Find average value for each row containing numeric values and print head
reviews.mean(axis=1).head()
```

Out[43]:
```
0    510.500
1    510.500
2    510.375
3    510.125
4    510.125
dtype: float64
```

Find lowest, highest, median, standard deviation of score column of reviews dataframe

In [44]:
```
#show median of "score" column of reviews dataframe
reviews['score'].median()
```

Out[44]: 7.3

In [45]:
```
#show minimum of "score" column of reviews dataframe
reviews['score'].min()
```

Out[45]: 0.5

In [46]:
```
#show maximum of "score" column of reviews dataframe
reviews['score'].max()
```

Out[46]: 10.0

In [47]:
```
#show standard deviation of "score" column of reviews dataframe
reviews['score'].std()
```

Out[47]: 1.7117358608045874

In [48]:
```
#How many non-null values in "score" column of reviews dataframe?
reviews['score'].notnull().sum()
```

Out[48]: 18625

In [49]:
```
#Show the summary of reviews dataframe
reviews.describe()
```

Out[49]:

	score	release_year	release_month	release_day
count	18625.000000	18625.000000	18625.00000	18625.000000
mean	6.950459	2006.515329	7.13847	15.603866
std	1.711736	4.587529	3.47671	8.690128
min	0.500000	1970.000000	1.00000	1.000000
25%	6.000000	2003.000000	4.00000	8.000000
50%	7.300000	2007.000000	8.00000	16.000000
75%	8.200000	2010.000000	10.00000	23.000000
max	10.000000	2016.000000	12.00000	31.000000

In [50]:
```
#Check if review score has any correlation with other columns of reviews
reviews.corr()
```

Out[50]:

	score	release_year	release_month	release_day
score	1.000000	0.062716	0.007632	0.020079
release_year	0.062716	1.000000	-0.115515	0.016867
release_month	0.007632	-0.115515	1.000000	-0.067964
release_day	0.020079	0.016867	-0.067964	1.000000

Math Operations on DF columns

In [51]:
```
#Divide the values of "score" column in reviews dataframe by 2. There will be too
M=reviews['score']/2
M.head()
```

Out[51]:
```
0    4.50
1    4.50
2    4.25
3    4.25
4    4.25
Name: score, dtype: float64
```

Boolean Indexing in Pandas

In [52]:
```
#Select all video games whose review score > 7, call it score_filter
score_filter=reviews['score']>7
```

In [53]:
```
#Print head of score_filter
score_filter.head()
```

Out[53]:
```
0    True
1    True
2    True
3    True
4    True
Name: score, dtype: bool
```

In [54]:
```
#Select all rows for score_filter column and print its head
filtered_review=reviews[score_filter]
```

In [55]:
```
filtered_review
```

27	Good	Tom Clancy's Ghost Recon Phantoms	/games/tom-clancys-ghost-recon-online/pc-109114	PC	7.5	Shooter
28	Great	Thirty Flights of Loving	/games/thirty-flights-of-loving/pc-138374	PC	8.0	Adventure
30	Amazing	The Walking Dead: The Game -- Episode 3: Long ...	/games/the-walking-dead-season-1-episode-3/mac...	Macintosh	9.0	Adventure
31	Great	World of Warcraft: Mists of Pandaria	/games/world-of-warcraft-mists-of-pandaria/pc-...	PC	8.7	RPG
33	Amazing	Pokemon White Version 2	/games/pokemon-white-version-2/nds-129228	Nintendo DS	9.6	RPG
		War of the	/games/war-of-the-			

In [56]:
```
#Show the size of filtered_reviews
filtered_review.shape
```

Out[56]:
```
(9800, 10)
```

In [57]:
```
#Show top 10 "title" from filtered_reviews
filtered_review['title'].head(10)
```

Out[57]:
```
0                                LittleBigPlanet PS Vita
1     LittleBigPlanet PS Vita -- Marvel Super Hero E...
2                                  Splice: Tree of Life
3                                               NHL 13
4                                               NHL 13
7                                           Guild Wars 2
10                              Tekken Tag Tournament 2
11                              Tekken Tag Tournament 2
13                                    Mark of the Ninja
14                                    Mark of the Ninja
Name: title, dtype: object
```

In [58]:
```
#First create a filter, called xbox_one_filter for the conditions
xbox_one_filter=(reviews['score']>7)&(reviews['platform']=='Xbox One')
filtered_reviews2=reviews[xbox_one_filter]
filtered_reviews2.head()
```

Out[58]:

	score_phrase	title	url	platform	score	genre	editors_choice	release_
17137	Amazing	Gone Home	/games/gone-home/xbox-one-20014361	Xbox One	9.5	Simulation	Y	2
17197	Amazing	Rayman Legends	/games/rayman-legends/xbox-one-20008449	Xbox One	9.5	Platformer	Y	2
17295	Amazing	LEGO Marvel Super Heroes	/games/lego-marvel-super-heroes/xbox-one-20000826	Xbox One	9.0	Action	Y	2
17313	Great	Dead Rising 3	/games/dead-rising-3/xbox-one-124306	Xbox One	8.3	Action	N	2
17317	Great	Killer Instinct	/games/killer-instinct-2013/xbox-one-20000538	Xbox One	8.4	Fighting	N	2

In [59]:
```
# What is the size of filtered_reviews2
filtered_reviews2.shape
```

Out[59]:
```
(140, 10)
```

In [60]:
```
#Select all video games which are 'Action' genre
action_reviews=reviews[reviews['genre']=='Action']
```

In [61]:
```
action_reviews.head()
```

Out[61]:

	score_phrase	title	url	platform	score	genre	editors_choice	release_year
17	Great	Avengers Initiative	/games/avengers-initiative/iphone-141579	iPhone	8.0	Action	N	2012
34	Good	War of the Roses	/games/war-of-the-roses-140577/pc-115849	PC	7.3	Action	N	2012
45	Amazing	Bad Piggies	/games/bad-piggies/iphone-141455	iPhone	9.2	Action	Y	2012
49	Okay	Demon's Score	/games/demons-score/iphone-118050	iPhone	6.9	Action	N	2012
69	Great	Hotline Miami	/games/hotline-miami/pc-139657	PC	8.8	Action	Y	2012

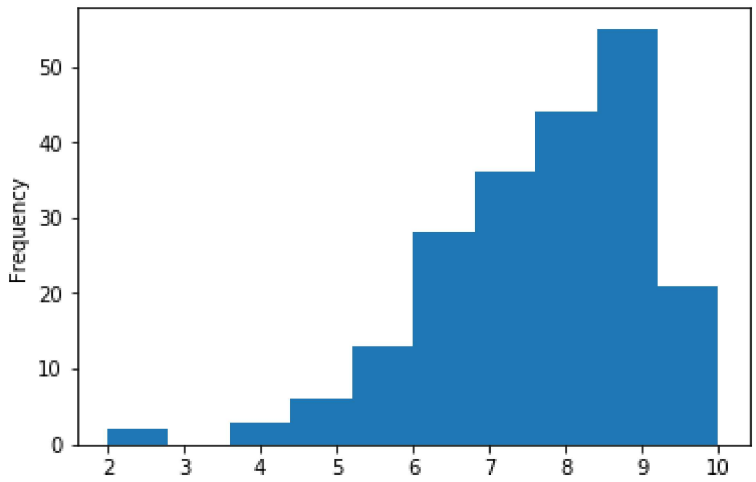
In [62]:
```
action_reviews.shape
```

Out[62]: (3797, 10)

In [63]:
```
## Import plotting libraries
#Plot Histogram for the frequencies of different score ranges of Xbox One platfor
import matplotlib.pyplot as plt
```

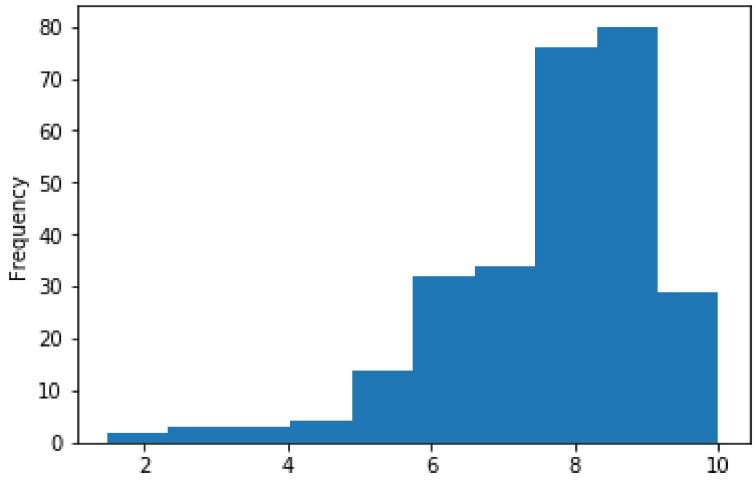
In [64]:
```
reviews[reviews["platform"]=="Xbox One"]["score"].plot(kind="hist")
```

Out[64]: <matplotlib.axes._subplots.AxesSubplot at 0x241effb4358>



In [65]:
```
#Plot Histogram for Frequencies of the scores of Play Station4 platform
reviews[reviews["platform"]=="PlayStation 4"]["score"].plot(kind="hist")
```

Out[65]: <matplotlib.axes._subplots.AxesSubplot at 0x241effd6630>



In [ ]: