

Hotel Data Analysis

Importing Libraries

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Data Exploration

```
In [3]: df_bookings = pd.read_csv("fact_bookings.csv")
df_agg_bookings = pd.read_csv("fact_aggregated_bookings.csv")
df_hotels = pd.read_csv("dim_hotels.csv")
df_rooms = pd.read_csv("dim_rooms.csv")
df_date = pd.read_csv("dim_date.csv")
```

```
In [6]: df_bookings.shape
```

```
Out[6]: (134590, 12)
```

```
In [7]: df_agg_bookings.shape
```

```
Out[7]: (9200, 5)
```

```
In [8]: df_hotels.shape
```

```
Out[8]: (25, 4)
```

```
In [9]: df_rooms.shape
```

```
Out[9]: (4, 2)
```

```
In [10]: df_date.shape
```

```
Out[10]: (92, 4)
```

```
In [11]: df_bookings.head(4)
```

```
Out[11]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booki
0	May012216558RT11	16558	27-04-22	1/5/2022	2/5/2022	-3.0	RT1	
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	
3	May012216558RT14	16558	28-04-22	1/5/2022	2/5/2022	-2.0	RT1	

```
In [12]: df_agg_bookings.head()
```

```
Out[12]:
```

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

In [13]: `df_rooms.head(3)`

Out[13]:

	room_id	room_class
0	RT1	Standard
1	RT2	Elite
2	RT3	Premium

In [14]: `df_rooms.room_class.unique()`

Out[14]: `array(['Standard', 'Elite', 'Premium', 'Presidential'], dtype=object)`

In [15]: `df_bookings.room_category.unique()`

Out[15]: `array(['RT1', 'RT2', 'RT3', 'RT4'], dtype=object)`

In [16]: `df_bookings.booking_platform.value_counts()`

Out[16]:

others	55066
makeyourtrip	26898
logtrip	14756
direct online	13379
tripster	9630
journey	8106
direct offline	6755

Name: booking_platform, dtype: int64

In [17]: `df_rooms.room_class.value_counts()`

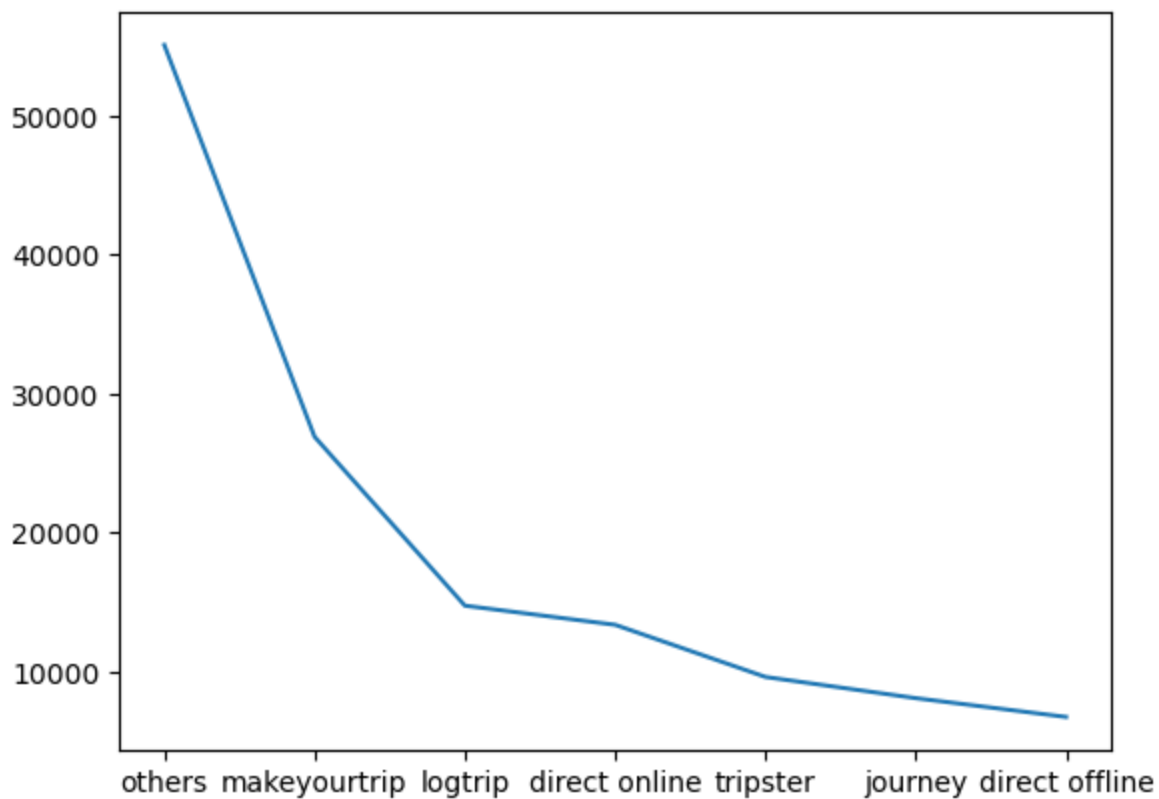
Out[17]:

Standard	1
Elite	1
Premium	1
Presidential	1

Name: room_class, dtype: int64

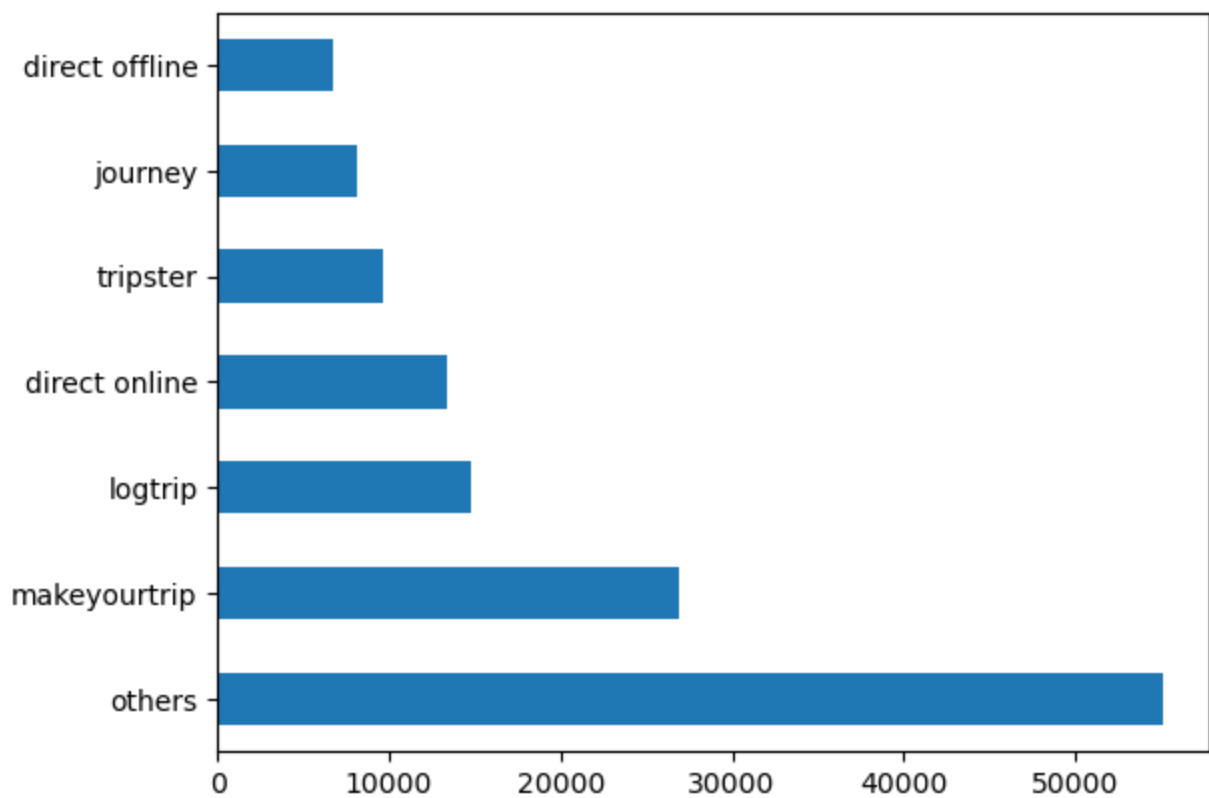
In [19]: `df_bookings.booking_platform.value_counts().plot()`

Out[19]: `<Axes: >`



```
In [22]: df_bookings.booking_platform.value_counts().plot(kind = "barh")
```

Out[22]: <Axes: >



```
In [23]: df_bookings.describe()
```

```
Out[23]:
```

	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 [24]: `df_agg_bookings.describe()`

Out[24]:

	property_id	successful_bookings	capacity
count	9200.000000	9200.000000	9198.000000
mean	18040.640000	14.655761	25.280496
std	1099.818325	7.736170	11.442080
min	16558.000000	1.000000	3.000000
25%	17558.000000	9.000000	18.000000
50%	17564.000000	14.000000	25.000000
75%	18563.000000	19.000000	34.000000
max	19563.000000	123.000000	50.000000

In [25]: `df_hotels.describe()`

Out[25]:

	property_id
count	25.000000
mean	18040.640000
std	1122.436371
min	16558.000000
25%	17558.000000
50%	17564.000000
75%	18563.000000
max	19563.000000

In [27]: `df_bookings.revenue_generated.min(), df_bookings.revenue_generated.max()`

Out[27]: (6500, 28560000)

Data Cleaning

In [28]: `df_bookings.describe()`

Out[28]:

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

Out[29]:

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

In [30]: `df_bookings.shape`

Out[30]: (134590, 12)

In [31]: `df_bookings=df_bookings[df_bookings.no_guests>0]`

In [32]: `df_bookings.shape`

Out[32]: (134578, 12)

In [33]: `df_bookings.revenue_generated.min(), df_bookings.revenue_generated.max(),`

Out[33]: (6500, 28560000)

In [34]: `avg , std = df_bookings.revenue_generated.mean(), df_bookings.revenue_generated.std(),`

In [36]: `avg , std`

Out[36]: (15378.036937686695, 93040.15493143328)

In [38]: `higher_limit=avg + 3*std
higher_limit`

Out[38]: 294498.50173198653

In [39]: `lower_limit= avg - 3*std`

```
In [40]: lower_limit
Out[40]: -263742.4278566132

In [41]: df_bookings[df_bookings.revenue_generated<0]
Out[41]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platfc
--	------------	-------------	--------------	---------------	---------------	-----------	---------------	----------------

```

In [42]: df_bookings[df_bookings.revenue_generated>=0]
Out[42]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1
...
134584	Jul312217564RT45	17564	30-07-22	31-07-22	1/8/2022	2.0	RT4
134585	Jul312217564RT46	17564	29-07-22	31-07-22	3/8/2022	1.0	RT4
134587	Jul312217564RT48	17564	30-07-22	31-07-22	2/8/2022	1.0	RT4
134588	Jul312217564RT49	17564	29-07-22	31-07-22	1/8/2022	2.0	RT4
134589	Jul312217564RT410	17564	31-07-22	31-07-22	1/8/2022	2.0	RT4

134578 rows × 12 columns

```

In [44]: df_bookings[df_bookings.revenue_generated>higher_limit]
Out[44]:
```

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

```

In [45]: df_bookings = df_bookings[df_bookings.revenue_generated<higher_limit]
df_bookings.shape
Out[45]: (134573, 12)

In [46]: df_bookings.revenue_realized.describe()
Out[46]:
```

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

```
Out[48]: 33479.3586618449
```

Data Transformation

```
In [49]: df_agg_bookings.head()
```

```
Out[49]:
```

	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

```
In [55]: df_agg_bookings["occ_pct"] = df_agg_bookings["occ_pct"].apply(lambda x: round(x*100,2))
```

```
In [56]: df_agg_bookings.head()
```

```
Out[56]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct
0	16559	1-May-22	RT1	25	30.0	83.0
1	19562	1-May-22	RT1	28	30.0	93.0
2	19563	1-May-22	RT1	23	30.0	77.0
3	17558	1-May-22	RT1	30	19.0	158.0
4	16558	1-May-22	RT1	18	19.0	95.0

Insights Generation

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

```
In [58]: df_agg_bookings.groupby("room_category")["occ_pct"].mean().round(2)
```

```
Out[58]: room_category  
RT1      58.24  
RT2      58.04  
RT3      58.01  
RT4      59.30  
Name: occ_pct, dtype: float64
```

```
In [67]: df_rooms
```

```
Out[67]:
```

	room_id	room_class
--	---------	------------

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

```
In [68]: df = pd.merge(df_agg_bookings,df_rooms,left_on="room_category" , right_on="room_id")
df.head()
```

```
Out[68]:
```

	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	83.0	RT1	Standard
1	19562	1-May-22	RT1	28	30.0	93.0	RT1	Standard
2	19563	1-May-22	RT1	23	30.0	77.0	RT1	Standard
3	17558	1-May-22	RT1	30	19.0	158.0	RT1	Standard
4	16558	1-May-22	RT1	18	19.0	95.0	RT1	Standard

```
In [69]: df.groupby("room_class")["occ pct"].mean().round(2)
```

```
Out[69]: room_class
Elite      58.04
Premium    58.01
Presidential 59.30
Standard   58.24
Name: occ pct, dtype: float64
```

```
In [70]: df.drop("room_class",axis=1,inplace=True)
df.head(4)
```

```
Out[70]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ pct	room_id
0	16559	1-May-22	RT1	25	30.0	83.0	RT1
1	19562	1-May-22	RT1	28	30.0	93.0	RT1
2	19563	1-May-22	RT1	23	30.0	77.0	RT1
3	17558	1-May-22	RT1	30	19.0	158.0	RT1

2. Print average occupancy rate per city

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

```
Out[59]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ pct
0	16559	1-May-22	RT1	25	30.0	83.0
1	19562	1-May-22	RT1	28	30.0	93.0
2	19563	1-May-22	RT1	23	30.0	77.0
3	17558	1-May-22	RT1	30	19.0	158.0
4	16558	1-May-22	RT1	18	19.0	95.0

```
In [61]: df_hotels.head()
```


Out[61]:

	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

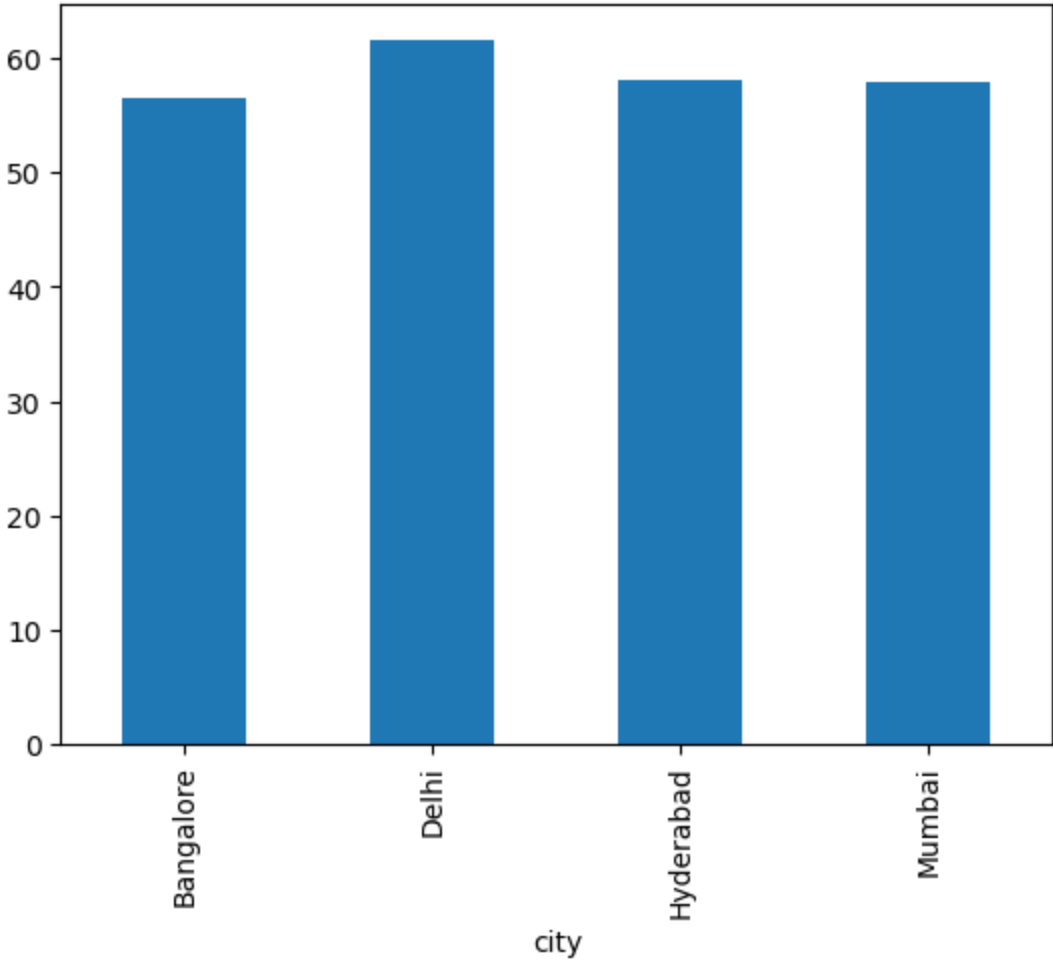
```
In [73]: df = pd.merge(df,df_hotels, on="property_id")
df.head(3)
```

Out[73]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ pct	room_id	property_name_x	ca
0	16559	1-May-22	RT1	25	30.0	83.0	RT1	Atliq Exotica	
1	16559	2-May-22	RT1	20	30.0	67.0	RT1	Atliq Exotica	
2	16559	3-May-22	RT1	17	30.0	57.0	RT1	Atliq Exotica	

```
In [75]: df.groupby("city")["occ pct"].mean().round(2).plot(kind="bar")
```

Out[75]: <Axes: xlabel='city'>



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

```
In [76]: df_date
```

Out[76]:

	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
...
87	27-Jul-22	Jul 22	W 31	weekeday
88	28-Jul-22	Jul 22	W 31	weekeday
89	29-Jul-22	Jul 22	W 31	weekeday
90	30-Jul-22	Jul 22	W 31	weekend
91	31-Jul-22	Jul 22	W 32	weekend

92 rows × 4 columns

```
In [77]: df.head(3)
```

Out[77]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ pct	room_id	property_name_x	ca
0	16559	1-May-22	RT1	25	30.0	83.0	RT1	Atliq Exotica	
1	16559	2-May-22	RT1	20	30.0	67.0	RT1	Atliq Exotica	
2	16559	3-May-22	RT1	17	30.0	57.0	RT1	Atliq Exotica	

```
In [78]: df = pd.merge(df,df_date,left_on="check_in_date" , right_on="date")
df.head()
```

Out[78]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ pct	room_id	property_name_x	ca
0	16559	10-May-22	RT1	18	30.0	60.0	RT1	Atliq Exotica	
1	16559	10-May-22	RT2	25	41.0	61.0	RT2	Atliq Exotica	
2	16559	10-May-22	RT3	20	32.0	62.0	RT3	Atliq Exotica	
3	16559	10-May-22	RT4	13	18.0	72.0	RT4	Atliq Exotica	
4	19562	10-May-22	RT1	18	30.0	60.0	RT1	Atliq Bay	

```
In [81]: df.groupby("day_type")["occ_pct"].mean().round(2)
```

```
Out[81]: day_type
weekday    50.90
weekend     72.41
Name: occ_pct, dtype: float64
```

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

```
In [83]: df["mmm yy"].unique()
```

```
Out[83]: array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

```
In [84]: df_june_22 = df[df["mmm yy"]=="Jun 22"]
df_june_22.head(3)
```

```
Out[84]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_id	property_name_x
2200	16559	10-Jun-22	RT1	20	30.0	67.0	RT1	Atliq Exotica
2201	16559	10-Jun-22	RT2	26	41.0	63.0	RT2	Atliq Exotica
2202	16559	10-Jun-22	RT3	20	32.0	62.0	RT3	Atliq Exotica

```
In [86]: df_june_22.groupby("city")["occ_pct"].mean().round(2)
```

```
Out[86]: city
Bangalore    56.58
Delhi         62.49
Hyderabad     58.47
Mumbai        58.39
Name: occ_pct, dtype: float64
```

5. Add the august data to the existing records

```
In [88]: df_august = pd.read_csv("new_data_august.csv")
df_august.head(3)
```

```
Out[88]:
```

	property_id	property_name	category	city	room_category	room_class	check_in_date	mmm yy	week no	da
0	16559	Atliq Exotica	Luxury	Mumbai	RT1	Standard	01-Aug-22	Aug-22	W 32	we
1	19562	Atliq Bay	Luxury	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	we
2	19563	Atliq Palace	Business	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	we

```
In [89]: latest_df = pd.concat([df, df_august], ignore_index=True, axis=0)
latest_df.tail(10)
```

Out[89]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ pct	room_id	property_name_x
6497	18560	31-Jul-22	RT2	34	40.0	85.0	RT2	Atliq City
6498	18560	31-Jul-22	RT3	17	24.0	71.0	RT3	Atliq City
6499	18560	31-Jul-22	RT4	12	15.0	80.0	RT4	Atliq City
6500	16559	01-Aug-22	RT1	30	30.0	NaN	NaN	NaN
6501	19562	01-Aug-22	RT1	21	30.0	NaN	NaN	NaN
6502	19563	01-Aug-22	RT1	23	30.0	NaN	NaN	NaN
6503	19558	01-Aug-22	RT1	30	40.0	NaN	NaN	NaN
6504	19560	01-Aug-22	RT1	20	26.0	NaN	NaN	NaN
6505	17561	01-Aug-22	RT1	18	26.0	NaN	NaN	NaN
6506	17564	01-Aug-22	RT1	10	16.0	NaN	NaN	NaN

10 rows × 22 columns

6. Print revenue realized per city

In [90]:

df_bookings.head(4)

Out[90]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booki
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1	

In [91]:

df_hotels.head(3)

Out[91]:

	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 [92]:

df_bookings_all=pd.merge(df_bookings,df_hotels,on="property_id")

```
df_bookings_all.head(3)
```

```
Out[92]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booki
0	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	
1	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	
2	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	

```
In [94]: df_bookings_all.groupby("city")["revenue_realized"].sum()
```

```
Out[94]:
```

city	
Bangalore	420383550
Delhi	294404488
Hyderabad	325179310
Mumbai	668569251

Name: revenue_realized, dtype: int64

7. Print month by month revenue

```
In [95]: df_bookings_all.head(3)
```

```
Out[95]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booki
0	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	
1	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	
2	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	

```
In [96]: df_date["mmm yy"].unique()
```

```
Out[96]: array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

```
In [97]: df_date.head(3)
```

```
Out[97]:
```

	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 [99]: pd.merge(df_bookings_all, df_date, left_on="check_in_date", right_on="date")
```

```
Out[99]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platfc
--	------------	-------------	--------------	---------------	---------------	-----------	---------------	----------------

```
In [100]: df_date["date"] = pd.to_datetime(df_date["date"])  
df_date.head(3)
```

```
Out[100]:
```

	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

```
In [101... 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
```

```
In [104... df_bookings_all["check_in_date"] = pd.to_datetime(df_bookings_all["check_in_date"])
df_date.head(3)
```

Out[104]:

	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

```
In [105... df_bookings_all.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 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 datetime64[ns]
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: datetime64[ns](1), float64(2), int64(3), object(9)
memory usage: 16.4+ MB
```

```
In [112... df_bookings_all = pd.merge(df_bookings_all,df_date,left_on="check_in_date",right_on="date")
df_bookings_all.head(3)
```

Out[112]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booki
0	May052216558RT11	16558	15-04-22	2022-05-05	7/5/2022	3.0	RT1	
1	May052216558RT12	16558	30-04-22	2022-05-05	7/5/2022	2.0	RT1	
2	May052216558RT13	16558	1/5/2022	2022-05-05	6/5/2022	3.0	RT1	

3 rows × 27 columns

```
In [113... df_bookings_all.groupby("mmm yy")["revenue_realized"].sum()
```

```
Out[113]: mmm yy  
Jul 22    389940912  
Jun 22    377191229  
May 22    408375641  
Name: revenue_realized, dtype: int64
```