python 3.11.9

Hotels Data Analysis Project

DATA ANALYSIS IN HOSPITALITY DOMAIN

- a company which onws multiple hotels in India
- they have many type of room .
- room booking by - website , 3rd party booking apps
- all booking data of verious platfrom connected to a single database(booking database)
- they have heavy compitiror in market
- and want to increse their revenue

==> 1. Data Import and Data Exploration

```
# import all liberey
import pandas as pd

# loading first data
df_booking = pd.read_csv("datasets/fact_bookings.csv")
print("data loaded - fact_bookings.csv")

data loaded - fact_bookings.csv

# looking for 4 rows to know what data i have
df_booking.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
```

- lets understand the dataframe
- unique booking id
- unique property id
- i have booking date , check in date , chackout date with no of guests
- i have unique room category maybe multiple
- i have muiltiple booking platform
- i have rating
- booking status
- based on booking status we have 2 type revenue
- 1 revenue that generated , 2 after cancelation charge /tips/ discount we have real revenue

```
# i want to know how many rows and columns i have
df_booking.shape

(134590, 12)

# as i saw  have room category , so i want to know how many room
category i have
df_booking.room_category.unique()

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

['RT1', 'RT2', 'RT3', 'RT4'] i have 4 room category

```
# now i want to know how many ways i have for booking a room in hotels
df_booking.booking_platform.unique()

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

['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip', 'journey', 'direct offline']

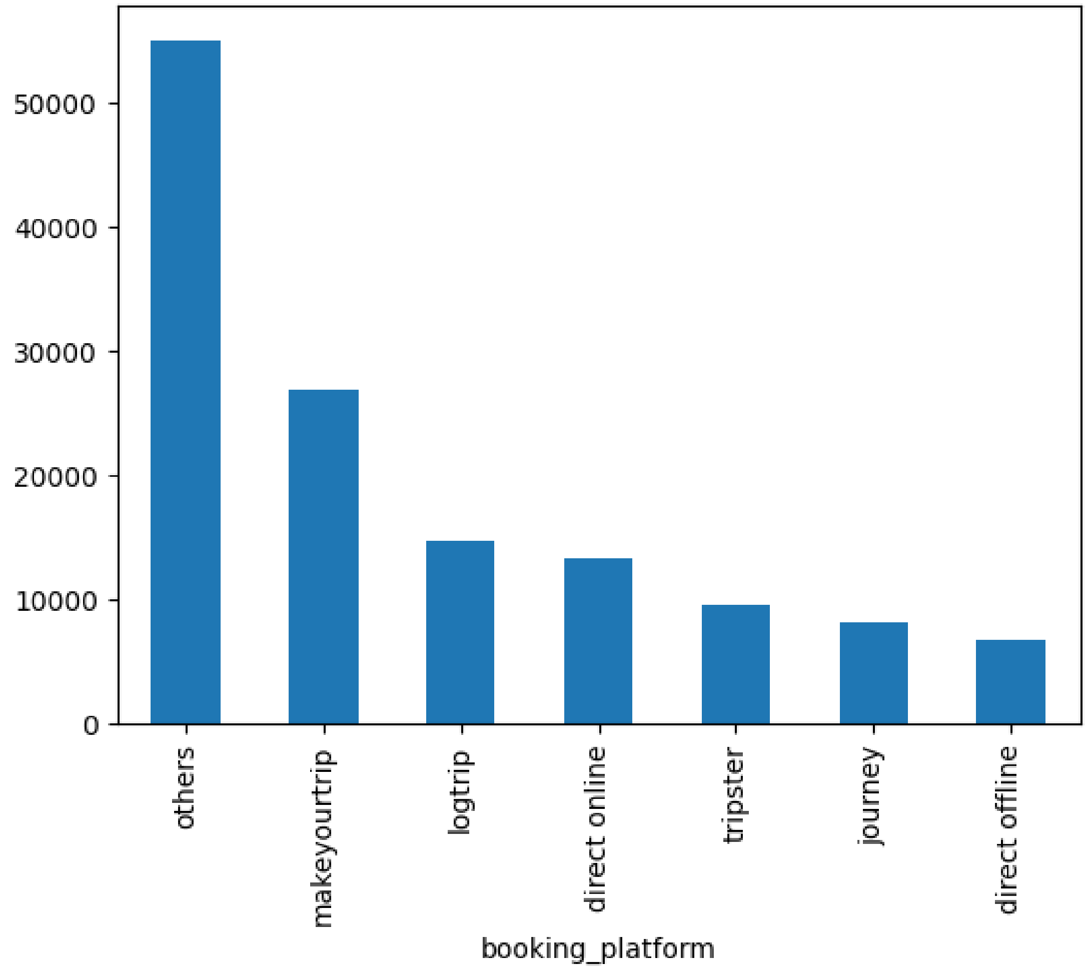
- so ihave 7 booking methods

```
# now i want to know the value of booking for eash platform
df_booking.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
```

```
# now i want to plot this
df_booking.booking_platform.value_counts().plot(kind='bar')
```

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



```
# now i want to get quick statistics
df_booking.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
```

- i can clearly see that my min rating is 1 , max is 5 , most important mean rate is 3.6
- so we have to work to get good rating atleast 4 to get a good impression
- min no guest is -17 means data error
- max no guest is 6 , and mean is 2

```
# reading revenue in this format is confusing so i'm brakeing it
df_booking.revenue_generated.min(),df_booking.revenue_generated.max()

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

- min booking amount is 6500
- max booking amount is 28560000 -- this maybe a error unreaalistic value for book a hotel room

```
# lets import my other files
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')

print("data loaded -dim_date.csv")
print("data loaded -dim_hotels.csv")
print("data loaded -dim_rooms.csv")
print("data loaded -fact_aggregated_bookings.csv")
```

```
data loaded -dim_date.csv
data loaded -dim_hotels.csv
data loaded -dim_rooms.csv
data loaded -fact_aggregated_bookings.csv
```

- let me explore hotels

```
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
```

we have total 16 luxury and 9 business hotels

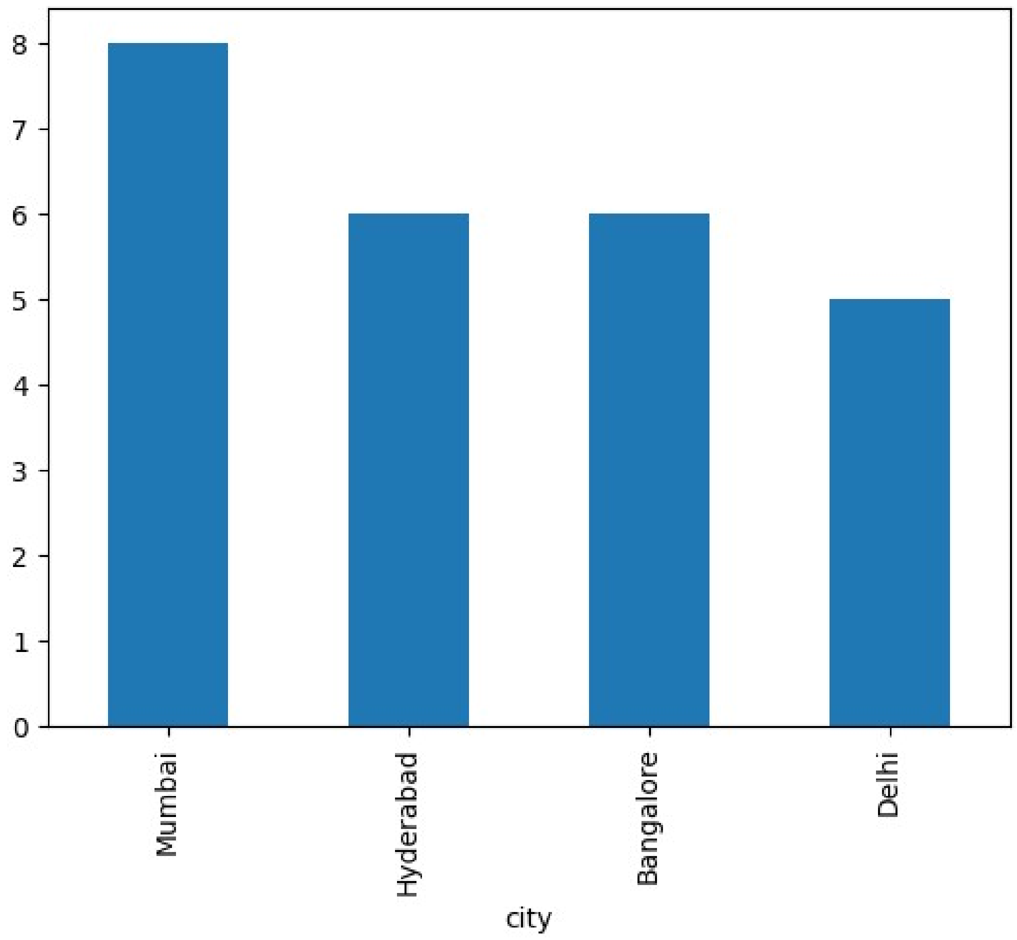
- now i want to know how many hotels we have in city wise

```
df_hotels.city.value_counts().sort_values()
```

```
city
Delhi        5
Hyderabad    6
Bangalore    6
Mumbai       8
Name: count, dtype: int64
```

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

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



Explore aggregate bookings ***

```
df_agg_bookings.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
```

```
df_agg_bookings.describe()
```

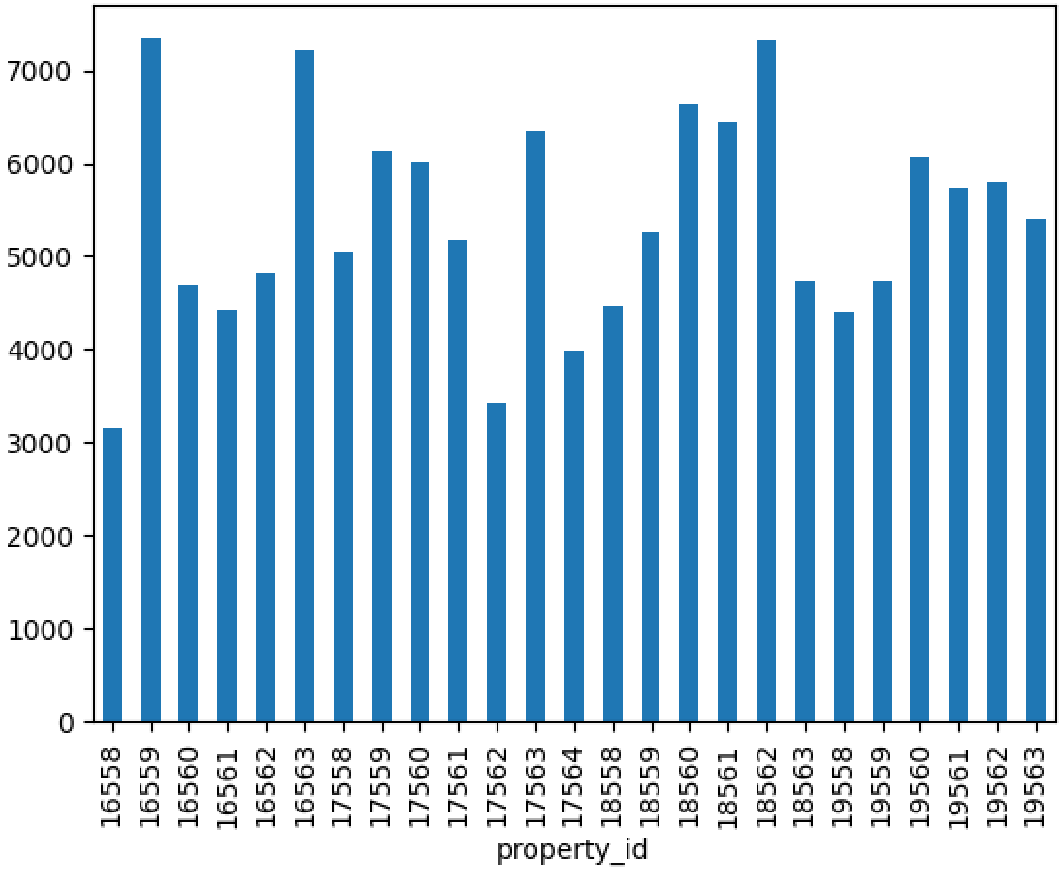
```
        property_id  successful_bookings     capacity
count   9200.000000          9200.000000  9198.000000
mean   18040.640000            14.655761    25.280496
std     1099.818325             7.736170    11.442080
min    16558.000000             1.000000     3.000000
```

```
25%    17558.000000            9.000000    18.000000
50%    17564.000000           14.000000    25.000000
75%    18563.000000           19.000000    34.000000
max    19563.000000          123.000000    50.000000
```

max capacity is 50 , min is 3 , mean 25

```
# now i want unique property id
print(df_agg_bookings.property_id.unique())

[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]

# so i want to see total booking as per property
df_agg_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

df_agg_bookings.groupby("property_id")
["successful_bookings"].sum().plot(kind='bar')
```

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



```
# lets find out in which dates we get booking then our capacity
df_agg_bookings[df_agg_bookings.successful_bookings>df_agg_bookings.ca
pacity]
```

```
      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
```

6 times we get lot of bookings that it is greater than our capacity

```
# let find out the property with highest capacity
print(df_agg_bookings.capacity.max())

50.0

df_agg_bookings[(df_agg_bookings.capacity ==
df_agg_bookings.capacity.max())]

      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 x 5 columns]
```

================================

data cleaning

================================

```
df_booking.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
```

- so i can clearly see that in no_guest , the min no of guest is -17 ---so this is a error

```
# unvalid record which are negative
df_booking[df_booking.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_booking.shape
```

```
(134590, 12)
```

```
# storing the valid data frame
df_booking= df_booking[df_booking.no_guests>=0]
df_booking.shape
```

```
(134578, 12)
```

i removed all the data rows which have -ve guests

```
df_booking.describe()
```

```
         property_id      no_guests  ratings_given  revenue_generated
\
count  134578.000000  134578.000000   56679.000000       1.345780e+05

mean    18061.143315       2.036744       3.619048       1.537804e+04

std      1093.053454       1.031710       1.234970       9.304015e+04

min     16558.000000       1.000000       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     134578.000000
mean       12696.011822
std         6927.841641
min         2600.000000
25%         7600.000000
50%        11700.000000
75%        15300.000000
max        45220.000000
```

as i saw early that i have max revenue generated on sinagle booking is too large so now i want to handel that error

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

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

# avg revenue generated  # standered deviation
avg , std =
df_booking.revenue_generated.mean(),df_booking.revenue_generated.std()
avg,std

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

# 3 standered deviation
higher_limit = avg + 3*std
higher_limit

np.float64(294498.50173207896)

# 3 standered deviation
lower_limit = avg - 3*std
lower_limit

np.float64(-263742.4278567056)
```

revenue should not be negative

```
df_booking[df_booking.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: []
```

so there is no -ve revenue

```
# revenue  greater than my higher limit , so this will show mw all my
outliers
df_booking[df_booking.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  
```

```
# clearing all wrong value which are higher than the limit
df_booking=df_booking[df_booking.revenue_generated<higher_limit]
df_booking
```

```
                 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]
```

```python
df_booking.shape
```

```
(134573, 12)
```

```python
# describing the column revenue realized of dataframe booking
df_booking.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
```

is my max and min value is correct , to know this i have do std again

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

np.float64(33479.358661845814)

lower_limit_limit =df_booking.revenue_realized.mean() - 
3*df_booking.revenue_realized.std()
lower_limit_limit

np.float64(-8087.391491611072)
```

so i'm getting my higher limit 33479 , but in dtaframe max is 45220 , that means this is a outlier

- but in a luxery hotel 45k for 1 night is okay

```
#  so i want to know how many revenue_realized is greater than my 
higher limit
df_booking[df_booking.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]
```

so there is 1299 times my revenue is greater than the higher limit

- the room category is rt4

```
df_rooms

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

RT4 IS A PRESEDENTIAL ROOM WHICH IS EXPENSIVE

```
#  now i will get std just for RT4 rooms
df_booking[df_booking.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

# mean + 3*std
23439 + 3*9048

50583
```

so the max limit is 50589 , and my max revenue from RT4 is 45220 that means it is not a outlier

- now it's time handel the NAN VALUE

```
# this will give true and flase
df_booking.isnull()

        booking_id  property_id  booking_date  check_in_date
checkout_date  \
1            False        False         False          False
False
4            False        False         False          False
False
5            False        False         False          False
False
6            False        False         False          False
False
7            False        False         False          False
False
...            ...          ...           ...            ...
...
134584       False        False         False          False
False
134585       False        False         False          False
False
134587       False        False         False          False
False
134588       False        False         False          False
False
134589       False        False         False          False
False

        no_guests  room_category  booking_platform  ratings_given  \
1           False          False             False           True
4           False          False             False          False
5           False          False             False          False
```

```
6           False          False             False           True
7           False          False             False           True
...           ...            ...               ...            ...
134584      False          False             False          False
134585      False          False             False          False
134587      False          False             False           True
134588      False          False             False          False
134589      False          False             False           True

        booking_status  revenue_generated  revenue_realized
1                False              False             False
4                False              False             False
5                False              False             False
6                False              False             False
7                False              False             False
...                ...                ...               ...
134584           False              False             False
134585           False              False             False
134587           False              False             False
134588           False              False             False
134589           False              False             False

[134573 rows x 12 columns]
```

```
# i want to ckeck the sum of true col wise
df_booking.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
```

So i have ratings_given is null value in 77897 times

- sometimes people forgot to give feedback or they don't want to give it , so it is fine , i don't need to handel this type of nan values

Total values in our dataframe is 134576. Out of that 77899 rows has null rating. Since there are many rows with null rating, we should not filter these values. Also we should not replace this rating with a median or mean rating etc

```
# checking my df_agg_bookings has null
df_agg_bookings.isnull().sum()

property_id            0
check_in_date          0
room_category          0
successful_bookings    0
capacity               2
dtype: int64
```

so capacity has null value , but it can't be null , so i can replace this with median

```
df_agg_bookings.capacity.median()

np.float64(25.0)

df_agg_bookings[df_agg_bookings.capacity.isna()]

    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

# filling the nan values with median and using inplace storing the
value directly
df_agg_bookings.capacity.fillna(df_agg_bookings.capacity.median(),inpl
ace=True)

df_agg_bookings.loc[[8,14]]

    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

df_agg_bookings[df_agg_bookings.capacity.isna()]

Empty DataFrame
Columns: [property_id, check_in_date, room_category,
successful_bookings, capacity]
Index: []
```

- as we saw early that my succesefull booking is greater than capacity so i have to filter that out

```
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

[9200 rows x 5 columns]
```

```
df_agg_bookings[df_agg_bookings.successful_bookings>df_agg_bookings.ca
pacity]
```

```
      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
```

```
df_agg_bookings.shape
```

```
(9200, 5)
```

```
df_agg_bookings =
df_agg_bookings[df_agg_bookings.successful_bookings<=df_agg_bookings.c
apacity]
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
4           16558      1-May-22           RT1                   18
19.0
5           17560      1-May-22           RT1                   28
40.0
...           ...           ...           ...                  ...
...
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

[9194 rows x 5 columns]
```

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
4        16558      1-May-22           RT1                   18
19.0
```

```
5        17560      1-May-22            RT1                      28
40.0

# occupancy percentage  = successefull_bookings / capacity

occ = [df_agg_bookings.successful_bookings/df_agg_bookings.capacity]

# now i will create a column inside my data occupancy_percentage
df_agg_bookings["occ_pct"] =
df_agg_bookings["successful_bookings"]/df_agg_bookings["capacity"]

C:\Users\ayush\AppData\Local\Temp\ipykernel_22400\56733706.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  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
4        16558      1-May-22            RT1                      18
19.0
5        17560      1-May-22            RT1                      28
40.0

    occ_pct
0  0.833333
1  0.933333
2  0.766667
4  0.947368
5  0.700000
```

occ_pct is in float , but i don't want that

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

```
C:\Users\ayush\AppData\Local\Temp\ipykernel_22400\2924502782.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_agg_bookings["occ_pct"] =
df_agg_bookings["occ_pct"] .apply(lambda x: round(x*100,2))

   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

   occ_pct
0    83.33
1    93.33
2    76.67
4    94.74
5    70.00
```

```
df_agg_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9194 entries, 0 to 9199
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   property_id          9194 non-null   int64
 1   check_in_date        9194 non-null   object
 2   room_category        9194 non-null   object
 3   successful_bookings  9194 non-null   int64
 4   capacity             9194 non-null   float64
 5   occ_pct              9194 non-null   float64
dtypes: float64(2), int64(2), object(2)
memory usage: 502.8+ KB
```

Insights Generation

what is an avg occupancy rate in each of thr room categoris?

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

   occ_pct
0    83.33
1    93.33
2    76.67
4    94.74
5    70.00

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

room_category
RT1    57.89
RT2    58.01
RT3    58.03
RT4    59.28
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='r
oom_id')
df.sample(5)

      property_id check_in_date room_category  successful_bookings
capacity  \
559         17561      6-May-22          RT3                   13
19.0
7362        16560     13-Jul-22          RT3                   10
20.0
7075        19558     10-Jul-22          RT4                    4
7.0
7344        19561     13-Jul-22          RT2                   24
45.0
7674        17561     16-Jul-22          RT4                    4
4.0

      occ_pct room_id    room_class
559     68.42     RT3       Premium
7362    50.00     RT3       Premium
7075    57.14     RT4  Presidential
7344    53.33     RT2         Elite
7674   100.00     RT4  Presidential
```

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

```
room_class
Elite           58.01
Premium         58.03
Presidential    59.28
Standard        57.89
Name: occ_pct, dtype: float64
```

```
df.drop('room_id', axis = 1,inplace=True)
df.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        16558      1-May-22           RT1                   18
19.0
4        17560      1-May-22           RT1                   28
40.0

   occ_pct room_class
0    83.33   Standard
1    93.33   Standard
2    76.67   Standard
3    94.74   Standard
4    70.00   Standard
```

print avg occupancy rate per city

```
df_hotels.head(3)
```

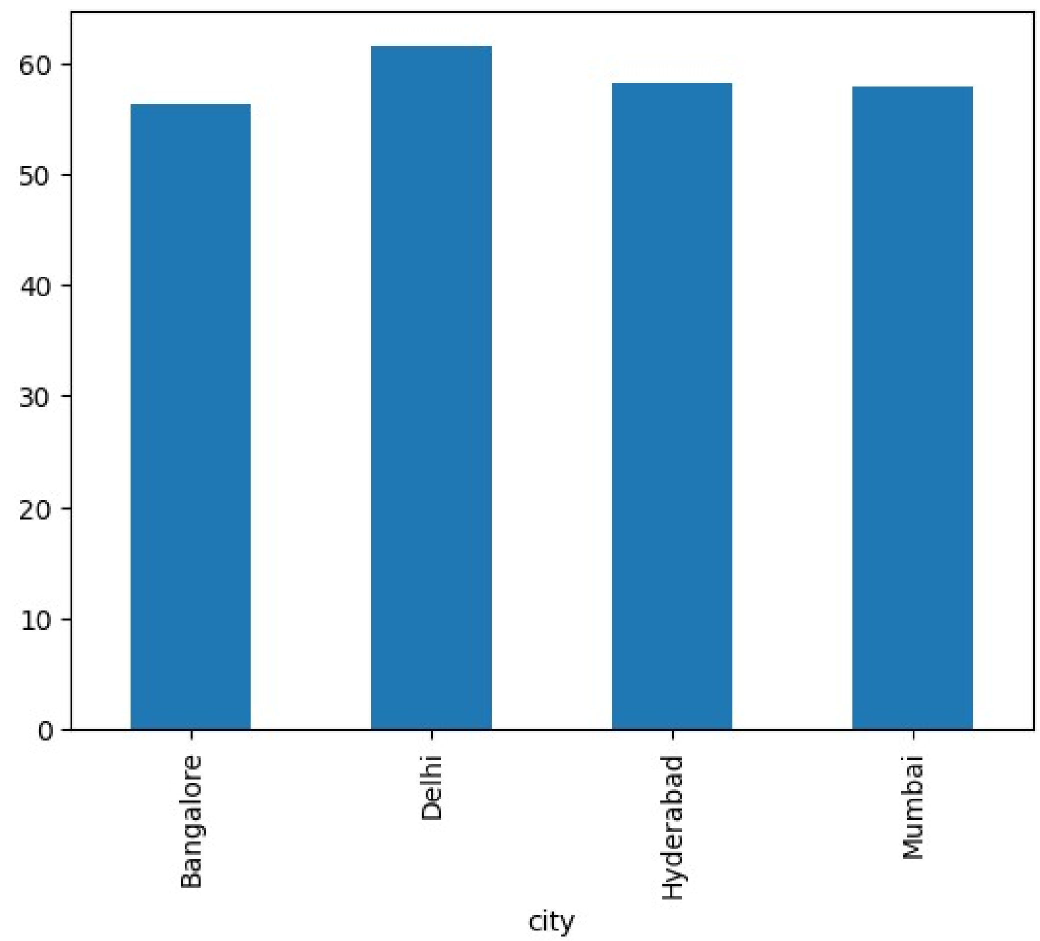
```
   property_id  property_name  category    city
0        16558   Atliq Grands    Luxury   Delhi
1        16559  Atliq Exotica    Luxury  Mumbai
2        16560     Atliq City  Business   Delhi
```

```
df = pd.merge(df,df_hotels, on= 'property_id')
df.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        16558      1-May-22           RT1                   18
19.0
4        17560      1-May-22           RT1                   28
40.0

   occ_pct room_class  property_name  category       city
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    94.74   Standard   Atliq Grands    Luxury      Delhi
4    70.00   Standard     Atliq City  Business     Mumbai
```

```
df.groupby('city')['occ_pct'].mean().plot(kind='bar')
```

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



when was the occupancy better? weekday or weekend?

```
df.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        16558      1-May-22           RT1                   18
19.0
4        17560      1-May-22           RT1                   28
40.0

   occ_pct room_class  property_name  category       city
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    94.74   Standard   Atliq Grands     Luxury      Delhi
4    70.00   Standard     Atliq City   Business     Mumbai

df_date.head()

        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.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_class property_name  category       city       date
mmm yy  \
0    51.72    Premium  Atliq Palace  Business  Bangalore  10-May-22
May 22
1    63.33   Standard    Atliq City  Business  Hyderabad  10-May-22
May 22
2    60.00   Standard     Atliq Bay    Luxury  Bangalore  10-May-22
May 22

  week no  day_type
0    W 20  weekeday
1    W 20  weekeday
2    W 20  weekeday

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

day_type
weekeday    50.88
weekend     72.34
Name: occ_pct, dtype: float64
```

in the month of june , what is the occupency fro different cities

```
df['mmm yy'].unique()

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

```
df_june22 = df[df['mmm yy']== 'Jun 22']
df_june22.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_class  property_name  category       city
date  \
2200    66.67   Standard  Atliq Exotica    Luxury     Mumbai  10-Jun-
22
2201    63.33   Standard      Atliq Bay    Luxury  Bangalore  10-Jun-
22
2202    56.67   Standard   Atliq Palace  Business  Bangalore  10-Jun-
22
2203    47.37   Standard   Atliq Grands    Luxury     Mumbai  10-Jun-
22

      mmm yy week no  day_type
2200  Jun 22    W 24  weekeday
2201  Jun 22    W 24  weekeday
2202  Jun 22    W 24  weekeday
2203  Jun 22    W 24  weekeday
```

```
df_june22.groupby('city')
['occ_pct'].mean().round(2).sort_values(ascending=False)
```

```
city
Delhi        62.47
Hyderabad    58.46
Mumbai       58.38
Bangalore    56.44
Name: occ_pct, dtype: float64
```

suppose i suddenly get the new data file of august

```
df_august = pd.read_csv("datasets/new_data_august.csv")
df_august.head(3)
```

```
   property_id  property_name  category       city room_category
room_class  \
0        16559  Atliq Exotica    Luxury     Mumbai           RT1
Standard
1        19562      Atliq Bay    Luxury  Bangalore           RT1
Standard
2        19563   Atliq Palace  Business  Bangalore           RT1
Standard

  check_in_date  mmm yy week no  day_type  successful_bookings
capacity  \
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

     occ%
0  100.00
1   70.00
2   76.67
```

```
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.columns
```

```
Index(['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'],
      dtype='object')
```

```
df_august.shape
```

```
(7, 13)
```

```
df.shape
```

```
(6497, 14)
```

```
latest_df = pd.concat([df,df_august],ignore_index=True,axis=0)
latest_df.tail(8)
```

```
      property_id check_in_date room_category  successful_bookings
capacity  \
6496        17561     31-Jul-22           RT4                    3
4.0
6497        16559     01-Aug-22           RT1                   30
30.0
6498        19562     01-Aug-22           RT1                   21
30.0
6499        19563     01-Aug-22           RT1                   23
30.0
6500        19558     01-Aug-22           RT1                   30
40.0
6501        19560     01-Aug-22           RT1                   20
26.0
6502        17561     01-Aug-22           RT1                   18
26.0
6503        17564     01-Aug-22           RT1                   10
16.0

      occ_pct    room_class  property_name  category       city
date  \
6496     75.0  Presidential      Atliq Blu    Luxury     Mumbai  31-
Jul-22
6497      NaN      Standard  Atliq Exotica    Luxury     Mumbai
NaN
6498      NaN      Standard      Atliq Bay    Luxury  Bangalore
NaN
6499      NaN      Standard   Atliq Palace  Business  Bangalore
NaN
6500      NaN      Standard   Atliq Grands    Luxury  Bangalore
NaN
6501      NaN      Standard     Atliq City  Business  Bangalore
NaN
6502      NaN      Standard      Atliq Blu    Luxury     Mumbai
NaN
6503      NaN      Standard  Atliq Seasons  Business     Mumbai
NaN

      mmm yy week no  day_type    occ%
6496  Jul 22    W 32   weekend     NaN
6497  Aug-22    W 32  weekeday  100.00
6498  Aug-22    W 32  weekeday   70.00
6499  Aug-22    W 32  weekeday   76.67
6500  Aug-22    W 32  weekeday   75.00
6501  Aug-22    W 32  weekeday   76.92
6502  Aug-22    W 32  weekeday   69.23
6503  Aug-22    W 32  weekeday   62.50
```

print revenue realized per city

```
df_booking.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
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

   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

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

```
df_hotels.head(3)
```

```
   property_id  property_name  category    city
0        16558   Atliq Grands    Luxury   Delhi
1        16559  Atliq Exotica    Luxury  Mumbai
2        16560     Atliq City  Business   Delhi
```

```
df_booking_all= pd.merge(df_booking,df_hotels,on ="property_id")
df_booking_all.head(3)
```

```
         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

   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

   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
```

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

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