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

```
df = pd.read_csv('hotel_books.csv') #uread the 'hotel_books.csv' file
df.head(30)
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 30,\n  \"fields\": [\n
{\n      \"column\": \"day\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 8,\n        \"min\": 1,\n
\"max\": 30,\n        \"num_unique_values\": 30,\n        \"samples\":
[\n          28,\n          16,\n          24\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"clients\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 13,\n
\"min\": 5,\n        \"max\": 49,\n        \"num_unique_values\": 20,\
n        \"samples\": [\n          33,\n          35,\n          38\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"total_bill\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
12441,\n        \"min\": 7534,\n        \"max\": 49450,\n
\"num_unique_values\": 30,\n        \"samples\": [\n          46577,\n
40749,\n          23499\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

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

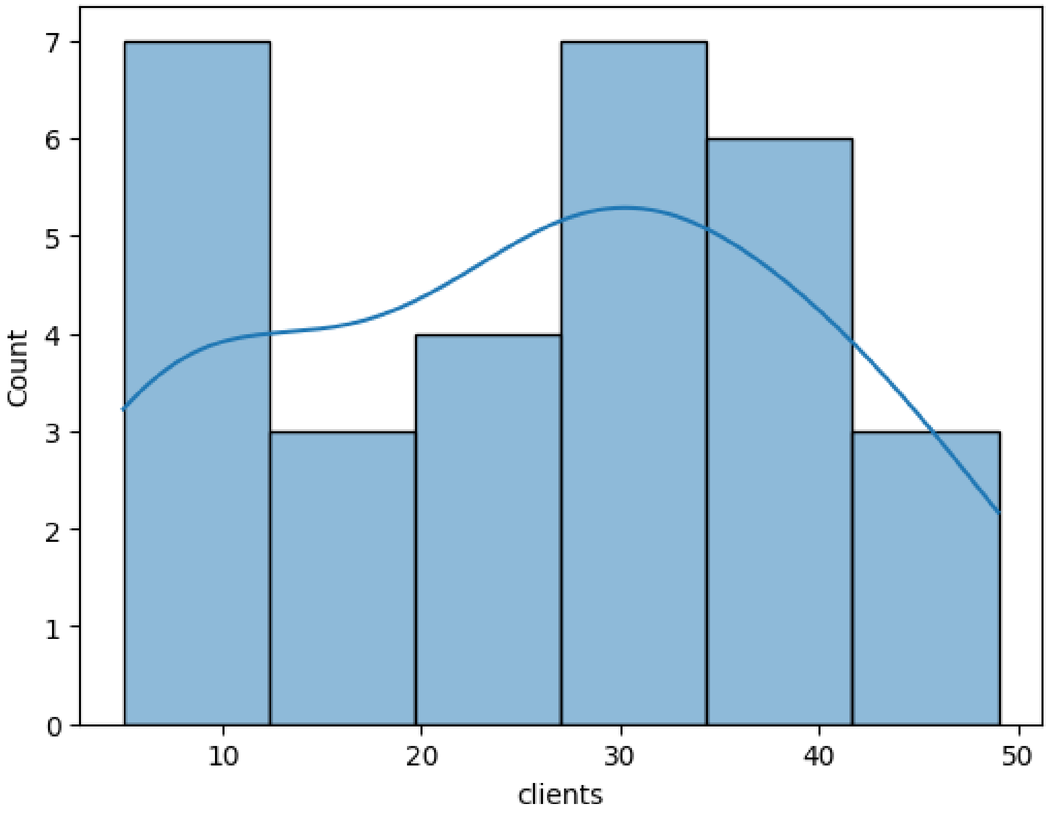
```
day              int64
clients          int64
total_bill       int64
dtype: object
```

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

```
day              0
clients          0
total_bill       0
dtype: int64
```

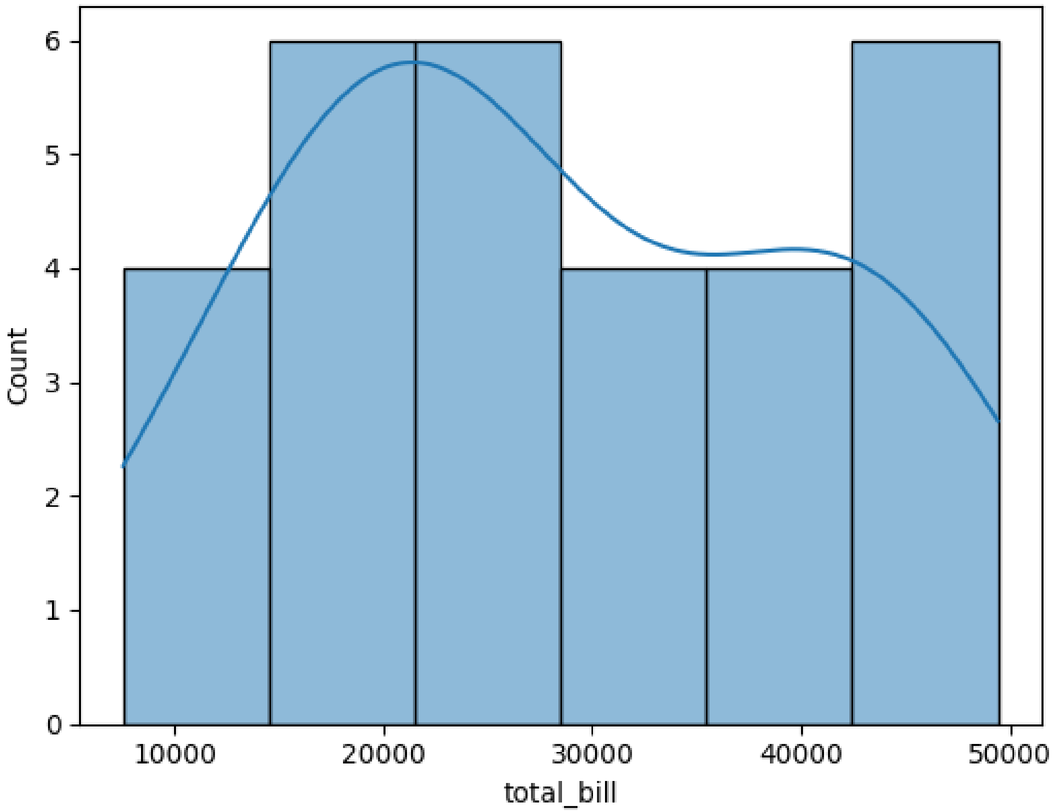
```
sns.histplot(df['clients'], kde=True) #generate histogram with kernel
density estimate (KDE) for number of hotel clients
```

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



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

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



```
#compute for skewness and kurtosis for number of clients
skew1 = df['clients'].skew()
kurt1 = df['clients'].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:-1.1388703400867874
Skewness for the number of hotel clients in a day:-0.05968808896371035

#compute for skewness and kurtosis for total number of bill
skew2 = df['total_bill'].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:0.18976914965853053

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

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"day\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 10.50923193300601,\n
\"min\": 1.0,\n        \"max\": 30.0,\n        \"num_unique_values\":
6,\n        \"samples\": [\n          30.0,\n          15.5,\n
22.75\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"clients\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 13.819360535969686,\n        \"min\": 5.0,\n
\"max\": 49.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          25.666666666666668,\n          28.0,\n
30.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total_bill\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 16554.093812127518,\n        \"min\":
30.0,\n        \"max\": 49450.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          28344.233333333334,\n          25841.5,\n
30.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```
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)
```