1. Data Import

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from matplotlib import style
```

In [2]:
```
dim_date=pd.read_excel('C:/Users/payal/Desktop/Payal/Internkaksha Project/Atli
dim_date.head(3)
```

Out[2]:

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

In [3]:
```
dim_date.shape
```

Out[3]: (92, 4)

In [4]:
```
dim_hotels=pd.read_excel('C:/Users/payal/Desktop/Payal/Internkaksha Project/At
dim_hotels.head(3)
```

Out[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

In [5]:
```
dim_rooms=pd.read_excel('C:/Users/payal/Desktop/Payal/Internkaksha Project/Atl
dim_rooms.head(3)
```

Out[5]:

	room_id	room_class
0	RT1	Standard
1	RT2	Elite
2	RT3	Premium

In [6]:
```
fact_aggregated_bookings=pd.read_excel('C:/Users/payal/Desktop/Payal/Internkak
fact_aggregated_bookings.head(3)
```

Out[6]:

	property_id	check_in_date	room_category	successful_bookings	capacity
0	16559	2022-05-01	RT1	25	30
1	19562	2022-05-01	RT1	28	30
2	19563	2022-05-01	RT1	23	30

In [7]:
```
fact_aggregated_bookings.shape
```

Out[7]:
```
(9200, 5)
```

In [8]:
```
fact_aggregated_bookings.room_category.unique()
```

Out[8]:
```
array(['RT1', 'RT2', 'RT3', 'RT4'], dtype=object)
```

In [9]:
```
fact_aggregated_bookings.groupby("room_category").successful_bookings.sum().so
```

Out[9]:
```
room_category
RT2    49505
RT1    38446
RT3    30566
RT4    16073
Name: successful_bookings, dtype: int64
```

In [10]:
```
fact_aggregated_bookings.groupby("room_category").capacity.sum().sort_values(a
```

Out[10]:
```
room_category
RT2    85928
RT1    66424
RT3    53084
RT4    27140
Name: capacity, dtype: int64
```

In [11]:
```
fact_aggregated_bookings.groupby("property_id").capacity.sum().sort_values(asc
```

Out[11]:
```
property_id
18559    11776
17560    11316
16559    11132
18562    11132
19561    10764
16563    10764
19563    10120
18560    10028
18561     9844
19558     9844
17563     9568
17558     9384
17559     9292
19560     9108
16562     9016
18563     8924
17564     8924
19562     8832
19559     8740
16560     8740
18558     8372
17561     7820
17562     7636
16561     6716
16558     4784
Name: capacity, dtype: int64
```

In [12]:
```
fact_bookings=pd.read_excel('C:/Users/payal/Desktop/Payal/Internkaksha Project
fact_bookings.head(3)
```

Out[12]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	roo
0	May012216558RT11	16558	2022-04-27	2022-05-01	2022-05-02	3	
1	May012216558RT12	16558	2022-04-30	2022-05-01	2022-05-02	2	
2	May012216558RT13	16558	2022-04-28	2022-05-01	2022-05-04	2	

In [13]:
```
fact_bookings.shape
```

Out[13]:
```
(134590, 12)
```

In [14]:
```
fact_bookings.booking_platform.unique()
```

Out[14]:
```
array(['direct online', 'others', 'logtrip', 'tripster', 'makeyourtrip',
       'journey', 'direct offline'], dtype=object)
```

In [15]:
```
fact_bookings.booking_platform.value_counts().sort_values(ascending=False)
```

Out[15]:
```
others            55066
makeyourtrip      26898
logtrip           14756
direct online     13379
tripster           9630
journey            8106
direct offline     6755
Name: booking_platform, dtype: int64
```

In [16]:
```
metrics_list=pd.read_excel('C:/Users/payal/Desktop/Payal/Internkaksha Project/
metrics_list.head(3)
```

Out[16]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2
0	Sno	Measuers	Description
1	1	Revenue	Sum of revenue_realized
2	2	Total Bookings	Count of booking_id in fact_bookings

2. Data Cleaning

In [17]:
```
fact_bookings.describe()
```

Out[17]:

	property_id	no_guests	ratings_given	revenue_generated	revenue_realized
count	134590.000000	134590.000000	56683.000000	134590.000000	134590.000000
mean	18061.113493	2.036808	3.619004	14916.013188	12696.123256
std	1093.055847	1.031766	1.235009	6452.868072	6928.108124
min	16558.000000	1.000000	1.000000	6500.000000	2600.000000
25%	17558.000000	1.000000	3.000000	9900.000000	7600.000000
50%	17564.000000	2.000000	4.000000	13500.000000	11700.000000
75%	18563.000000	2.000000	5.000000	18000.000000	15300.000000
max	19563.000000	6.000000	5.000000	45220.000000	45220.000000

In [18]:
```
fact_bookings[fact_bookings.revenue_realized==fact_bookings.revenue_realized.m
```

Out[18]:
```
array(['RT4'], dtype=object)
```

In [19]:
```
fact_bookings[fact_bookings.room_category=="RT4"].revenue_realized.describe()
```

Out[19]:
```
count    16073.000000
mean     23440.103652
std       9048.865206
min       7600.000000
25%      19000.000000
50%      26600.000000
75%      32300.000000
max      45220.000000
Name: revenue_realized, dtype: float64
```

 # to understand whether max revenue is an outlier or not

In [20]:
```
fact_bookings[fact_bookings.room_category=="RT4"].revenue_realized.mean()+\
3*fact_bookings[fact_bookings.room_category=="RT4"].revenue_realized.std()
```

Out[20]:
```
50586.69926930697
```

as our maximum revenue value (Rs.45220)< Rs.50586, it is OK

In [21]:
```
fact_bookings.isna().sum()
```

Out[21]:
```
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       77907
booking_status          0
revenue_generated       0
revenue_realized        0
dtype: int64
```

In [22]:
```
fact_aggregated_bookings.isnull().sum()
```

Out[22]:
```
property_id           0
check_in_date         0
room_category         0
successful_bookings   0
capacity              0
dtype: int64
```

3. Data Transformation and Visualization

3.1 Bookings per Room category

In [23]:
```
successful_bookings=fact_aggregated_bookings.groupby("room_category").successf
capacity=fact_aggregated_bookings.groupby("room_category").capacity.sum()
```

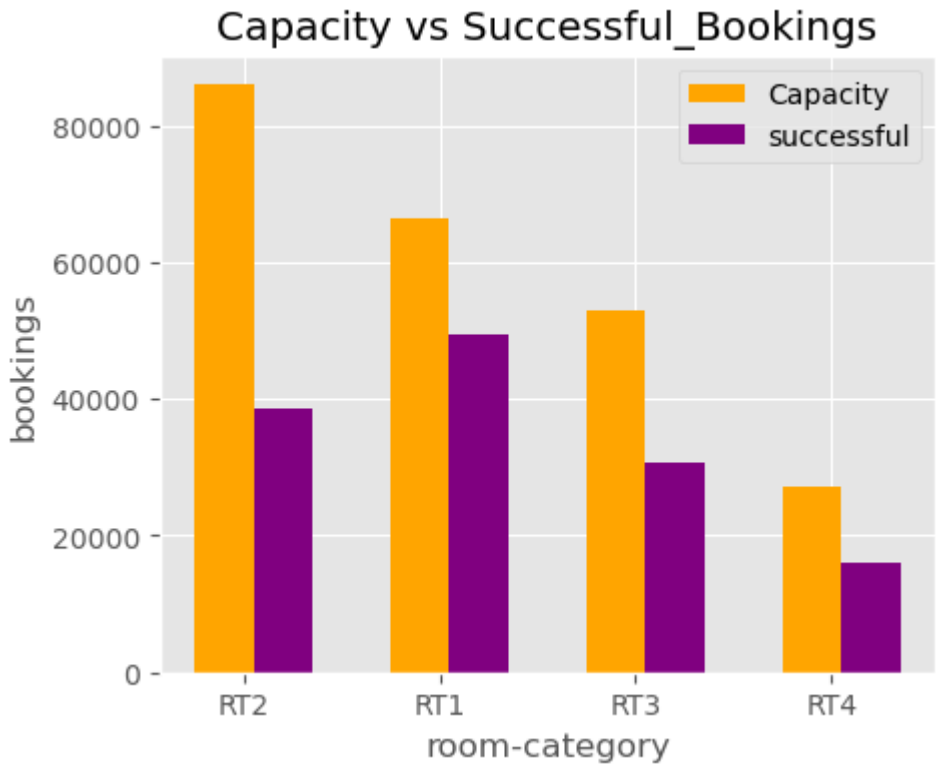
In [24]:
```
capacity=fact_aggregated_bookings.groupby("room_category").capacity.sum().sort
```

In [25]:
```
xplot=np.arange(4)
xplot
```

Out[25]:
```
array([0, 1, 2, 3])
```

In [26]:
```
style.use("ggplot")
plt.figure(figsize=(5,4))
plt.bar(xplot,capacity,width=0.3,color="orange",label="Capacity")
plt.bar(xplot+0.3,successful_bookings,width=0.3,color="purple",label="successf
plt.xticks(xplot+0.1,["RT2","RT1","RT3","RT4"])
plt.xlabel("room-category")
plt.ylabel("bookings")
plt.title("Capacity vs Successful_Bookings")
plt.legend()
```

Out[26]:
```
<matplotlib.legend.Legend at 0x1d0f47023d0>
```



3.2 Average Rating by City

merge two files fact_bookings and dim_hotels

In [27]:
```
fact_bookings_hotel=pd.merge(fact_bookings,dim_hotels,on="property_id")
fact_bookings_hotel.head(3)
```

Out[27]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	roo
0	May012216558RT11	16558	2022-04-27	2022-05-01	2022-05-02	3	
1	May012216558RT12	16558	2022-04-30	2022-05-01	2022-05-02	2	
2	May012216558RT13	16558	2022-04-28	2022-05-01	2022-05-04	2	

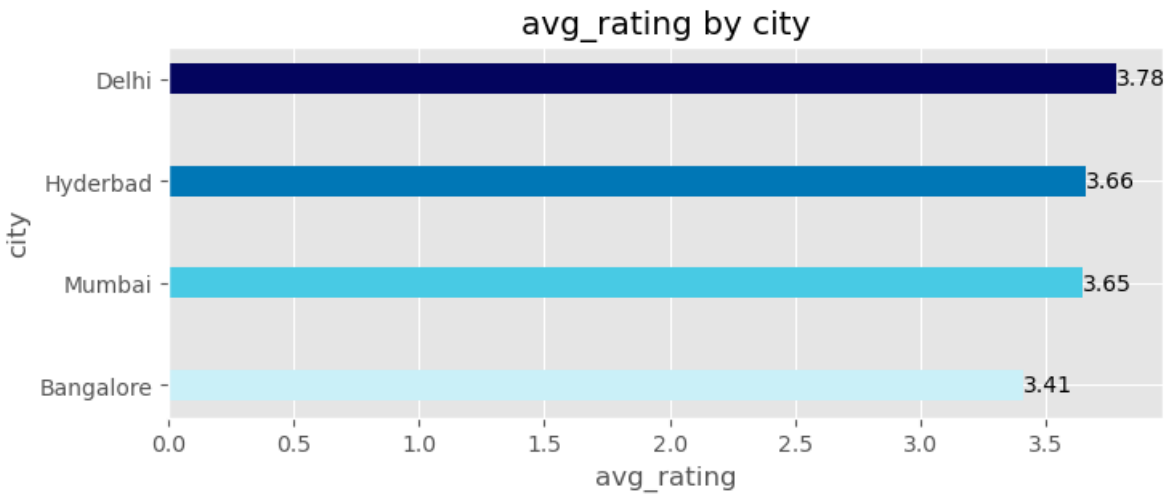
In [28]:
```
fact_bookings_hotel.shape
```

Out[28]:
```
(134590, 15)
```

In [29]:
```
avg_rating=fact_bookings_hotel.groupby("city").ratings_given.mean().round(2).s
avg_rating
```

Out[29]:
```
city
Bangalore    3.41
Mumbai       3.65
Hyderabad    3.66
Delhi        3.78
Name: ratings_given, dtype: float64
```

In [30]:
```
style.use("ggplot")
plt.figure(figsize=(8,3))
c=["#CAF0F8","#48CAE4","#0077B6","#03045E"]
plt.barh(xplot,avg_rating,height=0.3,color=c)
plt.yticks(xplot,["Bangalore","Mumbai","Hyderbad","Delhi"])
plt.xlabel("avg_rating")
plt.ylabel("city")
plt.title("avg_rating by city")
for index,value in enumerate(avg_rating):
    plt.text(value,index,str(value),va="center")
```

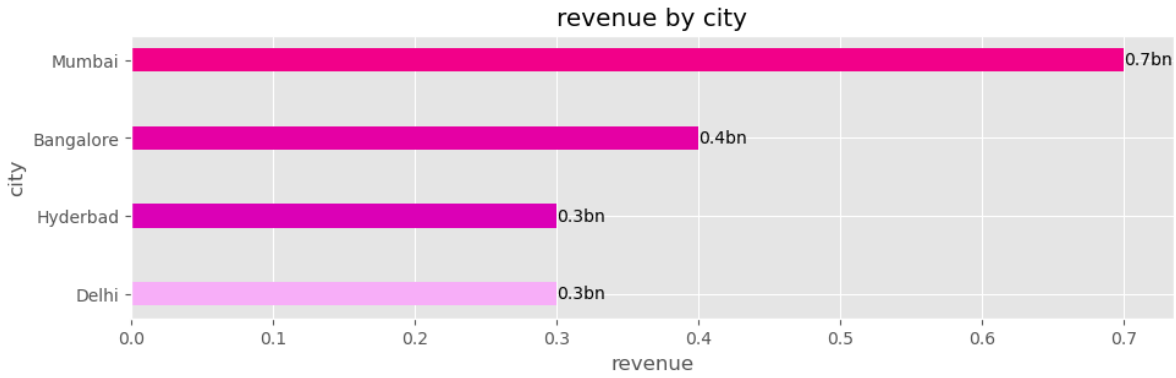


3.3 Revenue by city

In [31]:
```
revenue=fact_bookings_hotel.groupby("city").revenue_realized.sum().sort_values
revenue=round(revenue/1e9,1)
revenue
```

Out[31]:
```
city
Delhi        0.3
Hyderabad    0.3
Bangalore    0.4
Mumbai       0.7
Name: revenue_realized, dtype: float64
```

In [32]:
```
style.use("ggplot")
plt.figure(figsize=(11,3))
c2=["#F7AEF8","#DB00B6","#E500A4","#F20089"]
plt.barh(xplot,revenue,height=0.3,color=c2)
plt.yticks(xplot,["Delhi","Hyderbad","Bangalore","Mumbai"])
plt.xlabel("revenue")
plt.ylabel("city")
plt.title("revenue by city")
for index,value in enumerate(revenue):
    plt.text(value,index,str(value)+"bn",va="center")
```



3.4 Occupancy % by city

merge two files fact_aggregated_bookings and dim_hotels

In [33]:
```
fact_aggregated_bookings_city=pd.merge(fact_aggregated_bookings,dim_hotels,on=
fact_aggregated_bookings_city.head(3)
```

Out[33]:

	property_id	check_in_date	room_category	successful_bookings	capacity	property_name	ca
0	16559	2022-05-01	RT1	25	30	Atliq Exotica	
1	16559	2022-05-01	RT2	35	41	Atliq Exotica	
2	16559	2022-05-01	RT3	27	32	Atliq Exotica	

In [34]:
```
successful_bookings=fact_aggregated_bookings_city.groupby("city").successful_b
successful_bookings
```

Out[34]:
```
city
Bangalore    32016
Delhi        24231
Hyderabad    34888
Mumbai       43455
Name: successful_bookings, dtype: int64
```

In [35]:
```
capacity_of_bookings=fact_aggregated_bookings_city.groupby("city").capacity.su
capacity_of_bookings
```

Out[35]:
```
city
Bangalore    57408
Delhi        40020
Hyderabad    60076
Mumbai       75072
Name: capacity, dtype: int64
```

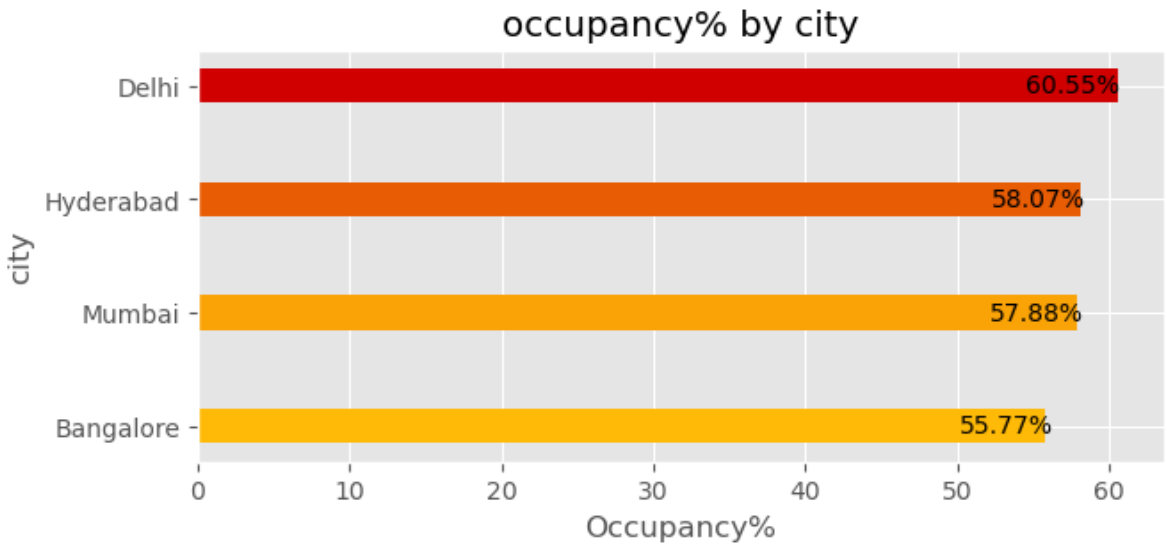
In [36]:
```
occupancy=round((successful_bookings/capacity_of_bookings)*100,2).sort_values(
occupancy
```

Out[36]:
```
city
Bangalore    55.77
Mumbai       57.88
Hyderabad    58.07
Delhi        60.55
dtype: float64
```

In [37]:
```
yplot=np.arange(4)
yplot
```

Out[37]:
```
array([0, 1, 2, 3])
```

In [38]:
```
style.use("ggplot")
plt.figure(figsize=(7,3))
c1=["#FFBA08","#FAA307","#E85D04","#D00000"]
x=["Bangalore","Mumbai","Hyderabad","Delhi"]
plt.barh(yplot,occupancy,color=c1,height=0.3)
plt.yticks(yplot,x)
plt.xlabel("Occupancy%")
plt.ylabel("city")
plt.title("occupancy% by city")
for index,value in enumerate(occupancy):
    plt.text(value*0.90,index,str(value)+"%",va="center")
```



3.5 Occupancy % by day_type

In [72]:
```
successful_bookings1=fact_aggregated_bookings_and_dim_date.groupby("day_type")
successful_bookings1
capacity_of_bookings1=fact_aggregated_bookings_and_dim_date.groupby("day_type"
capacity_of_bookings1
occupancy1=round((successful_bookings1/capacity_of_bookings1)*100,2)
occupancy1
```

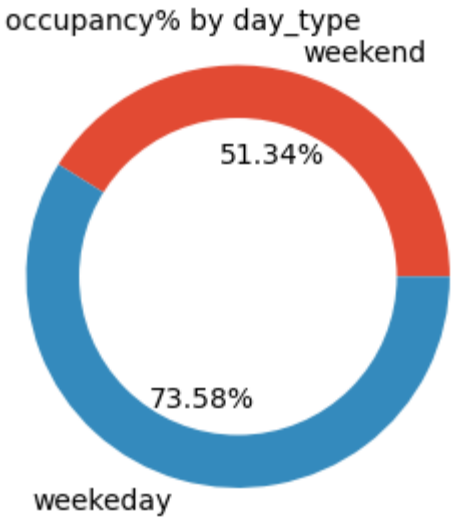
Out[72]:
```
day_type
weekeday    51.34
weekend     73.58
dtype: float64
```

In [74]:
```
day_type=fact_aggregated_bookings_and_dim_date.day_type.unique()
day_type
```

Out[74]:
```
array(['weekend', 'weekeday'], dtype=object)
```

In [75]:
```
style.use("ggplot")
plt.figure(figsize=(3,3))
plt.pie(x=occupancy1,autopct=lambda x: '{:.2%}'.format(x*occupancy1.sum()/1000
plt.axis("equal")
plt.title("occupancy% by day_type",size=10,loc="left")
circle=plt.Circle(xy=(0,0),radius=0.75,facecolor="white")
plt.gca().add_artist(circle)
```

Out[75]: <matplotlib.patches.Circle at 0x1d0f9850f90>



3.6 Booking % by Platform

In [41]:
```
fact_bookings.head(3)
```

Out[41]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	roo
0	May012216558RT11	16558	2022-04-27	2022-05-01	2022-05-02	3	
1	May012216558RT12	16558	2022-04-30	2022-05-01	2022-05-02	2	
2	May012216558RT13	16558	2022-04-28	2022-05-01	2022-05-04	2	

In [42]:
```
total_bookings=fact_bookings["booking_id"].count()
total_bookings
```

Out[42]: 134590

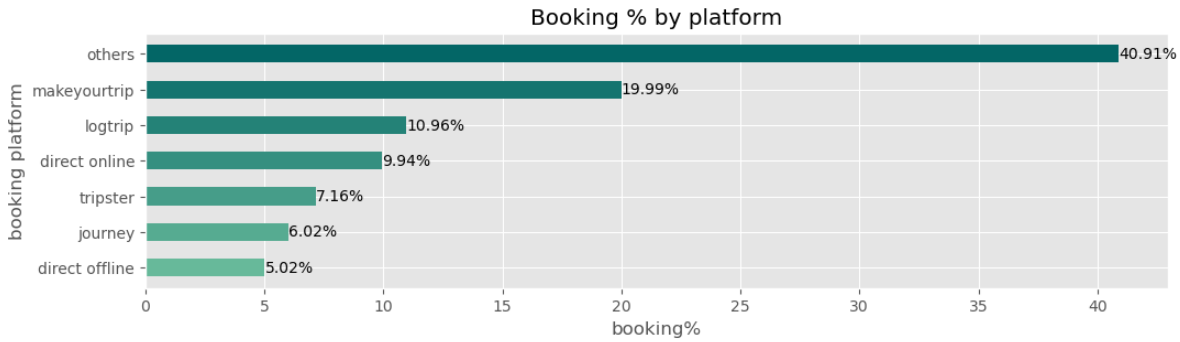
In [43]:
```
total_booking_by_platform=fact_bookings.groupby('booking_platform').booking_id
total_booking_by_platform
```

Out[43]:
```
booking_platform
direct offline     6755
direct online     13379
journey            8106
logtrip           14756
makeyourtrip      26898
others            55066
tripster           9630
Name: booking_id, dtype: int64
```

In [44]:
```
Booking_by_platform=round((total_booking_by_platform/total_bookings)*100,2)
Booking_by_platform=Booking_by_platform.sort_values()
Booking_by_platform
```

Out[44]:
```
booking_platform
direct offline     5.02
journey            6.02
tripster           7.16
direct online      9.94
logtrip           10.96
makeyourtrip      19.99
others            40.91
Name: booking_id, dtype: float64
```

In [45]:
```
yplot=np.arange(7)
plt.figure(figsize=(12,3))
style.use("ggplot")
x=["direct offline","journey","tripster","direct online","logtrip","makeyourtr
c=["#67B99A","#56AB91","#469D89","#358F80","#248277","#14746F","#036666"]
plt.barh(yplot,Booking_by_platform,color=c,height=0.5)
plt.yticks(yplot,x)
plt.ylabel("booking platform")
plt.xlabel("booking%")
plt.title("Booking % by platform")
for index,value in enumerate(Booking_by_platform):
    plt.text(value,index,str(value)+"%",va="center")
```



3.7 Revenue by City and Property name

In [46]:
```
import plotly.express as px
import plotly
```

merge two data 'fact_bookings_hotel' and 'dim_rooms'

In [47]:
```
fact_bookings_hotel_rooms=pd.merge(fact_bookings_hotel,dim_rooms,left_on="room
fact_bookings_hotel_rooms.head()
del fact_bookings_hotel_rooms["room_id"]
fact_bookings_hotel_rooms.head(3)
```

Out[47]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	roo
0	May012216558RT11	16558	2022-04-27	2022-05-01	2022-05-02	3	
1	May012216558RT12	16558	2022-04-30	2022-05-01	2022-05-02	2	
2	May012216558RT13	16558	2022-04-28	2022-05-01	2022-05-04	2	

In [48]:
```
city=fact_bookings_hotel_rooms["city"]
city.unique()
```

Out[48]:
```
array(['Delhi', 'Mumbai', 'Hyderabad', 'Bangalore'], dtype=object)
```

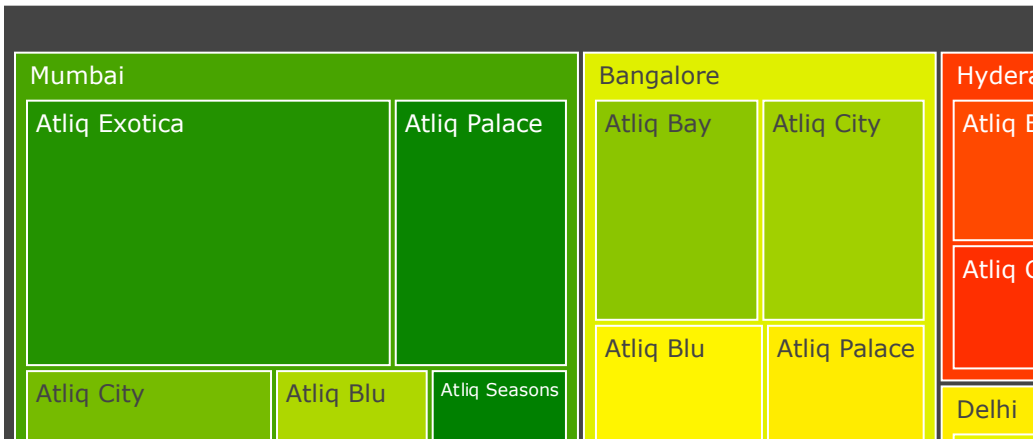
In [49]:
```
property_name=fact_bookings_hotel_rooms["property_name"]
property_name.unique()
```

Out[49]:
```
array(['Atliq Grands', 'Atliq Exotica', 'Atliq City', 'Atliq Blu',
       'Atliq Bay', 'Atliq Palace', 'Atliq Seasons'], dtype=object)
```

In [50]:
```
revenue_realized=fact_bookings_hotel_rooms["revenue_realized"]
```

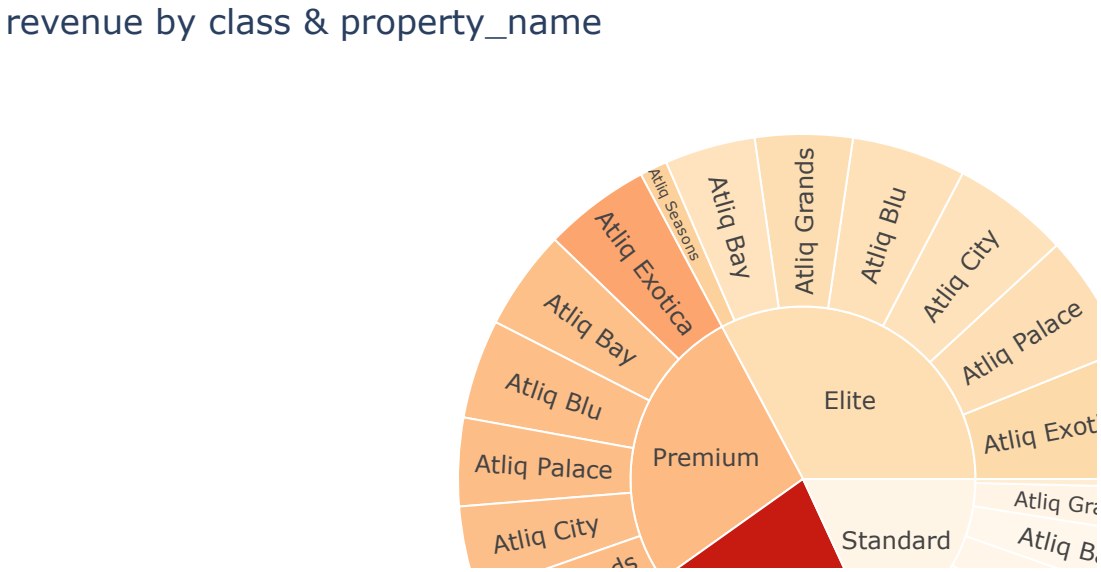
In [51]:
```
class_=fact_bookings_hotel_rooms["room_class"]
```

In [52]:
```
px.treemap(fact_bookings_hotel_rooms,path=[city,property_name],values=revenue_
color_continuous_scale=["red","yellow","green"],title="revenue by city & prope
```



3.8 Revenue by Class and Property name

In [53]:
```
px.sunburst(fact_bookings_hotel_rooms,path=[class_,property_name],values=reven
color_continuous_scale="orrd",title="revenue by class & property_name")
```



3.9 Realisation, Cancellation and No show percentage

In [54]:
```
total_cancelled_bookings=fact_bookings[fact_bookings.booking_status=="Cancelle
total_cancelled_bookings
```

Out[54]: 33420

In [55]:
```
total_bookings
print(total_bookings)
total_cancelled_bookings
print(total_cancelled_bookings)
```

```
134590
33420
```

In [56]:
```
Cancellation=round((total_cancelled_bookings/total_bookings)*100,2)
Cancellation
```

Out[56]:
```
24.83
```

In [57]:
```
total_noshow_bookings=fact_bookings[fact_bookings.booking_status=="No Show"].b
total_noshow_bookings
```

Out[57]:
```
6759
```

In [58]:
```
No_Show=round((total_noshow_bookings/total_bookings)*100,2)
No_Show
```

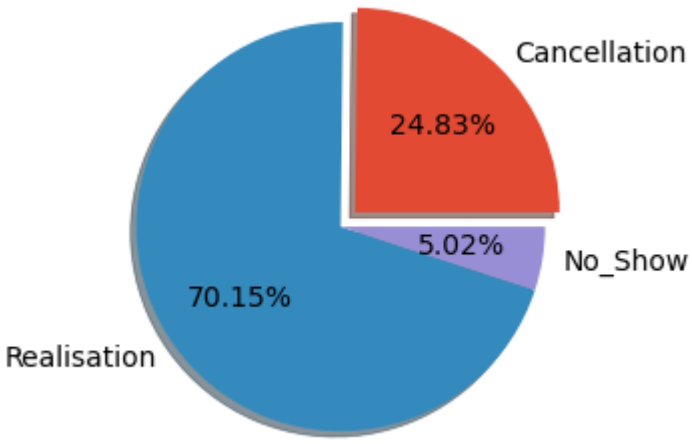
Out[58]:
```
5.02
```

In [59]:
```
Realisation=100-No_Show-Cancellation
Realisation
```

Out[59]:
```
70.15
```

In [60]:
```
values=[Cancellation,Realisation,No_Show,]
style.use("ggplot")
plt.figure(figsize=(3,3))
explode=(0.1,0,0)
plt.pie(x=values,explode=explode,shadow=True,autopct="%.2f%%",labels=["Cancell
plt.axis("equal")
```

Out[60]:
```
(-1.103554036774446,
 1.1746416626782494,
 -1.1035118765131204,
 1.173788640826471)
```



4.Occupancy % and Average Rating

In [61]:
```
fact_bookings_date=pd.merge(fact_bookings,dim_date,left_on="check_in_date",rig
fact_bookings_date.head(3)
```

Out[61]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	roo
0	May012216558RT11	16558	2022-04-27	2022-05-01	2022-05-02	3	
1	May012216558RT12	16558	2022-04-30	2022-05-01	2022-05-02	2	
2	May012216558RT13	16558	2022-04-28	2022-05-01	2022-05-04	2	

In [62]:
```
avg_rating=fact_bookings_date.groupby("week no").ratings_given.mean()
```

In [63]:
```
revenue_by_weekday=fact_bookings_date.groupby("week no").revenue_realized.sum(
revenue_by_weekday
```

Out[63]:
```
week no
W 19    138182064
W 20    139435920
W 21    114922175
W 22    138720126
W 23    115568569
W 24    139581703
W 25    138674279
W 26    114152421
W 27    139555632
W 28    139383916
W 29    139730590
W 30    114811148
W 31    115042325
W 32     21010361
Name: revenue_realized, dtype: int64
```

In [64]:
```
fact_aggregated_bookings_and_dim_date=pd.merge(fact_aggregated_bookings,dim_da
fact_aggregated_bookings_and_dim_date.head(3)
```

Out[64]:

	property_id	check_in_date	room_category	successful_bookings	capacity	date	mm
0	16559	2022-05-01	RT1	25	30	2022-05-01	2022-0
1	19562	2022-05-01	RT1	28	30	2022-05-01	2022-0
2	19563	2022-05-01	RT1	23	30	2022-05-01	2022-0

In [65]:
```
successful_bookings2=fact_aggregated_bookings_and_dim_date.groupby("week no").
successful_bookings2
capacity_of_bookings2=fact_aggregated_bookings_and_dim_date.groupby("week no")
capacity_of_bookings2
occupancy2=round((successful_bookings2/capacity_of_bookings2)*100,2)
occupancy2
```

Out[65]:
```
week no
W 19    61.96
W 20    61.92
W 21    51.10
W 22    61.79
W 23    51.36
W 24    62.39
W 25    61.84
W 26    50.96
W 27    61.95
W 28    61.76
W 29    62.26
W 30    50.97
W 31    50.98
W 32    65.31
dtype: float64
```

In [66]:
```
week_no=fact_bookings_date["week no"].unique()
week_no
```

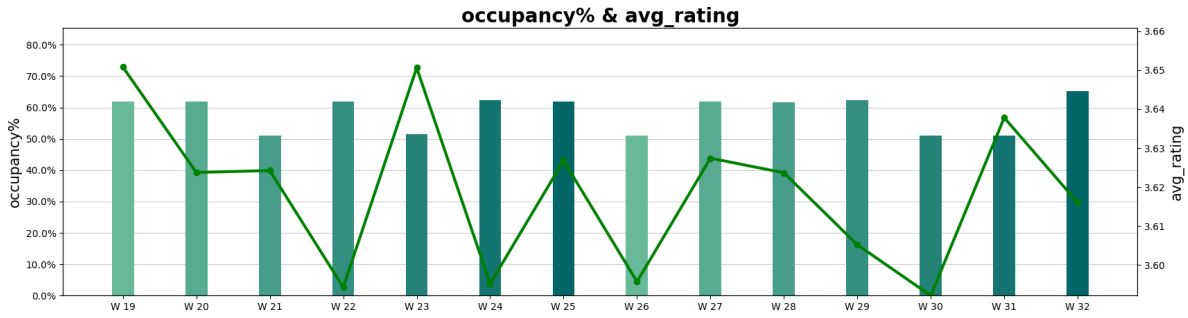
Out[66]:
```
array(['W 19', 'W 20', 'W 21', 'W 22', 'W 23', 'W 24', 'W 25', 'W 26',
       'W 27', 'W 28', 'W 29', 'W 30', 'W 31', 'W 32'], dtype=object)
```

In [67]:
```
from matplotlib.ticker import NullFormatter
```

In [68]:
```
def formatter1(x, pos):
    return str(x)+ "%"
```

In [69]:
```
style.use("default")
fig,ax1=plt.subplots(1,1,figsize=(20,5))
ax1.bar(week_no,occupancy2,width=0.3,color=c)
c=["#F20089","#E500A4","#DB00B6","#d100d1","#bc00dd","#b100e8","#a100f2","#890
ax1.set_ylim(0,occupancy2.max()+20)
ax1.yaxis.set_major_formatter(formatter1)
ax1.yaxis.set_minor_formatter(NullFormatter())
ax1.yaxis.grid(linewidth=0.5)
ax1.set_axisbelow(True)
ax1.set_ylabel("occupancy%",size=15)
ax2=ax1.twinx()
ax2.plot(week_no,avg_rating,"o-",linewidth=3,color="green")
ax2.set_ylim(avg_rating.min(),avg_rating.max()+0.01)
ax2.set_ylabel("avg_rating",size=15)
plt.title("occupancy% & avg_rating",fontsize=20,weight="bold")
```

Out[69]:
```
Text(0.5, 1.0, 'occupancy% & avg_rating')
```

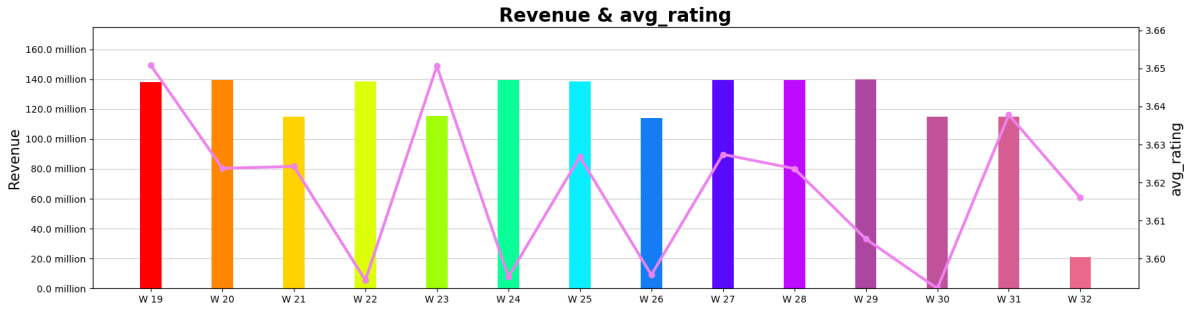


4.1 Revenue and Average Rating

In [70]:
```
def formatter(x, pos):
    return str(round(x / 1e6, 1))+ " million"
```

In [71]:
```
style.use("default")
fig2,ax3=plt.subplots(1,1,figsize=(20,5))
ax3.bar(week_no,revenue_by_weekday,width=0.3,color=["#ff0000","#ff8700","#ffd3
"#0aff99","#0aefff","#147df5","#580aff","#be0aff","#ac46a1","#c05299","#d55d92
ax3.set_ylim(0,revenue_by_weekday.max()*1.25)
ax3.yaxis.set_major_formatter(formatter)
ax3.yaxis.set_minor_formatter(NullFormatter())
ax3.yaxis.grid(linewidth=0.5)
ax3.set_axisbelow(True)
ax3.set_ylabel("Revenue",size=15)
ax4=ax3.twinx()
ax4.plot(week_no,avg_rating,"o-",linewidth=3,color="violet")
ax4.set_ylim(avg_rating.min(),avg_rating.max()+0.01)
ax4.set_ylabel("avg_rating",size=15)
plt.title("Revenue & avg_rating",fontsize=20,weight="bold")
```

Out[71]: Text(0.5, 1.0, 'Revenue & avg_rating')



In [ ]:

In [ ]: