

AtliQ Hotels Data Analysis Project

```
In [2]: import pandas as pd
```

```
In [3]: df_bookings=pd.read_csv("fact_bookings.csv")
df_bookings.head(4)
```

```
Out[3]:
```

| | booking_id | property_id | booking_date | check_in_date | checkout_date | no_guests |
|---|------------------|-------------|--------------|---------------|---------------|-----------|
| 0 | May012216558RT11 | 16558 | 27-04-22 | 1/5/2022 | 2/5/2022 | -3.0 |
| 1 | May012216558RT12 | 16558 | 30-04-22 | 1/5/2022 | 2/5/2022 | 2.0 |
| 2 | May012216558RT13 | 16558 | 28-04-22 | 1/5/2022 | 4/5/2022 | 2.0 |
| 3 | May012216558RT14 | 16558 | 28-04-22 | 1/5/2022 | 2/5/2022 | -2.0 |

◀ ▶

```
In [4]: df_bookings.shape
```

```
Out[4]: (134590, 12)
```

```
In [5]: df_bookings.room_category.unique()
```

```
Out[5]: array(['RT1', 'RT2', 'RT3', 'RT4'], dtype=object)
```

```
In [6]: df_bookings.booking_platform.unique()
```

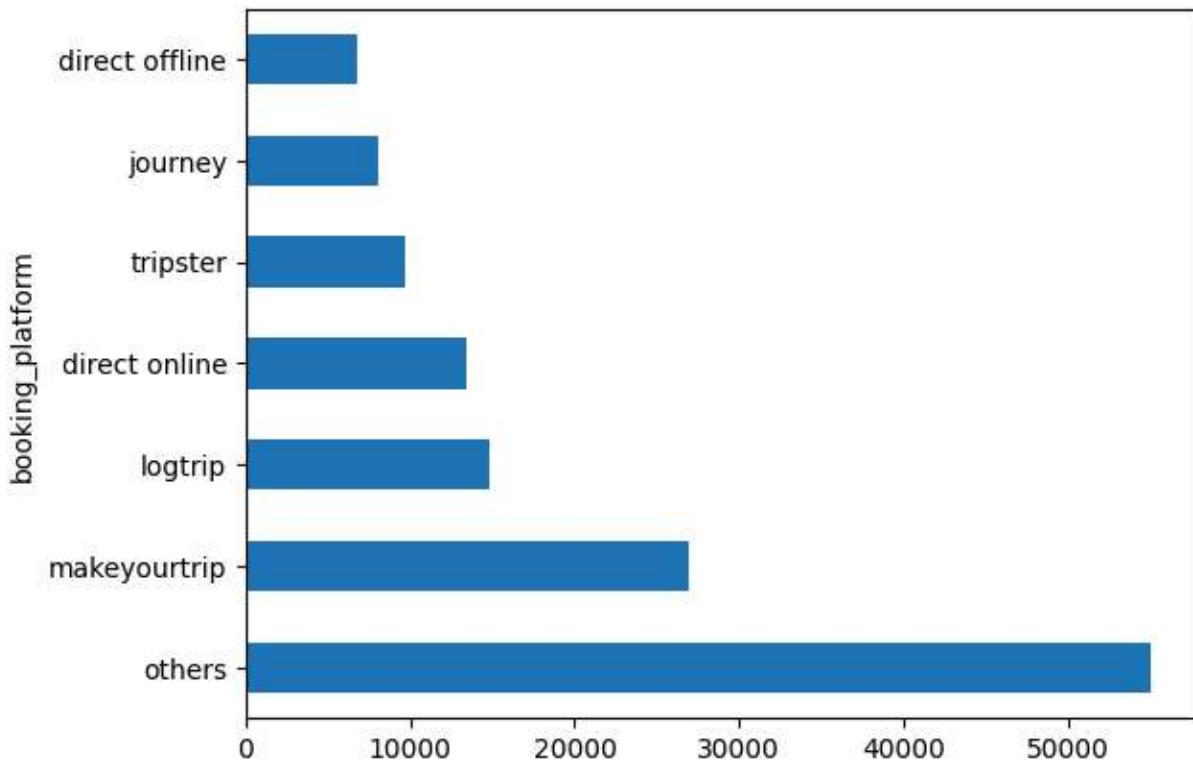
```
Out[6]: array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',
   'journey', 'direct offline'], dtype=object)
```

```
In [7]: df_bookings.booking_platform.value_counts()
```

```
Out[7]: booking_platform
others           55066
makeyourtrip     26898
logtrip          14756
direct online    13379
tripster         9630
journey          8106
direct offline   6755
Name: count, dtype: int64
```

```
In [8]: df_bookings.booking_platform.value_counts().plot(kind="barh")
```

```
Out[8]: <Axes: ylabel='booking_platform'>
```



In [9]: `df_bookings.describe()`

| | 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 [10]: `df_bookings.revenue_generated.min(), df_bookings.revenue_generated.max()`

Out[10]: `(np.int64(6500), np.int64(28560000))`

In [11]: `df_date = pd.read_csv('dim_date.csv')
df_hotels = pd.read_csv('dim_hotels.csv')
df_rooms = pd.read_csv('dim_rooms.csv')
df_agg_bookings = pd.read_csv('fact_aggregated_bookings.csv')`

In [12]: `df_hotels.shape`

Out[12]: (25, 4)

In [13]: df_hotels.head(4)

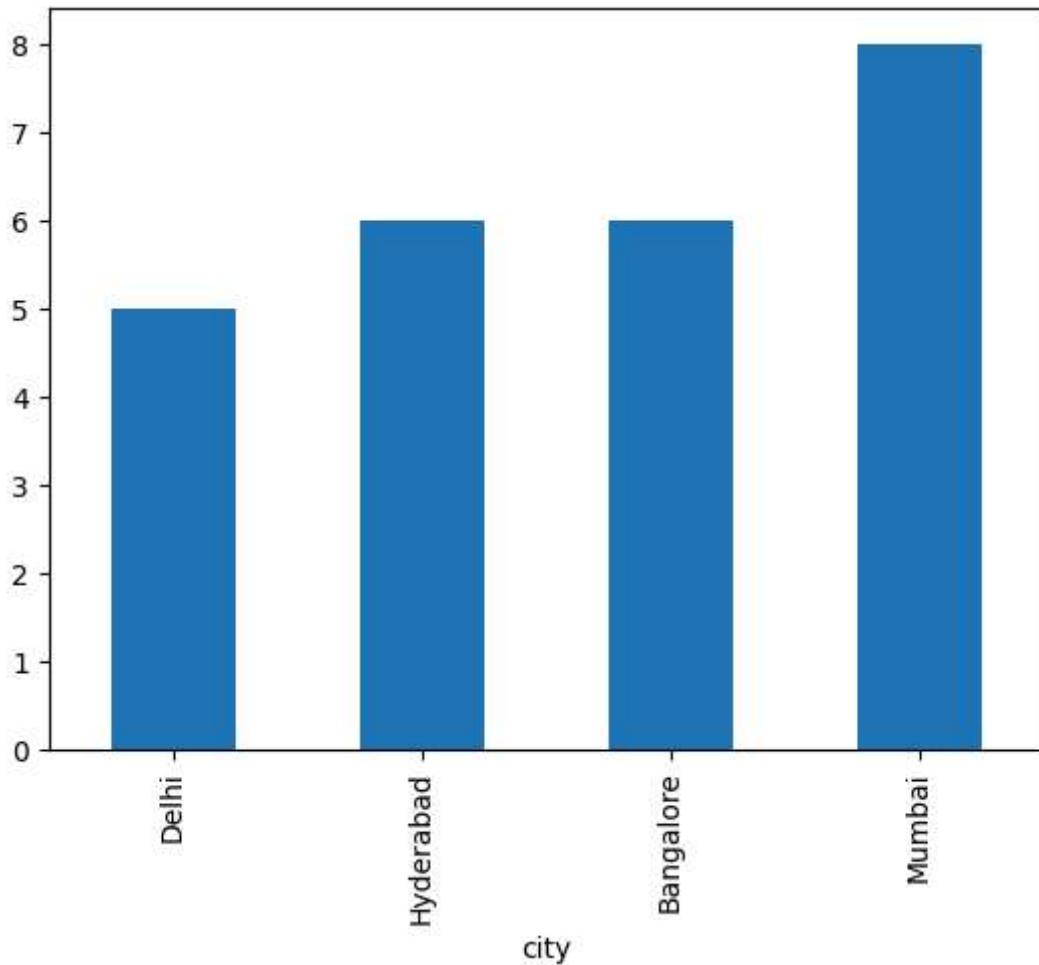
| | property_id | property_name | category | city |
|---|-------------|---------------|----------|--------|
| 0 | 16558 | Atliq Grands | Luxury | Delhi |
| 1 | 16559 | Atliq Exotica | Luxury | Mumbai |
| 2 | 16560 | Atliq City | Business | Delhi |
| 3 | 16561 | Atliq Blu | Luxury | Delhi |

In [14]: df_hotels.category.value_counts()

```
Out[14]: category
Luxury      16
Business     9
Name: count, dtype: int64
```

In [15]: df_hotels.city.value_counts().sort_values().plot(kind="bar")

Out[15]: <Axes: xlabel='city'>



Data Cleaning

In [16]: `df_bookings.describe()`

| | 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 [17]: `df_bookings[df_bookings.no_guests<=0]`

| | booking_id | property_id | booking_date | check_in_date | checkout_date | no_ |
|---------------|-------------------|--------------------|---------------------|----------------------|----------------------|------------|
| 0 | May012216558RT11 | 16558 | 27-04-22 | 1/5/2022 | 2/5/2022 | |
| 3 | May012216558RT14 | 16558 | 28-04-22 | 1/5/2022 | 2/5/2022 | |
| 17924 | May122218559RT44 | 18559 | 12/5/2022 | 12/5/2022 | 14-05-22 | |
| 18020 | May122218561RT22 | 18561 | 8/5/2022 | 12/5/2022 | 14-05-22 | |
| 18119 | May122218562RT311 | 18562 | 5/5/2022 | 12/5/2022 | 17-05-22 | |
| 18121 | May122218562RT313 | 18562 | 10/5/2022 | 12/5/2022 | 17-05-22 | |
| 56715 | Jun082218562RT12 | 18562 | 5/6/2022 | 8/6/2022 | 13-06-22 | |
| 119765 | Jul202219560RT220 | 19560 | 19-07-22 | 20-07-22 | 22-07-22 | |
| 134586 | Jul312217564RT47 | 17564 | 30-07-22 | 31-07-22 | 1/8/2022 | |

◀ ▶

In [18]: `df_bookings=df_bookings[df_bookings.no_guests>0]
df_bookings.shape`

Out[18]: (134578, 12)

In [19]: `df_bookings.revenue_generated.min(), df_bookings.revenue_generated.max()`

Out[19]: (np.int64(6500), np.int64(28560000))

In [20]: `df_bookings.revenue_generated.mean(), df_bookings.revenue_generated.median()`

```
Out[20]: (np.float64(15378.036937686695), np.float64(13500.0))
```

```
In [21]: avg, std = df_bookings.revenue_generated.mean(), df_bookings.revenue_generated.std()
```

```
In [22]: avg, std
```

```
Out[22]: (np.float64(15378.036937686695), np.float64(93040.1549314641))
```

```
In [23]: higher_limit = avg + 3*std  
higher_limit
```

```
Out[23]: np.float64(294498.50173207896)
```

```
In [24]: lower_limit = avg - 3*std  
lower_limit
```

```
Out[24]: np.float64(-263742.4278567056)
```

```
In [25]: df_bookings[df_bookings.revenue_generated > higher_limit]
```

| | booking_id | property_id | booking_date | check_in_date | checkout_date | no_ |
|---------------|-------------------|-------------|--------------|---------------|---------------|-----|
| 2 | May012216558RT13 | 16558 | 28-04-22 | 1/5/2022 | 4/5/2022 | |
| 111 | May012216559RT32 | 16559 | 29-04-22 | 1/5/2022 | 2/5/2022 | |
| 315 | May012216562RT22 | 16562 | 28-04-22 | 1/5/2022 | 4/5/2022 | |
| 562 | May012217559RT118 | 17559 | 26-04-22 | 1/5/2022 | 2/5/2022 | |
| 129176 | Jul282216562RT26 | 16562 | 21-07-22 | 28-07-22 | 29-07-22 | |

◀ ▶

```
In [26]: df_bookings = df_bookings[df_bookings.revenue_generated <= higher_limit]  
df_bookings.shape
```

```
Out[26]: (134573, 12)
```

```
In [27]: df_bookings.revenue_realized.describe()
```

```
Out[27]: 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 [28]: higher_limit = df_bookings.revenue_realized.mean() + 3*df_bookings.revenue_realized.std()  
higher_limit
```

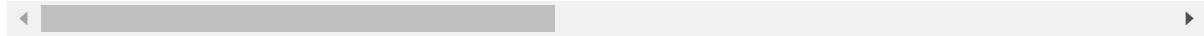
Out[28]: np.float64(33479.358661845814)

In [29]: df_bookings[df_bookings.revenue_realized>higher_limit]

Out[29]:

| | booking_id | property_id | booking_date | check_in_date | checkout_date | no_ |
|---------------|-------------------|-------------|--------------|---------------|---------------|-----|
| 137 | May012216559RT41 | 16559 | 27-04-22 | 1/5/2022 | 7/5/2022 | |
| 139 | May012216559RT43 | 16559 | 1/5/2022 | 1/5/2022 | 2/5/2022 | |
| 143 | May012216559RT47 | 16559 | 28-04-22 | 1/5/2022 | 3/5/2022 | |
| 149 | May012216559RT413 | 16559 | 24-04-22 | 1/5/2022 | 7/5/2022 | |
| 222 | May012216560RT45 | 16560 | 30-04-22 | 1/5/2022 | 3/5/2022 | |
| ... | ... | ... | ... | ... | ... | ... |
| 134328 | Jul312219560RT49 | 19560 | 31-07-22 | 31-07-22 | 2/8/2022 | |
| 134331 | Jul312219560RT412 | 19560 | 31-07-22 | 31-07-22 | 1/8/2022 | |
| 134467 | Jul312219562RT45 | 19562 | 28-07-22 | 31-07-22 | 1/8/2022 | |
| 134474 | Jul312219562RT412 | 19562 | 25-07-22 | 31-07-22 | 6/8/2022 | |
| 134581 | Jul312217564RT42 | 17564 | 31-07-22 | 31-07-22 | 1/8/2022 | |

1299 rows × 12 columns



In [30]: df_bookings[df_bookings.room_category=="RT4"].revenue_realized.describe()

Out[30]: 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 [31]: 23439+3*9048

Out[31]: 50583

In [32]: df_bookings.isnull().sum()

```
Out[32]: 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 []:

Data Transformation

In [33]: `df_agg_bookings.head(3)`

| | 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 [34]: `df_agg_bookings['occ_pct'] = df_agg_bookings.apply(lambda row: row['successful_bookings'] / row['capacity'] * 100, axis=1)`In [35]: `df_agg_bookings['occ_pct'] = df_agg_bookings['occ_pct'].apply(lambda x: round(x*100, 2))`

| | 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 []:

Insight Generation

1. Average occupancy rate in each of the room categories

In [36]: `df_agg_bookings.groupby("room_category")["occ_pct"].mean().round(2)`

```
Out[36]: room_category
RT1    58.22
RT2    58.04
RT3    58.03
RT4    59.30
Name: occ_pct, dtype: float64
```

```
In [37]: df = pd.merge(df_agg_bookings, df_rooms, left_on="room_category", right_on="room_id")
df.head(4)
```

| | property_id | check_in_date | room_category | successful_bookings | capacity | occ_pct | roon |
|----------|-------------|---------------|---------------|---------------------|----------|---------|--------|
| 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 |
| 3 | 17558 | 1-May-22 | RT1 | | 30 | 19.0 | 157.89 |

```
In [38]: df.groupby("room_class")["occ_pct"].mean().round(2)
```

```
Out[38]: room_class
Elite      58.04
Premium    58.03
Presidential 59.30
Standard   58.22
Name: occ_pct, dtype: float64
```

```
In [39]: df.drop("room_id", axis=1, inplace=True)
df.head(4)
```

| | property_id | check_in_date | room_category | successful_bookings | capacity | occ_pct | roon | |
|----------|-------------|---------------|---------------|---------------------|----------|---------|--------|----|
| 0 | 16559 | 1-May-22 | RT1 | | 25 | 30.0 | 83.33 | St |
| 1 | 19562 | 1-May-22 | RT1 | | 28 | 30.0 | 93.33 | St |
| 2 | 19563 | 1-May-22 | RT1 | | 23 | 30.0 | 76.67 | St |
| 3 | 17558 | 1-May-22 | RT1 | | 30 | 19.0 | 157.89 | St |

2. Print average occupancy rate per city

```
In [40]: df_hotels.head(3)
```

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

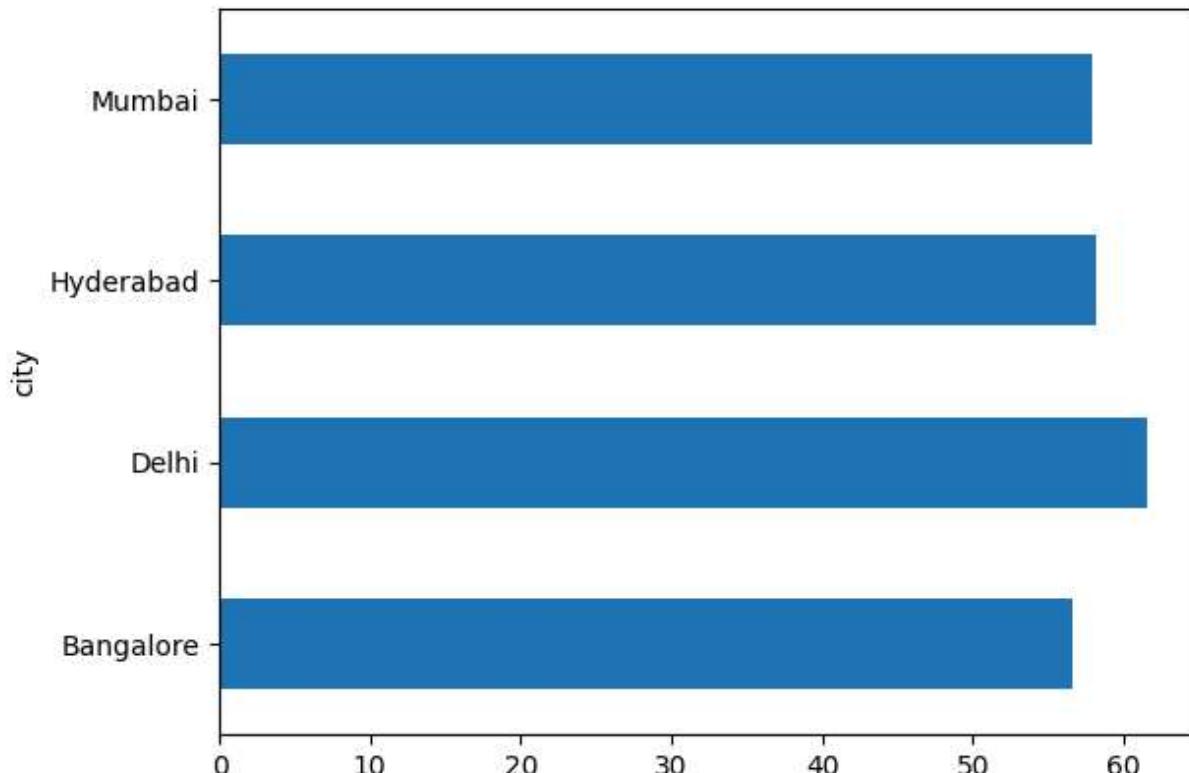
```
In [41]: df = pd.merge(df, df_hotels, on="property_id")
df.head(3)
```

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

◀ ▶

```
In [44]: df.groupby("city")["occ_pct"].mean().plot(kind="barh")
```

```
Out[44]: <Axes: ylabel='city'>
```



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

```
In [45]: df_date.head(3)
```

| | 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 [48]: df = pd.merge(df, df_date, left_on="check_in_date", right_on="date")
df.head(3)
```

| | property_id | check_in_date | room_category | successful_bookings | capacity | occ_pct | roon |
|---|-------------|---------------|---------------|---------------------|----------|---------|------|
| 0 | 19563 | 10-May-22 | RT3 | 15 | 29.0 | 51.72 | Pr |
| 1 | 18560 | 10-May-22 | RT1 | 19 | 30.0 | 63.33 | St |
| 2 | 19562 | 10-May-22 | RT1 | 18 | 30.0 | 60.00 | St |

```
In [49]: df.groupby("day_type")["occ_pct"].mean().round(2)
```

```
Out[49]: day_type
weekeday    50.90
weekend     72.39
Name: occ_pct, dtype: float64
```

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

```
In [50]: df_june_22 = df[df["mmm yy"]=="Jun 22"]
df_june_22.head(4)
```

Out[50]:

| | property_id | check_in_date | room_category | successful_bookings | capacity | occ_pct | r |
|------|-------------|---------------|---------------|---------------------|----------|---------|---|
| 2200 | 16559 | 10-Jun-22 | RT1 | 20 | 30.0 | 66.67 | |
| 2201 | 19562 | 10-Jun-22 | RT1 | 19 | 30.0 | 63.33 | |
| 2202 | 19563 | 10-Jun-22 | RT1 | 17 | 30.0 | 56.67 | |
| 2203 | 17558 | 10-Jun-22 | RT1 | 9 | 19.0 | 47.37 | |

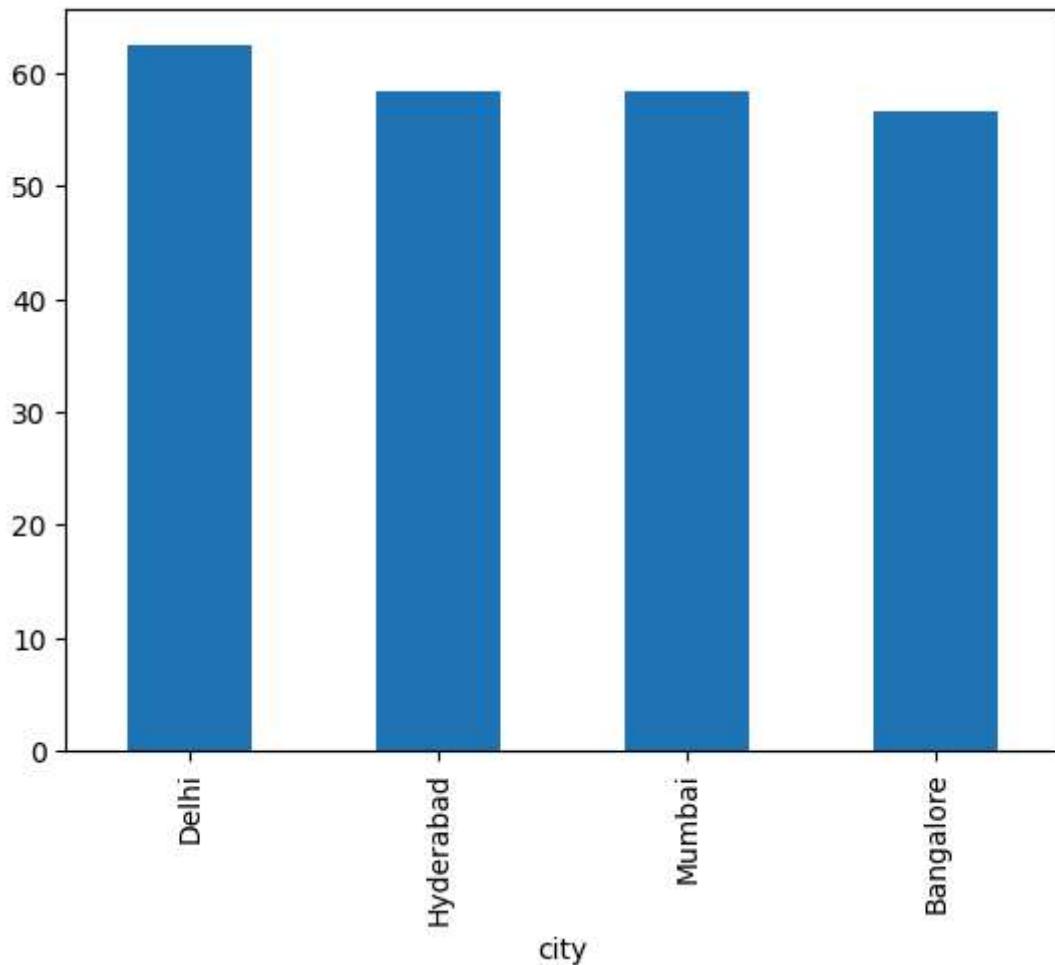


In [51]: df_june_22.groupby('city')['occ_pct'].mean().round(2).sort_values(ascending=False)

```
Out[51]: city
Delhi      62.47
Hyderabad  58.46
Mumbai     58.38
Bangalore   56.58
Name: occ_pct, dtype: float64
```

In [52]: df_june_22.groupby('city')['occ_pct'].mean().round(2).sort_values(ascending=False).

Out[52]: <Axes: xlabel='city'>



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

```
In [53]: df_august = pd.read_csv("new_data_august.csv")
df_august.head(3)
```

Out[53]:

| | property_id | property_name | category | city | room_category | room_class | check_in_date |
|---|-------------|---------------|----------|-----------|---------------|------------|---------------|
| 0 | 16559 | Atliq Exotica | Luxury | Mumbai | RT1 | Standard | 01-Aug-2024 |
| 1 | 19562 | Atliq Bay | Luxury | Bangalore | RT1 | Standard | 01-Aug-2024 |
| 2 | 19563 | Atliq Palace | Business | Bangalore | RT1 | Standard | 01-Aug-2024 |

◀ ▶

```
In [54]: df_august.columns
```

Out[54]: 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 [55]: `df.columns`

Out[55]: `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 [56]: `df_august.shape`

Out[56]: `(7, 13)`

In [57]: `latest_df = pd.concat([df, df_august], ignore_index = True, axis = 0)`
`latest_df.tail(10)`

Out[57]:

| | <code>property_id</code> | <code>check_in_date</code> | <code>room_category</code> | <code>successful_bookings</code> | <code>capacity</code> | <code>occ_pct</code> | <code>r</code> |
|-------------|--------------------------|----------------------------|----------------------------|----------------------------------|-----------------------|----------------------|----------------|
| 6497 | 17558 | 31-Jul-22 | RT4 | 3 | 6.0 | 50.0 | F |
| 6498 | 19563 | 31-Jul-22 | RT4 | 3 | 6.0 | 50.0 | F |
| 6499 | 17561 | 31-Jul-22 | RT4 | 3 | 4.0 | 75.0 | F |
| 6500 | 16559 | 01-Aug-22 | RT1 | 30 | 30.0 | NaN | |
| 6501 | 19562 | 01-Aug-22 | RT1 | 21 | 30.0 | NaN | |
| 6502 | 19563 | 01-Aug-22 | RT1 | 23 | 30.0 | NaN | |
| 6503 | 19558 | 01-Aug-22 | RT1 | 30 | 40.0 | NaN | |
| 6504 | 19560 | 01-Aug-22 | RT1 | 20 | 26.0 | NaN | |
| 6505 | 17561 | 01-Aug-22 | RT1 | 18 | 26.0 | NaN | |
| 6506 | 17564 | 01-Aug-22 | RT1 | 10 | 16.0 | NaN | |



| | | | | | | | |
|-------------|-------|-----------|-----|----|------|------|---|
| 6497 | 17558 | 31-Jul-22 | RT4 | 3 | 6.0 | 50.0 | F |
| 6498 | 19563 | 31-Jul-22 | RT4 | 3 | 6.0 | 50.0 | F |
| 6499 | 17561 | 31-Jul-22 | RT4 | 3 | 4.0 | 75.0 | F |
| 6500 | 16559 | 01-Aug-22 | RT1 | 30 | 30.0 | NaN | |
| 6501 | 19562 | 01-Aug-22 | RT1 | 21 | 30.0 | NaN | |
| 6502 | 19563 | 01-Aug-22 | RT1 | 23 | 30.0 | NaN | |
| 6503 | 19558 | 01-Aug-22 | RT1 | 30 | 40.0 | NaN | |
| 6504 | 19560 | 01-Aug-22 | RT1 | 20 | 26.0 | NaN | |
| 6505 | 17561 | 01-Aug-22 | RT1 | 18 | 26.0 | NaN | |
| 6506 | 17564 | 01-Aug-22 | RT1 | 10 | 16.0 | NaN | |

In [58]: `latest_df.shape`

Out[58]: `(6507, 15)`

6. Print revenue realized per city

```
In [59]: df_hotels.head(3)
df_bookings_all = pd.merge(df_bookings, df_hotels, on="property_id")
df_bookings_all.head(3)
```

Out[59]:

| | booking_id | property_id | booking_date | check_in_date | checkout_date | no_guests |
|----------|------------------|-------------|--------------|---------------|---------------|-----------|
| 0 | May012216558RT12 | 16558 | 30-04-22 | 1/5/2022 | 2/5/2022 | 2.0 |
| 1 | May012216558RT15 | 16558 | 27-04-22 | 1/5/2022 | 2/5/2022 | 4.0 |
| 2 | May012216558RT16 | 16558 | 1/5/2022 | 1/5/2022 | 3/5/2022 | 2.0 |

```
In [60]: df_bookings_all.groupby("city")["revenue_realized"].sum()
```

Out[60]: city

| | |
|-----------|-----------|
| Bangalore | 420383550 |
| Delhi | 294404488 |
| Hyderabad | 325179310 |
| Mumbai | 668569251 |

Name: revenue_realized, dtype: int64

7. Print month by month revenue

```
In [65]: df_date["mmm yy"].unique()
```

Out[65]: array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)

```
In [66]: 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 [75]: df_date["date"] = pd.to_datetime(df_date["date"], format='%d-%m-%y')
df_date.head(3)
```

Out[75]:

| | 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 [76]: df_date['formatted_date'] = df_date['date'].dt.strftime('%d-%m-%y')
print(df_date)
```

| | date | mmm | yy | week | no | day_type | formatted_date |
|----|------------|-----|-----|------|-----|----------|----------------|
| 0 | 2022-05-01 | May | 22 | W | 19 | weekend | 01-05-22 |
| 1 | 2022-05-02 | May | 22 | W | 19 | weekday | 02-05-22 |
| 2 | 2022-05-03 | May | 22 | W | 19 | weekday | 03-05-22 |
| 3 | 2022-05-04 | May | 22 | W | 19 | weekday | 04-05-22 |
| 4 | 2022-05-05 | May | 22 | W | 19 | weekday | 05-05-22 |
| .. | ... | ... | ... | ... | ... | ... | ... |
| 87 | 2022-07-27 | Jul | 22 | W | 31 | weekday | 27-07-22 |
| 88 | 2022-07-28 | Jul | 22 | W | 31 | weekday | 28-07-22 |
| 89 | 2022-07-29 | Jul | 22 | W | 31 | weekday | 29-07-22 |
| 90 | 2022-07-30 | Jul | 22 | W | 31 | weekend | 30-07-22 |
| 91 | 2022-07-31 | Jul | 22 | W | 32 | weekend | 31-07-22 |

[92 rows x 5 columns]

```
In [77]: print(df_bookings_all['check_in_date'].dtype)
```

object

```
In [82]: df_bookings_all["check_in_date"] = pd.to_datetime(df_bookings_all["check_in_date"],
```

```
In [83]: print(df_bookings_all['check_in_date'].dtype)
```

datetime64[ns]

```
In [84]: df_date['formatted_date'] = df_bookings_all['check_in_date'].dt.strftime('%d-%m-%y')
print(df_date)
```

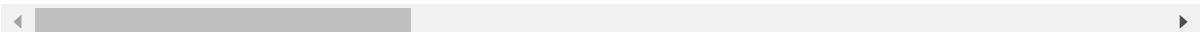
| | date | mmm | yy | week | no | day_type | formatted_date |
|----|------------|-----|-----|------|-----|----------|----------------|
| 0 | 2022-05-01 | May | 22 | W | 19 | weekend | 05-01-22 |
| 1 | 2022-05-02 | May | 22 | W | 19 | weekday | 05-01-22 |
| 2 | 2022-05-03 | May | 22 | W | 19 | weekday | 05-01-22 |
| 3 | 2022-05-04 | May | 22 | W | 19 | weekday | 05-01-22 |
| 4 | 2022-05-05 | May | 22 | W | 19 | weekday | 05-01-22 |
| .. | ... | ... | ... | ... | ... | ... | ... |
| 87 | 2022-07-27 | Jul | 22 | W | 31 | weekday | 05-01-22 |
| 88 | 2022-07-28 | Jul | 22 | W | 31 | weekday | 05-01-22 |
| 89 | 2022-07-29 | Jul | 22 | W | 31 | weekday | 05-01-22 |
| 90 | 2022-07-30 | Jul | 22 | W | 31 | weekend | 05-01-22 |
| 91 | 2022-07-31 | Jul | 22 | W | 32 | weekend | 05-01-22 |

[92 rows x 5 columns]

```
In [86]: df_bookings_all = pd.merge(df_bookings_all, df_date, left_on="check_in_date", right_on="check_in_date")
df_bookings_all.head(3)
```

Out[86]:

| | booking_id | property_id | booking_date | check_in_date | checkout_date | no_guests |
|---|------------------|-------------|--------------|---------------|---------------|-----------|
| 0 | May052216558RT11 | 16558 | 15-04-22 | 2022-05-05 | 7/5/2022 | 3.0 |
| 1 | May052216558RT12 | 16558 | 30-04-22 | 2022-05-05 | 7/5/2022 | 2.0 |
| 2 | May052216558RT13 | 16558 | 1/5/2022 | 2022-05-05 | 6/5/2022 | 3.0 |



In [87]: df_bookings_all.groupby("mmm yy")["revenue_realized"].sum()

```
Out[87]: mmm yy
Jul 22    389940912
Jun 22    377191229
May 22    408375641
Name: revenue_realized, dtype: int64
```

Revenue realized per hotel type

In [88]: df_bookings_all.groupby("property_name")["revenue_realized"].sum().round(2).sort_values(ascending=False)

```
Out[88]: 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
```

Average rating per city

In [89]: df_bookings_all.groupby("city")["ratings_given"].mean().round(2)

```
Out[89]: city
Bangalore      3.40
Delhi          3.78
Hyderabad      3.66
Mumbai          3.64
Name: ratings_given, dtype: float64
```

pie chart of revenue realized per booking platform

In [90]: df_bookings_all.groupby("booking_platform")["revenue_realized"].sum().plot(kind="pi

Out[90]: <Axes: ylabel='revenue_realized'>

