```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import requests
from io import StringIO

url = "https://raw.githubusercontent.com/campusx-official/100-days-of-machine-learning/refs/heads/main/day42-outlier-removal-using-zscore/placement.csv"
headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:66.0) Gecko/20100101 Firefox/66.0"}
req = requests.get(url, headers=headers)
data = StringIO(req.text)
df = pd.read_csv(data)
```

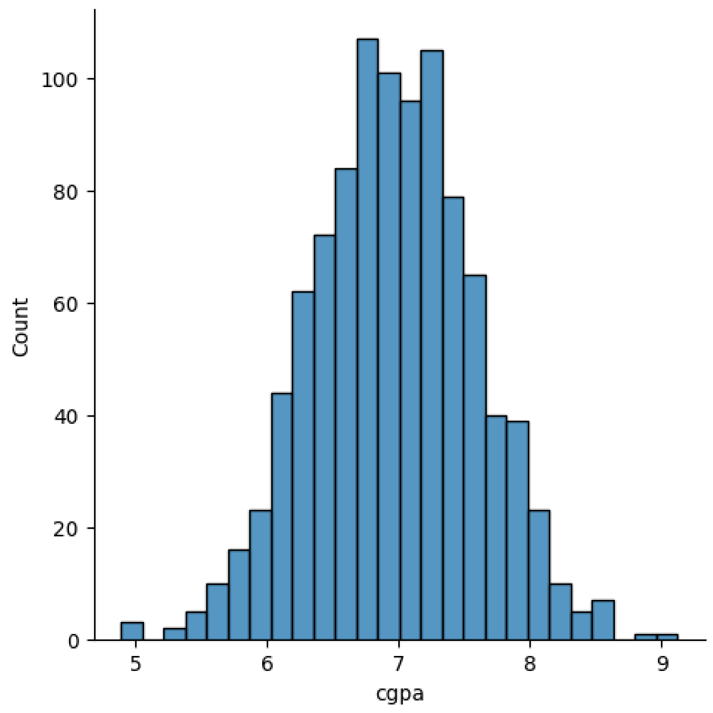
```
df.head()
```

	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

Next steps: Generate code with df New interactive sheet

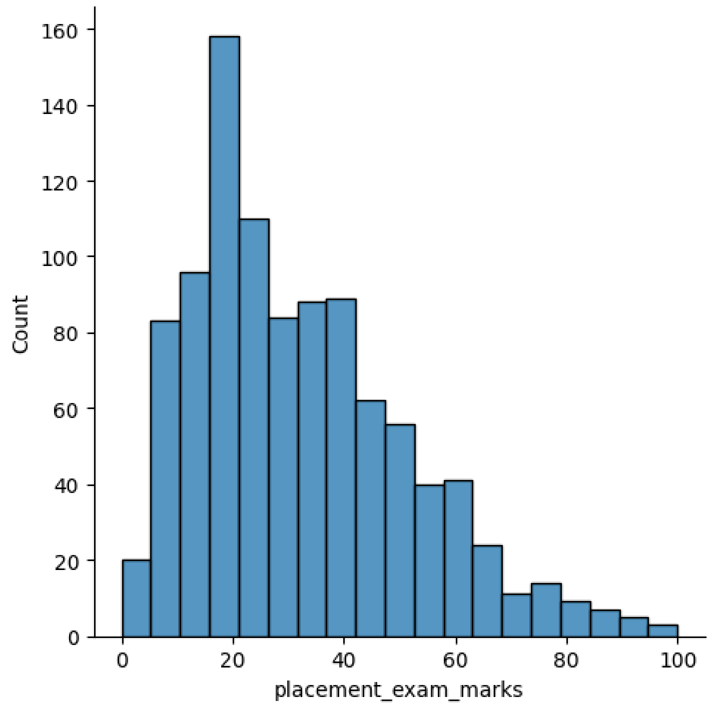
```
sns.displot(df['cgpa'])
```

```
<seaborn.axisgrid.FacetGrid at 0x7c15045e9250>
```



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

```
<seaborn.axisgrid.FacetGrid at 0x7c14eeb7bbc0>
```



Outlier detection using Zscore

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

```
df['cgpa_zscore'].head()
```

	cgpa_zscore
0	0.371425
1	0.809810
2	0.939701
3	-0.878782
4	0.436371

dtype: float64

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

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

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

	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

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

	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

Trimming

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

```
new_df.head()
```

	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

Next steps: Generate code with new_df New interactive sheet

capping

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

```
upper_limit
```

```
np.float64(8.808933625397168)
```

```
lower_limit
```

```
np.float64(5.113546374602832)
```

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

```
df.shape
```

```
(1000, 4)
```

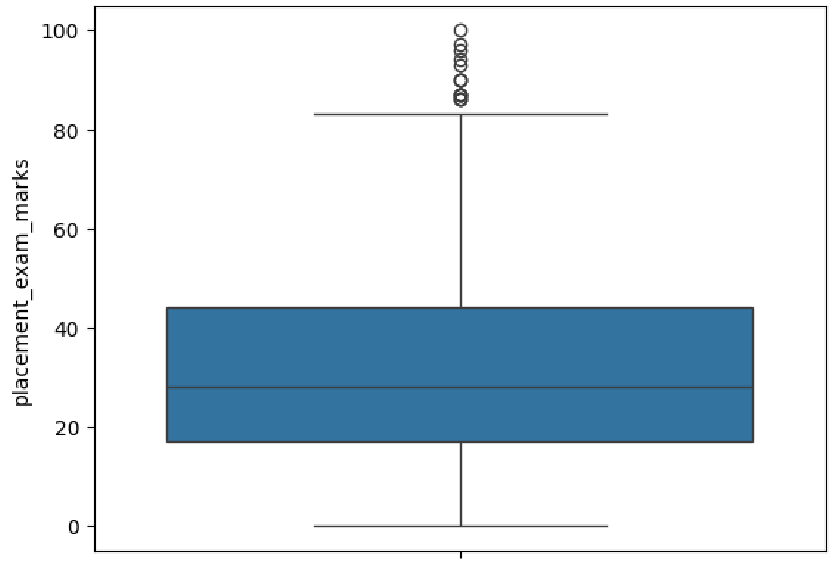
```
df['cgpa'].describe()
```

```
             cgpa
count  1000.000000
 mean     6.961499
  std     0.612688
  min     5.113546
  25%     6.550000
  50%     6.960000
  75%     7.370000
  max     8.808934

dtype: float64
```

Outlier detection using IQR

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

```
<Axes: ylabel='placement_exam_marks'>
```



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

```
       placement_exam_marks
count           1000.000000
 mean             32.225000
  std             19.130822
  min              0.000000
  25%             17.000000
  50%             28.000000
  75%             44.000000
  max            100.000000

dtype: float64
```

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

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

```
np.float64(27.0)
```

```
upper_limit = percentile75 + (1.5 * iqr)
lower_limit = percentile25 - (1.5 * iqr)
print('upper_limit:',upper_limit)
print('lower_limit',lower_limit)
```

```
upper_limit: 84.5
lower_limit -23.5
```

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

	cgpa	placement_exam_marks	placed	cgpa_zscore
9	7.75	94.0	1	1.280667
40	6.60	86.0	1	-0.586526
61	7.51	86.0	0	0.890992
134	6.33	93.0	0	-1.024910
162	7.80	90.0	0	1.361849
283	7.09	87.0	0	0.209061
290	8.38	87.0	0	2.303564
311	6.97	87.0	1	0.014223
324	6.64	90.0	0	-0.521580
630	6.56	96.0	1	-0.651472
685	6.05	87.0	1	-1.479531
730	6.14	90.0	1	-1.333403
771	7.31	86.0	1	0.566263
846	6.99	97.0	0	0.046696
917	5.95	100.0	0	-1.641896

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

	cgpa	placement_exam_marks	placed	cgpa_zscore

Trimming

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

```
new_df
```

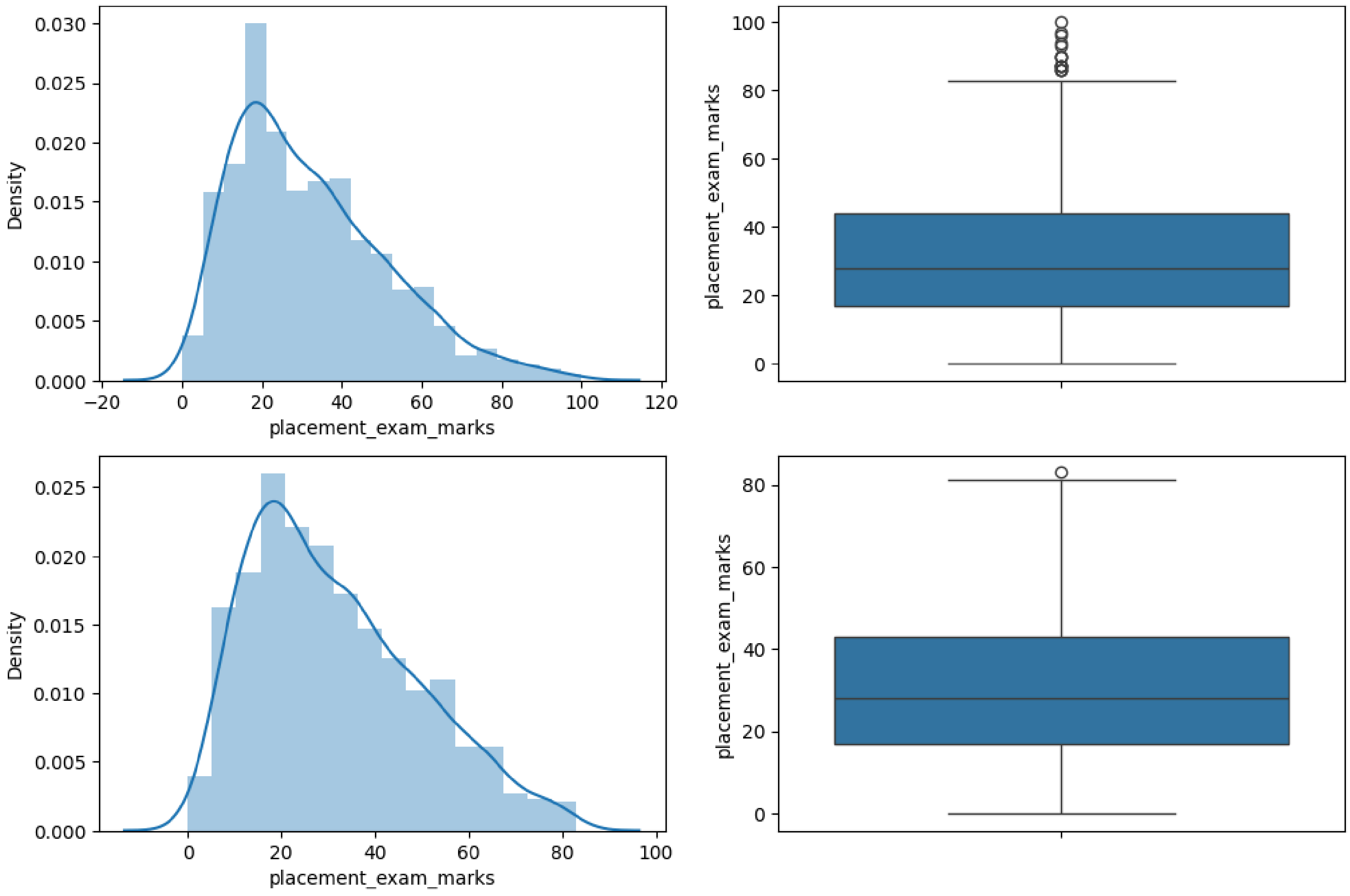
	cgpa	placement_exam_marks	placed	cgpa_zscore
0	7.190000	26.0	1	0.371425
1	7.460000	38.0	1	0.809810
2	7.540000	40.0	1	0.939701
3	6.420000	8.0	1	-0.878782
4	7.230000	17.0	0	0.436371
...	...	...	...	...
995	8.808934	44.0	1	3.099150
996	8.808934	65.0	1	3.505062
997	5.113546	34.0	0	-3.362960
998	8.620000	46.0	1	2.693239
999	5.113546	10.0	1	-3.346724

985 rows × 4 columns

Next steps: Generate code with new_df New interactive sheet

```
plt.figure(figsize=(12,8))
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['placement_exam_marks'])

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

plt.show()
```

```
/tmp/ipython-input-628301363.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['placement_exam_marks'])
/tmp/ipython-input-628301363.py:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(new_df['placement_exam_marks'])
```



Capping

```
new_df = df.copy()
```

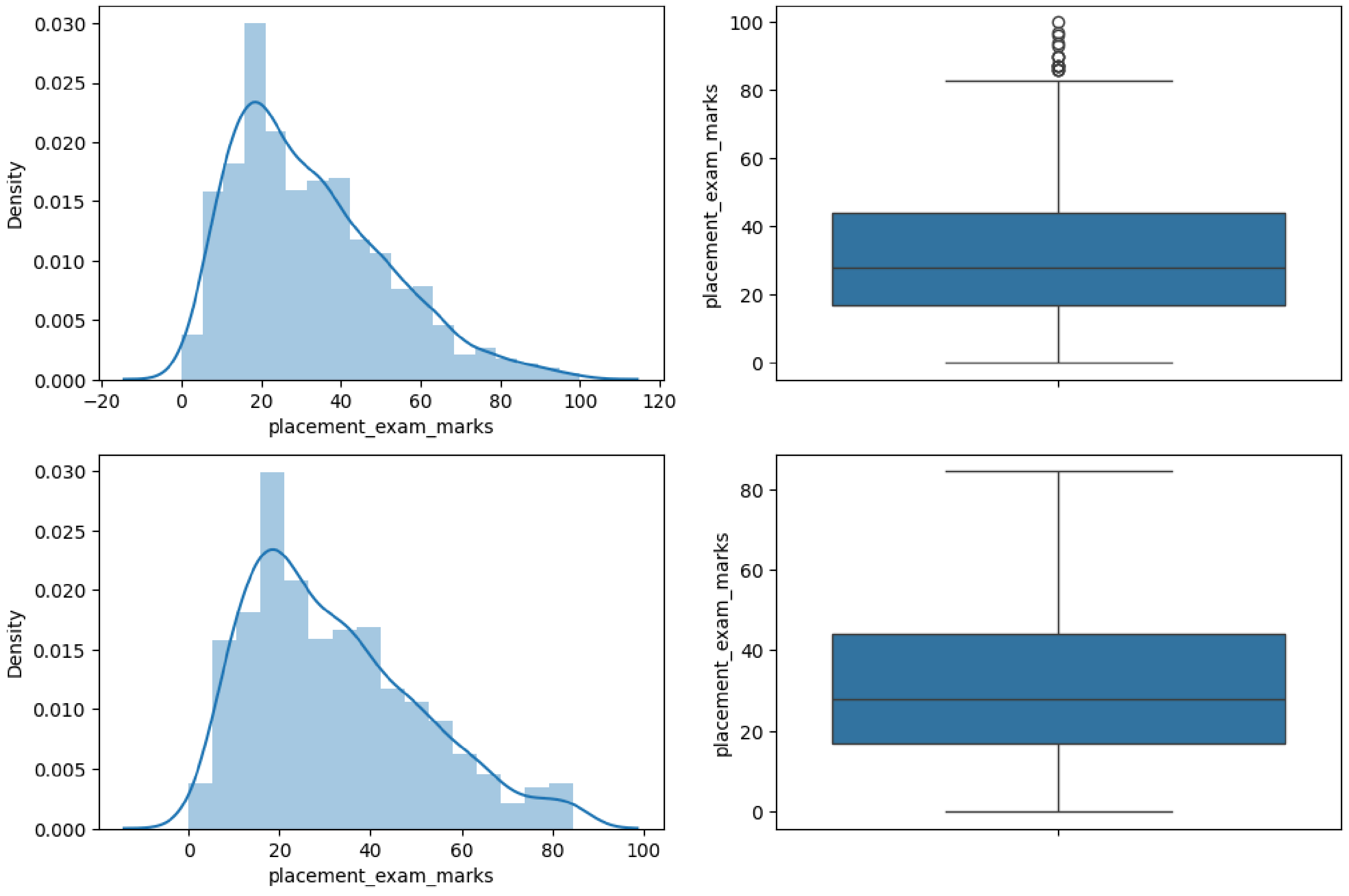
```
new_df['placement_exam_marks'] = np.where(
    new_df['placement_exam_marks'] > upper_limit,
    upper_limit ,
    np.where(
        new_df['placement_exam_marks'] < lower_limit ,
        lower_limit ,
        new_df['placement_exam_marks']
    )
)
```

```
new_df.shape
```

```
(1000, 4)
```

```
plt.figure(figsize=(12,8))
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['placement_exam_marks'])

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

plt.show()
```

```
/tmp/ipython-input-628301363.py:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['placement_exam_marks'])
/tmp/ipython-input-628301363.py:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(new_df['placement_exam_marks'])
```



Outlier detection using Percentile

```
df.head()
```

	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

Next steps: Generate code with df New interactive sheet

```
#upper & lower limit for cgpa
upper_limit_cgpa = df['cgpa'].quantile(0.99)
lower_limit_cgpa = df['cgpa'].quantile(0.01)
```

```
#upper & lower limit for marks
upper_limit_marks = df['placement_exam_marks'].quantile(0.99)
lower_limit_marks = df['placement_exam_marks'].quantile(0.01)
```

Trimming

```
new_df = df[(df['cgpa'] < upper_limit_cgpa) | (df['cgpa'] > lower_limit_cgpa)]
new_df
```

	cgpa	placement_exam_marks	placed	cgpa_zscore
0	7.190000	26.0	1	0.371425
1	7.460000	38.0	1	0.809810
2	7.540000	40.0	1	0.939701
3	6.420000	8.0	1	-0.878782
4	7.230000	17.0	0	0.436371
...	...	...	...	...
995	8.808934	44.0	1	3.099150
996	8.808934	65.0	1	3.505062
997	5.113546	34.0	0	-3.362960
998	8.620000	46.0	1	2.693239