


```
In [1]: import pandas as pd
```

```
In [3]: df_bookings= pd.read_csv('fact_bookings.csv')
```

```
In [4]: df_bookings.head(5)
```

```
Out[4]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_g
0	May012216558RT11	16558	27-04-22	1/5/2022	2/5/2022	-3.0	RT1	direct online	
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	
3	May012216558RT14	16558	28-04-22	1/5/2022	2/5/2022	-2.0	RT1	others	
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	



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

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

```
In [7]: df_bookings.room_category.unique()
```

```
Out[7]: array(['RT1', 'RT2', 'RT3', 'RT4'], dtype=object)
```

```
In [8]: df_bookings.booking_platform.unique()
```

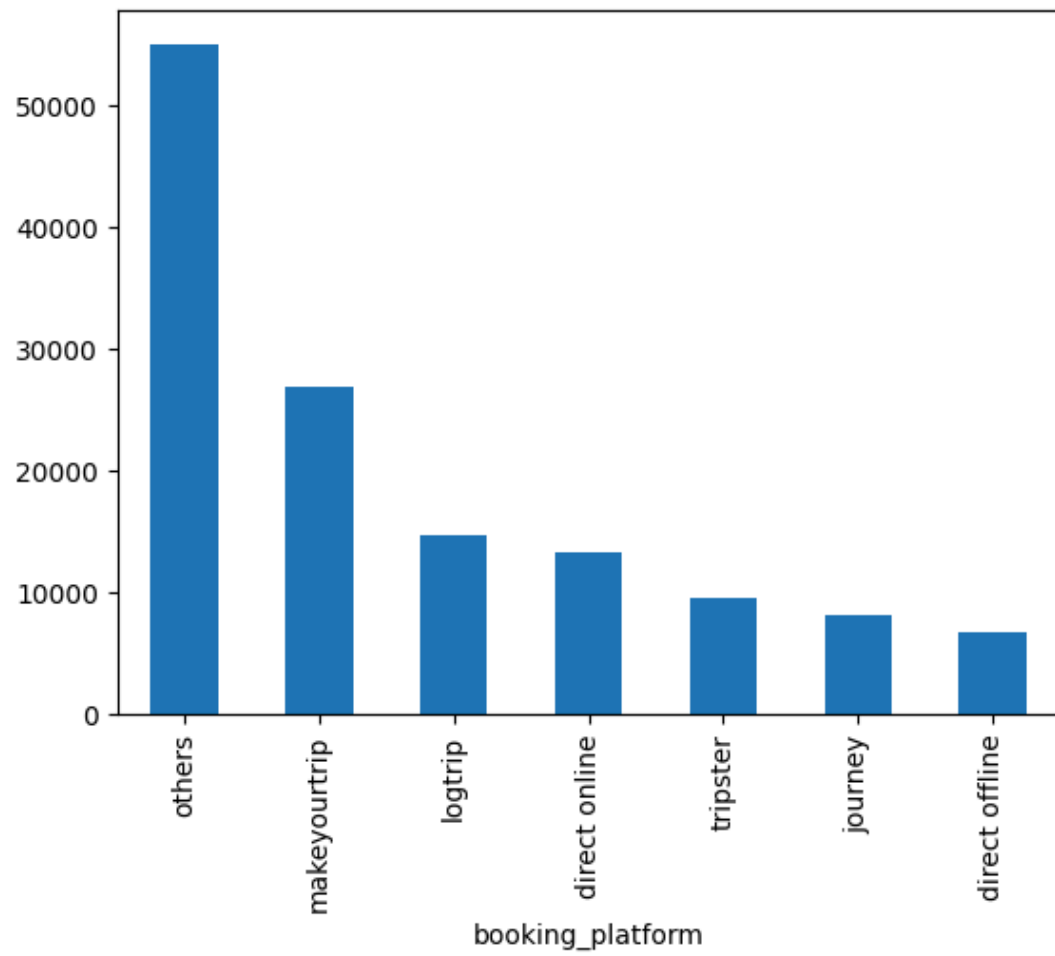
```
Out[8]: array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',  
              'journey', 'direct offline'], dtype=object)
```

```
In [14]: df_bookings.booking_platform.value_counts()
```

```
Out[14]: booking_platform
others      55066
makeyourtrip 26898
logtrip      14756
direct online 13379
tripster     9630
journey      8106
direct offline 6755
Name: count, dtype: int64
```

```
In [24]: #import matplotlib as plt
df_bookings.booking_platform.value_counts().plot(kind="bar")
```

```
Out[24]: <Axes: xlabel='booking_platform'>
```



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

```
Out[25]:
```

	property_id	no_guests	ratings_given	revenue_generated	revenue_realized
<b>count</b>	134590.000000	134587.000000	56683.000000	1.345900e+05	134590.000000
<b>mean</b>	18061.113493	2.036170	3.619004	1.537805e+04	12696.123256
<b>std</b>	1093.055847	1.034885	1.235009	9.303604e+04	6928.108124
<b>min</b>	16558.000000	-17.000000	1.000000	6.500000e+03	2600.000000
<b>25%</b>	17558.000000	1.000000	3.000000	9.900000e+03	7600.000000
<b>50%</b>	17564.000000	2.000000	4.000000	1.350000e+04	11700.000000
<b>75%</b>	18563.000000	2.000000	5.000000	1.800000e+04	15300.000000
<b>max</b>	19563.000000	6.000000	5.000000	2.856000e+07	45220.000000

```
In [33]: 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 [27]: df_hotels.shape
```

```
Out[27]: (25, 4)
```

```
In [29]: df_hotels.head(3)
```

```
Out[29]:
```

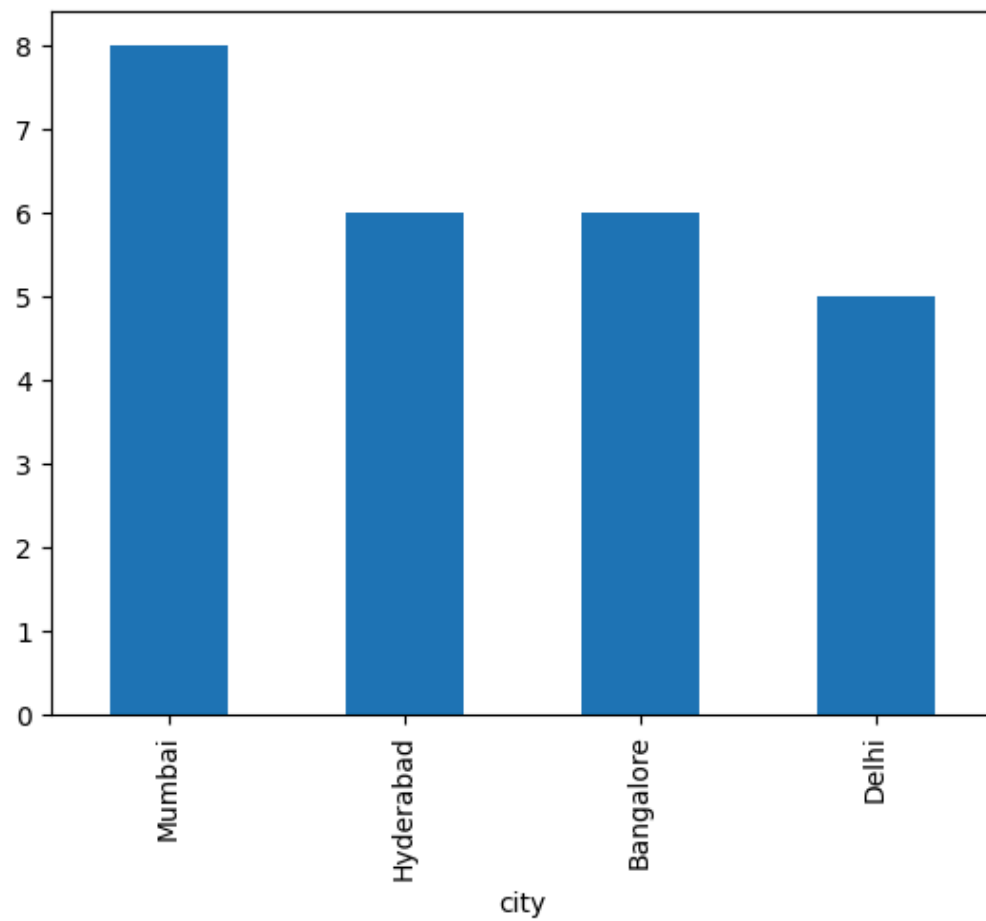
	property_id	property_name	category	city
<b>0</b>	16558	Atliq Grands	Luxury	Delhi
<b>1</b>	16559	Atliq Exotica	Luxury	Mumbai
<b>2</b>	16560	Atliq City	Business	Delhi

```
In [30]: df_hotels.category.value_counts()
```

```
Out[30]: category
Luxury      16
Business     9
Name: count, dtype: int64
```

```
In [31]: df_hotels.city.value_counts().plot(kind="bar")
```

```
Out[31]: <Axes: xlabel='city'>
```



```
In [34]: df_agg_bookings.head()
```

```
Out[34]:
```

	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

### unique property ids in aggregate bookings dataset

```
In [35]: df_agg_bookings.property_id.unique()
```

```
Out[35]: 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])
```

### total bookings per property\_id

```
In [37]: df_agg_bookings.groupby("property_id")["successful_bookings"].sum()
```

```
Out[37]: property_id
16558    3153
16559    7338
16560    4693
16561    4418
16562    4820
16563    7211
17558    5053
17559    6142
17560    6013
17561    5183
17562    3424
17563    6337
17564    3982
18558    4475
18559    5256
18560    6638
18561    6458
18562    7333
18563    4737
19558    4400
19559    4729
19560    6079
19561    5736
19562    5812
19563    5413
Name: successful_bookings, dtype: int64
```

**days on which bookings are greater than capacity**

```
In [42]: df_agg_bookings[df_agg_bookings.successful_bookings>df_agg_bookings.capacity]
```

Out[42]:

	property_id	check_in_date	room_category	successful_bookings	capacity
<b>3</b>	17558	1-May-22	RT1	30	19.0
<b>12</b>	16563	1-May-22	RT1	100	41.0
<b>4136</b>	19558	11-Jun-22	RT2	50	39.0
<b>6209</b>	19560	2-Jul-22	RT1	123	26.0
<b>8522</b>	19559	25-Jul-22	RT1	35	24.0
<b>9194</b>	18563	31-Jul-22	RT4	20	18.0

### properties that have highest capacity

```
In [43]: df_agg_bookings[df_agg_bookings.capacity==df_agg_bookings.capacity.max()]
```

Out[43]:

	property_id	check_in_date	room_category	successful_bookings	capacity
<b>27</b>	17558	1-May-22	RT2	38	50.0
<b>128</b>	17558	2-May-22	RT2	27	50.0
<b>229</b>	17558	3-May-22	RT2	26	50.0
<b>328</b>	17558	4-May-22	RT2	27	50.0
<b>428</b>	17558	5-May-22	RT2	29	50.0
...	...	...	...	...	...
<b>8728</b>	17558	27-Jul-22	RT2	22	50.0
<b>8828</b>	17558	28-Jul-22	RT2	21	50.0
<b>8928</b>	17558	29-Jul-22	RT2	23	50.0
<b>9028</b>	17558	30-Jul-22	RT2	32	50.0
<b>9128</b>	17558	31-Jul-22	RT2	30	50.0

92 rows × 5 columns



## Data cleaning

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

```
Out[44]:
```

	property_id	no_guests	ratings_given	revenue_generated	revenue_realized
<b>count</b>	134590.000000	134587.000000	56683.000000	1.345900e+05	134590.000000
<b>mean</b>	18061.113493	2.036170	3.619004	1.537805e+04	12696.123256
<b>std</b>	1093.055847	1.034885	1.235009	9.303604e+04	6928.108124
<b>min</b>	16558.000000	-17.000000	1.000000	6.500000e+03	2600.000000
<b>25%</b>	17558.000000	1.000000	3.000000	9.900000e+03	7600.000000
<b>50%</b>	17564.000000	2.000000	4.000000	1.350000e+04	11700.000000
<b>75%</b>	18563.000000	2.000000	5.000000	1.800000e+04	15300.000000
<b>max</b>	19563.000000	6.000000	5.000000	2.856000e+07	45220.000000

```
In [45]: df_bookings[df_bookings.no_guests<0]
```

```
Out[45]:
```

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

```
In [46]: df_bookings=df_bookings[df_bookings.no_guests>0]
```

```
In [47]: df_bookings.shape
```

```
Out[47]: (134578, 12)
```

### Outlier removal

```
In [48]: avg, std = df_bookings.revenue_generated.mean(),df_bookings.revenue_generated.std()
```

```
In [50]: higher_limit = avg + 3*std  
higher_limit
```

```
Out[50]: np.float64(294498.50173207896)
```

```
In [51]: lower_limit = avg - 3*std  
lower_limit
```

```
Out[51]: np.float64(-263742.4278567056)
```

```
In [52]: df_bookings = df_bookings[df_bookings.revenue_generated<= higher_limit]
```

```
In [53]: df_bookings.shape
```

```
Out[53]: (134573, 12)
```

```
In [54]: df_bookings.revenue_generated.describe()
```

```
Out[54]: count      134573.000000  
mean        14915.959776  
std         6452.676164  
min         6500.000000  
25%         9900.000000  
50%        13500.000000  
75%        18000.000000  
max        45220.000000  
Name: revenue_generated, dtype: float64
```

```
In [55]: df_bookings.revenue_realized.describe()
```

```
Out[55]: 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
```

### Null values

```
In [57]: df_agg_bookings.isnull().sum()
```

```
Out[57]: property_id      0
         check_in_date    0
         room_category    0
         successful_bookings 0
         capacity         2
         dtype: int64
```

```
In [58]: df_agg_bookings[df_agg_bookings.capacity.isna()]
```

```
Out[58]:
```

	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

```
In [59]: df_agg_bookings.capacity.median()
```

```
Out[59]: np.float64(25.0)
```

```
In [60]: df_agg_bookings.capacity.fillna(df_agg_bookings.capacity.median(), inplace=True)
```

C:\Users\kpatel\AppData\Local\Temp\ipykernel\_24056\2127972865.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)
```

```
In [61]: df_agg_bookings.isna().sum()
```

```
Out[61]: property_id      0  
check_in_date      0  
room_category      0  
successful_bookings  0  
capacity           0  
dtype: int64
```

```
In [62]: df_agg_bookings[df_agg_bookings.successful_bookings > df_agg_bookings.capacity]
```

```
Out[62]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity
<b>3</b>	17558	1-May-22	RT1	30	19.0
<b>12</b>	16563	1-May-22	RT1	100	41.0
<b>4136</b>	19558	11-Jun-22	RT2	50	39.0
<b>6209</b>	19560	2-Jul-22	RT1	123	26.0
<b>8522</b>	19559	25-Jul-22	RT1	35	24.0
<b>9194</b>	18563	31-Jul-22	RT4	20	18.0

```
In [63]: df_agg_bookings.shape
```

```
Out[63]: (9200, 5)
```

```
In [64]: df_agg_bookings = df_agg_bookings[df_agg_bookings.successful_bookings <= df_agg_bookings.capacity]  
df_agg_bookings.shape
```

Out[64]: (9194, 5)

## Data Transformation

### Occupancy Percentage

```
In [67]: df_agg_bookings.head()
```

```
Out[67]:
```

	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
4	16558	1-May-22	RT1	18	19.0
5	17560	1-May-22	RT1	28	40.0

```
In [68]: df_agg_bookings['occ_pct'] = df_agg_bookings.apply(lambda row : row['successful_bookings']/row['capacity'],axis=1)
```

C:\Users\kpatel\AppData\Local\Temp\ipykernel\_24056\232489575.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df_agg_bookings['occ_pct'] = df_agg_bookings.apply(lambda row : row['successful_bookings']/row['capacity'],axis=1)
```

```
In [69]: df_agg_bookings.head()
```

```
Out[69]:
```

	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
4	16558	1-May-22	RT1	18	19.0	0.947368
5	17560	1-May-22	RT1	28	40.0	0.700000

```
In [71]: df_agg_bookings['occ_pct'] = df_agg_bookings['occ_pct'].apply(lambda x: round(x*100,2 ))
```

```
In [72]: df_agg_bookings.head()
```

```
Out[72]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct
0	16559	1-May-22	RT1	25	30.0	83.33
1	19562	1-May-22	RT1	28	30.0	93.33
2	19563	1-May-22	RT1	23	30.0	76.67
4	16558	1-May-22	RT1	18	19.0	94.74
5	17560	1-May-22	RT1	28	40.0	70.00

### Insight Genartion

```
In [73]: df_agg_bookings.head()
```

```
Out[73]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct
0	16559	1-May-22	RT1	25	30.0	83.33
1	19562	1-May-22	RT1	28	30.0	93.33
2	19563	1-May-22	RT1	23	30.0	76.67
4	16558	1-May-22	RT1	18	19.0	94.74
5	17560	1-May-22	RT1	28	40.0	70.00

```
In [74]: df_agg_bookings.groupby("room_category")["occ_pct"].mean()
```

```
Out[74]: room_category
RT1      57.889643
RT2      58.009756
RT3      58.028213
RT4      59.277925
Name: occ_pct, dtype: float64
```

```
In [75]: df = pd.merge(df_agg_bookings,df_rooms, left_on="room_category", right_on="room_id")
df.head()
```

```
Out[75]:
```

	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	83.33	RT1	Standard
1	19562	1-May-22	RT1	28	30.0	93.33	RT1	Standard
2	19563	1-May-22	RT1	23	30.0	76.67	RT1	Standard
3	16558	1-May-22	RT1	18	19.0	94.74	RT1	Standard
4	17560	1-May-22	RT1	28	40.0	70.00	RT1	Standard

```
In [76]: df.drop("room_id",axis=1,inplace=True)
df.head()
```

```
Out[76]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class
0	16559	1-May-22	RT1	25	30.0	83.33	Standard
1	19562	1-May-22	RT1	28	30.0	93.33	Standard
2	19563	1-May-22	RT1	23	30.0	76.67	Standard
3	16558	1-May-22	RT1	18	19.0	94.74	Standard
4	17560	1-May-22	RT1	28	40.0	70.00	Standard

```
In [77]: df.groupby("room_class")["occ_pct"].mean()
```

```
Out[77]: room_class
Elite      58.009756
Premium    58.028213
Presidential 59.277925
Standard   57.889643
Name: occ_pct, dtype: float64
```

### Average Occupancy rate per city

```
In [78]: df_hotels.head()
```

```
Out[78]:
```

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

```
Out[83]:
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city
0	16559	1-May-22	RT1	25	30.0	83.33	Standard	Atliq Exotica	Luxury	Mumbai
1	19562	1-May-22	RT1	28	30.0	93.33	Standard	Atliq Bay	Luxury	Bangalore
2	19563	1-May-22	RT1	23	30.0	76.67	Standard	Atliq Palace	Business	Bangalore
3	16558	1-May-22	RT1	18	19.0	94.74	Standard	Atliq Grands	Luxury	Delhi
4	17560	1-May-22	RT1	28	40.0	70.00	Standard	Atliq City	Business	Mumbai

```
In [84]: df.groupby('city')['occ_pct'].mean()
```

```
Out[84]: city
Bangalore    56.332376
Delhi        61.507341
Hyderabad    58.120652
Mumbai       57.909181
Name: occ_pct, dtype: float64
```

### Occupancy distribution by weekday and weekend

```
In [85]: df_date.head()
```



Out[85]:

	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

```
In [86]: df = pd.merge(df,df_date, left_on='check_in_date', right_on='date')
df.head()
```

Out[86]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city
0	19563	10-May-22	RT3	15	29.0	51.72	Premium	Atliq Palace	Business	Bangalore
1	18560	10-May-22	RT1	19	30.0	63.33	Standard	Atliq City	Business	Hyderabad
2	19562	10-May-22	RT1	18	30.0	60.00	Standard	Atliq Bay	Luxury	Bangalore
3	19563	10-May-22	RT1	16	30.0	53.33	Standard	Atliq Palace	Business	Bangalore
4	17558	10-May-22	RT1	11	19.0	57.89	Standard	Atliq Grands	Luxury	Mumbai



```
In [87]: df.groupby('day_type')['occ_pct'].mean().round(2)
```

```
Out[87]: day_type
         weekday    50.88
         weekend     72.34
         Name: occ_pct, dtype: float64
```

### Occupancy in the month of June

```
In [89]: df_june_22 = df[df['mmm yy'] == "Jun 22"]
```

```
In [91]: df_june_22.occ_pct.mean().round(2)
```

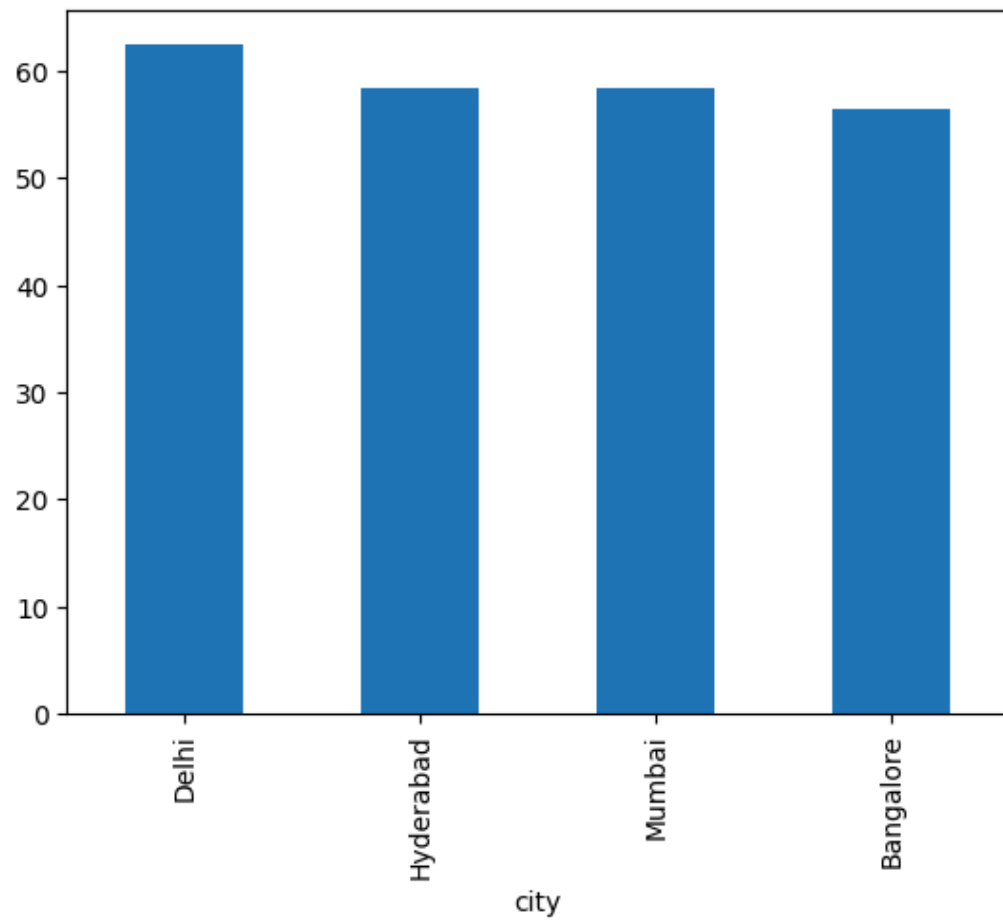
```
Out[91]: np.float64(58.75)
```

```
In [93]: df_june_22.groupby('city')['occ_pct'].mean().round(2).sort_values(ascending=False)
```

```
Out[93]: city
         Delhi      62.47
         Hyderabad  58.46
         Mumbai    58.38
         Bangalore  56.44
         Name: occ_pct, dtype: float64
```

```
In [94]: df_june_22.groupby('city')['occ_pct'].mean().round(2).sort_values(ascending=False).plot(kind='bar')
```

```
Out[94]: <Axes: xlabel='city'>
```




**Add new data for the month of August**

```
In [95]: df_august = pd.read_csv("new_data_august.csv")
df_august.head()
```

Out[95]:

	property_id	property_name	category	city	room_category	room_class	check_in_date	mmm yy	week no	day_type	successful_booking
0	16559	Atliq Exotica	Luxury	Mumbai	RT1	Standard	01-Aug-22	Aug-22	W 32	weekday	
1	19562	Atliq Bay	Luxury	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	weekday	
2	19563	Atliq Palace	Business	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	weekday	
3	19558	Atliq Grands	Luxury	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	weekday	
4	19560	Atliq City	Business	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	weekday	



In [97]: `df_august.columns`

Out[97]: 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 [98]: `df.columns`

Out[98]: 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 [99]: `df_august.shape`

Out[99]: (7, 13)

In [100]: `df.shape`

Out[100]: (6497, 14)

In [104]: `latest_df = pd.concat([df,df_august],ignore_index = True, axis=0)  
latest_df.tail()`

Out[104...

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	ci
<b>6499</b>	19563	01-Aug-22	RT1	23	30.0	NaN	Standard	Atliq Palace	Business	Bangalc
<b>6500</b>	19558	01-Aug-22	RT1	30	40.0	NaN	Standard	Atliq Grands	Luxury	Bangalc
<b>6501</b>	19560	01-Aug-22	RT1	20	26.0	NaN	Standard	Atliq City	Business	Bangalc
<b>6502</b>	17561	01-Aug-22	RT1	18	26.0	NaN	Standard	Atliq Blu	Luxury	Mumk
<b>6503</b>	17564	01-Aug-22	RT1	10	16.0	NaN	Standard	Atliq Seasons	Business	Mumk

In [105...

```
latest_df.shape
```

Out[105...

```
(6504, 15)
```

### Revenue realized per city

In [106...

```
df_bookings.head()
```

Out[106...

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_g
<b>1</b>	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	
<b>4</b>	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	
<b>5</b>	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	
<b>6</b>	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1	others	
<b>7</b>	May012216558RT18	16558	26-04-22	1/5/2022	3/5/2022	2.0	RT1	logtrip	

In [107...

```
df_hotels.head()
```

Out[107...

	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 [108...

```
df_bookings_all = pd.merge(df_bookings,df_hotels, on='property_id')
df_bookings_all.head()
```

Out[108...

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_g
0	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	
1	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	
2	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	
3	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1	others	
4	May012216558RT18	16558	26-04-22	1/5/2022	3/5/2022	2.0	RT1	logtrip	



In [111...

```
df_bookings_all.groupby('city')['revenue_realized'].sum()
```

Out[111...

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

### Month by month revenue

In [112...

```
df_date.head()
```

Out[112...

	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

In [113...

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

Out[113...

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

In [114...

```
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 [115...

```
df_date['date'] = pd.to_datetime(df_date['date'])
df_date.head()
```

```
C:\Users\kpatel\AppData\Local\Temp\ipykernel_24056\269778243.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[115...

	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
3	2022-05-04	May 22	W 19	weekeday
4	2022-05-05	May 22	W 19	weekeday

In [116...

```
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 [117...

```
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 [131...] df_bookings_all["check_in_date"] = pd.to_datetime(df_bookings_all["check_in_date"],format='%d-%m-%Y')
df_bookings_all.head(4)

```

Out[131...]

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_g
0	May012216558RT12	16558	30-04-22	2022-01-05	2/5/2022	2.0	RT1	others	
1	May012216558RT15	16558	27-04-22	2022-01-05	2/5/2022	4.0	RT1	direct online	
2	May012216558RT16	16558	1/5/2022	2022-01-05	3/5/2022	2.0	RT1	others	
3	May012216558RT17	16558	28-04-22	2022-01-05	6/5/2022	2.0	RT1	others	



```

In [132...] df_bookings_all = pd.merge(df_bookings_all, df_date, left_on='check_in_date',right_on='date')
df_bookings_all.head()

```

Out[132...

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_g
0	May052216558RT11	16558	15-04-22	2022-05-05	7/5/2022	3.0	RT1	tripster	
1	May052216558RT12	16558	30-04-22	2022-05-05	7/5/2022	2.0	RT1	others	
2	May052216558RT13	16558	1/5/2022	2022-05-05	6/5/2022	3.0	RT1	direct offline	
3	May052216558RT14	16558	3/5/2022	2022-05-05	6/5/2022	2.0	RT1	tripster	
4	May052216558RT15	16558	30-04-22	2022-05-05	10/5/2022	4.0	RT1	others	

In [133...

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

Out[133...

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

### Revenue realized per hotel

In [135...

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

Out[135...

```
property_name
Atliq Seasons    45920757
Atliq Grands    145860641
Atliq Blu        179203544
Atliq Bay        179416721
Atliq City       196555383
Atliq Palace     209474575
Atliq Exotica    219076161
Name: revenue_realized, dtype: int64
```

### Average rating per city

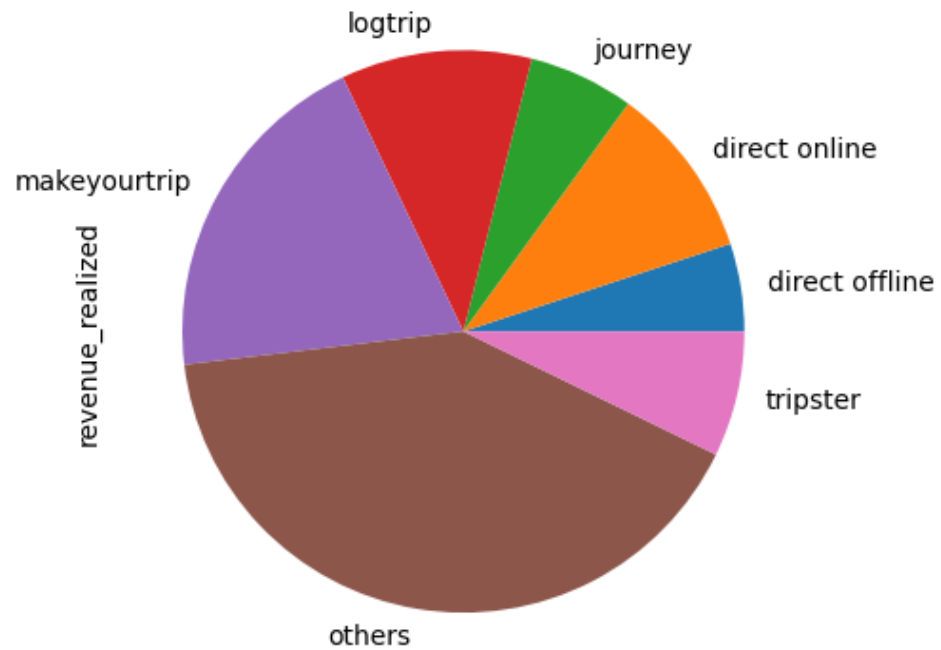
```
In [136... df_bookings_all.groupby('city')['ratings_given'].mean()
```

```
Out[136... city
Bangalore    3.403911
Delhi        3.775088
Hyderabad    3.664286
Mumbai       3.644350
Name: ratings_given, dtype: float64
```

### Pie chart of revenue realizer per booking platform

```
In [137... df_bookings_all.groupby('booking_platform')['revenue_realized'].sum().plot(kind="pie")
```

```
Out[137... <Axes: ylabel='revenue_realized'>
```



```
In [ ]:
```