Data Exploration

In [3]:
```
import pandas as pd
df_bookings = pd.read_csv("datasets/fact_bookings.csv")
df_bookings.head(4)
```

Out[3]:

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

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

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

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

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

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

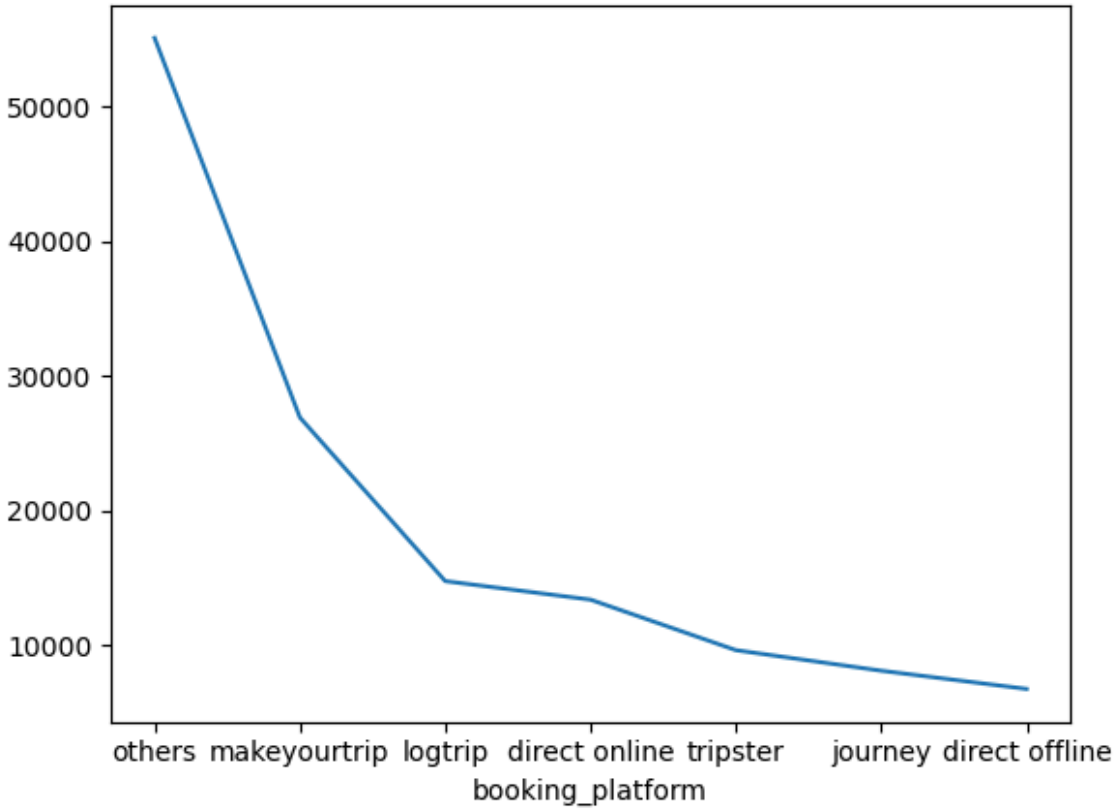
Out[7]:
```
array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',
       'journey', 'direct offline'], dtype=object)
```

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

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

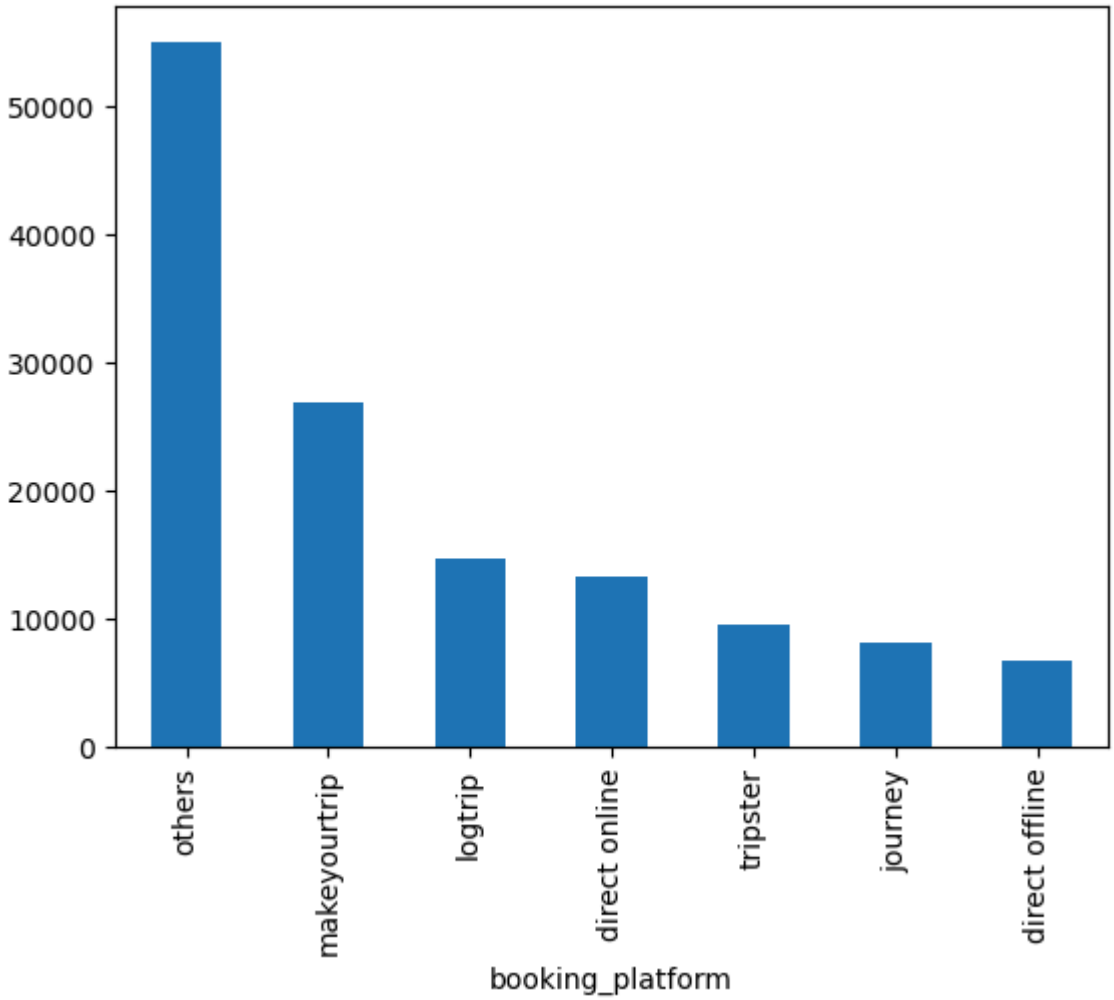
In [12]:
```
df_bookings.booking_platform.value_counts().plot()
```

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



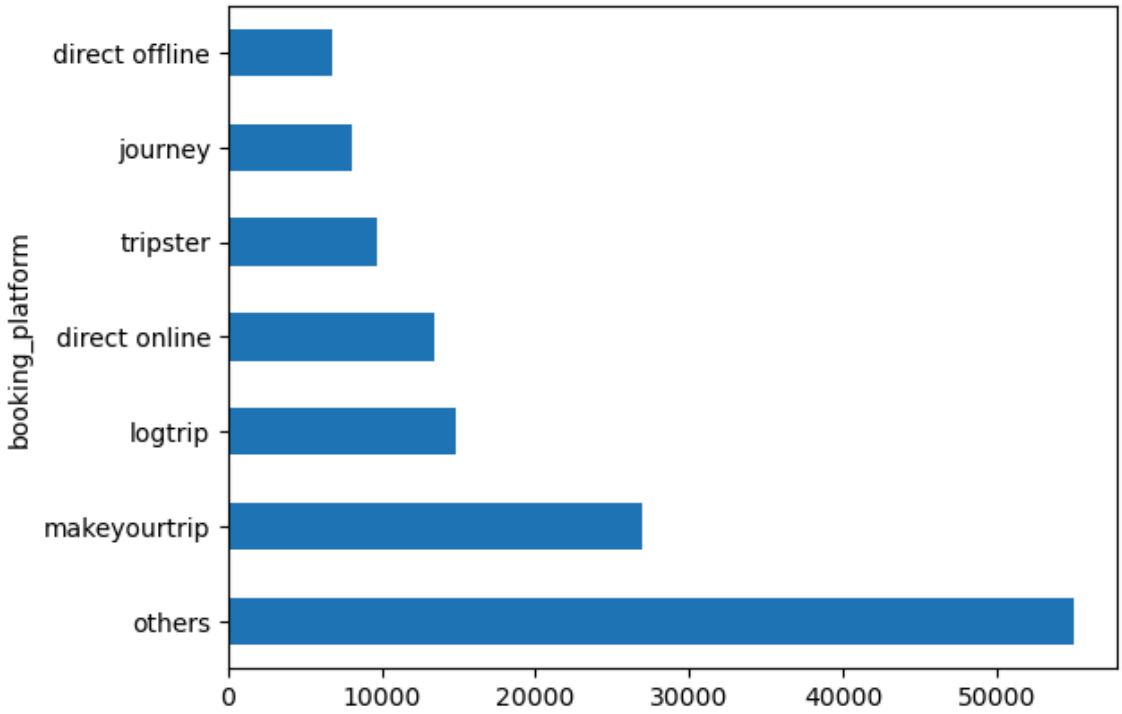
In [13]:
```
df_bookings.booking_platform.value_counts().plot(kind="bar")
```

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



In [14]:
```
df_bookings.booking_platform.value_counts().plot(kind="barh")
```

Out[14]:
```
<Axes: ylabel='booking_platform'>
```



In [15]:
```
df_bookings.describe()
```

Out[15]:

	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 [17]:
```
df_bookings.revenue_generated.min(),df_bookings.revenue_generated.max() # d
```

Out[17]:
```
(6500, 28560000)
```

In [20]:
```
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 [21]:
```
df_hotels.shape
```

Out[21]:
```
(25, 4)
```

In [22]:
```
df_hotels.head(4)
```

Out[22]:

	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 [23]:
```
df_hotels.category.value_counts()
```

Out[23]:
```
category
Luxury      16
Business     9
Name: count, dtype: int64
```

In [24]:
```
df_hotels.city.value_counts()
```

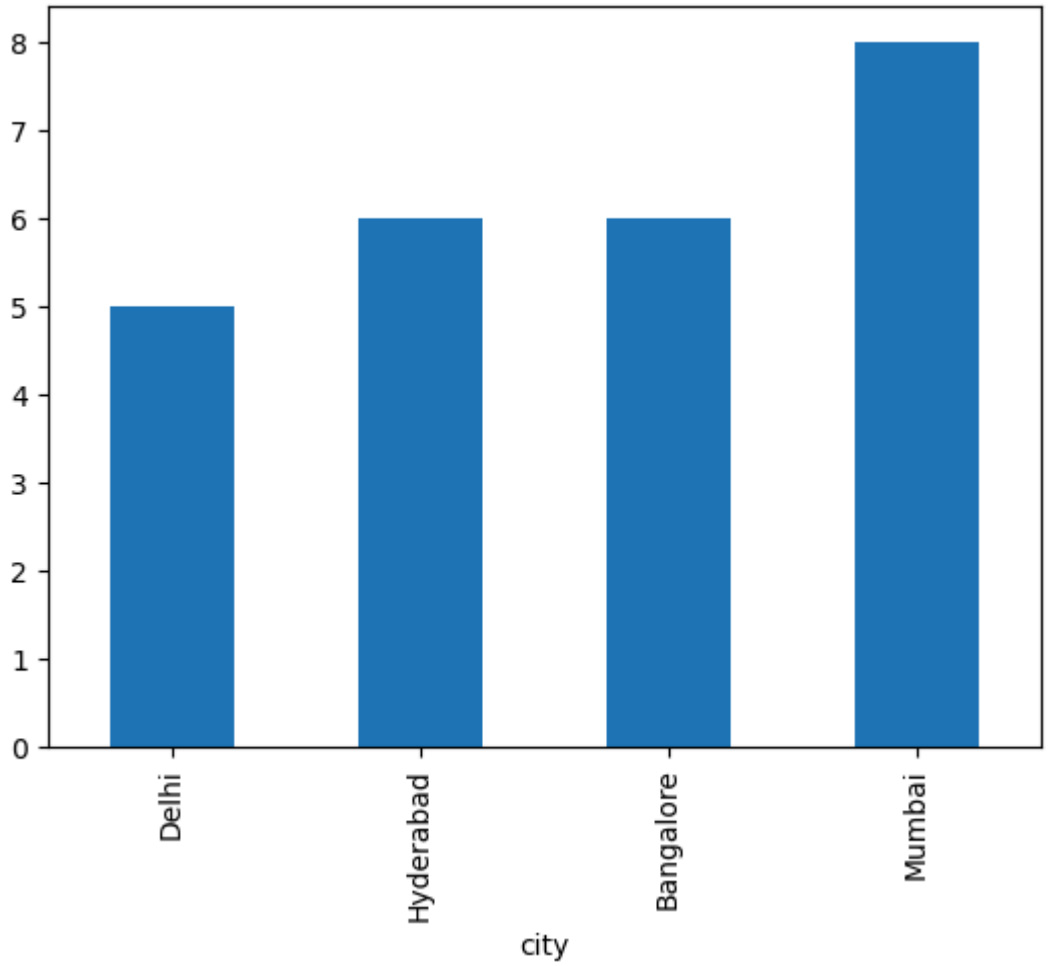
Out[24]:
```
city
Mumbai       8
Hyderabad    6
Bangalore    6
Delhi        5
Name: count, dtype: int64
```

In [25]:
```
df_hotels.city.value_counts().sort_values()
```

Out[25]:
```
city
Delhi        5
Hyderabad    6
Bangalore    6
Mumbai       8
Name: count, dtype: int64
```

In [26]:
```
df_hotels.city.value_counts().sort_values().plot(kind="bar")
```

Out[26]:
```
<Axes: xlabel='city'>
```



In [27]:
```
df_agg_bookings.head(3)
```

Out[27]:

	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

In [28]:
```
df_agg_bookings.property_id.unique()
```

Out[28]:
```
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], dtype=int64)
```

In [30]:
```
df_agg_bookings.property_id.value_counts()
```

Out[30]:
```
property_id
16559    368
17559    368
17564    368
19561    368
19559    368
18563    368
18562    368
18561    368
18559    368
18558    368
17563    368
17562    368
16563    368
19562    368
16562    368
16561    368
16560    368
17561    368
19560    368
19558    368
17560    368
16558    368
17558    368
19563    368
18560    368
Name: count, dtype: int64
```

In [34]:
```
df_agg_bookings.capacity.max()
```

Out[34]:
```
50.0
```

Data Cleaning

In [35]:
```
df_bookings.describe()   # no_guests has some negative values
```

Out[35]:

	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 [36]:
```
df_bookings[df_bookings.no_guests <= 0]
```

Out[36]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_gu
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	14-05-22	
18020	May122218561RT22	18561	8/5/2022	12/5/2022	14-05-22	
18119	May122218562RT311	18562	5/5/2022	12/5/2022	17-05-22	
18121	May122218562RT313	18562	10/5/2022	12/5/2022	17-05-22	
56715	Jun082218562RT12	18562	5/6/2022	8/6/2022	13-06-22	
119765	Jul202219560RT220	19560	19-07-22	20-07-22	22-07-22	
134586	Jul312217564RT47	17564	30-07-22	31-07-22	1/8/2022	

In [37]:
```
df_bookings.shape
```

Out[37]:
```
(134590, 12)
```

remove the negative values

In [38]:
```
df_bookings = df_bookings[df_bookings.no_guests >= 0]
df_bookings
```

Out[38]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_gu
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 [39]:
```
df_bookings.shape
```

Out[39]:
```
(134578, 12)
```

In [40]:
```
df_bookings.revenue_generated.min(), df_bookings.revenue_generated.max(),
```

Out[40]:
```
(6500, 28560000)
```

In [44]:
```
avg, std = df_bookings.revenue_generated.mean(), df_bookings.revenue_genera
```

In [45]:
```
avg , std
```

Out[45]:
```
(15378.036937686695, 93040.15493143328)
```

In [47]:
```
higher_limit = avg + 3*std
higher_limit
```

Out[47]:
```
294498.50173198653
```

In [48]:
```
lower_limit = avg - 3*std
lower_limit
```

Out[48]:
```
-263742.4278566132
```

In [49]:
```
df_bookings[df_bookings.revenue_generated>higher_limit] # revenue_generated
```

Out[49]:

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

In [51]:
```
df_bookings = df_bookings[df_bookings.revenue_generated<higher_limit]
df_bookings.shape
```

Out[51]:
```
(134573, 12)
```

In [52]:
```
df_bookings.revenue_realized.describe()
```

Out[52]:
```
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 [53]:
```
higher_limit = df_bookings.revenue_realized.mean() + 3*df_bookings.revenue_
higher_limit
```

Out[53]:
```
33479.3586618449
```

In [54]:
```
df_bookings[df_bookings.revenue_realized>higher_limit]
```

Out[54]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_g
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 × 12 columns

In [55]:
```
df_rooms
```

Out[55]:

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

In [57]:
```
df_bookings[df_bookings.room_category=="RT4"].revenue_realized
```

Out[57]:
```
47         10640
48         26600
49         10640
137        38760
138        12920
           ...
134584     32300
134585     32300
134587     12920
134588     32300
134589     12920
Name: revenue_realized, Length: 16071, dtype: int64
```

In [58]:
```
df_bookings[df_bookings.room_category=="RT4"].revenue_realized.describe()
```

Out[58]:
```
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 [60]:
```
23439 + 3*9048          #higher_limit check for RT4 rooms
```

Out[60]:
```
50583
```

In [62]:
```
df_bookings.isnull().sum()
```

Out[62]:
```
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
```

In [64]:
```
df_agg_bookings.isnull().sum()
```

Out[64]:
```
property_id            0
check_in_date          0
room_category          0
successful_bookings    0
capacity               2
dtype: int64
```

Data Transformation

In [98]:
```
df_agg_bookings.head()
```

Out[98]:

	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 [99]:
```
df_agg_bookings["occ_pct"] = df_agg_bookings["successful_bookings"]/df_agg_
```

In [100]:
```
df_agg_bookings.head()
```

Out[100]:

	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 [101]:
```
df_agg_bookings["occ_pct"] = df_agg_bookings["occ_pct"].apply(lambda x: rou
df_agg_bookings.head()
```

Out[101]:

	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
3	17558	1-May-22	RT1	30	19.0	157.89
4	16558	1-May-22	RT1	18	19.0	94.74

Insights Generation

Ad Hoc Analysis

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

In [103]:
```
df_agg_bookings.groupby("room_category")["occ_pct"].mean().round(2)
```

Out[103]:
```
room_category
RT1    58.22
RT2    58.04
RT3    58.03
RT4    59.30
Name: occ_pct, dtype: float64
```

In [104]:
```
df_rooms
```

Out[104]:

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

In [105]:
```
df = pd.merge(df_agg_bookings,df_rooms,left_on="room_category", right_on="r
df.head(4)
```

Out[105]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room
0	16559	1-May-22	RT1	25	30.0	83.33	F
1	19562	1-May-22	RT1	28	30.0	93.33	F
2	19563	1-May-22	RT1	23	30.0	76.67	F
3	17558	1-May-22	RT1	30	19.0	157.89	F

In [107]:
```
df.groupby("room_class")["occ_pct"].mean().round(2)
```

Out[107]:
```
room_class
Elite           58.04
Premium         58.03
Presidential    59.30
Standard        58.22
Name: occ_pct, dtype: float64
```

In [116]:
```
df.head(4)
```

Out[116]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room
0	16559	1-May-22	RT1	25	30.0	83.33	St
1	19562	1-May-22	RT1	28	30.0	93.33	St
2	19563	1-May-22	RT1	23	30.0	76.67	St
3	17558	1-May-22	RT1	30	19.0	157.89	St

2. Print average occupancy rate per city

In [117]:
```
df_hotels.head(4)
```

Out[117]:

	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 [118]:
```
df = pd.merge(df,df_hotels, on="property_id")
df.head(3)
```

Out[118]:

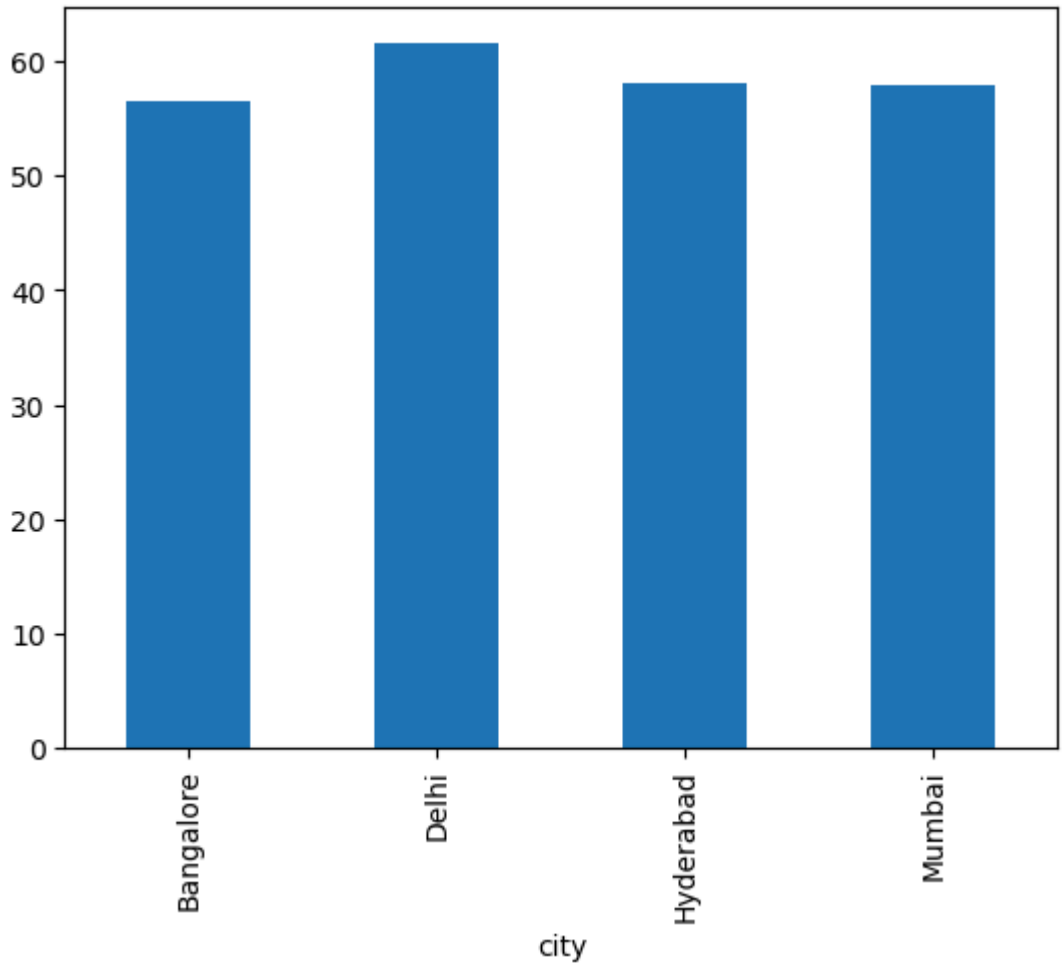
	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room
0	16559	1-May-22	RT1	25	30.0	83.33	St
1	16559	2-May-22	RT1	20	30.0	66.67	St
2	16559	3-May-22	RT1	17	30.0	56.67	St

In [120]:
```
df.groupby("city")["occ_pct"].mean().round(2)
```

Out[120]:
```
city
Bangalore    56.59
Delhi        61.61
Hyderabad    58.14
Mumbai       57.94
Name: occ_pct, dtype: float64
```

In [122]:
```
df.groupby("city")["occ_pct"].mean().round(2).plot(kind="bar")
```

Out[122]:
```
<Axes: xlabel='city'>
```



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

In [123]:
```
df_date.head(4)
```

Out[123]:

	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

In [124]:
```
df = pd.merge(df,df_date, left_on="check_in_date", right_on="date")
df.head(3)
```

Out[124]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room
0	16559	10-May-22	RT1	18	30.0	60.00	St
1	16559	10-May-22	RT2	25	41.0	60.98	
2	16559	10-May-22	RT3	20	32.0	62.50	Pr

In [125]:
```
df.drop("date",axis=1,inplace=True)
df
```

Out[125]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	ro
0	16559	10-May-22	RT1	18	30.0	60.00	
1	16559	10-May-22	RT2	25	41.0	60.98	
2	16559	10-May-22	RT3	20	32.0	62.50	
3	16559	10-May-22	RT4	13	18.0	72.22	P
4	19562	10-May-22	RT1	18	30.0	60.00	
...	...	...	...	...	...	...	
6495	17564	31-Jul-22	RT4	10	17.0	58.82	P
6496	18560	31-Jul-22	RT1	22	30.0	73.33	
6497	18560	31-Jul-22	RT2	34	40.0	85.00	
6498	18560	31-Jul-22	RT3	17	24.0	70.83	
6499	18560	31-Jul-22	RT4	12	15.0	80.00	P

6500 rows × 13 columns

In [126]:
```
df.groupby("day_type")["occ_pct"].mean().round(2)
```

Out[126]:
```
day_type
weekeday    50.90
weekend     72.39
Name: occ_pct, dtype: float64
```

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

In [127]:
```
df["mmm yy"].unique()
```

Out[127]:
```
array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

In [130]:
```
df_june_22 = df[df["mmm yy"]=="Jun 22"]
df_june_22.head(3)
```

Out[130]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	ro
2200	16559	10-Jun-22	RT1	20	30.0	66.67	
2201	16559	10-Jun-22	RT2	26	41.0	63.41	
2202	16559	10-Jun-22	RT3	20	32.0	62.50	

In [132]:
```
df_june_22.groupby("city")["occ_pct"].mean().round(2).sort_values(ascending
```

Out[132]:
```
city
Delhi        62.47
Hyderabad    58.46
Mumbai       58.38
Bangalore    56.58
Name: occ_pct, dtype: float64
```

In [134]:
```
df_august = pd.read_csv("datasets/new_data_august.csv")
df_august.head(3)
```

Out[134]:

	property_id	property_name	category	city	room_category	room_class	check_in_da
0	16559	Atliq Exotica	Luxury	Mumbai	RT1	Standard	01-Aug-2
1	19562	Atliq Bay	Luxury	Bangalore	RT1	Standard	01-Aug-2
2	19563	Atliq Palace	Business	Bangalore	RT1	Standard	01-Aug-2

In [135]:
```
df_august.columns
```

Out[135]:
```
Index(['property_id', 'property_name', 'category', 'city', 'room_categor
y',
       'room_class', 'check_in_date', 'mmm yy', 'week no', 'day_type',
       'successful_bookings', 'capacity', 'occ%'],
      dtype='object')
```

In [136]:
```
df.columns
```

Out[136]:
```
Index(['property_id', 'check_in_date', 'room_category', 'successful_bookin
gs',
       'capacity', 'occ_pct', 'room_class', 'property_name', 'category',
       'city', 'mmm yy', 'week no', 'day_type'],
      dtype='object')
```

In [137]:
```
df.shape
```

Out[137]: (6500, 13)

In [138]:
```
df_august.shape
```

Out[138]: (7, 13)

In [139]:
```
latest_df = pd.concat([df,df_august],ignore_index=True,axis=0)
latest_df
```

Out[139]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	ro
0	16559	10-May-22	RT1	18	30.0	60.00	
1	16559	10-May-22	RT2	25	41.0	60.98	
2	16559	10-May-22	RT3	20	32.0	62.50	
3	16559	10-May-22	RT4	13	18.0	72.22	P
4	19562	10-May-22	RT1	18	30.0	60.00	
...	...	...	...	...	...	...	
6502	19563	01-Aug-22	RT1	23	30.0	NaN	
6503	19558	01-Aug-22	RT1	30	40.0	NaN	
6504	19560	01-Aug-22	RT1	20	26.0	NaN	
6505	17561	01-Aug-22	RT1	18	26.0	NaN	
6506	17564	01-Aug-22	RT1	10	16.0	NaN	

6507 rows × 14 columns

6. Print revenue realized per city

In [140]:
```
df_bookings.head(3)
```

Out[140]:

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

In [141]:
```
df_hotels.head(3)
```

Out[141]:

	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 [142]:
```
df_bookings_all = pd.merge(df_bookings, df_hotels, on="property_id")
df_bookings_all.head(4)
```

Out[142]:

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

In [143]:
```
df_bookings_all.groupby("city")["revenue_realized"].sum()
```

Out[143]:
```
city
Bangalore    420383550
Delhi        294404488
Hyderabad    325179310
Mumbai       668569251
Name: revenue_realized, dtype: int64
```

7. Print month by month revenue

In [144]:
```
df_date["mmm yy"].unique()
```

Out[144]:
```
array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

In [145]:
```
df_date.head(3)
```

Out[145]:

	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 [146]:
```
df_bookings_all.head(3)
```

Out[146]:

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

In [147]:
```
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 [148]:
```
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 [150]:
```
df_date["date"] = pd.to_datetime(df_date["date"])
df_date.head(3)
```

Out[150]:

	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 [151]:
```
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 [157]:
```
df_bookings_all["check_in_date"] = pd.to_datetime(df_bookings_all["check_in
df_bookings_all.head(3)
```

Out[157]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests
0	May012216558RT12	16558	30-04-22	2022-01-05	2/5/2022	2.0
1	May012216558RT15	16558	27-04-22	2022-01-05	2/5/2022	4.0
2	May012216558RT16	16558	1/5/2022	2022-01-05	3/5/2022	2.0

In [158]:
```
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 [159]:
```
df_bookings_all = pd.merge(df_bookings_all,df_date, left_on="check_in_date"
df_bookings_all.head(4)
```

Out[159]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests
0	May052216558RT11	16558	15-04-22	2022-05-05	7/5/2022	3.0
1	May052216558RT12	16558	30-04-22	2022-05-05	7/5/2022	2.0
2	May052216558RT13	16558	1/5/2022	2022-05-05	6/5/2022	3.0
3	May052216558RT14	16558	3/5/2022	2022-05-05	6/5/2022	2.0

In [160]:
```
df_bookings_all.groupby("mmm yy")["revenue_realized"].sum()
```

Out[160]:
```
mmm yy
Jul 22    389940912
Jun 22    377191229
May 22    408375641
Name: revenue_realized, dtype: int64
```