

```
In [ ]: #AtliQ Hotels Data Analysis
```

```
In [94]: import pandas as pd
```

```
In [95]: # ==> 1. Data Import and Data Exploration
```

```
In [96]: # Datasets  
# We have 5 csv file  
  
# dim_date.csv  
# dim_hotels.csv  
# dim_rooms.csv  
# fact_aggregated_bookings  
# fact_bookings.csv
```

```
In [ ]:
```

```
In [97]: # Read bookings data in a dataframe
```

```
In [98]: df_bookings = pd.read_csv("datasets\\fact_bookings.csv")  
df_bookings.head(4)
```

```
Out[98]:
```

	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 [ ]:
```

```
In [99]: # Exploring Booking data
```

```
In [100... df_bookings.shape
```

```
Out[100... (134590, 12)
```

```
In [101... df_bookings.room_category.unique()
```

```
Out[101... array(['RT1', 'RT2', 'RT3', 'RT4'], dtype=object)
```

```
In [102... df_bookings.booking_platform.unique()
```

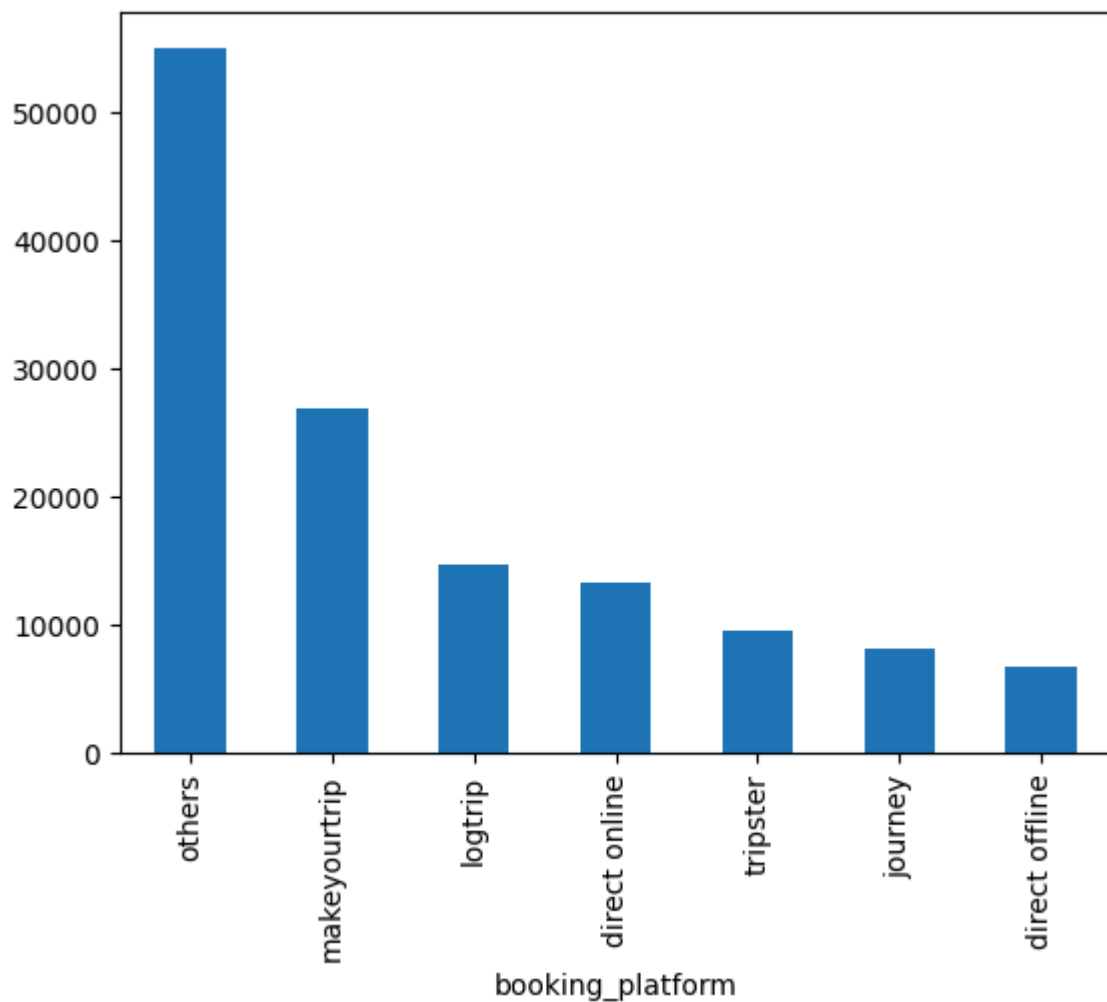
```
Out[102... array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',  
      'journey', 'direct offline'], dtype=object)
```

```
In [103... df_bookings.booking_platform.value_counts()
```

```
Out[103... booking_platform
others      55066
makeyourtrip 26898
logtrip     14756
direct online 13379
tripster    9630
journey      8106
direct offline 6755
Name: count, dtype: int64
```

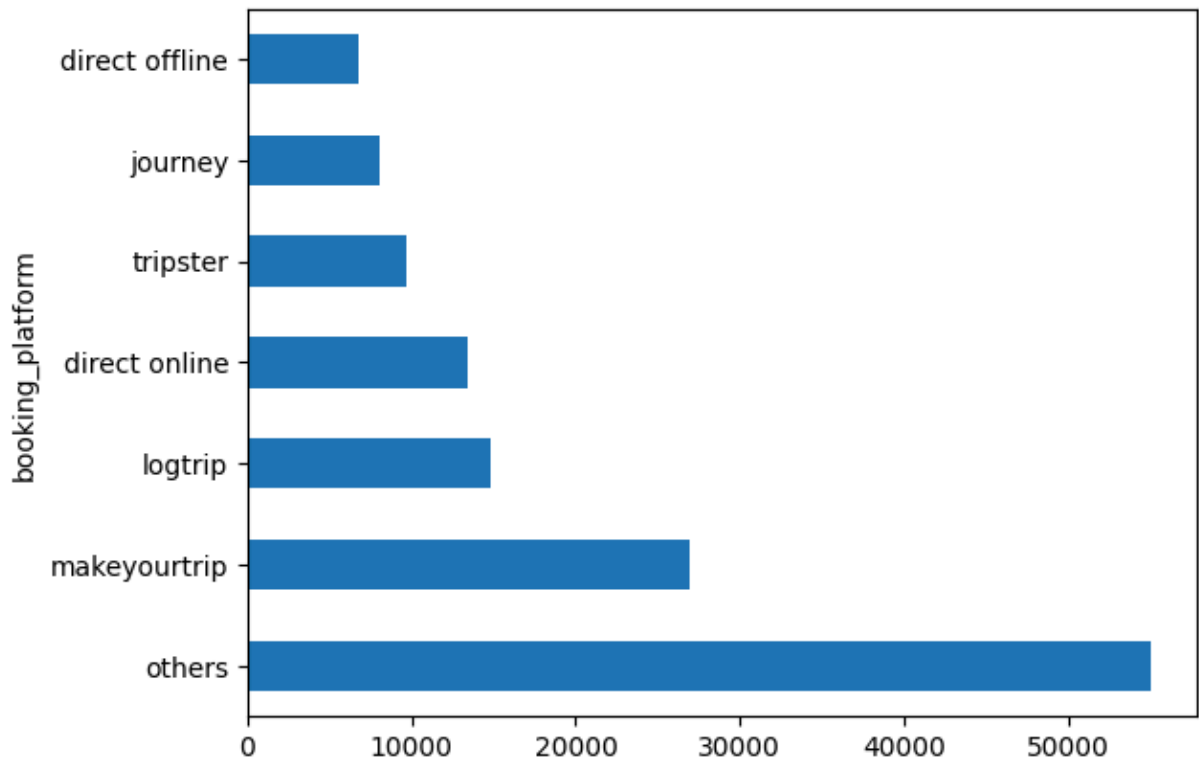
```
In [104... df_bookings.booking_platform.value_counts().plot(kind = "bar")
```

```
Out[104... <Axes: xlabel='booking_platform'>
```



```
In [105... df_bookings.booking_platform.value_counts().plot(kind = "barh")
```

```
Out[105... <Axes: ylabel='booking_platform'>
```



```
In [106...] df_bookings.describe()
```

```
Out[106...]

```

	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 [107...] df_bookings.revenue_generated.min() , df_bookings.revenue_generated.max()
```

```
Out[107...] (6500, 28560000)
```

```
In [108...] # Read rest of the files
```

```
In [109...] df_date = pd.read_csv("datasets\\dim_date.csv")
df_hotels = pd.read_csv("datasets\\dim_hotels.csv")
df_rooms = pd.read_csv("datasets\\dim_rooms.csv")
df_agg_bookings = pd.read_csv("datasets\\fact_aggregated_bookings.csv")
```

```
In [110...] df_hotels.shape
```

```
Out[110...] (25, 4)
```

```
In [111...] df_hotels.head()
```

```
Out[111...]
   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 [112...] df_hotels.category.value_counts()
```

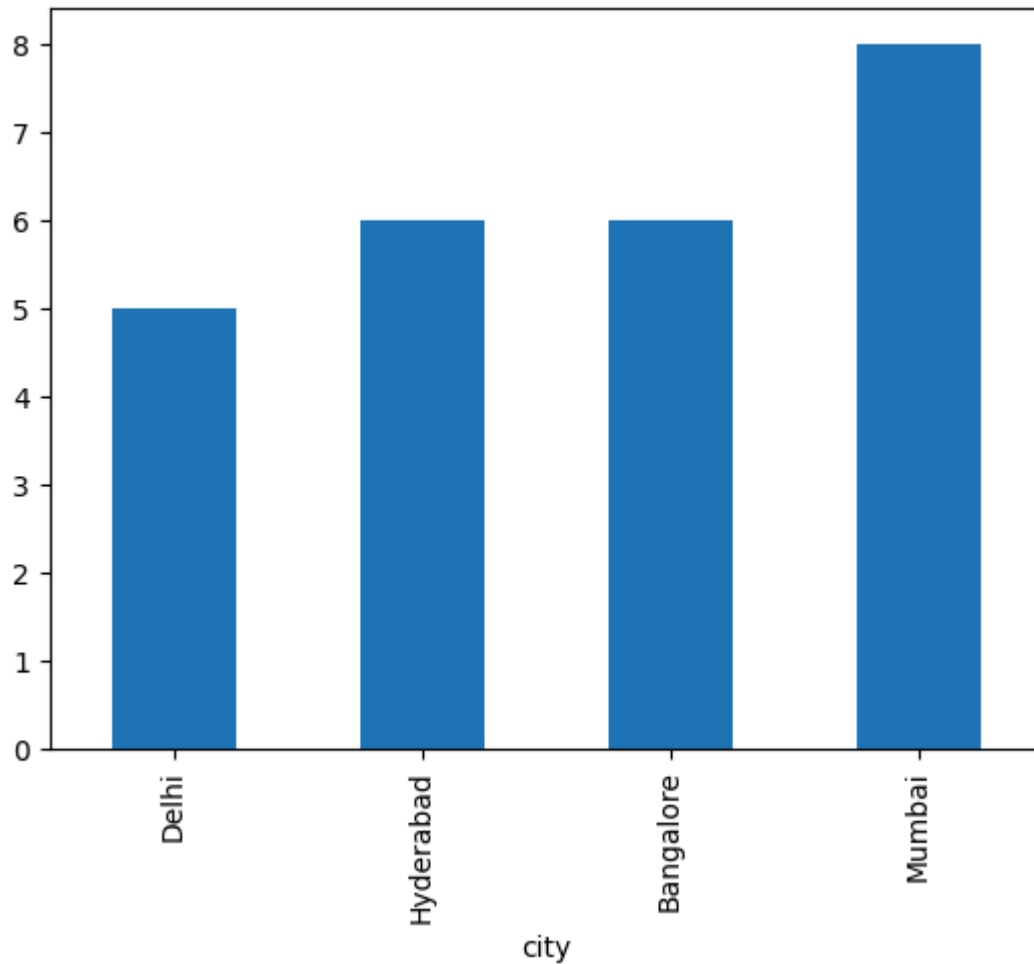
```
Out[112...]
category
Luxury      16
Business     9
Name: count, dtype: int64
```

```
In [113...] df_hotels.property_name.unique()
```

```
Out[113...]
array(['Atliq Grands', 'Atliq Exotica', 'Atliq City', 'Atliq Blu',
      'Atliq Bay', 'Atliq Palace', 'Atliq Seasons'], dtype=object)
```

```
In [114...] df_hotels.city.value_counts().sort_values().plot(kind = "bar")
```

```
Out[114...]
<Axes: xlabel='city'>
```



```
In [115... # Exercise: Explore aggregate bookings
```

```
In [116... df_agg_bookings.head(3)
```

```
Out[116...
   property_id  check_in_date  room_category  successful_bookings  capacity
0         16559      1-May-22           RT1                25         30.0
1         19562      1-May-22           RT1                28         30.0
2         19563      1-May-22           RT1                23         30.0
```

```
In [117... # Exercise-1. Find out unique property ids in aggregate bookings dataset
```

```
In [118... df_agg_bookings["property_id"].unique()
```

```
Out[118... array([16559, 19562, 19563, 17558, 16558, 17560, 19558, 19560, 17561,
        16560, 16561, 16562, 16563, 17559, 17562, 17563, 18558, 18559,
        18561, 18562, 18563, 19559, 19561, 17564, 18560], dtype=int64)
```

```
In [119... # Exercise-2. Find out total bookings per property_id
```

```
In [120... df_agg_bookings.groupby("property_id")["successful_bookings"].sum()
```

```
Out[120...] 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 [121...] # Exercise-3. Find out days on which bookings are greater than capacity
```

```
In [122...] df_agg_bookings[df_agg_bookings.successful_bookings > df_agg_bookings.capacity]
```

```
Out[122...]
   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
```

```
In [123...] # Exercise-4. Find out properties that have highest capacity
```

```
In [124...] df_agg_bookings[df_agg_bookings["capacity"] == df_agg_bookings["capacity"].max()]
```

Out[124...

	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

In []:

In []:

In [125...

2. Data Exploration/cleaning

In [126...

df_bookings.describe()

Out[126...

	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 [127...

(1) Clean invalid guests

In [128...

df_bookings[df_bookings.no_guests < 0]

Out[128...

	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 [129...

```
# As you can see above, number of guests having less than zero value represents dat
```

In [130...

```
df_bookings.shape
```

Out[130...

```
(134590, 12)
```

In [131...

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

Out[131...

```
(134578, 12)
```

In [132...

```
# (2) Outlier removal in revenue generated
```

In [133...

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

Out[133...

```
(6500, 28560000)
```

In [134...

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

In [135...

```
avg, std
```

Out[135...

```
(15378.036937686695, 93040.1549314641)
```

In [136...

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

Out[136...

```
294498.50173207896
```

In [137...

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

Out[137...

```
-263742.4278567056
```

In [138...

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



```
Out[138...      booking_id  property_id  booking_date  check_in_date  checkout_date  no_guests  room_c
```

```
In [139... df_bookings[df_bookings.revenue_generated > higher_limit]
```

```
Out[139...      booking_id  property_id  booking_date  check_in_date  checkout_date  no_
```

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

```
Out[140... (134573, 12)
```

```
In [141... df_bookings.revenue_realized.describe()
```

```
Out[141... count    134573.000000
mean      12695.983585
std        6927.791692
min         2600.000000
25%         7600.000000
50%        11700.000000
75%        15300.000000
max        45220.000000
Name: revenue_realized, dtype: float64
```

```
In [142... higher_limit = df_bookings.revenue_realized.mean() + 3 * df_bookings.revenue_realized
higher_limit
```

```
Out[142... 33479.358661845814
```

```
In [143... df_bookings[df_bookings.revenue_realized > higher_limit]
```

Out[143...

	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 [144...

```
# One observation we can have in above dataframe is that all rooms are RT4 which m
# Now since RT4 is a luxurious room it is likely their rent will be higher.
# To make a fair analysis, we need to do data analysis only on RT4 room types
```

In [145...

```
df_rooms
```

Out[145...

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

In [146...

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

Out[146...

```
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 [147...

```
23439 + 3* 9048
```

Out[147... 50583

In []:

In [148... `df_bookings.isnull().sum()`

Out[148...
booking_id 0
property_id 0
booking_date 0
check_in_date 0
checkout_date 0
no_guests 0
room_category 0
booking_platform 0
ratings_given 77897
booking_status 0
revenue_generated 0
revenue_realized 0
dtype: int64

In [149... *# Total values in our dataframe is 134576. Out of that 77899 rows has null rating.
we should not filter these values. Also we should not replace this rating with a*

In []:

In [150... *# Exercise-1. In aggregate bookings find columns that have null values. Fill these
substitute (possible ways is to use mean or median)*

In [151... `df_agg_bookings.head()`

Out[151...

	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 [152... `df_agg_bookings.isnull().sum()`

Out[152...
property_id 0
check_in_date 0
room_category 0
successful_bookings 0
capacity 2
dtype: int64

In [153... `df_agg_bookings[df_agg_bookings.capacity.isna()]`

```
Out[153...
```

	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 [154... df_agg_bookings.capacity.median()
```

```
Out[154... 25.0
```

```
In [155... df_agg_bookings["capacity"].fillna(df_agg_bookings["capacity"].median(), inplace =
```

```
In [156... df_agg_bookings.loc[[8,14]]
```

```
Out[156...
```

	property_id	check_in_date	room_category	successful_bookings	capacity
8	17561	1-May-22	RT1	22	25.0
14	17562	1-May-22	RT1	12	25.0

```
In [ ]:
```

```
In [157... # Exercise-2. In aggregate bookings find out records that have successful_bookings
```

```
In [158... df_agg_bookings[df_agg_bookings.successful_bookings > df_agg_bookings.capacity]
```

```
Out[158...
```

	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

```
In [159... df_agg_bookings.shape
```

```
Out[159... (9200, 5)
```

```
In [160... df_agg_bookings = df_agg_bookings[df_agg_bookings.successful_bookings <= df_agg_bookings.capacity]
df_agg_bookings.shape
```

```
Out[160... (9194, 5)
```

```
In [ ]:
```

```
In [161... # DATA TRANSFORMATION
```

```
In [162... # Create occupancy percentage column
```

```
In [163... df_agg_bookings.head()
```

```
Out[163... 
```

	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
4	16558	1-May-22	RT1	18	19.0
5	17560	1-May-22	RT1	28	40.0

```
In [164... df_agg_bookings["occt_pct"] = df_agg_bookings["successful_bookings"] / df_agg_bookin
```

```
In [165... df_agg_bookings.head()
```

```
Out[165... 
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occt_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
4	16558	1-May-22	RT1	18	19.0	0.947368
5	17560	1-May-22	RT1	28	40.0	0.700000

```
In [166... df_agg_bookings["occt_pct"] = df_agg_bookings["occt_pct"].apply(lambda x : round(x*100, 1))
df_agg_bookings.head()
```

```
Out[166... 
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occt_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
4	16558	1-May-22	RT1	18	19.0	94.74
5	17560	1-May-22	RT1	28	40.0	70.00

```
In [167... # There are various types of data transformations that you may have to perform base

# 1. Creating new columns
# 2. Normalization
# 3. Merging data
# 4. Aggregation
```

In []:

In [168... `# ==> 4. Insights Generation`

In [169... `1. What is an average occupancy rate in each of the room categories?`

Object `categories` not found.

In [170... `df_agg_bookings.groupby("room_category")["occt_pct"].mean().round(2)`

Out[170...
room_category
RT1 57.89
RT2 58.01
RT3 58.03
RT4 59.28
Name: occt_pct, dtype: float64

In [171... `df_rooms`

Out[171...

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

In [172... `df = pd.merge(df_agg_bookings, df_rooms, left_on = "room_category", right_on = "room_category", how="left")`
`df.head()`

Out[172...

	property_id	check_in_date	room_category	successful_bookings	capacity	occt_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	Elite
2	19563	1-May-22	RT1	23	30.0	76.67	Premium
3	16558	1-May-22	RT1	18	19.0	94.74	Presidential
4	17560	1-May-22	RT1	28	40.0	70.00	Standard

In [173... `df.groupby("room_class")["occt_pct"].mean().round(2)`

Out[173...
room_class
Elite 58.01
Premium 58.03
Presidential 59.28
Standard 57.89
Name: occt_pct, dtype: float64

In [174... `df.drop("room_id", axis = 1, inplace = True)`

In [175... `df.head()`

Out[175...

	property_id	check_in_date	room_category	successful_bookings	capacity	occt_pct	room
0	16559	1-May-22	RT1	25	30.0	83.33	S
1	19562	1-May-22	RT1	28	30.0	93.33	S
2	19563	1-May-22	RT1	23	30.0	76.67	S
3	16558	1-May-22	RT1	18	19.0	94.74	S
4	17560	1-May-22	RT1	28	40.0	70.00	S

In [176... `# 2. Print average occupancy rate per city`

In [177... `df_hotels.head()`

Out[177...

	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

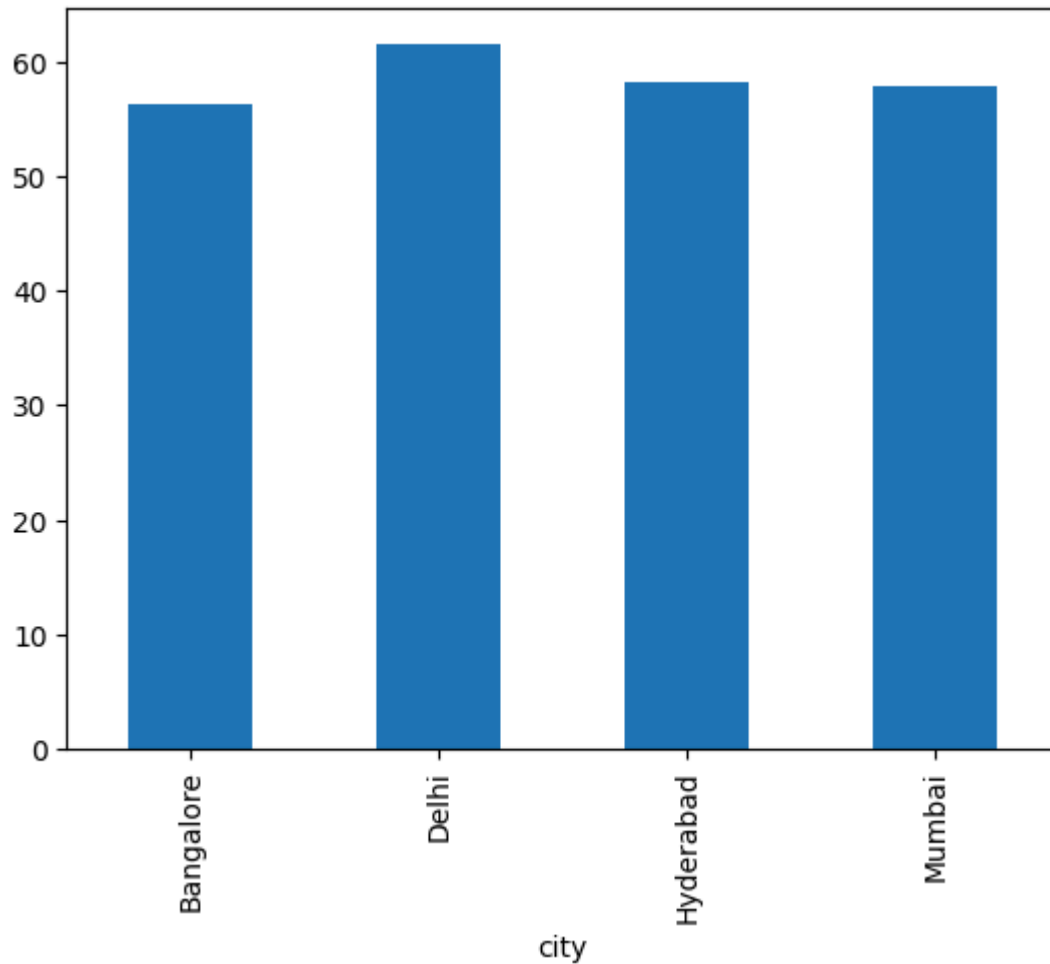
In [178... `df = pd.merge(df,df_hotels, on = "property_id")`
`df.head()`

Out[178...

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

In [179... `df.groupby("city")["occt_pct"].mean().round(2).plot(kind = "bar")`

Out[179... `<Axes: xlabel='city'>`



```
In [180... # **3. when was the occupancy better? Weekday or Weekend?**
```

```
In [181... df.head()
```

```
Out[181...
```

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

```
In [182... df_date
```


Out[182...

	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 × 4 columns

In [183...

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

Out[183...

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

In [184...

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

```
Out[184... day_type
weekeday    50.88
weekend      72.34
Name: occt_pct, dtype: float64
```

```
In [185... # **4.In month of june, what is the occupancy for different cities**
```

```
In [186... df["mmm yy"].unique()
```

```
Out[186... array(['May 22', 'Jun 22', 'Jul 22'], dtype=object)
```

```
In [187... df_june22 = df[df["mmm yy"] == "Jun 22"]
df_june22.head()
```

```
Out[187...
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occt_pct
2200	16559	10-Jun-22	RT1	20	30.0	66.67
2201	16559	10-Jun-22	RT2	26	41.0	63.41
2202	16559	10-Jun-22	RT3	20	32.0	62.50
2203	16559	10-Jun-22	RT4	11	18.0	61.11
2204	19562	10-Jun-22	RT1	19	30.0	63.33

```
In [188... df_june22.groupby("city")["occt_pct"].mean().round(2).sort_values(ascending = False)
```

```
Out[188... city
Delhi      62.47
Hyderabad  58.46
Mumbai     58.38
Bangalore  56.44
Name: occt_pct, dtype: float64
```

```
In [189... # Append the August data to existing data
```

```
In [190... df_august = pd.read_csv("datasets\\new_data_august.csv")
df_august.head()
```

Out[190...

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

In [191...

```
df_august.columns
```

Out[191...

```
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 [192...

```
df.columns
```

Out[192...

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

In [193...

```
df_august.shape
```

Out[193...

```
(7, 13)
```

In [194...

```
df.shape
```

Out[194...

```
(6497, 14)
```

In [195...

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

Out[195...

	property_id	check_in_date	room_category	successful_bookings	capacity	occt_pct
6494	16563	31-Jul-22	RT2	32	38.0	84.21
6495	16563	31-Jul-22	RT3	14	20.0	70.00
6496	16563	31-Jul-22	RT4	13	18.0	72.22
6497	16559	01-Aug-22	RT1	30	30.0	NaN
6498	19562	01-Aug-22	RT1	21	30.0	NaN
6499	19563	01-Aug-22	RT1	23	30.0	NaN
6500	19558	01-Aug-22	RT1	30	40.0	NaN
6501	19560	01-Aug-22	RT1	20	26.0	NaN
6502	17561	01-Aug-22	RT1	18	26.0	NaN
6503	17564	01-Aug-22	RT1	10	16.0	NaN

In [196...

```
latest_df.shape
```

Out[196...

```
(6504, 15)
```

In [197...

```
#Print revenue realized per city
```

In [198...

```
df_bookings.head()
```

Out[198...

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0
7	May012216558RT18	16558	26-04-22	1/5/2022	3/5/2022	2.0

```
In [199... df_hotels.head()
```

```
Out[199...
  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 [200... df_bookings_all = pd.merge(df_bookings, df_hotels, on ="property_id" )
df_bookings_all.head()
```

```
Out[200...
  booking_id  property_id  booking_date  check_in_date  checkout_date  no_guests
0  May012216558RT12      16558      30-04-22      1/5/2022      2/5/2022      2.0
1  May012216558RT15      16558      27-04-22      1/5/2022      2/5/2022      4.0
2  May012216558RT16      16558      1/5/2022      1/5/2022      3/5/2022      2.0
3  May012216558RT17      16558      28-04-22      1/5/2022      6/5/2022      2.0
4  May012216558RT18      16558      26-04-22      1/5/2022      3/5/2022      2.0
```

```
In [201... df_bookings_all.groupby("city")["revenue_realized"].sum()
```

```
Out[201...
city
Bangalore    420383550
Delhi        294404488
Hyderabad    325179310
Mumbai       668569251
Name: revenue_realized, dtype: int64
```

```
In [202... #7. print month by month revenue
```

```
In [203... df_bookings_all.tail()
```

```
Out[203...
  booking_id  property_id  booking_date  check_in_date  checkout_date  no_g
134568  Jul312217564RT45      17564      30-07-22      31-07-22      1/8/2022
134569  Jul312217564RT46      17564      29-07-22      31-07-22      3/8/2022
134570  Jul312217564RT48      17564      30-07-22      31-07-22      2/8/2022
134571  Jul312217564RT49      17564      29-07-22      31-07-22      1/8/2022
134572  Jul312217564RT410      17564      31-07-22      31-07-22      1/8/2022
```

```
In [204... df_date.head()
```

Out[204...

	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

In [205...

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

Out[205...

booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_c
------------	-------------	--------------	---------------	---------------	-----------	--------

In [206...

```
df_bookings_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134573 entries, 0 to 134572
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   booking_id            134573 non-null object
1   property_id           134573 non-null int64
2   booking_date          134573 non-null object
3   check_in_date         134573 non-null object
4   checkout_date         134573 non-null object
5   no_guests             134573 non-null float64
6   room_category         134573 non-null object
7   booking_platform      134573 non-null object
8   ratings_given         56676 non-null float64
9   booking_status        134573 non-null object
10  revenue_generated     134573 non-null int64
11  revenue_realized      134573 non-null int64
12  property_name         134573 non-null object
13  category              134573 non-null object
14  city                  134573 non-null object
dtypes: float64(2), int64(3), object(10)
memory usage: 15.4+ MB
```

In [207...

```
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 [209... df_date["date"] = pd.to_datetime(df_date["date"], format='mixed')
df_date.head()
```

```
Out[209...      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
4  2022-05-05   May 22    W 19  weekday
```

```
In [210... 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 [211... df_bookings_all.head()
```

```
Out[211...      booking_id  property_id  booking_date  check_in_date  checkout_date  no_guests
0  May012216558RT12      16558    30-04-22    1/5/2022    2/5/2022      2.0
1  May012216558RT15      16558    27-04-22    1/5/2022    2/5/2022      4.0
2  May012216558RT16      16558    1/5/2022    1/5/2022    3/5/2022      2.0
3  May012216558RT17      16558    28-04-22    1/5/2022    6/5/2022      2.0
4  May012216558RT18      16558    26-04-22    1/5/2022    3/5/2022      2.0
```

```
In [212... df_bookings_all["check_in_date"] = pd.to_datetime(df_bookings_all["check_in_date"],
df_bookings_all.head()
```

Out[212...

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests
0	May012216558RT12	16558	30-04-22	2022-01-05	2/5/2022	2.0
1	May012216558RT15	16558	27-04-22	2022-01-05	2/5/2022	4.0
2	May012216558RT16	16558	1/5/2022	2022-01-05	3/5/2022	2.0
3	May012216558RT17	16558	28-04-22	2022-01-05	6/5/2022	2.0
4	May012216558RT18	16558	26-04-22	2022-01-05	3/5/2022	2.0

In [213...

```
df_bookings_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134573 entries, 0 to 134572
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   booking_id            134573 non-null object  
1   property_id           134573 non-null int64   
2   booking_date          134573 non-null object  
3   check_in_date         134573 non-null datetime64[ns]
4   checkout_date         134573 non-null object  
5   no_guests             134573 non-null float64  
6   room_category         134573 non-null object  
7   booking_platform      134573 non-null object  
8   ratings_given         56676 non-null float64  
9   booking_status        134573 non-null object  
10  revenue_generated     134573 non-null int64   
11  revenue_realized      134573 non-null int64   
12  property_name         134573 non-null object  
13  category              134573 non-null object  
14  city                  134573 non-null object  
dtypes: datetime64[ns](1), float64(2), int64(3), object(9)
memory usage: 15.4+ MB
```

In [214...

```
df_bookings_all = pd.merge(df_bookings_all, df_date, left_on = "check_in_date" , r
df_bookings_all.head()
```


Out[214...

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests
0	May052216558RT11	16558	15-04-22	2022-05-05	7/5/2022	3.0
1	May052216558RT12	16558	30-04-22	2022-05-05	7/5/2022	2.0
2	May052216558RT13	16558	1/5/2022	2022-05-05	6/5/2022	3.0
3	May052216558RT14	16558	3/5/2022	2022-05-05	6/5/2022	2.0
4	May052216558RT15	16558	30-04-22	2022-05-05	10/5/2022	4.0

In [215...

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

Out[215...

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

In []:

In []: *# Exercise-1. Print revenue realized per hotel type*

In [216...

```
df_bookings_all.head()
```

Out[216...

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests
0	May052216558RT11	16558	15-04-22	2022-05-05	7/5/2022	3.0
1	May052216558RT12	16558	30-04-22	2022-05-05	7/5/2022	2.0
2	May052216558RT13	16558	1/5/2022	2022-05-05	6/5/2022	3.0
3	May052216558RT14	16558	3/5/2022	2022-05-05	6/5/2022	2.0
4	May052216558RT15	16558	30-04-22	2022-05-05	10/5/2022	4.0

In [223...

```
df_bookings_all.property_name.unique()
```

Out[223...

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

```
In [222... df_bookings_all.groupby("property_name")["revenue_realized"].sum().round(2).sort_va
```

```
Out[222... property_name
Atliq Seasons      45920757
Atliq Grands       145860641
Atliq Blu          179203544
Atliq Bay          179416721
Atliq City         196555383
Atliq Palace       209474575
Atliq Exotica      219076161
Name: revenue_realized, dtype: int64
```

```
In [ ]: # Exercise-2 Print average rating per city
```

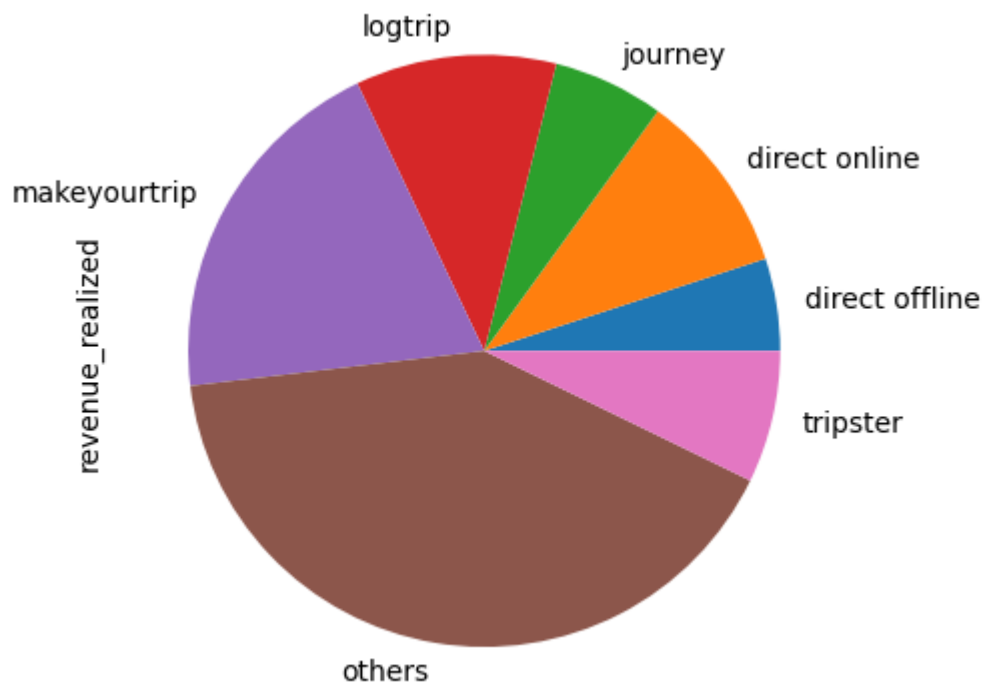
```
In [224... df_bookings_all.groupby("city")["ratings_given"].mean().round(2).sort_values()
```

```
Out[224... city
Bangalore      3.40
Mumbai         3.64
Hyderabad      3.66
Delhi          3.78
Name: ratings_given, dtype: float64
```

```
In [ ]: # Exercise-3 Print a pie chart of revenue realized per booking platform
```

```
In [226... df_bookings_all.groupby("booking_platform")["revenue_realized"].sum().round(2).plot
```

```
Out[226... <Axes: ylabel='revenue_realized'>
```



```
In [ ]:
```