atliq-analysis-final-report

March 4, 2024

1 Importing Libraries

[67]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import datetime as dt
from matplotlib import style
```

2 Importing Datas

Using the pd.read_csv() function to read in a CSV files

[2]:
```
df_date=pd.read_csv(r"dim_date.csv")
df_date
```

[2]:

	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 x 4 columns]

[3]:
```
# Show basic information of dataset like null value count of each column and␣
↪their data type
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
```

[4]:
```
df_hotels=pd.read_csv(r"dim_hotels.csv")
df_hotels
```

[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
4	16562	Atliq Bay	Luxury	Delhi
5	16563	Atliq Palace	Business	Delhi
6	17558	Atliq Grands	Luxury	Mumbai
7	17559	Atliq Exotica	Luxury	Mumbai
8	17560	Atliq City	Business	Mumbai
9	17561	Atliq Blu	Luxury	Mumbai
10	17562	Atliq Bay	Luxury	Mumbai
11	17563	Atliq Palace	Business	Mumbai
12	18558	Atliq Grands	Luxury	Hyderabad
13	18559	Atliq Exotica	Luxury	Hyderabad
14	18560	Atliq City	Business	Hyderabad
15	18561	Atliq Blu	Luxury	Hyderabad
16	18562	Atliq Bay	Luxury	Hyderabad
17	18563	Atliq Palace	Business	Hyderabad
18	19558	Atliq Grands	Luxury	Bangalore
19	19559	Atliq Exotica	Luxury	Bangalore
20	19560	Atliq City	Business	Bangalore
21	19561	Atliq Blu	Luxury	Bangalore
22	19562	Atliq Bay	Luxury	Bangalore
23	19563	Atliq Palace	Business	Bangalore
24	17564	Atliq Seasons	Business	Mumbai

[5]:
```
# Show basic information of dataset like null value count of each column and␣
↪their data type
df_hotels.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype
```

```
---  ------         --------------  -----
 0   property_id    25 non-null     int64
 1   property_name  25 non-null     object
 2   category       25 non-null     object
 3   city           25 non-null     object
dtypes: int64(1), object(3)
memory usage: 928.0+ bytes
```

[6]:
```
df_rooms=pd.read_csv(r"dim_rooms.csv")
df_rooms
```

[6]:

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

[7]:
```
# Show basic information of dataset like null value count of each column and␣
  ↪their data type
df_rooms.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   room_id     4 non-null      object
 1   room_class  4 non-null      object
dtypes: object(2)
memory usage: 192.0+ bytes
```

[8]:
```
df_aggre_bookings=pd.read_csv(r"fact_aggregated_bookings.csv")
df_aggre_bookings
```

[8]:

	property_id	check_in_date	room_category	successful_bookings	capacity
0	16559	01-May-22	RT1	25	30
1	19562	01-May-22	RT1	28	30
2	19563	01-May-22	RT1	23	30
3	17558	01-May-22	RT1	13	19
4	16558	01-May-22	RT1	18	19
…	…	…	…	…	…
9195	16563	31-Jul-22	RT4	13	18
9196	16559	31-Jul-22	RT4	13	18
9197	17558	31-Jul-22	RT4	3	6
9198	19563	31-Jul-22	RT4	3	6
9199	17561	31-Jul-22	RT4	3	4

```
[9200 rows x 5 columns]
```

[9]:
```
# Show basic information of dataset like null value count of each column and their data type
df_aggre_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9200 entries, 0 to 9199
Data columns (total 5 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   property_id          9200 non-null   int64 
 1   check_in_date        9200 non-null   object
 2   room_category        9200 non-null   object
 3   successful_bookings  9200 non-null   int64 
 4   capacity             9200 non-null   int64 
dtypes: int64(3), object(2)
memory usage: 359.5+ KB
```

[10]:
```
df_bookings=pd.read_csv(r"fact_bookings.csv")
df_bookings
```

[10]:

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given
0	May012216558RT11	16558	2022-04-27	2022-05-01	2022-05-02	3	RT1	direct online	1.0
1	May012216558RT12	16558	2022-04-30	2022-05-01	2022-05-02	2	RT1	others	NaN
2	May012216558RT13	16558	2022-04-28	2022-05-01	2022-05-04	2	RT1	logtrip	5.0
3	May012216558RT14	16558	2022-04-28	2022-05-01	2022-05-02	2	RT1	others	NaN
4	May012216558RT15	16558	2022-04-27	2022-05-01	2022-05-02	4	RT1	direct online	5.0
…	…	…	…	…	…	…	…	…	…
134585	Jul312217564RT46	17564	2022-07-29	2022-07-31	2022-08-03	1	RT4	makeyourtrip	2.0
134586	Jul312217564RT47	17564	2022-07-30	2022-07-31	2022-08-01	4	RT4	logtrip	2.0
134587	Jul312217564RT48	17564	2022-07-30	2022-07-31	2022-08-02	1	RT4	tripster	NaN
134588	Jul312217564RT49	17564	2022-07-29	2022-07-31	2022-08-01	2	RT4	logtrip	2.0
134589	Jul312217564RT410	17564	2022-07-31	2022-07-31	2022-08-01	2	RT4	makeyourtrip	NaN

```
       booking_status  revenue_generated  revenue_realized
0         Checked Out              10010             10010
1           Cancelled               9100              3640
2         Checked Out               9100              9100
3           Cancelled               9100              3640
4         Checked Out              10920             10920
…               …                  …                 …
134585    Checked Out              32300             32300
134586    Checked Out              38760             38760
134587      Cancelled              32300             12920
134588    Checked Out              32300             32300
134589      Cancelled              32300             12920

[134590 rows x 12 columns]
```

[11]:
```
# Show basic information of dataset like null value count of each column and␣
 ↪their data type
df_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134590 entries, 0 to 134589
Data columns (total 12 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   booking_id         134590 non-null  object
 1   property_id        134590 non-null  int64
 2   booking_date       134590 non-null  object
 3   check_in_date      134590 non-null  object
 4   checkout_date      134590 non-null  object
 5   no_guests          134590 non-null  int64
 6   room_category      134590 non-null  object
 7   booking_platform   134590 non-null  object
 8   ratings_given      56683 non-null   float64
 9   booking_status     134590 non-null  object
 10  revenue_generated  134590 non-null  int64
 11  revenue_realized   134590 non-null  int64
dtypes: float64(1), int64(4), object(7)
memory usage: 12.3+ MB
```

[12]:
```
df_bookings.describe()
```

[12]:
```
         property_id      no_guests  ratings_given  revenue_generated  \
count  134590.000000  134590.000000   56683.000000      134590.000000
mean    18061.113493       2.036808       3.619004       14916.013188
std      1093.055847       1.031766       1.235009        6452.868072
min     16558.000000       1.000000       1.000000        6500.000000
```

```
25%     17558.000000       1.000000       3.000000        9900.000000
50%     17564.000000       2.000000       4.000000       13500.000000
75%     18563.000000       2.000000       5.000000       18000.000000
max     19563.000000       6.000000       5.000000       45220.000000
```

```
       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
```

[13]:
```
# Checking out total null value in the booking table of dataset
df_bookings.isnull().sum()
```

[13]:
```
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
```

we can see in the column ratings_given there are 77907 null values

Changing the Data type of all dates columns in all tables

[14]:
```
df_date['date']=pd.to_datetime(df_date['date'])
df_aggre_bookings['check_in_date']=pd.
 ↪to_datetime(df_aggre_bookings['check_in_date'])
df_bookings['booking_date']=pd.to_datetime(df_bookings['booking_date'])
df_bookings['check_in_date']=pd.to_datetime(df_bookings['check_in_date'])
df_bookings['checkout_date']=pd.to_datetime(df_bookings['checkout_date'])
```

[15]:
```
df_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 134590 entries, 0 to 134589
Data columns (total 12 columns):
```

```
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   booking_id         134590 non-null  object
 1   property_id        134590 non-null  int64
 2   booking_date       134590 non-null  datetime64[ns]
 3   check_in_date      134590 non-null  datetime64[ns]
 4   checkout_date      134590 non-null  datetime64[ns]
 5   no_guests          134590 non-null  int64
 6   room_category      134590 non-null  object
 7   booking_platform   134590 non-null  object
 8   ratings_given      56683 non-null   float64
 9   booking_status     134590 non-null  object
 10  revenue_generated  134590 non-null  int64
 11  revenue_realized   134590 non-null  int64
dtypes: datetime64[ns](3), float64(1), int64(4), object(4)
memory usage: 12.3+ MB
```

[16]:
```
#filling the null value with 0
df_bookings['ratings_given']=df_bookings['ratings_given'].fillna(0)
```

[17]:
```
df_bookings.isnull().sum()
```

[17]:
```
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        0
booking_status       0
revenue_generated    0
revenue_realized     0
dtype: int64
```

Revenue Analysis

Merging 2 tables to do the analysis (Hotels & Bookings) to check the Revenue

[18]:
```
df_revenue=pd.merge(df_hotels,df_bookings,how='left',on='property_id')
df_revenue.head()
```

[18]:
```
   property_id property_name category    city        booking_id booking_date  \
0        16558  Atliq Grands   Luxury  Delhi  May012216558RT11   2022-04-27
1        16558  Atliq Grands   Luxury  Delhi  May012216558RT12   2022-04-30
2        16558  Atliq Grands   Luxury  Delhi  May012216558RT13   2022-04-28
3        16558  Atliq Grands   Luxury  Delhi  May012216558RT14   2022-04-28
```

```
4        16558  Atliq Grands   Luxury  Delhi  May012216558RT15   2022-04-27

  check_in_date checkout_date  no_guests room_category booking_platform  \
0    2022-05-01    2022-05-02          3           RT1    direct online
1    2022-05-01    2022-05-02          2           RT1           others
2    2022-05-01    2022-05-04          2           RT1          logtrip
3    2022-05-01    2022-05-02          2           RT1           others
4    2022-05-01    2022-05-02          4           RT1    direct online

   ratings_given booking_status  revenue_generated  revenue_realized
0            1.0    Checked Out              10010             10010
1            0.0      Cancelled               9100              3640
2            5.0    Checked Out               9100              9100
3            0.0      Cancelled               9100              3640
4            5.0    Checked Out              10920             10920
```

[19]:
```
df_revenue.shape
```

[19]:
```
(134590, 15)
```

[20]:
```
#Hotelwise Revenue
hotel_revenue=df_revenue.groupby(['property_name','city']).
 ↪agg({'revenue_realized':'sum'}).rename(columns={'revenue_realized':'Revenue␣
 ↪in Millions'})
hotel_revenue.reset_index(inplace=True)
hotel_revenue['Revenue in Millions']=hotel_revenue['Revenue in Millions']/
 ↪1000000
hotel_revenue['Revenue in Millions']=hotel_revenue['Revenue in Millions'].
 ↪round(2)
hotel_revenue
```

[20]:

	property_name	city	Revenue in Millions
0	Atliq Bay	Bangalore	82.44
1	Atliq Bay	Delhi	56.44
2	Atliq Bay	Hyderabad	69.26
3	Atliq Bay	Mumbai	51.91
4	Atliq Blu	Bangalore	72.96
5	Atliq Blu	Delhi	57.93
6	Atliq Blu	Hyderabad	56.04
7	Atliq Blu	Mumbai	73.92
8	Atliq City	Bangalore	81.88
9	Atliq City	Delhi	54.93
10	Atliq City	Hyderabad	61.01
11	Atliq City	Mumbai	88.00
12	Atliq Exotica	Bangalore	60.02
13	Atliq Exotica	Hyderabad	47.84
14	Atliq Exotica	Mumbai	212.44

```
15   Atliq Grands  Bangalore               54.49
16   Atliq Grands      Delhi               36.06
17   Atliq Grands  Hyderabad               46.25
18   Atliq Grands     Mumbai               74.73
19   Atliq Palace  Bangalore               68.60
20   Atliq Palace      Delhi               89.14
21   Atliq Palace  Hyderabad               44.84
22   Atliq Palace     Mumbai              101.51
23  Atliq Seasons     Mumbai               66.13
```

[21]:
```
revenue_pivot=hotel_revenue.
  ↪pivot(index='city',columns='property_name',values='Revenue in Millions')
revenue_pivot
```

[21]:
```
property_name  Atliq Bay  Atliq Blu  Atliq City  Atliq Exotica  Atliq Grands  \
city
Bangalore          82.44      72.96       81.88          60.02         54.49
Delhi              56.44      57.93       54.93            NaN         36.06
Hyderabad          69.26      56.04       61.01          47.84         46.25
Mumbai             51.91      73.92       88.00         212.44         74.73

property_name  Atliq Palace  Atliq Seasons
city
Bangalore             68.60            NaN
Delhi                 89.14            NaN
Hyderabad             44.84            NaN
Mumbai               101.51          66.13
```

[22]:
```
revenue_pivot.plot(kind='bar')
plt.ylabel('Revenue In Millions')
plt.title('City & Hotel Wise Revenue')
```

[22]:
```
Text(0.5, 1.0, 'City & Hotel Wise Revenue')
```

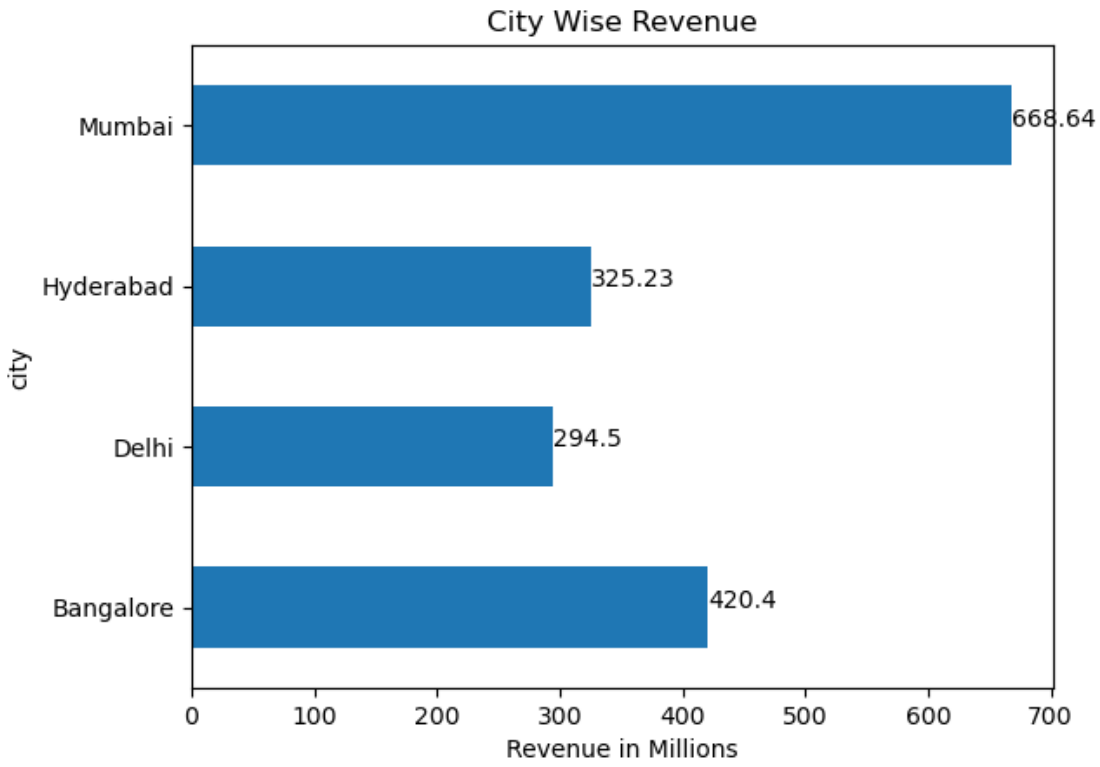
[23]:
```
# Citywise Revenue In Millions
city_revenue=df_revenue.groupby(['city']).agg({'revenue_realized':'sum'}).
  ↪rename(columns={'revenue_realized':'Revenue in Millions'})
city_revenue['Revenue in Millions']=city_revenue['Revenue in Millions']/1000000
city_revenue['Revenue in Millions']=city_revenue['Revenue in Millions'].round(2)
city_revenue.sort_values(by='Revenue in Millions',ascending=False)
```

[23]:

city	Revenue in Millions
Mumbai	668.64
Bangalore	420.40
Hyderabad	325.23
Delhi	294.50

[24]:
```
ax=city_revenue.plot(kind='barh',legend=False)
plt.xlabel('Revenue in Millions')
```

```
plt.title('City Wise Revenue')
for index, value in enumerate(city_revenue['Revenue in Millions']):
    ax.text(value, index, str(value))
plt.show()
```



Again merging 2 tables to do Revenue Analysis(df_revenue & df_date)

[25]:
```
revenue_tr=pd.
  ↪merge(df_date,df_revenue,how='left',left_on='date',right_on='check_in_date')
revenue_tr
```

[25]:

	date	mmm yy	week no	day_type	property_id	property_name	\
0	2022-05-01	May 22	W 19	weekend	16558	Atliq Grands	
1	2022-05-01	May 22	W 19	weekend	16558	Atliq Grands	
2	2022-05-01	May 22	W 19	weekend	16558	Atliq Grands	
3	2022-05-01	May 22	W 19	weekend	16558	Atliq Grands	
4	2022-05-01	May 22	W 19	weekend	16558	Atliq Grands	
…	…	…	…	…	…	…	
134585	2022-07-31	Jul 22	W 32	weekend	17564	Atliq Seasons	
134586	2022-07-31	Jul 22	W 32	weekend	17564	Atliq Seasons	
134587	2022-07-31	Jul 22	W 32	weekend	17564	Atliq Seasons	
134588	2022-07-31	Jul 22	W 32	weekend	17564	Atliq Seasons	

```
134589 2022-07-31  Jul 22    W 32  weekend        17564  Atliq Seasons

        category    city         booking_id booking_date check_in_date  \
0         Luxury   Delhi   May012216558RT11   2022-04-27    2022-05-01
1         Luxury   Delhi   May012216558RT12   2022-04-30    2022-05-01
2         Luxury   Delhi   May012216558RT13   2022-04-28    2022-05-01
3         Luxury   Delhi   May012216558RT14   2022-04-28    2022-05-01
4         Luxury   Delhi   May012216558RT15   2022-04-27    2022-05-01
…           …     …               …           …            …
134585  Business  Mumbai   Jul312217564RT46   2022-07-29    2022-07-31
134586  Business  Mumbai   Jul312217564RT47   2022-07-30    2022-07-31
134587  Business  Mumbai   Jul312217564RT48   2022-07-30    2022-07-31
134588  Business  Mumbai   Jul312217564RT49   2022-07-29    2022-07-31
134589  Business  Mumbai  Jul312217564RT410   2022-07-31    2022-07-31

       checkout_date  no_guests room_category booking_platform  ratings_given  \
0         2022-05-02          3           RT1    direct online            1.0
1         2022-05-02          2           RT1           others            0.0
2         2022-05-04          2           RT1          logtrip            5.0
3         2022-05-02          2           RT1           others            0.0
4         2022-05-02          4           RT1    direct online            5.0
…              …        …           …              …            …
134585    2022-08-03          1           RT4     makeyourtrip            2.0
134586    2022-08-01          4           RT4          logtrip            2.0
134587    2022-08-02          1           RT4         tripster            0.0
134588    2022-08-01          2           RT4          logtrip            2.0
134589    2022-08-01          2           RT4     makeyourtrip            0.0

       booking_status  revenue_generated  revenue_realized
0         Checked Out              10010             10010
1           Cancelled               9100              3640
2         Checked Out               9100              9100
3           Cancelled               9100              3640
4         Checked Out              10920             10920
…              …                 …                 …
134585    Checked Out              32300             32300
134586    Checked Out              38760             38760
134587      Cancelled              32300             12920
134588    Checked Out              32300             32300
134589      Cancelled              32300             12920

[134590 rows x 19 columns]
```

[26]:
```
revenue_tr.shape
```

[26]:
```
(134590, 19)
```

[27]:
```
#Weekwise Revenue
revenue_trend=revenue_tr.groupby(['week no','property_name']).
 ↪agg({'revenue_realized':'sum'}).rename(columns={'revenue_realized':'Revenue␣
 ↪in Millions'})
revenue_trend['Revenue in Millions']=revenue_trend['Revenue in Millions']/
 ↪1000000
revenue_trend['Revenue in Millions']=revenue_trend['Revenue in Millions'].
 ↪round(2)
revenue_trend.reset_index(inplace=True)
revenue_trend
```

[27]:

	week no	property_name	Revenue in Millions
0	W 19	Atliq Bay	20.87
1	W 19	Atliq Blu	20.98
2	W 19	Atliq City	23.32
3	W 19	Atliq Exotica	25.74
4	W 19	Atliq Grands	17.55
..	…	…	…
93	W 32	Atliq City	3.66
94	W 32	Atliq Exotica	3.85
95	W 32	Atliq Grands	2.63
96	W 32	Atliq Palace	3.79
97	W 32	Atliq Seasons	0.83

[98 rows x 3 columns]

[28]:
```
# Weekly Hotelwise Revenue Trend
pivot_data=revenue_trend.pivot(index='week␣
 ↪no',columns='property_name',values='Revenue in Millions')
pivot_data
```

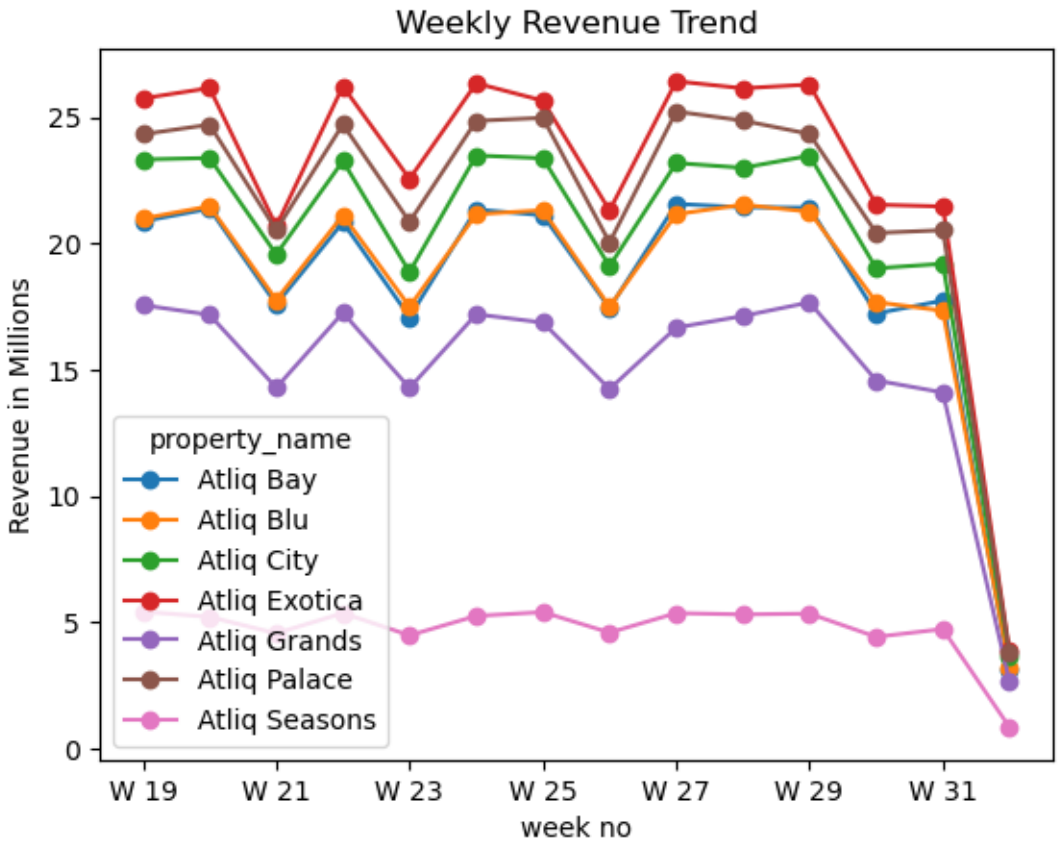
[28]:

property_name	Atliq Bay	Atliq Blu	Atliq City	Atliq Exotica	Atliq Grands	\
week no						
W 19	20.87	20.98	23.32	25.74	17.55	
W 20	21.37	21.47	23.38	26.15	17.17	
W 21	17.58	17.76	19.56	20.68	14.28	
W 22	20.83	21.11	23.26	26.22	17.25	
W 23	17.08	17.49	18.86	22.55	14.27	
W 24	21.34	21.15	23.48	26.34	17.19	
W 25	21.11	21.31	23.37	25.65	16.86	
W 26	17.43	17.49	19.09	21.30	14.23	
W 27	21.56	21.15	23.18	26.42	16.66	
W 28	21.43	21.53	22.99	26.14	17.12	
W 29	21.41	21.25	23.47	26.29	17.66	
W 30	17.22	17.67	19.01	21.53	14.57	
W 31	17.73	17.33	19.19	21.46	14.09	
W 32	3.10	3.16	3.66	3.85	2.63	

```
property_name  Atliq Palace  Atliq Seasons
week no
W 19                  24.32           5.41
W 20                  24.70           5.20
W 21                  20.52           4.55
W 22                  24.71           5.35
W 23                  20.85           4.47
W 24                  24.85           5.24
W 25                  24.97           5.39
W 26                  20.03           4.58
W 27                  25.22           5.35
W 28                  24.86           5.30
W 29                  24.33           5.33
W 30                  20.41           4.41
W 31                  20.52           4.72
W 32                   3.79           0.83
```

[29]:
```
pivot_data.plot(kind='line',marker='o')
plt.ylabel('Revenue in Millions')
plt.title('Weekly Revenue Trend')
```

[29]:
```
Text(0.5, 1.0, 'Weekly Revenue Trend')
```



[30]:
```
# Week over Week Revenue Trend
atliq_revenue_trend=revenue_tr.groupby(['week no']).agg({'revenue_realized':
 ↪'sum'}).rename(columns={'revenue_realized':'Revenue in Millions'})
atliq_revenue_trend['Revenue in Millions']=atliq_revenue_trend['Revenue in␣
 ↪Millions']/1000000
atliq_revenue_trend['Revenue in Millions']=atliq_revenue_trend['Revenue in␣
 ↪Millions'].round(2)
atliq_revenue_trend['Prev week Revenue']=atliq_revenue_trend['Revenue in␣
 ↪Millions'].shift(1)
atliq_revenue_trend['Change Percentage']=((atliq_revenue_trend['Revenue in␣
 ↪Millions']/atliq_revenue_trend['Prev week Revenue'])-1)*100
atliq_revenue_trend
```

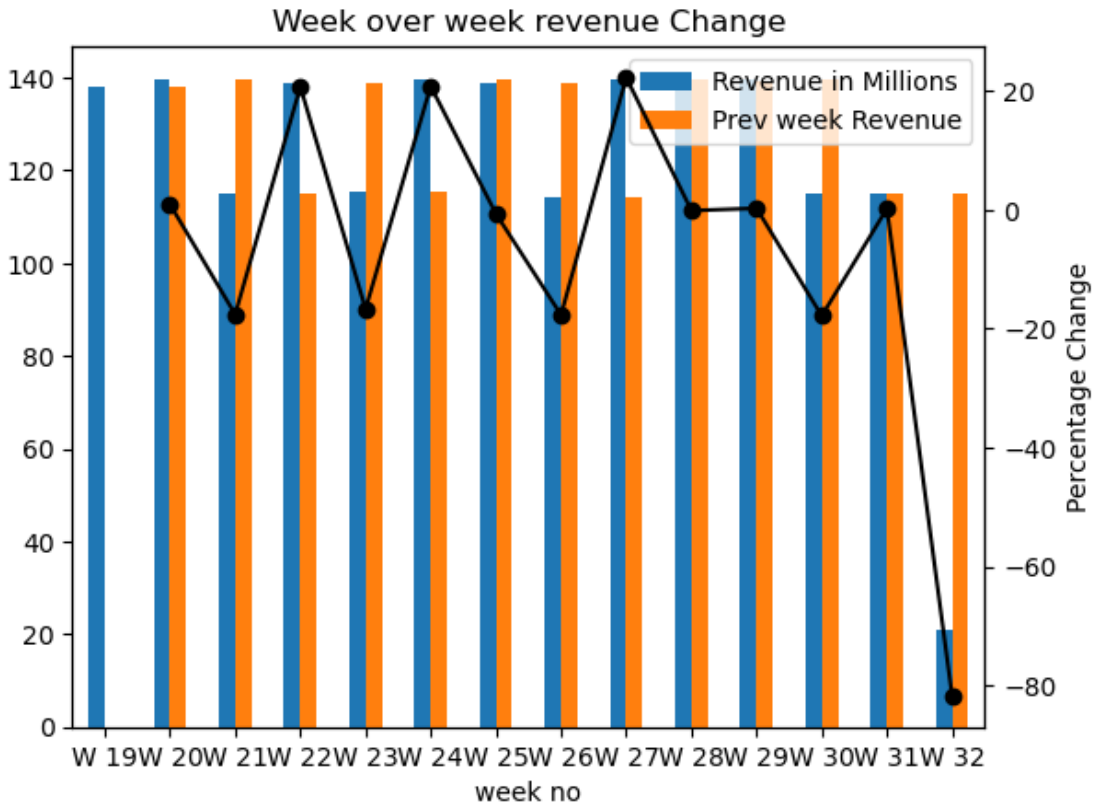
[30]:

week no	Revenue in Millions	Prev week Revenue	Change Percentage
W 19	138.18	NaN	NaN
W 20	139.44	138.18	0.911854
W 21	114.92	139.44	-17.584624
W 22	138.72	114.92	20.710059

W 23	115.57	138.72	-16.688293
W 24	139.58	115.57	20.775288
W 25	138.67	139.58	-0.651956
W 26	114.15	138.67	-17.682267
W 27	139.56	114.15	22.260184
W 28	139.38	139.56	-0.128977
W 29	139.73	139.38	0.251112
W 30	114.81	139.73	-17.834395
W 31	115.04	114.81	0.200331
W 32	21.01	115.04	-81.736787

[31]:
```
atliq_revenue_trend[['Revenue in Millions','Prev week Revenue']].
 ↪plot(kind='bar')
atliq_revenue_trend['Change Percentage'].
 ↪plot(secondary_y=True,color='black',marker='o')
plt.ylabel('Percentage Change')
plt.title('Week over week revenue Change')
```

[31]: Text(0.5, 1.0, 'Week over week revenue Change')



Booking Analysis

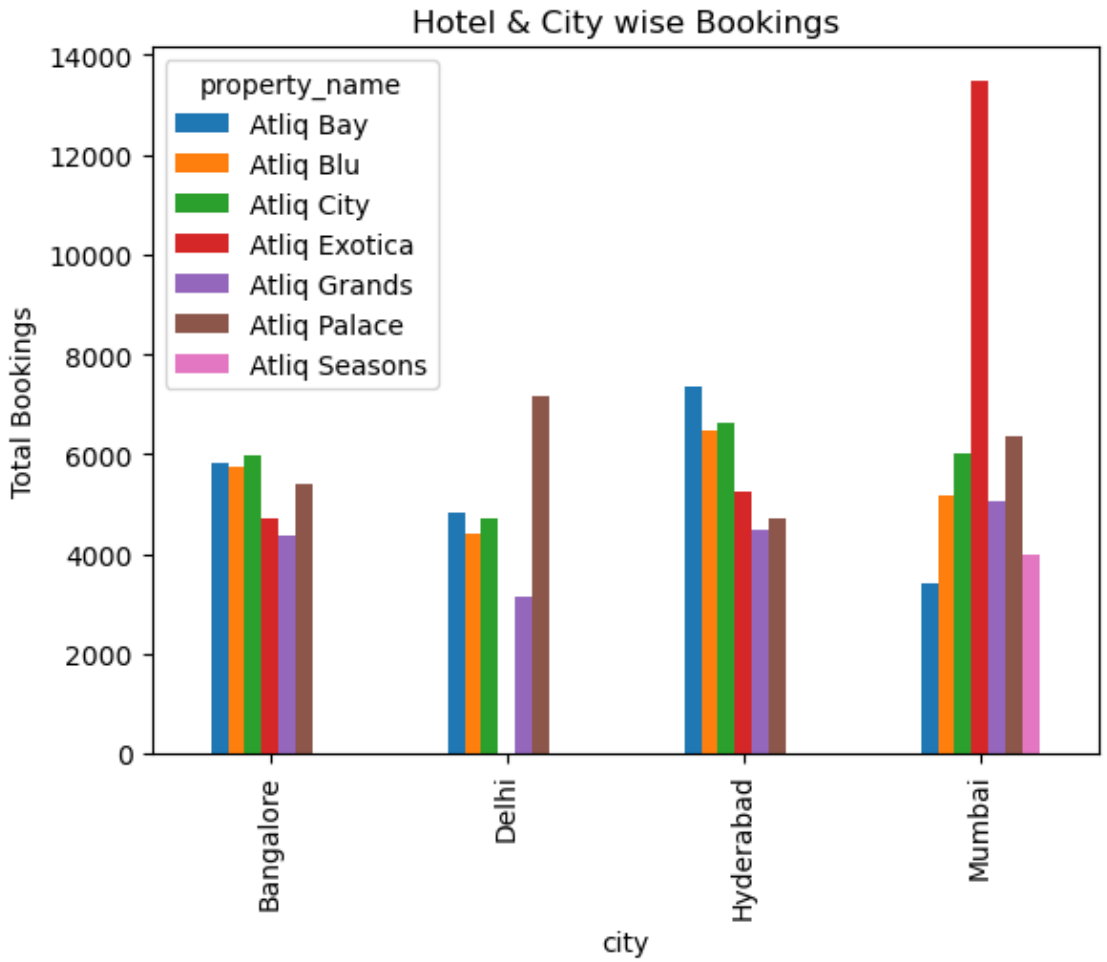
[32]:
```
# Hotelwise Bookings
hotel_bookings=df_revenue.groupby(['property_name','city']).agg({'booking_id':
 ↪'nunique'}).rename(columns={'booking_id':'Total bookings'})
hotel_bookings.reset_index(inplace=True)
hotel_bookings
```

[32]:

	property_name	city	Total bookings
0	Atliq Bay	Bangalore	5812
1	Atliq Bay	Delhi	4820
2	Atliq Bay	Hyderabad	7333
3	Atliq Bay	Mumbai	3424
4	Atliq Blu	Bangalore	5736
5	Atliq Blu	Delhi	4418
6	Atliq Blu	Hyderabad	6458
7	Atliq Blu	Mumbai	5183
8	Atliq City	Bangalore	5979
9	Atliq City	Delhi	4693
10	Atliq City	Hyderabad	6638
11	Atliq City	Mumbai	6013
12	Atliq Exotica	Bangalore	4705
13	Atliq Exotica	Hyderabad	5256
14	Atliq Exotica	Mumbai	13480
15	Atliq Grands	Bangalore	4371
16	Atliq Grands	Delhi	3153
17	Atliq Grands	Hyderabad	4475
18	Atliq Grands	Mumbai	5036
19	Atliq Palace	Bangalore	5413
20	Atliq Palace	Delhi	7147
21	Atliq Palace	Hyderabad	4728
22	Atliq Palace	Mumbai	6337
23	Atliq Seasons	Mumbai	3982

[33]:
```
booking_pivot=hotel_bookings.
 ↪pivot(index='city',columns='property_name',values='Total bookings')
booking_pivot.plot(kind='bar')
plt.ylabel('Total Bookings')
plt.title('Hotel & City wise Bookings')
```

[33]:
```
Text(0.5, 1.0, 'Hotel & City wise Bookings')
```



Occupancy % by City

[34]:
```
aggregated_booking_city=pd.merge(df_aggre_bookings,df_hotels,on="property_id")
aggregated_booking_city
```

[34]:

	property_id	check_in_date	room_category	successful_bookings	capacity	\
0	16559	2022-05-01	RT1	25	30	
1	16559	2022-05-01	RT2	35	41	
2	16559	2022-05-01	RT3	27	32	
3	16559	2022-05-01	RT4	17	18	
4	16559	2022-05-02	RT1	20	30	
…	…	…	…	…	…	
9195	18560	2022-07-30	RT4	9	15	
9196	18560	2022-07-31	RT1	22	30	
9197	18560	2022-07-31	RT2	34	40	
9198	18560	2022-07-31	RT3	17	24	
9199	18560	2022-07-31	RT4	12	15	

```
      property_name  category       city
0     Atliq Exotica    Luxury     Mumbai
1     Atliq Exotica    Luxury     Mumbai
2     Atliq Exotica    Luxury     Mumbai
3     Atliq Exotica    Luxury     Mumbai
4     Atliq Exotica    Luxury     Mumbai
…               …       …         …
9195     Atliq City  Business  Hyderabad
9196     Atliq City  Business  Hyderabad
9197     Atliq City  Business  Hyderabad
9198     Atliq City  Business  Hyderabad
9199     Atliq City  Business  Hyderabad

[9200 rows x 8 columns]
```

[35]:
```
successful_bookings=aggregated_booking_city.groupby("city").successful_bookings.
↪sum()
successful_bookings
```

[35]:
```
city
Bangalore    32016
Delhi        24231
Hyderabad    34888
Mumbai       43455
Name: successful_bookings, dtype: int64
```

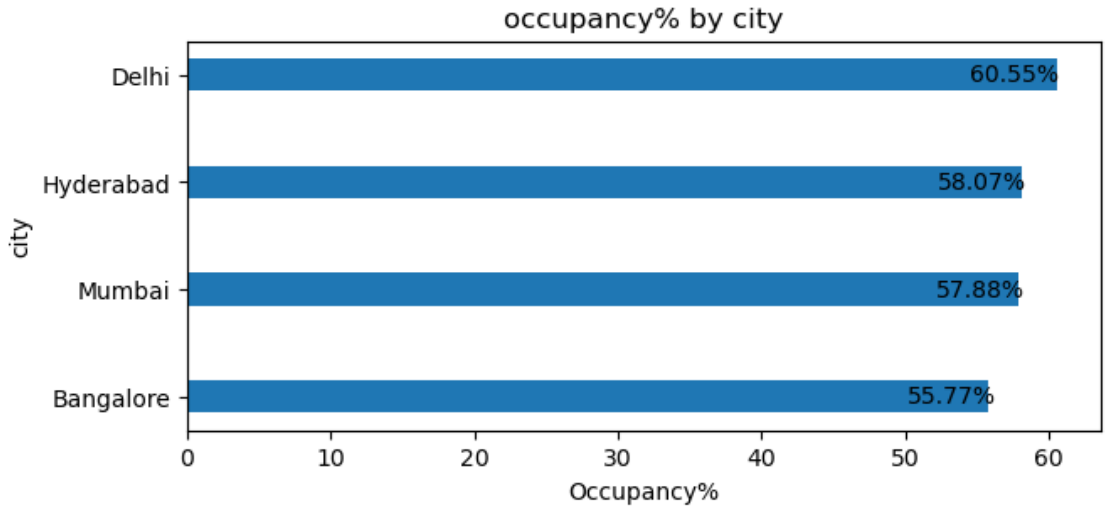
[36]:
```
capacity_of_bookings=aggregated_booking_city.groupby("city").capacity.sum()
capacity_of_bookings
```

[36]:
```
city
Bangalore    57408
Delhi        40020
Hyderabad    60076
Mumbai       75072
Name: capacity, dtype: int64
```

[37]:
```
occupancy=round((successful_bookings/capacity_of_bookings)*100,2).sort_values()
occupancy
```

[37]:
```
city
Bangalore    55.77
Mumbai       57.88
Hyderabad    58.07
Delhi        60.55
dtype: float64
```

[38]:
```
yplot=np.arange(4)
plt.figure(figsize=(7,3))
x=["Bangalore","Mumbai","Hyderabad","Delhi"]
plt.barh(yplot,occupancy,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")
```



Booking % by Platform

[39]:
```
df_bookings.head()
```

[39]:

	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
0	May012216558RT11	16558	2022-04-27	2022-05-01	2022-05-02	3	RT1	direct online	1.0	Checked Out		
1	May012216558RT12	16558	2022-04-30	2022-05-01	2022-05-02	2	RT1	others	0.0	Cancelled		
2	May012216558RT13	16558	2022-04-28	2022-05-01	2022-05-04	2	RT1	logtrip	5.0	Checked Out		
3	May012216558RT14	16558	2022-04-28	2022-05-01	2022-05-02	2	RT1	others	0.0	Cancelled		
4	May012216558RT15	16558	2022-04-27	2022-05-01	2022-05-02	4	RT1	direct online	5.0	Checked Out		

```
0              10010             10010
1               9100              3640
2               9100              9100
3               9100              3640
4              10920             10920
```

[40]:
```
total_booking=df_bookings["booking_id"].count()
total_booking
```

[40]:
```
134590
```

[41]:
```
total_booking_by_platform=df_bookings.groupby('booking_platform').booking_id.
 ↪count()
total_booking_by_platform
```

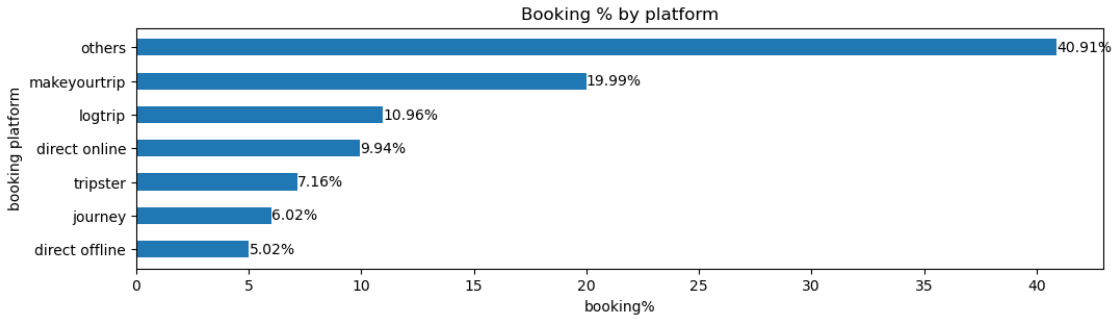
[41]:
```
booking_platform
direct offline     6755
direct online     13379
journey            8106
logtrip           14756
makeyourtrip      26898
others            55066
tripster           9630
Name: booking_id, dtype: int64
```

[42]:
```
Booking_by_platform=round((total_booking_by_platform/total_booking)*100,2)
Booking_by_platform=Booking_by_platform.sort_values()
Booking_by_platform
```

[42]:
```
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
```

[43]:
```
yplot=np.arange(7)
plt.figure(figsize=(12,3))
x=["direct offline","journey","tripster","direct␣
 ↪online","logtrip","makeyourtrip","others"]
plt.barh(yplot,Booking_by_platform,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")
```



[44]:
```
rooms_df=pd.
 ↪merge(df_rooms,df_bookings,how='left',left_on='room_id',right_on='room_category')
rooms_df.head()
```

[44]:

	room_id	room_class	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
0	RT1	Standard	May012216558RT11	16558	2022-04-27	2022-05-01	2022-05-02	3	RT1	direct online	1.0	Checked Out	10010	10010
1	RT1	Standard	May012216558RT12	16558	2022-04-30	2022-05-01	2022-05-02	2	RT1	others	0.0	Cancelled	9100	3640
2	RT1	Standard	May012216558RT13	16558	2022-04-28	2022-05-01	2022-05-04	2	RT1	logtrip	5.0	Checked Out	9100	9100
3	RT1	Standard	May012216558RT14	16558	2022-04-28	2022-05-01	2022-05-02	2	RT1	others	0.0	Cancelled	9100	3640
4	RT1	Standard	May012216558RT15	16558	2022-04-27	2022-05-01	2022-05-02	4	RT1	direct online	5.0	Checked Out	10920	10920

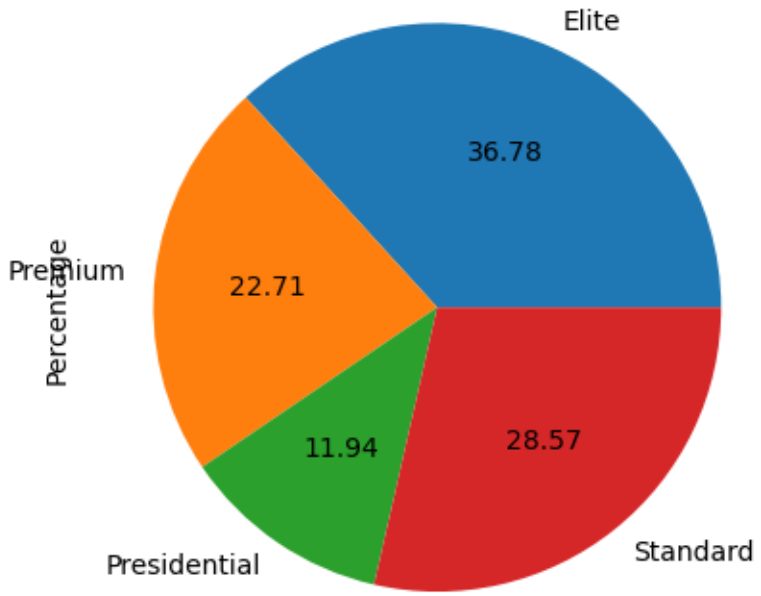
[45]:
```
#Room Categorywise Bookings
room_bookings=rooms_df.groupby(['room_class']).agg({'booking_id':'nunique'})
room_bookings['Percentage']=room_bookings['booking_id']*100/
 ↪room_bookings['booking_id'].sum()
room_bookings['Percentage']=room_bookings['Percentage'].round(2)
room_bookings
```

[45]:

room_class	booking_id	Percentage
Elite	49505	36.78
Premium	30566	22.71
Presidential	16073	11.94
Standard	38446	28.57

[47]:
```
#Room Caegory wise Booking Percentage
room_bookings['Percentage'].plot(kind='pie',subplots=True,autopct='%0.2f')
```

[47]:
```
array([<Axes: ylabel='Percentage'>], dtype=object)
```



[50]:
```
# Booking Trends weekly
weekly_bookings=revenue_tr.groupby(['week no','property_name']).
 ↪agg({'booking_id':'nunique'}).rename(columns={'booking_id':'Total Bookings'})
weekly_bookings.reset_index(inplace=True)
weekly_bookings_pivot=weekly_bookings.pivot(index='week␣
 ↪no',columns='property_name',values='Total Bookings')
weekly_bookings_pivot
```

[50]:

property_name	Atliq Bay	Atliq Blu	Atliq City	Atliq Exotica	Atliq Grands	\
week no						
W 19	1745	1771	1897	1907	1411	

```
W 20                1746       1789        1906          1906          1384
W 21                1441       1465        1586          1546          1134
W 22                1726       1754        1896          1913          1389
W 23                1421       1475        1559          1624          1149
W 24                1756       1784        1916          1937          1394
W 25                1734       1779        1897          1895          1381
W 26                1424       1459        1568          1574          1141
W 27                1758       1777        1883          1911          1379
W 28                1753       1781        1867          1902          1380
W 29                1750       1784        1922          1921          1408
W 30                1427       1457        1569          1569          1145
W 31                1451       1450        1566          1557          1130
W 32                 257        270         291           279           210
```

```
property_name  Atliq Palace  Atliq Seasons
week no
W 19                   1913            321
W 20                   1908            319
W 21                   1600            270
W 22                   1933            323
W 23                   1591            270
W 24                   1929            325
W 25                   1930            327
W 26                   1581            270
W 27                   1933            321
W 28                   1924            322
W 29                   1909            324
W 30                   1585            268
W 31                   1593            274
W 32                    296             48
```

[51]:
```
weekly_bookings_pivot.plot(kind='line',marker='o')
plt.ylabel('Total Bookings')
plt.title('Weekly Hotel Bookings')
```

[51]:
```
Text(0.5, 1.0, 'Weekly Hotel Bookings')
```

[52]:
```
x=revenue_tr.groupby(['week no','booking_status']).agg({'booking_id':'nunique'})
x.reset_index(inplace=True)
x
```

[52]:

	week no	booking_status	booking_id
0	W 19	Cancelled	2779
1	W 19	Checked Out	7628
2	W 19	No Show	558
3	W 20	Cancelled	2711
4	W 20	Checked Out	7699
5	W 20	No Show	548
6	W 21	Cancelled	2226
7	W 21	Checked Out	6331
8	W 21	No Show	485
9	W 22	Cancelled	2772
10	W 22	Checked Out	7585
11	W 22	No Show	577
12	W 23	Cancelled	2222
13	W 23	Checked Out	6399
14	W 23	No Show	468

15	W 24	Cancelled	2797
16	W 24	Checked Out	7688
17	W 24	No Show	556
18	W 25	Cancelled	2748
19	W 25	Checked Out	7657
20	W 25	No Show	538
21	W 26	Cancelled	2299
22	W 26	Checked Out	6292
23	W 26	No Show	426
24	W 27	Cancelled	2691
25	W 27	Checked Out	7734
26	W 27	No Show	537
27	W 28	Cancelled	2643
28	W 28	Checked Out	7757
29	W 28	No Show	529
30	W 29	Cancelled	2733
31	W 29	Checked Out	7777
32	W 29	No Show	508
33	W 30	Cancelled	2210
34	W 30	Checked Out	6348
35	W 30	No Show	462
36	W 31	Cancelled	2193
37	W 31	Checked Out	6347
38	W 31	No Show	481
39	W 32	Cancelled	396
40	W 32	Checked Out	1169
41	W 32	No Show	86

[53]:
```
x=revenue_tr.groupby(['week no','booking_status']).agg({'booking_id':'nunique'})
x.reset_index(inplace=True)
x
```

[53]:

	week no	booking_status	booking_id
0	W 19	Cancelled	2779
1	W 19	Checked Out	7628
2	W 19	No Show	558
3	W 20	Cancelled	2711
4	W 20	Checked Out	7699
5	W 20	No Show	548
6	W 21	Cancelled	2226
7	W 21	Checked Out	6331
8	W 21	No Show	485
9	W 22	Cancelled	2772
10	W 22	Checked Out	7585
11	W 22	No Show	577
12	W 23	Cancelled	2222
13	W 23	Checked Out	6399

14	W 23	No Show	468
15	W 24	Cancelled	2797
16	W 24	Checked Out	7688
17	W 24	No Show	556
18	W 25	Cancelled	2748
19	W 25	Checked Out	7657
20	W 25	No Show	538
21	W 26	Cancelled	2299
22	W 26	Checked Out	6292
23	W 26	No Show	426
24	W 27	Cancelled	2691
25	W 27	Checked Out	7734
26	W 27	No Show	537
27	W 28	Cancelled	2643
28	W 28	Checked Out	7757
29	W 28	No Show	529
30	W 29	Cancelled	2733
31	W 29	Checked Out	7777
32	W 29	No Show	508
33	W 30	Cancelled	2210
34	W 30	Checked Out	6348
35	W 30	No Show	462
36	W 31	Cancelled	2193
37	W 31	Checked Out	6347
38	W 31	No Show	481
39	W 32	Cancelled	396
40	W 32	Checked Out	1169
41	W 32	No Show	86

[54]:
```
y=revenue_tr.groupby(['week no']).agg({'booking_id':'nunique'}).
  ↪rename(columns={'booking_id':'Total Bookings'})
y.reset_index(inplace=True)
y
```

[54]:

	week no	Total Bookings
0	W 19	10965
1	W 20	10958
2	W 21	9042
3	W 22	10934
4	W 23	9089
5	W 24	11041
6	W 25	10943
7	W 26	9017
8	W 27	10962
9	W 28	10929
10	W 29	11018
11	W 30	9020

```
12    W 31            9021
13    W 32            1651
```

Occupancy % and Average Rating

[55]:
```
df_booking_date=pd.
 ↪merge(df_bookings,df_date,left_on="check_in_date",right_on="date")
df_booking_date.head()
```

[55]:

	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	date	mmm yy	week no	day_type
0	May012216558RT11	16558	2022-04-27	2022-05-01	2022-05-02	3	RT1	direct online	1.0	Checked Out	10010	10010	2022-05-01	May 22	W 19	weekend
1	May012216558RT12	16558	2022-04-30	2022-05-01	2022-05-02	2	RT1	others	0.0	Cancelled	9100	3640	2022-05-01	May 22	W 19	weekend
2	May012216558RT13	16558	2022-04-28	2022-05-01	2022-05-04	2	RT1	logtrip	5.0	Checked Out	9100	9100	2022-05-01	May 22	W 19	weekend
3	May012216558RT14	16558	2022-04-28	2022-05-01	2022-05-02	2	RT1	others	0.0	Cancelled	9100	3640	2022-05-01	May 22	W 19	weekend
4	May012216558RT15	16558	2022-04-27	2022-05-01	2022-05-02	4	RT1	direct online	5.0	Checked Out	10920	10920	2022-05-01	May 22	W 19	weekend

[56]:
```
avg_rating=df_booking_date.groupby("week no").ratings_given.mean()
```

[57]:
```
revenue_by_weekday=df_booking_date.groupby("week no").revenue_realized.sum()
revenue_by_weekday
```

[57]:
```
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
```

[59]:
```
aggregated_booking_and_date=pd.
 ↪merge(df_aggre_bookings,df_date,left_on="check_in_date",right_on="date")

aggregated_booking_and_date.head()
```

[59]:
```
   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
3        17558    2022-05-01           RT1                   13        19
4        16558    2022-05-01           RT1                   18        19

        date  mmm yy week no day_type
0 2022-05-01  May 22    W 19  weekend
1 2022-05-01  May 22    W 19  weekend
2 2022-05-01  May 22    W 19  weekend
3 2022-05-01  May 22    W 19  weekend
4 2022-05-01  May 22    W 19  weekend
```

[60]:
```
successful_bookings2=aggregated_booking_and_date.groupby("week no").
 ↪successful_bookings.sum()
successful_bookings2
capacity_of_bookings2=aggregated_booking_and_date.groupby("week no").capacity.
 ↪sum()
capacity_of_bookings
occupancy2=round((successful_bookings2/capacity_of_bookings2)*100,2)
occupancy2
```

[60]:
```
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
```

[62]:
```
week_no=df_booking_date["week no"].unique()
week_no
```

[62]:
```
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)
```

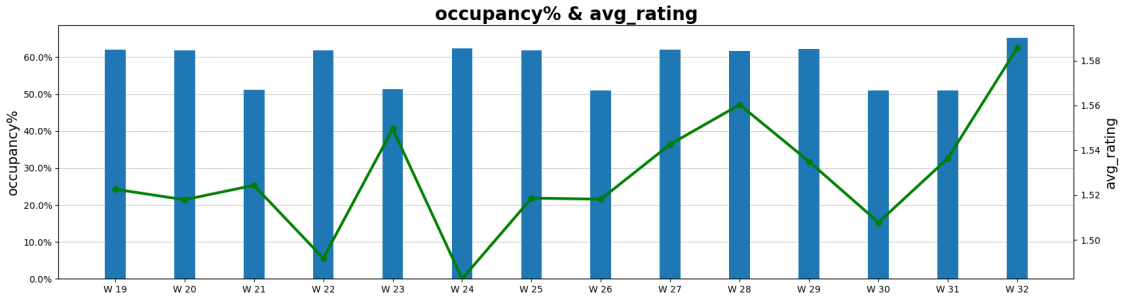
[63]:
```
from matplotlib.ticker import NullFormatter
```

[64]:
```
def formatter1(x, pos):
 return str(x)+ "%"
```

[70]:
```
style.use("default")
fig,ax1=plt.subplots(1,1,figsize=(20,5))
ax1.bar(week_no,occupancy2,width=0.3)

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")
```

[70]:
```
Text(0.5, 1.0, 'occupancy% & avg_rating')
```

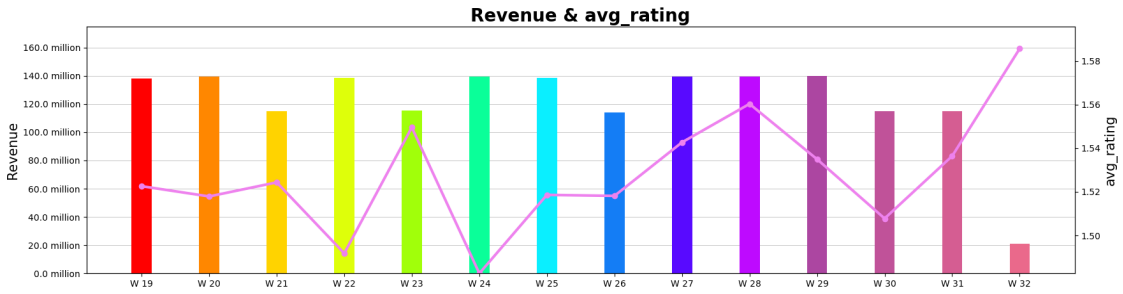


Revenue and Average Rating

[71]:
```
def formatter(x, pos):
 return str(round(x / 1e6, 1))+ " million"
```

[72]:
```
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","#ffd300","#deff0a","#a1ff0a",
"#0aff99","#0aefff","#147df5","#580aff","#be0aff","#ac46a1","#c05299","#d55d92","#ea698b"])
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")
```

[72]:
```
Text(0.5, 1.0, 'Revenue & avg_rating')
```



[ ]: