

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import skew, kurtosis
import seaborn as sns #This is for generating Histogram with Ker
```

```
import pandas as pd
```

```
df = pd.read_csv ('hotel_books.csv') #read the 'hotel_books.csv' file
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
df.dtypes #check for data types
```

```

0
day      int64
clients  int64
total_bill int64
```

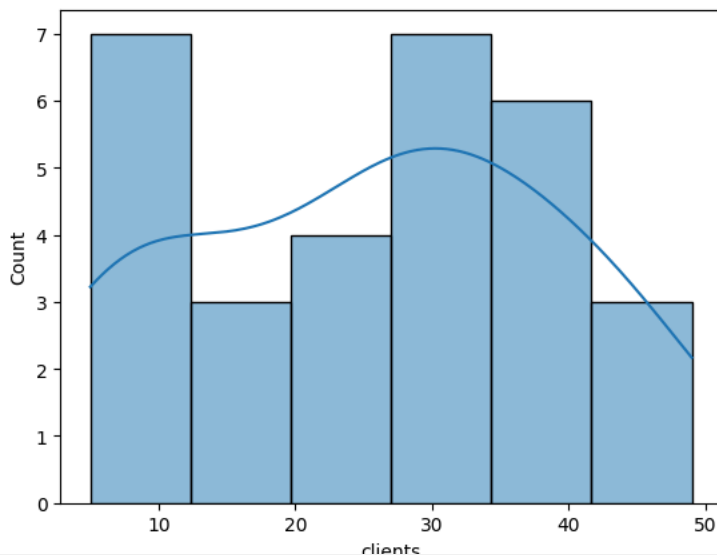
```
df.isnull().sum() #check for missing values
```

```

0
day      0
clients  0
total_bill 0
```

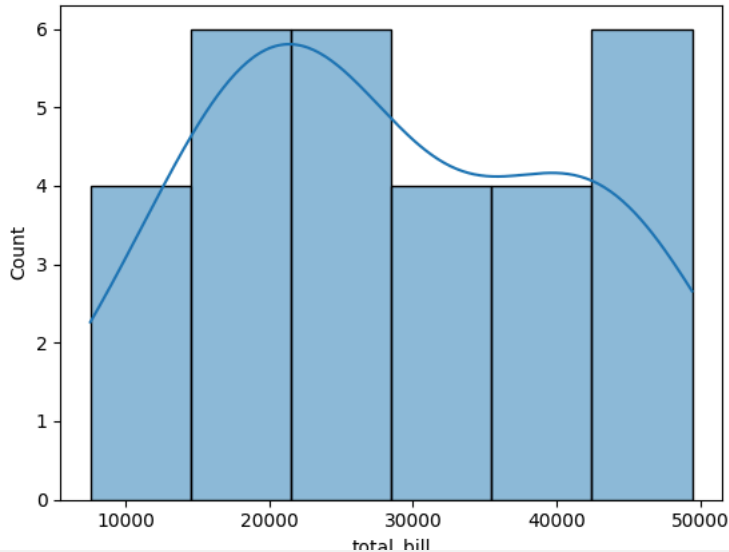
```
sns.histplot(df['clients'], kde=True) #generate histogram with kernel density estimate (KDE) for number of hotel clients
```

<Axes: xlabel='clients', ylabel='Count'>



```
sns.histplot(df['total_bill'], kde=True) #generate histogram with kernel density estimate (KDE) for total bill connected
```

<Axes: xlabel='total_bill', ylabel='Count'>



```
#compute for skewness and kurtosis for number of clients
skew1 = df['clients'].skew()
kurt1 = df['total_bill'].kurt()
print(f'Kurtosis for the number of hotel clients in a day:{kurt1}')
print(f'Skewness for the number of hotel clients in a day:{skew1}')
```

Kurtosis for the number of hotel clients in a day:-1.130219880444574
Skewness for the number of hotel clients in a day:-0.05968808896371035

```
#compute for skewness and kurtosis for total number of bill
skew2 = df['clients'].skew()
kurt2 = df['total_bill'].kurt()
print(f'Kurtosis for the total bill collected from clients per day:{kurt2}')
print(f'Skewness for the total bill collected from clients per day:{skew2}')
```

Kurtosis for the total bill collected from clients per day:-1.130219880444574
Skewness for the total bill collected from clients per day:-0.05968808896371035

```
df.describe() #generate summary measure and observe the mean and 50% (median)
```

	day	clients	total_bill
count	30.000000	30.000000	30.000000
mean	15.500000	25.666667	28344.233333
std	8.803408	13.557879	12441.769892
min	1.000000	5.000000	7534.000000
25%	8.250000	16.000000	18335.000000
50%	15.500000	28.000000	25841.500000
75%	22.750000	35.750000	39810.250000
max	30.000000	40.000000	40450.000000

```
stats.mode(df['clients']) #compute for mode
```

ModeResult(mode=8, count=4)

```
stats.mode(df['total_bill']) #compute for mode
```

ModeResult(mode=7534, count=1)

```
df.dtypes #check for data types
```

```
df.info()
Out[1]:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30 entries, 0 to 29
Data columns (total 4 columns):
 #   column  non-null    dtype  object
 0   day      30         int64  int64
 1   clients  30         int64  int64
 2   total_bill  30        int64  int64
```

```
df.isnull().sum() #check for missing values
```

```
Out[2]:
day      0
clients  0
total_bill  0
```

```
#check dataset again for confirmation
df.head(30)
```

```
Out[3]:
   day  clients  total_bill
0     1         33      23958
1     2         25      26812
2     3          5      24871
3     4         17      17954
4     5         28      29416
5     6         16      19478
6     7         41      12904
7     8          8      49386
8     9          5      22155
9    10         36       7534
10   11         46      35116
11   12          7      30492
12   13         22      14039
13   14         29      20932
14   15         30      42492
15   16         20      40749
16   17         38       8733
17   18         23      44654
18   19          8      17652
19   20         28      21636
20   21         41      16992
21   22         16      15366
22   23          8      49450
23   24         35      23499
24   25         49      37999
25   26          8      37056
26   27         31      43434
27   28         41      46577
28   29         30      28577
29   30         46      40414
```

