Page No.
Date

Name:- Rohini Janardan Devkar
Roll no:- 23272
PRN no:- 72030818G
TE-2 (comp)
DSBDA Lab

Practical No:- 2.
Data Wrangling II

* Aim:-

Data Wrangling II :-
Perform the following operations using Python on any source dataset (e.g. data.csv)

1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them.
2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them.
3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution.

Reason and document your approach properly.

* Pre-requisite:-

1. Basic of python programming
2. Concept of data preprocessing, data formatting, data normalization, and data cleaning.

Page No.
Date

* Theory :-

1. Creation of Dataset using Microsoft Excel :-

The dataset is created in "CSV" format.

- The name of dataset is StudentsPerformance.
- The features of the dataset are :- ~~Math_Score~~ Attendance, Phy-marks, Che-marks, EM1_marks, PPS-marks, SME-marks, Rollno, Name, Gender, Branch, Total marks, percentage.
- Number of Instances :- 200
- The response variable is :- Percentage.
- Range of Values :-

Phy_marks [-100, 100], Che-marks [-100, 100], EM1-marks [-100, 100], PPS-marks [-100, 100], SME-marks [-100, 100], Total marks [-500, 500].

- The response variable is the number of placement offers facilitated to particular students, which is largely depend on Percentage.

The step to create the dataset are as follows :-

Step1 :- Open Microsoft Excel and click on SaveAs. Select other. Formats.

Step 2 : Enter the name of the dataset and save the dataset astype CSV(MS-DOS).

step3 : Enter the name of features as column header.

Step 4 : Fill the data by using RANDOMBETWEEN function. For every Feature, fill the data by considering above specified range.

Step 5 : In 20% data, fill the impurities.

Step 6 : To violate the ruleof response variable, update fea values.

The dataset is created with the given description.

2. Identification and Handling of Null Values :-

Missing data can occur when no information is provided for one or more items or for a whole unit. Missing data can also refer to as NA values in pandas.

In pandas missing data is represented by two value :

1. None :- None is a python singleton object that is often used for missing data in python code.
2. NaN :- NaN is a special floating-point value recognized by all systems that use the standard IEEE floating-point representation.

Pandas treat None and NaN as essentially interchangeable for indicating missing or null values.

1. Checking for missing values using isnull() and notnull() :-

- Checking for missing values using isnull() :-

In order to check null values in pandas Dataframe, isnull() function is used. This function return dataframe of Boolean values which are True for NaN values.

- Checking for missing values using notnull() :-

In order to check null values in pandas dataframe, notnull() function is used. This function return dataframe of Boolean values which are False for NaN values.

3. Identification and Handling of outliers :-

1) Identification of outliers :-

One of the most important steps as part of data preprocessing is detecting and treating the outliers as they can negatively affect the statistical analysis and the training process of a machine learning algorithm resulting in lower accuracy.

2) Handling of outliers:-

For removing the outlier, one must follow the same process of removing an entry from the dataset using its exact position in the dataset because in all the above methods of detecting the outliers end result is the list of all those data items that satisfy the outlier definition according to the method used.

4. Data transformation for the purpose of:

Data transformation is the process of converting raw data into a format or structure that would be more suitable for model building and also data discovery in general. The process of data transformation can also be referred to as extract / transform / load (ETL). The data transformation involves steps that are:-

1) Smoothing:- It is a process that is used to remove noise from the dataset using some algorithms.

2) Aggregation:- It is a method of storing and presenting data in a summary format.

3) Generalization:- It converts low-level data attributes to high-level data attributes using concept-hierarchy.

4) Normalization:- Data normalization involves converting all data variables into a given range. There are some techniques:-

1) min-max normalization:-

2) Z-score normalization:.

3) Normalization by decimal scaling:-

5) Attribute or feature construction:.

New attribute constructed from the given ones.

Page No.
Date

4.1. To change the scale for better understanding.

4.2. To decrease the skewness and convert distribution into normal distribution.

Assignment 2 Data Wrangling II

Importing pandas and numpy libs

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Reading the dataset and loading into pandas dataframe

In [2]:
```
df=pd.read_csv("31456_Dataset(Academic Performance).csv")
```

Displaying the first 20 rows using head() function

In [3]:
```
df.head(20)
```

Out[3]:

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_mar
0	1	Mohammed	M	Comp	72.0	62.0	98.0	63.0	89
1	2	Reyansh	M	IT	58.0	62.0	83.0	83.0	88
2	3	Aarav	M	IT	57.0	-20.0	100.0	NaN	56
3	4	Atharv	M	IT	60.0	89.0	83.0	70.0	33
4	5	Vivaan	M	Comp	85.0	90.0	NaN	78.0	23
5	6	Advik	M	ENTC	94.0	99.0	84.0	100.0	56
6	7	Ansh	M	ENTC	98.0	88.0	95.0	81.0	78
7	8	Ishaan	M	ENTC	75.0	66.0	66.0	83.0	-99
8	9	Dhruv	M	ENTC	63.0	NaN	NaN	97.0	56
9	10	Siddharth	M	ENTC	96.0	67.0	78.0	95.0	Na
10	11	Vihaan	M	ENTC	82.0	54.0	70.0	88.0	55
11	12	Arjun	M	IT	75.0	64.0	67.0	71.0	66
12	13	Aarush	M	IT	67.0	56.0	81.0	NaN	90
13	14	Leo	M	IT	98.0	-34.0	70.0	94.0	77
14	15	Maryam	F	IT	64.0	87.0	60.0	90.0	65
15	16	Saanvi	F	Comp	66.0	90.0	95.0	67.0	99
16	17	Zaranew	F	Comp	93.0	54.0	NaN	75.0	90
17	18	Inaya	F	Comp	74.0	67.0	93.0	93.0	87
18	19	Aarya	F	Comp	72.0	88.0	84.0	81.0	80
19	20	Pari	F	Comp	53.0	76.0	81.0	93.0	65

Displaying the number of rows and columns in the dataset using shape

In [4]:
```
df.shape
```

Out[4]:
```
(200, 12)
```

Displaying the data type of each column in the data set using dtypes

In [5]:
```
df.dtypes.value_counts()
```

Out[5]:
```
float64    7
object     3
int64      2
dtype: int64
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Rollno       200 non-null    int64
 1   Name         200 non-null    object
 2   Gender       200 non-null    object
 3   Branch       200 non-null    object
 4   Attendence   195 non-null    float64
 5   Phy_marks    194 non-null    float64
 6   Che_marks    188 non-null    float64
 7   EM1_marks    193 non-null    float64
 8   PPS_marks    195 non-null    float64
 9   SME_marks    199 non-null    float64
 10  Total Marks  200 non-null    int64
 11  Percentage   200 non-null    float64
dtypes: float64(7), int64(2), object(3)
memory usage: 18.9+ KB
```

In [7]:
```
df.describe()
```

Out[7]:

	Rollno	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_marks	SME_marks	Tot
count	200.000000	195.000000	194.000000	188.000000	193.000000	195.000000	199.000000	20
mean	100.500000	80.861538	74.309278	79.047872	86.756477	81.241026	67.723618	37
std	57.879185	86.095286	89.061818	18.144779	103.421356	117.895368	50.536905	18
min	1.000000	-10.000000	-55.000000	-69.000000	-34.000000	-99.000000	-88.000000	11
25%	50.750000	62.000000	55.000000	71.000000	69.000000	56.000000	55.000000	31
50%	100.500000	73.000000	70.000000	81.000000	81.000000	77.000000	67.000000	35
75%	150.250000	90.000000	88.750000	90.000000	90.000000	89.500000	87.500000	39
max	200.000000	1250.000000	1000.000000	100.000000	1500.000000	1111.000000	444.000000	184

Handle the Missing value

Detect the missing value

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Rollno          0
Name            0
Gender          0
Branch          0
Attendence      5
Phy_marks       6
Che_marks      12
EM1_marks       7
PPS_marks       5
SME_marks       1
Total Marks     0
Percentage      0
dtype: int64
```

Make a list of column having missing value

In [9]:
```
data=df
coln=[]
miss=[]
coln.extend(data.columns)
#print(miss)
for i in coln:
    t=data[i].isnull().sum()
    if t!=0:
        miss.append(i)
    else:
        continue
print(miss)
```

```
['Attendence', 'Phy_marks', 'Che_marks', 'EM1_marks', 'PPS_marks', 'SME_marks']
```

Removal of out of bound element

In [10]:
```
pd.options.mode.chained_assignment = None
for j in miss:
    q=data[j].dtypes
    if (q=='int64' or q=='float64') :
        f=data[j]
        for k in range(data.shape[0]):
            if (f[k]<0 or f[k]>100) :
                f[k]=(np.nan)
    else:
        continue
```

fill the missing value using mean for float and int datatypes and for other backword fill

In [11]:
```
for j in miss:
    q=data[j].dtypes
    if (q=='int64' or q=='float64') :
        data[j].fillna((data[j].mean()),inplace=True)
    else:
        data.fillna(method='bfill')

data.head(10)
```

Out[11]:

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_mark
0	1	Mohammed	M	Comp	72.0	62.000000	98.000000	63.000000	89.00000

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_mark
1	2	Reyansh	M	IT	58.0	62.000000	83.000000	83.000000	88.00000(
2	3	Aarav	M	IT	57.0	70.854749	100.000000	79.989529	56.00000(
3	4	Atharv	M	IT	60.0	89.000000	83.000000	70.000000	33.00000(
4	5	Vivaan	M	Comp	85.0	90.000000	79.839572	78.000000	23.00000(
5	6	Advik	M	ENTC	94.0	99.000000	84.000000	100.000000	56.00000(
6	7	Ansh	M	ENTC	98.0	88.000000	95.000000	81.000000	78.00000(
7	8	Ishaan	M	ENTC	75.0	66.000000	66.000000	83.000000	72.75409(
8	9	Dhruv	M	ENTC	63.0	70.854749	79.839572	97.000000	56.00000(
9	10	Siddharth	M	ENTC	96.0	67.000000	78.000000	95.000000	72.75409(

Correction in Total Marks, Percentage after filling missing value

In [12]:
```
data['Total Marks']=data['Phy_marks']+data['Che_marks']+data['EM1_marks']+data['PPS_
data['Percentage']=data['Total Marks']/5
```

In [13]:
```
data
```

Out[13]:

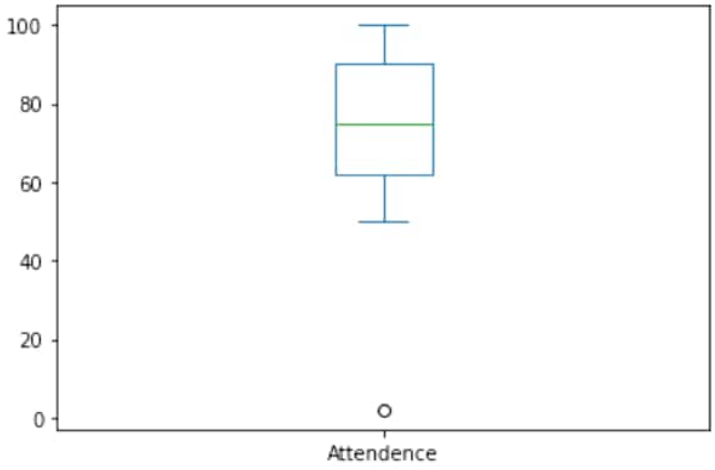
	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_m
0	1	Mohammed	M	Comp	72.0	62.000000	98.000000	63.000000	89.00(
1	2	Reyansh	M	IT	58.0	62.000000	83.000000	83.000000	88.00(
2	3	Aarav	M	IT	57.0	70.854749	100.000000	79.989529	56.00(
3	4	Atharv	M	IT	60.0	89.000000	83.000000	70.000000	33.00(
4	5	Vivaan	M	Comp	85.0	90.000000	79.839572	78.000000	23.00(
...	...	...	...	...	...	...	...	...	
195	196	Meghanew	F	ENTC	69.0	97.000000	68.000000	78.000000	85.00(
196	197	Thendralnew	F	IT	77.0	90.000000	78.000000	88.000000	72.754
197	198	Pranavinew	F	ENTC	67.0	77.000000	89.000000	83.000000	90.00(
198	199	Gunjannew	F	IT	82.0	62.000000	100.000000	93.000000	77.00(
199	200	Dhatrinew	F	Comp	69.0	62.000000	80.000000	90.000000	62.00(

200 rows × 12 columns

OUTLIERS DETECTION

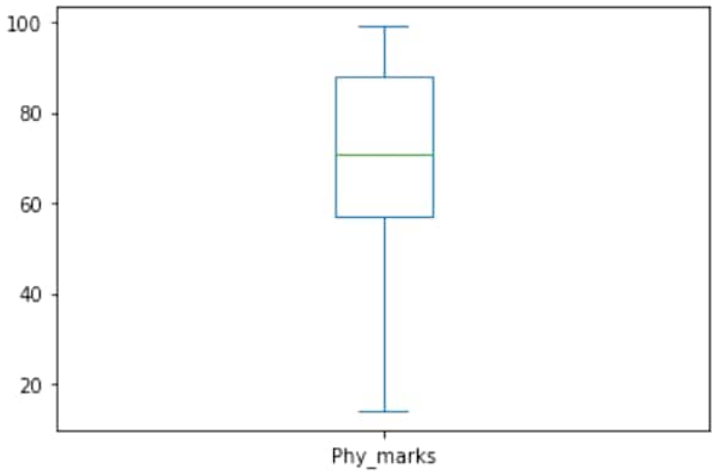
In [14]:
```
data['Attendence'].plot(kind='box')
```

Out[14]: <AxesSubplot:>



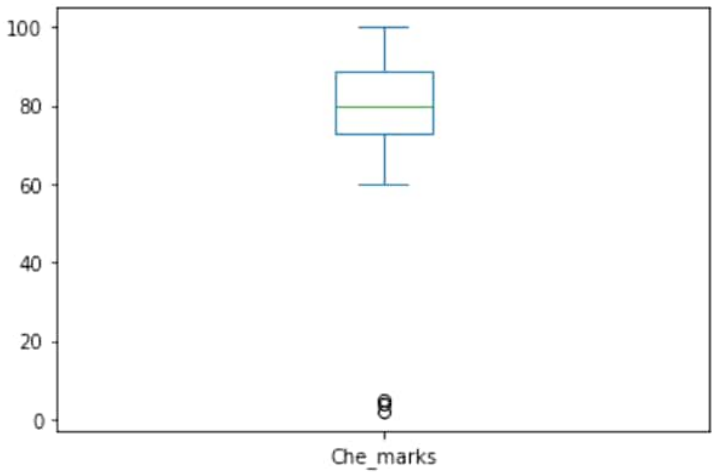
In [15]:
```
data['Phy_marks'].plot(kind='box')
```

Out[15]: <AxesSubplot:>



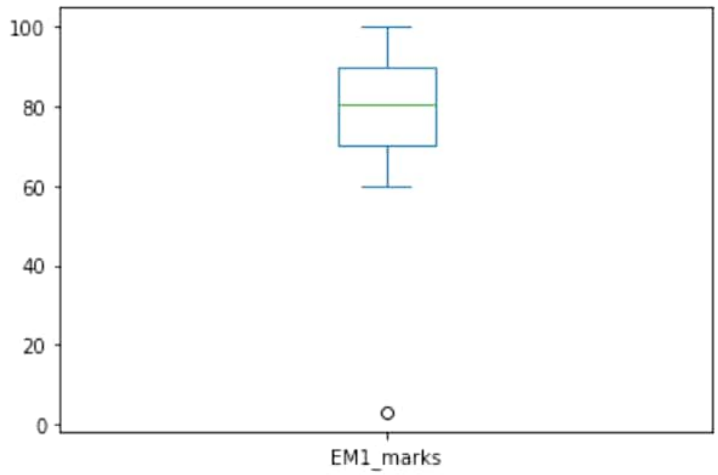
In [16]:
```
data['Che_marks'].plot(kind='box')
```

Out[16]: <AxesSubplot:>



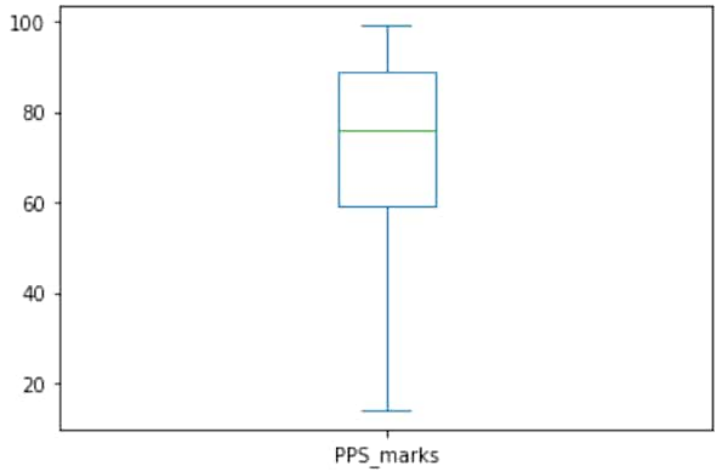
In [17]:
```
data['EM1_marks'].plot(kind='box')
```

Out[17]: <AxesSubplot:>



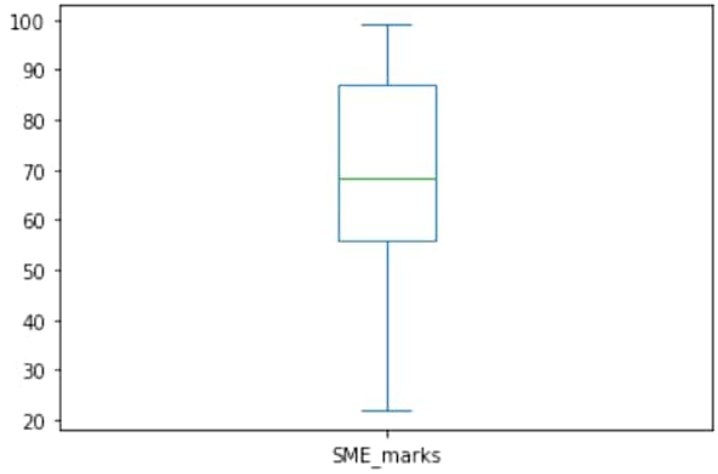
In [18]:
```
data['PPS_marks'].plot(kind='box')
```

Out[18]: <AxesSubplot:>



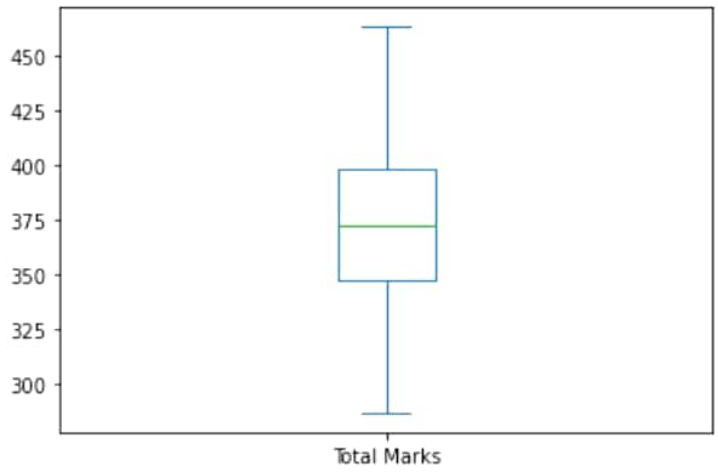
In [19]:
```
data['SME_marks'].plot(kind='box')
```

Out[19]: <AxesSubplot:>



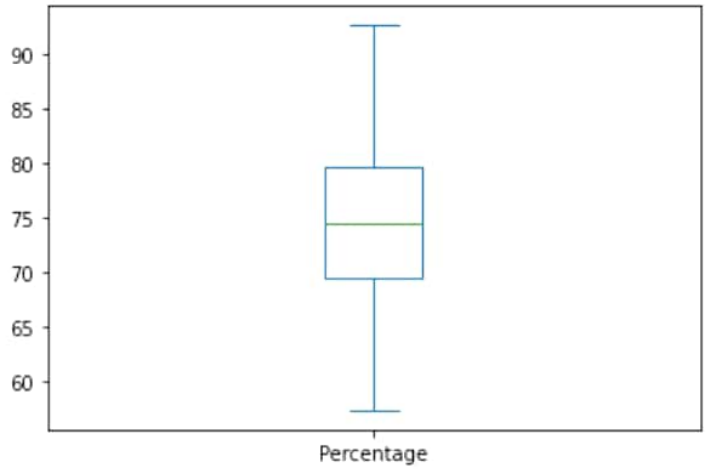
In [20]:
```
data['Total Marks'].plot(kind='box')
```

Out[20]: <AxesSubplot:>



In [21]:
```
data['Percentage'].plot(kind='box')
```

Out[21]: <AxesSubplot:>



OUTLIERS REMOVAL

REMOVAL OF OUTLIERS FROM ATTENDENCE

In [22]:
```
Q1 = data['Attendence'].quantile(0.25)
Q3 = data['Attendence'].quantile(0.75)
IQR = Q3 - Q1

Lower_limit = Q1 - 1.5*IQR
Upper_limit = Q3 + 1.5*IQR

print("Q1 :",Q1,"\nQ3 :",Q3,"\nIQR :",IQR,"\nLower_limit :",Lower_limit,"\nUpper_lim
```

```
Q1 : 62.0
Q3 : 90.0
IQR : 28.0
Lower_limit : 20.0
Upper_limit : 132.0
```

In [23]:
```
data[(data['Attendence']<Lower_limit)|(data['Attendence']>Upper_limit)]
```

Out[23]:

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_marks	S

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_marks	S
69	70	Daniel	M	ENTC	2.0	90.0	96.0	78.0	88.0	

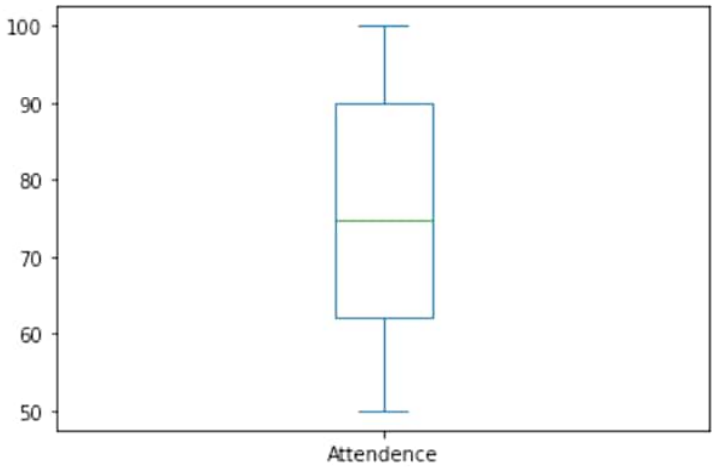
In [24]:
```
data=data[(data['Attendence']>Lower_limit)&(data['Attendence']<Upper_limit)]
data[60:70]
```

Out[24]:

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_mark
60	61	Mehernew	F	Comp	96.0	85.000000	64.000000	94.0	33.(
61	62	Aradhya	F	IT	98.0	83.000000	66.000000	99.0	66.(
62	63	Aanya	F	IT	57.0	76.000000	79.839572	100.0	88.(
63	64	Prisha	F	Comp	58.0	70.854749	73.000000	87.0	99.(
64	65	Pranav	M	ENTC	100.0	34.000000	97.000000	88.0	65.(
65	66	David	M	IT	60.0	56.000000	78.000000	76.0	76.(
66	67	Ahaan	M	ENTC	89.0	23.000000	85.000000	99.0	32.(
67	68	Noah	M	IT	94.0	56.000000	74.000000	100.0	45.(
68	69	Jay	M	Comp	54.0	99.000000	66.000000	85.0	78.(
70	71	Aaron	M	ENTC	78.0	70.854749	89.000000	98.0	99.(

In [25]:
```
data['Attendence'].plot(kind='box')
```

Out[25]: <AxesSubplot:>



REMOVAL OF OUTLIERS FROM Che_marks

In [26]:
```
Q1 = data['Che_marks'].quantile(0.25)
Q3 = data['Che_marks'].quantile(0.75)
IQR = Q3 - Q1

Lower_limit = Q1 - 1.5*IQR
Upper_limit = Q3 + 1.5*IQR
```

```
print("Q1 :",Q1,"\nQ3 :",Q3,"\nIQR :",IQR,"\nLower_limit :",Lower_limit,"\nUpper_lim
```

```
Q1 : 73.0
Q3 : 89.0
IQR : 16.0
Lower_limit : 49.0
Upper_limit : 113.0
```

In [27]:

```
data[(data['Che_marks']<Lower_limit)|(data['Che_marks']>Upper_limit)]
```

Out[27]:

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_m
144	145	Shivanyanew	F	COMP	53.0	77.0	2.0	87.0	
168	169	Shreshtha	F	ENTC	65.0	99.0	4.0	64.0	
185	186	Avanew	F	IT	97.0	87.0	5.0	65.0	

In [28]:

```
data=data[(data['Che_marks']>Lower_limit)&(data['Che_marks']<Upper_limit)]
data[140:170]
```

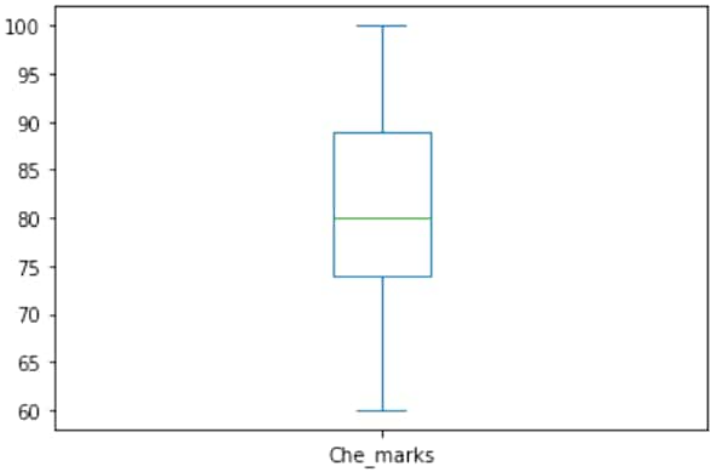
Out[28]:

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_r
141	142	Shivaninew	F	IT	52.000000	85.000000	65.000000	97.0	54.0(
142	143	Norah	F	COMP	71.000000	97.000000	77.000000	87.0	85.0(
143	144	Nakshatranew	F	ENTC	69.000000	90.000000	91.000000	63.0	97.0(
145	146	Noor	F	COMP	50.000000	62.000000	76.000000	80.0	77.0(
146	147	Mishka	F	Comp	95.000000	62.000000	90.000000	67.0	62.0(
147	148	Hannah	F	Comp	88.000000	33.000000	87.000000	87.0	62.0(
148	149	Surya	M	Comp	55.000000	44.000000	83.000000	64.0	75.0(
149	150	Neil	M	Comp	90.000000	78.000000	98.000000	84.0	89.0(
150	151	Yug	M	Comp	98.000000	14.000000	92.000000	66.0	72.7!
151	152	Shlok	M	Comp	94.000000	70.854749	88.000000	81.0	99.0(
152	153	Shaashwat	M	Comp	74.885417	70.854749	80.000000	98.0	88.0(
153	154	Parth	M	Comp	59.000000	96.000000	92.000000	3.0	66.0(
154	155	Luke	M	Comp	87.000000	90.000000	78.000000	88.0	98.0(
155	156	Armaan	M	ENTC	94.000000	87.000000	75.000000	69.0	96.0(
156	157	Aayush	M	IT	78.000000	80.000000	79.839572	60.0	90.0(
157	158	Mivaan	M	ENTC	58.000000	44.000000	66.000000	95.0	87.0(
158	159	Pranit	M	IT	98.000000	76.000000	92.000000	80.0	44.0(
159	160	Ved-	M	IT	84.000000	55.000000	69.000000	93.0	72.7!
160	161	Nivaan	M	ENTC	60.000000	50.000000	78.000000	77.0	99.0(
161	162	Sananew	F	ENTC	79.000000	33.000000	91.000000	94.0	54.0(

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_r
162	163	Zairanew	F	IT	62.000000	55.000000	93.000000	82.0	34.0(
163	164	Ishaninew	F	IT	58.000000	77.000000	83.000000	76.0	67.0(
164	165	Sara	F	ENTC	94.000000	70.854749	77.000000	97.0	34.0(
165	166	Aarohi	F	ENTC	68.000000	55.000000	76.000000	67.0	67.0(
166	167	Yashikanew	F	ENTC	59.000000	88.000000	94.000000	97.0	99.0(
167	168	Aira	F	IT	72.000000	22.000000	90.000000	90.0	32.0(
169	170	Michellenew	F	ENTC	76.000000	54.000000	66.000000	64.0	78.0(
170	171	Scarlettnew	F	Comp	98.000000	34.000000	99.000000	65.0	88.0(
171	172	Srishtinew	F	Comp	64.000000	62.000000	99.000000	77.0	89.0(
172	173	Garginew	F	Comp	83.000000	62.000000	63.000000	70.0	88.0(

In [29]:
```
data['Che_marks'].plot(kind='box')
```

Out[29]: <AxesSubplot:>



REMOVAL OF OUTLIERS FROM EM1 marks

In [30]:
```
Q1 = data['EM1_marks'].quantile(0.25)
Q3 = data['EM1_marks'].quantile(0.75)
IQR = Q3 - Q1

Lower_limit = Q1 - 1.5*IQR
Upper_limit = Q3 + 1.5*IQR

print("Q1 :",Q1,"\nQ3 :",Q3,"\nIQR :",IQR,"\nLower_limit :",Lower_limit,"\nUpper_lim
```

```
Q1 : 70.0
Q3 : 90.0
IQR : 20.0
Lower_limit : 40.0
Upper_limit : 120.0
```

In [31]:
```
data[(data['EM1_marks']<Lower_limit)|(data['EM1_marks']>Upper_limit)]
```

Out[31]:

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_marks
153	154	Parth	M	Comp	59.0	96.0	92.0	3.0	66.0

In [32]:

```
data=data[(data['EM1_marks']>Lower_limit)&(data['EM1_marks']<Upper_limit)]
data[150:160]
```

Out[32]:

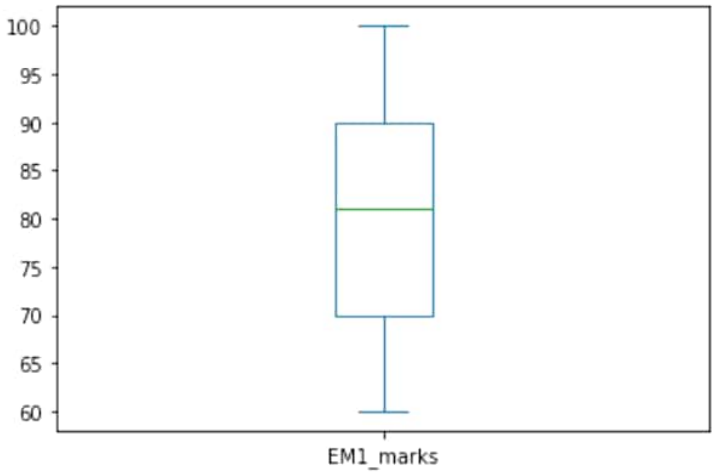
	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_marl
152	153	Shaashwat	M	Comp	74.885417	70.854749	80.000000	98.0	88.0000(
154	155	Luke	M	Comp	87.000000	90.000000	78.000000	88.0	98.0000(
155	156	Armaan	M	ENTC	94.000000	87.000000	75.000000	69.0	96.0000(
156	157	Aayush	M	IT	78.000000	80.000000	79.839572	60.0	90.0000(
157	158	Mivaan	M	ENTC	58.000000	44.000000	66.000000	95.0	87.0000(
158	159	Pranit	M	IT	98.000000	76.000000	92.000000	80.0	44.0000(
159	160	Ved-	M	IT	84.000000	55.000000	69.000000	93.0	72.7540!
160	161	Nivaan	M	ENTC	60.000000	50.000000	78.000000	77.0	99.0000(
161	162	Sananew	F	ENTC	79.000000	33.000000	91.000000	94.0	54.0000(
162	163	Zairanew	F	IT	62.000000	55.000000	93.000000	82.0	34.0000(

In [33]:

```
data['EM1_marks'].plot(kind='box')
```

Out[33]:

```
<AxesSubplot:>
```



BINNING USING FREQUENCY

In [34]:

```
def BinningFunction(column,cut_points,labels=None):
    minper=column.min()
    maxper=column.max()
    break_points=[minper]+cut_points+[maxper]
    print("Gradding According to percentage \n>60 = F \n60-70 = B \n70-80 = A\n80-10
```

```
    t=pd.cut(column,bins=break_points,labels=labels,include_lowest=True)
    return t
```

In [35]:

```
cut_points=[60,70,80]
labels=['F','B','A','O']
data["Grade"]=BinningFunction(data['Percentage'],cut_points,labels)
data
```

```
Gradding According to percentage
>60 = F
60-70 = B
70-80 = A
80-100 = O
```

Out[35]:

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_m
0	1	Mohammed	M	Comp	72.0	62.000000	98.000000	63.000000	89.00(
1	2	Reyansh	M	IT	58.0	62.000000	83.000000	83.000000	88.00(
2	3	Aarav	M	IT	57.0	70.854749	100.000000	79.989529	56.00(
3	4	Atharv	M	IT	60.0	89.000000	83.000000	70.000000	33.00(
4	5	Vivaan	M	Comp	85.0	90.000000	79.839572	78.000000	23.00(
...	...	...	...	...	...	...	...	...	
195	196	Meghanew	F	ENTC	69.0	97.000000	68.000000	78.000000	85.00(
196	197	Thendralnew	F	IT	77.0	90.000000	78.000000	88.000000	72.754
197	198	Pranavinew	F	ENTC	67.0	77.000000	89.000000	83.000000	90.00(
198	199	Gunjannew	F	IT	82.0	62.000000	100.000000	93.000000	77.00(
199	200	Dhatrinew	F	Comp	69.0	62.000000	80.000000	90.000000	62.00(

195 rows × 13 columns

MIN MAX SCALING

In [36]:

```
new_data1=data
from sklearn.preprocessing import MinMaxScaler
```

In [37]:

```
scaler = MinMaxScaler()
```

In [38]:

```
column=['Attendence','Phy_marks','Che_marks','EM1_marks','PPS_marks','SME_marks','To
scaler.fit(new_data1[column])
new_data1[column] = scaler.transform(new_data1[column])
```

In [39]:

```
new_data1
```

Out[39]:

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_m
0	1	Mohammed	M	Comp	0.44	0.564706	0.950000	0.075000	0.882

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_m
1	2	Reyansh	M	IT	0.16	0.564706	0.575000	0.575000	0.87(
2	3	Aarav	M	IT	0.14	0.668879	1.000000	0.499738	0.49⁴
3	4	Atharv	M	IT	0.20	0.882353	0.575000	0.250000	0.22³
4	5	Vivaan	M	Comp	0.70	0.894118	0.495989	0.450000	0.10⁵
...	...	...	...	...	...	...	...	...	
195	196	Meghanew	F	ENTC	0.38	0.976471	0.200000	0.450000	0.83⁵
196	197	Thendralnew	F	IT	0.54	0.894118	0.450000	0.700000	0.69¹
197	198	Pranavinew	F	ENTC	0.34	0.741176	0.725000	0.575000	0.89⁴
198	199	Gunjannew	F	IT	0.64	0.564706	1.000000	0.825000	0.74¹
199	200	Dhatrinew	F	Comp	0.38	0.564706	0.500000	0.750000	0.56⁴

195 rows × 13 columns

Z-SCORE SCALING

In [40]:
```
new_data2=data
```

In [41]:
```
from scipy import stats
column=['Attendence','Phy_marks','Che_marks','EM1_marks','PPS_marks','SME_marks','To
new_data2[column] = stats.zscore(new_data2[column])
new_data2
```

Out[41]:

	Rollno	Name	Gender	Branch	Attendence	Phy_marks	Che_marks	EM1_marks	PPS_m
0	1	Mohammed	M	Comp	-0.224299	-0.422745	1.631550	-1.501918	0.80⁷
1	2	Reyansh	M	IT	-1.150562	-0.422745	0.203187	0.212848	0.75⁷
2	3	Aarav	M	IT	-1.216723	0.024538	1.821999	-0.045264	-0.83⁷
3	4	Atharv	M	IT	-1.018238	0.941115	0.203187	-0.901750	-1.98⁴
4	5	Vivaan	M	Comp	0.635802	0.991628	-0.097762	-0.215843	-2.48²
...	...	...	...	...	...	...	...	...	
195	196	Meghanew	F	ENTC	-0.422784	1.345221	-1.225177	-0.215843	0.60⁷
196	197	Thendralnew	F	IT	0.106509	0.991628	-0.272934	0.641540	-0.00²
197	198	Pranavinew	F	ENTC	-0.555107	0.334955	0.774532	0.212848	0.85⁷
198	199	Gunjannew	F	IT	0.437317	-0.422745	1.821999	1.070232	0.20⁹
199	200	Dhatrinew	F	Comp	-0.422784	-0.422745	-0.082486	0.813017	-0.53⁸

195 rows × 13 columns

Page No.
Date

Conclusion :-

In this way we have explored the functions of the python library for data identifying and handling the outliers. Data transformations techniques are explored with the purpose of creating the new variable and reducing the skewness from datasets.