In [75]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [76]:
```
df = pd.read_csv('placement.csv')
```

In [77]:
```
df.shape
```

Out[77]: (1000, 3)

In [78]:
```
df.sample(5)
```

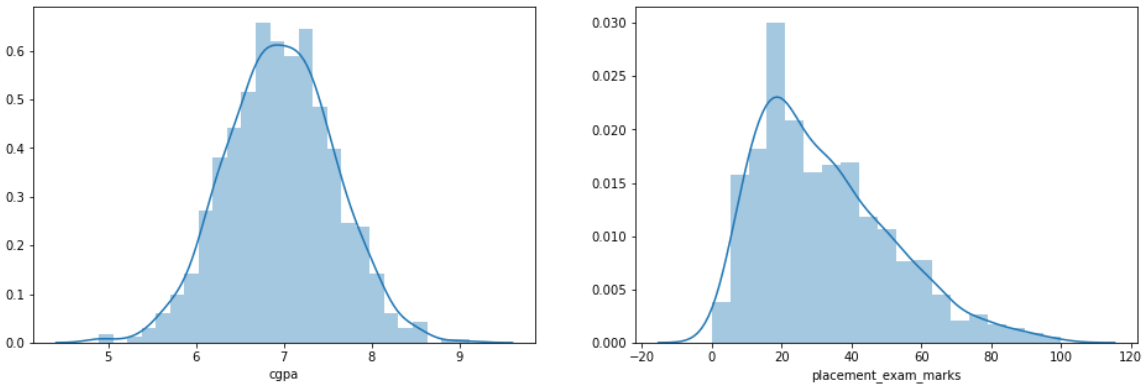
Out[78]:

	cgpa	placement_exam_marks	placed
689	8.02	67.0	0
111	6.48	33.0	0
991	7.04	57.0	0
835	6.67	65.0	1
772	6.63	26.0	0

In [79]:
```
plt.figure(figsize=(16,5))
plt.subplot(1,2,1)
sns.distplot(df['cgpa'])

plt.subplot(1,2,2)
sns.distplot(df['placement_exam_marks'])

plt.show()
```



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

In [57]:
```
print("Mean value of cgpa",df['cgpa'].mean())
print("Std value of cgpa",df['cgpa'].std())
```

```
print("Min value of cgpa",df['cgpa'].min())
print("Max value of cgpa",df['cgpa'].max())
```

```
Mean value of cgpa 6.96124000000001
Std value of cgpa 0.6158978751323894
Min value of cgpa 4.89
Max value of cgpa 9.12
```

In [58]:

```
# Finding the boundary values
print("Highest allowed",df['cgpa'].mean() + 3*df['cgpa'].std())
print("Lowest allowed",df['cgpa'].mean() - 3*df['cgpa'].std())
```

```
Highest allowed 8.808933625397177
Lowest allowed 5.113546374602842
```

In [59]:

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

Out[59]:

	cgpa	placement_exam_marks	placed
485	4.92	44.0	1
995	8.87	44.0	1
996	9.12	65.0	1
997	4.89	34.0	0
999	4.90	10.0	1

Trimming

In [60]:

```
# Trimming

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

Out[60]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0
...	...	...	...
991	7.04	57.0	0
992	6.26	12.0	0
993	6.73	21.0	1
994	6.48	63.0	0

998	8.62	46.0	1

995 rows × 3 columns

In [62]:
```
# Approach 2

# Calculating the Zscore

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

In [63]:
```
df.head()
```

Out[63]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
0	7.19	26.0	1	0.371425
1	7.46	38.0	1	0.809810
2	7.54	40.0	1	0.939701
3	6.42	8.0	1	-0.878782
4	7.23	17.0	0	0.436371

In [64]:
```
df[df['cgpa_zscore'] > 3]
```

Out[64]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
995	8.87	44.0	1	3.099150
996	9.12	65.0	1	3.505062

In [65]:
```
df[df['cgpa_zscore'] < -3]
```

Out[65]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
485	4.92	44.0	1	-3.314251
997	4.89	34.0	0	-3.362960
999	4.90	10.0	1	-3.346724

In [66]:
```
df[(df['cgpa_zscore'] > 3) | (df['cgpa_zscore'] < -3)]
```

Out[66]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
485	4.92	44.0	1	-3.314251
995	8.87	44.0	1	3.099150
996	9.12	65.0	1	3.505062
997	4.89	34.0	0	-3.362960

999	4.90	10.0	1	-3.346724

In [67]:
```
# Trimming
new_df = df[(df['cgpa_zscore'] < 3) & (df['cgpa_zscore'] > -3)]
```

In [68]:
```
new_df
```

Out[68]:

	cgpa	placement_exam_marks	placed	cgpa_zscore
0	7.19	26.0	1	0.371425
1	7.46	38.0	1	0.809810
2	7.54	40.0	1	0.939701
3	6.42	8.0	1	-0.878782
4	7.23	17.0	0	0.436371
...	...	...	...	...
991	7.04	57.0	0	0.127878
992	6.26	12.0	0	-1.138565
993	6.73	21.0	1	-0.375452
994	6.48	63.0	0	-0.781363
998	8.62	46.0	1	2.693239

995 rows × 4 columns

Capping

In [69]:
```
upper_limit = df['cgpa'].mean() + 3*df['cgpa'].std()
lower_limit = df['cgpa'].mean() - 3*df['cgpa'].std()
```

In [71]:
```
lower_limit
```

Out[71]:
```
5.113546374602842
```

In [72]:
```
df['cgpa'] = np.where(
    df['cgpa']>upper_limit,
    upper_limit,
    np.where(
        df['cgpa']<lower_limit,
        lower_limit,
        df['cgpa']
    )
)
```

In [73]:
```
df.shape
```

```
df.shape
```

Out[73]: (1000, 4)

In [74]:
```
df['cgpa'].describe()
```

Out[74]:
```
count    1000.000000
mean        6.961499
std         0.612688
min         5.113546
25%         6.550000
50%         6.960000
75%         7.370000
max         8.808934
Name: cgpa, dtype: float64
```

In [ ]:
```

```