Aim:Find the outliers from the given dataset using trimming and capping method

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
df = pd.read_csv('placement - placement.csv')
df.head()
```

Out[2]:

	cgpa	placement_exam_marks	placed
0	7.19	26	1
1	7.46	38	1
2	7.54	40	1
3	6.42	8	1
4	7.23	17	0

In [3]:

```
import seaborn as sns
#from matplotlib import pyplot as plt
import matplotlib.pyplot as plt
```

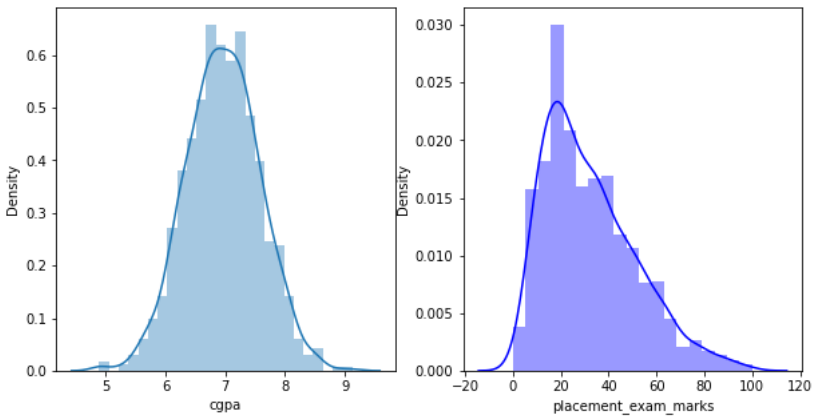
In [4]:

```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.distplot(df['cgpa'])

plt.subplot(1,2,2)
sns.distplot(df['placement_exam_marks'],color='blue')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[4]:

```
<AxesSubplot:xlabel='placement_exam_marks', ylabel='Density'>
```



In [5]:

```
df['placement_exam_marks'].describe()
```

Out[5]:

```
count    1000.000000
mean       32.225000
std        19.130822
min         0.000000
25%        17.000000
50%        28.000000
75%        44.000000
max       100.000000
Name: placement_exam_marks, dtype: float64
```

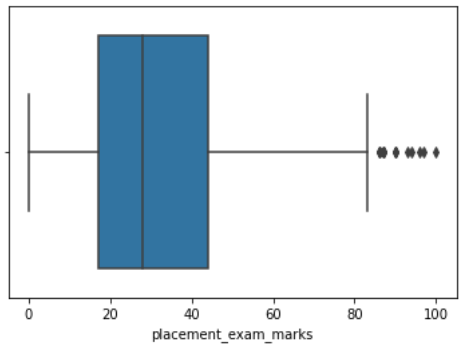
In [6]:

```
sns.boxplot(df['placement_exam_marks'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keywo
rd arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[6]:

```
<AxesSubplot:xlabel='placement_exam_marks'>
```



In [7]:

```
# finding boundaries values
print("Highest Boundary value of cgpa ",df['cgpa'].mean() + 3*df['cgpa'].std())
print("Lowest Boundary value of cgpa ",df['cgpa'].mean() - 3*df['cgpa'].std())
```

```
Highest Boundary value of cgpa  8.808933625397177
Lowest Boundary value of cgpa  5.113546374602842
```

In [8]:

```
# finding outliers
df[(df['cgpa']>8.80)|(df['cgpa']<5.11)]
```

Out[8]:

	cgpa	placement_exam_marks	placed
485	4.92	44	1
995	8.87	44	1
996	9.12	65	1
997	4.89	34	0
999	4.90	10	1

Trimming:

In [9]:

```
df.shape
```

Out[9]:

```
(1000, 3)
```

In [10]:

```
new_df = df[(df['cgpa']<8.80)&(df['cgpa']>5.11)]
new_df
```

Out[10]:

	cgpa	placement_exam_marks	placed
0	7.19	26	1
1	7.46	38	1
2	7.54	40	1
3	6.42	8	1
4	7.23	17	0
...	...	...	...
991	7.04	57	0
992	6.26	12	0
993	6.73	21	1
994	6.48	63	0
998	8.62	46	1

995 rows × 3 columns

In [11]:

```
new_df.shape
```

Out[11]:

```
(995, 3)
```

Z-score

zi = xi-x_mean/S.D

In [12]:

```
df['cgpa_zscore']= (df['cgpa']-df['cgpa'].mean())/df['cgpa'].std()
df
```

Out[12]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
0	7.19	26	1	0.371425
1	7.46	38	1	0.809810
2	7.54	40	1	0.939701
3	6.42	8	1	-0.878782
4	7.23	17	0	0.436371
...	...	...	...	...
995	8.87	44	1	3.099150
996	9.12	65	1	3.505062
997	4.89	34	0	-3.362960
998	8.62	46	1	2.693239
999	4.90	10	1	-3.346724

1000 rows × 4 columns

In [13]:

```
df['cgpa_zscore']
```

Out[13]:

```
0      0.371425
1      0.809810
2      0.939701
3     -0.878782
4      0.436371
         ...
995    3.099150
996    3.505062
997   -3.362960
998    2.693239
999   -3.346724
Name: cgpa_zscore, Length: 1000, dtype: float64
```

In [14]:

```
df[df['cgpa_zscore']>3]
```

Out[14]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
995	8.87	44	1	3.099150
996	9.12	65	1	3.505062

In [15]:

```
df[df['cgpa_zscore']<-3]
```

Out[15]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
485	4.92	44	1	-3.314251
997	4.89	34	0	-3.362960
999	4.90	10	1	-3.346724

In [16]:

```
new_dff = df[(df['cgpa_zscore']<3) | (df['cgpa_zscore']>-3)]
new_dff
```

Out[16]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
0	7.19	26	1	0.371425
1	7.46	38	1	0.809810
2	7.54	40	1	0.939701
3	6.42	8	1	-0.878782
4	7.23	17	0	0.436371
...	...	...	...	...
995	8.87	44	1	3.099150
996	9.12	65	1	3.505062
997	4.89	34	0	-3.362960
998	8.62	46	1	2.693239
999	4.90	10	1	-3.346724

1000 rows × 4 columns

In [17]:

```
new_dff.shape
```

Out[17]:

```
(1000, 4)
```

Capping

In [18]:

```
upper_limit=df['cgpa'].mean() +3 *df['cgpa'].std()
lower_limit=df['cgpa'].mean() -3 *df['cgpa'].std()
lower_limit
```

Out[18]:

```
5.113546374602842
```

In [19]:

```
df['cgpa_cap'] = np.where(
  df['cgpa']>upper_limit,
  upper_limit,
  np.where(
  df['cgpa']<lower_limit,
     lower_limit,df['cgpa']
  )
)
```

In [20]:

```
df.describe()
```

Out[20]:

	cgpa	placement_exam_marks	placed	cgpa_zscore	cgpa_cap
count	1000.000000	1000.000000	1000.000000	1.000000e+03	1000.000000
mean	6.961240	32.225000	0.489000	-1.600275e-14	6.961499
std	0.615898	19.130822	0.500129	1.000000e+00	0.612688
min	4.890000	0.000000	0.000000	-3.362960e+00	5.113546
25%	6.550000	17.000000	0.000000	-6.677081e-01	6.550000
50%	6.960000	28.000000	0.000000	-2.013321e-03	6.960000
75%	7.370000	44.000000	1.000000	6.636815e-01	7.370000
max	9.120000	100.000000	1.000000	3.505062e+00	8.808934

In [21]:

```
df['placement_exam_marks'].skew()
```

Out[21]:

```
0.8356419499466834
```

In [22]:

```
df['cgpa'].skew()
```

Out[22]:

```
-0.014529938929314918
```

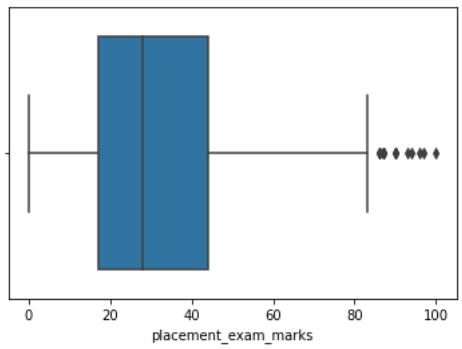
In [23]:

```
sns.boxplot(df['placement_exam_marks'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keywo
rd arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[23]:

```
<AxesSubplot:xlabel='placement_exam_marks'>
```



In [24]:

```
percentile25=df['placement_exam_marks'].quantile(0.25)
percentile75=df['placement_exam_marks'].quantile(0.75)
```

In [25]:

```
percentile75
```

Out[25]:

```
44.0
```

In [26]:

```
percentile25
```

Out[26]:

```
17.0
```

In [27]:

```
iqr = percentile75 - percentile25
iqr
```

Out[27]:

```
27.0
```

In [28]:

```
upper_limit = percentile75 + 1.5 *iqr
upper_limit
```

Out[28]:

```
84.5
```

In [29]:

```
lower_limit = percentile25 - 1.5 *iqr
lower_limit
```

Out[29]:

```
-23.5
```

In [30]:

```
print('upper limit of Height is:' ,upper_limit)
print('lower_limit of Height is:',lower_limit)
```

```
upper limit of Height is: 84.5
lower_limit of Height is: -23.5
```

In [31]:

```
df[df['placement_exam_marks']> upper_limit]
```

Out[31]:

	cgpa	placement_exam_marks	placed	cgpa_zscore	cgpa_cap
9	7.75	94	1	1.280667	7.75
40	6.60	86	1	-0.586526	6.60
61	7.51	86	0	0.890992	7.51
134	6.33	93	0	-1.024910	6.33
162	7.80	90	0	1.361849	7.80
283	7.09	87	0	0.209061	7.09
290	8.38	87	0	2.303564	8.38
311	6.97	87	1	0.014223	6.97
324	6.64	90	0	-0.521580	6.64
630	6.56	96	1	-0.651472	6.56
685	6.05	87	1	-1.479531	6.05
730	6.14	90	1	-1.333403	6.14
771	7.31	86	1	0.566263	7.31
846	6.99	97	0	0.046696	6.99
917	5.95	100	0	-1.641896	5.95

In [32]:

```
df[df['placement_exam_marks'] < lower_limit]
```

Out[32]:

	cgpa	placement_exam_marks	placed	cgpa_zscore	cgpa_cap

Trimming

In [35]:

```
new_dff= df[df['placement_exam_marks'] < upper_limit]
new_dff
```

Out[35]:

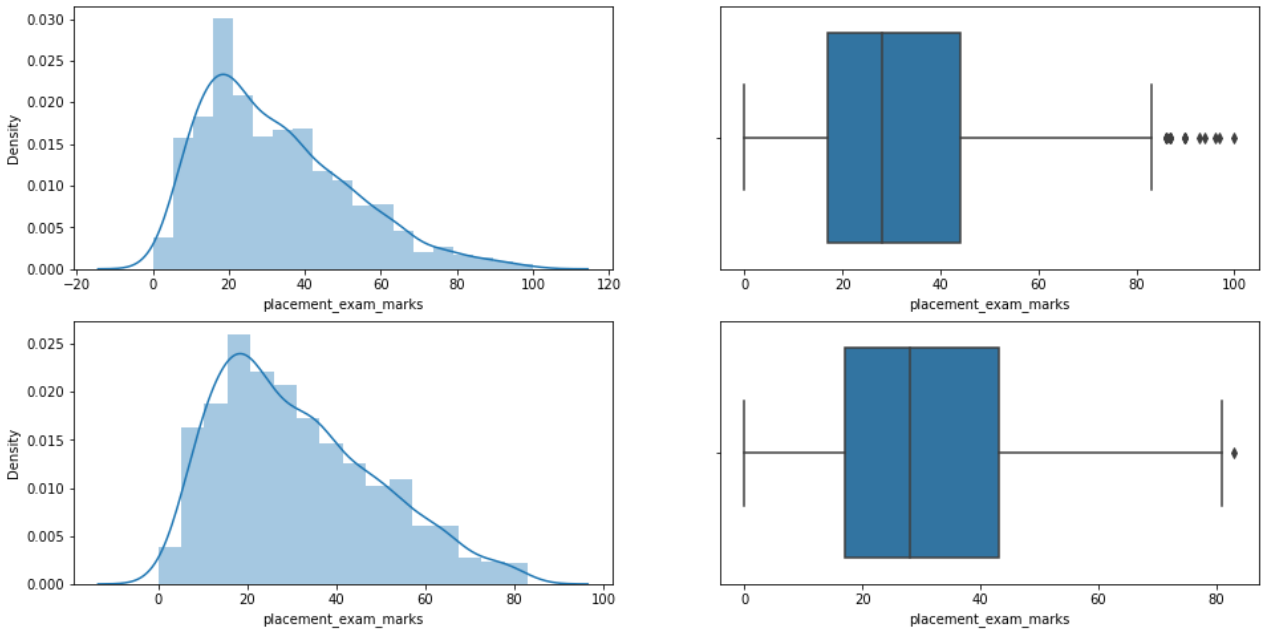
	cgpa	placement_exam_marks	placed	cgpa_zscore	cgpa_cap
0	7.19	26	1	0.371425	7.190000
1	7.46	38	1	0.809810	7.460000
2	7.54	40	1	0.939701	7.540000
3	6.42	8	1	-0.878782	6.420000
4	7.23	17	0	0.436371	7.230000
...	...	...	...	...	...
995	8.87	44	1	3.099150	8.808934
996	9.12	65	1	3.505062	8.808934
997	4.89	34	0	-3.362960	5.113546
998	8.62	46	1	2.693239	8.620000
999	4.90	10	1	-3.346724	5.113546

985 rows × 5 columns

In [39]:

```
# Comparing

plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['placement_exam_marks'])

plt.subplot(2,2,2)
sns.boxplot(df['placement_exam_marks'])

plt.subplot(2,2,3)
sns.distplot(new_dff['placement_exam_marks'])

plt.subplot(2,2,4)
sns.boxplot(new_dff['placement_exam_marks'])

plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keywo
rd arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keywo
rd arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Capping

In [40]:

```
new_df_cap = df.copy()

new_df_cap['placement_exam_marks'] = np.where(
    new_df_cap['placement_exam_marks'] > upper_limit,
    upper_limit,
    np.where(
        new_df_cap['placement_exam_marks'] < lower_limit,
        lower_limit,
        new_df_cap['placement_exam_marks']
    )
)
```

In [44]:

```
new_df_cap.shape
```

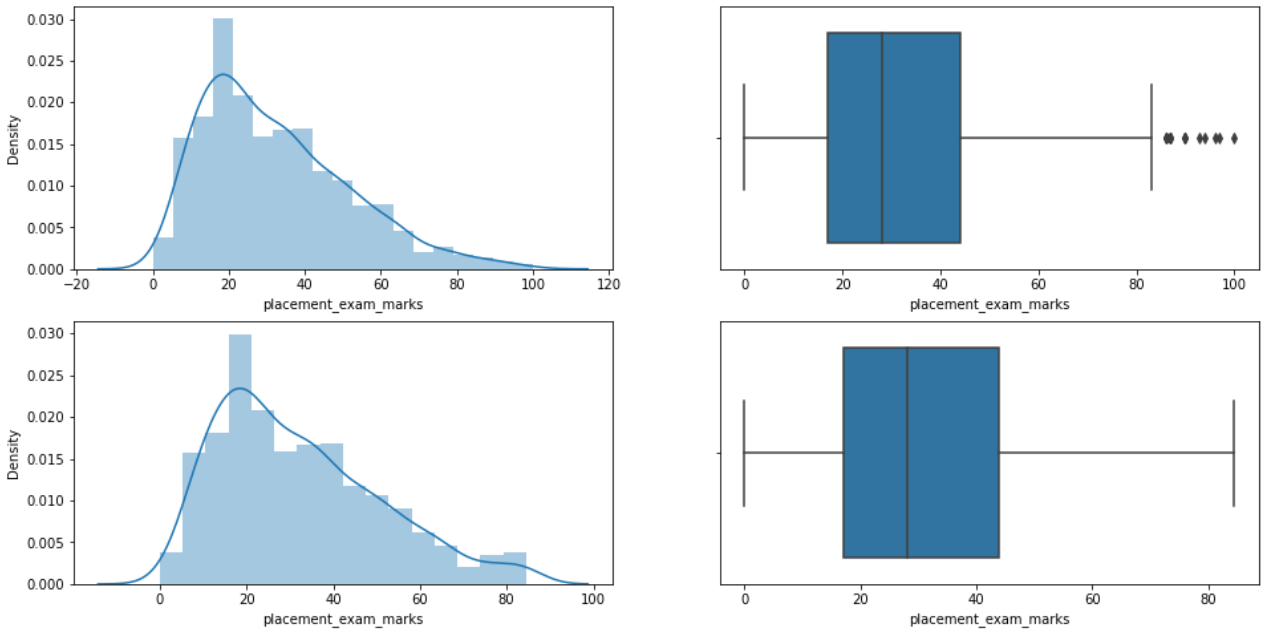
Out[44]:

```
(1000, 5)
```

In [43]:

```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df['placement_exam_marks'])

plt.subplot(2,2,2)
sns.boxplot(df['placement_exam_marks'])

plt.subplot(2,2,3)
sns.distplot(new_df_cap['placement_exam_marks'])

plt.subplot(2,2,4)
sns.boxplot(new_df_cap['placement_exam_marks'])

plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keywo
rd arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keywo
rd arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [ ]: