Name :sandip gadadare. Div:A
Roll no: CO3AO9. BATCH:B1

---

**Assignment No : 02**

**Create an "Academic performance" dataset of students and perform the following operations using Python.**
**1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them.**
**2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them.**
**3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution.**

**Code :**

```
In [1]:
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import zscore
from scipy.stats import boxcox
```

```
In [2]:
data={
    'stud_id':range(1,10),
    'CNS-marks':[90,56,78,89,np.nan,77,84,67,np.nan],
    'DSBDA-marks':[97,91,90,56,78,89,np.nan,77,84],
    'Attendance':[90,56,78,89,65,77,84,77,67]
}
```

```
In [3]:
data
Out[3]:
{'stud_id': range(1, 10),
 'CNS-marks': [90, 56, 78, 89, nan, 77, 84, 67, nan],
 'DSBDA-marks': [97, 91, 90, 56, 78, 89, nan, 77, 84],
 'Attendance': [90, 56, 78, 89, 65, 77, 84, 77, 67]}
```

```
In [4]:
df=pd.DataFrame(data)
```

In [5]:
```
df
```
Out[5]:

| | stud_id | CNS-marks | DSBDA-marks | Attendance |
|---|---|---|---|---|
| **0** | 1 | 90.0 | 97.0 | 90 |
| **1** | 2 | 56.0 | 91.0 | 56 |
| **2** | 3 | 78.0 | 90.0 | 78 |
| **3** | 4 | 89.0 | 56.0 | 89 |
| **4** | 5 | NaN | 78.0 | 65 |
| **5** | 6 | 77.0 | 89.0 | 77 |
| **6** | 7 | 84.0 | NaN | 84 |
| **7** | 8 | 67.0 | 77.0 | 77 |
| **8** | 9 | NaN | 84.0 | 67 |

In [6]:
```
  df.isnull().sum()
```
Out[6]:
```
stud_id        0
CNS-marks      2
DSBDA-marks    1
Attendance     0
dtype: int64
```

In [7]:
```
df['CNS-marks'].fillna(df['CNS-marks'].mean(),inplace=True)
```

In [8]:
```
df['DSBDA-marks'].fillna(df['DSBDA-marks'].mean(),inplace=True)
```

In [9]:
```
df
```
Out[9]:

| | stud_id | CNS-marks | DSBDA-marks | Attendance |
|---|---|---|---|---|
| **0** | 1 | 90.000000 | 97.00 | 90 |
| **1** | 2 | 56.000000 | 91.00 | 56 |
| **2** | 3 | 78.000000 | 90.00 | 78 |
| **3** | 4 | 89.000000 | 56.00 | 89 |
| **4** | 5 | 77.285714 | 78.00 | 65 |
| **5** | 6 | 77.000000 | 89.00 | 77 |
| **6** | 7 | 84.000000 | 82.75 | 84 |

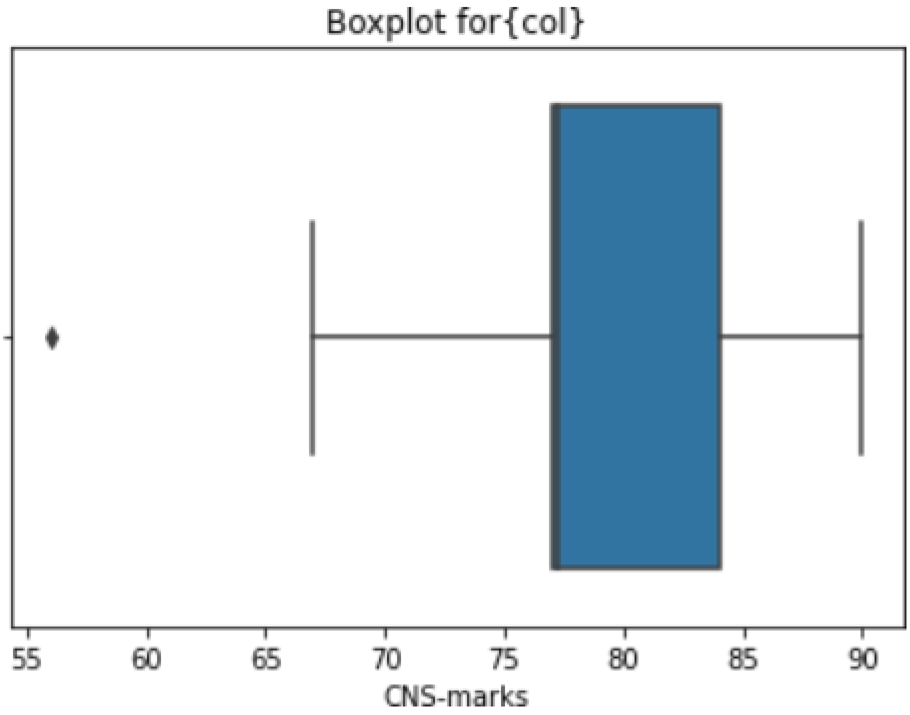
|   | stud_id | CNS-marks | DSBDA-marks | Attendance |
|---|---|---|---|---|
| **7** | 8 | 67.000000 | 77.00 | 77 |
| **8** | 9 | 77.285714 | 84.00 | 67 |

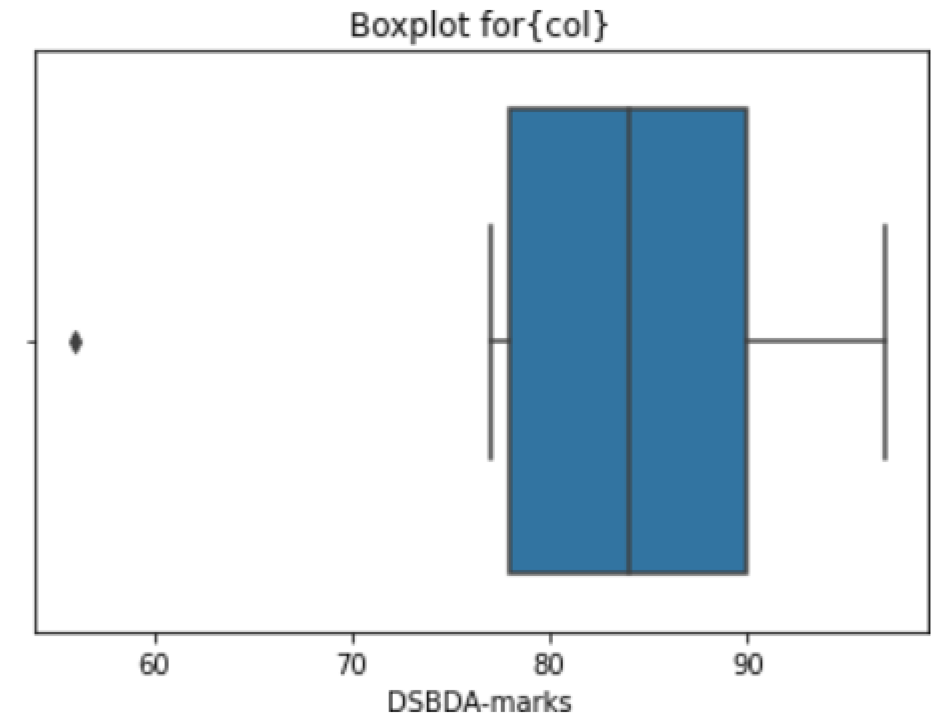
In [10]:

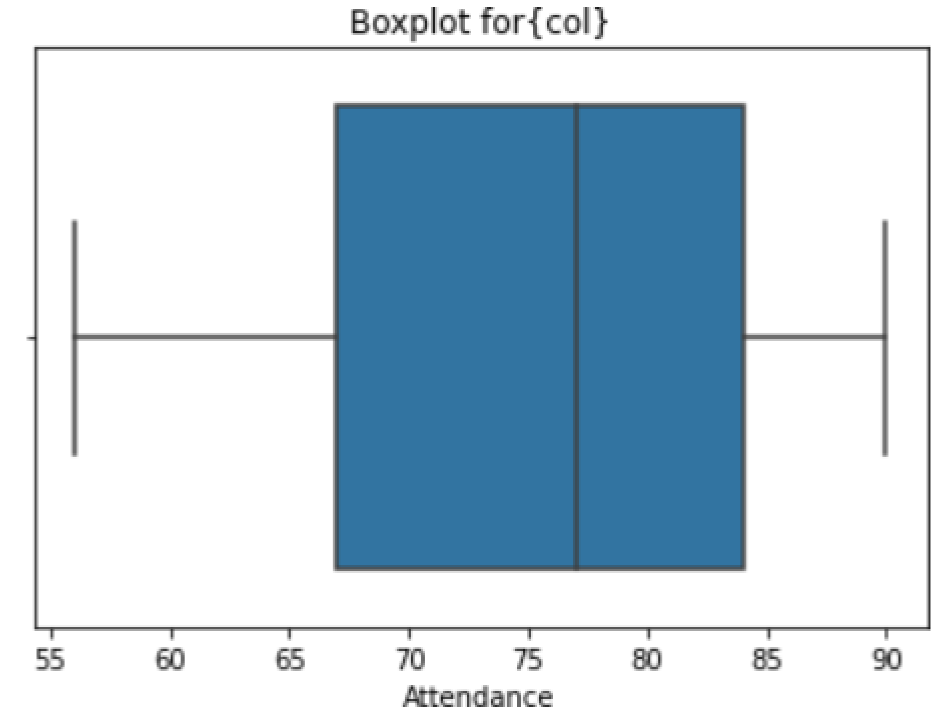
```
column=['CNS-marks','DSBDA-marks','Attendance']
```

In [11]:

```
for col in column:
   sns.boxplot(df[col])
   plt.title("Boxplot for{col}")
   plt.show()
```







In [12]:

```
z_scores = np.abs(zscore(df[column]))
```

In [13]:

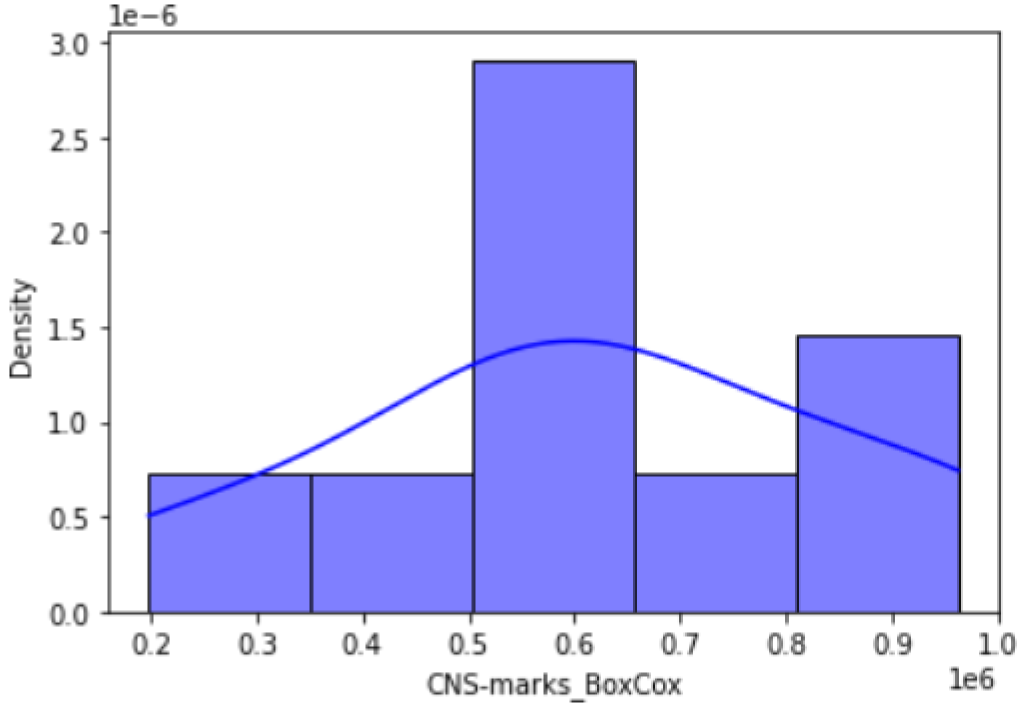
```
outlier_threshold=3
```

```
In [14]:
outliers=(z_scores>outlier_threshold).any(axis=1)
```

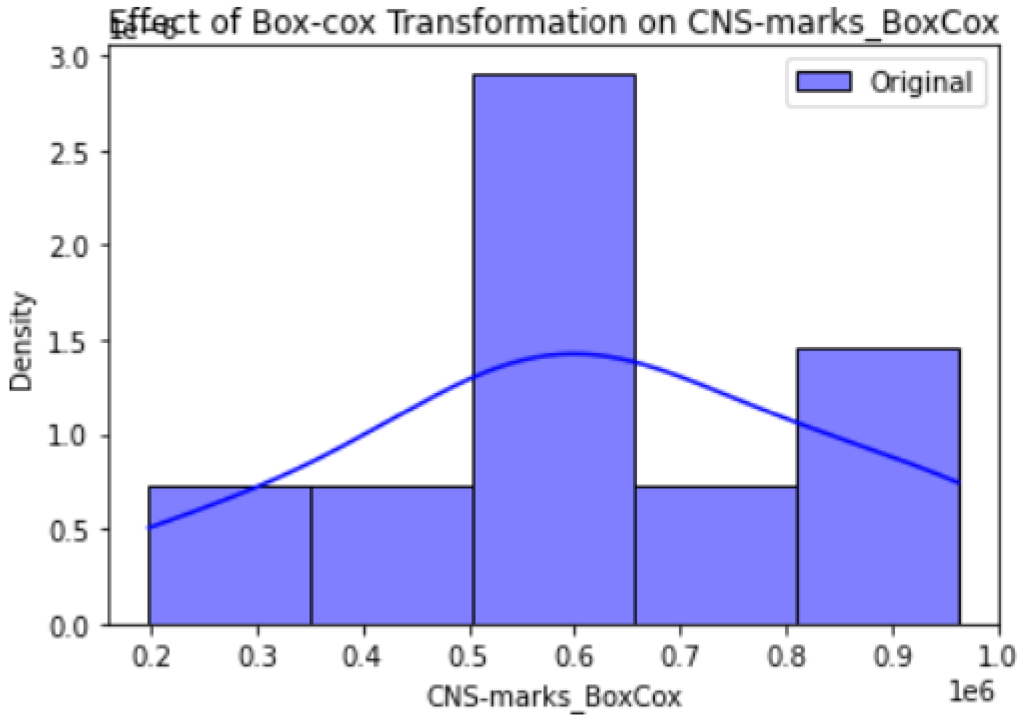
```
In [15]:
print(f"\nRows with outliers:\n{df[outliers]}")
Rows with outliers:
Empty DataFrame
Columns: [stud_id, CNS-marks, DSBDA-marks, Attendance]
Index: []
```

```
In [22]:
df['CNS-marks_BoxCox'],lambda_val=boxcox(df['CNS-marks'])
print(f"\nBox-cox transformation applied to 'Study_Hours' with lambda = {lambda_val:.4f}")
Box-cox transformation applied to 'Study_Hours' with lambda = 3.3290
```

```
In [24]:
sns.histplot(df['CNS-marks_BoxCox'], kde=True, color='blue',label='Original',
stat="density")
Out[24]:
<AxesSubplot:xlabel='CNS-marks_BoxCox', ylabel='Density'>
```



```
In [25]:
sns.histplot(df['CNS-marks_BoxCox'], kde=True, color='blue',label='Original',
stat="density")
plt.legend()
plt.title("Effect of Box-cox Transformation on CNS-marks_BoxCox")
plt.show()
print("\nFinal Dataset after Data Wrangling:")
print(df)
```



```
Final Dataset after Data Wrangling:
   stud_id  CNS-marks  DSBDA-marks  Attendance  CNS-marks_BoxCox
0        1  90.000000        97.00          90     962411.544161
1        2  56.000000        91.00          56     198337.468953
2        3  78.000000        90.00          78     597682.184000
3        4  89.000000        56.00          89     927271.331681
4        5  77.285714        78.00          65     579655.136186
5        6  77.000000        89.00          77     572552.068616
6        7  84.000000        82.75          84     764915.029804
7        8  67.000000        77.00          77     360321.519483
8        9  77.285714        84.00          67     579655.136186
```