HOSPITALITY DATA ANALYTICS

DATA EXPLORATION

In [73]:
```
import pandas as pd
df_booking = pd.read_csv("C:\\source-code\\3_project_hospitality_analysis\\datasets\\fact_bookings.csv")
df_booking.head(4)
```

Out[73]:

booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_generated	rev
27-04-22	1/5/2022	2/5/2022	-3.0	RT1	direct online	1.0	Checked Out	10010	
30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	9100	
28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	5.0	Checked Out	9100000	
28-04-22	1/5/2022	2/5/2022	-2.0	RT1	others	NaN	Cancelled	9100	

In [74]:
```
df_booking.shape
```

Out[74]:
```
(134590, 12)
```

In [75]:
```
df_booking.room_category.unique()
```

Out[75]:
```
array(['RT1', 'RT2', 'RT3', 'RT4'], dtype=object)
```

In [76]:
```
df_booking.booking_platform.unique()
```

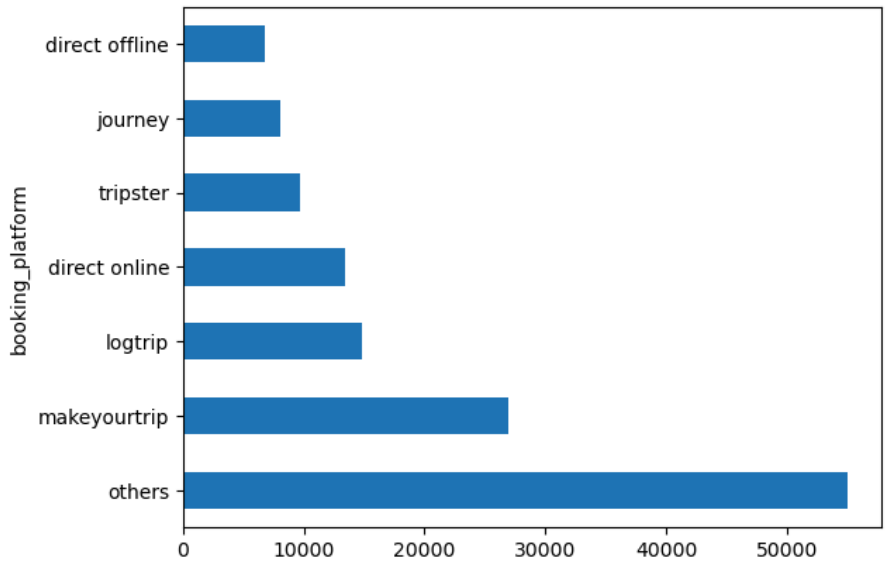
Out[76]:
```
array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',
       'journey', 'direct offline'], dtype=object)
```

In [77]:
```
df_booking.booking_platform.value_counts()
```

Out[77]:
```
booking_platform
others            55066
makeyourtrip      26898
logtrip           14756
direct online     13379
tripster           9630
journey            8106
direct offline     6755
Name: count, dtype: int64
```

In [78]:
```
df_booking.booking_platform.value_counts().plot(kind='barh')
```

Out[78]:
```
<Axes: ylabel='booking_platform'>
```



In [79]:
```
df_booking.describe()
```

Out[79]:

	property_id	no_guests	ratings_given	revenue_generated	revenue_realized
count	134590.000000	134587.000000	56683.000000	1.345900e+05	134590.000000
mean	18061.113493	2.036170	3.619004	1.537805e+04	12696.123256
std	1093.055847	1.034885	1.235009	9.303604e+04	6928.108124
min	16558.000000	-17.000000	1.000000	6.500000e+03	2600.000000
25%	17558.000000	1.000000	3.000000	9.900000e+03	7600.000000
50%	17564.000000	2.000000	4.000000	1.350000e+04	11700.000000
75%	18563.000000	2.000000	5.000000	1.800000e+04	15300.000000
max	19563.000000	6.000000	5.000000	2.856000e+07	45220.000000

In [80]:
```
df_booking['revenue_generated'].min(),df_booking['revenue_generated'].max()
```

Out[80]:
```
(6500, 28560000)
```

In [23]:
```
df_date = pd.read_csv("C:\\source-code\\3_project_hospitality_analysis\\datasets\\dim_date.csv")
df_hotels = pd.read_csv("C:\\source-code\\3_project_hospitality_analysis\\datasets\\dim_hotels.csv")
df_rooms = pd.read_csv("C:\\source-code\\3_project_hospitality_analysis\\datasets\\dim_rooms.csv")
df_agg_bookings = pd.read_csv("C:\\source-code\\3_project_hospitality_analysis\\datasets\\fact_aggregated_bookings.cs
```

In [24]:
```
df_hotels.shape
```

Out[24]:
```
(25, 4)
```

In [25]:
```
df_hotels.head(4)
```

Out[25]:

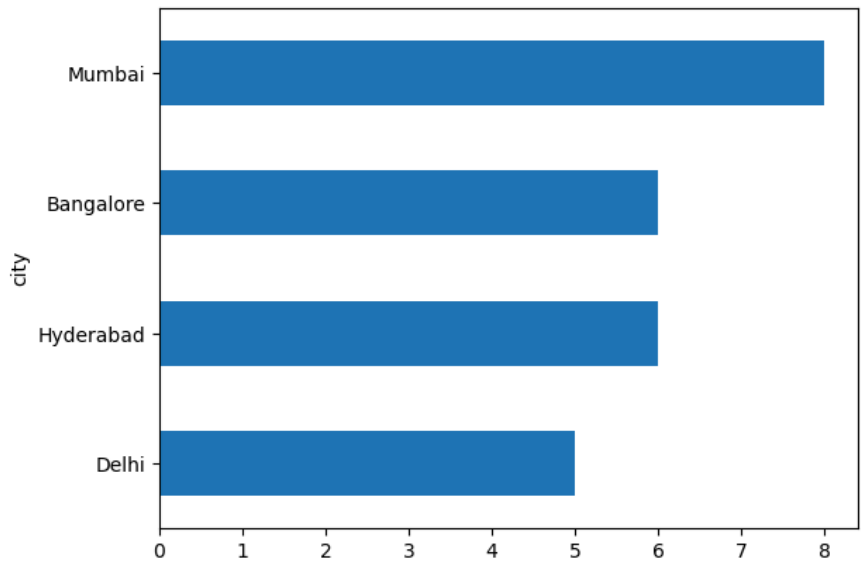
	property_id	property_name	category	city
0	16558	Atliq Grands	Luxury	Delhi
1	16559	Atliq Exotica	Luxury	Mumbai
2	16560	Atliq City	Business	Delhi
3	16561	Atliq Blu	Luxury	Delhi

In [26]:
```
df_hotels['category'].value_counts()
```

Out[26]:
```
category
Luxury      16
Business     9
Name: count, dtype: int64
```

In [34]:
```
df_hotels['city'].value_counts().sort_values().plot(kind='barh')
```

Out[34]:
```
<Axes: ylabel='city'>
```



In [82]:
```
df_agg_bookings.head(4)
```

Out[82]:

	property_id	check_in_date	room_category	successful_bookings	capacity
0	16559	1-May-22	RT1	25	30.0
1	19562	1-May-22	RT1	28	30.0
2	19563	1-May-22	RT1	23	30.0
3	17558	1-May-22	RT1	30	19.0

In [36]:
```
df_agg_bookings['property_id'].unique()
```

Out[36]:
```
array([16559, 19562, 19563, 17558, 16558, 17560, 19558, 19560, 17561,
       16560, 16561, 16562, 16563, 17559, 17562, 17563, 18558, 18559,
       18561, 18562, 18563, 19559, 19561, 17564, 18560], dtype=int64)
```

In [50]:
```
g = df_agg_bookings.groupby('property_id')["successful_bookings"].sum()
g
```

Out[50]:
```
property_id
16558    3153
16559    7338
16560    4693
16561    4418
16562    4820
16563    7211
17558    5053
17559    6142
17560    6013
17561    5183
17562    3424
17563    6337
17564    3982
18558    4475
18559    5256
18560    6638
18561    6458
18562    7333
18563    4737
19558    4400
19559    4729
19560    6079
19561    5736
19562    5812
19563    5413
Name: successful_bookings, dtype: int64
```

In [52]:
```
df_agg_bookings[df_agg_bookings['successful_bookings']>df_agg_bookings['capacity']]
```

Out[52]:

	property_id	check_in_date	room_category	successful_bookings	capacity
3	17558	1-May-22	RT1	30	19.0
12	16563	1-May-22	RT1	100	41.0
4136	19558	11-Jun-22	RT2	50	39.0
6209	19560	2-Jul-22	RT1	123	26.0
8522	19559	25-Jul-22	RT1	35	24.0
9194	18563	31-Jul-22	RT4	20	18.0

In [53]:
```
df_agg_bookings[df_agg_bookings['capacity']==df_agg_bookings['capacity'].max()]
```

Out[53]:

	property_id	check_in_date	room_category	successful_bookings	capacity
27	17558	1-May-22	RT2	38	50.0
128	17558	2-May-22	RT2	27	50.0
229	17558	3-May-22	RT2	26	50.0
328	17558	4-May-22	RT2	27	50.0
428	17558	5-May-22	RT2	29	50.0
...	...	...	...	...	...
8728	17558	27-Jul-22	RT2	22	50.0
8828	17558	28-Jul-22	RT2	21	50.0
8928	17558	29-Jul-22	RT2	23	50.0
9028	17558	30-Jul-22	RT2	32	50.0
9128	17558	31-Jul-22	RT2	30	50.0

92 rows × 5 columns

DATA CLEANING

In [83]:
```
df_booking.describe()
```

Out[83]:

	property_id	no_guests	ratings_given	revenue_generated	revenue_realized
count	134590.000000	134587.000000	56683.000000	1.345900e+05	134590.000000
mean	18061.113493	2.036170	3.619004	1.537805e+04	12696.123256
std	1093.055847	1.034885	1.235009	9.303604e+04	6928.108124
min	16558.000000	-17.000000	1.000000	6.500000e+03	2600.000000
25%	17558.000000	1.000000	3.000000	9.900000e+03	7600.000000
50%	17564.000000	2.000000	4.000000	1.350000e+04	11700.000000
75%	18563.000000	2.000000	5.000000	1.800000e+04	15300.000000
max	19563.000000	6.000000	5.000000	2.856000e+07	45220.000000

In [85]:
```
df_booking[df_booking['no_guests']<0]
```

Out[85]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_give
0	May012216558RT11	16558	27-04-22	1/5/2022	2/5/2022	-3.0	RT1	direct online	1.
3	May012216558RT14	16558	28-04-22	1/5/2022	2/5/2022	-2.0	RT1	others	Na
17924	May122218559RT44	18559	12/5/2022	12/5/2022	14-05-22	-10.0	RT4	direct online	Na
18020	May122218561RT22	18561	8/5/2022	12/5/2022	14-05-22	-12.0	RT2	makeyourtrip	Na
18119	May122218562RT311	18562	5/5/2022	12/5/2022	17-05-22	-6.0	RT3	direct offline	5.
18121	May122218562RT313	18562	10/5/2022	12/5/2022	17-05-22	-4.0	RT3	direct online	Na
56715	Jun082218562RT12	18562	5/6/2022	8/6/2022	13-06-22	-17.0	RT1	others	Na
119765	Jul202219560RT220	19560	19-07-22	20-07-22	22-07-22	-1.0	RT2	others	Na
134586	Jul312217564RT47	17564	30-07-22	31-07-22	1/8/2022	-4.0	RT4	logtrip	2.

In [86]:
```
df_booking.shape
```

Out[86]: (134590, 12)

In [89]:
```
df_booking = df_booking[df_booking['no_guests']>0]                    #removing records that have negative number o
df_booking.shape
```

Out[89]: (134578, 12)

In [90]:
```
df_booking['revenue_generated'].min(),df_booking['revenue_generated'].max()
```

Out[90]: (6500, 28560000)

In [91]:
```
avg, std = df_booking['revenue_generated'].mean(), df_booking['revenue_generated'].std()
avg , std                                                                    #here std is one standar
```

Out[91]: (15378.036937686695, 93040.15493143328)

In [92]:
```
higher_limit , lower_limit = avg + 3*std , avg - 3*std
higher_limit , lower_limit
```

Out[92]:
```
(294498.50173198653, -263742.4278566132)
```

In [66]:
```
df_booking[df_booking['revenue_generated']<0]                    #checking if revenue_ggenerated column has its v
```

Out[66]:

booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_stat

In [93]:
```
df_booking[df_booking['revenue_generated']>higher_limit]                    #finding outliers in revenue_gener
```

Out[93]:

booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_generated	rev
28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	5.0	Checked Out	9100000	
29-04-22	1/5/2022	2/5/2022	6.0	RT3	direct online	NaN	Checked Out	28560000	
28-04-22	1/5/2022	4/5/2022	2.0	RT2	direct offline	3.0	Checked Out	12600000	
26-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	2000000	
21-07-22	28-07-22	29-07-22	2.0	RT2	direct online	3.0	Checked Out	10000000	

In [94]:
```
df_booking = df_booking[df_booking['revenue_generated']<higher_limit]
df_booking.shape                                                    #removing outliers in revenue_gener
```

Out[94]:
```
(134573, 12)
```

In [95]:
```
df_booking['revenue_realized'].describe()
```

Out[95]:
```
count    134573.000000
mean      12695.983585
std        6927.791692
min        2600.000000
25%        7600.000000
50%       11700.000000
75%       15300.000000
max       45220.000000
Name: revenue_realized, dtype: float64
```

In [96]:
```
avg1, std1 = df_booking['revenue_realized'].mean(), df_booking['revenue_realized'].std()
avg1 , std1
```

Out[96]:
```
(12695.983585117372, 6927.791692242509)
```

In [97]:
```
higher_limit1 = avg1 + 3*std1
higher_limit1                                    #higher limit1 > maximum value of revenue_realized => outliers could
```

Out[97]:
```
33479.3586618449
```

In [98]:
```
df_booking[df_booking['revenue_realized']>higher_limit1]
```

Out[98]:

perty_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_generat
16559	27-04-22	1/5/2022	7/5/2022	4.0	RT4	others	NaN	Checked Out	387
16559	1/5/2022	1/5/2022	2/5/2022	6.0	RT4	tripster	3.0	Checked Out	452
16559	28-04-22	1/5/2022	3/5/2022	3.0	RT4	others	5.0	Checked Out	355
16559	24-04-22	1/5/2022	7/5/2022	5.0	RT4	logtrip	NaN	Checked Out	419
16560	30-04-22	1/5/2022	3/5/2022	5.0	RT4	others	3.0	Checked Out	345
...	...	...	...	...	...	...	...	...	
19560	31-07-22	31-07-22	2/8/2022	6.0	RT4	direct online	5.0	Checked Out	399
19560	31-07-22	31-07-22	1/8/2022	6.0	RT4	others	2.0	Checked Out	399
19562	28-07-22	31-07-22	1/8/2022	6.0	RT4	makeyourtrip	4.0	Checked Out	399
19562	25-07-22	31-07-22	6/8/2022	5.0	RT4	direct offline	5.0	Checked Out	370
17564	31-07-22	31-07-22	1/8/2022	4.0	RT4	makeyourtrip	4.0	Checked Out	387

In [100]:
```
#1299 rows have revenue_generated > higher_limit1 but almost all rows room_category is RT4 which is a presidential su
#(has higher rent than other suites)
```

In [101]:
```
df_booking[df_booking['room_category']=='RT4'].revenue_realized.describe()
```

Out[101]:
```
count    16071.000000
mean     23439.308444
std       9048.599076
min       7600.000000
25%      19000.000000
50%      26600.000000
75%      32300.000000
max      45220.000000
Name: revenue_realized, dtype: float64
```

In [102]:
```
high = 23439.308444 + 3*9048.599076
high                  #higher limit for RT4 room_category rows is ~ 50585. Thus, there are no outliers in revenue_rea
```

Out[102]:
```
50585.105672000005
```

In [103]:
```
df_booking.isnull().sum()             #ratings_given has 77897 NA values but it is alright as only very few people g
```

Out[103]:
```
booking_id               0
property_id              0
booking_date             0
check_in_date            0
checkout_date            0
no_guests                0
room_category            0
booking_platform         0
ratings_given        77897
booking_status           0
revenue_generated        0
revenue_realized         0
dtype: int64
```

In [107]:
```
df_agg_bookings.isnull().sum()
```

Out[107]:
```
property_id            0
check_in_date          0
room_category          0
successful_bookings    0
capacity               2
dtype: int64
```

In [110]:
```
df_agg_bookings = df_agg_bookings.fillna(df_agg_bookings['capacity'].mean())
df_agg_bookings
```

Out[110]:

	property_id	check_in_date	room_category	successful_bookings	capacity
0	16559	1-May-22	RT1	25	30.0
1	19562	1-May-22	RT1	28	30.0
2	19563	1-May-22	RT1	23	30.0
3	17558	1-May-22	RT1	30	19.0
4	16558	1-May-22	RT1	18	19.0
...	...	...	...	...	...
9195	16563	31-Jul-22	RT4	13	18.0
9196	16559	31-Jul-22	RT4	13	18.0
9197	17558	31-Jul-22	RT4	3	6.0
9198	19563	31-Jul-22	RT4	3	6.0
9199	17561	31-Jul-22	RT4	3	4.0

In [111]:
```
df_agg_bookings.isnull().sum()
```

Out[111]:
```
property_id            0
check_in_date          0
room_category          0
successful_bookings    0
capacity               0
dtype: int64
```

In [112]:
```
df_agg_bookings[df_agg_bookings['successful_bookings']>df_agg_bookings['capacity']]
```

Out[112]:

	property_id	check_in_date	room_category	successful_bookings	capacity
3	17558	1-May-22	RT1	30	19.0
12	16563	1-May-22	RT1	100	41.0
4136	19558	11-Jun-22	RT2	50	39.0
6209	19560	2-Jul-22	RT1	123	26.0
8522	19559	25-Jul-22	RT1	35	24.0
9194	18563	31-Jul-22	RT4	20	18.0

DATA TRANSFORMATION

In [113]:
```
df_agg_bookings.head(4)
```

Out[113]:

	property_id	check_in_date	room_category	successful_bookings	capacity
0	16559	1-May-22	RT1	25	30.0
1	19562	1-May-22	RT1	28	30.0
2	19563	1-May-22	RT1	23	30.0
3	17558	1-May-22	RT1	30	19.0

In [117]:
```
df_agg_bookings['occ_pct'] = df_agg_bookings['successful_bookings']/df_agg_bookings['capacity']
df_agg_bookings.head(4)
```

Out[117]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct
0	16559	1-May-22	RT1	25	30.0	0.833333
1	19562	1-May-22	RT1	28	30.0	0.933333
2	19563	1-May-22	RT1	23	30.0	0.766667
3	17558	1-May-22	RT1	30	19.0	1.578947

In [118]:
```
df_agg_bookings['occ_pct'] = df_agg_bookings['occ_pct'].apply(lambda x: round(x*100, 2))
df_agg_bookings.head(3)                                                                    #transforming occ
```

Out[118]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct
0	16559	1-May-22	RT1	25	30.0	83.33
1	19562	1-May-22	RT1	28	30.0	93.33
2	19563	1-May-22	RT1	23	30.0	76.67

INSIGHTS GENERATION

1. what is the average occupancy rate in each of the room categories?

In [120]:
```
df_agg_bookings.groupby('room_category')['occ_pct'].mean().round(2)
```

Out[120]:
```
room_category
RT1    58.23
RT2    58.04
RT3    58.03
RT4    59.30
Name: occ_pct, dtype: float64
```

In [121]:
```
df_rooms
```

Out[121]:

	room_id	room_class
0	RT1	Standard
1	RT2	Elite
2	RT3	Premium
3	RT4	Presidential

In [126]:
```
df = pd.merge(df_agg_bookings,df_rooms,left_on='room_category',right_on ='room_id')
df
```

Out[126]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_id	room_class
0	16559	1-May-22	RT1	25	30.0	83.33	RT1	Standard
1	19562	1-May-22	RT1	28	30.0	93.33	RT1	Standard
2	19563	1-May-22	RT1	23	30.0	76.67	RT1	Standard
3	17558	1-May-22	RT1	30	19.0	157.89	RT1	Standard
4	16558	1-May-22	RT1	18	19.0	94.74	RT1	Standard
...	...	...	...	...	...	...	...	...
9195	16563	31-Jul-22	RT4	13	18.0	72.22	RT4	Presidential
9196	16559	31-Jul-22	RT4	13	18.0	72.22	RT4	Presidential
9197	17558	31-Jul-22	RT4	3	6.0	50.00	RT4	Presidential
9198	19563	31-Jul-22	RT4	3	6.0	50.00	RT4	Presidential
9199	17561	31-Jul-22	RT4	3	4.0	75.00	RT4	Presidential

9200 rows × 8 columns

In [127]:
```
df.groupby('room_class')['occ_pct'].mean().round(2)
```

Out[127]:
```
room_class
Elite           58.04
Premium         58.03
Presidential    59.30
Standard        58.23
Name: occ_pct, dtype: float64
```

In [128]:
```
df.drop('room_id',axis=1,inplace=True)
df.head(3)
```

Out[128]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class
0	16559	1-May-22	RT1	25	30.0	83.33	Standard
1	19562	1-May-22	RT1	28	30.0	93.33	Standard
2	19563	1-May-22	RT1	23	30.0	76.67	Standard

2. Printing average occupancy rate per city

In [129]:
```
df_hotels.head(2)
```

Out[129]:

	property_id	property_name	category	city
0	16558	Atliq Grands	Luxury	Delhi
1	16559	Atliq Exotica	Luxury	Mumbai

In [130]:
```
df.head(2)
```

Out[130]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class
0	16559	1-May-22	RT1	25	30.0	83.33	Standard
1	19562	1-May-22	RT1	28	30.0	93.33	Standard

In [131]:
```
df1 = pd.merge(df_hotels,df, on= 'property_id')
df1.head(2)
```

Out[131]:

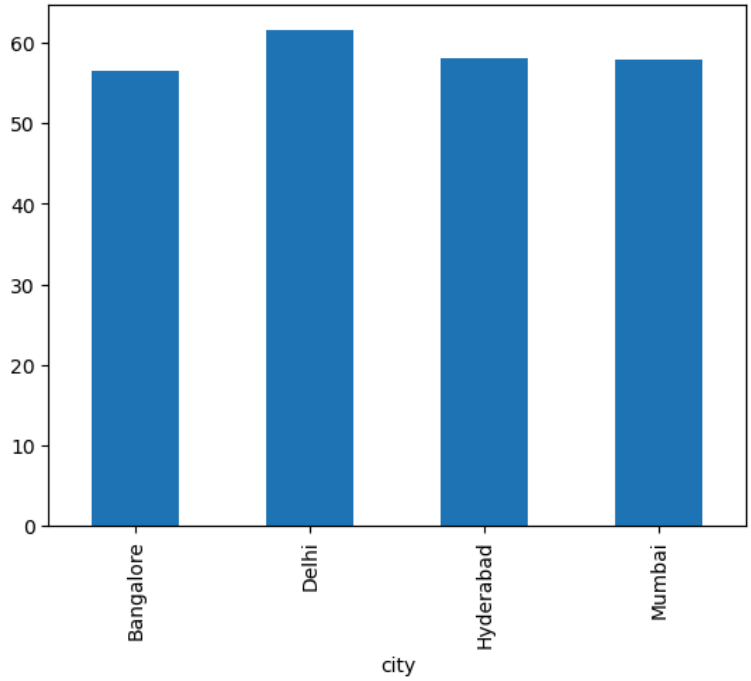
	property_id	property_name	category	city	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class
0	16558	Atliq Grands	Luxury	Delhi	1-May-22	RT1	18	19.0	94.74	Standard
1	16558	Atliq Grands	Luxury	Delhi	2-May-22	RT1	12	19.0	63.16	Standard

In [133]:
```
df1.groupby('city').occ_pct.mean().round(2)
```

Out[133]:
```
city
Bangalore    56.59
Delhi        61.61
Hyderabad    58.14
Mumbai       57.94
Name: occ_pct, dtype: float64
```

In [134]:
```
df1.groupby('city').occ_pct.mean().round(2).plot(kind='bar')
```

Out[134]: <Axes: xlabel='city'>



3. When was the occupancy rate better? weekday or weekend?

In [138]:
```
df_date.head(2)
```

Out[138]:

	date	mmm yy	week no	day_type
0	01-May-22	May 22	W 19	weekend
1	02-May-22	May 22	W 19	weekeday

In [139]:
```
df1.head(2)
```

Out[139]:

	property_id	property_name	category	city	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class
0	16558	Atliq Grands	Luxury	Delhi	1-May-22	RT1	18	19.0	94.74	Standard
1	16558	Atliq Grands	Luxury	Delhi	2-May-22	RT1	12	19.0	63.16	Standard

In [141]:
```
df2 = pd.merge(df1,df_date,left_on='check_in_date',right_on ='date')
df2.head(2)
```

Out[141]:

property_id	property_name	category	city	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	date	mmm yy	we
16558	Atliq Grands	Luxury	Delhi	10-May-22	RT1	10	19.0	52.63	Standard	10-May-22	May 22	W
16558	Atliq Grands	Luxury	Delhi	10-May-22	RT2	12	22.0	54.55	Elite	10-May-22	May 22	W

In [143]:
```
df2.groupby('day_type').occ_pct.mean().round(2)
```

Out[143]:
```
day_type
weekeday    50.90
weekend     72.39
Name: occ_pct, dtype: float64
```

4.In the month of June, what was the occupancy for different cities?

In [156]:
```
df2['mmm yy'].unique()
```

Out[156]:
```
array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

In [158]:
```
df_june22 = df2[df2['mmm yy'] == 'Jun 22'].groupby('city').occ_pct.mean().round(2).sort_values(ascending=False)
df_june22
```

Out[158]:
```
city
Delhi        62.47
Hyderabad    58.46
Mumbai       58.38
Bangalore    56.58
Name: occ_pct, dtype: float64
```

4. adding august month data to the existigng dataframe

In [160]:
```
df_august = pd.read_csv("C:\\source-code\\3_project_hospitality_analysis\\datasets\\new_data_august.csv")
df_august.head(2)
```

Out[160]:

	property_id	property_name	category	city	room_category	room_class	check_in_date	mmm yy	week no	day_type	successful_bookings	ca
0	16559	Atliq Exotica	Luxury	Mumbai	RT1	Standard	01-Aug-22	Aug-22	W 32	weekeday	30	
1	19562	Atliq Bay	Luxury	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	weekeday	21	

In [161]:
```
df_august.columns
```

Out[161]:
```
Index(['property_id', 'property_name', 'category', 'city', 'room_category',
       'room_class', 'check_in_date', 'mmm yy', 'week no', 'day_type',
       'successful_bookings', 'capacity', 'occ%'],
      dtype='object')
```

In [162]:
```
df2.columns
```

Out[162]:
```
Index(['property_id', 'property_name', 'category', 'city', 'check_in_date',
       'room_category', 'successful_bookings', 'capacity', 'occ_pct',
       'room_class', 'date', 'mmm yy', 'week no', 'day_type'],
      dtype='object')
```

In [166]:
```
df_august.shape, df2.shape
```

Out[166]:
```
((7, 13), (6500, 14))
```

In [171]:
```
latest_df = pd.concat([df2,df_august],ignore_index=True,axis=0)
latest_df.tail(4)
```

Out[171]:

	property_id	property_name	category	city	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	date
6503	19558	Atliq Grands	Luxury	Bangalore	01-Aug-22	RT1	30	40.0	NaN	Standard	NaN
6504	19560	Atliq City	Business	Bangalore	01-Aug-22	RT1	20	26.0	NaN	Standard	NaN
6505	17561	Atliq Blu	Luxury	Mumbai	01-Aug-22	RT1	18	26.0	NaN	Standard	NaN
6506	17564	Atliq Seasons	Business	Mumbai	01-Aug-22	RT1	10	16.0	NaN	Standard	NaN

In [172]:
```
latest_df.shape
```

Out[172]:
```
(6507, 15)
```

5. print revenue realised per city

In [176]:
```
df_booking.head(2)
```

Out[176]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	boo
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	

In [177]:
```
df_hotels.head(2)
```

Out[177]:

	property_id	property_name	category	city
0	16558	Atliq Grands	Luxury	Delhi
1	16559	Atliq Exotica	Luxury	Mumbai

In [179]:
```
df_rev = pd.merge(df_booking,df_hotels,on='property_id')
df_rev.head(2)
```

Out[179]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	boo
0	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	
1	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	

In [181]:
```
df_rev.groupby('city')['revenue_realized'].sum()
```

Out[181]:
```
city
Bangalore    420383550
Delhi        294404488
Hyderabad    325179310
Mumbai       668569251
Name: revenue_realized, dtype: int64
```

6.print month by month revenue

In [187]:
```
df_date.head(3)
```

Out[187]:

	date	mmm yy	week no	day_type
0	01-May-22	May 22	W 19	weekend
1	02-May-22	May 22	W 19	weekeday
2	03-May-22	May 22	W 19	weekeday

In [184]:
```
latest_df['mmm yy'].unique()
```

Out[184]:
```
array(['May 22', 'Jun 22', 'Jul 22', 'Aug-22'], dtype=object)
```

In [186]:
```
df_rev.head(2)
```

Out[186]:

d	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_generated	re
8	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	9100	
8	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	10920	

In [188]:
```
pd.merge(df_rev,df_date,left_on='check_in_date',right_on='date')
```

Out[188]:

booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_stat

In [191]:
```
df_rev['check_in_date'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 134573 entries, 0 to 134572
Series name: check_in_date
Non-Null Count   Dtype
--------------   -----
134573 non-null  object
dtypes: object(1)
memory usage: 1.0+ MB
```

In [192]:
```
df_date['date'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 92 entries, 0 to 91
Series name: date
Non-Null Count  Dtype
--------------  -----
92 non-null     object
dtypes: object(1)
memory usage: 868.0+ bytes
```

In [193]:
```
df_date['date'] = pd.to_datetime(df_date['date'])
df_date.head(2)
```

```
C:\Users\Shreya\AppData\Local\Temp\ipykernel_8920\33639996.py:1: UserWarning: Could not infer format, so each elemen
t will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please s
pecify a format.
  df_date['date'] = pd.to_datetime(df_date['date'])
```

Out[193]:

	date	mmm yy	week no	day_type
0	2022-05-01	May 22	W 19	weekend
1	2022-05-02	May 22	W 19	weekeday

In [194]:
```
df_date['date'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 92 entries, 0 to 91
Series name: date
Non-Null Count  Dtype
--------------  -----
92 non-null     datetime64[ns]
dtypes: datetime64[ns](1)
memory usage: 868.0 bytes
```

In [213]:
```
df_rev['check_in_date'] = pd.to_datetime(df_rev["check_in_date"],format='mixed')
# The format='mixed' option is used in pd.to_datetime() to handle columns with mixed date formats.
df_rev.head(2)
```

Out[213]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	boo
0	May012216558RT12	16558	30-04-22	2022-01-05	2/5/2022	2.0	RT1	others	NaN	
1	May012216558RT15	16558	27-04-22	2022-01-05	2/5/2022	4.0	RT1	direct online	5.0	

In [214]:
```
df_rev['check_in_date'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 134573 entries, 0 to 134572
Series name: check_in_date
Non-Null Count  Dtype
--------------  -----
55790 non-null  datetime64[ns]
dtypes: datetime64[ns](1)
memory usage: 1.0 MB
```

In [215]:
```
df_month = pd.merge(df_rev,df_date,left_on='check_in_date',right_on='date')
df_month.head(3)
```

Out[215]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	boo
0	May052216558RT11	16558	15-04-22	2022-05-05	7/5/2022	3.0	RT1	tripster	5.0	
1	May052216558RT12	16558	30-04-22	2022-05-05	7/5/2022	2.0	RT1	others	NaN	
2	May052216558RT13	16558	1/5/2022	2022-05-05	6/5/2022	3.0	RT1	direct offline	5.0	

In [220]:
```
df_month.groupby('mmm yy')['revenue_realized'].sum().sort_values()
```

Out[220]:
```
mmm yy
Jun 22    52903014
Jul 22    60278496
May 22    60961428
Name: revenue_realized, dtype: int64
```

7.print revenue realize per hotel type

In [225]:
```
df_month.groupby('property_name')['revenue_realized'].sum()
```

Out[225]:
```
property_name
Atliq Bay        26936115
Atliq Blu        26459751
Atliq City       29047727
Atliq Exotica    32436799
Atliq Grands     21644446
Atliq Palace     30945855
Atliq Seasons     6672245
Name: revenue_realized, dtype: int64
```

8.Print average rate per city

In [226]:
```
df_month.groupby("city")["ratings_given"].mean().round(2)
```

Out[226]:
```
city
Bangalore    3.41
Delhi        3.79
Hyderabad    3.65
Mumbai       3.63
Name: ratings_given, dtype: float64
```

9. Print a pie chart of revenue realized per booking platform

In [230]:
```
df_month.groupby("booking_platform")["revenue_realized"].sum().plot(kind ='pie')
```

Out[230]:
```
<Axes: ylabel='revenue_realized'>
```