AtliQ Hotel Analysis by Python

July 30, 2024

#ATLIQ GRAND HOTEL ANALYSIS

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

[3]:
```
df_bookings = pd.read_csv(r"C:
↪\Users\KONGEGOWTAM\Downloads\64101194a2364\source-code\3_project_hospitality_analysis\datas
↪csv")
```

[4]:
```
df_bookings.head()
```

[4]:
```
          booking_id  property_id booking_date check_in_date checkout_date  \
0  May012216558RT11        16558     27-04-22      1/5/2022      2/5/2022
1  May012216558RT12        16558     30-04-22      1/5/2022      2/5/2022
2  May012216558RT13        16558     28-04-22      1/5/2022      4/5/2022
3  May012216558RT14        16558     28-04-22      1/5/2022      2/5/2022
4  May012216558RT15        16558     27-04-22      1/5/2022      2/5/2022

   no_guests room_category booking_platform  ratings_given booking_status  \
0       -3.0           RT1    direct online            1.0    Checked Out
1        2.0           RT1           others            NaN      Cancelled
2        2.0           RT1          logtrip            5.0    Checked Out
3       -2.0           RT1           others            NaN      Cancelled
4        4.0           RT1    direct online            5.0    Checked Out

   revenue_generated  revenue_realized
0              10010             10010
1               9100              3640
2            9100000              9100
3               9100              3640
4              10920             10920
```

[ ]:
```

```

[5]:
```
df_bookings.shape
```

[5]:
```
(134590, 12)
```

[6]:
```
df_bookings.room_category.unique()
```

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

```
[7]: df_bookings.booking_platform.unique()
```

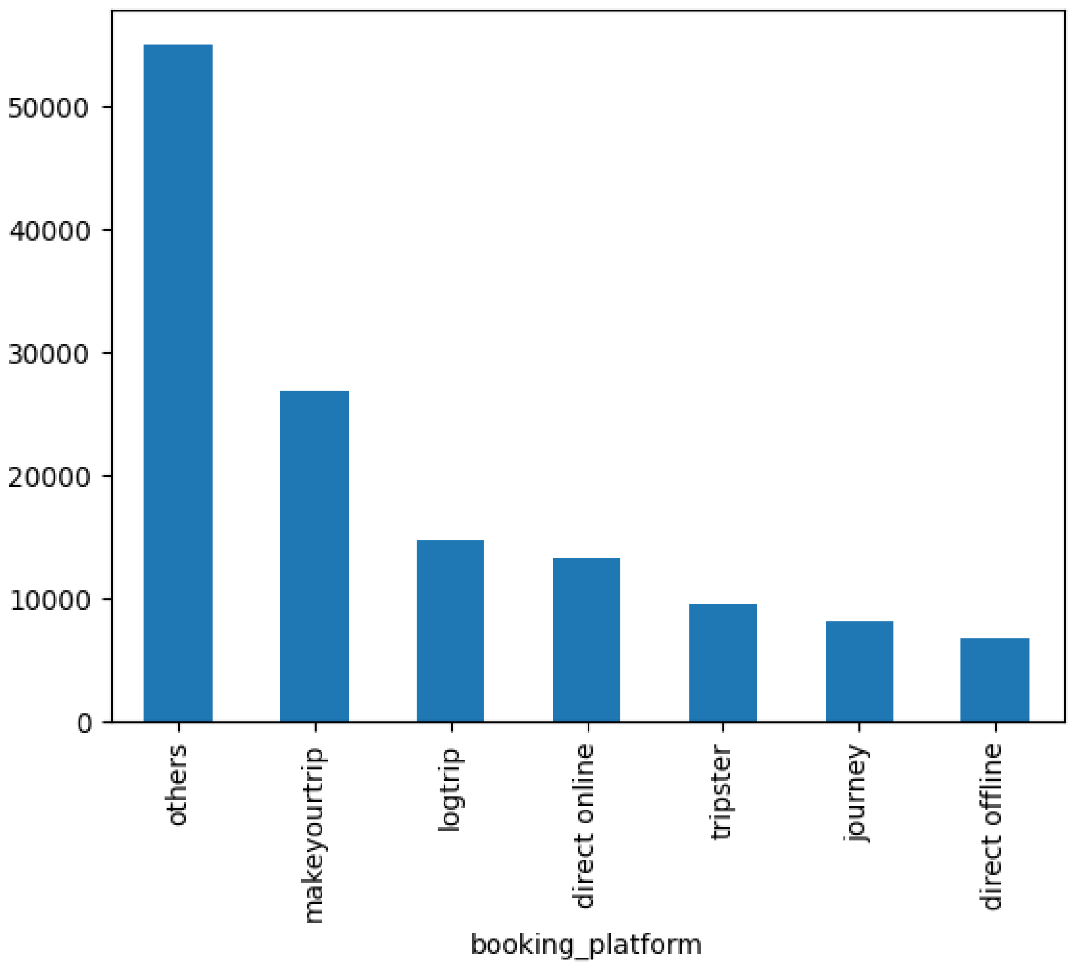
```
[7]: array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',
            'journey', 'direct offline'], dtype=object)
```

```
[8]: df_bookings.booking_platform.value_counts()
```

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

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

```
[9]: <Axes: xlabel='booking_platform'>
```



[10]:
```
df_bookings.describe()
```

[10]:
```
         property_id      no_guests  ratings_given  revenue_generated  \
count  134590.000000  134587.000000   56683.000000       1.345900e+05
mean    18061.113493       2.036170       3.619004       1.537805e+04
std      1093.055847       1.034885       1.235009       9.303604e+04
min     16558.000000     -17.000000       1.000000       6.500000e+03
25%     17558.000000       1.000000       3.000000       9.900000e+03
50%     17564.000000       2.000000       4.000000       1.350000e+04
75%     18563.000000       2.000000       5.000000       1.800000e+04
max     19563.000000       6.000000       5.000000       2.856000e+07

       revenue_realized
count     134590.000000
mean       12696.123256
std         6928.108124
min         2600.000000
```

```
25%        7600.000000
50%       11700.000000
75%       15300.000000
max       45220.000000
```

[11]:
```
df_bookings.revenue_generated.min(),df_bookings.revenue_generated.max()
```

[11]:
```
(np.int64(6500), np.int64(28560000))
```

[12]:
```
df_bookings = pd.read_csv(r"C:
 ↪\Users\KONGEGOWTAM\Downloads\64101194a2364\source-code\3_project_hospitality_analysis\datas
 ↪csv")
df_date = pd.read_csv(r"C:
 ↪\Users\KONGEGOWTAM\Downloads\64101194a2364\source-code\3_project_hospitality_analysis\datas
 ↪csv")
df_hotels = pd.read_csv(r"C:
 ↪\Users\KONGEGOWTAM\Downloads\64101194a2364\source-code\3_project_hospitality_analysis\datas
 ↪csv")
df_rooms = pd.read_csv(r"C:
 ↪\Users\KONGEGOWTAM\Downloads\64101194a2364\source-code\3_project_hospitality_analysis\datas
 ↪csv")
df_agg_bookings = pd.read_csv(r"C:
 ↪\Users\KONGEGOWTAM\Downloads\64101194a2364\source-code\3_project_hospitality_analysis\datas
 ↪csv")
```

[13]:
```
df_hotels.shape
```

[13]:
```
(25, 4)
```

[14]:
```
df_date.head()
```

[14]:
```
        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
3  04-May-22  May 22     W 19  weekeday
4  05-May-22  May 22     W 19  weekeday
```

[15]:
```
df_hotels.head()
```

[15]:
```
   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
4        16562      Atliq Bay    Luxury   Delhi
```

```
[123]: df_rooms
```

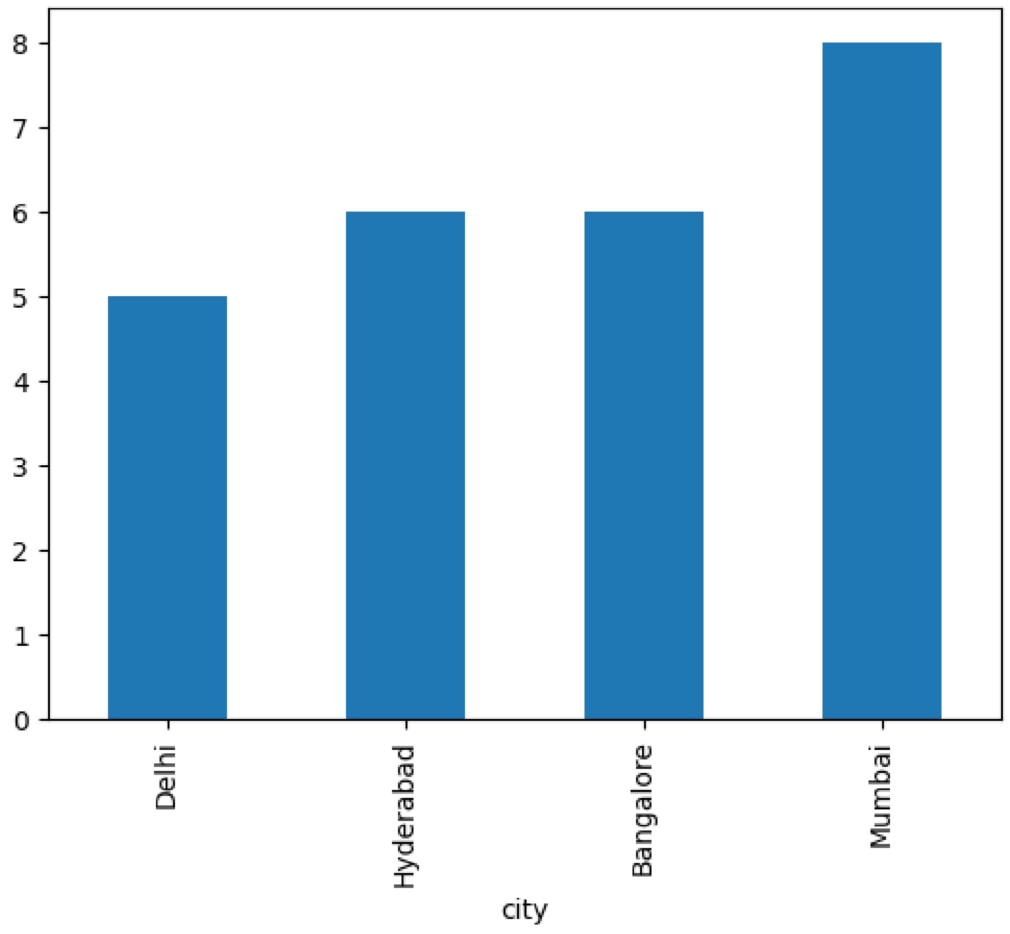
```
[123]:   room_id    room_class
       0     RT1      Standard
       1     RT2         Elite
       2     RT3       Premium
       3     RT4  Presidential
```

```
[18]: df_hotels.category.value_counts()
```

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

```
[19]: df_hotels.city.value_counts().sort_values().plot(kind = 'bar')
```

```
[19]: <Axes: xlabel='city'>
```



1 Exercise: Explore aggregate bookings

```
[20]: df_agg_bookings.head()
```

```
[20]:    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
```

2 Exercise-1. Find out unique property ids in aggregate bookings dataset

```
[21]: df_agg_bookings['property_id'].unique()
```

```
[21]: 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])
```

3 Exercise-2. Find out total bookings per property_id

```
[22]: total_bookings_per_property_id = df_agg_bookings.
        ↪groupby('property_id')['successful_bookings'].sum().reset_index()
      total_bookings_per_property_id
```

```
[22]:     property_id  successful_bookings
      0         16558                 3153
      1         16559                 7338
      2         16560                 4693
      3         16561                 4418
      4         16562                 4820
      5         16563                 7211
      6         17558                 5053
      7         17559                 6142
      8         17560                 6013
      9         17561                 5183
      10        17562                 3424
      11        17563                 6337
      12        17564                 3982
      13        18558                 4475
      14        18559                 5256
      15        18560                 6638
      16        18561                 6458
      17        18562                 7333
      18        18563                 4737
```

```
19         19558                    4400
20         19559                    4729
21         19560                    6079
22         19561                    5736
23         19562                    5812
24         19563                    5413
```

4 Exercise-3. Find out days on which bookings are greater than capacity

[23]:
```
df_agg_bookings[df_agg_bookings.successful_bookings>df_agg_bookings.capacity]
```

[23]:
```
      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
```

**Exercise-4. Find out properties that have highest capacity*

[24]:
```
df_agg_bookings["capacity"].max()
```

[24]:
```
np.float64(50.0)
```

5 Data Cleaning

[25]:
```
df_bookings.describe()
```

[25]:
```
         property_id      no_guests  ratings_given  revenue_generated  \
count  134590.000000  134587.000000   56683.000000       1.345900e+05
mean    18061.113493       2.036170       3.619004       1.537805e+04
std      1093.055847       1.034885       1.235009       9.303604e+04
min     16558.000000     -17.000000       1.000000       6.500000e+03
25%     17558.000000       1.000000       3.000000       9.900000e+03
50%     17564.000000       2.000000       4.000000       1.350000e+04
75%     18563.000000       2.000000       5.000000       1.800000e+04
max     19563.000000       6.000000       5.000000       2.856000e+07

       revenue_realized
count     134590.000000
mean       12696.123256
std         6928.108124
min         2600.000000
25%         7600.000000
```

```
50%          11700.000000
75%          15300.000000
max          45220.000000
```

[33]:
```
df_bookings[df_bookings.no_guests<=0]
```

[33]:
```
                  booking_id  property_id booking_date check_in_date  \
0         May012216558RT11        16558     27-04-22      1/5/2022
3         May012216558RT14        16558     28-04-22      1/5/2022
17924     May122218559RT44        18559    12/5/2022     12/5/2022
18020     May122218561RT22        18561     8/5/2022     12/5/2022
18119    May122218562RT311        18562     5/5/2022     12/5/2022
18121    May122218562RT313        18562    10/5/2022     12/5/2022
56715     Jun082218562RT12        18562     5/6/2022      8/6/2022
119765   Jul202219560RT220        19560     19-07-22      20-07-22
134586    Jul312217564RT47        17564     30-07-22      31-07-22

       checkout_date  no_guests room_category booking_platform  ratings_given  \
0           2/5/2022       -3.0           RT1    direct online            1.0
3           2/5/2022       -2.0           RT1           others            NaN
17924       14-05-22      -10.0           RT4    direct online            NaN
18020       14-05-22      -12.0           RT2      makeyourtrip            NaN
18119       17-05-22       -6.0           RT3   direct offline            5.0
18121       17-05-22       -4.0           RT3    direct online            NaN
56715       13-06-22      -17.0           RT1           others            NaN
119765      22-07-22       -1.0           RT2           others            NaN
134586      1/8/2022       -4.0           RT4          logtrip            2.0

       booking_status  revenue_generated  revenue_realized
0         Checked Out              10010             10010
3           Cancelled               9100              3640
17924         No Show              20900             20900
18020       Cancelled               9000              3600
18119     Checked Out              16800             16800
18121       Cancelled              14400              5760
56715     Checked Out               6500              6500
119765    Checked Out              13500             13500
134586    Checked Out              38760             38760
```

[34]:
```
df_bookings.shape
```

[34]:
```
(134590, 12)
```

[35]:
```
df_bookings=df_bookings[df_bookings.no_guests>0]
```

[36]:
```
df_bookings.shape
```

[36]:
```
(134578, 12)
```

[37]:
```
df_bookings.revenue_generated.min(),df_bookings.revenue_generated.max()
```

[37]:
```
(np.int64(6500), np.int64(28560000))
```

[39]:
```
avg,std = df_bookings.revenue_generated.mean(),df_bookings.revenue_generated.
↪std()
```

[ ]:
```
avg, std
```

[40]:
```
higher_limit = avg+3*std
higher_limit
```

[40]:
```
np.float64(294498.50173207896)
```

[41]:
```
lower_limit = avg-3*std
lower_limit
```

[41]:
```
np.float64(-263742.4278567056)
```

[42]:
```
df_bookings[df_bookings.revenue_generated>higher_limit]
```

[42]:
```
               booking_id  property_id booking_date check_in_date  \
2        May012216558RT13        16558     28-04-22      1/5/2022
111      May012216559RT32        16559     29-04-22      1/5/2022
315      May012216562RT22        16562     28-04-22      1/5/2022
562     May012217559RT118        17559     26-04-22      1/5/2022
129176   Jul282216562RT26        16562     21-07-22      28-07-22

       checkout_date  no_guests room_category booking_platform  ratings_given  \
2           4/5/2022        2.0           RT1          logtrip            5.0
111         2/5/2022        6.0           RT3    direct online            NaN
315         4/5/2022        2.0           RT2   direct offline            3.0
562         2/5/2022        2.0           RT1           others            NaN
129176      29-07-22        2.0           RT2    direct online            3.0

       booking_status  revenue_generated  revenue_realized
2         Checked Out            9100000              9100
111       Checked Out           28560000             28560
315       Checked Out           12600000             12600
562         Cancelled            2000000              4420
129176    Checked Out           10000000             12600
```

[43]:
```
df_bookigs = df_bookings[df_bookings.revenue_generated<higher_limit]
df_bookings.shape
```

[43]:
```
(134578, 12)
```

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

[44]:
```
count    134578.000000
mean      12696.011822
std        6927.841641
min        2600.000000
25%        7600.000000
50%       11700.000000
75%       15300.000000
max       45220.000000
Name: revenue_realized, dtype: float64
```

[45]:
```
higher_limit = df_bookings.revenue_realized.mean() +3*df_bookings.
 ↪revenue_realized.std()
higher_limit
```

[45]:
```
np.float64(33479.53674501214)
```

[46]:
```
df_bookings[df_bookings.revenue_realized>higher_limit]
```

[46]:
```
                booking_id  property_id booking_date check_in_date  \
137      May012216559RT41        16559     27-04-22      1/5/2022
139      May012216559RT43        16559     1/5/2022      1/5/2022
143      May012216559RT47        16559     28-04-22      1/5/2022
149     May012216559RT413        16559     24-04-22      1/5/2022
222      May012216560RT45        16560     30-04-22      1/5/2022
…                      …           …          …           …
134328   Jul312219560RT49        19560     31-07-22      31-07-22
134331  Jul312219560RT412        19560     31-07-22      31-07-22
134467   Jul312219562RT45        19562     28-07-22      31-07-22
134474  Jul312219562RT412        19562     25-07-22      31-07-22
134581   Jul312217564RT42        17564     31-07-22      31-07-22

       checkout_date  no_guests room_category booking_platform  ratings_given  \
137         7/5/2022        4.0           RT4           others            NaN
139         2/5/2022        6.0           RT4         tripster            3.0
143         3/5/2022        3.0           RT4           others            5.0
149         7/5/2022        5.0           RT4          logtrip            NaN
222         3/5/2022        5.0           RT4           others            3.0
…                …        …            …               …              …
134328      2/8/2022        6.0           RT4    direct online            5.0
134331      1/8/2022        6.0           RT4           others            2.0
134467      1/8/2022        6.0           RT4     makeyourtrip            4.0
134474      6/8/2022        5.0           RT4   direct offline            5.0
134581      1/8/2022        4.0           RT4     makeyourtrip            4.0

       booking_status  revenue_generated  revenue_realized
```

```
137        Checked Out             38760             38760
139        Checked Out             45220             45220
143        Checked Out             35530             35530
149        Checked Out             41990             41990
222        Checked Out             34580             34580
…                …                …                …
134328     Checked Out             39900             39900
134331     Checked Out             39900             39900
134467     Checked Out             39900             39900
134474     Checked Out             37050             37050
134581     Checked Out             38760             38760

[1299 rows x 12 columns]
```

[47]:
```
df_rooms
```

[47]:
```
  room_id    room_class
0     RT1      Standard
1     RT2         Elite
2     RT3       Premium
3     RT4  Presidential
```

[48]:
```
df_bookings[df_bookings.room_category =="RT4"].revenue_realized.describe()
```

[48]:
```
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
```

[49]:
```
23439+3*9048
```

[49]:
```
50583
```

[50]:
```
df_bookings.isnull().sum()
```

[50]:
```
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        77899
booking_status           0
revenue_generated        0
revenue_realized         0
dtype: int64
```

[51]:
```
df_agg_bookings[df_agg_bookings.capacity.isna()]
```

[51]:
```
    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
```

[52]:
```
df_agg_bookings.capacity.median()
```

[52]:
```
np.float64(25.0)
```

[53]:
```
df_agg_bookings.capacity.fillna(df_agg_bookings.capacity.median(),inplace =␣
  ↪True)
```

```
C:\Users\KONGEGOWTAM\AppData\Local\Temp\ipykernel_15560\4155640710.py:1:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df_agg_bookings.capacity.fillna(df_agg_bookings.capacity.median(),inplace =
True)
```

[54]:
```
df_agg_bookings.loc[[8,14]]
```

[54]:
```
    property_id check_in_date room_category  successful_bookings  capacity
8         17561      1-May-22           RT1                   22      25.0
14        17562      1-May-22           RT1                   12      25.0
```

6 Data Transformation

[55]:
```
df_agg_bookings.head()
```

[55]:
```
   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
```

[56]:
```
df_agg_bookings["occ_pct"] = df_agg_bookings["successful_bookings"]/
 ↪df_agg_bookings["capacity"]
```

[57]:
```
df_agg_bookings.head()
```

[57]:
```
   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

    occ_pct
0  0.833333
1  0.933333
2  0.766667
3  1.578947
4  0.947368
```

[58]:
```
df_agg_bookings["occ_pct"] = df_agg_bookings["occ_pct"].apply(lambda x:␣
 ↪round(x*100,2))
df_agg_bookings.head(4)
```

[58]:
```
   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

   occ_pct
0    83.33
1    93.33
2    76.67
3   157.89
```

[59]:
```
df_agg_bookings["occ_pct"] = df_agg_bookings["occ_pct"].apply(lambda x: round(x␣
 ↪* 100, 2))
df_agg_bookings.head(4)
```

[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
3        17558      1-May-22           RT1                   30      19.0
```

```
   occ_pct
0   8333.0
1   9333.0
2   7667.0
3  15789.0
```

7 what is an average occupancy rate in each of the room categoreis?

[60]:
```
df_agg_bookings.groupby("room_category")["occ_pct"].mean().round(2)
```

[60]:
```
room_category
RT1    5823.27
RT2    5804.03
RT3    5802.82
RT4    5930.05
Name: occ_pct, dtype: float64
```

[61]:
```
df_rooms
```

[61]:
```
  room_id    room_class
0     RT1      Standard
1     RT2         Elite
2     RT3       Premium
3     RT4  Presidential
```

[62]:
```
df = pd.merge(df_agg_bookings, df_rooms, left_on="room_category",␣
 ↪right_on="room_id")
df.tail(4)
```

[62]:
```
      property_id check_in_date room_category  successful_bookings  capacity  \
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

      occ_pct room_id    room_class
9196   7222.0     RT4  Presidential
9197   5000.0     RT4  Presidential
9198   5000.0     RT4  Presidential
9199   7500.0     RT4  Presidential
```

[63]:
```
df.groupby("room_class")["occ_pct"].mean().round(2)
```

[63]:
```
room_class
Elite           5804.03
Premium         5802.82
Presidential    5930.05
Standard        5823.27
Name: occ_pct, dtype: float64
```

[64]:
```
df.drop("room_id", axis=1, inplace = True)
df.head(4)
```

[64]:
```
   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

   occ_pct room_class
0   8333.0   Standard
1   9333.0   Standard
2   7667.0   Standard
3  15789.0   Standard
```

8 print average occupancy rate per city

[65]:
```
df_hotels.head(3)
```

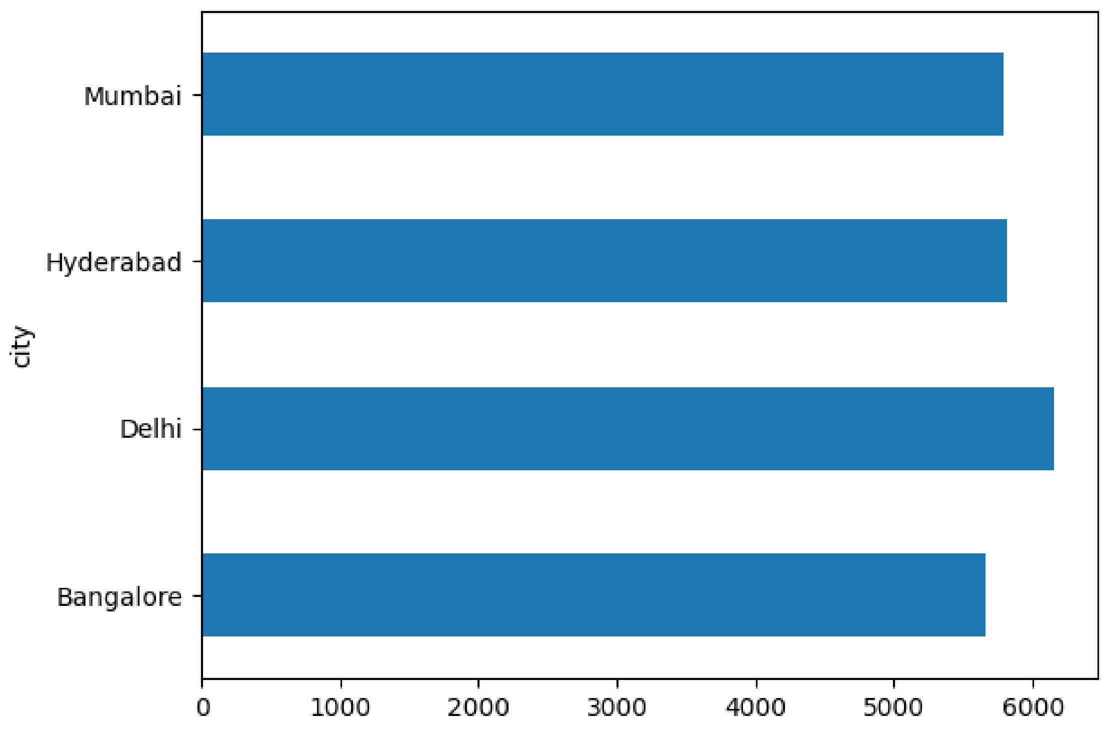
[65]:
```
   property_id  property_name  category    city
0        16558   Atliq Grands    Luxury   Delhi
1        16559  Atliq Exotica    Luxury  Mumbai
2        16560     Atliq City  Business   Delhi
```

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

[66]:
```
   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

   occ_pct room_class  property_name  category       city
0   8333.0   Standard  Atliq Exotica    Luxury     Mumbai
1   9333.0   Standard      Atliq Bay    Luxury  Bangalore
2   7667.0   Standard   Atliq Palace  Business  Bangalore
```

[67]:
```
df.groupby("city")["occ_pct"].mean().plot(kind="barh")
```

[67]:
```
<Axes: ylabel='city'>
```



9 when was the occupancy better? Weekday or Weekend?

```
[111]: df.head
```

```
[111]: <bound method NDFrame.head of    property_name  revenue_realized
       3  Atliq Exotica          32436799
       5   Atliq Palace          30945855
       2     Atliq City          29047727
       0      Atliq Bay          26936115
       1      Atliq Blu          26459751
       4   Atliq Grands          21644446
       6  Atliq Seasons           6672245>
```

```
[108]: df_date.head(3)
```

```
[108]:         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
       2 2022-05-03  May 22    W 19  weekeday
```

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

```
[69]:    property_id check_in_date room_category  successful_bookings  capacity  \
      0        19563     10-May-22           RT3                   15      29.0
      1        18560     10-May-22           RT1                   19      30.0
      2        19562     10-May-22           RT1                   18      30.0

         occ_pct room_class property_name  category       city       date  mmm yy  \
      0   5172.0    Premium  Atliq Palace  Business  Bangalore  10-May-22  May 22
      1   6333.0   Standard    Atliq City  Business  Hyderabad  10-May-22  May 22
      2   6000.0   Standard     Atliq Bay    Luxury  Bangalore  10-May-22  May 22

        week no  day_type
      0    W 20  weekeday
      1    W 20  weekeday
      2    W 20  weekeday
```

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

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

```
[70]: df["mmm yy"].unique()
```

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

```
[71]: df_june_22 = df[df["mmm yy"] == "Jun 22"]
      df_june_22.head(3)
```

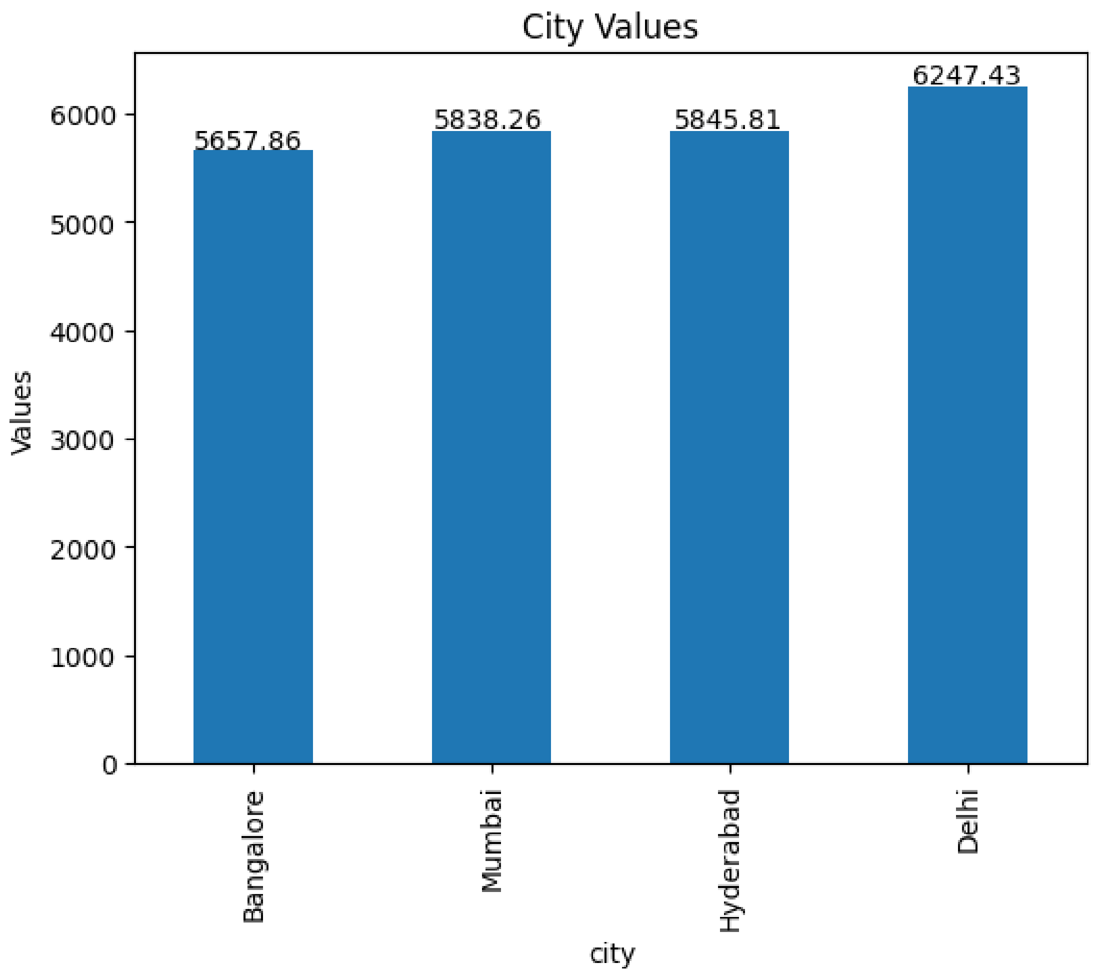
```
[71]:       property_id check_in_date room_category  successful_bookings  capacity  \
      2200        16559     10-Jun-22           RT1                   20      30.0
      2201        19562     10-Jun-22           RT1                   19      30.0
      2202        19563     10-Jun-22           RT1                   17      30.0

            occ_pct room_class property_name  category       city       date  \
      2200   6667.0   Standard Atliq Exotica    Luxury     Mumbai  10-Jun-22
      2201   6333.0   Standard     Atliq Bay    Luxury  Bangalore  10-Jun-22
      2202   5667.0   Standard  Atliq Palace  Business  Bangalore  10-Jun-22

            mmm yy week no  day_type
      2200  Jun 22    W 24  weekeday
      2201  Jun 22    W 24  weekeday
      2202  Jun 22    W 24  weekeday
```

```
[72]: df_june_22.groupby("city")["occ_pct"].mean().round(2).sort_values(ascending =␣
       ↪False)
```

```
[72]: city
      Delhi        6247.43
      Hyderabad    5845.81
```

```
Mumbai       5838.26
Bangalore    5657.86
Name: occ_pct, dtype: float64
```

[83]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Sample data
data = {
    'city': ['Delhi', 'Hyderabad', 'Mumbai', 'Bangalore'],
    'value': [6247.43, 5845.81, 5838.26, 5657.86]
}
df_hotels = pd.DataFrame(data)

# Sort values
df_hotels = df_hotels.sort_values(by='value')

# Create the bar plot
ax = df_hotels.plot(kind='bar', x='city', y='value', legend=False)

# Add data labels
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() * 1.005, p.get_height() * 1.
  ↪005))

# Display the plot
plt.ylabel('Values')
plt.title('City Values')
plt.show()
```



```
[74]: df_august = pd.read_csv(r"C:
       ↪\Users\KONGEGOWTAM\Downloads\64101194a2364\source-code\3_project_hospitality_analysis\datas
       ↪csv")
      df_august.head(3)
```

```
[74]:    property_id  property_name  category       city room_category room_class  \
      0        16559  Atliq Exotica    Luxury     Mumbai           RT1   Standard
      1        19562      Atliq Bay    Luxury  Bangalore           RT1   Standard
      2        19563   Atliq Palace  Business  Bangalore           RT1   Standard

        check_in_date  mmm yy week no  day_type  successful_bookings  capacity  \
      0     01-Aug-22  Aug-22    W 32  weekeday                   30        30
      1     01-Aug-22  Aug-22    W 32  weekeday                   21        30
      2     01-Aug-22  Aug-22    W 32  weekeday                   23        30

           occ%
```

```
0   100.00
1    70.00
2    76.67
```

[112]:
```
df_august.columns
```

[112]:
```
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')
```

[75]:
```
df_august.shape
```

[75]:
```
(7, 13)
```

[76]:
```
df.shape
```

[76]:
```
(6500, 14)
```

[120]:
```
df.rein
```

[120]:
```
   property_name  revenue_realized
3  Atliq Exotica          32436799
5   Atliq Palace          30945855
2     Atliq City          29047727
0      Atliq Bay          26936115
1      Atliq Blu          26459751
4   Atliq Grands          21644446
6  Atliq Seasons           6672245
```

[114]:
```
df_august
```

[114]:
```
   property_id  property_name  category       city room_category room_class  \
0        16559  Atliq Exotica    Luxury     Mumbai           RT1   Standard
1        19562      Atliq Bay    Luxury  Bangalore           RT1   Standard
2        19563   Atliq Palace  Business  Bangalore           RT1   Standard
3        19558   Atliq Grands    Luxury  Bangalore           RT1   Standard
4        19560     Atliq City  Business  Bangalore           RT1   Standard
5        17561      Atliq Blu    Luxury     Mumbai           RT1   Standard
6        17564  Atliq Seasons  Business     Mumbai           RT1   Standard

  check_in_date  mmm yy week no  day_type  successful_bookings  capacity  \
0     01-Aug-22  Aug-22    W 32  weekeday                   30        30
1     01-Aug-22  Aug-22    W 32  weekeday                   21        30
2     01-Aug-22  Aug-22    W 32  weekeday                   23        30
3     01-Aug-22  Aug-22    W 32  weekeday                   30        40
4     01-Aug-22  Aug-22    W 32  weekeday                   20        26
5     01-Aug-22  Aug-22    W 32  weekeday                   18        26
```

```
6     01-Aug-22  Aug-22    W 32  weekeday                   10        16

     occ%
0  100.00
1   70.00
2   76.67
3   75.00
4   76.92
5   69.23
6   62.50
```

[77]:
```
latest_df = pd.concat([df,df_august], ignore_index=True, axis =0)
latest_df.head(5)
```

[77]:
```
   property_id check_in_date room_category  successful_bookings  capacity  \
0        19563     10-May-22           RT3                   15      29.0
1        18560     10-May-22           RT1                   19      30.0
2        19562     10-May-22           RT1                   18      30.0
3        19563     10-May-22           RT1                   16      30.0
4        17558     10-May-22           RT1                   11      19.0

   occ_pct room_class property_name  category       city       date  mmm yy  \
0   5172.0    Premium  Atliq Palace  Business  Bangalore  10-May-22  May 22
1   6333.0   Standard    Atliq City  Business  Hyderabad  10-May-22  May 22
2   6000.0   Standard     Atliq Bay    Luxury  Bangalore  10-May-22  May 22
3   5333.0   Standard  Atliq Palace  Business  Bangalore  10-May-22  May 22
4   5789.0   Standard  Atliq Grands    Luxury     Mumbai  10-May-22  May 22

  week no  day_type  occ%
0    W 20  weekeday   NaN
1    W 20  weekeday   NaN
2    W 20  weekeday   NaN
3    W 20  weekeday   NaN
4    W 20  weekeday   NaN
```

[78]:
```
latest_df.shape
```

[78]:
```
(6507, 15)
```

11 print revenue realized per city

[79]:
```
df_bookings.head(4)
```

[79]:
```
         booking_id  property_id booking_date check_in_date checkout_date  \
1  May012216558RT12        16558     30-04-22      1/5/2022      2/5/2022
2  May012216558RT13        16558     28-04-22      1/5/2022      4/5/2022
4  May012216558RT15        16558     27-04-22      1/5/2022      2/5/2022
```

```
5  May012216558RT16        16558     1/5/2022      1/5/2022      3/5/2022

   no_guests room_category booking_platform  ratings_given booking_status  \
1        2.0           RT1           others            NaN      Cancelled
2        2.0           RT1          logtrip            5.0    Checked Out
4        4.0           RT1    direct online            5.0    Checked Out
5        2.0           RT1           others            4.0    Checked Out

   revenue_generated  revenue_realized
1               9100              3640
2            9100000              9100
4              10920             10920
5               9100              9100
```

[80]:
```
df_hotels.head(8)
```

[80]:
```
   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
4        16562      Atliq Bay    Luxury   Delhi
5        16563   Atliq Palace  Business   Delhi
6        17558   Atliq Grands    Luxury  Mumbai
7        17559  Atliq Exotica    Luxury  Mumbai
```

[81]:
```
df_bookings_all = pd.merge(df_bookings,df_hotels, on ="property_id")
df_bookings_all.head(3)
```

[81]:
```
         booking_id  property_id booking_date check_in_date checkout_date  \
0  May012216558RT12        16558     30-04-22      1/5/2022      2/5/2022
1  May012216558RT13        16558     28-04-22      1/5/2022      4/5/2022
2  May012216558RT15        16558     27-04-22      1/5/2022      2/5/2022

   no_guests room_category booking_platform  ratings_given booking_status  \
0        2.0           RT1           others            NaN      Cancelled
1        2.0           RT1          logtrip            5.0    Checked Out
2        4.0           RT1    direct online            5.0    Checked Out

   revenue_generated  revenue_realized property_name category   city
0               9100              3640  Atliq Grands   Luxury  Delhi
1            9100000              9100  Atliq Grands   Luxury  Delhi
2              10920             10920  Atliq Grands   Luxury  Delhi
```

[82]:
```
df_bookings_all.groupby("city")["revenue_realized"].sum()
```

```
[82]: city
      Bangalore    420383550
      Delhi        294438788
      Hyderabad    325179310
      Mumbai       668602231
      Name: revenue_realized, dtype: int64
```

12 print month by month revenue

```
[84]: df_bookings_all.head(3)
```

```
[84]:          booking_id  property_id booking_date check_in_date checkout_date  \
      0  May012216558RT12        16558     30-04-22      1/5/2022      2/5/2022
      1  May012216558RT13        16558     28-04-22      1/5/2022      4/5/2022
      2  May012216558RT15        16558     27-04-22      1/5/2022      2/5/2022

         no_guests room_category booking_platform  ratings_given booking_status  \
      0        2.0           RT1           others            NaN      Cancelled
      1        2.0           RT1          logtrip            5.0    Checked Out
      2        4.0           RT1   direct online             5.0    Checked Out

         revenue_generated  revenue_realized property_name category   city
      0               9100              3640  Atliq Grands   Luxury  Delhi
      1            9100000              9100  Atliq Grands   Luxury  Delhi
      2              10920             10920  Atliq Grands   Luxury  Delhi
```

```
[85]: df_date["mmm yy"]
```

```
[85]: 0     May 22
      1     May 22
      2     May 22
      3     May 22
      4     May 22
             …
      87    Jul 22
      88    Jul 22
      89    Jul 22
      90    Jul 22
      91    Jul 22
      Name: mmm yy, Length: 92, dtype: object
```

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

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

```
[87]: df_date.head(3)
```

```
[87]:         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
```

```
[88]: df_bookings_all.head(3)
```

```
[88]:          booking_id  property_id booking_date check_in_date checkout_date  \
      0  May012216558RT12        16558     30-04-22      1/5/2022      2/5/2022
      1  May012216558RT13        16558     28-04-22      1/5/2022      4/5/2022
      2  May012216558RT15        16558     27-04-22      1/5/2022      2/5/2022

         no_guests room_category booking_platform  ratings_given booking_status  \
      0        2.0           RT1           others            NaN      Cancelled
      1        2.0           RT1          logtrip            5.0    Checked Out
      2        4.0           RT1    direct online            5.0    Checked Out

         revenue_generated  revenue_realized property_name category   city
      0               9100              3640  Atliq Grands   Luxury  Delhi
      1            9100000              9100  Atliq Grands   Luxury  Delhi
      2              10920             10920  Atliq Grands   Luxury  Delhi
```

```
[89]: 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
```

```
[90]: df_bookings_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134578 entries, 0 to 134577
Data columns (total 15 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   booking_id         134578 non-null  object
 1   property_id        134578 non-null  int64
 2   booking_date       134578 non-null  object
 3   check_in_date      134578 non-null  object
 4   checkout_date      134578 non-null  object
 5   no_guests          134578 non-null  float64
 6   room_category      134578 non-null  object
 7   booking_platform   134578 non-null  object
 8   ratings_given      56679 non-null   float64
 9   booking_status     134578 non-null  object
 10  revenue_generated  134578 non-null  int64
 11  revenue_realized   134578 non-null  int64
 12  property_name      134578 non-null  object
 13  category           134578 non-null  object
 14  city               134578 non-null  object
dtypes: float64(2), int64(3), object(10)
memory usage: 15.4+ MB
```

[91]:
```
df_date["date"] = pd.to_datetime(df_date["date"])
df_date.head(3)
```

```
C:\Users\KONGEGOWTAM\AppData\Local\Temp\ipykernel_15560\173964601.py:1:
UserWarning: Could not infer format, so each element will be parsed
individually, falling back to `dateutil`. To ensure parsing is consistent and
as-expected, please specify a format.
  df_date["date"] = pd.to_datetime(df_date["date"])
```

[91]:
```
        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
2 2022-05-03  May 22    W 19  weekeday
```

[92]:
```
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     datetime64[ns]
 1   mmm yy    92 non-null     object
 2   week no   92 non-null     object
 3   day_type  92 non-null     object
dtypes: datetime64[ns](1), object(3)
memory usage: 3.0+ KB
```

[93]:
```
df_bookings_all["check_in_date"] = pd.
 ↪to_datetime(df_bookings_all["check_in_date"], errors='coerce')
df_bookings_all.head(3)
```

[93]:
```
          booking_id  property_id booking_date check_in_date checkout_date  \
0  May012216558RT12        16558     30-04-22    2022-01-05      2/5/2022
1  May012216558RT13        16558     28-04-22    2022-01-05      4/5/2022
2  May012216558RT15        16558     27-04-22    2022-01-05      2/5/2022
```

```
   no_guests room_category booking_platform  ratings_given booking_status  \
0        2.0           RT1           others            NaN      Cancelled
1        2.0           RT1          logtrip            5.0    Checked Out
2        4.0           RT1    direct online            5.0    Checked Out

   revenue_generated  revenue_realized property_name category   city
0               9100              3640  Atliq Grands   Luxury  Delhi
1            9100000              9100  Atliq Grands   Luxury  Delhi
2              10920             10920  Atliq Grands   Luxury  Delhi
```

[94]:
```
df_bookings_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134578 entries, 0 to 134577
Data columns (total 15 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   booking_id         134578 non-null  object
 1   property_id        134578 non-null  int64
 2   booking_date       134578 non-null  object
 3   check_in_date      55794 non-null   datetime64[ns]
 4   checkout_date      134578 non-null  object
 5   no_guests          134578 non-null  float64
 6   room_category      134578 non-null  object
 7   booking_platform   134578 non-null  object
 8   ratings_given      56679 non-null   float64
 9   booking_status     134578 non-null  object
 10  revenue_generated  134578 non-null  int64
 11  revenue_realized   134578 non-null  int64
 12  property_name      134578 non-null  object
 13  category           134578 non-null  object
 14  city               134578 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(3), object(9)
memory usage: 15.4+ MB
```

[95]:
```
# Ensure both columns are in datetime format
df_bookings_all["check_in_date"] = pd.
 ↪to_datetime(df_bookings_all["check_in_date"], errors='coerce')
df_date["date"] = pd.to_datetime(df_date["date"], errors='coerce')

# Perform the merge
df_bookings_all = pd.merge(df_bookings_all, df_date, left_on="check_in_date",␣
 ↪right_on="date")
df_bookings_all.head(3)
```

[95]:
```
        booking_id  property_id booking_date check_in_date checkout_date  \
0  May052216558RT11        16558     15-04-22    2022-05-05      7/5/2022
```

```
1  May052216558RT12        16558     30-04-22    2022-05-05      7/5/2022
2  May052216558RT13        16558     1/5/2022    2022-05-05      6/5/2022

   no_guests room_category booking_platform  ratings_given booking_status  \
0        3.0           RT1         tripster            5.0    Checked Out
1        2.0           RT1           others            NaN      Cancelled
2        3.0           RT1   direct offline            5.0    Checked Out

   revenue_generated  revenue_realized property_name category   city  \
0              10010             10010  Atliq Grands   Luxury  Delhi
1               9100              3640  Atliq Grands   Luxury  Delhi
2              10010             10010  Atliq Grands   Luxury  Delhi

        date  mmm yy week no  day_type
0 2022-05-05  May 22    W 19  weekeday
1 2022-05-05  May 22    W 19  weekeday
2 2022-05-05  May 22    W 19  weekeday
```

[ ]:

```
[100]: df_bookings_all.groupby("mmm yy")["revenue_realized"].sum()
```

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

Exercise-1. Print revenue realized per hotel type

```
[101]: df_bookings_all.groupby("property_name")["revenue_realized"].sum().round(2).
        ↪reset_index()
```

```
[101]:    property_name  revenue_realized
0      Atliq Bay          26936115
1      Atliq Blu          26459751
2     Atliq City          29047727
3  Atliq Exotica          32436799
4   Atliq Grands          21644446
5   Atliq Palace          30945855
6  Atliq Seasons           6672245
```

Exercise-2 Print average rating per city

```
[97]: df_bookings_all.groupby("city")["ratings_given"].mean()
```

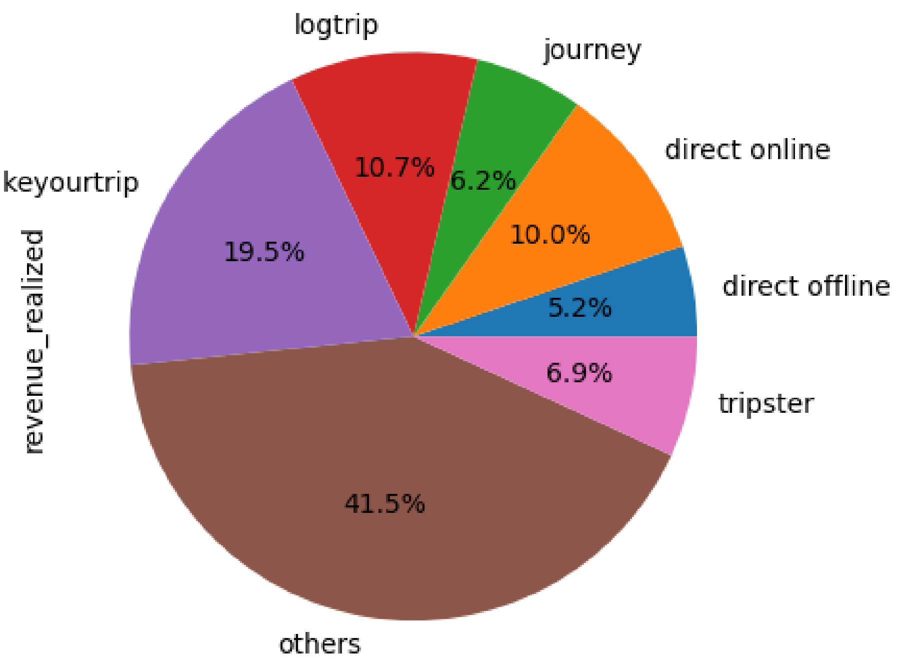
```
[97]: city
Bangalore    3.410464
```

```
Delhi        3.785784
Hyderabad    3.653743
Mumbai       3.629671
Name: ratings_given, dtype: float64
```

Exercise-3 Print a pie chart of revenue realized per booking platform

[98]:
```
df_bookings_all.groupby("booking_platform")["revenue_realized"].sum().
 ↪plot(kind="pie", autopct='%1.1f%%')
```

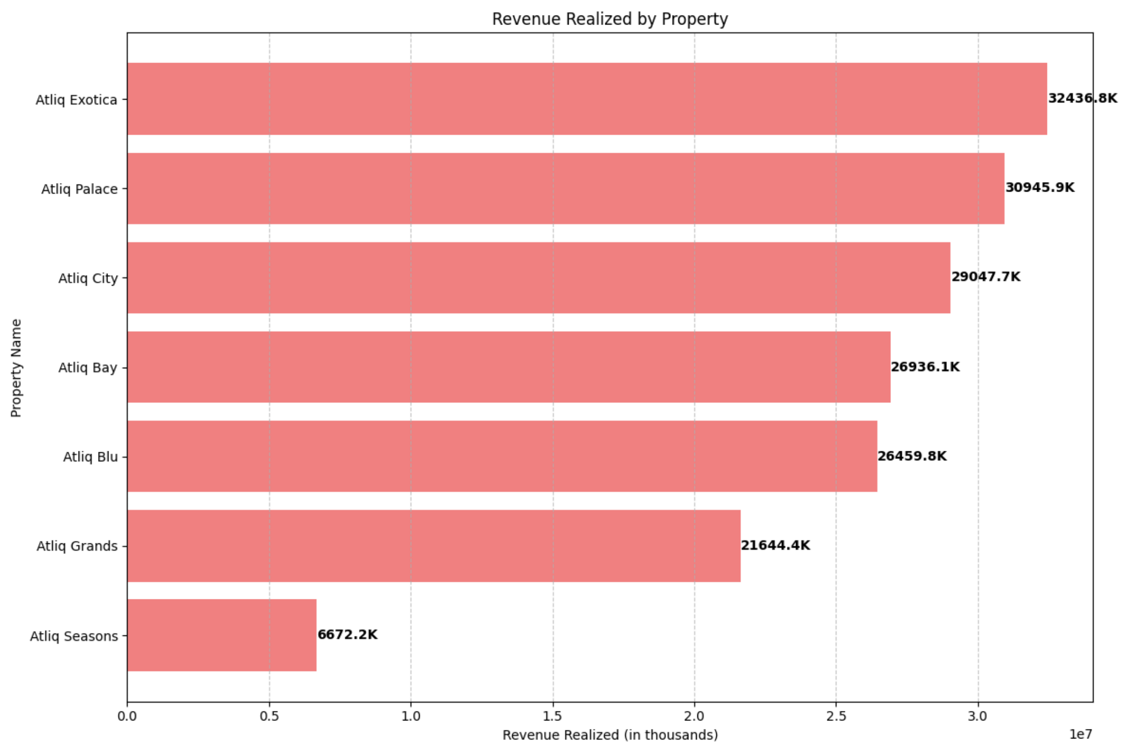
[98]:
```
<Axes: ylabel='revenue_realized'>
```



[106]:
```
df_bookings_all.groupby("booking_platform")["revenue_realized"].sum()
```

[106]:
```
booking_platform
direct offline     8986465
direct online     17488976
journey           10757858
logtrip           18605339
makeyourtrip      34034257
others            72310965
tripster          11959078
Name: revenue_realized, dtype: int64
```

[105]:
```python
import pandas as pd
import matplotlib.pyplot as plt

# Sample data
data = {
    'property_name': ['Atliq Bay', 'Atliq Blu', 'Atliq City', 'Atliq Exotica',␣
 ↪'Atliq Grands', 'Atliq Palace', 'Atliq Seasons'],
    'revenue_realized': [26936115, 26459751, 29047727, 32436799, 21644446,␣
 ↪30945855, 6672245]
}
df = pd.DataFrame(data)

# Sort values
df = df.sort_values(by='revenue_realized', ascending=False)

# Create the bar plot
plt.figure(figsize=(12, 8))
bars = plt.barh(df['property_name'], df['revenue_realized'],␣
 ↪color='lightcoral')  # Change color here

# Add data labels in thousands
for bar in bars:
    width = bar.get_width()
    plt.text(width, bar.get_y() + bar.get_height()/2, f'{width/1000:.1f}K',
             ha='left', va='center', color='black', fontweight='bold')

# Customize the plot
plt.xlabel('Revenue Realized (in thousands)')
plt.ylabel('Property Name')
plt.title('Revenue Realized by Property')
plt.gca().invert_yaxis()
plt.grid(True, axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()

# Display the plot
plt.show()
```


Revenue Realized by Property
Atliq Exotica
Atliq Palace
Atliq City
Atliq Bay
Atliq Blu
Atliq Grands
Atliq Seasons
32436.8K
30945.9K
29047.7K
26936.1K
26459.8K
21644.4K
6672.2K
Property Name
Revenue Realized (in thousands)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e7

```
[ ]:
```