



AtliQ Grands Hospitality Analysis

Problem Statement

AtliQ Grands, a well known hotel chain with a presence across several Indian cities and brands like AtliQ Seasons, AtliQ Exotica, AtliQ Bay, and AtliQ Palace, has experienced a decline in both revenue and market share. To tackle this issue and strengthen his financial performance, the company plans to harness data analytics for more informed decision-making. The initiative focuses on examining booking records sourced from both the official website and third-party platforms to uncover insights that can drive revenue growth and improve competitiveness in the market.

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
```

Dataset

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

Data Exploration

```
In [2]: df_bookings = pd.read_csv('fact_bookings.csv')
df_bookings.head(4)
```

```
Out[2]:
```

	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

```
In [3]: df_bookings.shape
```

```
Out[3]: (134590, 12)
```

```
In [4]: df_bookings.room_category.unique()
```

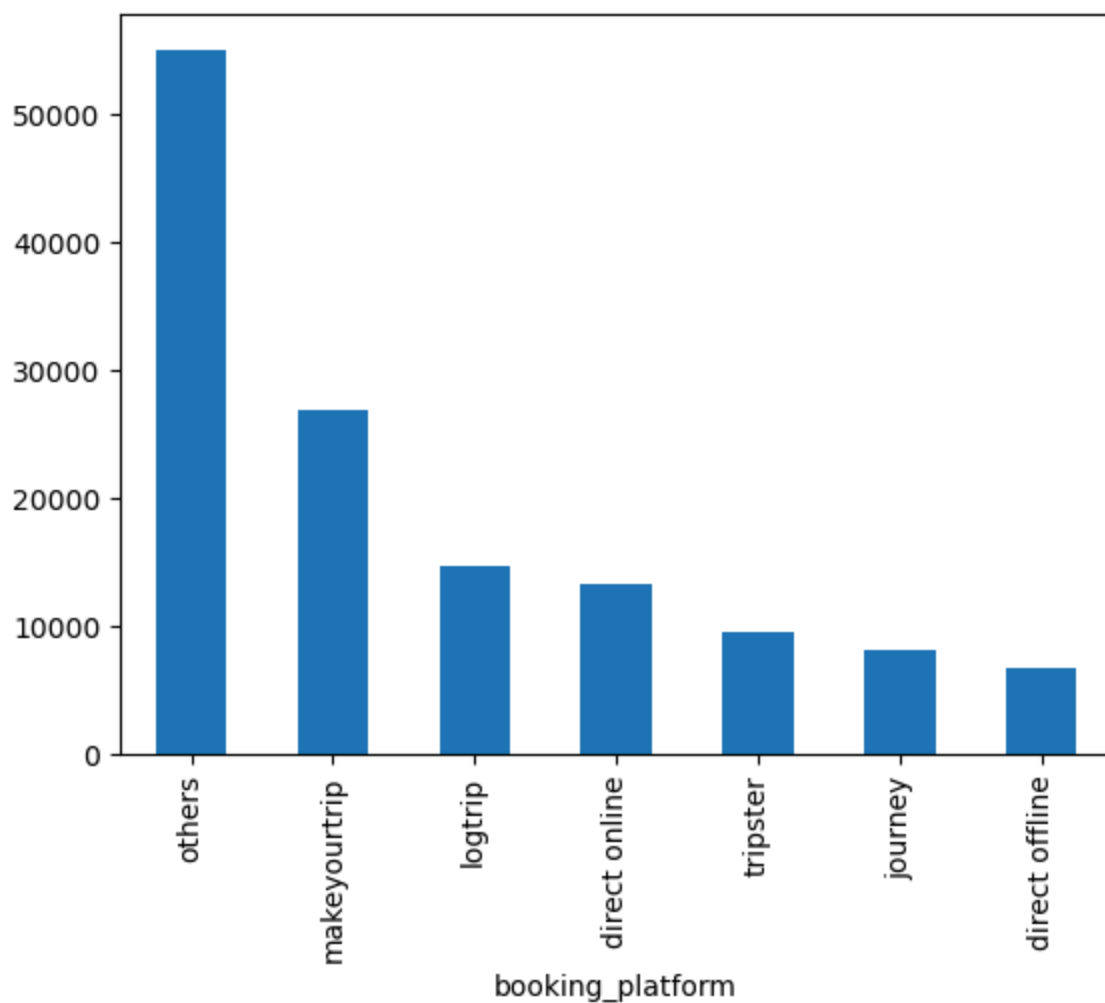
```
Out[4]: array(['RT1', 'RT2', 'RT3', 'RT4'], dtype=object)
```

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

```
Out[5]: array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',  
              'journey', 'direct offline'], dtype=object)
```

```
In [6]: df_bookings.booking_platform.value_counts().plot(kind = 'bar')
```

```
Out[6]: <Axes: xlabel='booking_platform'>
```



```
In [7]: df_bookings.describe()
```

```
Out[7]:
```

	property_id	no_guests	ratings_given	revenue_generated	revenue
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 [8]: df_bookings.revenue_generated.min().item(),df_bookings.revenue_generated.max()
```

```
Out[8]: (6500, 28560000)
```

```
In [9]: 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 [10]: df_hotels.shape
```

```
Out[10]: (25, 4)
```

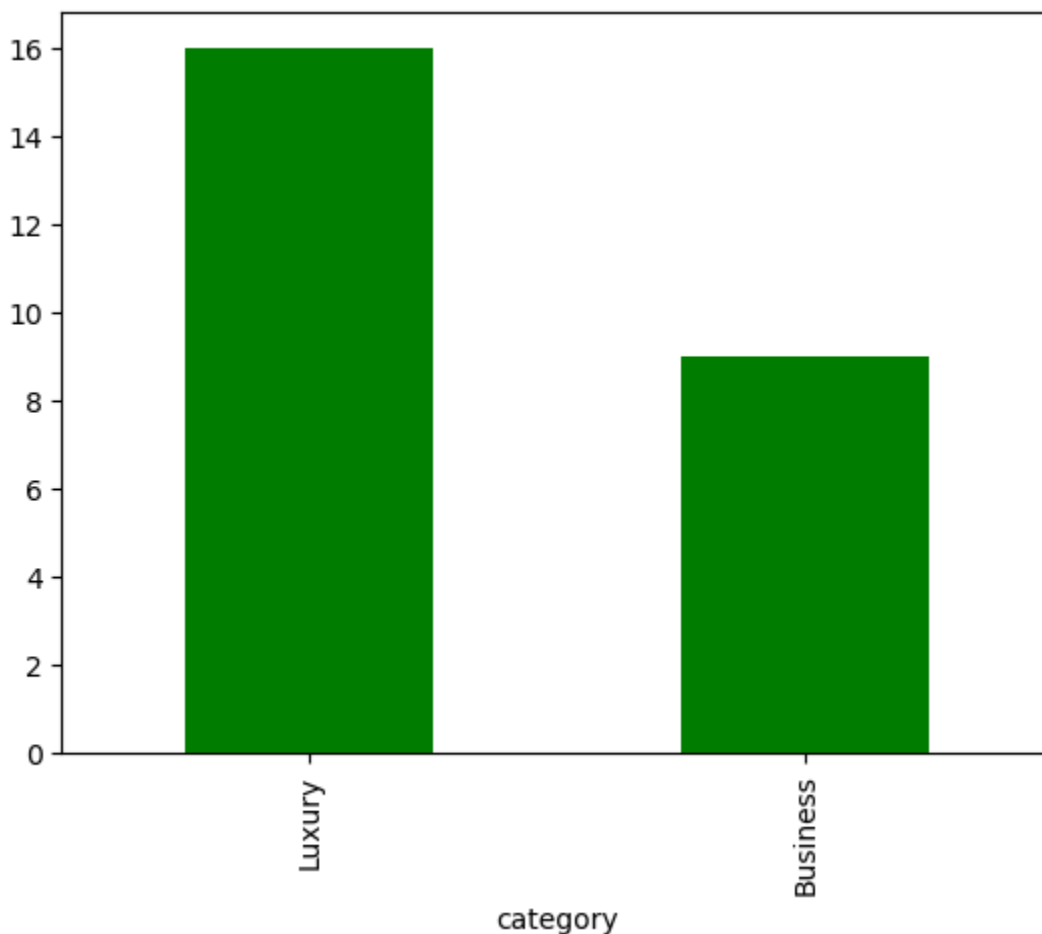
```
In [11]: df_hotels.head(4)
```

```
Out[11]:
```

	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

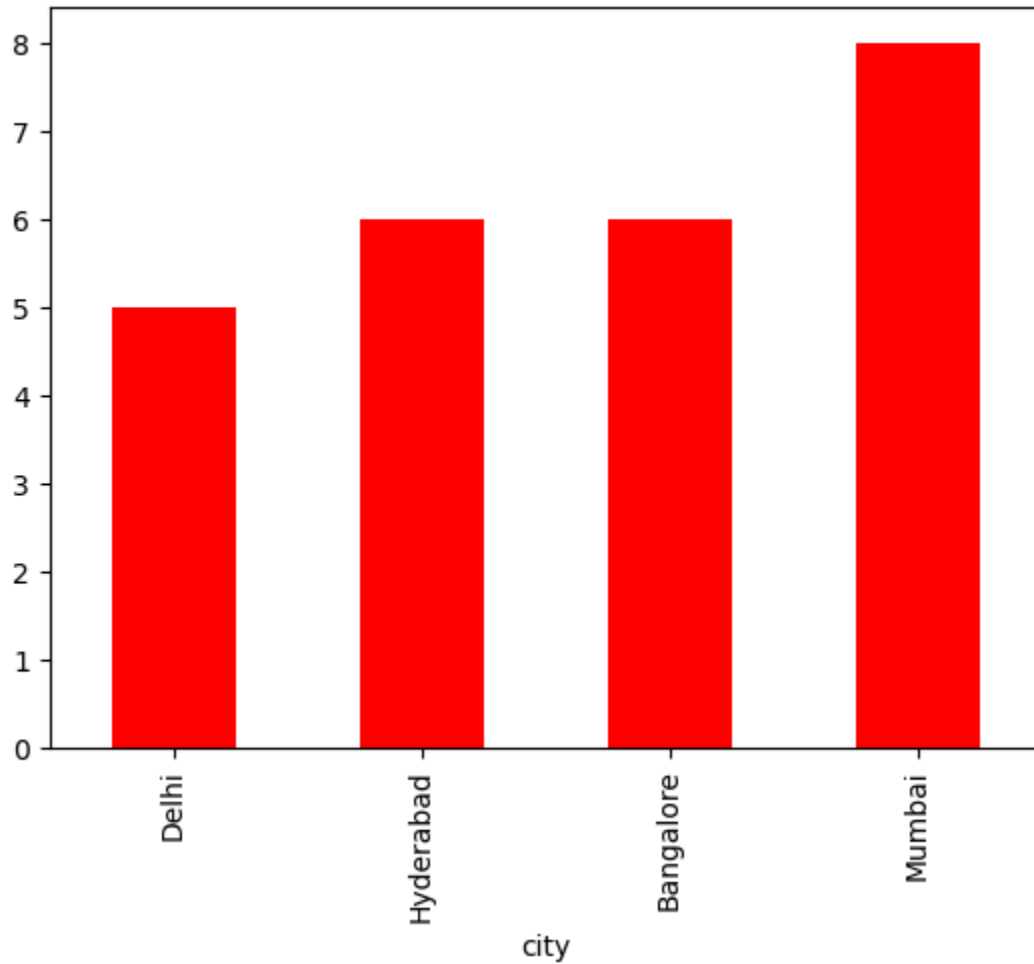
```
In [12]: df_hotels.category.value_counts().plot(kind = 'bar', color = 'green')
```

```
Out[12]: <Axes: xlabel='category'>
```



```
In [13]: df_hotels.city.value_counts().sort_values().plot(kind = 'bar', color = 'red')
```

```
Out[13]: <Axes: xlabel='city'>
```



Data Cleaning

```
In [14]: df_bookings.describe()
```

Out[14]:

	property_id	no_guests	ratings_given	revenue_generated	revenue
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 [15]: `df_bookings[df_bookings.no_guests <= 0]`

Out[15]:

	booking_id	property_id	booking_date	check_in_date	checkout_date
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	14/5/2022
18020	May122218561RT22	18561	8/5/2022	12/5/2022	14/5/2022
18119	May122218562RT311	18562	5/5/2022	12/5/2022	17/5/2022
18121	May122218562RT313	18562	10/5/2022	12/5/2022	17/5/2022
56715	Jun082218562RT12	18562	5/6/2022	8/6/2022	13/6/2022
119765	Jul202219560RT220	19560	19-07-22	20-07-22	22/07/2022
134586	Jul312217564RT47	17564	30-07-22	31-07-22	1/8/2022

In [16]: `df_bookings = df_bookings[df_bookings.no_guests > 0]`
`df_bookings`

```
Out[16]:
```

	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
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022
...
134584	Jul312217564RT45	17564	30-07-22	31-07-22	1/8/2022
134585	Jul312217564RT46	17564	29-07-22	31-07-22	3/8/2022
134587	Jul312217564RT48	17564	30-07-22	31-07-22	2/8/2022
134588	Jul312217564RT49	17564	29-07-22	31-07-22	1/8/2022
134589	Jul312217564RT410	17564	31-07-22	31-07-22	1/8/2022

134578 rows × 12 columns

```
In [17]: df_bookings = df_bookings[df_bookings.no_guests > 0]
df_bookings.shape
```

```
Out[17]: (134578, 12)
```

```
In [18]: df_bookings.revenue_generated.min().item(), df_bookings.revenue_generated.max().item()
```

```
Out[18]: (6500, 28560000)
```

```
In [19]: avg, std = df_bookings.revenue_generated.mean().item(), df_bookings.revenue_generated.std().item()
```

```
In [20]: avg, std
```

```
Out[20]: (15378.036937686695, 93040.1549314641)
```

```
In [21]: higher_limit = avg + 3*std
higher_limit
```

```
Out[21]: 294498.50173207896
```

```
In [22]: lower_limit = avg - 3*std
lower_limit
```

```
Out[22]: -263742.4278567056
```

```
In [23]: df_bookings[df_bookings.revenue_generated > higher_limit]
```

Out[23]:

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

In [24]: `df_bookings = df_bookings[df_bookings.revenue_generated < higher_limit]`
`df_bookings.shape`

Out[24]: (134573, 12)

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

Out[25]:

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

Out[26]: 33479.358661845814

In [27]: `df_bookings[df_bookings.revenue_realized > higher_limit]`


```
Out[27]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date
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 × 6 columns

```
In [28]: df_rooms
```

```
Out[28]:
```

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

```
In [29]: df_bookings[df_bookings.room_category == 'RT4'].revenue_realized.describe()
```

```
Out[29]: 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 [30]: df_bookings.isnull().sum()
```

```
Out[30]: 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
```

Data Transformation

```
In [31]: df_agg_bookings.head()
```

```
Out[31]:
```

	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 [32]: df_agg_bookings['occ_pct'] = df_agg_bookings['successful_bookings']/df_agg_bookings['capacity']
```

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

```
Out[33]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct
0	16559	1-May-22	RT1	25	30.0	0.833333
1	19562	1-May-22	RT1	28	30.0	0.933333
2	19563	1-May-22	RT1	23	30.0	0.766667
3	17558	1-May-22	RT1	30	19.0	1.578947
4	16558	1-May-22	RT1	18	19.0	0.947368

```
In [34]: df_agg_bookings["occ_pct"] = df_agg_bookings["occ_pct"].apply(lambda x: round(x, 2))
df_agg_bookings.head(4)
```

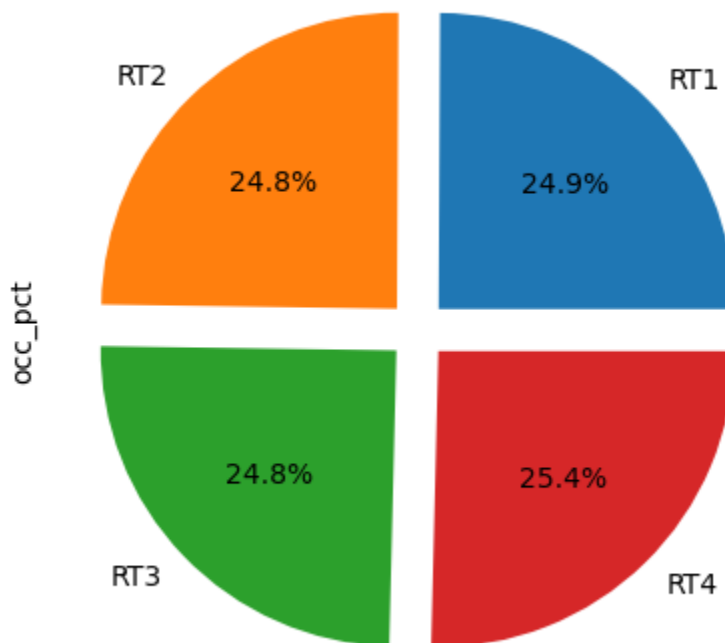
```
Out[34]:
```

	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

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

```
In [35]: df_agg_bookings.groupby('room_category')['occ_pct'].mean().round(2).plot(kind
```

```
Out[35]: <Axes: ylabel='occ_pct'>
```



```
In [36]: df = pd.merge(df_agg_bookings, df_rooms, left_on = 'room_category', right_on = df.head(4))
```

```
Out[36]:
```

	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 [37]: df.groupby('room_class')['occ_pct'].mean().round(2)
```

```
Out[37]: room_class
Elite      58.04
Premium    58.03
Presidential 59.30
Standard   58.22
Name: occ_pct, dtype: float64
```

```
In [38]: df.drop('room_id', axis = 1, inplace = True)
df.head(4)
```

```
Out[38]:
```

	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 per city

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

```
Out[39]:
```

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

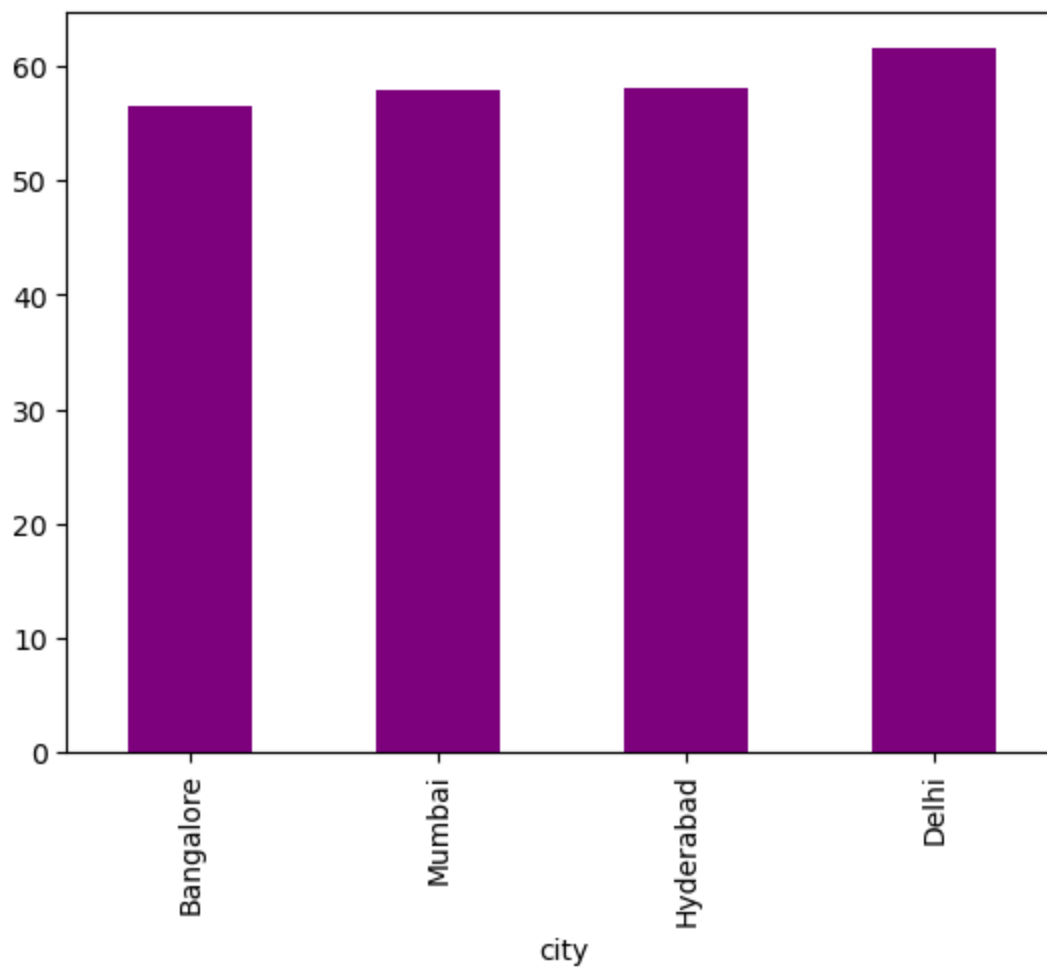
```
In [40]: df = pd.merge(df, df_hotels, on = 'property_id')
df.head(3)
```

```
Out[40]:
```

	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 [41]: df.groupby('city')['occ_pct'].mean().sort_values().plot(kind = 'bar', color =
```

```
Out[41]: <Axes: xlabel='city'>
```



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

```
In [42]: df.head(3)
```

```
Out[42]:
```

	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	

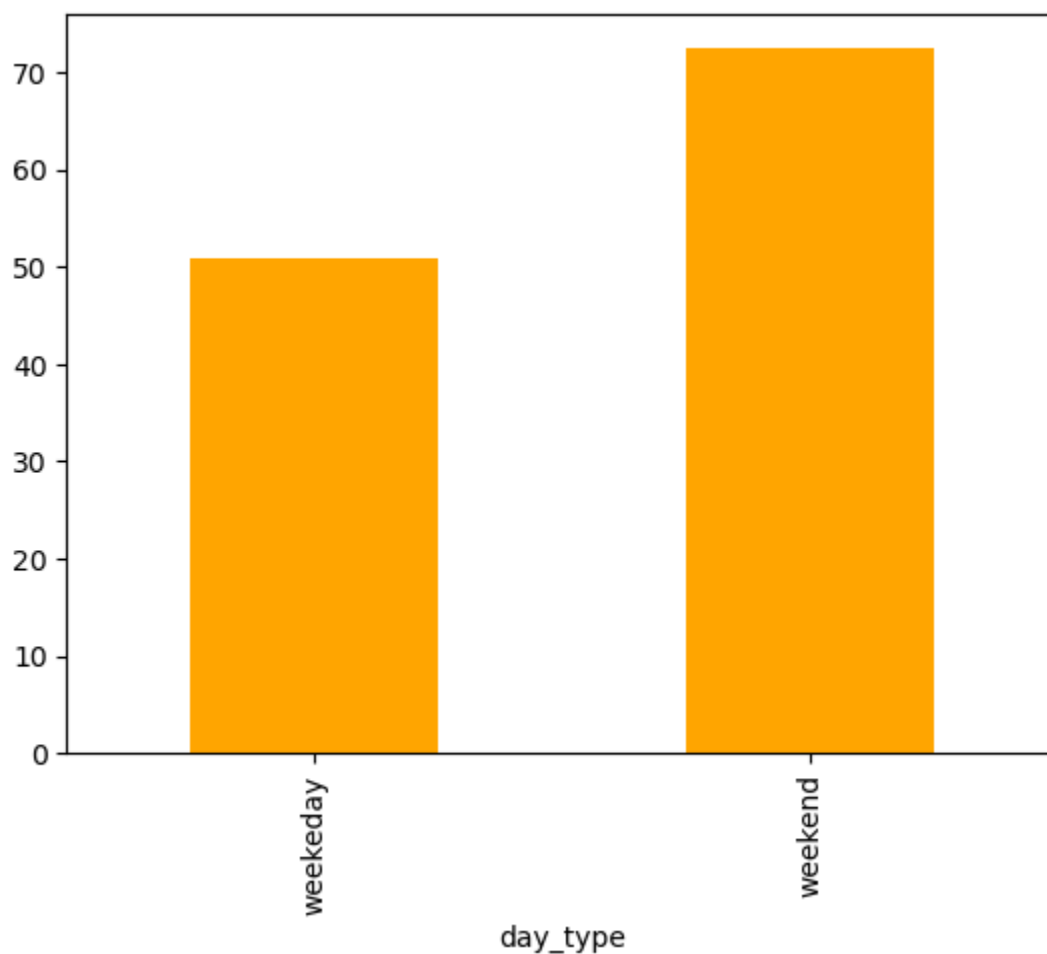
```
In [43]: df = pd.merge(df, df_date, left_on = 'check_in_date', right_on = 'date')
df.head(3)
```

```
Out[43]:
```

	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 [44]: df.groupby('day_type')['occ_pct'].mean().round(2).plot(kind = 'bar', color = 'orange')
```

```
Out[44]: <Axes: xlabel='day_type'>
```



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

```
In [45]: df['mmm yy'].unique()
```

```
Out[45]: array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

```
In [46]: df_june_22 = df[df['mmm yy'] == 'Jun 22']
df_june_22
```

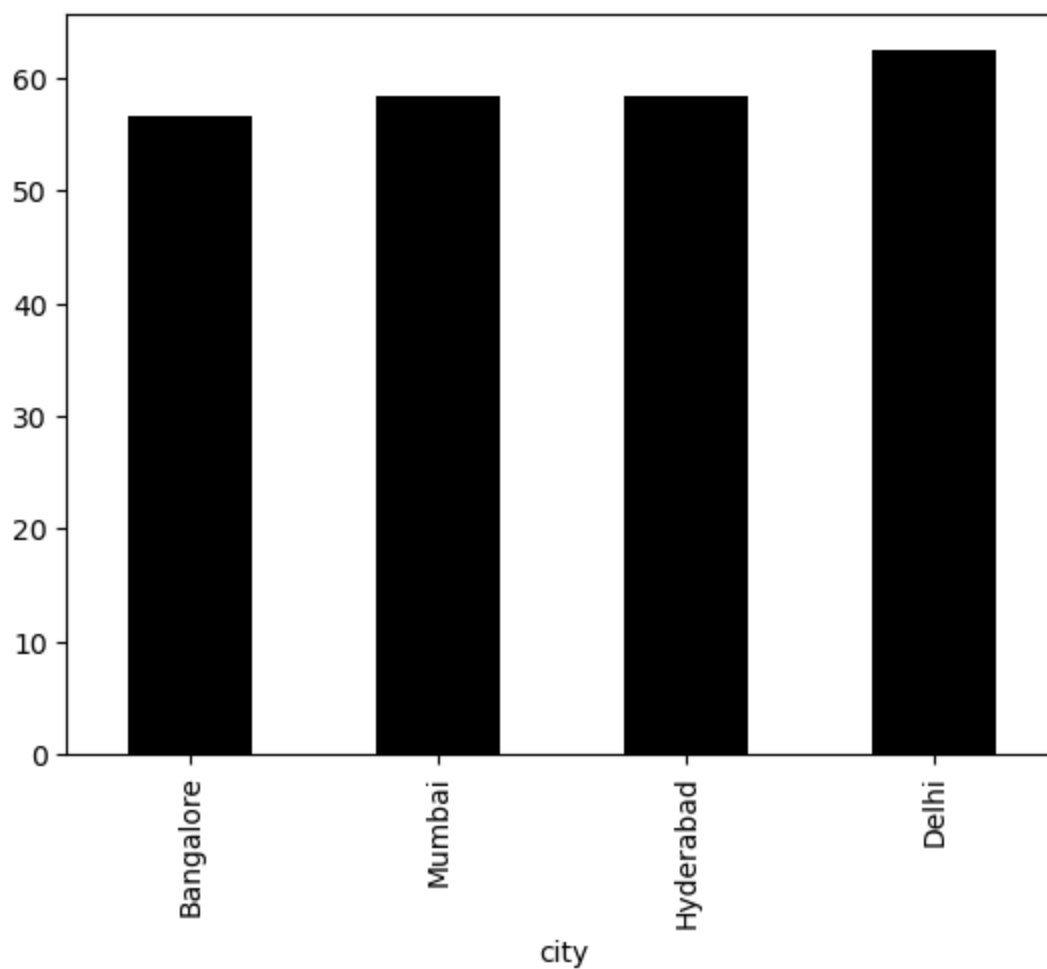
```
Out[46]:
```

	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
2203	17558	10-Jun-22	RT1	9	19.0
2204	16558	10-Jun-22	RT1	11	19.0
...
4295	17562	30-Jun-22	RT4	3	6.0
4296	19563	30-Jun-22	RT4	3	6.0
4297	16560	30-Jun-22	RT4	3	7.0
4298	19558	30-Jun-22	RT4	3	7.0
4299	17561	30-Jun-22	RT4	3	4.0

2100 rows × 14 columns

```
In [47]: df_june_22.groupby('city')['occ_pct'].mean().round(2).sort_values().plot(kind
```

```
Out[47]: <Axes: xlabel='city'>
```



```
In [48]: df_august = pd.read_csv('new_data_august.csv')
df_august.head(3)
```

```
Out[48]:
```

	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 [49]: df.columns
```

```
Out[49]: 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 [50]: df_august.columns
```



```
Out[50]: 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 [51]: df_august.shape
```

```
Out[51]: (7, 13)
```

```
In [52]: df.shape
```

```
Out[52]: (6500, 14)
```

```
In [53]: latest_df = pd.concat([df, df_august], ignore_index = True, axis = 0)
latest_df.tail(4)
```

```
Out[53]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity	
	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 [54]: latest_df.shape
```

```
Out[54]: (6507, 15)
```

5. Print revenue realized per city

```
In [55]: df_bookings.head(4)
```

```
Out[55]:
```

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

```
Out[56]:
```

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

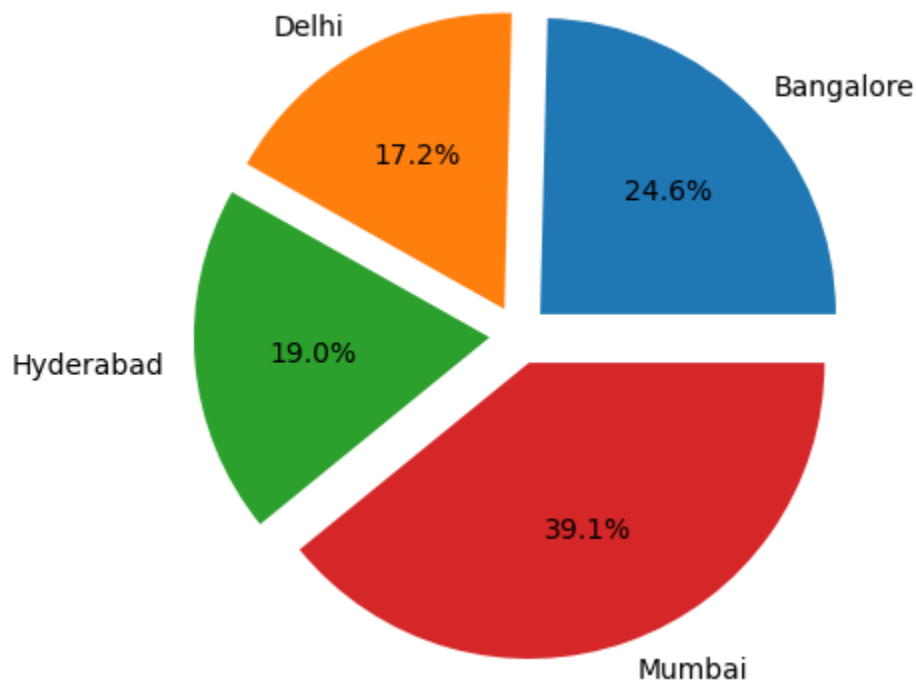
```
In [57]: df_bookings_all = pd.merge(df_bookings, df_hotels, on = 'property_id')
df_bookings_all.head(3)
```

```
Out[57]:
```

	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 [58]: df_bookings_all.groupby('city')['revenue_realized'].sum().plot(kind = 'pie', y
```

```
Out[58]: <Axes: >
```



6. Print month by month revenue

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

```
Out[59]:
```

	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 [60]: df_date['mmm yy'].unique()
```

```
Out[60]: array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

```
In [61]: df_date.head(3)
```

```
Out[61]:
```

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

```
Out[62]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_gues
--	------------	-------------	--------------	---------------	---------------	---------

```
In [63]: df_bookings_all.info()
```

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

```
In [64]: 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 [65]: df_date['date'] = pd.to_datetime(df_date['date'])
df_date.head(3)
```

C:\Users\sheri\AppData\Local\Temp\ipykernel_11820\248606528.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'])
```

```
Out[65]:
```

	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 [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    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 [67]: df_bookings_all['check_in_date'] = pd.to_datetime(
df_bookings_all['check_in_date'],
format='%d-%m-%y', # day-month-2digit year
errors='coerce'
)
df_bookings_all.head(4)
```

```
Out[67]:
```

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

```
In [68]: 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         78783 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 [69]: df_bookings_all = pd.merge(df_bookings_all, df_date, left_on = 'check_in_date'
df_bookings_all.head(3)
```

```
Out[69]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date
0	May132216558RT11	16558	10/5/2022	2022-05-13	15-05-22
1	May132216558RT12	16558	9/5/2022	2022-05-13	14-05-22
2	May132216558RT13	16558	9/5/2022	2022-05-13	14-05-22

```
In [70]: df_bookings_all.groupby('mmm yy')['revenue_realized'].sum()
```

```
Out[70]: mmm yy
Jul 22    329662416
Jun 22    324288215
May 22    347414213
Name: revenue_realized, dtype: int64
```

```
In [71]: df_bookings_all.groupby('mmm yy')['revenue_realized'].sum().sort_values(ascending=True)
kind='bar',
color='teal'
)
plt.xlabel('Month')
plt.ylabel('Revenue Realized')
plt.show()
```

