

Data Exploration

```
import pandas as pd

df_bookings =
pd.read_csv("C:/Users/user/Downloads/64101194a2364/source-code/3_project_hospitality_analysis/datasets/fact_bookings.csv")
df_bookings.head(4)
```

	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

	no_guests	room_category	booking_platform	ratings_given	booking_status
0	-3.0	RT1	direct online	1.0	Checked Out
1	2.0	RT1	others	NaN	Cancelled
2	2.0	RT1	logtrip	5.0	Checked Out
3	-2.0	RT1	others	NaN	Cancelled

	revenue_generated	revenue_realized
0	10010	10010
1	9100	3640
2	9100000	9100
3	9100	3640

```
df_bookings.shape
```

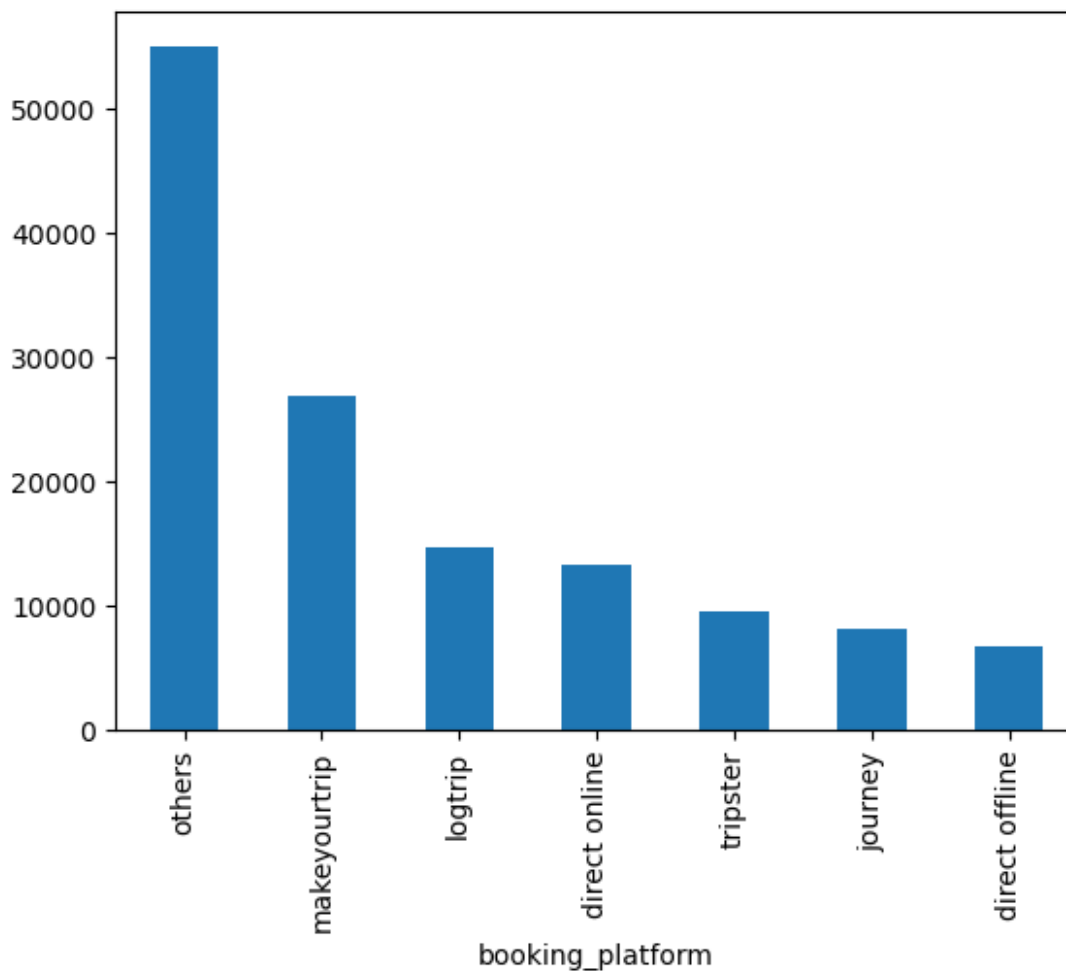
```
(134590, 12)
```

```
df_bookings.room_category.unique()
```

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

```
df_bookings.booking_platform.unique()
```

```
array(['direct online', 'others', 'logtrip', 'tripster',  
      'makeyourtrip',  
      'journey', 'direct offline'], dtype=object)  
  
df_bookings.booking_platform.value_counts()  
  
booking_platform  
others          55066  
makeyourtrip    26898  
logtrip         14756  
direct online   13379  
tripster        9630  
journey         8106  
direct offline  6755  
Name: count, dtype: int64  
  
df_bookings.booking_platform.value_counts().plot(kind = "bar")  
<Axes: xlabel='booking_platform'>
```



```
df_bookings.describe()
```

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

	revenue_realized
count	134590.000000
mean	12696.123256
std	6928.108124
min	2600.000000
25%	7600.000000
50%	11700.000000
75%	15300.000000
max	45220.000000

```
df_date = pd.read_csv(r"C:\Users\user\Downloads\64101194a2364\source-  
code\3_project_hospitality_analysis\datasets\dim_date.csv")  
df_hotels = pd.read_csv(r"C:\Users\user\Downloads\64101194a2364\  
source-code\3_project_hospitality_analysis\datasets\dim_hotels.csv")  
df_rooms = pd.read_csv(r"C:\Users\user\Downloads\64101194a2364\source-  
code\3_project_hospitality_analysis\datasets\dim_rooms.csv")  
df_agg_bookings = pd.read_csv(r"C:\Users\user\Downloads\64101194a2364\  
source-code\3_project_hospitality_analysis\datasets\  
fact_aggregated_bookings.csv")
```

```
df_date.shape
```

```
(92, 4)
```

```
df_hotels.shape
```

```
(25, 4)
```

```
df_rooms.shape
```

```
(4, 2)
```

```
df_agg_bookings.shape
```

```
(9200, 5)
```

```
df_hotels.head(4)
```

	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

```
df_hotels.category.value_counts()
```

```
category
```

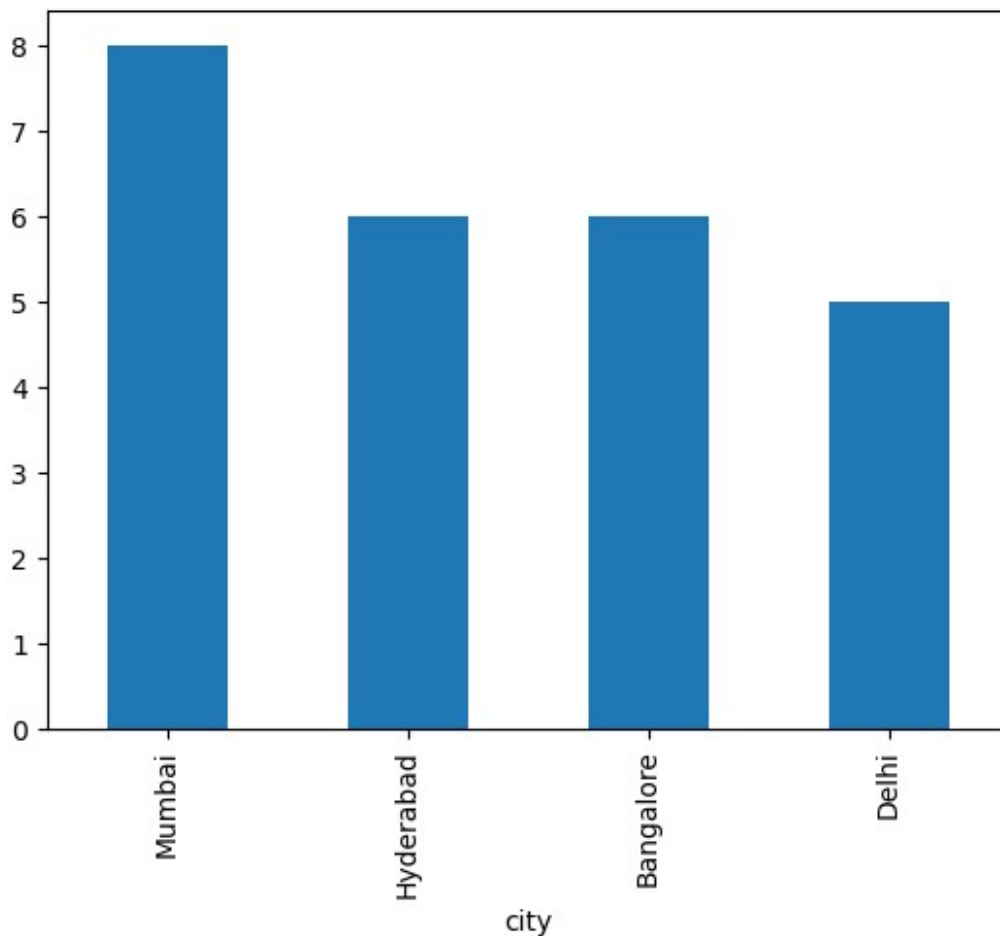
```
Luxury      16
```

```
Business     9
```

```
Name: count, dtype: int64
```

```
df_hotels.city.value_counts().plot(kind = "bar")
```

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



```
df_agg_bookings.head(4)
```

	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				

1.find out unique property id's in aggregate booking dataset

```
df_agg_bookings.property_id.unique()
```

```
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)
```

2.find out total bookings per property id.

```
df_agg_bookings.property_id.value_counts()
```

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

```

3. find out days on which bookings are greater than the capacity

```

over_capacity = df_agg_bookings[df_agg_bookings['successful_bookings']
> df_agg_bookings['capacity']]
over_capacity

```

	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				

4.find out the property which has highest capacity.

```

max_capacity_row =
df_agg_bookings.loc[df_agg_bookings['capacity'].idxmax()]
max_capacity_row

```

```

property_id      17558
check_in_date    1-May-22
room_category    RT2
successful_bookings    38
capacity         50.0
Name: 27, dtype: object

```

Data Cleaning

```
df_bookings.describe()
```

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

	revenue_realized
count	134590.000000
mean	12696.123256
std	6928.108124
min	2600.000000
25%	7600.000000
50%	11700.000000
75%	15300.000000
max	45220.000000

identify the negative values from our data which can drastically impact our analysis.

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

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

	checkout_date	no_guests	room_category	booking_platform
ratings_given	\			

0	2/5/2022	-3.0	RT1	direct online
1.0				
3	2/5/2022	-2.0	RT1	others
NaN				
17924	14-05-22	-10.0	RT4	direct online
NaN				
18020	14-05-22	-12.0	RT2	makeyourtrip
NaN				
18119	17-05-22	-6.0	RT3	direct offline
5.0				
18121	17-05-22	-4.0	RT3	direct online
NaN				
56715	13-06-22	-17.0	RT1	others
NaN				
119765	22-07-22	-1.0	RT2	others
NaN				
134586	1/8/2022	-4.0	RT4	logtrip
2.0				

	booking_status	revenue_generated	revenue_realized
0	Checked Out	10010	10010
3	Cancelled	9100	3640
17924	No Show	20900	20900
18020	Cancelled	9000	3600
18119	Checked Out	16800	16800
18121	Cancelled	14400	5760
56715	Checked Out	6500	6500
119765	Checked Out	13500	13500
134586	Checked Out	38760	38760

```
df_bookings.shape
```

```
(134590, 12)
```

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

```
df_bookings.shape
```

```
(134578, 12)
```

in these above code we have stored a positive values in the same file df_bookings which is (134578, 12) in order to do analysis only on positive data's

```
df_bookings.head(4)
```

	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				

	no_guests	room_category	booking_platform	ratings_given	
booking_status \					
1	2.0	RT1	others	NaN	
Cancelled					
2	2.0	RT1	logtrip	5.0	Checked
Out					
4	4.0	RT1	direct online	5.0	Checked
Out					
5	2.0	RT1	others	4.0	Checked
Out					

	revenue_generated	revenue_realized
1	9100	3640
2	9100000	9100
4	10920	10920
5	9100	9100

```
df_bookings.revenue_generated.min(),df_bookings.revenue_generated.max(
)
```

```
(6500, 285600000)
```

in this revenue generated column has inappropriate values where 285600000 amount can't anyone pay for a single night. so need to rectify this.

```
avg,std =
df_bookings.revenue_generated.mean(),df_bookings.revenue_generated.std
()
```

```
avg,std
```

```
(15378.036937686695, 93040.15493143328)
```

```
higher_limit = avg+3*std
higher_limit
```

```
294498.50173198653
```

```
lower_limit = avg-3*std
lower_limit
```

```
-263742.4278566132
```

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

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

	checkout_date	no_guests	room_category	booking_platform	ratings_given	\
2	4/5/2022	2.0	RT1	logtrip	5.0	
111	2/5/2022	6.0	RT3	direct online	NaN	
315	4/5/2022	2.0	RT2	direct offline	3.0	
562	2/5/2022	2.0	RT1	others	NaN	
129176	29-07-22	2.0	RT2	direct online	3.0	

	booking_status	revenue_generated	revenue_realized
2	Checked Out	9100000	9100
111	Checked Out	28560000	28560
315	Checked Out	12600000	12600
562	Cancelled	2000000	4420
129176	Checked Out	10000000	12600

we found the insane values in revenue generated column which we have to fix so what we can do a reverse we can see or store the only values which are less than the Higher_limit.

```
df_bookings = df_bookings[df_bookings.revenue_generated<higher_limit]
df_bookings.shape
```

```
(134573, 12)
```

```
df_bookings.shape
```

```
(134573, 12)
```

```
df_bookings.isnull().sum()
```

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

```
df_agg_bookings.head()
```

	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				

```
# to add additional column as "Occupancy_pct"
```

```
df_agg_bookings["OCC_Pct"] = df_agg_bookings["successful_bookings"] /  
df_agg_bookings["capacity"]
```

```
df_agg_bookings.head()
```

	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				

	OCC_Pct
0	0.833333
1	0.933333
2	0.766667
3	1.578947
4	0.947368

```
# to conver it into the exact percentae we will use apply function
with lambda, round function
```

```
df_agg_bookings["OCC_Pct"] = df_agg_bookings["OCC_Pct"].apply(lambda
x: round(x*100,2))
df_agg_bookings.head()
```

	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				

	OCC_Pct
0	83.33
1	93.33
2	76.67
3	157.89
4	94.74

Insights Generation

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

```
df_agg_bookings
```

	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				
...
...				
9195	16563	31-Jul-22	RT4	13

18.0				
9196	16559	31-Jul-22	RT4	13
18.0				
9197	17558	31-Jul-22	RT4	3
6.0				
9198	19563	31-Jul-22	RT4	3
6.0				
9199	17561	31-Jul-22	RT4	3
4.0				

	OCC_Pct
0	83.33
1	93.33
2	76.67
3	157.89
4	94.74
...	...
9195	72.22
9196	72.22
9197	50.00
9198	50.00
9199	75.00

[9200 rows x 6 columns]

```
df_agg_bookings.groupby("room_category")["OCC_Pct"].mean()
```

```
room_category
RT1    58.224247
RT2    58.040278
RT3    58.028213
RT4    59.300461
Name: OCC_Pct, dtype: float64
```

#I don't understand RT1, RT2 etc. Print room categories such as Standard, Premium, Elite etc along with average occupancy percentage

```
df = pd.merge(df_agg_bookings, df_rooms, left_on = "room_category",
right_on = "room_id")
df.head(4)
```

	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				

	OCC_Pct	room_id	room_class
0	83.33	RT1	Standard
1	93.33	RT1	Standard
2	76.67	RT1	Standard
3	157.89	RT1	Standard

```
df.groupby("room_class")["OCC_Pct"].mean()
```

room_class	
Elite	58.040278
Premium	58.028213
Presidential	59.300461
Standard	58.224247

Name: OCC_Pct, dtype: float64

```
df[df.room_class=="Standard"].OCC_Pct.mean()
```

58.22424717145344

#2. Print average occupancy rate per city

```
df_agg_bookings
```

	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				
...
...				
9195	16563	31-Jul-22	RT4	13
18.0				
9196	16559	31-Jul-22	RT4	13
18.0				
9197	17558	31-Jul-22	RT4	3
6.0				
9198	19563	31-Jul-22	RT4	3
6.0				
9199	17561	31-Jul-22	RT4	3
4.0				

OCC_Pct

```

0      83.33
1      93.33
2      76.67
3     157.89
4      94.74
...
9195    72.22
9196    72.22
9197    50.00
9198    50.00
9199    75.00

```

```
[9200 rows x 6 columns]
```

```
df
```

```

      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
...          ...          ...          ...          ...
...
9195          16563      31-Jul-22          RT4                   13
18.0
9196          16559      31-Jul-22          RT4                   13
18.0
9197          17558      31-Jul-22          RT4                    3
6.0
9198          19563      31-Jul-22          RT4                    3
6.0
9199          17561      31-Jul-22          RT4                    3
4.0

```

```

      OCC_Pct room_id  room_class
0      83.33    RT1    Standard
1      93.33    RT1    Standard
2      76.67    RT1    Standard
3     157.89    RT1    Standard
4      94.74    RT1    Standard
...
9195    72.22    RT4  Presidential
9196    72.22    RT4  Presidential

```

```

9197    50.00    RT4  Presidential
9198    50.00    RT4  Presidential
9199    75.00    RT4  Presidential

```

```
[9200 rows x 8 columns]
```

```

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

```

```

   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

```

```

   OCC_Pct  room_id  room_class  property_name  category  city
0    83.33    RT1    Standard  Atliq Exotica    Luxury  Mumbai
1    93.33    RT1    Standard    Atliq Bay    Luxury  Bangalore
2    76.67    RT1    Standard  Atliq Palace  Business  Bangalore

```

```
df.groupby("city")["OCC_Pct"].mean()
```

```

city
Bangalore    56.594207
Delhi        61.606467
Hyderabad    58.144651
Mumbai       57.936305
Name: OCC_Pct, dtype: float64

```

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

```

df_date

```

	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
...
87	27-Jul-22	Jul	22	W	31	weekeday
88	28-Jul-22	Jul	22	W	31	weekeday
89	29-Jul-22	Jul	22	W	31	weekeday
90	30-Jul-22	Jul	22	W	31	weekend
91	31-Jul-22	Jul	22	W	32	weekend

```
[92 rows x 4 columns]
```



```
df=pd.merge(df, df_date, left_on = "check_in_date", right_on = "date")
df.head(3)
```

	property_id	check_in_date	room_category	successful_bookings
capacity \				
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				

	OCC_Pct	room_id	room_class	property_name	category	city
date \						
0	51.72	RT3	Premium	Atliq Palace	Business	Bangalore
10-May-22						
1	63.33	RT1	Standard	Atliq City	Business	Hyderabad
10-May-22						
2	60.00	RT1	Standard	Atliq Bay	Luxury	Bangalore
10-May-22						

	mmm	yy	week	no	day_type
0	May	22	W	20	weekeday
1	May	22	W	20	weekeday
2	May	22	W	20	weekeday

```
df.groupby("day_type")["OCC_Pct"].mean().round(2)
```

```
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

```
df
```

	property_id	check_in_date	room_category	successful_bookings
capacity \				
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				
3	19563	10-May-22	RT1	16
30.0				
4	17558	10-May-22	RT1	11
19.0				
...

...					
6495	16563	31-Jul-22	RT4		13
18.0					
6496	16559	31-Jul-22	RT4		13
18.0					
6497	17558	31-Jul-22	RT4		3
6.0					
6498	19563	31-Jul-22	RT4		3
6.0					
6499	17561	31-Jul-22	RT4		3
4.0					

	OCC_Pct	room_id	room_class	property_name	category
city \					
0	51.72	RT3	Premium	Atliq Palace	Business
Bangalore					
1	63.33	RT1	Standard	Atliq City	Business
Hyderabad					
2	60.00	RT1	Standard	Atliq Bay	Luxury
Bangalore					
3	53.33	RT1	Standard	Atliq Palace	Business
Bangalore					
4	57.89	RT1	Standard	Atliq Grands	Luxury
Mumbai					

...
.						
6495	72.22	RT4	Presidential	Atliq Palace	Business	
Delhi						
6496	72.22	RT4	Presidential	Atliq Exotica	Luxury	
Mumbai						
6497	50.00	RT4	Presidential	Atliq Grands	Luxury	
Mumbai						
6498	50.00	RT4	Presidential	Atliq Palace	Business	
Bangalore						
6499	75.00	RT4	Presidential	Atliq Blu	Luxury	
Mumbai						

	date	mmm	yy	week	no	day_type
0	10-May-22	May	22	W	20	weekeday
1	10-May-22	May	22	W	20	weekeday
2	10-May-22	May	22	W	20	weekeday
3	10-May-22	May	22	W	20	weekeday
4	10-May-22	May	22	W	20	weekeday
...	...					
6495	31-Jul-22	Jul	22	W	32	weekend
6496	31-Jul-22	Jul	22	W	32	weekend
6497	31-Jul-22	Jul	22	W	32	weekend
6498	31-Jul-22	Jul	22	W	32	weekend
6499	31-Jul-22	Jul	22	W	32	weekend

```
[6500 rows x 15 columns]
```

```
df_june_22 = df[df["mmm yy"]=="Jun 22"]  
df_june_22.head(4)
```

	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				

	OCC_Pct	room_id	room_class	property_name	category
city \					
2200	66.67	RT1	Standard	Atliq Exotica	Luxury Mumbai
2201	63.33	RT1	Standard	Atliq Bay	Luxury Bangalore
2202	56.67	RT1	Standard	Atliq Palace	Business Bangalore
2203	47.37	RT1	Standard	Atliq Grands	Luxury Mumbai

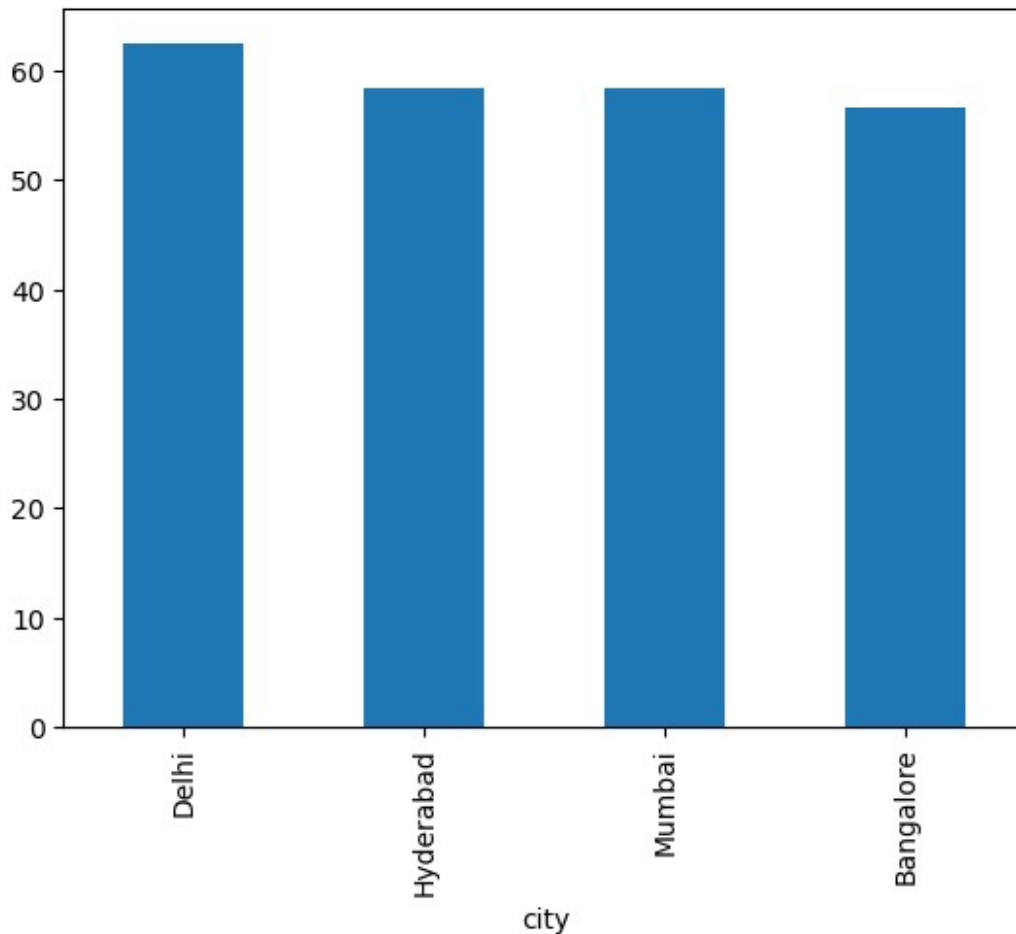
	date	mmm	yy	week	no	day_type
2200	10-Jun-22	Jun	22	W	24	weekeday
2201	10-Jun-22	Jun	22	W	24	weekeday
2202	10-Jun-22	Jun	22	W	24	weekeday
2203	10-Jun-22	Jun	22	W	24	weekeday

```
df_june_22.groupby('city')  
['OCC_Pct'].mean().round(2).sort_values(ascending=False)
```

```
city  
Delhi      62.47  
Hyderabad  58.46  
Mumbai     58.38  
Bangalore  56.58  
Name: OCC_Pct, dtype: float64
```

```
df_june_22.groupby('city')  
['OCC_Pct'].mean().round(2).sort_values(ascending=False).plot(kind="bar")
```

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



#5: We got new data for the month of august. Append that to existing data

```
df_august = pd.read_csv(r"C:\Users\user\Downloads\64101194a2364\
source-code\3_project_hospitality_analysis\datasets\
new_data_august.csv")
df_august.head(4)
```

	property_id	property_name	category	city	room_category
0	16559	Atliq Exotica	Luxury	Mumbai	RT1
1	19562	Atliq Bay	Luxury	Bangalore	RT1
2	19563	Atliq Palace	Business	Bangalore	RT1
3	19558	Atliq Grands	Luxury	Bangalore	RT1

	check_in_date	mmm yy	week no	day_type	successful_bookings
0	01-Aug-22	Aug-22	W 32	weekday	30

```

30
1      01-Aug-22  Aug-22    W 32  weekday          21
30
2      01-Aug-22  Aug-22    W 32  weekday          23
30
3      01-Aug-22  Aug-22    W 32  weekday          30
40

```

```

      occ%
0  100.00
1   70.00
2   76.67
3   75.00

```

```
df_august.columns
```

```

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')

```

```
df_august.shape
```

```
(7, 13)
```

```
df.shape
```

```
(6500, 15)
```

```

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

```

```

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

```

6505	17561	01-Aug-22	RT1	18
26.0				
6506	17564	01-Aug-22	RT1	10
16.0				

	OCC_Pct	room_id	room_class	property_name	category
city \					
6497	50.0	RT4	Presidential	Atliq Grands	Luxury
Mumbai					
6498	50.0	RT4	Presidential	Atliq Palace	Business
Bangalore					
6499	75.0	RT4	Presidential	Atliq Blu	Luxury
Mumbai					
6500	NaN	NaN	Standard	Atliq Exotica	Luxury
Mumbai					
6501	NaN	NaN	Standard	Atliq Bay	Luxury
Bangalore					
6502	NaN	NaN	Standard	Atliq Palace	Business
Bangalore					
6503	NaN	NaN	Standard	Atliq Grands	Luxury
Bangalore					
6504	NaN	NaN	Standard	Atliq City	Business
Bangalore					
6505	NaN	NaN	Standard	Atliq Blu	Luxury
Mumbai					
6506	NaN	NaN	Standard	Atliq Seasons	Business
Mumbai					

	date	mmm	yy	week	no	day_type	occ%
6497	31-Jul-22	Jul	22	W	32	weekend	NaN
6498	31-Jul-22	Jul	22	W	32	weekend	NaN
6499	31-Jul-22	Jul	22	W	32	weekend	NaN
6500	NaN	Aug	-22	W	32	weekeday	100.00
6501	NaN	Aug	-22	W	32	weekeday	70.00
6502	NaN	Aug	-22	W	32	weekeday	76.67
6503	NaN	Aug	-22	W	32	weekeday	75.00
6504	NaN	Aug	-22	W	32	weekeday	76.92
6505	NaN	Aug	-22	W	32	weekeday	69.23
6506	NaN	Aug	-22	W	32	weekeday	62.50

latest_df.shape

(6507, 16)

#6. Print revenue realized per city

df_bookings

	booking_id	property_id	booking_date	check_in_date	\
1	May012216558RT12	16558	30-04-22	1/5/2022	

4	May012216558RT15	16558	27-04-22	1/5/2022
5	May012216558RT16	16558	1/5/2022	1/5/2022
6	May012216558RT17	16558	28-04-22	1/5/2022
7	May012216558RT18	16558	26-04-22	1/5/2022
...
134584	Jul312217564RT45	17564	30-07-22	31-07-22
134585	Jul312217564RT46	17564	29-07-22	31-07-22
134587	Jul312217564RT48	17564	30-07-22	31-07-22
134588	Jul312217564RT49	17564	29-07-22	31-07-22
134589	Jul312217564RT410	17564	31-07-22	31-07-22

	checkout_date	no_guests	room_category	booking_platform
ratings_given \				
1	2/5/2022	2.0	RT1	others
NaN				
4	2/5/2022	4.0	RT1	direct online
5.0				
5	3/5/2022	2.0	RT1	others
4.0				
6	6/5/2022	2.0	RT1	others
NaN				
7	3/5/2022	2.0	RT1	logtrip
NaN				
...
...				
134584	1/8/2022	2.0	RT4	others
2.0				
134585	3/8/2022	1.0	RT4	makeyourtrip
2.0				
134587	2/8/2022	1.0	RT4	tripster
NaN				
134588	1/8/2022	2.0	RT4	logtrip
2.0				
134589	1/8/2022	2.0	RT4	makeyourtrip
NaN				

	booking_status	revenue_generated	revenue_realized
1	Cancelled	9100	3640
4	Checked Out	10920	10920
5	Checked Out	9100	9100
6	Cancelled	9100	3640
7	No Show	9100	9100
...
134584	Checked Out	32300	32300
134585	Checked Out	32300	32300
134587	Cancelled	32300	12920
134588	Checked Out	32300	32300
134589	Cancelled	32300	12920

[134573 rows x 12 columns]

```
df_hotels
```

	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
5	16563	Atliq Palace	Business	Delhi
6	17558	Atliq Grands	Luxury	Mumbai
7	17559	Atliq Exotica	Luxury	Mumbai
8	17560	Atliq City	Business	Mumbai
9	17561	Atliq Blu	Luxury	Mumbai
10	17562	Atliq Bay	Luxury	Mumbai
11	17563	Atliq Palace	Business	Mumbai
12	18558	Atliq Grands	Luxury	Hyderabad
13	18559	Atliq Exotica	Luxury	Hyderabad
14	18560	Atliq City	Business	Hyderabad
15	18561	Atliq Blu	Luxury	Hyderabad
16	18562	Atliq Bay	Luxury	Hyderabad
17	18563	Atliq Palace	Business	Hyderabad
18	19558	Atliq Grands	Luxury	Bangalore
19	19559	Atliq Exotica	Luxury	Bangalore
20	19560	Atliq City	Business	Bangalore
21	19561	Atliq Blu	Luxury	Bangalore
22	19562	Atliq Bay	Luxury	Bangalore
23	19563	Atliq Palace	Business	Bangalore
24	17564	Atliq Seasons	Business	Mumbai

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

	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				
3	May012216558RT17	16558	28-04-22	1/5/2022
6/5/2022				
4	May012216558RT18	16558	26-04-22	1/5/2022
3/5/2022				

	no_guests	room_category	booking_platform	ratings_given
booking_status \				
0	2.0	RT1	others	NaN
Cancelled				
1	4.0	RT1	direct online	5.0
Checked				


```

Out
2      2.0      RT1      others      4.0      Checked
Out
3      2.0      RT1      others      NaN
Cancelled
4      2.0      RT1      logtrip      NaN      No
Show

```

	revenue_generated	revenue_realized	property_name	category	city
0	9100	3640	Atliq Grands	Luxury	Delhi
1	10920	10920	Atliq Grands	Luxury	Delhi
2	9100	9100	Atliq Grands	Luxury	Delhi
3	9100	3640	Atliq Grands	Luxury	Delhi
4	9100	9100	Atliq Grands	Luxury	Delhi

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

```

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

```