

**PROJECT
PRESENTATION ON
PYTHON -
POWERED REVIVAL
:ATLIQ GRANDS
DATA-DRIVEN
STRATEGY**

**PRESENTED BY
ABHINAV GHOSH**



Problem Statement:

AtliQ Grands, a distinguished hospitality brand with over two decades of expertise in luxury and business accommodations, is currently navigating the challenges of retaining its market share amidst rising competition. To regain its competitive edge, the company is turning to the power of business and data intelligence, aiming to enhance strategic decision-making.

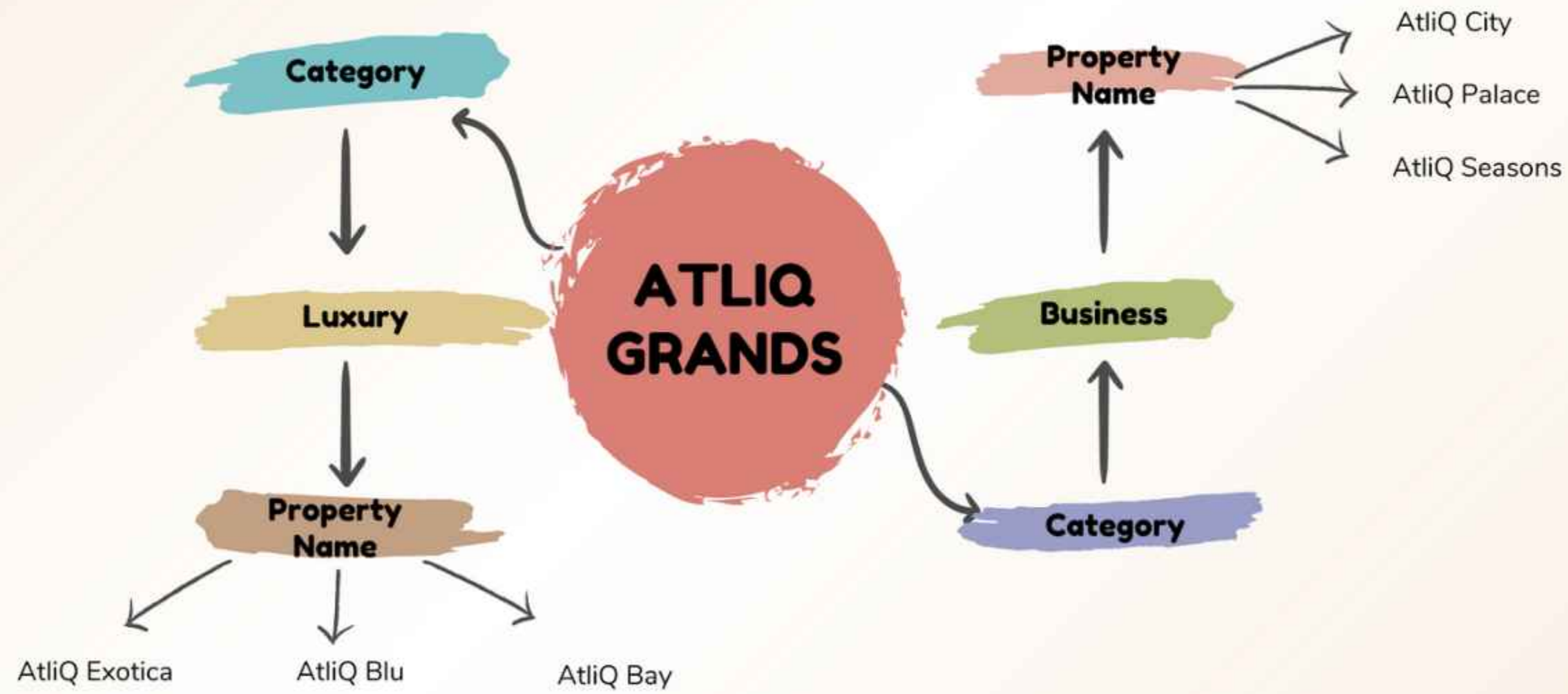
Recognizing the value of specialized insight, AtliQ Grands is exploring a partnership with a third-party data analytics provider. This collaboration will allow the company to unlock actionable insights from its historical data, empowering its revenue management team to make informed, data-driven decisions and drive optimal performance. Through this strategic alliance, AtliQ Grands aims to reassert its position as a leader in the hospitality industry.



Project Goal:

To elevate its market position and increase revenue in the luxury and business hotel sector, AtliQ Grands is dedicated to implementing data-driven strategies and leveraging business intelligence. By harnessing the power of data analytics, the company aims to outpace competitors, make informed decisions, and ultimately achieve sustainable growth and success.

AtliQ Grands Business Model:



AtliQ Hotels Data Analysis Project



```
import pandas as pd
```

1. Data Import and Data Exploration.

```
df_bookings = pd.read_csv(r"C:\Code\Pandas_Matplotlib_Seaborn\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	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
0	May012216558RT11	16558	27-04-22	1/5/2022	2/5/2022	-3.0	RT1	direct online	1.0	Checked Out	
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	5.0	Checked Out	
3	May012216558RT14	16558	28-04-22	1/5/2022	2/5/2022	-2.0	RT1	others	NaN	Cancelled	

```
df_bookings.shape
```

```
(134590, 12)
```

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

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



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

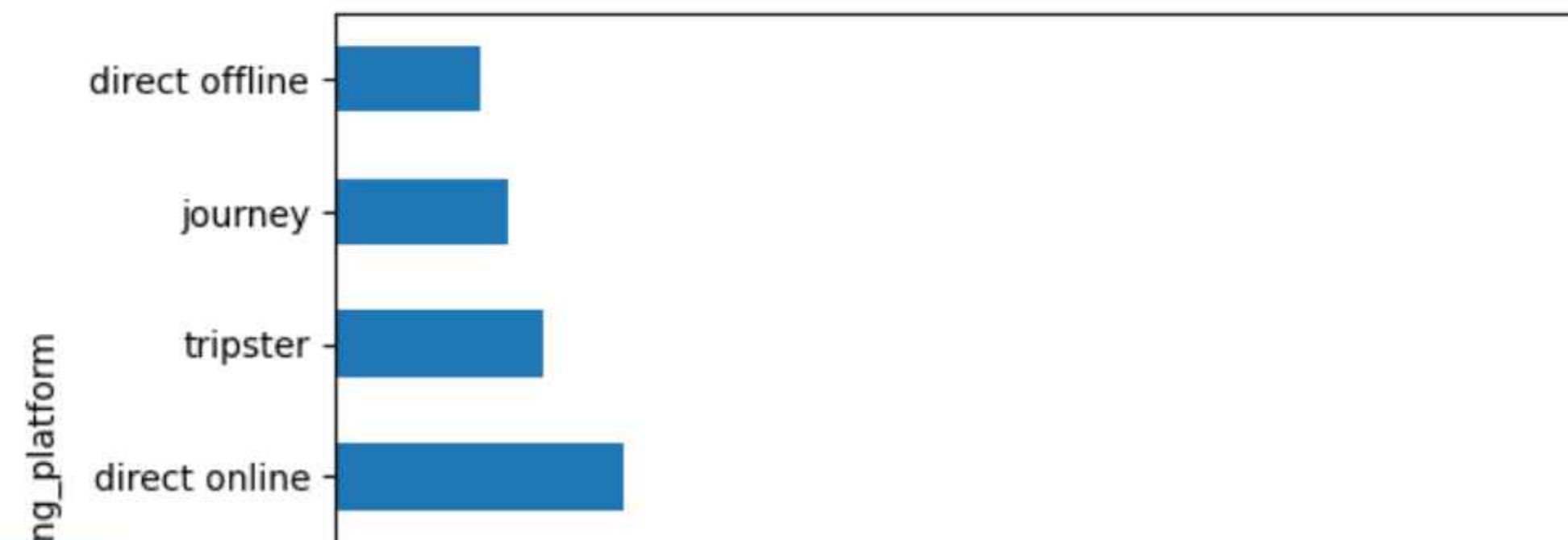
```
array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',  
      'journey', 'direct offline'], dtype=object)
```

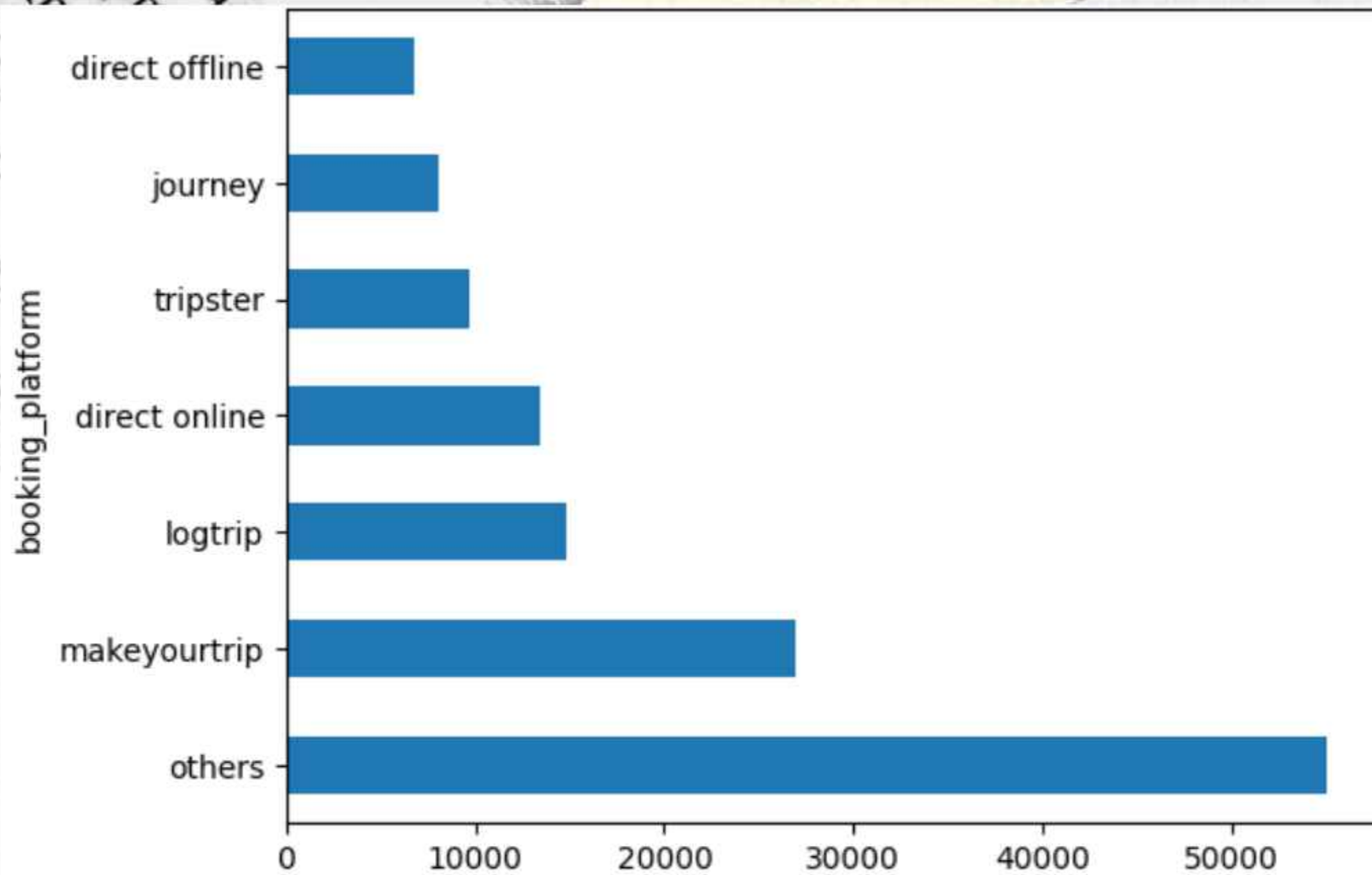
```
df_bookings.booking_platform.value_counts()
```

```
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="barh")
```

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





```
df_bookings.describe()
```

	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

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

```
(np.int64(6500), np.int64(28560000))
```

```
df_date = pd.read_csv(r"C:\Code\Pandas_Matplotlib_Seaborn\source-code\3_project_hospitality_analysis\datasets\dim_date.csv")
df_hotels = pd.read_csv(r"C:\Code\Pandas_Matplotlib_Seaborn\source-code\3_project_hospitality_analysis\datasets\dim_hotels.csv")
df_rooms = pd.read_csv(r"C:\Code\Pandas_Matplotlib_Seaborn\source-code\3_project_hospitality_analysis\datasets\dim_rooms.csv")
df_agg_bookings = pd.read_csv(r"C:\Code\Pandas_Matplotlib_Seaborn\source-code\3_project_hospitality_analysis\datasets\fact_aggregated_bookings.csv")
```

```
df_hotels.shape
```

```
(25, 4)
```

```
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

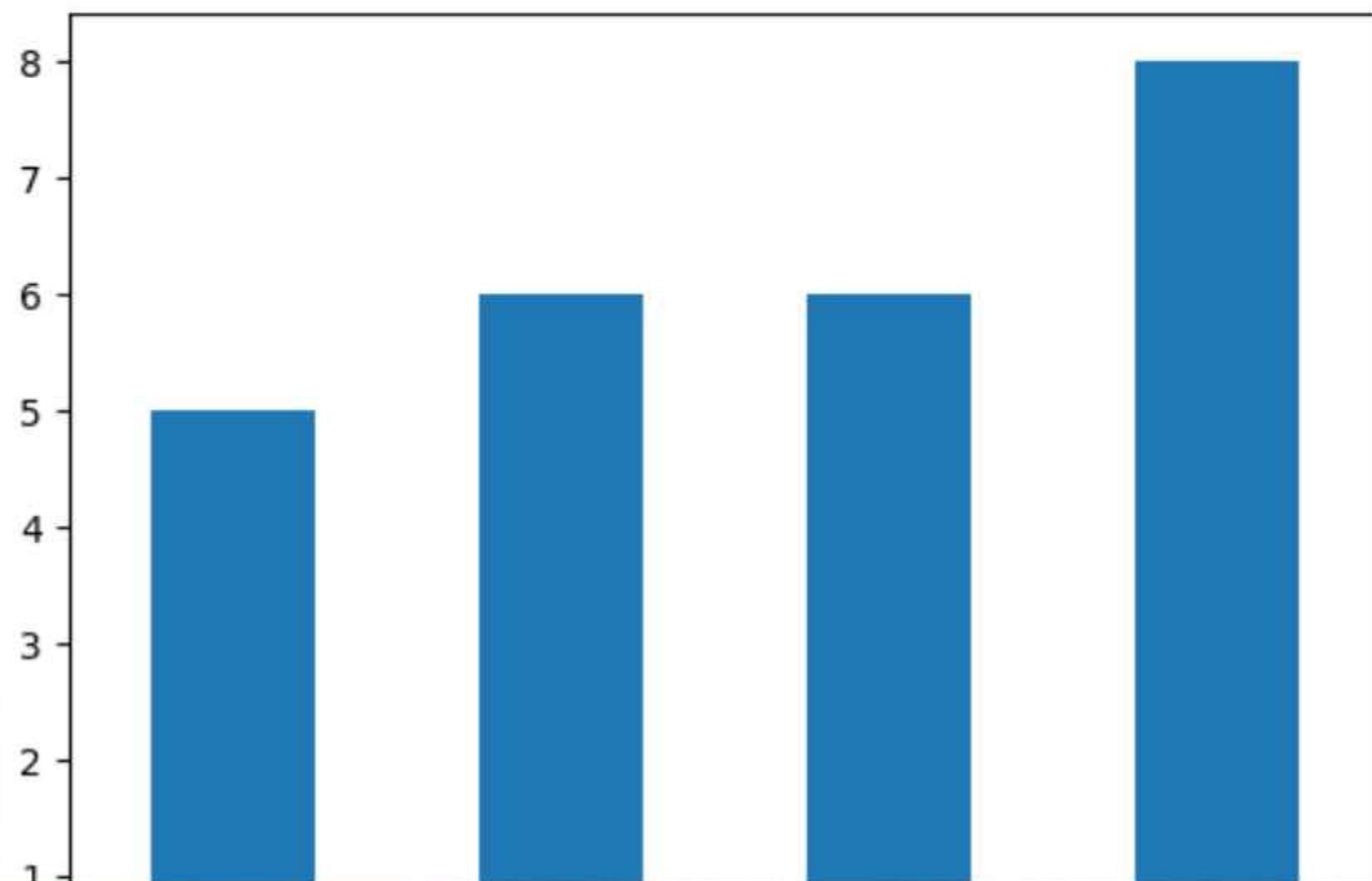
3 16561 Atliq Blu Luxury Delhi

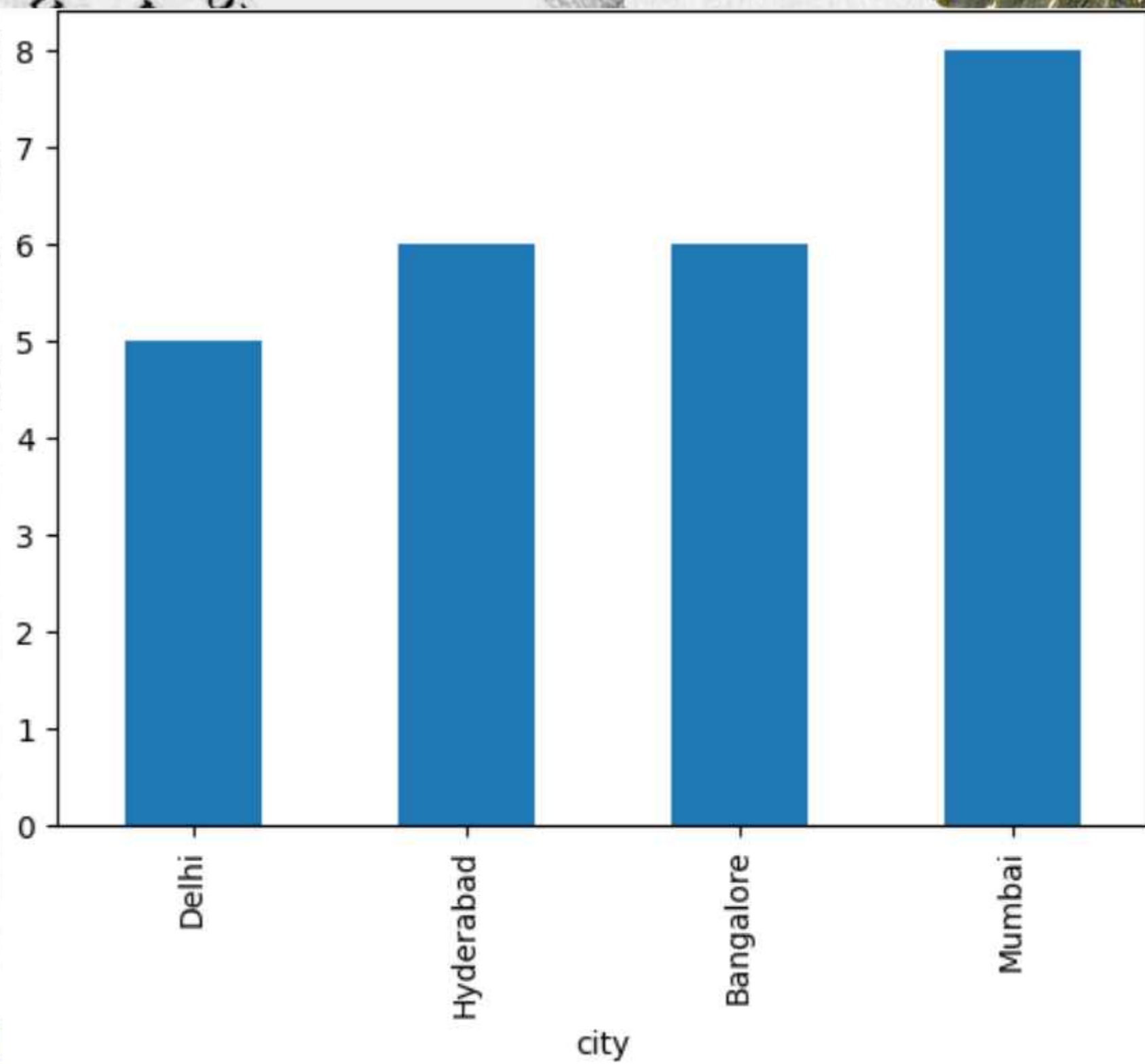
```
df_hotels.category.value_counts()
```

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

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

<Axes: xlabel='city'>





2. Data Cleaning.

```
df_bookings.describe()
```


	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

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

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue
0	May012216558RT11	16558	27-04-22	1/5/2022	2/5/2022	-3.0	RT1	direct online	1.0	Checked Out	
3	May012216558RT14	16558	28-04-22	1/5/2022	2/5/2022	-2.0	RT1	others	NaN	Cancelled	
17924	May122218559RT44	18559	12/5/2022	12/5/2022	14-05-22	-10.0	RT4	direct online	NaN	No Show	
18020	May122218561RT22	18561	8/5/2022	12/5/2022	14-05-22	-12.0	RT2	makeyourtrip	NaN	Cancelled	
18119	May122218562RT311	18562	5/5/2022	12/5/2022	17-05-22	-6.0	RT3	direct offline	5.0	Checked Out	
18121	May122218562RT313	18562	10/5/2022	12/5/2022	17-05-22	-4.0	RT3	direct online	NaN	Cancelled	
56715	Jun082218562RT12	18562	5/6/2022	8/6/2022	13-06-22	-17.0	RT1	others	NaN	Checked Out	

119765	Jul202219560RT220	19560	19-07-22	20-07-22	22-07-22	-1.0	RT2	others	NaN	Checked Out
134586	Jul312217564RT47	17564	30-07-22	31-07-22	1/8/2022	-4.0	RT4	logtrip	2.0	Checked Out

df_bookings.shape

(134590, 12)

```
df_bookings = df_bookings[df_bookings.no_guests>0]
df_bookings.head(4)
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	5.0	Checked Out	
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	4.0	Checked Out	

df_bookings.shape

(134578, 12)

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

(np.int64(6500), np.int64(28560000))


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

```
(np.float64(15378.036937686695), np.float64(93040.1549314641))
```

```
higher_limit_rg = avg+3*std
higher_limit_rg
```

```
np.float64(294498.50173207896)
```

```
lower_limit_rg = avg-3*std
lower_limit_rg
```

```
np.float64(-263742.4278567056)
```

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

booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_generated
------------	-------------	--------------	---------------	---------------	-----------	---------------	------------------	---------------	----------------	-------------------

```
df_bookings[df_bookings.revenue_generated>higher_limit_rg]
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	rev
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	5.0	Checked Out	
111	May012216559RT32	16559	29-04-22	1/5/2022	2/5/2022	6.0	RT3	direct online	NaN	Checked Out	
315	May012216562RT22	16562	28-04-22	1/5/2022	4/5/2022	2.0	RT2	direct offline	3.0	Checked Out	
562	May012217559RT118	17559	26-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	


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

```
(np.float64(15378.036937686695), np.float64(93040.1549314641))
```

```
higher_limit_rg = avg+3*std
higher_limit_rg
```

```
np.float64(294498.50173207896)
```

```
lower_limit_rg = avg-3*std
lower_limit_rg
```

```
np.float64(-263742.4278567056)
```

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

booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_generated
------------	-------------	--------------	---------------	---------------	-----------	---------------	------------------	---------------	----------------	-------------------

```
df_bookings[df_bookings.revenue_generated>higher_limit_rg]
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	rev
	2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	5.0	Checked Out
	111	May012216559RT32	16559	29-04-22	1/5/2022	2/5/2022	6.0	RT3	direct online	NaN	Checked Out
	315	May012216562RT22	16562	28-04-22	1/5/2022	4/5/2022	2.0	RT2	direct offline	3.0	Checked Out
	562	May012217559RT118	17559	26-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled

129176 Jul282216562RT26 16562 21-07-22 28-07-22 29-07-22 2.0 RT2 direct online 3.0 Checked Out

```
df_bookings = df_bookings[df_bookings.revenue_generated<=higher_limit_rg]
df_bookings.head(4)
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	4.0	Checked Out	
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1	others	NaN	Cancelled	

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

```
np.float64(33479.358661845814)
```

```
df_bookings[df_bookings.revenue_realized>higher_limit_rr]
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	rev
	137	May012216559RT41	16559	27-04-22	1/5/2022	7/5/2022	4.0	RT4	others	NaN	Checked Out
	139	May012216559RT43	16559	1/5/2022	1/5/2022	2/5/2022	6.0	RT4	tripster	3.0	Checked Out
	143	May012216559RT47	16559	28-04-22	1/5/2022	3/5/2022	3.0	RT4	others	5.0	Checked Out
	149	May012216559RT413	16559	24-04-22	1/5/2022	7/5/2022	5.0	RT4	logtrip	NaN	Checked Out
	222	May012216560RT45	16560	30-04-22	1/5/2022	3/5/2022	5.0	RT4	others	3.0	Checked Out

	134328	Jul312219560RT49	19560	31-07-22	31-07-22	2/8/2022	6.0	RT4	direct online	5.0	Checked Out
	134331	Jul312219560RT412	19560	31-07-22	31-07-22	1/8/2022	6.0	RT4	others	2.0	Checked Out
	134467	Jul312219562RT45	19562	28-07-22	31-07-22	1/8/2022	6.0	RT4	makeyourtrip	4.0	Checked Out
	134474	Jul312219562RT412	19562	25-07-22	31-07-22	6/8/2022	5.0	RT4	direct offline	5.0	Checked Out
	134581	Jul312217564RT42	17564	31-07-22	31-07-22	1/8/2022	4.0	RT4	makeyourtrip	4.0	Checked Out

1299 rows × 12 columns


```
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
```

3. 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

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

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

	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

```
df_agg_bookings["occ_pct"] = df_agg_bookings["occ_pct"].apply(lambda x: round(x*100, 2))
df_agg_bookings.head(4)
```

	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

```
df_agg_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 9200 entries, 0 to 9199
```

```
Data columns (total 6 columns):
```

#	Column	Non-Null Count	Dtype
0	property_id	9200 non-null	int64
1	check_in_date	9200 non-null	object
2	room_category	9200 non-null	object
3	successful_bookings	9200 non-null	int64
4	capacity	9198 non-null	float64

5 occ_pct 9198 non-null float64
dtypes: float64(2), int64(2), object(2)
memory usage: 431.4+ KB

df_rooms

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

df_agg_bookings

	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
...
9195	16563	31-Jul-22	RT4	13	18.0	72.22
9196	16559	31-Jul-22	RT4	13	18.0	72.22

9197	17558	31-Jul-22	RT4	3	6.0	50.00
9198	19563	31-Jul-22	RT4	3	6.0	50.00
9199	17561	31-Jul-22	RT4	3	4.0	75.00

200 rows × 6 columns

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

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

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

df_rooms

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

```
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	occ_pct	room_id	room_class
0	16559	1-May-22	RT1	25	30.0	83.33	RT1	Standard
1	19562	1-May-22	RT1	28	30.0	93.33	RT1	Standard
2	19563	1-May-22	RT1	23	30.0	76.67	RT1	Standard
3	17558	1-May-22	RT1	30	19.0	157.89	RT1	Standard

```
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
```

```
df.drop("room_id", axis=1, inplace=True)
df.head(4)
```

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

2. Print average occupancy rate per city.

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

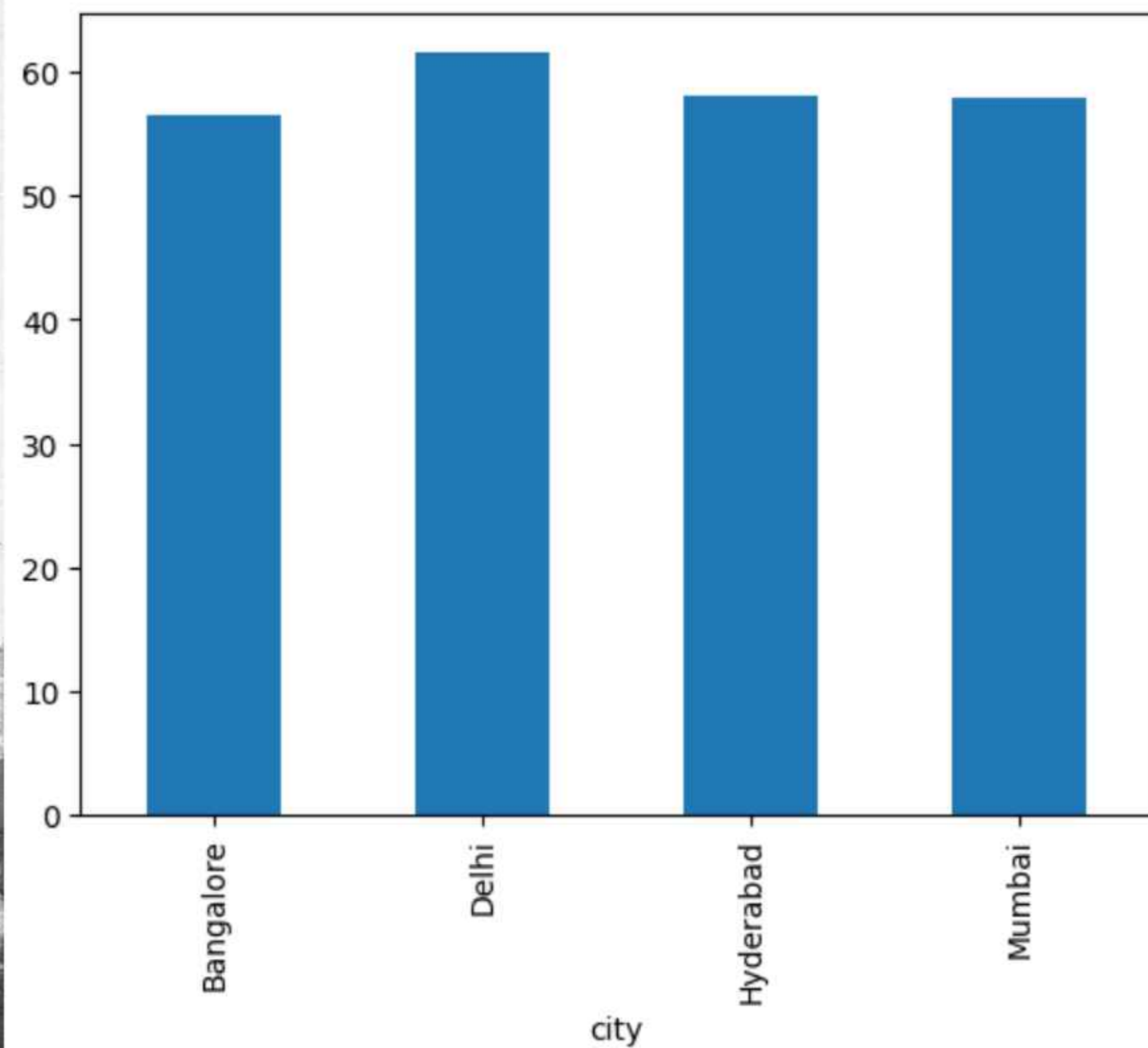
	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city
0	16559	1-May-22	RT1	25	30.0	83.33	Standard	Atliq Exotica	Luxury	Mumbai
1	19562	1-May-22	RT1	28	30.0	93.33	Standard	Atliq Bay	Luxury	Bangalore
2	19563	1-May-22	RT1	23	30.0	76.67	Standard	Atliq Palace	Business	Bangalore
3	17558	1-May-22	RT1	30	19.0	157.89	Standard	Atliq Grands	Luxury	Mumbai

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

<Axes: xlabel='city'>



<Axes: xlabel='city'>



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


```
df.head(4)
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city
0	16559	1-May-22	RT1	25	30.0	83.33	Standard	Atliq Exotica	Luxury	Mumbai
1	19562	1-May-22	RT1	28	30.0	93.33	Standard	Atliq Bay	Luxury	Bangalore
2	19563	1-May-22	RT1	23	30.0	76.67	Standard	Atliq Palace	Business	Bangalore
3	17558	1-May-22	RT1	30	19.0	157.89	Standard	Atliq Grands	Luxury	Mumbai

```
df_date.head(4)
```

	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

```
df = pd.merge(df, df_date, left_on="check_in_date", right_on="date")
df.head(4)
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city	date	mmm yy	week no	day_type
0	19563	10-May-22	RT3	15	29.0	51.72	Premium	Atliq Palace	Business	Bangalore	10-May-22	May 22	W 20	weekeday

1	18560	10-May-22	RT1	19	30.0	63.33	Standard	Atliq City	Business	Hyderabad	10-May-22	May 22	W 20	weekday
2	19562	10-May-22	RT1	18	30.0	60.00	Standard	Atliq Bay	Luxury	Bangalore	10-May-22	May 22	W 20	weekday
3	19563	10-May-22	RT1	16	30.0	53.33	Standard	Atliq Palace	Business	Bangalore	10-May-22	May 22	W 20	weekday

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

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

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

```
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(4)
```

property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city	date	mmm yy	week no	day_typ
-------------	---------------	---------------	---------------------	----------	---------	------------	---------------	----------	------	------	--------	---------	---------

10-

2200	16559	10-Jun-22	RT1	20	30.0	66.67	Standard	Atliq Exotica	Luxury	Mumbai	10-Jun-22	Jun 22	W 24	weekeda
2201	19562	10-Jun-22	RT1	19	30.0	63.33	Standard	Atliq Bay	Luxury	Bangalore	10-Jun-22	Jun 22	W 24	weekeda
2202	19563	10-Jun-22	RT1	17	30.0	56.67	Standard	Atliq Palace	Business	Bangalore	10-Jun-22	Jun 22	W 24	weekeda
2203	17558	10-Jun-22	RT1	9	19.0	47.37	Standard	Atliq Grands	Luxury	Mumbai	10-Jun-22	Jun 22	W 24	weekeda

Types of content: blog posts, videos

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

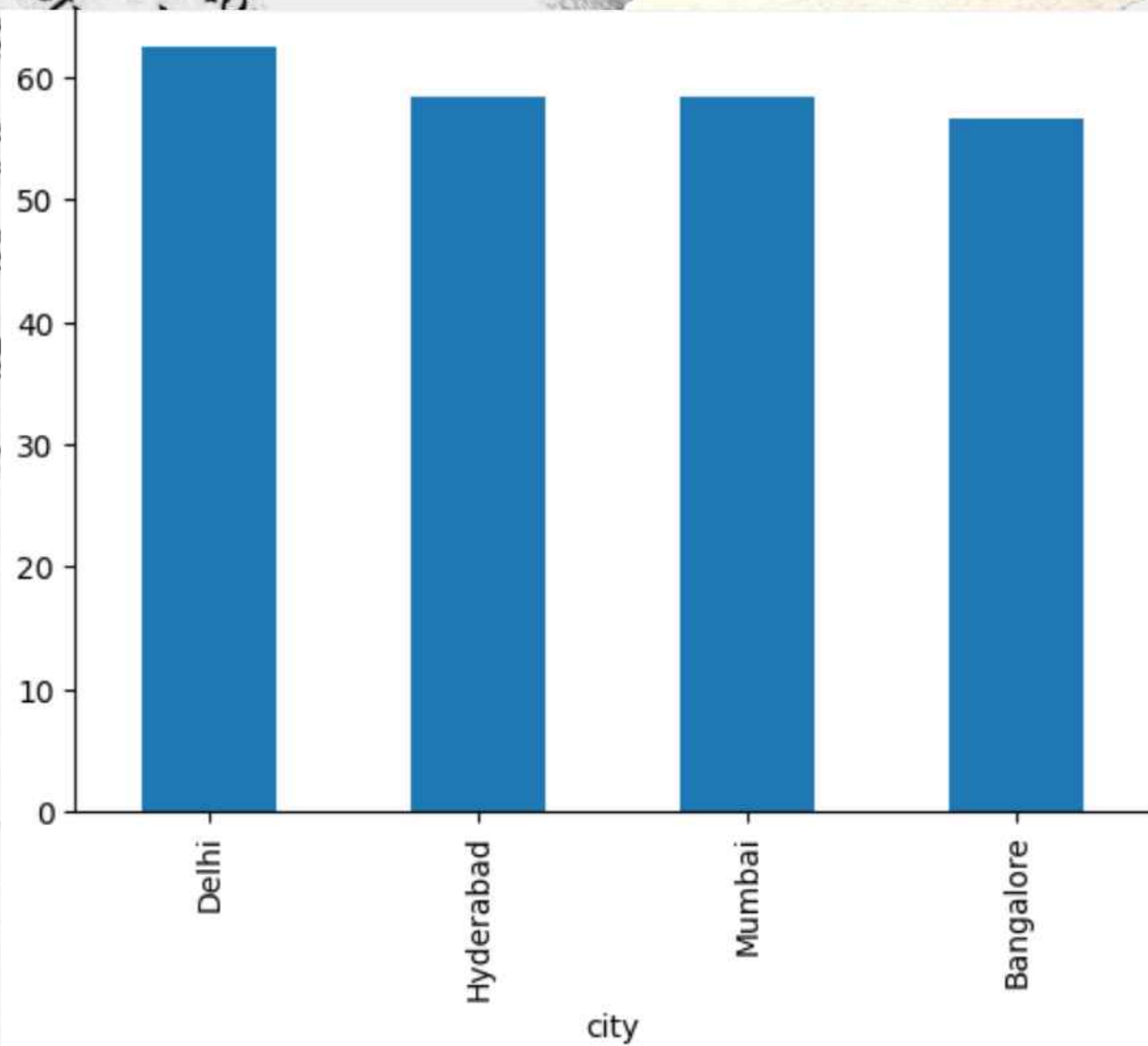
infographics, etc.

```
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'>





```
df_august = pd.read_csv(r"C:\Code\Pandas_Matplotlib_Seaborn\source-code\3_project_hospitality_analysis\datasets\new_data_august.csv")
df_august.head(4)
```

property_id	property_name	category	city	room_category	room_class	check_in_date	mmm yy	week no	day_type	successful_bookings	capacity	occ%
-------------	---------------	----------	------	---------------	------------	---------------	--------	---------	----------	---------------------	----------	------

0	16559	Atliq Exotica	Luxury	Mumbai	RT1	Standard	01-Aug-22	Aug-22	W 32	weekeday		30	30	100.00
1	19562	Atliq Bay	Luxury	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	weekeday		21	30	70.00
2	19563	Atliq Palace	Business	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	weekeday		23	30	76.67
3	19558	Atliq Grands	Luxury	Bangalore	RT1	Standard	01-Aug-22	Aug-22	W 32	weekeday		30	40	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)
```

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

```
latest_df.shape
```

```
(6507, 15)
```

5. Print revenue realized per city.

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

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	

4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	4.0	Checked Out
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1	others	NaN	Cancelled

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_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	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
0	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	
1	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	
2	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	4.0	Checked Out	
3	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1	others	NaN	Cancelled	


```
df_bookings_all.groupby("city")["revenue_realized"].sum().sort_values(ascending=False)
```

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

6. Print month by month revenue.

```
df_bookings_all.head(4)
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
0	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	
1	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	
2	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	4.0	Checked Out	
3	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1	others	NaN	Cancelled	

```
df_date.head(4)
```

	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 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    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(4)
```

C:\Users\lenovo\AppData\Local\Temp\ipykernel_6540\2639837338.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
3	2022-05-04	May 22	W 19	weekday


```
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_bookings_all["check_in_date"] = pd.to_datetime(df_bookings_all["check_in_date"], dayfirst=True, errors='coerce')
```

```
df_bookings_all.head(4)
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
0	May012216558RT12	16558	30-04-22	2022-05-01	2/5/2022	2.0	RT1	others	NaN	Cancelled	
1	May012216558RT15	16558	27-04-22	2022-05-01	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	

3 May012216558RT17 16558 28-04-22 2022-05-01 6/5/2022 2.0 RT1 others NaN Cancelled

```
df_bookings_all = pd.merge(df_bookings_all, df_date, left_on="check_in_date", right_on="date")
df_bookings_all.head(4)
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
0	May012216558RT12	16558	30-04-22	2022-05-01	2/5/2022	2.0	RT1	others	NaN	Cancelled	
1	May012216558RT15	16558	27-04-22	2022-05-01	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	
2	May012216558RT16	16558	1/5/2022	2022-05-01	3/5/2022	2.0	RT1	others	4.0	Checked Out	
3	May012216558RT17	16558	28-04-22	2022-05-01	6/5/2022	2.0	RT1	others	NaN	Cancelled	

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

```
mmm yy
Jul 22    243180932
Jun 22    229637640
May 22    234353183
Name: revenue_realized, dtype: int64
```


Thank you.

