

AtliQ Grands



Exploratory Data Analysis in Hospitality Domain using Python

created by
Sowmya Satyamurthi



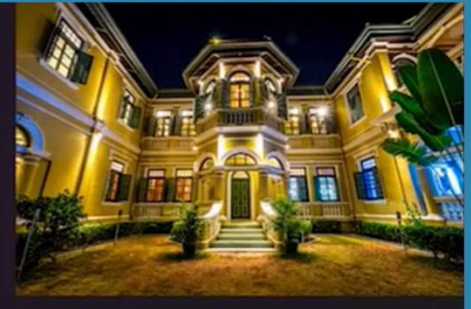
AtliQ Seasons



AtliQ Exotica



AtliQ Bay



AtliQ Palace

Introduction

AtliQ Grands is a leading hotel chain with a presence across multiple cities in India.

Various room categories



Standard



Elite

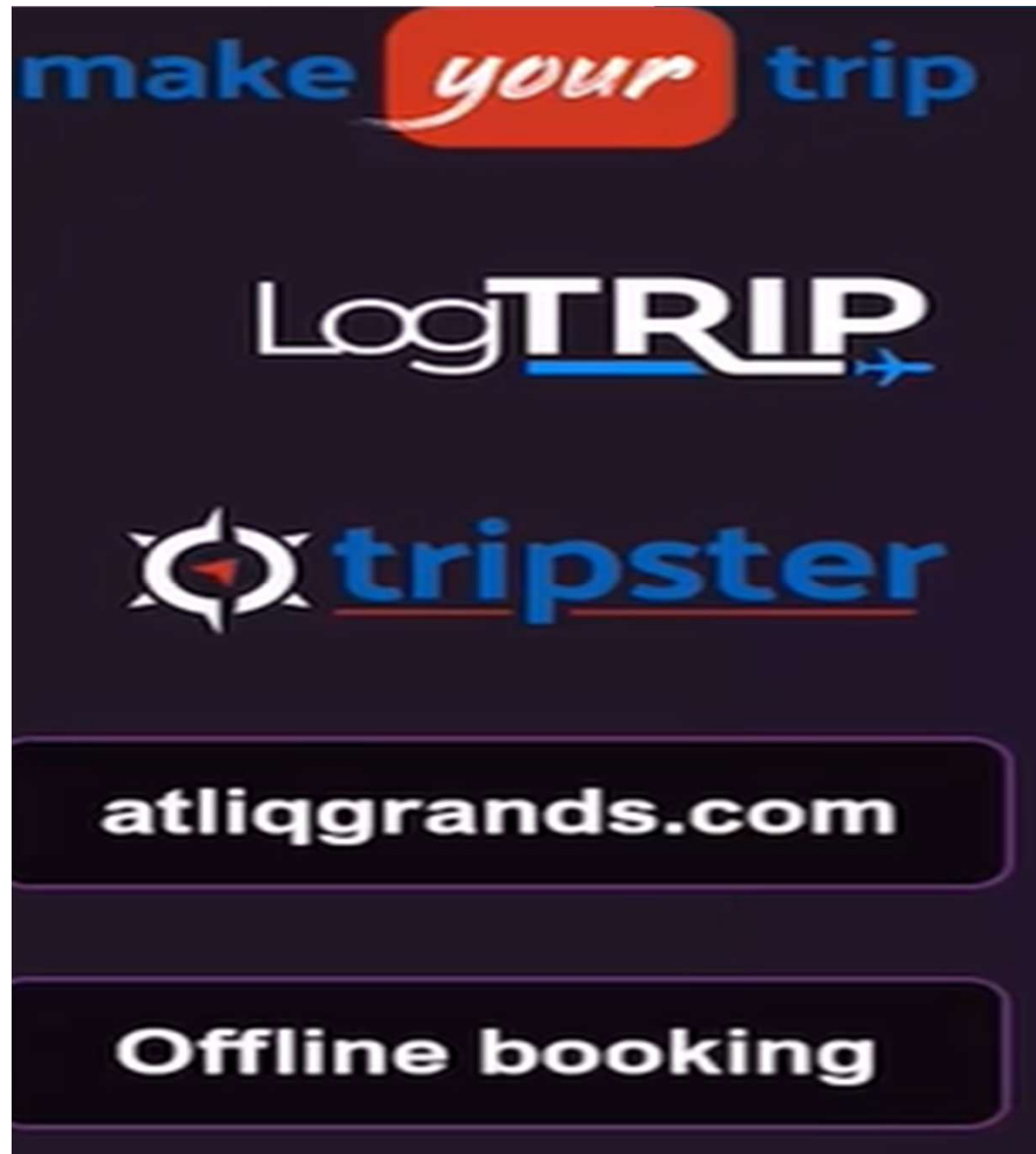


Premium



Presidential

Various
mediums of
hotel booking

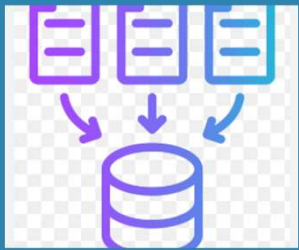




Problem Statement

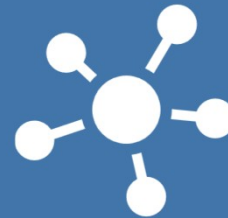
- Facing strong competition
- Losing revenue and market share
- Hired a data analyst
- Shift towards data-driven decision-making
- Improve decision quality
- Regain revenue growth

Extraction Transformation Loading(ETL)



Data collection and
Understanding

Data cleaning and
Exploration

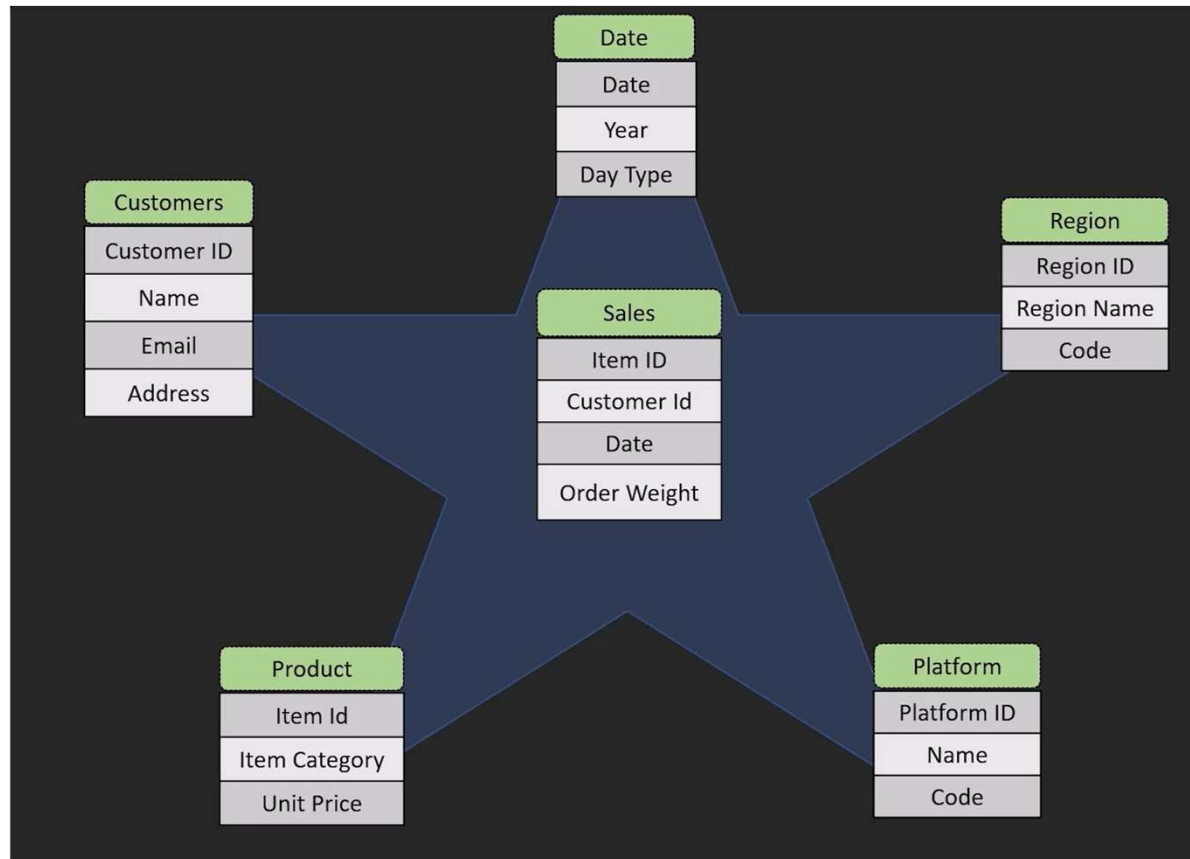


Data Transformation

Collect Insights



Schema used



How?

PYTHON

JUPYTER

NOTEBOOK



Data Exploration

ter Exploratory Data Analysis_Exercise Last Checkpoint: 1 hour ago

View Run Kernel Settings Help

Markdown

JupyterLab Python

AtliQ Hotels Data Analysis Project

```
import pandas as pd
```

=> 1. Data Import and Data Exploration

Datasets

We have 5 csv file

- dim_date.csv
- dim_hotels.csv
- dim_rooms.csv
- fact_aggregated_bookings
- fact_bookings.csv

Read bookings data in a datagrame

```
df_bookings = pd.read_csv('datasets/fact_bookings.csv')
```

Data from CSV files is read into data frames using Pandas

The functions `shape`, `unique()`, `value_counts()` help in exploring the dataset by showing its size, identifying distinct values, and summarizing their frequency.

jupyter Exploratory Data Analysis_Exercise Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help

Explore bookings data

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

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings
0	May012216558RT11	16558	27-04-22	1/5/2022	2/5/2022	-3.0	RT1	direct online	
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	logtrip	
3	May012216558RT14	16558	28-04-22	1/5/2022	2/5/2022	-2.0	RT1	others	
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	

```
[248]: df_bookings.shape
```

```
[248]: (134590, 12)
```

```
[249]: df_bookings.room_category.unique()
```

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

```
[250]: df_bookings.booking_platform.unique()
```

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

```
[5]: df_bookings.booking_platform.value_counts()
```

```
[5]: booking_platform
```

Data Cleaning

Identifying and rectifying erroneous values in the dataset

Jupyter Exploratory Data Analysis Exercise Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Not Trusted

JupyterLab Python 3 (ipykernel)

(1) Clean invalid guests

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

```
[266]:
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	re
0	May012216558RT11	16558	27-04-22	1/5/2022	2/5/2022	-3.0	RT1	direct online	1.0	Checked Out	
3	May012216558RT14	16558	28-04-22	1/5/2022	2/5/2022	-2.0	RT1	others	NaN	Cancelled	
17924	May122218559RT44	18559	12/5/2022	12/5/2022	14-05-22	-10.0	RT4	direct online	NaN	No Show	
18020	May122218561RT22	18561	8/5/2022	12/5/2022	14-05-22	-12.0	RT2	makeyourtrip	NaN	Cancelled	
18119	May122218562RT311	18562	5/5/2022	12/5/2022	17-05-22	-6.0	RT3	direct offline	5.0	Checked Out	
18121	May122218562RT313	18562	10/5/2022	12/5/2022	17-05-22	-4.0	RT3	direct online	NaN	Cancelled	
56715	Jun082218562RT12	18562	5/6/2022	8/6/2022	13-06-22	-17.0	RT1	others	NaN	Checked Out	
119765	Jul202219560RT220	19560	19-07-22	20-07-22	22-07-22	-1.0	RT2	others	NaN	Checked Out	
134586	Jul312217564RT47	17564	30-07-22	31-07-22	1/8/2022	-4.0	RT4	logtrip	2.0	Checked Out	

As you can see above, number of guests having less than zero value represents data error. We can ignore these records.

```
[267]: df_bookings = df_bookings[df_bookings.no_guests>0]
```

```
[268]: df_bookings.shape
```

```
[268]: (134578, 12)
```

Data Cleaning

```
Jupyter Exploratory Data Analysis Exercise Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Python 3 (ipykernel)
(1345/8, 12)

(2) Outlier removal in revenue generated

[269]: df_bookings.revenue_generated.min(), df_bookings.revenue_generated.max()
[269]: (6500, 28560000)
[270]: df_bookings.revenue_generated.mean(), df_bookings.revenue_generated.median()
[270]: (15378.036937686695, 13500.0)
[271]: avg, std = df_bookings.revenue_generated.mean(), df_bookings.revenue_generated.std()
[272]: higher_limit = avg + 3*std
[272]: higher_limit
[272]: 294498.50173207896
[273]: lower_limit = avg - 3*std
[273]: lower_limit
[273]: -263742.4278567056
[274]: df_bookings[df_bookings.revenue_generated<=0]
[274]: booking_id property_id booking_date check_in_date checkout_date no_guests room_category booking_platform ratings_given booking_status revenue_generated
[275]: df_bookings[df_bookings.revenue_generated>higher_limit]
[275]: booking_id property_id booking_date check_in_date checkout_date no_guests room_category booking_platform ratings_given booking_status revenue_generated
```

Detecting and correcting revenue outliers to ensure reliable insights

Applying techniques such as standard deviation to ensure correct data loading.

```
Jupyter Exploratory Data Analysis Exercise Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Python 3 (ipykernel)
1299 rows x 12 columns

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

[280]: df_bookings[df_bookings.room_category=="RT4"].revenue_realized.describe()
[280]: 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
[281]: # mean + 3*standard deviation
[281]: 23439+3*9048
[281]: 50583
```

Here higher limit comes to be 50583 and in our dataframe above we can see that max value for revenue realized is 45220. Hence we can conclude that there is no outlier and we don't need to do any data cleaning on this particular column

Data Transformation

Jupyter Exploratory Data Analysis Exercise Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Not Trusted

JupyterLab Python 3 (ipykernel)

```
[295]: df_agg_bookings['occ_pct'] = df_agg_bookings['occ_pct'].apply(lambda x: round(x*100, 2))
df_agg_bookings.head(3)
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct
0	16559	1-May-22	RT1	25	30.0	83.33
1	19562	1-May-22	RT1	28	30.0	93.33
2	19563	1-May-22	RT1	23	30.0	76.67

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

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	4.0	Checked Out	
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1	others	NaN	Cancelled	
7	May012216558RT18	16558	26-04-22	1/5/2022	3/5/2022	2.0	RT1	logtrip	NaN	No Show	

```
[297]: df_agg_bookings.info()
```

Data transformation is the process of modifying data through business rules to make it suitable for analysis.

Created New column called "occ_pct"

Common transformations are column creation, normalization, data merging, and aggregation.

Insights

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

```
exercise_solution Last Checkpoint: 18 seconds ago

ew Run Kernel Settings Help

] Code Python 3

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

room_class
lite      58.01
premium   58.03
residential 59.28
standard   57.89
Name: occ_pct, dtype: float64

import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors

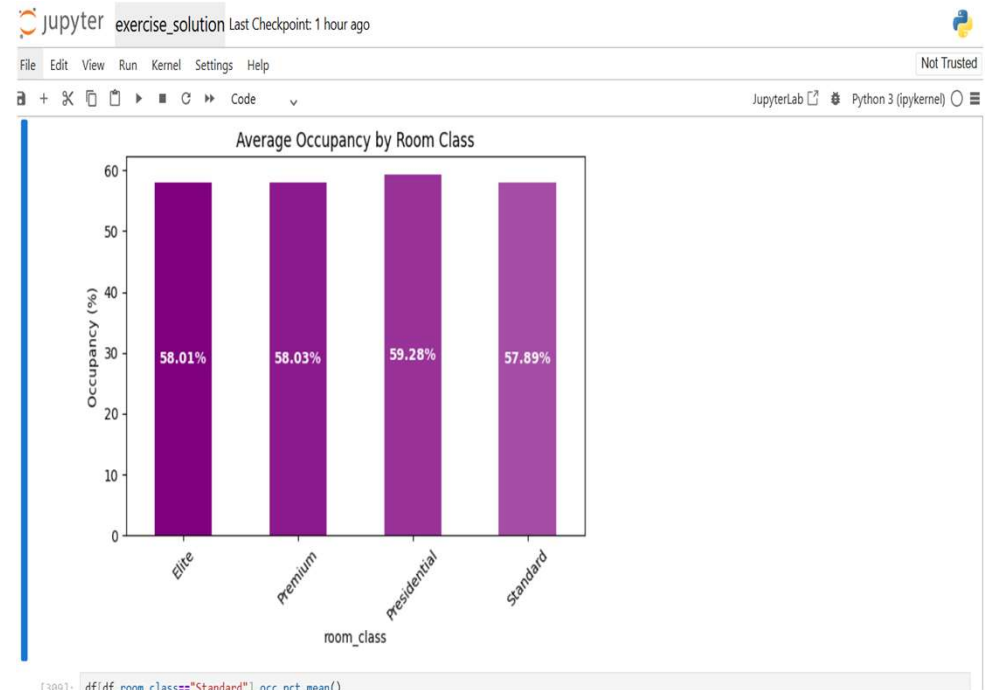
# Sample data
data = {
    'room_class': ['Elite', 'Premium', 'Presidential', 'Standard'],
    'occ_pct': [58.01, 58.03, 59.28, 57.89]
}
df = pd.DataFrame(data)

# Generate shades of a single base color
base_color = mcolors.to_rgb('purple') # blue
shades = [mcolors.to_hex((base_color[0] * (1 - i*0.1) + i*0.1,
                           base_color[1] * (1 - i*0.1) + i*0.1,
                           base_color[2] * (1 - i*0.1) + i*0.1))
           for i in range(len(df))]

# Create the plot
x = df.groupby("room_class")["occ_pct"].mean().round(2).plot(
    kind="bar",
    title="Average Occupancy by Room Class",
    ylabel="Occupancy (%)",
```

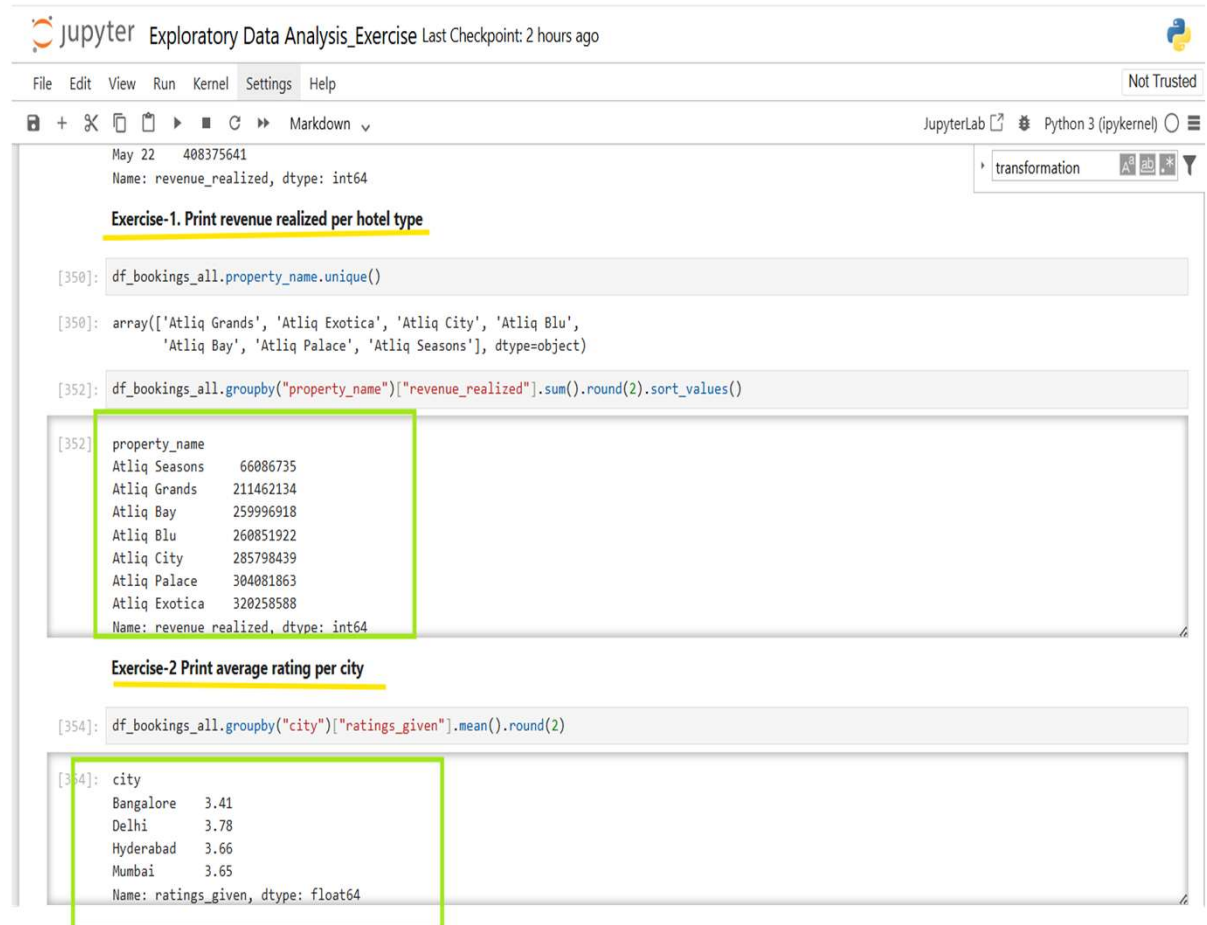
Presidential rooms outperform peers with the highest occupancy at 59.28%, indicating strong guest preference for luxury offerings.

Occupancy rates are tightly clustered between 57.89% and 59.28%, suggesting uniform demand across all room classes and limited sensitivity to class-based pricing.



2. Print revenue realized per hotel type

1. *Atliq Exotica* generated the highest revenue (~32.03M).
2. Most hotels are performing in the 25–30M range, except *Atliq Seasons*, which is significantly lower.
3. Customer satisfaction is relatively higher in Delhi and lower in Bangalore. Management can investigate why Bangalore ratings are lower and improve services there.



Jupyter Exploratory Data Analysis Exercise Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

May 22 408375641
Name: revenue_realized, dtype: int64

transformation

Exercise-1. Print revenue realized per hotel type

```
[350]: df_bookings_all.property_name.unique()
[350]: array(['Atliq Grands', 'Atliq Exotica', 'Atliq City', 'Atliq Blu',
          'Atliq Bay', 'Atliq Palace', 'Atliq Seasons'], dtype=object)
[352]: df_bookings_all.groupby("property_name")["revenue_realized"].sum().round(2).sort_values()
```

```
[352]: property_name
      Atliq Seasons    66086735
      Atliq Grands   211462134
      Atliq Bay      259996918
      Atliq Blu      260851922
      Atliq City     285798439
      Atliq Palace   304081863
      Atliq Exotica  320258588
      Name: revenue realized, dtype: int64
```

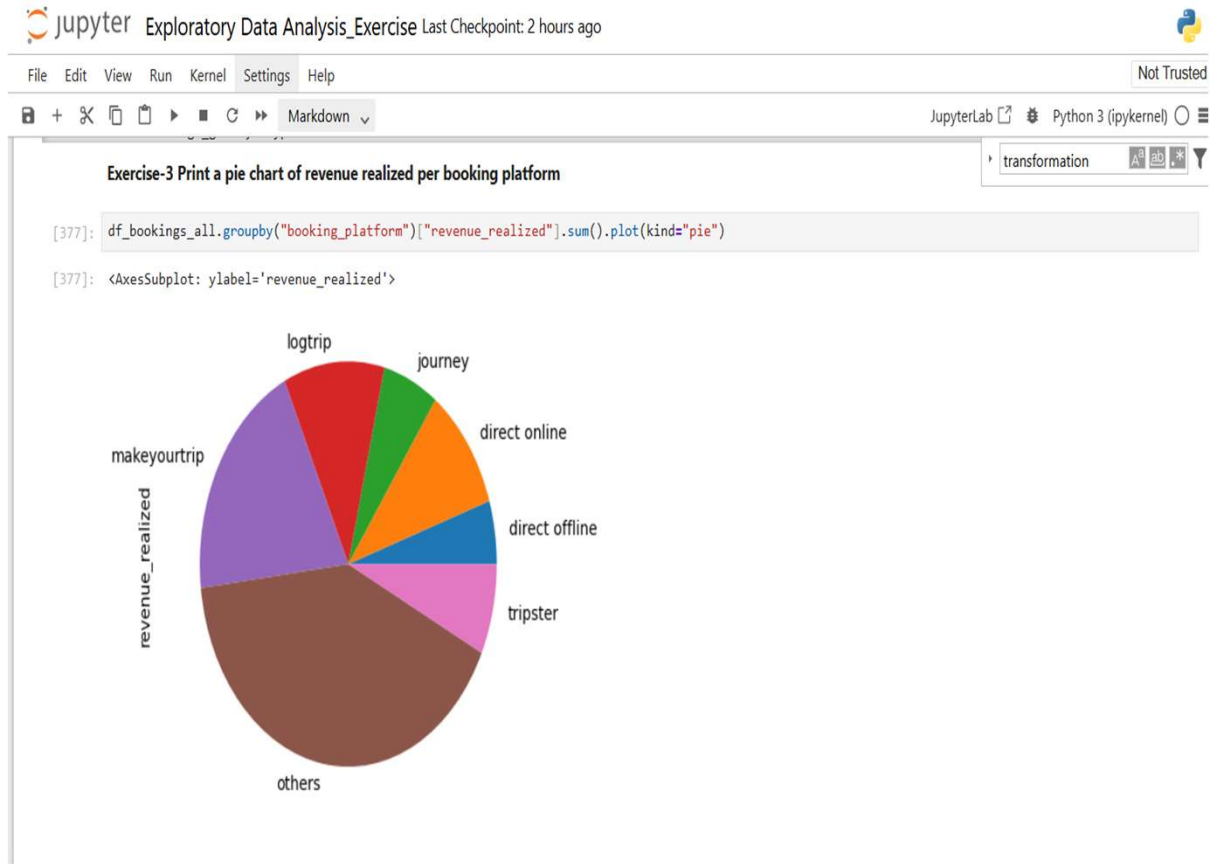
Exercise-2 Print average rating per city

```
[354]: df_bookings_all.groupby("city")["ratings_given"].mean().round(2)
```

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

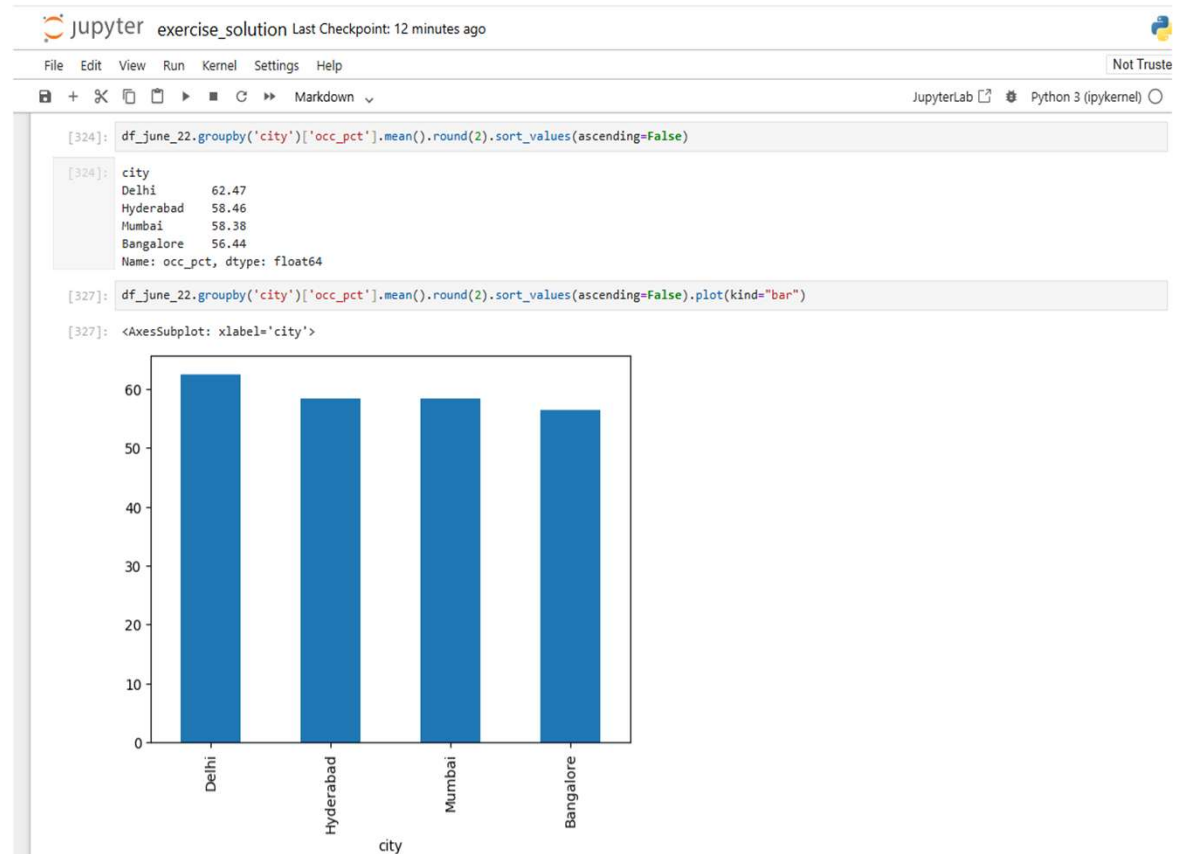

3. Print a pie chart of revenue realized per booking platform

1. **"Others"** platform dominates revenue realization, contributing the largest share compared to all other platforms.
2. **Makeyourtrip** is the second-largest contributor, with a significant portion of total revenue.
3. Platforms like **Tripster, Logtrip, and Journey** have relatively smaller contributions to overall revenue.
4. **Direct offline** contributes the least among all categories, indicating less reliance on offline channels.



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

1. Delhi Leads with Strong Demand: At 62.47% average occupancy.
2. Hyderabad and Mumbai Cluster Around 58% ,show nearly identical performance.
3. Bangalore Lags Slightly. Targeted promotions could help close this gap and boost utilization.





Thank you

