

```
import pandas as pd
```

```
df_bookings = pd.read_csv("datasets/fact_bookings.csv")
```

```
df_bookings.head()
```

	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
4	May012216558RT15	16558	27-04-22	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
4	4.0	RT1	direct online	5.0	Checked Out

	revenue_generated	revenue_realized
0	10010	10010
1	9100	3640
2	9100000	9100
3	9100	3640
4	10920	10920

```
df_bookings.shape
```

```
(134590, 12)
```

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

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

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

```

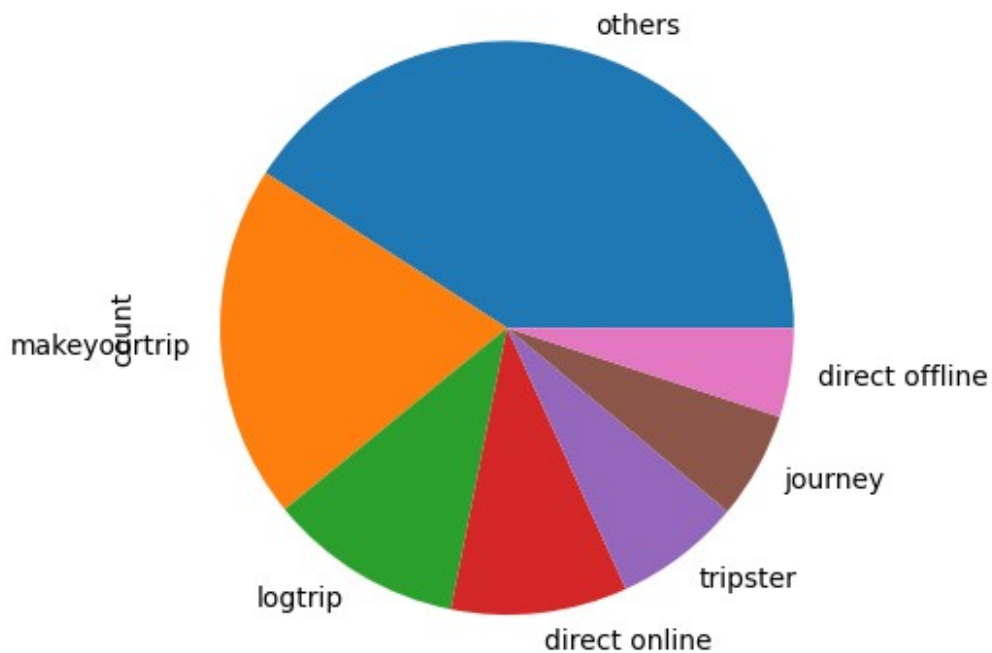
array(['direct online', 'others', 'logtrip', 'tripster',
      'makeyourtrip',
      'journey', 'direct offline'], dtype=object)

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="pie")
<Axes: ylabel='count'>

```



```

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

(6500, 28560000)
```

```
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_aggregated_bookings =
pd.read_csv("datasets/fact_aggregated_bookings.csv")
```

```
df_hotels.shape
```

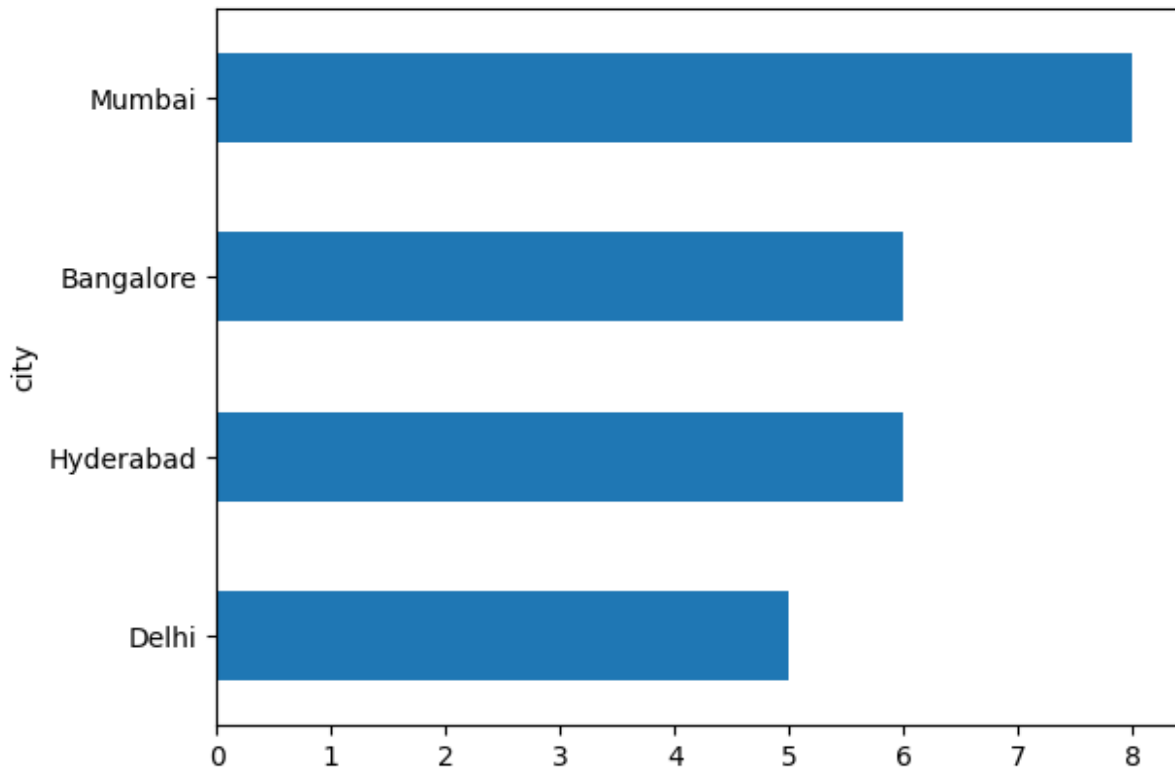
```
(25, 4)
```

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

```
category
Luxury      16
Business     9
Name: count, dtype: int64
```

```
df_hotels.city.value_counts().sort_values().plot(kind="barh")
```

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



```
df_aggregated_bookings.head()
```

	property_id	check_in_date	room_category	successful_bookings
0	16559	1-May-22	RT1	25
1	19562	1-May-22	RT1	28
2	19563	1-May-22	RT1	23
3	17558	1-May-22	RT1	30
4	16558	1-May-22	RT1	18

```
df_aggregated_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)
```

Total Booking per Property ID

```
df_aggregated_bookings.groupby("property_id")
["successful_bookings"].sum()
```

```

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

```

Booking is greater than capacity

```
df_aggregated_bookings[df_aggregated_bookings.successful_bookings>df_aggregated_bookings.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				

```
## Property with highest capacity
```

```
df_aggregated_bookings.capacity.max()
```

```
50.0
```

Data Cleaning

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

		booking_id	property_id	booking_date	check_in_date	\
0	May01	2216558RT11	16558	27-04-22	1/5/2022	
3	May01	2216558RT14	16558	28-04-22	1/5/2022	
17924	May12	2218559RT44	18559	12/5/2022	12/5/2022	
18020	May12	2218561RT22	18561	8/5/2022	12/5/2022	
18119	May12	2218562RT311	18562	5/5/2022	12/5/2022	
18121	May12	2218562RT313	18562	10/5/2022	12/5/2022	
56715	Jun08	2218562RT12	18562	5/6/2022	8/6/2022	
119765	Jul20	2219560RT220	19560	19-07-22	20-07-22	
134586	Jul31	2217564RT47	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)

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

(6500, 28560000)

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

avg, std
(15378.036937686695, 93040.1549314641)

higher_limit = avg + 3*std
higher_limit
294498.50173207896

lower_limit = avg - 3*std
lower_limit
-263742.4278567056

df_bookings[df_bookings.revenue_generated<0]

Empty DataFrame
Columns: [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]
Index: []

df_bookings[df_bookings.revenue_generated>0]

```

	booking_id	property_id	booking_date	check_in_date	\
1	May012216558RT12	16558	30-04-22	1/5/2022	
2	May012216558RT13	16558	28-04-22	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	
...
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
1	2/5/2022	2.0	RT1	others
NaN				
2	4/5/2022	2.0	RT1	logtrip
5.0				
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				
...
...				
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
2	Checked Out	9100000	9100
4	Checked Out	10920	10920
5	Checked Out	9100	9100
6	Cancelled	9100	3640
...
134584	Checked Out	32300	32300
134585	Checked Out	32300	32300
134587	Cancelled	32300	12920
134588	Checked Out	32300	32300
134589	Cancelled	32300	12920

[134578 rows x 12 columns]

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

(134573, 12)

```
df_bookings.revenue_realized.describe()
```

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
```

```
higher_limit = df_bookings.revenue_realized.mean() +
3*df_bookings.revenue_realized.std()
higher_limit
```

```
33479.358661845814
```

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

	booking_id	property_id	booking_date	check_in_date	\
137	May012216559RT41	16559	27-04-22	1/5/2022	
139	May012216559RT43	16559	1/5/2022	1/5/2022	
143	May012216559RT47	16559	28-04-22	1/5/2022	
149	May012216559RT413	16559	24-04-22	1/5/2022	
222	May012216560RT45	16560	30-04-22	1/5/2022	
...	
134328	Jul312219560RT49	19560	31-07-22	31-07-22	
134331	Jul312219560RT412	19560	31-07-22	31-07-22	
134467	Jul312219562RT45	19562	28-07-22	31-07-22	
134474	Jul312219562RT412	19562	25-07-22	31-07-22	
134581	Jul312217564RT42	17564	31-07-22	31-07-22	

	checkout_date	no_guests	room_category	booking_platform	ratings_given \
137	7/5/2022	4.0	RT4	others	
NaN					
139	2/5/2022	6.0	RT4	tripster	
3.0					
143	3/5/2022	3.0	RT4	others	
5.0					
149	7/5/2022	5.0	RT4	logtrip	
NaN					
222	3/5/2022	5.0	RT4	others	
3.0					
...	
...					
134328	2/8/2022	6.0	RT4	direct online	
5.0					
134331	1/8/2022	6.0	RT4	others	
2.0					
134467	1/8/2022	6.0	RT4	makeyourtrip	
4.0					
134474	6/8/2022	5.0	RT4	direct offline	

```
5.0
134581      1/8/2022      4.0      RT4      makeyourtrip
4.0
```

	booking_status	revenue_generated	revenue_realized
137	Checked Out	38760	38760
139	Checked Out	45220	45220
143	Checked Out	35530	35530
149	Checked Out	41990	41990
222	Checked Out	34580	34580
...
134328	Checked Out	39900	39900
134331	Checked Out	39900	39900
134467	Checked Out	39900	39900
134474	Checked Out	37050	37050
134581	Checked Out	38760	38760

```
[1299 rows x 12 columns]
```

```
df_rooms
```

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

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

```
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
```

```
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 Transforming

```
df_aggregated_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				

```
df_aggregated_bookings["occ_pct"]=
df_aggregated_bookings["successful_bookings"]/df_aggregated_bookings["
capacity"]
```

```
df_aggregated_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

```
df_aggregated_bookings["occ_pct"]
=df_aggregated_bookings["occ_pct"].apply(lambda x: round(x*100,2))
df_aggregated_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

##Ad Hoc Analysis (Insights)

What is an average occupancy rate in each Room category ?

```
df_aggregated_bookings.groupby("room_category")
["occ_pct"].mean().round(2)
```

room_category	occ_pct
RT1	58.22
RT2	58.04
RT3	58.03
RT4	59.30

Name: occ_pct, dtype: float64

Mearging df_room and df_aggregated_bookings

```
df = pd.merge(df_aggregated_bookings, df_rooms,
left_on="room_category", right_on="room_id")
df.tail(5)
```

	property_id	check_in_date	room_category	successful_bookings
capacity \				
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
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

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

room_class	
Elite	58.04
Premium	58.03
Presidential	59.30
Standard	58.22

Name: occ_pct, dtype: float64

Dropping a column

```
df.drop("room_id", axis=1, inplace=True)
df.head(5)
```

	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	room_class
0	83.33	Standard
1	93.33	Standard
2	76.67	Standard
3	157.89	Standard
4	94.74	Standard

```
df_hotels.head(5)
```

	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

Merging df_hotels with above df table

```
df = pd.merge(df,df_hotels, on="property_id")
df.head(5)
```

	property_id	check_in_date	room_category	successful_bookings
0	16559	1-May-22	RT1	25
1	19562	1-May-22	RT1	28
2	19563	1-May-22	RT1	23
3	17558	1-May-22	RT1	30
4	16558	1-May-22	RT1	18

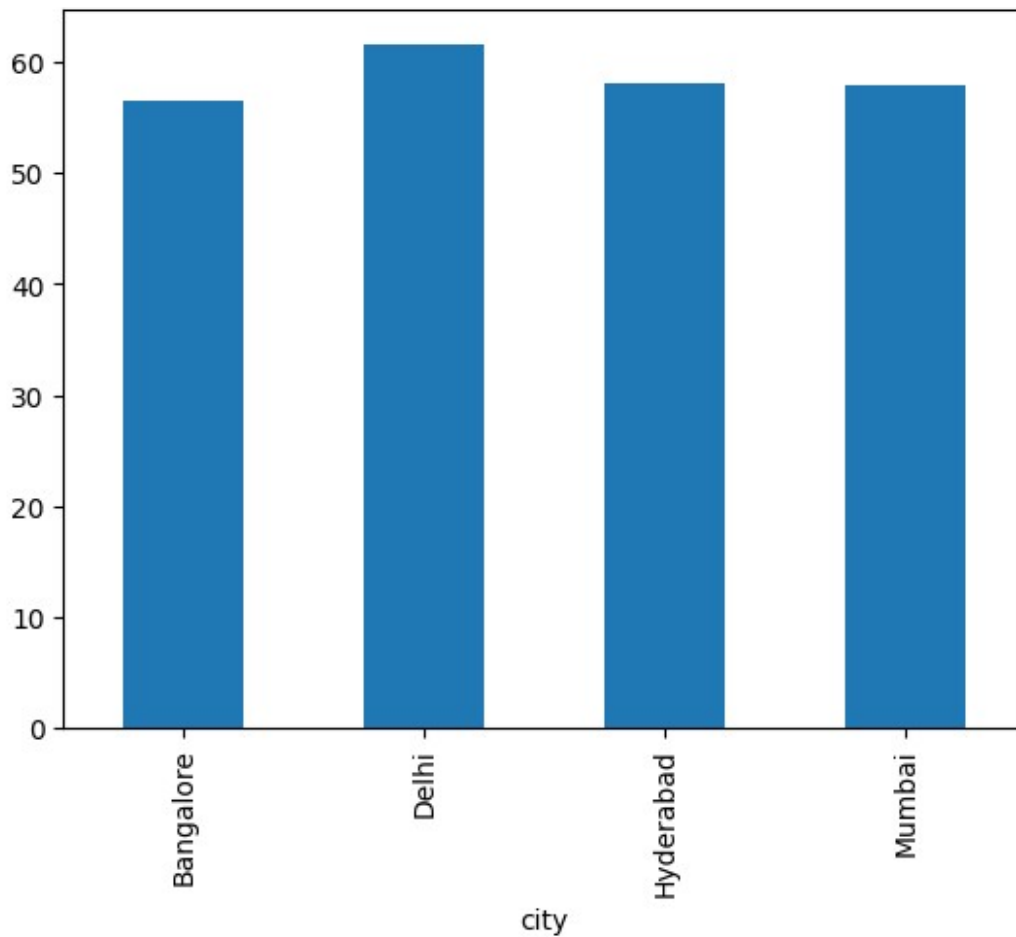
	occ_pct	room_class	property_name_x	category_x	city_x
0	83.33	Standard	Atliq Exotica	Luxury	Mumbai
1	93.33	Standard	Atliq Bay	Luxury	Bangalore
2	76.67	Standard	Atliq Palace	Business	Bangalore
3	157.89	Standard	Atliq Grands	Luxury	Mumbai
4	94.74	Standard	Atliq Grands	Luxury	Delhi

	category_y	city_y	property_name	category	city
0	Luxury	Mumbai	Atliq Exotica	Luxury	Mumbai
1	Luxury	Bangalore	Atliq Bay	Luxury	Bangalore
2	Business	Bangalore	Atliq Palace	Business	Bangalore
3	Luxury	Mumbai	Atliq Grands	Luxury	Mumbai
4	Luxury	Delhi	Atliq Grands	Luxury	Delhi

Print average occupancy rate per city

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

<Axes: xlabel='city'>



When was the Occupancy better ? Weekday or Weekend ?

```
df.head(5)
```

	property_id	check_in_date	room_category	successful_bookings
0	16559	1-May-22	RT1	25
1	19562	1-May-22	RT1	28
2	19563	1-May-22	RT1	23
3	17558	1-May-22	RT1	30
4	16558	1-May-22	RT1	18

occ_pct	room_class	property_name_x	category_x	city_x
---------	------------	-----------------	------------	--------

	property_name_y \					
0	83.33	Standard	Atliq Exotica	Luxury	Mumbai	Atliq Exotica
1	93.33	Standard	Atliq Bay	Luxury	Bangalore	Atliq Bay
2	76.67	Standard	Atliq Palace	Business	Bangalore	Atliq Palace
3	157.89	Standard	Atliq Grands	Luxury	Mumbai	Atliq Grands
4	94.74	Standard	Atliq Grands	Luxury	Delhi	Atliq Grands

	category_y	city_y	property_name	category	city
0	Luxury	Mumbai	Atliq Exotica	Luxury	Mumbai
1	Luxury	Bangalore	Atliq Bay	Luxury	Bangalore
2	Business	Bangalore	Atliq Palace	Business	Bangalore
3	Luxury	Mumbai	Atliq Grands	Luxury	Mumbai
4	Luxury	Delhi	Atliq Grands	Luxury	Delhi

df_date.head(5)

	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

df = pd.merge(df, df_date, left_on="check_in_date", right_on="date")
df.tail(5)

	property_id	check_in_date	room_category	successful_bookings
capacity \				
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_class	property_name_x	category_x	city_x \
6495	72.22	Presidential	Atliq Palace	Business	Delhi
6496	72.22	Presidential	Atliq Exotica	Luxury	Mumbai
6497	50.00	Presidential	Atliq Grands	Luxury	Mumbai
6498	50.00	Presidential	Atliq Palace	Business	Bangalore
6499	75.00	Presidential	Atliq Blu	Luxury	Mumbai

	property_name_y	category_y	city_y	property_name	category	\
6495	Atliq Palace	Business	Delhi	Atliq Palace	Business	
6496	Atliq Exotica	Luxury	Mumbai	Atliq Exotica	Luxury	
6497	Atliq Grands	Luxury	Mumbai	Atliq Grands	Luxury	
6498	Atliq Palace	Business	Bangalore	Atliq Palace	Business	
6499	Atliq Blu	Luxury	Mumbai	Atliq Blu	Luxury	

	city	date	mmm	yy	week	no	day_type
6495	Delhi	31-Jul-22	Jul	22	W	32	weekend
6496	Mumbai	31-Jul-22	Jul	22	W	32	weekend
6497	Mumbai	31-Jul-22	Jul	22	W	32	weekend
6498	Bangalore	31-Jul-22	Jul	22	W	32	weekend
6499	Mumbai	31-Jul-22	Jul	22	W	32	weekend

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

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

In the month of June, what is the occupancy for different city ?

```
df["mmm yy"].unique()
array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
df_June_22 = df[df["mmm yy"]=="Jun 22"]
df_June_22.head(5)
```

	property_id	check_in_date	room_category	successful_bookings
capacity \				
2200	16559	10-Jun-22	RT1	20
30.0				
2201	19562	10-Jun-22	RT1	19
30.0				
2202	19563	10-Jun-22	RT1	17
30.0				
2203	17558	10-Jun-22	RT1	9
19.0				
2204	16558	10-Jun-22	RT1	11
19.0				

	occ_pct	room_class	property_name_x	category_x	city_x	\
2200	66.67	Standard	Atliq Exotica	Luxury	Mumbai	
2201	63.33	Standard	Atliq Bay	Luxury	Bangalore	
2202	56.67	Standard	Atliq Palace	Business	Bangalore	
2203	47.37	Standard	Atliq Grands	Luxury	Mumbai	
2204	57.89	Standard	Atliq Grands	Luxury	Delhi	

	property_name_y	category_y	city_y	property_name	category	\
2200	Atliq Exotica	Luxury	Mumbai	Atliq Exotica	Luxury	
2201	Atliq Bay	Luxury	Bangalore	Atliq Bay	Luxury	
2202	Atliq Palace	Business	Bangalore	Atliq Palace	Business	
2203	Atliq Grands	Luxury	Mumbai	Atliq Grands	Luxury	
2204	Atliq Grands	Luxury	Delhi	Atliq Grands	Luxury	

	city	date	mmm	yy	week	no	day_type
2200	Mumbai	10-Jun-22	Jun	22	W	24	weekeday
2201	Bangalore	10-Jun-22	Jun	22	W	24	weekeday
2202	Bangalore	10-Jun-22	Jun	22	W	24	weekeday
2203	Mumbai	10-Jun-22	Jun	22	W	24	weekeday
2204	Delhi	10-Jun-22	Jun	22	W	24	weekeday

```
df_June_22.groupby("city")["occ_pct"].mean().round(2).sort_values()
```

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

```
df_august = pd.read_csv("datasets/new_data_august.csv")
df_august.head(5)
```

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

	check_in_date	mmm	yy	week	no	day_type	successful_bookings
0	01-Aug-22	Aug	-22	W	32	weekeday	30
30							
1	01-Aug-22	Aug	-22	W	32	weekeday	21
30							
2	01-Aug-22	Aug	-22	W	32	weekeday	23
30							
3	01-Aug-22	Aug	-22	W	32	weekeday	30
40							

```
4      01-Aug-22  Aug-22    W 32  weekday          20
26
```

```
      occ%
0  100.00
1   70.00
2   76.67
3   75.00
4   76.92
```

```
df_august.shape
```

```
(7, 13)
```

Append Dataset for new month

```
latest_df = pd.concat([df, df_august], ignore_index= True, axis=0)
latest_df.tail(5)
```

```
      property_id check_in_date room_category  successful_bookings
capacity \
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_class property_name_x category_x city_x  ...
category_y \
6502      NaN  Standard          NaN          NaN  NaN  ...
NaN
6503      NaN  Standard          NaN          NaN  NaN  ...
NaN
6504      NaN  Standard          NaN          NaN  NaN  ...
NaN
6505      NaN  Standard          NaN          NaN  NaN  ...
NaN
6506      NaN  Standard          NaN          NaN  NaN  ...
NaN
```

```
      city_y  property_name  category      city date  mmm yy week
no \
6502      NaN  Atliq Palace  Business  Bangalore  NaN  Aug-22  W 32
6503      NaN  Atliq Grands   Luxury  Bangalore  NaN  Aug-22  W 32
```

6504	NaN	Atliq City	Business	Bangalore	NaN	Aug-22	W 32
6505	NaN	Atliq Blu	Luxury	Mumbai	NaN	Aug-22	W 32
6506	NaN	Atliq Seasons	Business	Mumbai	NaN	Aug-22	W 32

	day_type	occ%
6502	weekeday	76.67
6503	weekeday	75.00
6504	weekeday	76.92
6505	weekeday	69.23
6506	weekeday	62.50

[5 rows x 21 columns]

latest_df.shape

(6507, 21)

Print revenue realize per city

df_bookings.head(5)

	booking_id	property_id	booking_date	check_in_date	checkout_date \
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022
7	May012216558RT18	16558	26-04-22	1/5/2022	3/5/2022

	no_guests	room_category	booking_platform	ratings_given	booking_status \
1	2.0	RT1	others	NaN	Cancelled
4	4.0	RT1	direct online	5.0	Checked Out
5	2.0	RT1	others	4.0	Checked Out
6	2.0	RT1	others	NaN	Cancelled
7	2.0	RT1	logtrip	NaN	No Show

	revenue_generated	revenue_realized
1	9100	3640
4	10920	10920
5	9100	9100
6	9100	3640
7	9100	9100

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

	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

	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

	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

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

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

Month by month revenue

```
df_date.head(3)
```

```
   date mmm yy week no day_type
0 01-May-22 May 22 W 19 weekend
1 02-May-22 May 22 W 19 weekday
2 03-May-22 May 22 W 19 weekday
```

```
pd.merge(df_bookings_all, df_date, left_on="check_in_date",
right_on="date")
```

Empty DataFrame

Columns: [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]
Index: []

```
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
```

```
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

```

```

df_date["date"] = pd.to_datetime(df_date["date"])
df_date.head(3)

```

C:\Users\DELL\AppData\Local\Temp\ipykernel_9660\173964601.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"])

```

```

      date  mmm yy week no  day_type
0 2022-05-01  May 22   W 19  weekend
1 2022-05-02  May 22   W 19  weekday
2 2022-05-03  May 22   W 19  weekday

```

```

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

```

```

df_bookings_all["check_in_date"] =
pd.to_datetime(df_bookings_all["check_in_date"], errors='coerce')
df_bookings_all.head(4)

```

```

      booking_id  property_id booking_date check_in_date
checkout_date \
0  May012216558RT12      16558    30-04-22    2022-01-05
2/5/2022
1  May012216558RT15      16558    27-04-22    2022-01-05
2/5/2022
2  May012216558RT16      16558    1/5/2022    2022-01-05
3/5/2022
3  May012216558RT17      16558    28-04-22    2022-01-05
6/5/2022

```

```

      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					

	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

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

	booking_id	property_id	booking_date	check_in_date
checkout_date \				
0 May0522	16558RT11	16558	15-04-22	2022-05-05
7/5/2022				
1 May0522	16558RT12	16558	30-04-22	2022-05-05
7/5/2022				
2 May0522	16558RT13	16558	1/5/2022	2022-05-05
6/5/2022				

	no_guests	room_category	booking_platform	ratings_given
booking_status \				
0	3.0	RT1	tripster	5.0
Out				
1	2.0	RT1	others	NaN
Cancelled				
2	3.0	RT1	direct offline	5.0
Out				

	revenue_generated	revenue_realized	property_name	category
city \				
0	10010	10010	Atliq Grands	Luxury
1	9100	3640	Atliq Grands	Luxury
2	10010	10010	Atliq Grands	Luxury

	date	mmm	yy	week	no	day_type
0	2022-05-05	May	22	W	19	weekeday
1	2022-05-05	May	22	W	19	weekeday
2	2022-05-05	May	22	W	19	weekeday

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

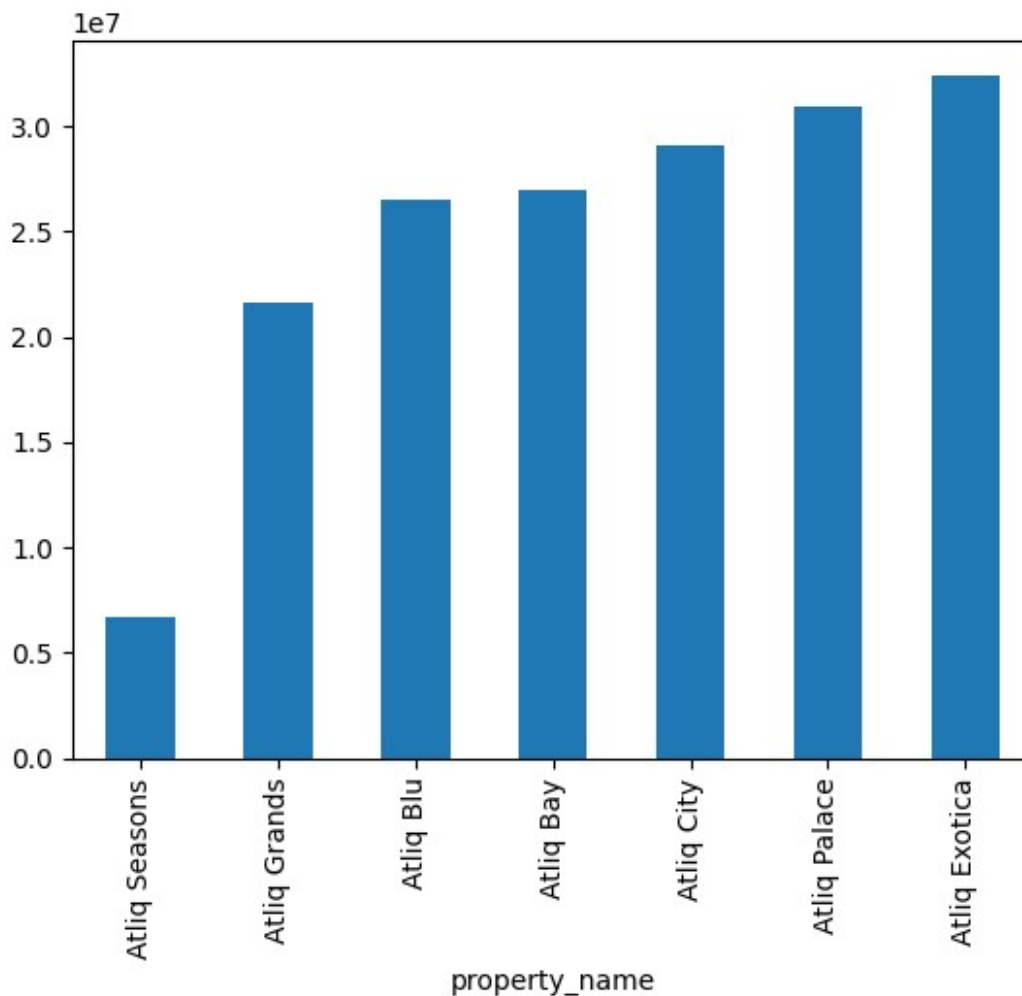


```
mmm yy
Jul 22    60278496
Jun 22    52903014
May 22    60961428
Name: revenue_realized, dtype: int64
```

Print revenue realized per hotel type

```
df_bookings_all.property_name.unique()
array(['Atliq Grands', 'Atliq Exotica', 'Atliq City', 'Atliq Blu',
      'Atliq Bay', 'Atliq Palace', 'Atliq Seasons'], dtype=object)

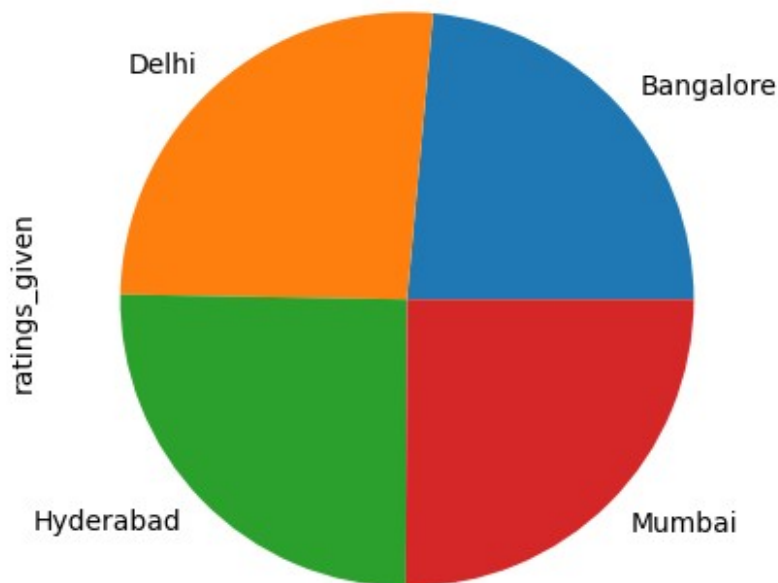
df_bookings_all.groupby("property_name")
["revenue_realized"].sum().round(2).sort_values().plot(kind="bar")
<Axes: xlabel='property_name'>
```



Print average rating per city

```
df_bookings_all.groupby("city")["ratings_given"].mean().round(2)
city
Bangalore    3.41
Delhi        3.79
Hyderabad    3.65
Mumbai       3.63
Name: ratings_given, dtype: float64

df_bookings_all.groupby("city")
["ratings_given"].mean().round(2).plot(kind="pie")
<Axes: ylabel='ratings_given'>
```



Print a pie chart of revenue realized per booking platform

```
df_bookings_all.groupby("booking_platform")
["revenue_realized"].sum().plot(kind="pie")
<Axes: ylabel='revenue_realized'>
```

