```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder


df=pd.read_csv('stud.csv')
df
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement_Offer_Count	Gender
0	79	86	66	76	2020	2	male
1	76	85	78	96	2018	3	female
2	60	81	67	76	2020	2	female
3	66	76	72	93	2020	3	male
4	64	84	61	78	2020	2	male
5	66	94	78	80	2019	2	female
6	70	76	69	82	2019	2	male
7	63	83	72	90	2018	3	male
8	61	83	67	81	2020	2	female
9	60	93	75	95	2018	3	female
10	80	94	74	98	2018	3	male
11	62	91	78	82	2019	2	male
12	66	77	65	96	2020	3	female
13	70	78	71	100	2020	3	male
14	77	81	69	85	2021	2	male
15	72	87	71	99	2020	3	female
16	63	85	70	87	2018	3	female
17	77	86	68	96	2020	3	male
18	70	93	73	85	2020	2	male
19	61	91	70	79	2020	2	female
20	70	84	80	97	2019	3	male
21	75	78	75	79	2019	2	male
22	64	93	66	76	2021	2	female
23	70	81	61	99	2019	3	female
24	72	89	60	89	2018	3	male
25	62	84	79	98	2019	3	male
26	72	92	74	87	2018	3	female
27	65	95	68	89	2019	3	male
28	75	91	65	85	2019	2	male
29	79	76	66	86	2019	3	female

```
df.head()
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement_Offer_Count	Gender
0	79	86	66	76	2020	2	male
1	76	85	78	96	2018	3	female
2	60	81	67	76	2020	2	female
3	66	76	72	93	2020	3	male
4	64	84	61	78	2020	2	male

```
df.tail()
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement_Offer_Count	Gender
25	62	84	79	98	2019	3	male
26	72	92	74	87	2018	3	female
27	65	95	68	89	2019	3	male
28	75	91	65	85	2019	2	male
29	79	76	66	86	2019	3	female

```
df.columns
```

```
Index(['Math_Score', 'Reading_Score', 'Writing_Score', 'Placement_Score',
       'Club_Join_Date', 'Placement_Offer_Count', 'Gender'],
      dtype='object')
```

```
df.index
```

```
RangeIndex(start=0, stop=30, step=1)
```

```
df.describe()
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement_Offer_Count
count	30.00000	30.000000	30.000000	30.000000	30.000000	30.000000
mean	68.90000	85.566667	70.266667	87.966667	2019.266667	2.566667
std	6.31009	6.123349	5.445234	7.989145	0.907187	0.504007
min	60.00000	76.000000	60.000000	76.000000	2018.000000	2.000000
25%	63.25000	81.000000	66.250000	81.250000	2019.000000	2.000000
50%	70.00000	85.000000	70.000000	87.000000	2019.000000	3.000000
75%	74.25000	91.000000	74.000000	96.000000	2020.000000	3.000000
max	80.00000	95.000000	80.000000	100.000000	2021.000000	3.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 7 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   Math_Score            30 non-null     int64 
 1   Reading_Score         30 non-null     int64 
 2   Writing_Score         30 non-null     int64 
 3   Placement_Score       30 non-null     int64 
 4   Club_Join_Date        30 non-null     int64 
 5   Placement_Offer_Count  30 non-null     int64 
 6   Gender                30 non-null     object
```

```
dtypes: int64(6), object(1)
memory usage: 1.8+ KB
```

```
df.isnull()
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placemen
0	False	False	False	False	False	
1	False	False	False	False	False	
2	False	False	False	False	False	
3	False	False	False	False	False	
4	False	False	False	False	False	
5	False	False	False	False	False	
6	False	False	False	False	False	
7	False	False	False	False	False	
8	False	False	False	False	False	
9	False	False	False	False	False	
10	False	False	False	False	False	
11	False	False	False	False	False	
12	False	False	False	False	False	
13	False	False	False	False	False	
14	False	False	False	False	False	
15	False	False	False	False	False	
16	False	False	False	False	False	
17	False	False	False	False	False	
18	False	False	False	False	False	
19	False	False	False	False	False	
20	False	False	False	False	False	
21	False	False	False	False	False	
22	False	False	False	False	False	
23	False	False	False	False	False	
24	False	False	False	False	False	
25	False	False	False	False	False	
26	False	False	False	False	False	
27	False	False	False	False	False	
28	False	False	False	False	False	
29	False	False	False	False	False	

```
df.isnull().sum()
```

```
Math_Score              0
Reading_Score           0
Writing_Score           0
Placement_Score         0
Club_Join_Date          0
Placement_Offer_Count   0
Gender                  0
dtype: int64
```

```
# Check if the 'Math_Score' column is null
series = pd.isnull(df['Math_Score'])
print(series)
```

```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16    False
17    False
18    False
19    False
20    False
21    False
22    False
23    False
24    False
25    False
26    False
27    False
28    False
29    False
Name: Math_Score, dtype: bool
```

```
df.notnull()
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement_Offer_Count	Gender
0	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True
5	True	True	True	True	True	True	True
6	True	True	True	True	True	True	True
7	True	True	True	True	True	True	True
8	True	True	True	True	True	True	True
9	True	True	True	True	True	True	True
10	True	True	True	True	True	True	True
11	True	True	True	True	True	True	True
12	True	True	True	True	True	True	True
13	True	True	True	True	True	True	True
14	True	True	True	True	True	True	True
15	True	True	True	True	True	True	True
16	True	True	True	True	True	True	True
17	True	True	True	True	True	True	True
18	True	True	True	True	True	True	True
19	True	True	True	True	True	True	True
20	True	True	True	True	True	True	True
21	True	True	True	True	True	True	True
22	True	True	True	True	True	True	True
23	True	True	True	True	True	True	True
24	True	True	True	True	True	True	True
25	True	True	True	True	True	True	True
26	True	True	True	True	True	True	True
27	True	True	True	True	True	True	True
28	True	True	True	True	True	True	True
29	True	True	True	True	True	True	True

```
series1=pd.notnull(df['Math_Score'])
df[series1]
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement_Offer_Count	Gender
0	79	86	66	76	2020	2	male
1	76	85	78	96	2018	3	female
2	60	81	67	76	2020	2	female
3	66	76	72	93	2020	3	male
4	64	84	61	78	2020	2	male
5	66	94	78	80	2019	2	female
6	70	76	69	82	2019	2	male
7	63	83	72	90	2018	3	male
8	61	83	67	81	2020	2	female
9	60	93	75	95	2018	3	female
10	80	94	74	98	2018	3	male
11	62	91	78	82	2019	2	male
12	66	77	65	96	2020	3	female
13	70	78	71	100	2020	3	male
14	77	81	69	85	2021	2	male
15	72	87	71	99	2020	3	female
16	63	85	70	87	2018	3	female
17	77	86	68	96	2020	3	male
18	70	93	73	85	2020	2	male
19	61	91	70	79	2020	2	female
20	70	84	80	97	2019	3	male
21	75	78	75	79	2019	2	male
22	64	93	66	76	2021	2	female
23	70	81	61	99	2019	3	female
24	72	89	60	89	2018	3	male
25	62	84	79	98	2019	3	male
26	72	92	74	87	2018	3	female
27	65	95	68	89	2019	3	male
28	75	91	65	85	2019	2	male
29	79	76	66	86	2019	3	female

```
from sklearn.preprocessing import LabelEncoder
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Gender']=le.fit_transform(df['Gender']) #transform gender categorical column in int
newdf=df
```

```
newdf
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placemen
0	79	86	66	76	2020	
1	76	85	78	96	2018	
2	60	81	67	76	2020	
3	66	76	72	93	2020	
4	64	84	61	78	2020	
5	66	94	78	80	2019	
6	70	76	69	82	2019	
7	63	83	72	90	2018	
8	61	83	67	81	2020	
9	60	93	75	95	2018	
10	80	94	74	98	2018	
11	62	91	78	82	2019	
12	66	77	65	96	2020	
13	70	78	71	100	2020	
14	77	81	69	85	2021	
15	72	87	71	99	2020	
16	63	85	70	87	2018	
17	77	86	68	96	2020	
18	70	93	73	85	2020	
19	61	91	70	79	2020	
20	70	84	80	97	2019	
21	75	78	75	79	2019	
22	64	93	66	76	2021	
23	70	81	61	99	2019	
24	72	89	60	89	2018	
25	62	84	79	98	2019	
26	72	92	74	87	2018	
27	65	95	68	89	2019	
28	75	91	65	85	2019	
29	79	76	66	86	2019	

```
#for filling the missing values
ndf=df
ndf.fillna(0)
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placemen
0	79	86	66	76	2020	
1	76	85	78	96	2018	
2	60	81	67	76	2020	
3	66	76	72	93	2020	
4	64	84	61	78	2020	
5	66	94	78	80	2019	
6	70	76	69	82	2019	
7	63	83	72	90	2018	
8	61	83	67	81	2020	
9	60	93	75	95	2018	
10	80	94	74	98	2018	
11	62	91	78	82	2019	
12	66	77	65	96	2020	
13	70	78	71	100	2020	
14	77	81	69	85	2021	
15	72	87	71	99	2020	
16	63	85	70	87	2018	
17	77	86	68	96	2020	
18	70	93	73	85	2020	
19	61	91	70	79	2020	
20	70	84	80	97	2019	
21	75	78	75	79	2019	
22	64	93	66	76	2021	
23	70	81	61	99	2019	
24	72	89	60	89	2018	
25	62	84	79	98	2019	
26	72	92	74	87	2018	
27	65	95	68	89	2019	
28	75	91	65	85	2019	
29	79	76	66	86	2019	

```
df['Math_Score'] = df['Math_Score'].fillna(df['Math_Score'].mean())
```

```
df.dropna()
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placemen
0	79	86	66	76	2020	
1	76	85	78	96	2018	
2	60	81	67	76	2020	
3	66	76	72	93	2020	
4	64	84	61	78	2020	
5	66	94	78	80	2019	
6	70	76	69	82	2019	
7	63	83	72	90	2018	
8	61	83	67	81	2020	
9	60	93	75	95	2018	
10	80	94	74	98	2018	
11	62	91	78	82	2019	
12	66	77	65	96	2020	
13	70	78	71	100	2020	
14	77	81	69	85	2021	
15	72	87	71	99	2020	
16	63	85	70	87	2018	
17	77	86	68	96	2020	
18	70	93	73	85	2020	
19	61	91	70	79	2020	
20	70	84	80	97	2019	
21	75	78	75	79	2019	
22	64	93	66	76	2021	
23	70	81	61	99	2019	
24	72	89	60	89	2018	
25	62	84	79	98	2019	
26	72	92	74	87	2018	
27	65	95	68	89	2019	
28	75	91	65	85	2019	
29	79	76	66	86	2019	

```
newdf.dropna(axis=1)
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placemen
0	79	86	66	76	2020	
1	76	85	78	96	2018	
2	60	81	67	76	2020	
3	66	76	72	93	2020	
4	64	84	61	78	2020	
5	66	94	78	80	2019	
6	70	76	69	82	2019	
7	63	83	72	90	2018	
8	61	83	67	81	2020	
9	60	93	75	95	2018	
10	80	94	74	98	2018	
11	62	91	78	82	2019	
12	66	77	65	96	2020	
13	70	78	71	100	2020	
14	77	81	69	85	2021	
15	72	87	71	99	2020	
16	63	85	70	87	2018	
17	77	86	68	96	2020	
18	70	93	73	85	2020	
19	61	91	70	79	2020	
20	70	84	80	97	2019	
21	75	78	75	79	2019	
22	64	93	66	76	2021	
23	70	81	61	99	2019	
24	72	89	60	89	2018	
25	62	84	79	98	2019	
26	72	92	74	87	2018	
27	65	95	68	89	2019	
28	75	91	65	85	2019	
29	79	76	66	86	2019	

```
newdf.dropna(axis=0)
```

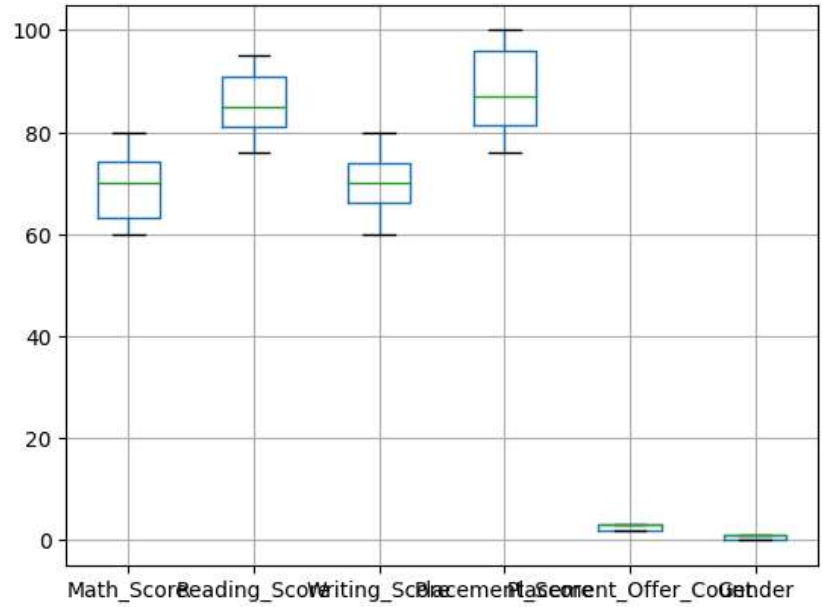
	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placement_Offer_Count	Gender
0	79	86	66	76	2020	2	1
1	76	85	78	96	2018	3	0
2	60	81	67	76	2020	2	0
3	66	76	72	93	2020	3	1
4	64	84	61	78	2020	2	1
5	66	94	78	80	2019	2	0
6	70	76	69	82	2019	2	1
7	63	83	72	90	2018	3	1
8	61	83	67	81	2020	2	0
9	60	93	75	95	2018	3	0
10	80	94	74	98	2018	3	1
11	62	91	78	82	2019	2	1
12	66	77	65	96	2020	3	0
13	70	78	71	100	2020	3	1
14	77	81	69	85	2021	2	1
15	72	87	71	99	2020	3	0
16	63	85	70	87	2018	3	0
17	77	86	68	96	2020	3	1
18	70	93	73	85	2020	2	1
19	61	91	70	79	2020	2	0
20	70	84	80	97	2019	3	1
21	75	78	75	79	2019	2	1
22	64	93	66	76	2021	2	0
23	70	81	61	99	2019	3	0
24	72	89	60	89	2018	3	1
25	62	84	79	98	2019	3	1
26	72	92	74	87	2018	3	0
27	65	95	68	89	2019	3	1
28	75	91	65	85	2019	2	1
29	79	76	66	86	2019	3	0

```
df.describe()
```

	Math_Score	Reading_Score	Writing_Score	Placement_Score	Club_Join_Date	Placer
count	30.000000	30.000000	30.000000	30.000000	30.000000	

```
col=['Math_Score','Reading_Score','Writing_Score','Placement_Score','Placement_Offer_Count',    'Gender']
df.boxplot(col)
```

```
<Axes: >
```




```
print(np.where(df['Math_Score']>90))
print(np.where(df['Reading_Score']<25))
print(np.where(df['Writing_Score']<30))
#to print the outliers but as we dont have any outliers the array will be empty
```

```
(array([], dtype=int64),)
(array([], dtype=int64),)
(array([], dtype=int64),)
```

```
#detecting outliers using the scatterplot
fig,ax=plt.subplots(figsize=(18,10))
ax.scatter(df['Placement_Score'],df['Placement_Offer_Count'])
ax.set_xlabel('placement score')
ax.set_ylabel('placement offer count')
plt.show()
```