AtliQ Hotels Data Analysis Project

In [782…
```
import pandas as pd
import matplotlib.pyplot as plt
```

---

1. Data Import and Data Exploration

Datasets We have 5 CSV files:

1. `dim_date.csv`
2. `dim_hotels.csv`
3. `dim_rooms.csv`
4. `fact_aggregated_bookings.csv`
5. `fact_bookings.csv`

Read bookings data in a datagrame

In [784…
```
df_bookings = pd.read_csv("datasets/fact_bookings.csv")
df_bookings.head(4)
```

Out[784…

	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

Explore bookings data

In [860…
```
df_bookings.head()
```

Out[860…

	booking_id	property_id	booking_date	check_in_date	checkout_date
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022
7	May012216558RT18	16558	26-04-22	1/5/2022	3/5/2022

In [785…
```
df_bookings.shape
```

Out[785…
```
(134590, 12)
```

In [786…
```
df_bookings.room_category.unique()
```

Out[786…
```
array(['RT1', 'RT2', 'RT3', 'RT4'], dtype=object)
```

In [787…
```
df_bookings.booking_platform.unique()
```

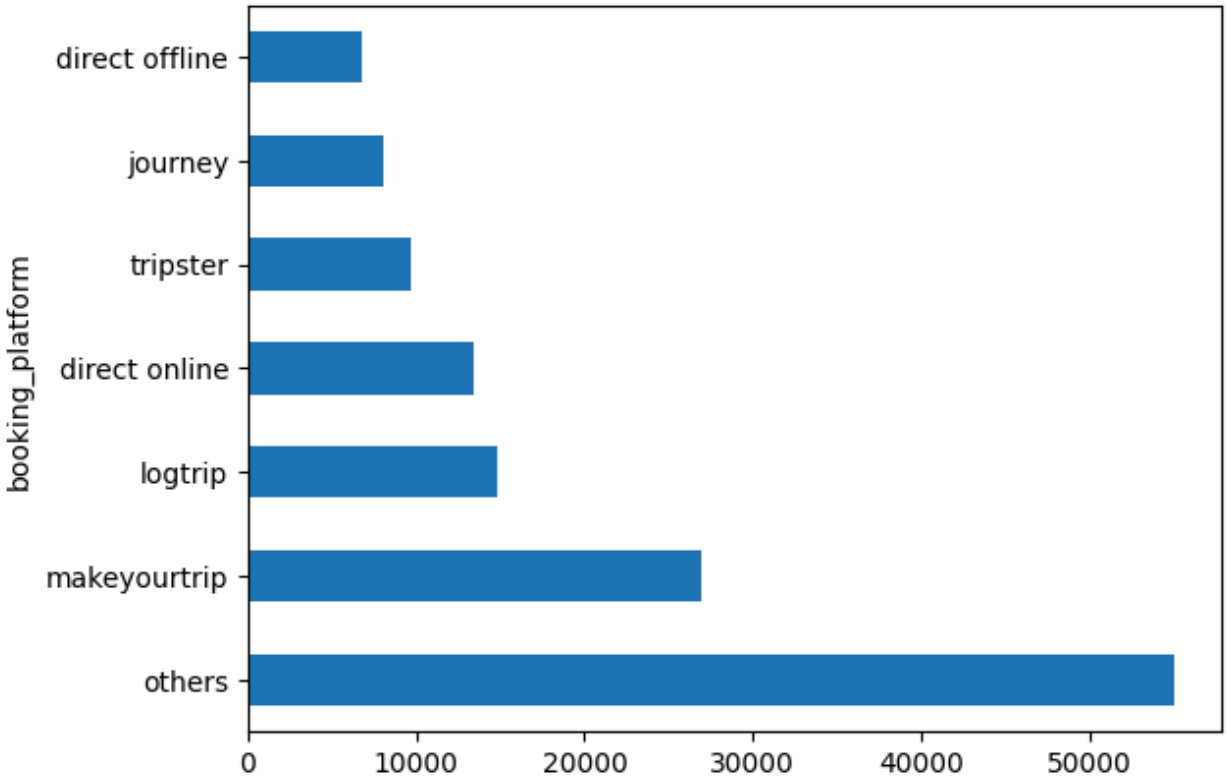
Out[787…
```
array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',
       'journey', 'direct offline'], dtype=object)
```

In [788…
```
df_bookings.booking_platform.value_counts()
```

Out[788…
```
booking_platform
others            55066
makeyourtrip      26898
logtrip           14756
direct online     13379
tripster           9630
journey            8106
direct offline     6755
Name: count, dtype: int64
```

In [789…
```
df_bookings.booking_platform.value_counts().plot(kind="barh")
""" another way can be:
    import matplotlib.pyplot as plt
    platform_counts = df_bookings.booking_platform.value_counts()
    plt.barh(platform_counts.index, platform_counts.values)
    plt.xlabel("Number of Bookings")
    plt.ylabel("Booking Platform")
    plt.title("Bookings per Platform")
    plt.show()"""
```

Out[789…
```
' another way can be:\n    import matplotlib.pyplot as plt\n    platform_coun
ts = df_bookings.booking_platform.value_counts()\n    plt.barh(platform_count
s.index, platform_counts.values)\n    plt.xlabel("Number of Bookings")\n    p
lt.ylabel("Booking Platform")\n    plt.title("Bookings per Platform")\n    pl
t.show()'
```



In [790…
```
df_bookings.describe()
```

Out[790…

	property_id	no_guests	ratings_given	revenue_generated	revenu
count	134590.000000	134587.000000	56683.000000	1.345900e+05	1345
mean	18061.113493	2.036170	3.619004	1.537805e+04	126
std	1093.055847	1.034885	1.235009	9.303604e+04	69
min	16558.000000	-17.000000	1.000000	6.500000e+03	26
25%	17558.000000	1.000000	3.000000	9.900000e+03	76
50%	17564.000000	2.000000	4.000000	1.350000e+04	117
75%	18563.000000	2.000000	5.000000	1.800000e+04	153
max	19563.000000	6.000000	5.000000	2.856000e+07	452

In [791…
```
df_bookings.revenue_generated.min(),df_bookings.revenue_generated.max()
```

Out[791…
```
(np.int64(6500), np.int64(28560000))
```

In [792…
```
import pandas as pd
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 [793…
```
df_hotels.shape
```

Out[793…
```
(25, 4)
```

In [794…
```
df_hotels.head(4)
```

Out[794…

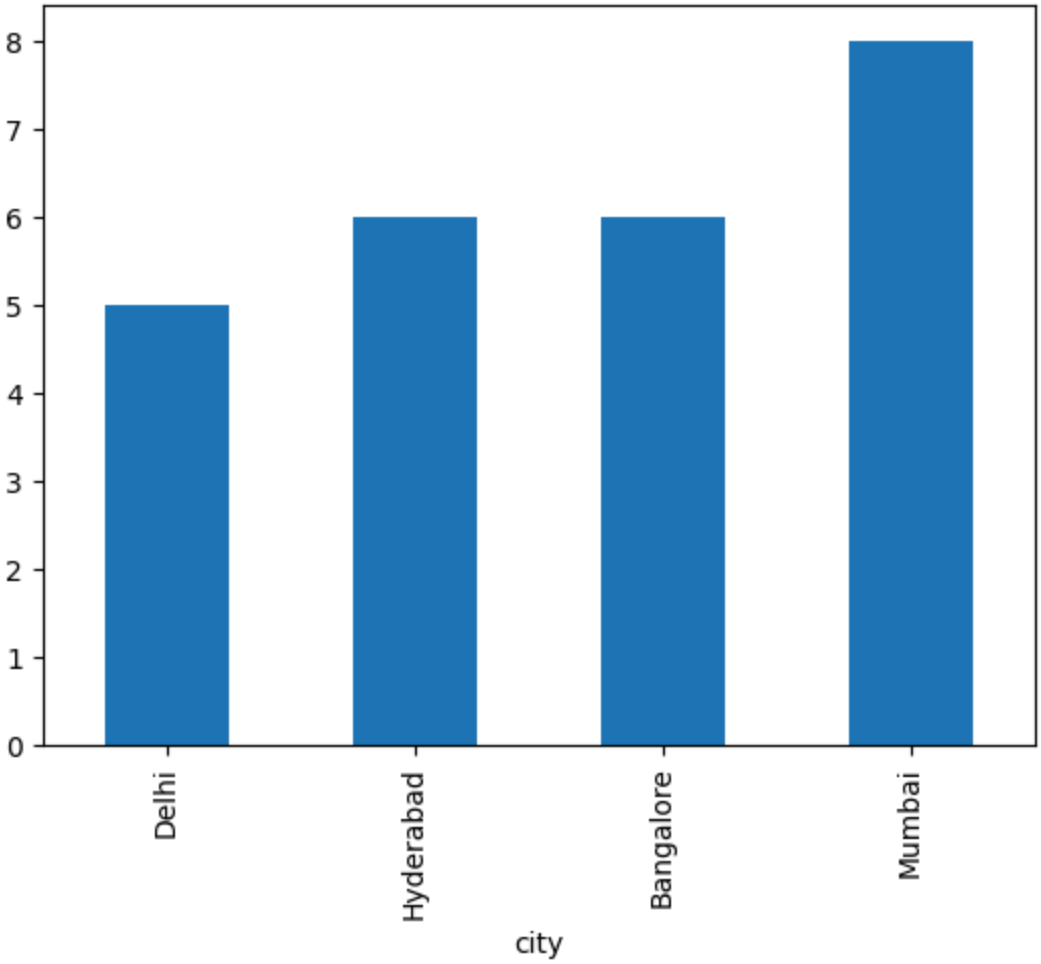
	property_id	property_name	category	city
0	16558	Atliq Grands	Luxury	Delhi
1	16559	Atliq Exotica	Luxury	Mumbai
2	16560	Atliq City	Business	Delhi
3	16561	Atliq Blu	Luxury	Delhi

In [795…
```
df_hotels.category.value_counts()
```

Out[795…
```
category
Luxury      16
Business     9
Name: count, dtype: int64
```

In [796…
```
df_hotels.city.value_counts().sort_values().plot(kind="bar")
```

Out[796…
```
<Axes: xlabel='city'>
```



2. Data Cleaning

In [797…

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

Out[797…

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

In [798…
```
df_bookings.shape
```

Out[798…
```
(134590, 12)
```

In [799…
```
df_bookings = df_bookings[df_bookings.no_guests>0]
df_bookings
```

Out[799…

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

134578 rows × 12 columns

In [800…
```
df_bookings.shape
```

Out[800…
```
(134578, 12)
```

In [801…
```
df_bookings.revenue_generated.min(),df_bookings.revenue_generated.max()
```

Out[801…
```
(np.int64(6500), np.int64(28560000))
```

In [802…
```
avg, std = df_bookings.revenue_generated.mean(), df_bookings.revenue_generated
```

In [803…
```
avg, std
```

Out[803…
```
(np.float64(15378.036937686695), np.float64(93040.1549314641))
```

In [804…
```
higher_limit = avg + 3*std
higher_limit
```

Out[804…
```
np.float64(294498.50173207896)
```

In [805…
```
lower_limit = avg - 3*std
lower_limit
```

Out[805…
```
np.float64(-263742.4278567056)
```

In [806…
```
df_bookings[df_bookings.revenue_generated>higher_limit]
```

Out[806…

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

In [807…
```
df_bookings = df_bookings[df_bookings.revenue_generated<higher_limit]
df_bookings.shape
```

Out[807…
```
(134573, 12)
```

In [808…
```
df_bookings.revenue_realized.describe()
```

Out[808…
```
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 [809…
```
higher_limit = df_bookings.revenue_realized.mean() + 3*df_bookings.revenue_rea
```

```
higher_limit
```

Out[809…
```
np.float64(33479.358661845814)
```

In [810…
```
df_bookings[df_bookings.revenue_realized>higher_limit]
```

Out[810…

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

1299 rows × 12 columns

In [811…
```
df_rooms
```

Out[811…

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

In [812…
```
df_bookings[df_bookings.room_category=="RT4"].revenue_realized.describe()
```

Out[812…
```
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 [813…
```
23439 + 3*9048
```

Out[813…
```
50583
```

In [814…
```
df_bookings.isnull().sum()
```

Out[814…
```
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
```

---

3. Data Transformation

In [815…
```
df_agg_bookings.head()
```

Out[815…

	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 [816…
```
df_agg_bookings["Occ_pct"] = df_agg_bookings["successful_bookings"]/df_agg_boo
```

In [817…
```
df_agg_bookings.head()
```

Out[817…

	property_id	check_in_date	room_category	successful_bookings	capacity	O
0	16559	1-May-22	RT1	25	30.0	0.8
1	19562	1-May-22	RT1	28	30.0	0.9
2	19563	1-May-22	RT1	23	30.0	0.7
3	17558	1-May-22	RT1	30	19.0	1.5
4	16558	1-May-22	RT1	18	19.0	0.9

In [818…
```
df_agg_bookings["Occ_pct"] = df_agg_bookings["Occ_pct"].apply(lambda x: round(
df_agg_bookings.head(4)
```

Out[818…

	property_id	check_in_date	room_category	successful_bookings	capacity	Oc
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	1

In [858…
```
df_bookings.head()
```

Out[858…

	booking_id	property_id	booking_date	check_in_date	checkout_date
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022
7	May012216558RT18	16558	26-04-22	1/5/2022	3/5/2022

In [859…
```
df_agg_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9200 entries, 0 to 9199
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   property_id          9200 non-null   int64  
 1   check_in_date        9200 non-null   object 
 2   room_category        9200 non-null   object 
 3   successful_bookings  9200 non-null   int64  
 4   capacity             9198 non-null   float64
 5   Occ_pct              9198 non-null   float64
dtypes: float64(2), int64(2), object(2)
memory usage: 431.4+ KB
```

In [ ]:
```
There are various types of data transformations that you may have to perform b

1. Creating new columns
1. Normalization
1. Merging data
1. Aggregation
```

---

4. Insights Generation

---

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

In [819…
```
df_agg_bookings.groupby("room_category")["Occ_pct"].mean().round(2)
```

Out[819…
```
room_category
RT1    58.22
RT2    58.04
RT3    58.03
RT4    59.30
Name: Occ_pct, dtype: float64
```

In [820…
```
df_rooms
```

Out[820…

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

In [821…
```
df = pd.merge(df_agg_bookings, df_rooms, left_on="room_category", right_on="rc
df.head(4)
```

Out[821…

	property_id	check_in_date	room_category	successful_bookings	capacity	Oc
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	1

In [822…
```
df.groupby("room_class")["Occ_pct"].mean().round(2)
```

Out[822…
```
room_class
Elite           58.04
Premium         58.03
Presidential    59.30
Standard        58.22
Name: Occ_pct, dtype: float64
```

In [823…
```
df.drop("room_id", axis=1, inplace=True)
df.head(4)
```

Out[823…

	property_id	check_in_date	room_category	successful_bookings	capacity	Oc
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	1

2. Print Average Occupancy rate by city.

In [824…

```
df_hotels.head(3)
```

Out[824…

	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 [825…

```
df = pd.merge(df, df_hotels, on="property_id")
df.head(3)
```

Out[825…

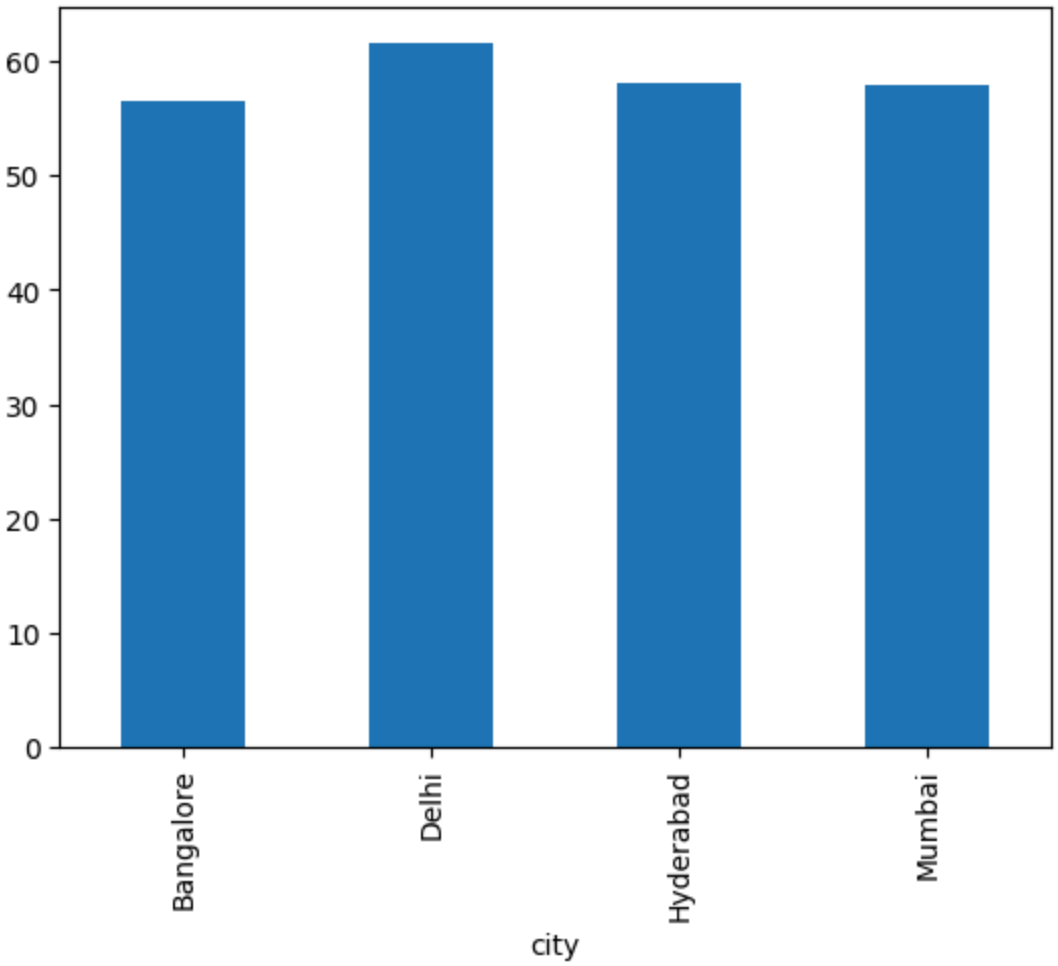
	property_id	check_in_date	room_category	successful_bookings	capacity	Oc
0	16559	1-May-22	RT1	25	30.0	
1	19562	1-May-22	RT1	28	30.0	
2	19563	1-May-22	RT1	23	30.0	

In [826…

```
df.groupby("city")["Occ_pct"].mean().plot(kind="bar")
```

Out[826…

```
<Axes: xlabel='city'>
```



3. when was the occupancy better?

In [827…
```
df.head(3)
```

Out[827…

	property_id	check_in_date	room_category	successful_bookings	capacity	Oc
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 [828…
```
df = pd.merge(df, df_date, left_on="check_in_date", right_on="date")
df.head(3)
```

Out[828…

	property_id	check_in_date	room_category	successful_bookings	capacity	Oc
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	

In [829…
```
df.groupby("day_type")["Occ_pct"].mean().round(2)
```

Out[829…
```
day_type
weekeday    50.90
weekend     72.39
Name: Occ_pct, dtype: float64
```

4. In the month of june, what is the occupany in different cities

In [830…
```
df["mmm yy"].unique()
```

Out[830…
```
array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

In [831…
```
df_june_22 = df[df["mmm yy"]=='Jun 22']
df_june_22.head(3)
```

Out[831…

	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

In [832…
```
df_june_22.groupby("city")["Occ_pct"].mean().round(2).sort_values(ascending=Fa
```

Out[832…
```
city
Delhi        62.47
Hyderabad    58.46
Mumbai       58.38
Bangalore    56.58
Name: Occ_pct, dtype: float64
```

In [833…
```
df_august = pd.read_csv("datasets/new_data_august.csv")
df_august.head(3)
```

Out[833…

	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

In [834…
```
df_august.columns
```

Out[834…
```
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 [835…
```
df.columns
```

Out[835…
```
Index(['property_id', 'check_in_date', 'room_category', 'successful_booking
s',
       'capacity', 'Occ_pct', 'room_class', 'property_name', 'category',
       'city', 'date', 'mmm yy', 'week no', 'day_type'],
      dtype='object')
```

In [836…
```
df_august.shape
```

Out[836…
```
(7, 13)
```

In [837…
```
df.shape
```

Out[837…
```
(6500, 14)
```

In [838…
```
latest_df = pd.concat([df, df_august], ignore_index=True, axis=0)
latest_df.tail(10)
```

Out[838…

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

In [839…
```
latest_df.shape
```

Out[839… (6507, 15)

6. Print revenue realized per city

In [840…
```
df_bookings.head(4)
```

Out[840…

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

In [841…
```
df_hotels.head(3)
```

Out[841…

	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 [842…
```
df_bookings_all = pd.merge(df_bookings, df_hotels, on="property_id")
df_bookings_all.head(3)
```

Out[842…

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

In [843…
```
df_bookings_all.groupby("city")["revenue_realized"].sum()
```

Out[843…
```
city
Bangalore    420383550
Delhi        294404488
Hyderabad    325179310
Mumbai       668569251
Name: revenue_realized, dtype: int64
```

7. Print month by month revenue

In [844…
```
df_date.head(3)
```

Out[844…

	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 [845…
```
df_date["mmm yy"].unique()
```

Out[845…
```
array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

In [846…
```
df_bookings_all.head(3)
```

Out[846…

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

In [847…
```
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 [848…
```
df_bookings_all["check_in_date"] = pd.to_datetime(df_bookings_all["check_in_da
df_bookings_all.head(4)
```

Out[848…

	booking_id	property_id	booking_date	check_in_date	checkout_date
0	May012216558RT12	16558	30-04-22	2022-05-01	2/5/2022
1	May012216558RT15	16558	27-04-22	2022-05-01	2/5/2022
2	May012216558RT16	16558	1/5/2022	2022-05-01	3/5/2022
3	May012216558RT17	16558	28-04-22	2022-05-01	6/5/2022

In [758…
```
df_date["date"] = pd.to_datetime(df_date["date"])
df_date.head(3)
```

Out[758…

	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 [850…
```
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  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: 15.4+ MB
```

In [851…

```
df_bookings_all["check_in_date"] = pd.to_datetime(df_bookings_all["check_in_da
df_bookings_all.head(4)
```

Out[851…

	booking_id	property_id	booking_date	check_in_date	checkout_date
0	May012216558RT12	16558	30-04-22	2022-05-01	2/5/2022
1	May012216558RT15	16558	27-04-22	2022-05-01	2/5/2022
2	May012216558RT16	16558	1/5/2022	2022-05-01	3/5/2022
3	May012216558RT17	16558	28-04-22	2022-05-01	6/5/2022

1. Print revenue realized per hotel type

In [746…

```
df_bookings_all.groupby("property_name")["revenue_realized"].sum().sort_values
```

Out[746…

```
property_name
Atliq Exotica    302316584
Atliq Palace     293666481
Atliq City       278111153
Atliq Blu        255511060
Atliq Bay        253888950
Atliq Grands     206977678
Atliq Seasons     62430375
Name: revenue_realized, dtype: int64
```

2.Print average rating per city

In [734…
```
df_bookings_all.groupby("city")["ratings_given"].mean().round(2).sort_values(a
```

Out[734…
```
city
Delhi        3.78
Hyderabad    3.66
Mumbai       3.65
Bangalore    3.40
Name: ratings_given, dtype: float64
```

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

In [582…
```
revenue_by_platform = df_bookings_all.groupby("booking_platform")["revenue_rea

revenue_by_platform.plot(kind="pie", autopct="%1.1f%%")
plt.ylabel("")   # removes y-axis label
plt.xticks(rotation=0)
plt.show()
```