In [ ]:
```
import pandas as pd
import numpy as np
```

In [ ]:
```
df_bookings = pd.read_csv("C:\\Users\\user\\Downloads\\Hotel dataset\\fact_bookings
df_bookings.head(4)
```

Out[ ]:

	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 [ ]:
```
df_bookings.shape
```

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

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

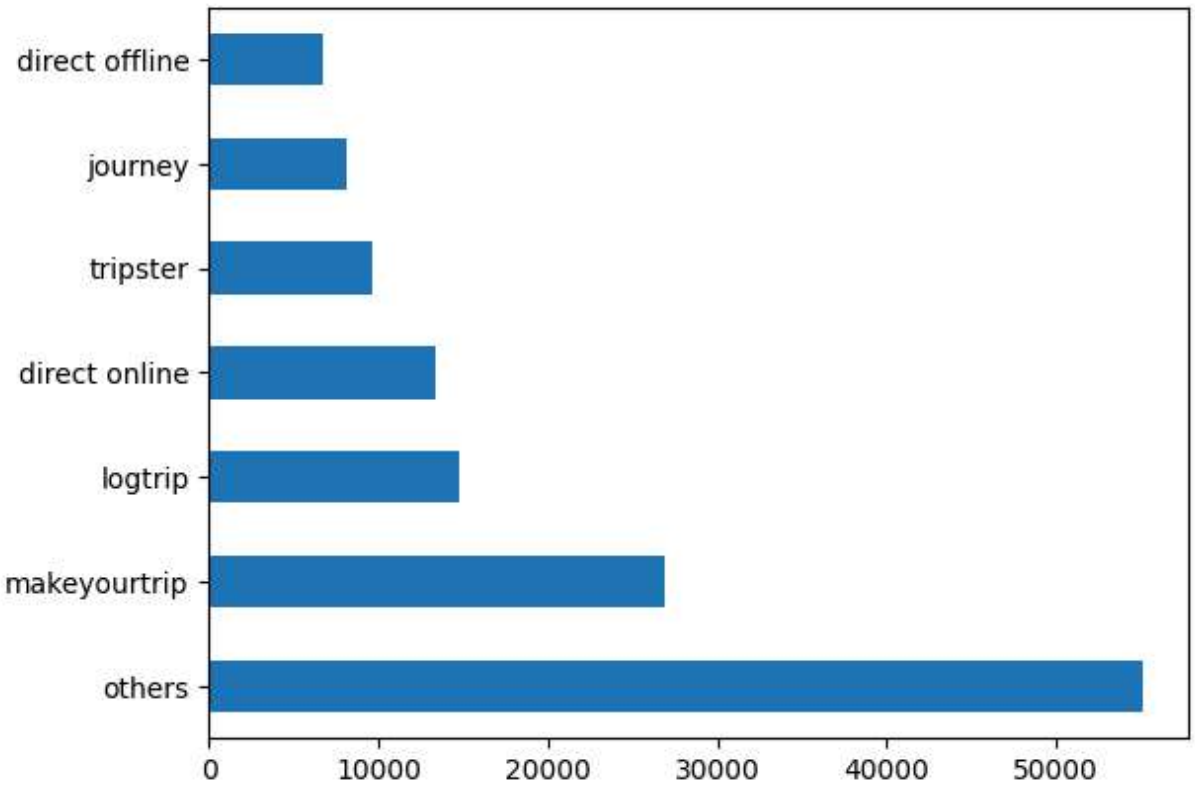
Out[ ]:
```
array(['RT1', 'RT2', 'RT3', 'RT4'], dtype=object)
```

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

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

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

Out[ ]:
```
<AxesSubplot:>
```



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

Out[ ]:

	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 [ ]:
```
df_date = pd.read_csv("C:\\Users\\user\\Downloads\\Hotel dataset\\dim_date.csv")
df_hotels = pd.read_csv("C:\\Users\\user\\Downloads\\Hotel dataset\\dim_hotels.csv"
df_rooms = pd.read_csv("C:\\Users\\user\\Downloads\\Hotel dataset\\dim_rooms.csv")
df_agg_bookings = pd.read_csv("C:\\Users\\user\Downloads\\Hotel dataset\\fact_aggre
```

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

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

In [ ]:
```
df_hotels.head(5)
```

Out[ ]:

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

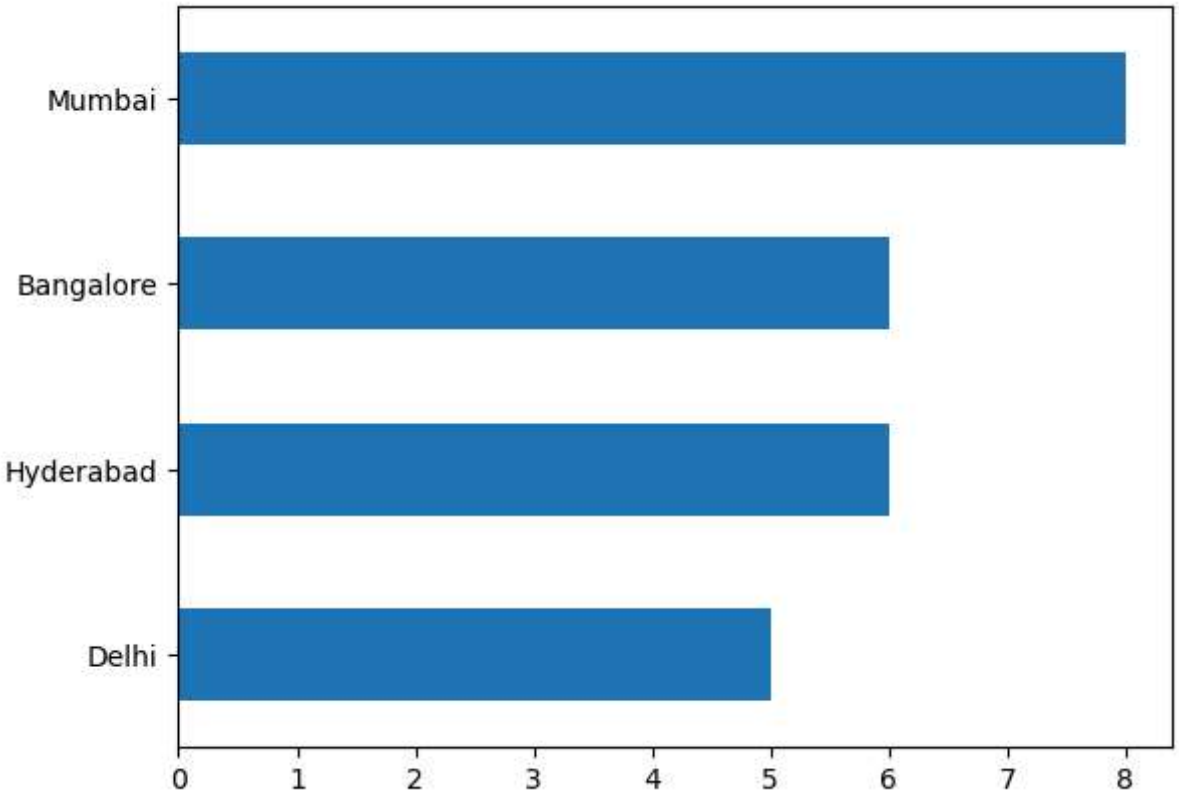
Out[ ]:
```
Luxury      16
Business     9
Name: category, dtype: int64
```

In [ ]:
```
df_hotels.property_name.value_counts()
```

Out[ ]:
```
Atliq Grands     4
Atliq Exotica    4
Atliq City       4
Atliq Blu        4
Atliq Bay        4
Atliq Palace     4
Atliq Seasons    1
Name: property_name, dtype: int64
```

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

Out[ ]:
```
<AxesSubplot:>
```



In [ ]:
```
df_agg_bookings.head(5)
```

Out[ ]:

	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

Excerise -1 Find out unique property ids in aggregate bookings dataset

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

Out[ ]:
```
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)
```

Excerise - 2 Find out total bookings per property id

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

Out[ ]:
```
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
```

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

Out[ ]:

	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

Excercise - 3 Find out days on which bookings are greater than capacity

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

Out[ ]:

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

Excecise - 4 Find out properties that have highest capacity

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

Out[ ]: 50.0

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

Out[ ]:

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

92 rows × 5 columns

==> Data Cleaning

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

Out[ ]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_
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 [ ]:
```
df_bookings = df_bookings[df_bookings.no_guests > 0]
df_bookings
```

Out[ ]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_g
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 [ ]:
```
df_bookings.revenue_generated.min(), df_bookings.revenue_generated.max()
```

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

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

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

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

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

Out[ ]:
```
294498.50173207896
```

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

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

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

Out[ ]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_
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 [ ]:
```
 df_bookings[df_bookings.revenue_generated < higher_limit]
df_bookings
```

Out[ ]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_g
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 [ ]:
```
df_bookings.revenue_realized.describe()
```

Out[ ]:
```
count    134578.000000
mean      12696.011822
std        6927.841641
min        2600.000000
25%        7600.000000
50%       11700.000000
75%       15300.000000
max       45220.000000
Name: revenue_realized, dtype: float64
```

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

Out[ ]:
```
33479.53674501214
```

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

Out[ ]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_
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 [ ]:
```
df_rooms
```

Out[ ]:

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

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

Out[ ]:
```
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 [ ]:
```
23439 + 3 *9048
```

Out[ ]:
```
50583
```

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

Out[ ]:
```
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       77899
booking_status          0
revenue_generated       0
revenue_realized        0
dtype: int64
```

Total values in our datawfram is 134578. Out of that 77899 is the Null or NA rating. As we have the null value or null rating we are not goona remove that and not gonna replace that with mean median or mode

Exercise-1. In aggregate bookings find columns that have null values. Fill these null values with whatever you think is the appropriate subtitute (possible ways is to use mean or median)

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

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

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

Out[ ]:

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

In [ ]:
```
df_agg_bookings.loc[[8,15]]
```

Out[ ]:

	property_id	check_in_date	room_category	successful_bookings	capacity
8	17561	1-May-22	RT1	22	25.0
15	17563	1-May-22	RT1	21	25.0

Exercise-2. In aggregate bookings find out records that have successful_bookings value greater than capacity. Filter those records

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

Out[ ]:

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

==> Data Transformation

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

Out[ ]:

	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 [ ]:
```
df_agg_bookings["occ_pct"] = df_agg_bookings['successful_bookings']/df_agg_bookings
```

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

Out[ ]:

	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 [ ]:
```
df_agg_bookings["occ_pct"] = df_agg_bookings['occ_pct'].apply(lambda x : round (x*1
df_agg_bookings.head()
```

Out[ ]:

	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

---

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

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

Out[ ]:
```
room_category
RT1    58.232748
RT2    58.040278
RT3    58.028213
RT4    59.300461
Name: occ_pct, dtype: float64
```

In [ ]:
```
df_rooms
```

Out[ ]:

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

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

Out[ ]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	roon
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	

2. Print average occupancy rate per city

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

Out[ ]:

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

Out[ ]:

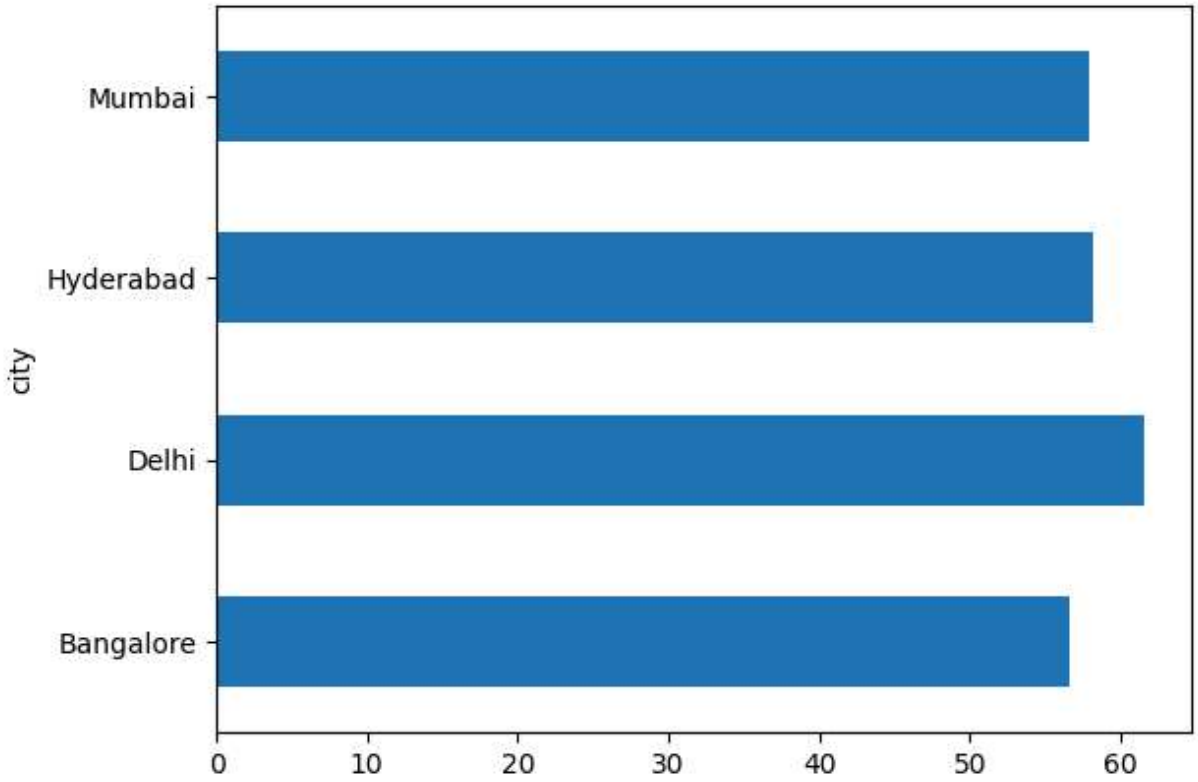
	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	roon
0	16559	1-May-22	RT1	25	30.0	83.33	
1	16559	2-May-22	RT1	20	30.0	66.67	
2	16559	3-May-22	RT1	17	30.0	56.67	
3	16559	4-May-22	RT1	21	30.0	70.00	
4	16559	5-May-22	RT1	16	30.0	53.33	

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

Out[ ]:
```
city
Bangalore    56.594207
Delhi        61.606467
Hyderabad    58.144651
Mumbai       57.943142
Name: occ_pct, dtype: float64
```

In [ ]:
```
df.groupby("city")["occ_pct"].mean().plot(kind = "barh")
```

Out[ ]:
```
<AxesSubplot:ylabel='city'>
```



3. When was the occupancy better? weekday or weekdays ?

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

Out[ ]:

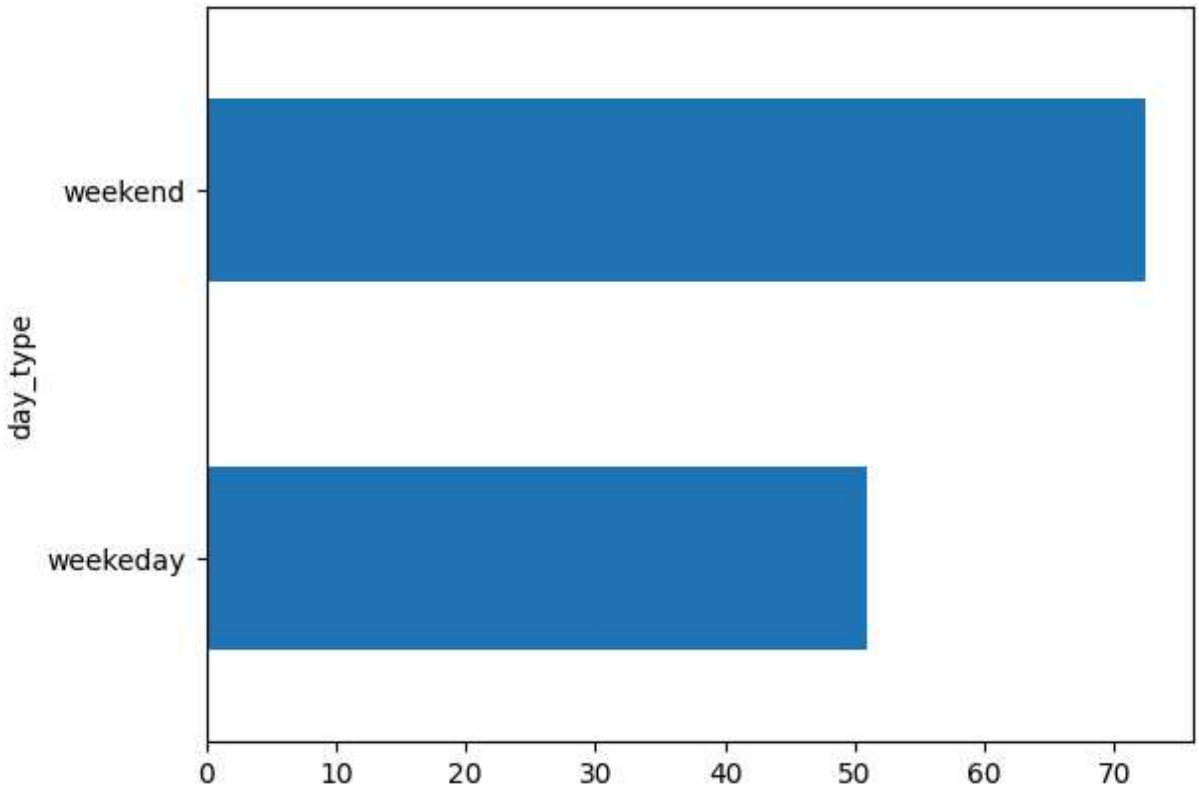
	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	roon
0	16559	10-May-22	RT1	18	30.0	60.00	
1	16559	10-May-22	RT2	25	41.0	60.98	
2	16559	10-May-22	RT3	20	32.0	62.50	
3	16559	10-May-22	RT4	13	18.0	72.22	
4	19562	10-May-22	RT1	18	30.0	60.00	

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

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

In [ ]:
```
df.groupby("day_type")["occ_pct"].mean().round(2).plot(kind = "barh")
```

Out[ ]:
```
<AxesSubplot:ylabel='day_type'>
```



4. In the month of june what was the occupancy rate for different cities?

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

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

In [ ]:
```
df_june_22 = df[df["mmm yy"] == "Jun 22"]
df_june_22.head(2)
```

Out[ ]:

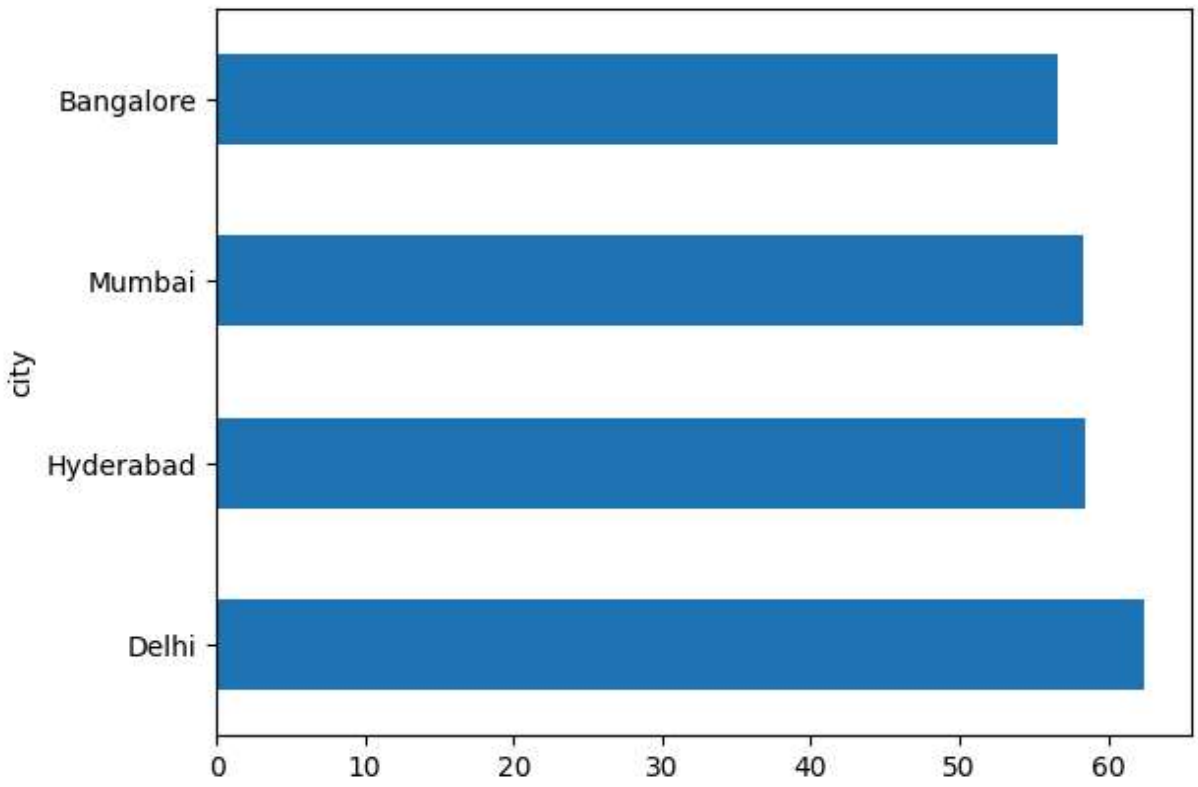
	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	r
2200	16559	10-Jun-22	RT1	20	30.0	66.67	
2201	16559	10-Jun-22	RT2	26	41.0	63.41	

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

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

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

Out[ ]:
```
<AxesSubplot:ylabel='city'>
```



In [ ]:
```
df_august = pd.read_csv("C:\\Users\\user\\Downloads\\Hotel dataset\\new_data_august
df_august.head()
```

Out[ ]:

	property_id	property_name	category	city	room_category	room_class	check_in_
0	16559	Atliq Exotica	Luxury	Mumbai	RT1	Standard	01-Au
1	19562	Atliq Bay	Luxury	Bangalore	RT1	Standard	01-Au
2	19563	Atliq Palace	Business	Bangalore	RT1	Standard	01-Au
3	19558	Atliq Grands	Luxury	Bangalore	RT1	Standard	01-Au
4	19560	Atliq City	Business	Bangalore	RT1	Standard	01-Au

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

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

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

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

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

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

Out[ ]:
```
(6500, 15)
```

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

Out[ ]:

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	r
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	

In [ ]:
```
latest_df.shape
```

Out[ ]:
```
(6507, 16)
```

5. print revenue realized per city

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

Out[ ]:

	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
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	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
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0

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

Out[ ]:

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

Out[ ]:

	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	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0
2	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0
3	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0
4	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0

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

Out[ ]:
```
city
Bangalore    420383550
Delhi        294438788
Hyderabad    325179310
Mumbai       668602231
Name: revenue_realized, dtype: int64
```

6. print revenue month by month

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

Out[ ]:

	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	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0
2	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0

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

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

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

Out[ ]:

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 134578 entries, 0 to 134577
Data columns (total 15 columns):
 #   Column             Non-Null Count   Dtype  
---  ------             --------------   -----  
 0   booking_id         134578 non-null  object 
 1   property_id        134578 non-null  int64  
 2   booking_date       134578 non-null  object 
 3   check_in_date      134578 non-null  object 
 4   checkout_date      134578 non-null  object 
 5   no_guests          134578 non-null  float64
 6   room_category      134578 non-null  object 
 7   booking_platform   134578 non-null  object 
 8   ratings_given      56679 non-null   float64
 9   booking_status     134578 non-null  object 
 10  revenue_generated  134578 non-null  int64  
 11  revenue_realized   134578 non-null  int64  
 12  property_name      134578 non-null  object 
 13  category           134578 non-null  object 
 14  city               134578 non-null  object 
dtypes: float64(2), int64(3), object(10)
memory usage: 16.4+ MB
```

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

Out[ ]:
```
0    2022-05-01
1    2022-05-02
2    2022-05-03
3    2022-05-04
4    2022-05-05
        ...    
87   2022-07-27
88   2022-07-28
89   2022-07-29
90   2022-07-30
91   2022-07-31
Name: date, Length: 92, dtype: datetime64[ns]
```

In [ ]:
```
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 [ ]:
```
df_bookings_all["check_in_date"] = pd.to_datetime(df_bookings_all["check_in_date"])
df_bookings_all["check_in_date"]
```

Out[ ]:
```
0        2022-01-05
1        2022-01-05
2        2022-01-05
3        2022-01-05
4        2022-01-05
            ...
134573   2022-07-31
134574   2022-07-31
134575   2022-07-31
134576   2022-07-31
134577   2022-07-31
Name: check_in_date, Length: 134578, dtype: datetime64[ns]
```

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 134578 entries, 0 to 134577
Data columns (total 15 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   booking_id         134578 non-null  object
 1   property_id        134578 non-null  int64
 2   booking_date       134578 non-null  object
 3   check_in_date      134578 non-null  datetime64[ns]
 4   checkout_date      134578 non-null  object
 5   no_guests          134578 non-null  float64
 6   room_category      134578 non-null  object
 7   booking_platform   134578 non-null  object
 8   ratings_given      56679 non-null   float64
 9   booking_status     134578 non-null  object
 10  revenue_generated  134578 non-null  int64
 11  revenue_realized   134578 non-null  int64
 12  property_name      134578 non-null  object
 13  category           134578 non-null  object
 14  city               134578 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(3), object(9)
memory usage: 16.4+ MB
```

In [ ]:
```
df_bookings_all = pd.merge(df_bookings_all,df_date, left_on= "check_in_date",right_
df_bookings_all.head(2)
```

Out[ ]:

	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

In [ ]:
```
df_bookings_all.columns
```

Out[ ]:
```
Index(['booking_id', 'property_id', 'booking_date', 'check_in_date',
       'checkout_date', 'no_guests', 'room_category', 'booking_platform',
       'ratings_given', 'booking_status', 'revenue_generated',
       'revenue_realized', 'property_name', 'category', 'city', 'date',
       'mmm yy', 'week no', 'day_type'],
      dtype='object')
```

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

Out[ ]:
```
mmm yy
Jul 22    389953512
Jun 22    377191229
May 22    408375641
Name: revenue_realized, dtype: int64
```

In [ ]:
```
df_bookings_rev_real_per_hotel_type = pd.merge(df_hotels,df_bookings,on ="property_
df_bookings_rev_real_per_hotel_type.head(2)
```

Out[ ]:

	property_id	property_name	category	city	booking_id	booking_date	check_i
0	16558	Atliq Grands	Luxury	Delhi	May012216558RT12	30-04-22	1/5
1	16558	Atliq Grands	Luxury	Delhi	May012216558RT13	28-04-22	1/5

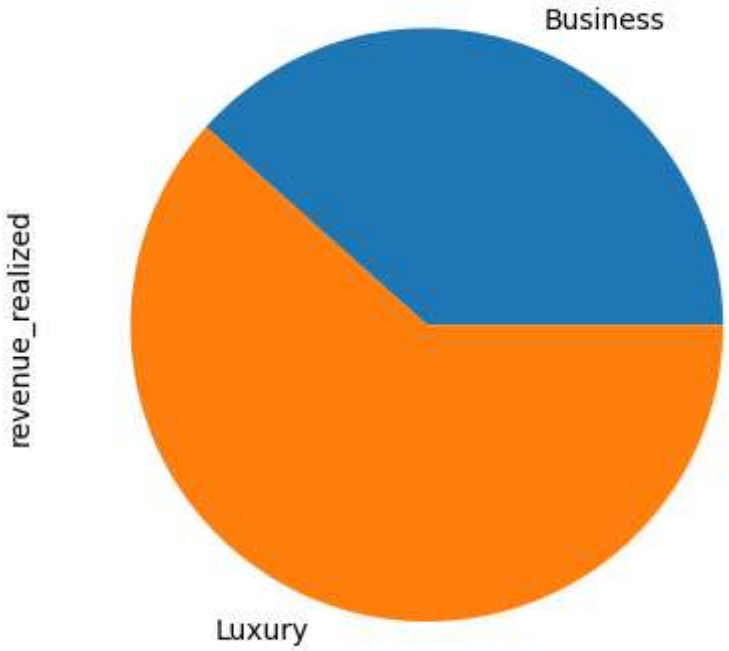
7. Print Revenue Realized per hotel type

In [ ]:
```
df_bookings_rev_real_per_hotel_type.groupby("category")["revenue_realized"].sum()
```

Out[ ]:
```
category
Business     655967037
Luxury      1052636842
Name: revenue_realized, dtype: int64
```

In [ ]:
```
df_bookings_rev_real_per_hotel_type.groupby("category")["revenue_realized"].sum().p
```

Out[ ]:
```
<AxesSubplot:ylabel='revenue_realized'>
```



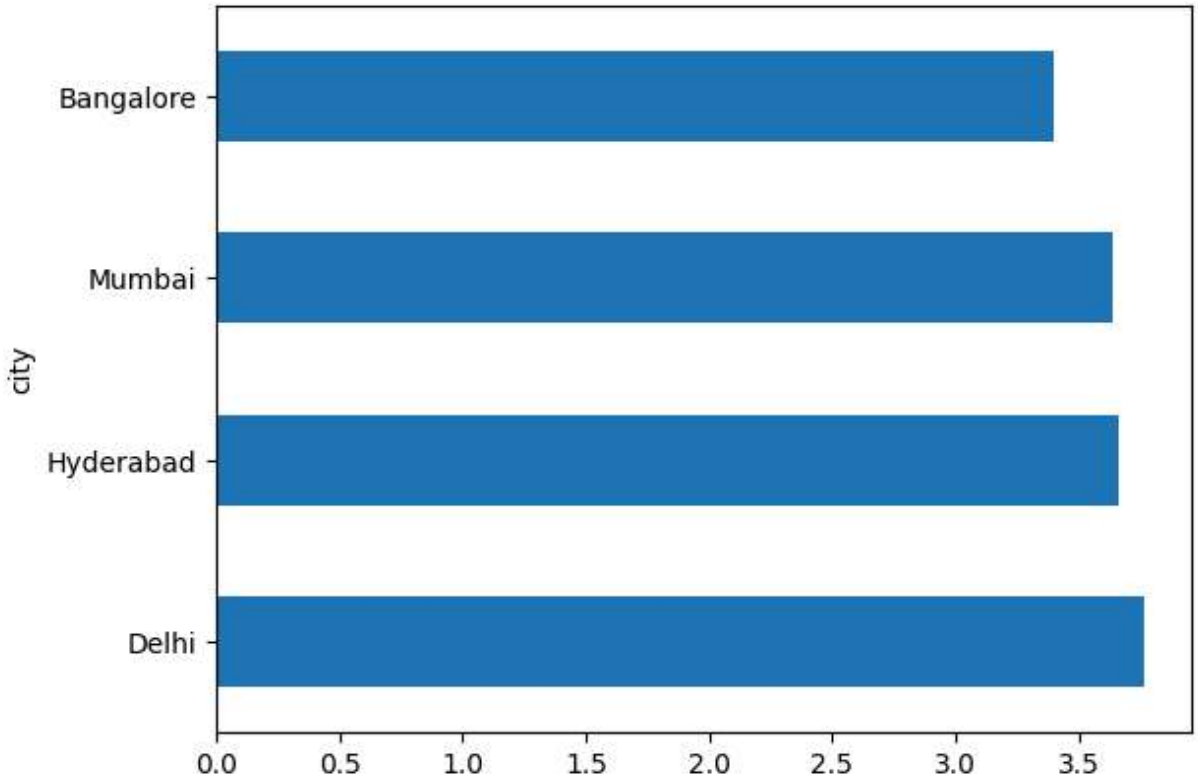
8. Print average rating per city

In [ ]:
```
df_bookings_all.groupby("city")["ratings_given"].mean().round(2).sort_values(ascend
```

Out[ ]:
```
city
Delhi        3.77
Hyderabad    3.66
Mumbai       3.64
Bangalore    3.40
Name: ratings_given, dtype: float64
```

In [ ]:
```
df_bookings_all.groupby("city")["ratings_given"].mean().round(2).sort_values(ascend
```

Out[ ]:
```
<AxesSubplot:ylabel='city'>
```



9. print a pie chart of revenue realized per booking platform

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

Out[ ]:
```
<AxesSubplot:ylabel='revenue_realized'>
```