AtliQ Hotels Data Analysis Project

In [3]:
```
import pandas as pd
```

---

1. Data Import and Data Exploration

Datasets

We have 5 csv file

- dim_date.csv
- dim_hotels.csv
- dim_rooms.csv
- fact_aggregated_bookings
- fact_bookings.csv

Read bookings data in a dataframe

In [107…
```
df_bookings = pd.read_csv('datasets/fact_bookings.csv')
```

Explore bookings data

In [30]:
```
df_bookings.head()
```

Out[30]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given
0	May012216558RT11	16558	27-04-22	1/5/2022	2/5/2022	-3.0	RT1	direct online	1.0
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	5.0
3	May012216558RT14	16558	28-04-22	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 [32]:
```
df_bookings.shape
```

Out[32]:
```
(134590, 12)
```

In [34]:
```
df_bookings.room_category.unique()
```

Out[34]:
```
array(['RT1', 'RT2', 'RT3', 'RT4'], dtype=object)
```

In [36]:
```
df_bookings.booking_platform.unique()
```

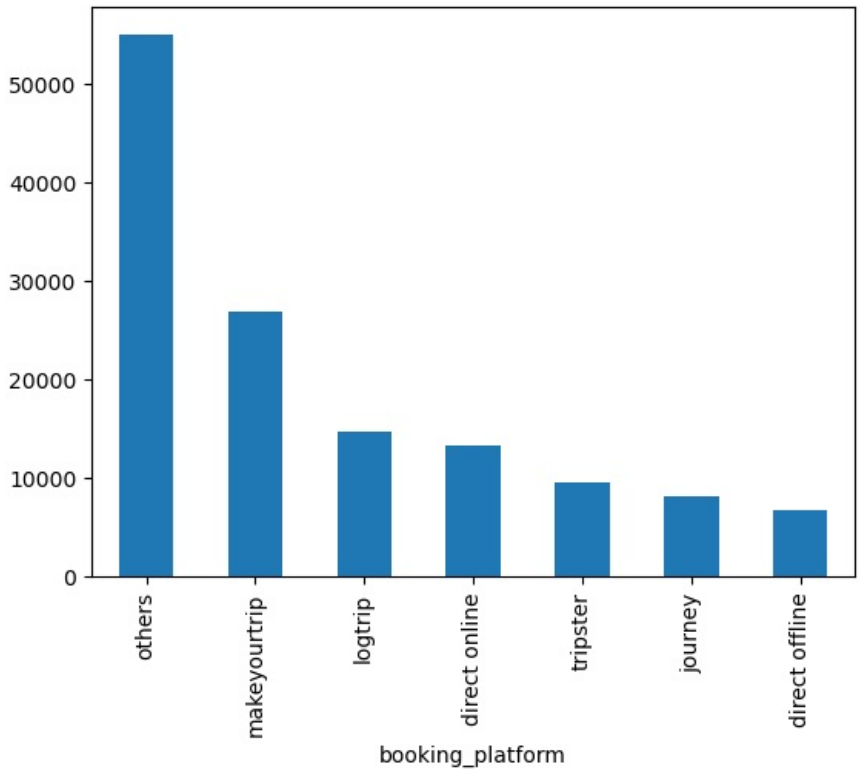
Out[36]:
```
array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',
       'journey', 'direct offline'], dtype=object)
```

In [38]:
```
df_bookings.booking_platform.value_counts()
```

Out[38]:
```
booking_platform
others            55066
makeyourtrip      26898
logtrip           14756
direct online     13379
tripster           9630
journey            8106
direct offline     6755
Name: count, dtype: int64
```

In [42]:
```
df_bookings.booking_platform.value_counts().plot(kind = 'bar')
```

Out[42]:
```
<Axes: xlabel='booking_platform'>
```



In [44]:
```
df_bookings.describe()
```

Out[44]:

	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

Read rest of the files

In [5]:
```
df_date = pd.read_csv('datasets/dim_date.csv')
df_hotels = pd.read_csv('datasets/dim_hotels.csv')
df_rooms = pd.read_csv('datasets/dim_rooms.csv')
df_agg_bookings = pd.read_csv('datasets/fact_aggregated_bookings.csv')
```

In [49]:
```
df_hotels.shape
```

Out[49]:
```
(25, 4)
```

In [53]:
```
df_hotels.head(3)
```

Out[53]:

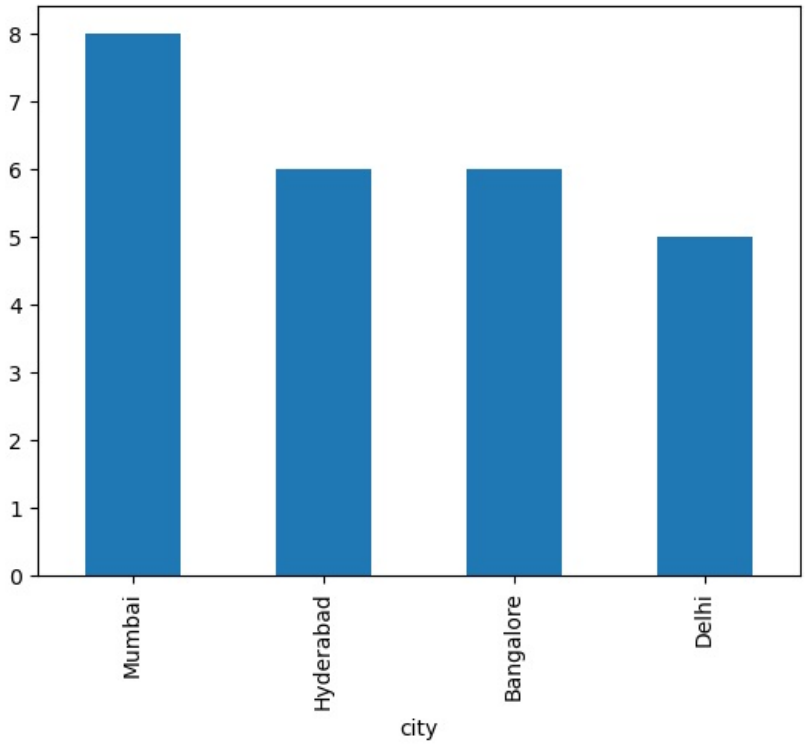
	property_id	property_name	category	city
0	16558	Atliq Grands	Luxury	Delhi
1	16559	Atliq Exotica	Luxury	Mumbai
2	16560	Atliq City	Business	Delhi

In [55]:
```
df_hotels.category.value_counts()
```

Out[55]:
```
category
Luxury      16
Business     9
Name: count, dtype: int64
```

In [57]:
```
df_hotels.city.value_counts().plot(kind = 'bar')
```

Out[57]:
```
<Axes: xlabel='city'>
```



In [59]:
```
df_agg_bookings.head(3)
```

Out[59]:

	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

In [61]:
```
df_agg_bookings.property_id.unique()
```

Out[61]:
```
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 [65]:
```
df_agg_bookings.groupby('property_id')[['successful_bookings']].sum()
```

Out[65]:

property_id	successful_bookings
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

In [69]:
```
df_agg_bookings[df_agg_bookings.successful_bookings > df_agg_bookings.capacity]
```

Out[69]:

	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 [119…
```
df_agg_bookings[df_agg_bookings.capacity == df_agg_bookings.capacity.max()]
```

Out[119]:

	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

2. Data Cleaning

In [76]:
```
df_bookings.describe()
```

Out[76]:

	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

(1) Clean data error in 'no_guests' column

In [111…
```
df_bookings[df_bookings.no_guests <= 0]
```

Out[111]:

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

As you can see above, number of guests having less than zero value represents data error. We can ignore these records.

In [113…
```
df_bookings = df_bookings[df_bookings.no_guests > 0]
```

In [115…
```
df_bookings.shape
```

Out[115]:
```
(134578, 12)
```

(2) Outlier removal in 'revenue_generated' column

In [117…
```
df_bookings.revenue_generated.min(), df_bookings.revenue_generated.max()
```

Out[117]:
```
(6500, 28560000)
```

In [119…
```
df_bookings.revenue_generated.mean(), df_bookings.revenue_generated.median()
```

Out[119]:
```
(15378.036937686695, 13500.0)
```

In [121…
```
avg, std = df_bookings.revenue_generated.mean(), df_bookings.revenue_generated.std()
```

In [123…
```
higher_limit = avg + 3*std
higher_limit
```

Out[123]:
```
294498.50173198653
```

In [125…
```
lower_limit = avg - 3*std
lower_limit
```

Out[125]:
```
-263742.4278566132
```

In [127…
```
df_bookings[df_bookings.revenue_generated <= 0]
```

Out[127]:

booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking

In [129…
```
df_bookings[df_bookings.revenue_generated > higher_limit]
```

Out[129]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	
111	May012216559RT32	16559	29-04-22	1/5/2022	2/5/2022	6.0	RT3	direct online	
315	May012216562RT22	16562	28-04-22	1/5/2022	4/5/2022	2.0	RT2	direct offline	
562	May012217559RT118	17559	26-04-22	1/5/2022	2/5/2022	2.0	RT1	others	
129176	Jul282216562RT26	16562	21-07-22	28-07-22	29-07-22	2.0	RT2	direct online	

In [131…
```
df_bookings = df_bookings[df_bookings.revenue_generated <= higher_limit]
df_bookings.shape
```

Out[131]:
```
(134573, 12)
```

(3) Checking for outliers in 'revenue_generated' column

In [134…
```
df_bookings.revenue_realized.describe()
```

Out[134]:
```
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 [136…
```
higher_limit = df_bookings.revenue_realized.mean() + 3*df_bookings.revenue_realized.std()
higher_limit
```

Out[136]:
```
33479.3586618449
```

In [138…
```
df_bookings[df_bookings.revenue_realized > higher_limit]
```

Out[138]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings
137	May012216559RT41	16559	27-04-22	1/5/2022	7/5/2022	4.0	RT4	others	
139	May012216559RT43	16559	1/5/2022	1/5/2022	2/5/2022	6.0	RT4	tripster	
143	May012216559RT47	16559	28-04-22	1/5/2022	3/5/2022	3.0	RT4	others	
149	May012216559RT413	16559	24-04-22	1/5/2022	7/5/2022	5.0	RT4	logtrip	
222	May012216560RT45	16560	30-04-22	1/5/2022	3/5/2022	5.0	RT4	others	
...	...	...	...	...	...	...	...	...	
134328	Jul312219560RT49	19560	31-07-22	31-07-22	2/8/2022	6.0	RT4	direct online	
134331	Jul312219560RT412	19560	31-07-22	31-07-22	1/8/2022	6.0	RT4	others	
134467	Jul312219562RT45	19562	28-07-22	31-07-22	1/8/2022	6.0	RT4	makeyourtrip	
134474	Jul312219562RT412	19562	25-07-22	31-07-22	6/8/2022	5.0	RT4	direct offline	
134581	Jul312217564RT42	17564	31-07-22	31-07-22	1/8/2022	4.0	RT4	makeyourtrip	

1299 rows × 12 columns

From the above dataframe it is observed that all rooms are RT4 which means presidential suit. Now since RT4 is a luxurious room it is likely their rent will be higher. To make a fair analysis, we need to do data analysis only on RT4 room types.

In [141…
```
df_bookings[df_bookings.room_category == 'RT4'].revenue_realized.describe()
```

Out[141]:
```
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 [143…
```
# mean + 3*standard deviation

23439 + 3*9048
```

Out[143]:
```
50583
```

The higher limit comes to be 50583 and in our dataframe above we can see that max value for revenue realized is 45220. Hence we can conclude that there is no outlier and we don't need to do any data cleaning on this particular column

In [146…
```
df_bookings.isnull().sum()
```

Out[146]:
```
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
```

Total values in our dataframe is 134576. Out of that 77899 rows has null rating. Since there are many rows with null rating, we should not filter these values. Also we should not replace this rating with a median or mean rating.

(4) Cleaning 'fact_aggregated_bookings' dataset

In [7]:
```
df_agg_bookings.isnull().sum()
```

Out[7]:
```
property_id            0
check_in_date          0
room_category          0
successful_bookings    0
capacity               2
dtype: int64
```

In [9]:
```
df_agg_bookings[df_agg_bookings.capacity.isna()]
```

Out[9]:

	property_id	check_in_date	room_category	successful_bookings	capacity
8	17561	1-May-22	RT1	22	NaN
14	17562	1-May-22	RT1	12	NaN

In [11]:
```
df_agg_bookings.capacity.median()
```

Out[11]:
```
25.0
```

In [13]:
```
df_agg_bookings.capacity.fillna(df_agg_bookings.capacity.median(), inplace = True)
```

In [15]:
```
df_agg_bookings.loc[[8,15]]
```

Out[15]:

	property_id	check_in_date	room_category	successful_bookings	capacity
8	17561	1-May-22	RT1	22	25.0
15	17563	1-May-22	RT1	21	25.0

In [17]:
```
df_agg_bookings[df_agg_bookings.successful_bookings > df_agg_bookings.capacity]
```

Out[17]:

	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 [19]:
```
df_agg_bookings.shape
```

Out[19]:
```
(9200, 5)
```

In [21]:
```
df_agg_bookings = df_agg_bookings[df_agg_bookings.successful_bookings <= df_agg_bookings.capacity]
df_agg_bookings.shape
```

Out[21]:
```
(9194, 5)
```

---

3. Data Transformation

Create occupancy percentage column

In [26]:
```
df_agg_bookings.head(3)
```

In [26]:
```
df_agg_bookings.head(3)
```

Out[26]:

	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

In [33]:
```
df_agg_bookings['occ_pct'] = df_agg_bookings['successful_bookings']/df_agg_bookings['capacity']
df_agg_bookings.head(3)
```

Out[33]:

	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

Convert it to a percentage value

In [35]:
```
df_agg_bookings['occ_pct'] = df_agg_bookings['occ_pct'].apply(lambda x: round(x*100, 2))
df_agg_bookings.head(3)
```

Out[35]:

	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

---

4. Insights Generation

---

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

In [39]:
```
df_agg_bookings.head(3)
```

Out[39]:

	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

In [47]:
```
df_agg_bookings.groupby('room_category')['occ_pct'].mean()
```

Out[47]:
```
room_category
RT1    57.889643
RT2    58.009756
RT3    58.028213
RT4    59.277925
Name: occ_pct, dtype: float64
```

In [57]:
```
df = pd.merge(df_agg_bookings, df_rooms,
              left_on = 'room_category', right_on = 'room_id')
df.head(3)
```

Out[57]:

	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

In [59]:
```
df.drop('room_id', axis = 1, inplace = True)
df.head(4)
```

Out[59]:

	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
3	16558	1-May-22	RT1	18	19.0	94.74	Standard

In [61]:
```
df.groupby('room_class')['occ_pct'].mean()
```

Out[61]:
```
room_class
Elite           58.009756
Premium         58.028213
Presidential    59.277925
Standard        57.889643
Name: occ_pct, dtype: float64
```

2. Print average occupancy rate per city

In [64]:
```
df_hotels.head(3)
```

Out[64]:

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

In [66]:
```
df = pd.merge(df, df_hotels, on = 'property_id')
df.head(3)
```

Out[66]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city
0	16559	1-May-22	RT1	25	30.0	83.33	Standard	Atliq Exotica	Luxury	Mumbai
1	16559	2-May-22	RT1	20	30.0	66.67	Standard	Atliq Exotica	Luxury	Mumbai
2	16559	3-May-22	RT1	17	30.0	56.67	Standard	Atliq Exotica	Luxury	Mumbai

In [70]:
```
df.groupby('city')['occ_pct'].mean().round(2)
```

Out[70]:
```
city
Bangalore    56.33
Delhi        61.51
Hyderabad    58.12
Mumbai       57.91
Name: occ_pct, dtype: float64
```

3. When was the occupancy better? Weekday or Weekend?

In [73]:
```
df_date.head(3)
```

Out[73]:

	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 [75]:
```
df = pd.merge(df, df_date,
              left_on = 'check_in_date', right_on = 'date')
df.head(3)
```

Out[75]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city	date
0	16559	10-May-22	RT1	18	30.0	60.00	Standard	Atliq Exotica	Luxury	Mumbai	10-May-22
1	16559	10-May-22	RT2	25	41.0	60.98	Elite	Atliq Exotica	Luxury	Mumbai	10-May-22
2	16559	10-May-22	RT3	20	32.0	62.50	Premium	Atliq Exotica	Luxury	Mumbai	10-May-22

In [77]:
```
df.groupby('day_type')['occ_pct'].mean().round(2)
```

Out[77]:
```
day_type
weekeday    50.88
weekend     72.34
Name: occ_pct, dtype: float64
```

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

In [80]:
```
df_june_22 = df[df['mmm yy'] == 'Jun 22']
df_june_22.head(3)
```

Out[80]:

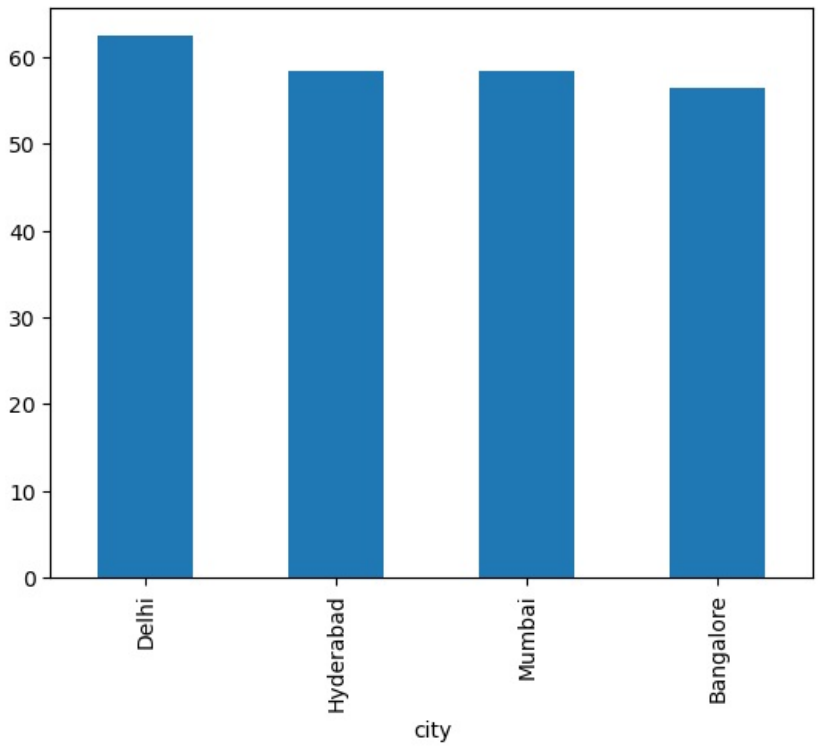
	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city	da
2200	16559	10-Jun-22	RT1	20	30.0	66.67	Standard	Atliq Exotica	Luxury	Mumbai	1 Ju 2
2201	16559	10-Jun-22	RT2	26	41.0	63.41	Elite	Atliq Exotica	Luxury	Mumbai	1 Ju 2
2202	16559	10-Jun-22	RT3	20	32.0	62.50	Premium	Atliq Exotica	Luxury	Mumbai	1 Ju 2

In [82]:
```
df_june_22.groupby('city')['occ_pct'].mean().round(2).sort_values(ascending = False)
```

Out[82]:
```
city
Delhi        62.47
Hyderabad    58.46
Mumbai       58.38
Bangalore    56.44
Name: occ_pct, dtype: float64
```

In [86]:
```
df_june_22.groupby('city')['occ_pct'].mean().round(2).sort_values(ascending = False).plot(kind = 'bar')
```

Out[86]:
```
<Axes: xlabel='city'>
```



5. We got new data for the month of August. Append that to existing data.

In [90]:
```
df_august = pd.read_csv("datasets/new_data_august.csv")
df_august.head(3)
```

Out[90]:

	property_id	property_name	category	city	room_category	room_class	check_in_date	mmm yy	week no	day_type	successful_bookings
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
2	19563	Atliq Palace	Business	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	weekeday	23

In [92]:
```
df_august.columns
```

Out[92]:
```
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 [94]:
```
df.columns
```

Out[94]:
```
Index(['property_id', 'check_in_date', 'room_category', 'successful_bookings',
       'capacity', 'occ_pct', 'room_class', 'property_name', 'category',
       'city', 'date', 'mmm yy', 'week no', 'day_type'],
      dtype='object')
```

In [96]:
```
df_august.shape
```

Out[96]:
```
(7, 13)
```

In [98]:
```
df.shape
```

Out[98]:
```
(6497, 14)
```

In [100…
```
latest_df = pd.concat([df, df_august], ignore_index = True, axis = 0)
latest_df.tail(3)
```

Out[100]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city
6501	19560	01-Aug-22	RT1	20	26.0	NaN	Standard	Atliq City	Business	Bangalore
6502	17561	01-Aug-22	RT1	18	26.0	NaN	Standard	Atliq Blu	Luxury	Mumbai
6503	17564	01-Aug-22	RT1	10	16.0	NaN	Standard	Atliq Seasons	Business	Mumbai

In [102…
```
latest_df.shape
```

Out[102]:
```
(6504, 15)
```

6. Print revenue realized per city

In [149…
```
df_bookings.head(3)
```

Out[149]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given
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
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	4.0

In [151…
```
df_hotels.head(3)
```

Out[151]:

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

In [153…
```
df_bookings_all = pd.merge(df_bookings, df_hotels,
                           on = 'property_id')
df_bookings_all.head(3)
```

Out[153]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given
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
2	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	4.0

In [155…
```
df_bookings_all.groupby('city')['revenue_realized'].sum()
```

Out[155]:
```
city
Bangalore    420383550
Delhi        294404488
Hyderabad    325179310
Mumbai       668569251
Name: revenue_realized, dtype: int64
```

7. Print month by month revenue.

In [158…
```
df_date.head(3)
```

Out[158]:

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

In [160…
```
df_date['mmm yy'].unique()
```

Out[160]:
```
array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

In [162…
```
df_bookings_all.head(3)
```

Out[162]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given
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
2	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	4.0

In [164…
```
df_date.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 92 entries, 0 to 91
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   date      92 non-null     object
 1   mmm yy    92 non-null     object
 2   week no   92 non-null     object
 3   day_type  92 non-null     object
dtypes: object(4)
memory usage: 3.0+ KB
```

In [168…
```
df_date['date'] = pd.to_datetime(df_date['date'], format = '%d/%m/%Y')
df_date.head(3)
```

Out[168]:

	date	mmm yy	week no	day_type
0	2022-05-01	May 22	W 19	weekend
1	2022-05-02	May 22	W 19	weekday
2	2022-05-03	May 22	W 19	weekday

In [170…
```
df_bookings_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134573 entries, 0 to 134572
Data columns (total 15 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   booking_id         134573 non-null  object 
 1   property_id        134573 non-null  int64  
 2   booking_date       134573 non-null  object 
 3   check_in_date      134573 non-null  object 
 4   checkout_date      134573 non-null  object 
 5   no_guests          134573 non-null  float64
 6   room_category      134573 non-null  object 
 7   booking_platform   134573 non-null  object 
 8   ratings_given      56676 non-null   float64
 9   booking_status     134573 non-null  object 
 10  revenue_generated  134573 non-null  int64  
 11  revenue_realized   134573 non-null  int64  
 12  property_name      134573 non-null  object 
 13  category           134573 non-null  object 
 14  city               134573 non-null  object 
dtypes: float64(2), int64(3), object(10)
memory usage: 15.4+ MB
```

In [186…
```
df_bookings_all['check_in_date'] = pd.to_datetime(df_bookings_all['check_in_date'], format = 'mixed')
df_bookings_all.head(3)
```

Out[186]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given
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
2	May012216558RT16	16558	1/5/2022	2022-01-05	3/5/2022	2.0	RT1	others	4.0

In [188…
```
df_bookings_all = pd.merge(df_bookings_all, df_date,
                           left_on = 'check_in_date', right_on = 'date')
df_bookings_all.head(3)
```

Out[188]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given
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 [190…
```
df_bookings_all.groupby('mmm yy')['revenue_realized'].sum()
```

Out[190]:
```
mmm yy
Jul 22    389940912
Jun 22    377191229
May 22    408375641
Name: revenue_realized, dtype: int64
```

8. Revenue realized per hotel type

In [195…
```
df_bookings_all.property_name.unique()
```

Out[195]:
```
array(['Atliq Grands', 'Atliq Exotica', 'Atliq City', 'Atliq Blu',
       'Atliq Bay', 'Atliq Palace', 'Atliq Seasons'], dtype=object)
```

In [199…
```
df_bookings_all.groupby('property_name')['revenue_realized'].sum().sort_values()
```

Out[199]:
```
property_name
Atliq Seasons     45920757
Atliq Grands     145860641
Atliq Blu        179203544
Atliq Bay        179416721
Atliq City       196555383
Atliq Palace     209474575
Atliq Exotica    219076161
Name: revenue_realized, dtype: int64
```

9. Average rating per city

In [204…
```
df_bookings_all.groupby('city')['ratings_given'].mean().round(2)
```

Out[204]:
```
city
Bangalore    3.40
Delhi        3.78
Hyderabad    3.66
Mumbai       3.64
Name: ratings_given, dtype: float64
```

10. Revenue realized per booking platform

In [207…
```
df_bookings_all.groupby('booking_platform')['revenue_realized'].sum().plot(
    kind = 'pie')
```

Out[207]:
```
<Axes: ylabel='revenue_realized'>
```