In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [22]:
```python
df=pd.read_excel(r"C:\Users\HP\Downloads\Hospitality (1).xlsx")
```

In [23]:
```python
df.head()
```

Out[23]:

	Avg Room Rate	reservation_id	check_in_date	stay_duration	adults	children	room_type	special_requests_flag	booking_channel	reservation_status	advanced_booking	Property	Date	Rate Type
0	71.10	779087-Y5-9824-SA	4/15/2020	13	3	4	Single	Yes	Call Center	Completed	Yes	The Chord	4/15/2020	Weekday
1	71.10	984023-QO-5015-YG	2020-12-06 00:00:00	3	3	3	Single	No	Travel Agent	Completed	Yes	The Chord	2020-12-06 00:00:00	Weekday
2	172.38	518066-UQ-2315-FK	2/25/2020	11	4	2	Queen	No	Call Center	Completed	Yes	The Sankey	2/25/2020	Weekday
3	172.38	130339-H9-2116-KE	9/15/2020	11	4	1	Queen	No	Website	No-Show	Yes	The Sankey	9/15/2020	Weekday
4	199.00	961051-40-0956-EO	2020-05-01 00:00:00	14	2	4	Double	Yes	Walk-in	Completed	No	The Sankey	2020-05-01 00:00:00	Weekend

In [24]:
```python
df.columns
```

Out[24]:
```
Index(['Avg Room Rate', 'reservation_id', 'check_in_date', 'stay_duration',
       'adults', 'children', 'room_type', 'special_requests_flag',
       'booking_channel', 'reservation_status', 'advanced_booking', 'Property',
       'Date', 'Rate Type'],
      dtype='object')
```

In [25]:
```python
df.describe()
```

Out[25]:

	Avg Room Rate	stay_duration	adults	children
count	50000.000000	50000.000000	50000.000000	50000.000000
mean	147.147144	7.473600	2.507560	2.493260
std	48.316584	4.050861	1.119164	1.117434
min	71.100000	1.000000	1.000000	1.000000
25%	103.950000	4.000000	2.000000	1.000000
50%	152.100000	7.000000	3.000000	2.000000
75%	177.450000	11.000000	4.000000	3.000000
max	288.550000	14.000000	4.000000	4.000000

In [26]:
```python
df.isnull().sum()
```

Out[26]:
```
Avg Room Rate            0
reservation_id           0
check_in_date            0
stay_duration            0
adults                   0
children                 0
room_type                0
special_requests_flag    0
booking_channel          0
reservation_status       0
advanced_booking         0
Property                 0
Date                     0
Rate Type                0
dtype: int64
```

In [27]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 14 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Avg Room Rate          50000 non-null  float64
 1   reservation_id         50000 non-null  object 
 2   check_in_date          50000 non-null  object 
 3   stay_duration          50000 non-null  int64  
 4   adults                 50000 non-null  int64  
 5   children               50000 non-null  int64  
 6   room_type              50000 non-null  object 
 7   special_requests_flag  50000 non-null  object 
 8   booking_channel        50000 non-null  object 
 9   reservation_status     50000 non-null  object 
 10  advanced_booking       50000 non-null  object 
 11  Property               50000 non-null  object 
 12  Date                   50000 non-null  object 
 13  Rate Type              50000 non-null  object 
dtypes: float64(1), int64(3), object(10)
memory usage: 5.3+ MB
```

Task

Calculate and summarize the basic descriptive statistics for the ‘Avg Room Rate’ column. Identify the maximum, minimum, mean, and median room rates.

In [28]:
```python
df["Avg Room Rate"].max()
```

Out[28]:
```
288.55
```

In [29]:
```python
df["Avg Room Rate"].min()
```

Out[29]:
```
71.1
```

In [30]:
```python
df["Avg Room Rate"].mean()
```

Out[30]:
```
147.1471442
```

In [31]:
```python
df["Avg Room Rate"].median()
```

Out[31]:
```
152.1
```

In [32]:
```python
df["Avg Room Rate"].describe()
```

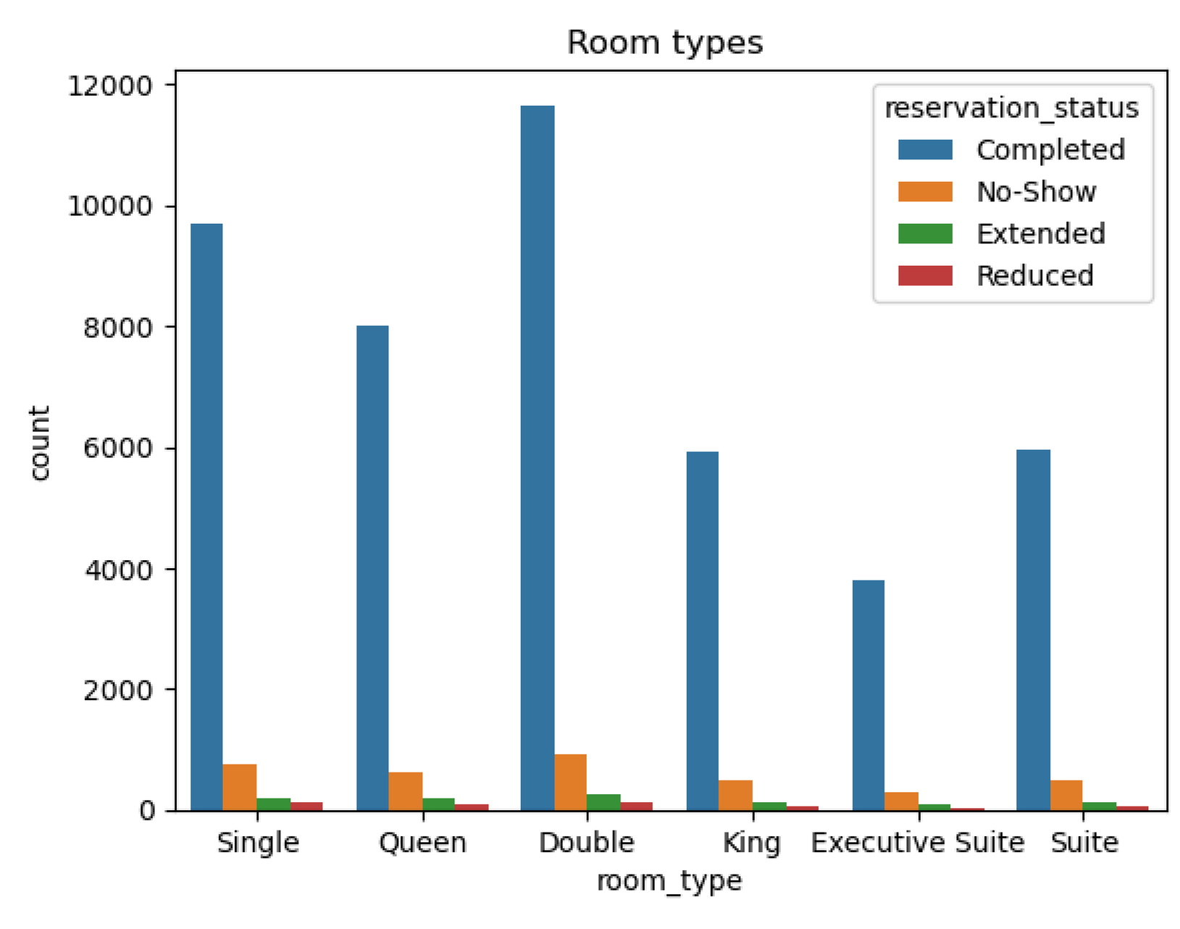
Out[32]:
```
count    50000.000000
mean       147.147144
std         48.316584
min         71.100000
25%        103.950000
50%        152.100000
75%        177.450000
max        288.550000
Name: Avg Room Rate, dtype: float64
```

In [33]:
```python
df.room_type.value_counts()
```

Out[33]:
```
Double            12936
Single            10764
Queen              8872
Suite              6623
King               6565
Executive Suite    4240
Name: room_type, dtype: int64
```

In [34]:
```python
sns.countplot(x="room_type",y=None,hue="reservation_status",data=df)
plt.title("Room types ")
```

Out[34]:
```
Text(0.5, 1.0, 'Room types ')
```



Room Type Analysis: Analyze the distribution of room types (‘Room Type’ column). Determine the most and least popular room types based on the number of reservations.

In [35]:
```python
df["booking_channel"].value_counts()
```

Out[35]:
```
Phone App       15391
Travel Agent    11574
Website         11532
Call Center      7700
Walk-in          3803
Name: booking_channel, dtype: int64
```

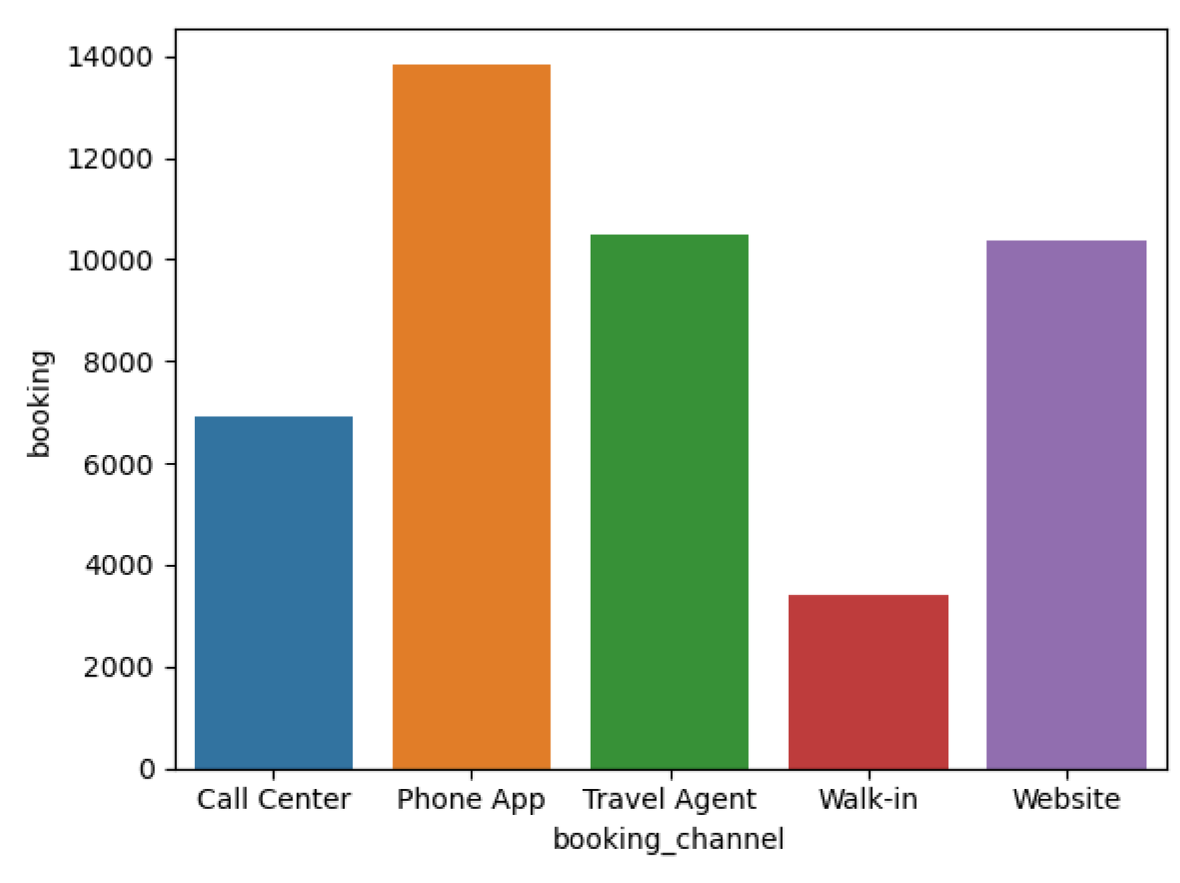
In [36]:
```python
b=df[df["reservation_status"]=="Completed"].groupby("booking_channel").size().reset_index().rename(columns={0:"booking"})
```

In [37]:
```python
print(b)
```

```
  booking_channel  booking
0     Call Center     6920
1       Phone App    13848
2    Travel Agent    10488
3         Walk-in     3400
4         Website    10379
```

In [38]:
```python
sns.barplot(x="booking_channel",y="booking",data=b)
```

Out[38]:
```
<Axes: xlabel='booking_channel', ylabel='booking'>
```



Booking Channel Insights: Investigate the distribution of booking channels (‘Booking Channel’ column). Identify the most effective booking channel based on the number of completed reservations.

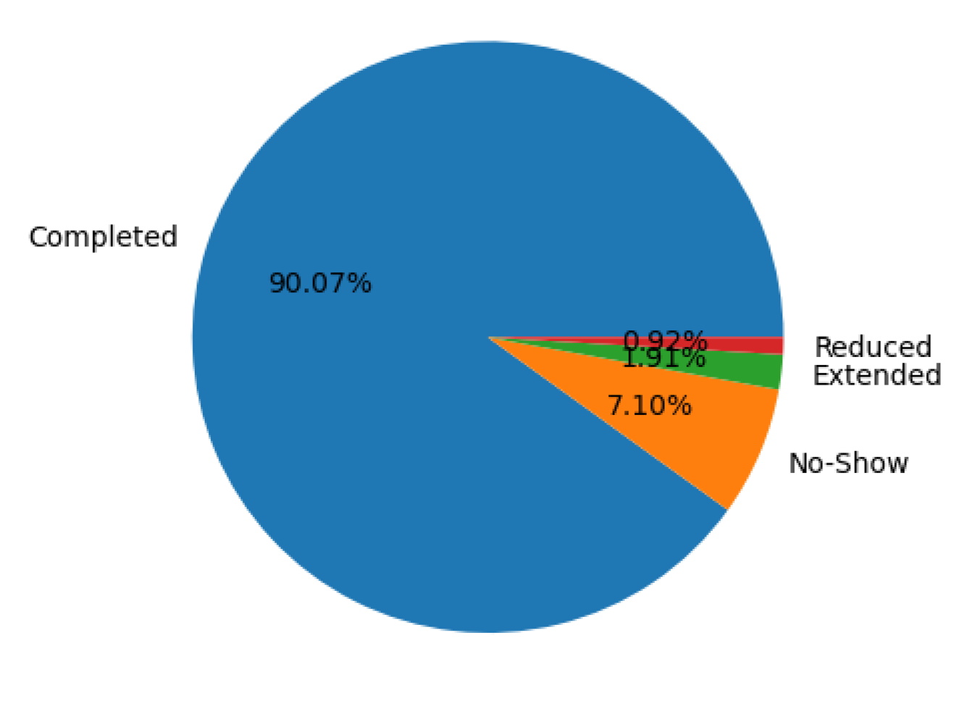
In [48]:
```python
reservation=df.reservation_status.value_counts()
names=df["reservation_status"].value_counts().index
```

In [49]:
```python
reservation
```

Out[49]:
```
Completed    45035
No-Show       3550
Extended       953
Reduced        462
Name: reservation_status, dtype: int64
```

In [50]:
```python
plt.pie(reservation,labels=names,autopct="%1.2f%%")
```

Out[50]:
```
([<matplotlib.patches.Wedge at 0x2643e0ec2d0>,
  <matplotlib.patches.Wedge at 0x2643e0e0f90>,
  <matplotlib.patches.Wedge at 0x2643e0eefd0>,
  <matplotlib.patches.Wedge at 0x2643e0f8890>],
 [Text(-1.0469071293725583, 0.33761733141074063, 'Completed'),
  Text(1.0127951594386826, -0.42923882049224504, 'No-Show'),
  Text(1.0923590271578676, -0.12942857407356972, 'Extended'),
  Text(1.0995365740026752, -0.03192682932048767, 'Reduced')],
 [Text(-0.5710402523850318, 0.18415490804222215, '90.07%'),
  Text(0.5524337233301905, -0.2341302657230427, '7.10%'),
  Text(0.5958321966315641, -0.07059740404012893, '1.91%'),
  Text(0.5997472221832774, -0.017414634174811452, '0.92%')])
```



Reservation Status Analysis: Explore the distribution of reservation statuses (‘Reservation Status’ column). Calculate the percentage of completed, no-show, and other reservaΘon statuses.

In [58]:
```python
df[df["Rate Type"]=="Weekday"].groupby("Rate Type")["Avg Room Rate"].mean()
```

Out[58]:
```
Rate Type
Weekday    142.031001
Name: Avg Room Rate, dtype: float64
```

In [57]:
```python
df[df["Rate Type"]=="Weekend"].groupby("Rate Type")["Avg Room Rate"].mean()
```

Out[57]:
```
Rate Type
Weekend    159.810204
Name: Avg Room Rate, dtype: float64
```

observation= There are 17.779203 stays difference between weekend and weedays

In [67]:
```python
df.Property.value_counts()
```

Out[67]:
```
The Sankey       27400
The Marimekko    13666
The Chord         8934
Name: Property, dtype: int64
```

In [71]:
```python
df["Avg Room Rate"].mean()
```

Out[71]:
```
147.1471442
```

Analyze the performance of different properties (‘Property’ column).

 Identify the property with the highest average room rate and the one with the most reservations.

In [72]:
```python
df[df["Property"]=="The Chord"].groupby("Property")["Avg Room Rate"].mean()
```

Out[72]:
```
Property
The Chord    88.133999
Name: Avg Room Rate, dtype: float64
```

In [73]:
```python
df[df["Property"]=="The Marimekko"].groupby("Property")["Avg Room Rate"].mean()
```

Out[73]:
```
Property
The Marimekko    110.153924
Name: Avg Room Rate, dtype: float64
```

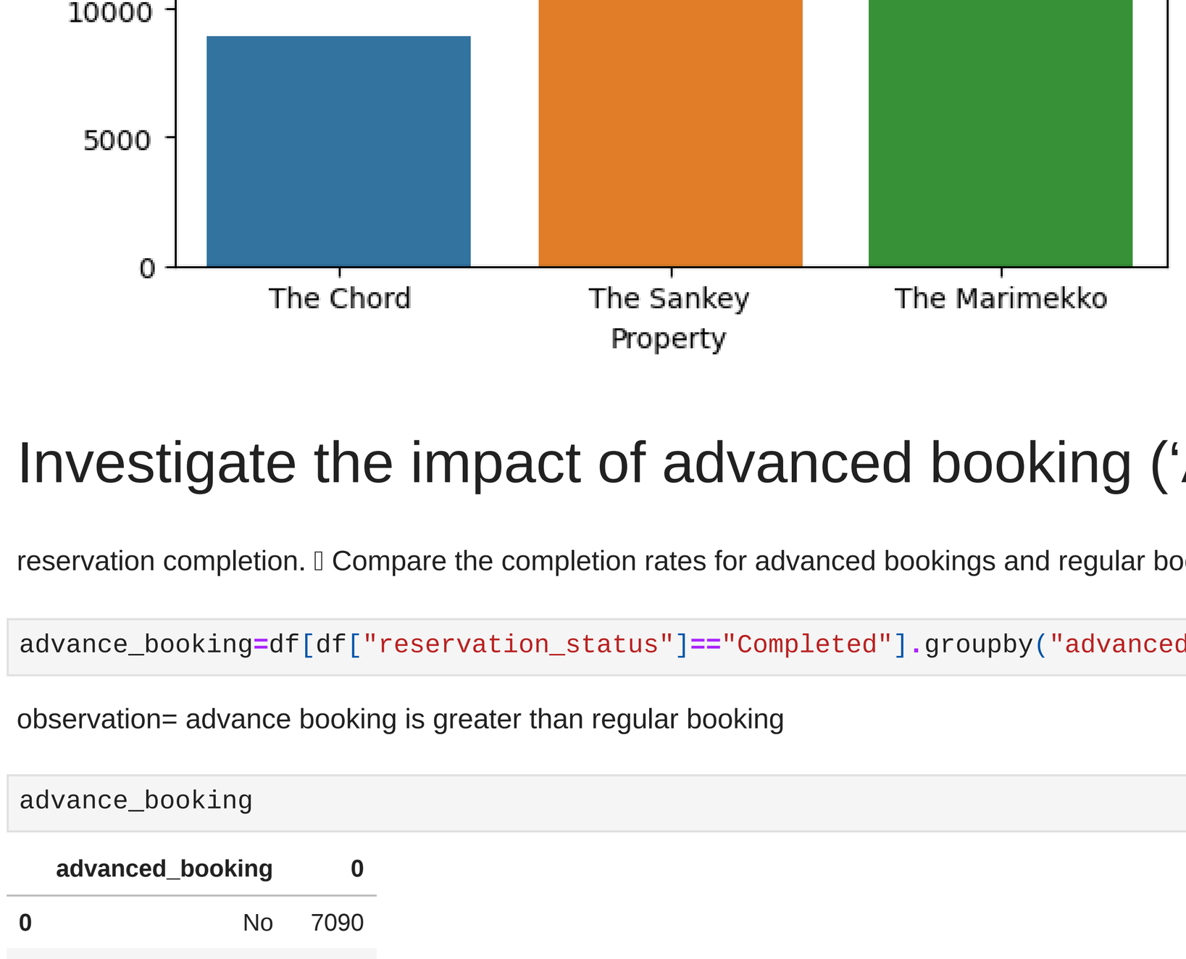
In [74]:
```python
df[df["Property"]=="The Sankey"].groupby("Property")["Avg Room Rate"].mean()
```

Out[74]:
```
Property
The Sankey    184.839581
Name: Avg Room Rate, dtype: float64
```

observation= The SAnkey has the highest average room rate

In [77]:
```python
sns.countplot(x="Property",y=None,data=df)
```

Out[77]:
```
<Axes: xlabel='Property', ylabel='count'>
```



Investigate the impact of advanced booking (‘Advanced Booking’ column) on

reservation completion. Compare the completion rates for advanced bookings and regular bookings.

In [93]:
```python
advance_booking=df[df["reservation_status"]=="Completed"].groupby("advanced_booking").size().reset_index()
```

observation= advance booking is greater than regular booking

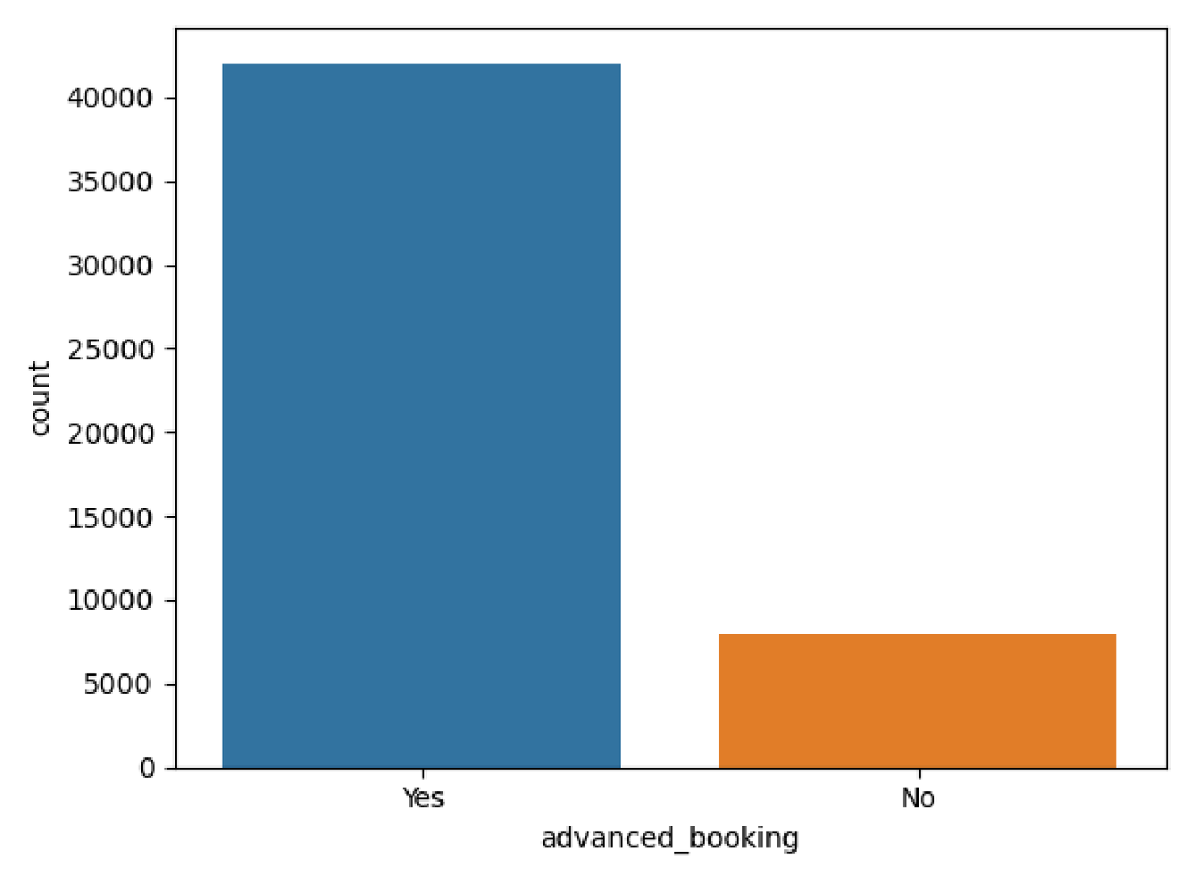
In [94]:
```python
advance_booking
```

Out[94]:

	advanced_booking	0
0	No	7090
1	Yes	37945

In [96]:
```python
sns.countplot(x="advanced_booking",y=None,data=df)
```

Out[96]:
```
<Axes: xlabel='advanced_booking', ylabel='count'>
```



In [ ]: