

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

## ✓ TASK : Exploratory Data Analysis (EDA)

Load Prepared Data: Load the cleaned and integrated data from Component 1 into Python.

Visualize Data: Use libraries such as Matplotlib and Seaborn to create visualizations that reveal booking trends, seasonal patterns, and high-demand classes.

Statistical Analysis: Perform descriptive statistics to summarize data features (e.g., mean, median, standard deviation). Identify outliers and anomalies using statistical tests or visual methods like box plots.

### ✓ Load Prepared Data:

Load the cleaned and integrated data from Component 1 into Python.

```
df = pd.read_csv('Merged_Data_Avinash.xlsx', index_col='Unnamed: 0')
```

```
df.head()
```



	ActivitySiteID	ActivityDescription	BookingEndDateTime	BookingStartTime	MaxBook
0	HXP	20-20-20 2.45pm-3.45pm	2018-04-08	14:45:00	-0.483
1	HXP	20-20-20 2.45pm-3.45pm	2018-04-15	14:45:00	-0.483
2	HXP	20-20-20 2.45pm-3.45pm	2018-04-22	14:45:00	-0.483

```
# drop unnecessary columns
```

```
df.drop(['ActivityDescription', 'MaxBookees', 'BookingStartTime'], axis=1, inplace=True)
```

```
# encoding ActivitySiteID
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
df['ActivitySiteID'] = le.fit_transform(df['ActivitySiteID'])
```

```
df['BookingEndDateTime'] = pd.to_datetime(df['BookingEndDateTime'])
```

```
df.head()
```



	ActivitySiteID	BookingEndDateTime	Number Booked	Price (INR)
0	1	2018-04-08	-0.593867	-1.692847
1	1	2018-04-15	-0.282487	-1.692847
2	1	2018-04-22	-0.386280	-1.692847
3	1	2018-04-29	-0.905246	-1.692847
4	1	2018-05-06	-1.112833	-1.692847

### ✓ Visualize Data:

Use libraries such as Matplotlib and Seaborn to create visualizations that reveal booking trends, seasonal patterns, and high-demand classes.

```
# Plotting the number of bookings over time
```

```
plt.figure(figsize=(12, 6))
```

```
sns.lineplot(data=df, x='BookingEndDateTime', y='Number Booked')
```

```
plt.title('Booking Trends Over Time')
```

```
plt.xlabel('Date')
```

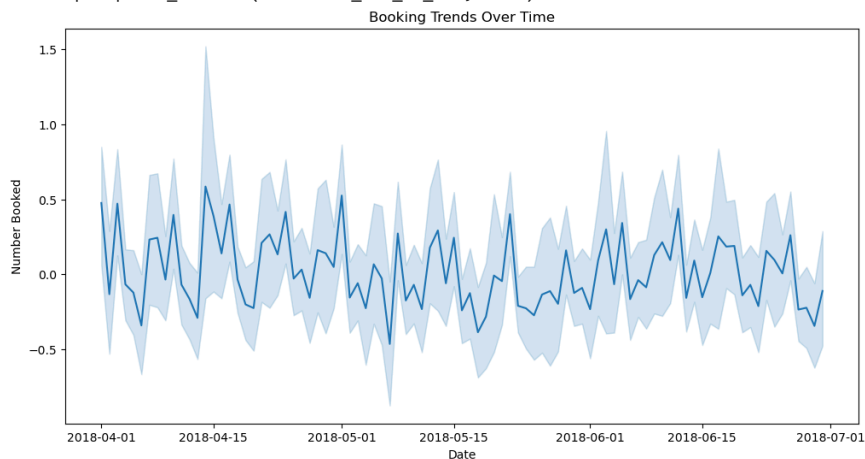
```
plt.ylabel('Number Booked')
```

```
plt.show()
```

```

c:\Users\SUNIL\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
with pd.option_context('mode.use_inf_as_na', True):
c:\Users\SUNIL\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
with pd.option_context('mode.use_inf_as_na', True):

```



## Statistical Analysis:

Perform descriptive statistics to summarize data features (e.g., mean, median, standard deviation). Identify outliers and anomalies using statistical tests or visual methods like box plots.

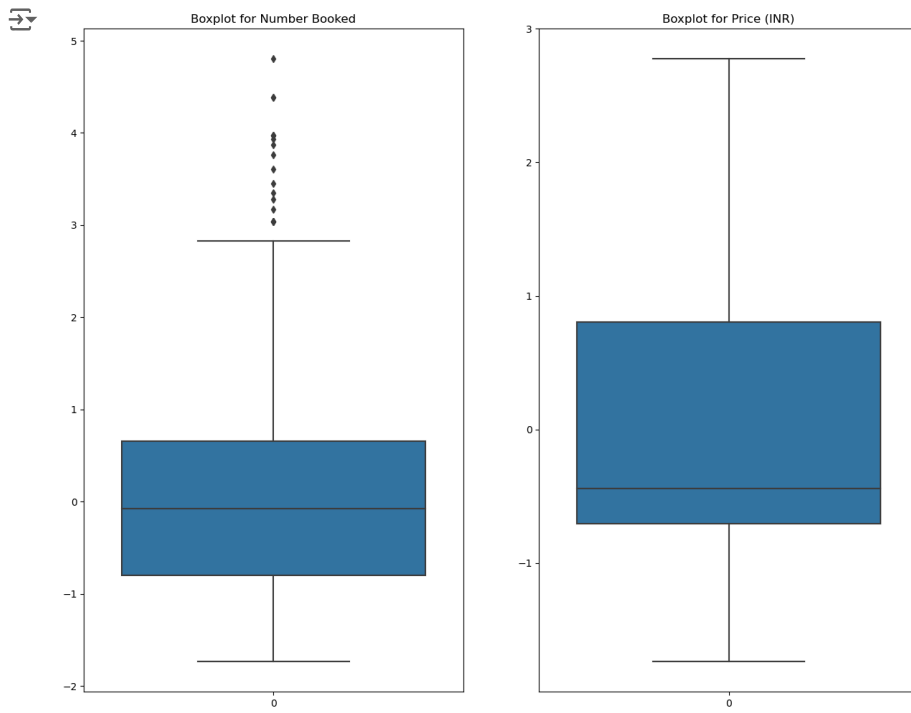
```
df.describe()
```

	ActivitySiteID	BookingEndDateTime	Number Booked	Price (INR)
count	3271.000000	3271	3.271000e+03	3.271000e+03
mean	1.856924	2018-05-16 11:00:47.324977152	-8.906222e-17	1.477130e-16
min	0.000000	2018-04-01 00:00:00	-1.735592e+00	-1.739450e+00
25%	1.000000	2018-04-24 00:00:00	-8.014531e-01	-7.075652e-01
50%	2.000000	2018-05-16 00:00:00	-7.490036e-02	-4.447742e-01
75%	3.000000	2018-06-08 00:00:00	6.579793e-01	8.032989e-01
max	4.000000	2018-06-30 00:00:00	4.803382e+00	2.775048e+00
std	1.398453	NaN	1.000153e+00	1.000153e+00

```

plt.figure(figsize=(15,12))
num_columns = ['Number Booked', 'Price (INR)']
for i, col in enumerate(num_columns, 1):
    plt.subplot(1,2,i)
    sns.boxplot(df[col])
    plt.title(f'Boxplot for {col}')

```

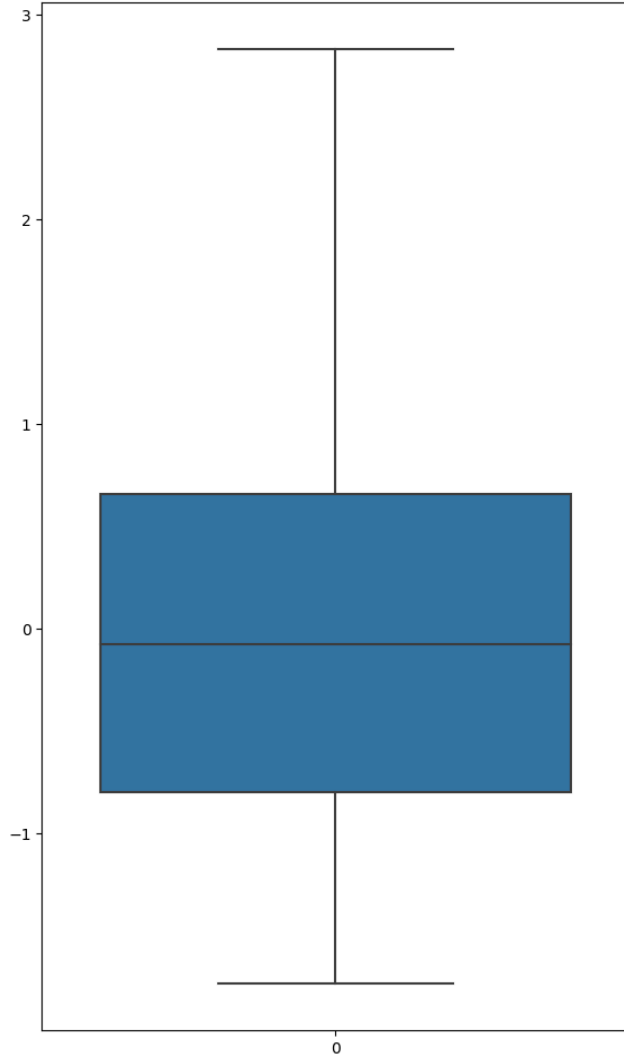


```
Q1 = df['Number Booked'].quantile(0.25)
Q3 = df['Number Booked'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5*IQR
upper_bound = Q3 + 1.5*IQR
df = df[(df['Number Booked'] > lower_bound) & (df['Number Booked'] < upper_bound)]
```

```
plt.figure(figsize=(15,12))
num_columns = ['Number Booked', 'Price (INR)']
for i, col in enumerate(num_columns, 1):
    plt.subplot(1,2,i)
    sns.boxplot(df[col])
    plt.title(f'Boxplot for {col}')
```



Boxplot for Number Booked



Boxplot for Price (INR)

