```
import pandas as pd
import numpy as np
missing values = ["Na", "na"]
pd.read_excel("/home/ubuntu/TCOB35/Student_Performance.xlsx",na_values
= missing values)
df
    math_score reading_score writing_score placement_score
club_join_year
                          73.0
                                          69.0
            77
                                                           65.0
2020
            75
                          70.0
                                          58.0
                                                           94.0
1
2018
            75
                          63.0
                                          61.0
                                                           69.0
2019
            69
                          70.0
                                          71.0
                                                           73.0
2019
            97
                          50.0
                                          64.0
                                                           77.0
2018
            78
                          75.0
                                          62.0
                                                           78.0
2019
            77
                          71.0
                                          74.0
                                                           66.0
6
2019
            66
                          70.0
                                           NaN
                                                           77.0
2019
                                          69.0
            70
                          67.0
                                                           62.0
2019
            55
                          20.0
                                          74.0
                                                           75.0
2021
            73
                          77.0
                                          77.0
                                                           60.0
10
2018
            64
                          60.0
                                          73.0
                                                           75.0
11
2021
            73
                          60.0
                                          73.0
                                                           95.0
12
2021
            72
13
                          71.0
                                          61.0
                                                           80.0
2019
            74
                                                           70.0
14
                          69.0
                                          68.0
2021
15
            73
                          63.0
                                          67.0
                                                            NaN
2018
16
            65
                          86.0
                                          61.0
                                                           77.0
2019
            87
                          61.0
                                          90.0
                                                           77.0
17
2018
18
            71
                          69.0
                                          78.0
                                                             NaN
2019
```

19	67	77.0	69.0	79.0
2020 20	62	NaN	78.0	75.0
2021				
21 2018	67	68.0	64.0	64.0
22	66	69.0	72.0	79.0
2021 23	63	62.0	61.0	77.0
2019				
24 2018	67	76.0	70.0	65.0
25	55	74.0	66.0	77.0
2020 26	96	60.0	20.0	69.0
2021	90	00.0	20.0	09.0
27	75	79.0	73.0	80.0
2019 28	70	79.0	57.0	63.0
2019	70			
29 2020	70	74.0	75.0	69.0
Place 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24	ment-offe	r-count 2 0 2 0 2 0 0 2 0 2 0 2 0 0 2 0 0 2 0 0 2 0 0 2 0 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2 0 0 2 0 2		

25 26 27 28 29		0 2 0 2 2		
series df[ser		ull(df["placeme	nt_score"])	
	th_score oin_year 73	reading_score \ 63.0	writing_score 67.0	placement_score
2018	13	03.0	07.0	ivaiv
18 2019	71	69.0	78.0	NaN
Pl: 15 18	acement-o	ffer-count 0 2		
df.not	null()			
	th_score oin_year	\		placement_score
0 True	True	True	True	True
1 True	True	True	True	True
2 True	True	True	True	True
3 True	True	True	True	True
4 True	True	True	True	True
5 True	True	True	True	True
6 True	True	True	True	True
7 True	True	True	False	True
8 True	True	True	True	True
9 True	True	True	True	True
10 True	True	True	True	True
11 True	True	True	True	True
12 True	True	True	True	True

13	True	True	True	True	
True 14	True	True	True	True	
True		11.00		1140	
15	True	True	True	False	
True 16	True	True	True	True	
True	True	True	True	True	
17	True	True	True	True	
True	_	_	_		
18 True	True	True	True	False	
True 19	True	True	True	True	
True	1146	1140	TTUC	Truc	
20	True	False	True	True	
True	_	_	_	_	
21	True	True	True	True	
True 22	Truo	Truo	Truo	Truo	
True	True	True	True	True	
23	True	True	True	True	
True					
24	True	True	True	True	
True	_	_	_	_	
25 Taura	True	True	True	True	
True 26	True	True	True	True	
True	TTUE	True	TTUE	True	
27	True	True	True	True	
True					
28	True	True	True	True	
True	T	Tour	T	T ::	
29 True	True	True	True	True	
TTUE					
Pla	acement-offer	^-count			
0		True			
1		True			
2		True			
3 1		True True			
5		True			
1 2 3 4 5 6		True			
7		True			
8		True			
9		True			
10 11		True True			
12		True			
12		1140			

13	True
14	True
15	True
16	True
17	True
18	True
19	True
20	True
21	True
22	True
23	True
24	True
25	True
26	True
27	True
28	True
29	True
series1 = nd notnull	(df["math score"])

series1 = pd.notnull(df["math\_score"])
df[series1]

			writing_score	placement_score
0	_join_year 77	73.0	69.0	65.0
2020				
1	75	70.0	58.0	94.0
2018		62.0	61.0	60.0
2010	75	63.0	61.0	69.0
2019 3	69	70.0	71.0	73.0
2019		70.0	71.0	73.0
4	97	50.0	64.0	77.0
2018		30.0	0410	7710
5	78	75.0	62.0	78.0
2019		,510	02.0	7515
6	77	71.0	74.0	66.0
2019				
7	66	70.0	NaN	77.0
2019				
8	70	67.0	69.0	62.0
2019				
9	55	20.0	74.0	75.0
2021				
10	73	77.0	77.0	60.0
2018		CO 0	72.0	75.0
11	64	60.0	73.0	75.0
2021 12	73	60.0	72 A	95.0
2021		60.0	73.0	95.0
13	72	71.0	61.0	80.0
15	12	/1.0	01.0	30.0

2010					
2019 14	74	69.0	68.0	70.0	
2021	7 -	03.0	00.0	7010	
15	73	63.0	67.0	NaN	
2018					
16	65	86.0	61.0	77.0	
2019 17	87	61.0	90.0	77.0	
2018	07	01.0	3010	7710	
18	71	69.0	78.0	NaN	
2019	67	77.0	60.0	70.0	
19	67	77.0	69.0	79.0	
2020 20	62	NaN	78.0	75.0	
2021	02	Nan	70.0	75.0	
21	67	68.0	64.0	64.0	
2018					
22	66	69.0	72.0	79.0	
2021 23	63	62.0	61.0	77.0	
2019	05	02.0	01.0	77.0	
24	67	76.0	70.0	65.0	
2018					
25	55	74.0	66.0	77.0	
2020 26	96	60.0	20.0	69.0	
2021	30	00.0	20.0	03.0	
27	75	79.0	73.0	80.0	
2019	70	70.0		62.0	
28 2019	70	79.0	57.0	63.0	
2019	70	74.0	75.0	69.0	
2020	, 0	, 110	7310	0310	
	ement-offer				
1		2 0			
2		0 2 2 0 0			
3		2			
4		0			
5		0			
7		2			
8		2			
0 1 2 3 4 5 6 7 8 9		0 2 0 2			
11 12		0 0			
13		0			
13		<b>J</b>			

14 15 16 17	2 0 0
18	2
19	0
20	0
21	2
22	0
23	0
24	2
25	0
26	2
27	0
28	2
29	2

ndf=df
ndf.fillna(0)

math_score	reading_score	writing_score	placement_score
club_join_year	\		
0 77	73.0	69.0	65.0
2020			
1 75	70.0	58.0	94.0
2018			
2 75	63.0	61.0	69.0
2019			
3 69	70.0	71.0	73.0
2019			
4 97	50.0	64.0	77.0
2018			
5 78	75.0	62.0	78.0
2019			
6 77	71.0	74.0	66.0
2019			
7 66	70.0	0.0	77.0
2019			
8 70	67.0	69.0	62.0
2019			
9 55	20.0	74.0	75.0
2021			
10 73	77.0	77.0	60.0
2018			
11 64	60.0	73.0	75.0
2021			
12 73	60.0	73.0	95.0
2021			
13 72	71.0	61.0	80.0
2019			

14	74	69.0	68.0	70.0	
2021 15	73	63.0	67.0	0.0	
2018	65	96.0	61.0		
16 2019	65	86.0	61.0	77.0	
17	87	61.0	90.0	77.0	
2018 18	71	69.0	78.0	0.0	
2019	67	77.0	60.0		
19 2020	67	77.0	69.0	79.0	
20 2021	62	0.0	78.0	75.0	
21 2018	67	68.0	64.0	64.0	
22 2021	66	69.0	72.0	79.0	
23 2019	63	62.0	61.0	77.0	
24	67	76.0	70.0	65.0	
2018 25	55	74.0	66.0	77.0	
2020 26	96	60.0	20.0	69.0	
2021 27	75	79.0	73.0	80.0	
2019 28	70	79.0	57.0	63.0	
2019 29	70	74.0	75.0	69.0	
2020	70	7110	73.0	0310	
Place	ement-offer	-count			
Θ		2			
1 2 3 4 5 6 7 8 9		0			
3		2			
4		0			
5		0			
6		2			
8		2			
9		0			
10		2			
11		0 0			
12 13		0 2 2 0 0 2 0 2 0 2 0 0			
14		2			

```
15
                         0
16
                         0
17
                         0
                         2
18
19
                         0
20
                         0
                         2
21
22
                         0
23
                         0
                         2
24
25
                         0
                         2
26
27
                         0
                         2
28
29
                         2
# df['reading score'] =
df['reading score'].fillna(df['reading score'].mean())
# df['reading score'] =
df['reading score'].fillna(df['reading score'].median())
# df['reading score'] =
df['reading score'].fillna(df['reading score'].std())
# df['reading_score'] =
df['reading_score'].fillna(df['reading_score'].min())
df['reading score'] =
df['reading_score'].fillna(df['reading_score'].max())
df
                 reading score writing score placement score
    math score
club join year
            77
                          73.0
                                          69.0
                                                            65.0
2020
            75
                          70.0
                                          58.0
                                                            94.0
2018
            75
                          63.0
                                          61.0
                                                            69.0
2019
            69
                          70.0
                                          71.0
                                                            73.0
2019
            97
                          50.0
                                          64.0
                                                            77.0
2018
            78
                          75.0
                                          62.0
                                                            78.0
2019
            77
                          71.0
                                          74.0
                                                            66.0
2019
            66
                          70.0
                                           NaN
                                                            77.0
2019
            70
                          67.0
                                          69.0
                                                            62.0
2019
```

9		5	20.0	74.0	75.0
202 10	7.	3	77.0	77.0	60.0
201		4	60.0	72.0	75.0
11 202	6 <sub>4</sub>	4	60.0	73.0	75.0
12 202	7.	3	60.0	73.0	95.0
13	7.	2	71.0	61.0	80.0
201 14	9 7,	4	69.0	68.0	70.0
202					
15	7.	3	63.0	67.0	NaN
201		_	00.0	61.0	77.0
16	6.	5	86.0	61.0	77.0
201 17	9 8'	7	61.0	90.0	77.0
201		1	01.0	90.0	77.0
18	7	1	69.0	78.0	NaN
201		1	03.0	70.0	IVAIV
19	6	7	77.0	69.0	79.0
202					
20	6:	2	86.0	78.0	75.0
202					
21	6	7	68.0	64.0	64.0
201		_			
22	6	6	69.0	72.0	79.0
202		2	62.0	61.0	77.0
23 201	6	3	62.0	61.0	77.0
24	9 6	7	76.0	70.0	65.0
201		,	70.0	70.0	05.0
25	5.	5	74.0	66.0	77.0
202		_	-		
26	9	6	60.0	20.0	69.0
202					
27	7.	5	79.0	73.0	80.0
201		0	70.0	F7 0	62.0
28	7	O	79.0	57.0	63.0
201 29	9 7	۵	74.0	75.0	69.0
202		U	74.0	73.0	09.0
202	o .				
	Placement	-offer-count			
0		2			
1					
2		0 2 2			
0 1 2 3 4					
4		0			

```
5
6
7
                            0
                            2
8
                            2
9
                            0
                            2
10
                            0
11
12
                            0
                            0
13
                            2
14
                            0
15
                            0
16
17
                            0
                            2
0
18
19
                            0
20
                            2
21
22
                            0
                            0
23
                            2
24
25
                            0
26
                            2
27
                            0
                            2
28
                            2
29
m_v=df['placement_score'].mean()
df['placement_score'].fillna(value=m_v, inplace=True)
df
                   reading score writing score placement score
    math score
club join year
                                               69.0
                                                              65.000000
              77
                             73.0
0
2020
              75
                             70.0
                                               58.0
                                                              94.000000
1
2018
              75
                             63.0
                                               61.0
                                                              69.000000
2019
              69
                             70.0
                                               71.0
                                                              73.000000
2019
              97
                             50.0
                                               64.0
                                                              77.000000
2018
              78
                             75.0
                                               62.0
                                                              78.000000
2019
              77
                             71.0
                                               74.0
                                                              66.000000
2019
              66
                             70.0
                                                 NaN
                                                              77.000000
7
2019
                                               69.0
              70
                             67.0
                                                              62.000000
2019
```

9	55	20.0	74.0	75.000000
2021				
10	73	77.0	77.0	60.000000
2018 11	64	60.0	73.0	75.000000
2021	04	00.0	75.0	75.000000
12	73	60.0	73.0	95.000000
2021				
13	72	71.0	61.0	80.000000
2019 14	74	69.0	68.0	70.000000
2021	74	09.0	00.0	70.00000
15	73	63.0	67.0	73.821429
2018				
16	65	86.0	61.0	77.000000
2019 17	87	61.0	90.0	77.000000
2018	07	01.0	90.0	77.000000
18	71	69.0	78.0	73.821429
2019				
19	67	77.0	69.0	79.000000
2020	60	06.0	70.0	75 000000
20 2021	62	86.0	78.0	75.000000
21	67	68.0	64.0	64.000000
2018				
22	66	69.0	72.0	79.000000
2021	63	62.0	C1 0	77 000000
23 2019	63	62.0	61.0	77.000000
24	67	76.0	70.0	65.000000
2018				
25	55	74.0	66.0	77.000000
2020	0.0	60.0	20.0	60,000000
26 2021	96	60.0	20.0	69.000000
27	75	79.0	73.0	80.000000
2019				
28	70	79.0	57.0	63.000000
2019	70	74.0	75.0	60.00000
29	70	74.0	75.0	69.000000
2020				
Placeme	ent-offer-coun	it		
0		2		
1		0		
2		0 2 2		
0 1 2 3 4		0		
•		•		

```
5
                          0
6
                          2
7
                          0
8
                          2
9
                          0
10
                          2
                          0
11
12
                          0
                          0
13
                          2
14
                          0
15
                          0
16
17
                          0
                          2
18
                          0
19
20
                          0
                          2
21
22
                          0
23
                          0
                          2
24
25
                          0
26
                          2
27
                          0
                          2
28
                          2
29
ndf1=df
ndf1.replace(to_replace = np.nan, value = -99)
ndf1=ndf1.replace(to_replace = 77, value = np.nan)
ndf1
    math_score
                 reading score writing score placement score
club_join_year
                                           69.0
            NaN
                           73.0
                                                        65.000000
2020
           75.0
                           70.0
                                           58.0
                                                        94.000000
2018
           75.0
                           63.0
                                                        69.000000
                                           61.0
2019
3
           69.0
                           70.0
                                           71.0
                                                        73.000000
2019
                           50.0
                                           64.0
           97.0
                                                               NaN
2018
                           75.0
           78.0
                                           62.0
                                                         78.000000
2019
                           71.0
                                           74.0
                                                        66.000000
            NaN
2019
           66.0
                           70.0
                                            NaN
                                                               NaN
2019
                                           69.0
                                                         62.000000
           70.0
                           67.0
```

2010					
2019 9	55.0	20.0	74.0	75.000000	
2021	33.0	2010	, 110	73100000	
10	73.0	NaN	NaN	60.000000	
2018					
11	64.0	60.0	73.0	75.000000	
2021 12	73.0	60.0	73.0	95.000000	
2021	75.0	0010	7310	33.00000	
13	72.0	71.0	61.0	80.000000	
2019					
14	74.0	69.0	68.0	70.000000	
2021 15	73.0	63.0	67.0	73.821429	
2018	75.0	0310	0710	751021125	
16	65.0	86.0	61.0	NaN	
2019	07.0	61.0	00.0		
17	87.0	61.0	90.0	NaN	
2018 18	71.0	69.0	78.0	73.821429	
2019	,1.0	0310	7010	751021125	
19	67.0	NaN	69.0	79.000000	
2020	60.0	06.0	70.0	75 000000	
20 2021	62.0	86.0	78.0	75.000000	
2021	67.0	68.0	64.0	64.000000	
2018	07.10	00.0	0110	01100000	
22	66.0	69.0	72.0	79.000000	
2021	63.0	62.0	C1 0	N - N	
23 2019	63.0	62.0	61.0	NaN	
24	67.0	76.0	70.0	65.000000	
2018					
25	55.0	74.0	66.0	NaN	
2020	06.0	60.0	20.0	60 000000	
26 2021	96.0	60.0	20.0	69.000000	
27	75.0	79.0	73.0	80.000000	
2019					
28	70.0	79.0	57.0	63.000000	
2019	70.0	74.0	75.0	60,000000	
29 2020	70.0	74.0	75.0	69.000000	
2020					
	ement-offe				
0 1		2			
2		0 2			
3		2			
_		<del>-</del>			

Δ	0
4 5 6 7 8 9 10	0
6	0 2
7	0
8	2
q	0
10	2
11	0
12	0
13	0
14	2
15	0
15 16	0 2 0 2 0 0 0 2 0 0 0 0 2 0 0 0 2
17	0
10	2
18 19	0
20	0
21	2
22	0
23	0
23 24 25	2
25	0
26	2
27	0
28	2
29	0 0 2 0 2 0 2 2 2
23	_

# 1. Dropping rows with at least 1 null value
ndf1.dropna()

	ath_score join year	reading_score	writing_score	placement_score
1 2018	75.0	70.0	58.0	94.000000
2	75.0	63.0	61.0	69.000000
2019	69.0	70.0	71.0	73.000000
2019	78.0	75.0	62.0	78.000000
2019 8	70.0	67.0	69.0	62.000000
2019 9	55.0	20.0	74.0	75.000000
2021 11	64.0	60.0	73.0	75.000000
2021 12	73.0	60.0	73.0	95.000000
2021 13	72.0	71.0	61.0	80.000000
2019	7210	7110	0110	23100000

14	74.0	69.0	68.0	70.000000
2021 15	73.0	63.0	67.0	73.821429
2018	7310	0310	07.10	731021123
18	71.0	69.0	78.0	73.821429
2019 20	62.0	86.0	78.0	75.000000
2021	02.0	00.0	70.0	73.00000
21	67.0	68.0	64.0	64.000000
2018				
22 2021	66.0	69.0	72.0	79.000000
24	67.0	76.0	70.0	65.000000
2018				
26	96.0	60.0	20.0	69.000000
2021 27	75.0	79.0	73.0	80.000000
2019	75.0	73.0	75.0	00.00000
28	70.0	79.0	57.0	63.000000
2019	70.0	74.0	75 0	60 000000
29 2020	70.0	74.0	75.0	69.000000
2020				

1 2 3 5	Placement-offer-count 0 2 2
	0
8 9	2
11	0
12	9
13	0
14	2
15	0
18	2
20 21	0 2
22	0
24	2
26	2
27	0
28	2
29	2

## # 2. Dropping rows if all values in that row are missing ndfl.dropna(how = 'all')

math\_score reading\_score writing\_score placement\_score
club\_join\_year \

0	NaN	73.0	69.0	65.000000
2020 1	75.0	70.0	58.0	94.000000
2018	73.0	70.0	50.0	54.000000
2	75.0	63.0	61.0	69.000000
2019	60.0	70.0	71.0	72 22222
3 2019	69.0	70.0	71.0	73.000000
4	97.0	50.0	64.0	NaN
2018				
5	78.0	75.0	62.0	78.000000
2019	N - N	71 0	74.0	66 000000
2019	NaN	71.0	74.0	66.000000
013	66.0	70.0	NaN	NaN
019				
010	70.0	67.0	69.0	62.000000
019	EE O	20.0	74.0	75 000000
021	55.0	20.0	74.0	75.000000
0	73.0	NaN	NaN	60.000000
018				
1	64.0	60.0	73.0	75.000000
021 2	73.0	60.0	73.0	95.000000
021	73.0	00.0	73.0	93.00000
3	72.0	71.0	61.0	80.000000
019				
4	74.0	69.0	68.0	70.000000
921 5	73.0	63.0	67.0	73.821429
918	75.0	03.0	07.0	75.021425
5	65.0	86.0	61.0	NaN
019	07.0	61.0	00.0	
7 018	87.0	61.0	90.0	NaN
8	71.0	69.0	78.0	73.821429
019	,	33.0	. 3.10	.5.521.25
9	67.0	NaN	69.0	79.000000
2020	62.0	06.0	70.0	75 000000
0 021	62.0	86.0	78.0	75.000000
21	67.0	68.0	64.0	64.000000
2018				
22	66.0	69.0	72.0	79.000000
2021	62.0	62.0	61.0	N - Al
23 2019	63.0	62.0	61.0	NaN
24	67.0	76.0	70.0	65.000000

```
2018
                            74.0
                                            66.0
           55.0
                                                                 NaN
25
2020
26
           96.0
                            60.0
                                             20.0
                                                          69.000000
2021
27
           75.0
                            79.0
                                             73.0
                                                          80.000000
2019
28
           70.0
                            79.0
                                             57.0
                                                          63.000000
2019
29
           70.0
                            74.0
                                             75.0
                                                          69.000000
2020
    Placement-offer-count
0
1
                           0
2
                           2
3
                           2
4
                           0
5
                           0
6
                           2
7
                           0
8
                           2
9
                           0
10
                           2
                           0
11
                           0
12
                           0
13
                           2
14
15
                           0
                           0
16
                           0
17
                           2
18
                           0
19
                           0
20
                           2
21
                           0
22
23
                           0
                           2
0
24
25
                           2
26
                           0
27
                           2
28
29
# 3. Dropping columns with at least 1 null value.
ndfl.dropna(axis = 1)
    club_join_year
                      Placement-offer-count
               2020
0
                                             2
1
               2018
                                             0
```

2 3 4 5 6 7 8 9	2019 2019 2018 2019 2019	2 2 0 0 2
7	2019	0
8	2019	2
9	2021	0
10	2018	2
11	2021	0
12	2021	0
13	2019	0
14	2021	2
15	2018	0
16	2019	0
17	2018	0
18	2019	2
19	2020	0
20	2021	0
21	2018	2
22	2021	0
23	2019	0
24	2018	2
25	2020	0
26	2021	2
27 28 29	2019 2019 2019 2020	0 2 2

# 4. Dropping Rows with at least 1 null value in CSV file new\_data = ndfl.dropna(axis = 0, how ='any') new\_data

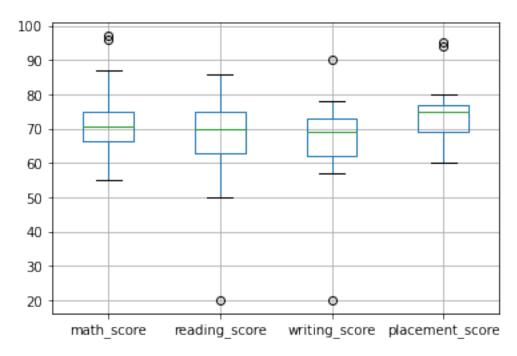
	math score	reading score	writing score	placement score
_1	<b>—</b>	Leauring_score	wilting_score	pracement_score
ctu	b_join_year	\		0.4.000000
1	75.0	70.0	58.0	94.000000
201	8			
2	75.0	63.0	61.0	69.000000
201	9			
3	69.0	70.0	71.0	73.000000
201	9			
5	78.0	75.0	62.0	78.000000
201	9			
8	70.0	67.0	69.0	62.000000
201	9			
9	55.0	20.0	74.0	75.000000
202	1			
11	64.0	60.0	73.0	75.000000
202	1			
12	73.0	60.0	73.0	95.000000

2021				
13	72.0	71.0	61.0	80.000000
2019				
14	74.0	69.0	68.0	70.000000
2021				
15	73.0	63.0	67.0	73.821429
2018				
18	71.0	69.0	78.0	73.821429
2019	62.0	06.0	70.0	75 000000
20	62.0	86.0	78.0	75.000000
2021	67.0	60.0	64.0	64 000000
21 2018	67.0	68.0	64.0	64.000000
2018	66.0	69.0	72.0	79.000000
2021	00.0	03.0	72.0	73.00000
24	67.0	76.0	70.0	65.000000
2018	07.10	, 0.0	70.0	05100000
26	96.0	60.0	20.0	69.000000
2021				
27	75.0	79.0	73.0	80.000000
2019				
28	70.0	79.0	57.0	63.000000
2019				
29	70.0	74.0	75.0	69.000000
2020				

	Placement-offer-count
1	0
2	2
2 3 5	2
	Θ
8	2
9	Θ
11	Θ
12	Θ
13	Θ
14	2
15	Θ
18	2
20	Θ
21	2
22	0
24	2
26	2
27	Θ
28	2
29	2

#Outliers d1= df

```
#Boxplot
col = ['math_score', 'reading_score',
'writing_score','placement_score']
df.boxplot(col)
<AxesSubplot:>
```

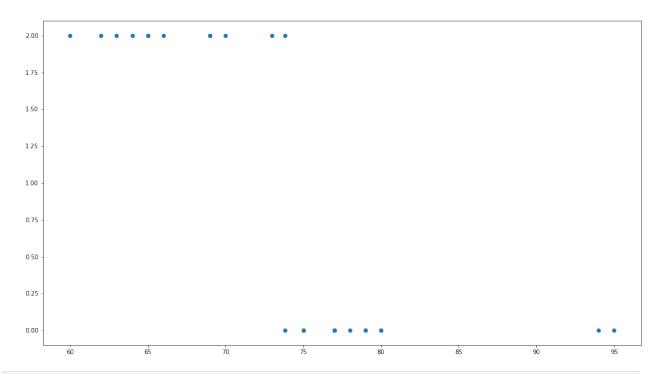


```
print(np.where(df['math_score']>90))
print(np.where(df['reading_score']<25))
print(np.where(df['writing_score']<30))

(array([ 4, 26]),)
  (array([9]),)
  (array([26]),)

import matplotlib.pyplot as plt

#scatterplot
fig, ax = plt.subplots(figsize = (18,10))
ax.scatter(d1['placement_score'], d1['Placement-offer-count'])
plt.show()
ax.set_xlabel('(Proportion non-retail business acres)/(town)')
ax.set_ylabel('(Full-value property-tax rate)/($10,000)')</pre>
```



Text(3.2000000000017, 0.5, '(Full-value property-tax rate)/(\$10,000)')

df					
	math_score	reading_score	writing_score	placement_score	
clu 0	b_join_year 77	73.0	69.0	65.000000	
202		75.0	05.0	03.000000	
1	75	70.0	58.0	94.000000	
201 2	.8 75	62.0	61.0	60 000000	
201		63.0	61.0	69.000000	
3	69	70.0	71.0	73.000000	
201		50.0	64.0	77 000000	
4 201	97	50.0	64.0	77.000000	
5	78	75.0	62.0	78.000000	
201		71.0	74.0	66 00000	
6 201	77	71.0	74.0	66.000000	
7	.9	70.0	NaN	77.000000	
201					
8 201	70	67.0	69.0	62.000000	
9	.9 55	20.0	74.0	75.000000	
202					
10	73	77.0	77.0	60.000000	
201	.8				

11	64	60.0	73.0	75.000000	
2021 12	73	60.0	73.0	95.000000	
2021 13	72	71.0	61.0	80.000000	
2019 14	74	69.0	68.0	70.000000	
2021 15	73	63.0	67.0	73.821429	
2018 16	65	86.0	61.0	77.000000	
2019 17	87	61.0	90.0	77.000000	
2018 18	71	69.0	78.0	73.821429	
2019 19	67	77.0	69.0	79.000000	
2020 20	62	86.0	78.0	75.000000	
2021 21	67	68.0	64.0	64.000000	
2018 22	66	69.0	72.0	79.000000	
2021					
23 2019	63	62.0	61.0	77.000000	
24 2018	67	76.0	70.0	65.000000	
25 2020	55	74.0	66.0	77.000000	
26 2021	96	60.0	20.0	69.000000	
27 2019	75	79.0	73.0	80.000000	
28 2019	70	79.0	57.0	63.000000	
29 2020	70	74.0	75.0	69.000000	
	nent-offe	r-count			
0		2			
2		0 2 2			
4		0			
1 2 3 4 5 6 7		0 2 0			
8		2			

```
9
                          0
10
                          2
                          0
11
12
                          0
                          0
13
14
                          2
                          0
15
16
                          0
17
                          0
                          2
18
                          0
19
                          0
20
21
                          2
                          0
22
                          0
23
24
                          2
                          0
25
                          2
26
27
                          0
                          2
28
                          2
29
print(np.where((df['placement_score']<50) & (df['Placement-offer-</pre>
count']>1)))
print(np.where((df['placement_score']>85) & (df['Placement-offer-
count']<3)))
(array([], dtype=int64),)
(array([ 1, 12]),)
#Z-score
from scipy import stats
z = np.abs(stats.zscore(df['math_score']))
print(z)
0
      0.578370
1
      0.362828
2
      0.362828
3
      0.283797
4
      2.733787
5
      0.686141
6
      0.578370
7
      0.607109
8
      0.176026
9
      1.792588
10
      0.147287
11
      0.822651
12
      0.147287
13
      0.039516
14
      0.255058
```

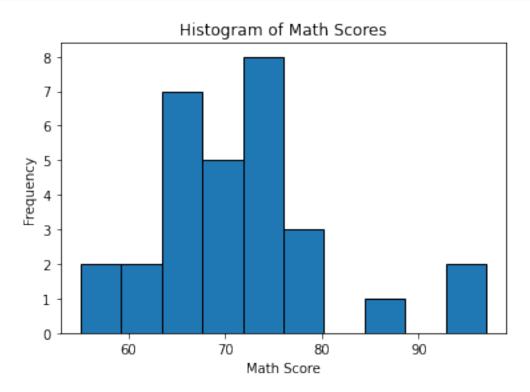
```
15
      0.147287
16
      0.714880
17
      1.656078
18
      0.068255
19
      0.499338
20
      1.038192
21
      0.499338
22
      0.607109
23
      0.930421
24
      0.499338
25
      1.792588
26
     2.626016
27
      0.362828
28
      0.176026
29
      0.176026
Name: math score, dtype: float64
threshold = 0.18
sample outliers = np.where(z <threshold)</pre>
sample outliers
(array([ 6, 15, 24, 25]),)
# Detecting outliers using Inter Quantile Range(IQR):
sorted_rscore= sorted(d1['reading_score'])
sorted rscore
q1 = np.percentile(sorted rscore, 25)
q3 = np.percentile(sorted_rscore, 75)
print(q1,q3)
IQR = q3-q1
lwr bound = q1-(1.5*IQR)
upr bound = q3+(1.5*IQR)
print(lwr bound, upr bound)
r_outliers = []
for i in sorted rscore:
    if (i<lwr_bound or i>upr_bound):
        r outliers.append(i)
print(r outliers)
63.0 74.75
45.375 92.375
[20.0]
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

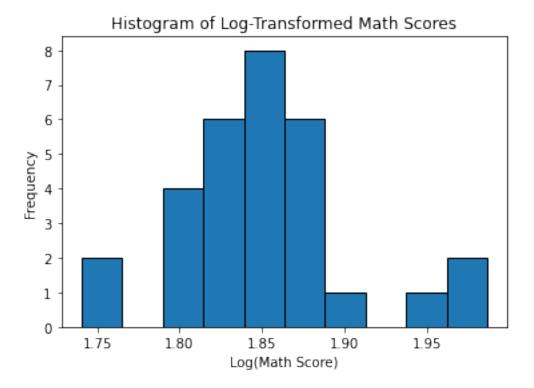
```
df =
pd.read_excel("/home/ubuntu/TCOB35/DSBDAL/Student_Performance.xlsx")

df['math_score'].plot(kind='hist', title='Histogram of Math Scores',
edgecolor='black')
plt.xlabel('Math Score')
plt.ylabel('Frequency')
plt.show()

df['log_math'] = np.log10(df['math_score'])

df['log_math'].plot(kind='hist', title='Histogram of Log-Transformed
Math Scores', edgecolor='black')
plt.xlabel('Log(Math Score)')
plt.ylabel('Frequency')
plt.show()
```





```
#Handling of Outliers
sample_outliers=[9,1,26,4] #contains index of row where outlier is
present
# 1. Trimming/Removing the Outliers:
new_df = df
for i in sample outliers:
    new df.drop(i, inplace=True)
new_df
                reading_score writing_score placement_score
    math_score
club join year
                          73.0
                                         69.0
                                                            65
            77
0
2020
            75
                          63.0
                                         61.0
                                                            69
2019
            69
                          70.0
                                         71.0
                                                            73
2019
            78
                          75.0
                                          62.0
                                                            78
2019
            77
                          71.0
                                          74.0
                                                            66
2019
                                                            77
            66
                          70.0
                                           NaN
2019
                                         69.0
            70
                          67.0
                                                            62
8
2019
            73
                          77.0
                                          77.0
                                                            60
10
2018
```

	<b>.</b> .	60.0	72.0	
11 2021	64	60.0	73.0	75
12	73	60.0	73.0	95
2021				
13	72	71.0	61.0	80
2019	7.4	60 0	60 0	70
14 2021	74	69.0	68.0	70
15	73	63.0	67.0	na
2018				
16	65	86.0	61.0	77
2019	0.7	61.0	00.0	77
17 2018	87	61.0	90.0	77
18	71	69.0	78.0	NaN
2019	, 1	0310	7010	real e
19	67	77.0	69.0	79
2020				
20	62	NaN	78.0	75
2021 21	67	68.0	64.0	64
2018	07	00.0	04.0	04
22	66	69.0	72.0	79
2021				
23	63	62.0	61.0	77
2019	67	76 0	70 0	6 E
24 2018	67	76.0	70.0	65
25	55	74.0	66.0	77
2020				
27	75	79.0	73.0	80
2019	70	70.0	F7 0	62
28 2019	70	79.0	57.0	63
29	70	74.0	75.0	69
2020				
	ent-offer-cou	J_		
0		2 1.886491 2 1.875061		
3		2 1.838849		
5		0 1.892095		
0 2 3 5 6 7		2 1.886491		
7		0 1.819544		
8		2 1.845098		
10 11		2 1.863323 0 1.806180		
12		0 1.863323		
13		0 1.857332		

```
14
                            1.869232
15
                         0
                           1.863323
16
                           1.812913
17
                         0
                            1.939519
18
                         2
                           1.851258
19
                            1.826075
20
                         0
                           1.792392
21
                         2
                            1.826075
                            1.819544
22
                         0
23
                         0
                           1.799341
24
                         2
                           1.826075
25
                         0
                           1.740363
27
                            1.875061
                         0
28
                         2
                            1.845098
29
                         2
                            1.845098
# 2. Quantile-Based Flooring and Capping:
import numpy as np
df stud = df
ninetieth percentile = np.percentile(df stud['math score'], 90)
# Cap values above the 90th percentile
b = np.where(df_stud['math_score'] > ninetieth percentile,
ninetieth percentile, df stud['math score'])
print("New array:", b)
# Insert the capped data back into the DataFrame
df stud.insert(1, "m score", b, True)
df stud
New array: [77. 75. 69. 77. 77. 66. 70. 73. 64. 73. 72. 74. 73. 65.
77. 71. 67. 62.
67. 66. 63. 67. 55. 75. 70. 70.1
    math_score m score reading_score writing_score placement score
\
0
            77
                   77.0
                                   73.0
                                                   69.0
                                                                      65
2
            75
                   75.0
                                   63.0
                                                   61.0
                                                                      69
3
            69
                   69.0
                                   70.0
                                                   71.0
                                                                      73
5
            78
                   77.0
                                   75.0
                                                   62.0
                                                                      78
            77
                                                   74.0
                                                                      66
                   77.0
                                   71.0
7
            66
                   66.0
                                   70.0
                                                    NaN
                                                                      77
8
            70
                   70.0
                                   67.0
                                                   69.0
                                                                      62
```

10	73	73.0	77.0	77.0	60
11	64	64.0	60.0	73.0	75
12	73	73.0	60.0	73.0	95
13	72	72.0	71.0	61.0	80
14	74	74.0	69.0	68.0	70
15	73	73.0	63.0	67.0	na
16	65	65.0	86.0	61.0	77
17	87	77.0	61.0	90.0	77
18	71	71.0	69.0	78.0	NaN
19	67	67.0	77.0	69.0	79
20	62	62.0	NaN	78.0	75
21	67	67.0	68.0	64.0	64
22	66	66.0	69.0	72.0	79
23	63	63.0	62.0	61.0	77
24	67	67.0	76.0	70.0	65
25	55	55.0	74.0	66.0	77
27	75	75.0	79.0	73.0	80
28	70	70.0	79.0	57.0	63
29	70	70.0	74.0	75.0	69
0 2 3 5 6 7 8 10 11 12	club_join_year 2020 2019 2019 2019 2019 2019 2018 2021 2021		nt-offer-count 2 2 2 0 2 0 2 2 0	log_math 1.886491 1.875061 1.838849 1.892095 1.886491 1.819544 1.845098 1.863323 1.806180 1.863323	

```
13
              2019
                                            1.857332
14
              2021
                                         2 1.869232
15
              2018
                                            1.863323
16
              2019
                                         0
                                            1.812913
17
              2018
                                            1.939519
                                         2
18
              2019
                                            1.851258
19
              2020
                                         0
                                           1.826075
20
              2021
                                         0
                                            1.792392
21
              2018
                                         2
                                            1.826075
22
              2021
                                            1.819544
              2019
23
                                            1.799341
                                         0
                                         2 1.826075
24
              2018
25
              2020
                                           1.740363
27
              2019
                                         0
                                            1.875061
28
              2019
                                            1.845098
29
              2020
                                            1.845098
# 3. Mean/Median Imputation:
import numpy as np
col = ['reading score']
df.boxplot(col)
sorted rscore = sorted(df['reading_score'])
median = np.median(sorted rscore)
print("Median:", median)
refined df = df.copy()
refined df['reading score'] = np.where(refined df['reading score'] >
upr bound, median, refined df['reading score'])
refined df['reading score'] = np.where(refined df['reading score'] <</pre>
lwr bound, median, refined df['reading score'])
refined df.boxplot(col)
Median: 70.0
<AxesSubplot:>
```

