## Load and Preview the Data

```
import pandas as pd

# Load the data
dim_hotels = pd.read_csv('dim_hotels.csv')
dim_date = pd.read_csv('dim_date.csv')
dim_rooms = pd.read_csv('dim_rooms.csv')
fact_aggregated_bookings = pd.read_csv('fact_aggregated_bookings.csv')
fact_bookings = pd.read_csv('fact_bookings.csv')

# Display the first few rows of each dataset
dim_hotels.head(), dim_rooms.head(),dim_date.head(),
fact_aggregated_bookings.head(),fact_bookings.head(),
```

```
(   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,
   room_id    room_class
 0     RT1      Standard
 1     RT2         Elite
 2     RT3       Premium
 3     RT4  Presidential,
         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,
    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,
          booking_id  property_id booking_date check_in_date
checkout_date  \
 0  May012216558RT11        16558   2022-04-27    2022-05-01    2022-
05-02
 1  May012216558RT12        16558   2022-04-30    2022-05-01    2022-
```

```
05-02
 2  May012216558RT13        16558   2022-04-28    2022-05-01    2022-
05-04
 3  May012216558RT14        16558   2022-04-28    2022-05-01    2022-
05-02
 4  May012216558RT15        16558   2022-04-27    2022-05-01    2022-
05-02

    no_guests room_category booking_platform  ratings_given
booking_status  \
 0          3           RT1    direct online            1.0    Checked
Out
 1          2           RT1           others            NaN
Cancelled
 2          2           RT1          logtrip            5.0    Checked
Out
 3          2           RT1           others            NaN
Cancelled
 4          4           RT1    direct online            5.0    Checked
Out

    revenue_generated  revenue_realized
 0              10010             10010
 1               9100              3640
 2               9100              9100
 3               9100              3640
 4              10920             10920  )
```

## Data Merging and Cleaning

```
# Merge fact_aggregated_bookings with dim_hotels
bookings_hotels = fact_aggregated_bookings.merge(dim_hotels,
on='property_id')

# Merge the resulting dataframe with dim_rooms
complete_data = bookings_hotels.merge(dim_rooms,
left_on='room_category', right_on='room_id')

# Display the first few rows of the merged dataset
complete_data.head()
```

```
   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

   property_name  category       city room_id room_class
0  Atliq Exotica    Luxury     Mumbai     RT1   Standard
1      Atliq Bay    Luxury  Bangalore     RT1   Standard
2   Atliq Palace  Business  Bangalore     RT1   Standard
3   Atliq Grands    Luxury     Mumbai     RT1   Standard
4   Atliq Grands    Luxury      Delhi     RT1   Standard
```

```
# Merge complete_data with dim_date on check_in_date
complete_data = complete_data.merge(dim_date, left_on='check_in_date',
right_on='date')

# Display the first few rows to check the merge
print(complete_data.head())
```

```
   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

   property_name  category       city room_id room_class       date
mmm yy  \
0  Atliq Exotica    Luxury     Mumbai     RT1   Standard  01-May-22
May 22
1      Atliq Bay    Luxury  Bangalore     RT1   Standard  01-May-22
May 22
2   Atliq Palace  Business  Bangalore     RT1   Standard  01-May-22
May 22
3   Atliq Grands    Luxury     Mumbai     RT1   Standard  01-May-22
May 22
4   Atliq Grands    Luxury      Delhi     RT1   Standard  01-May-22
May 22

  week no day_type
0    W 19  weekend
1    W 19  weekend
2    W 19  weekend
```

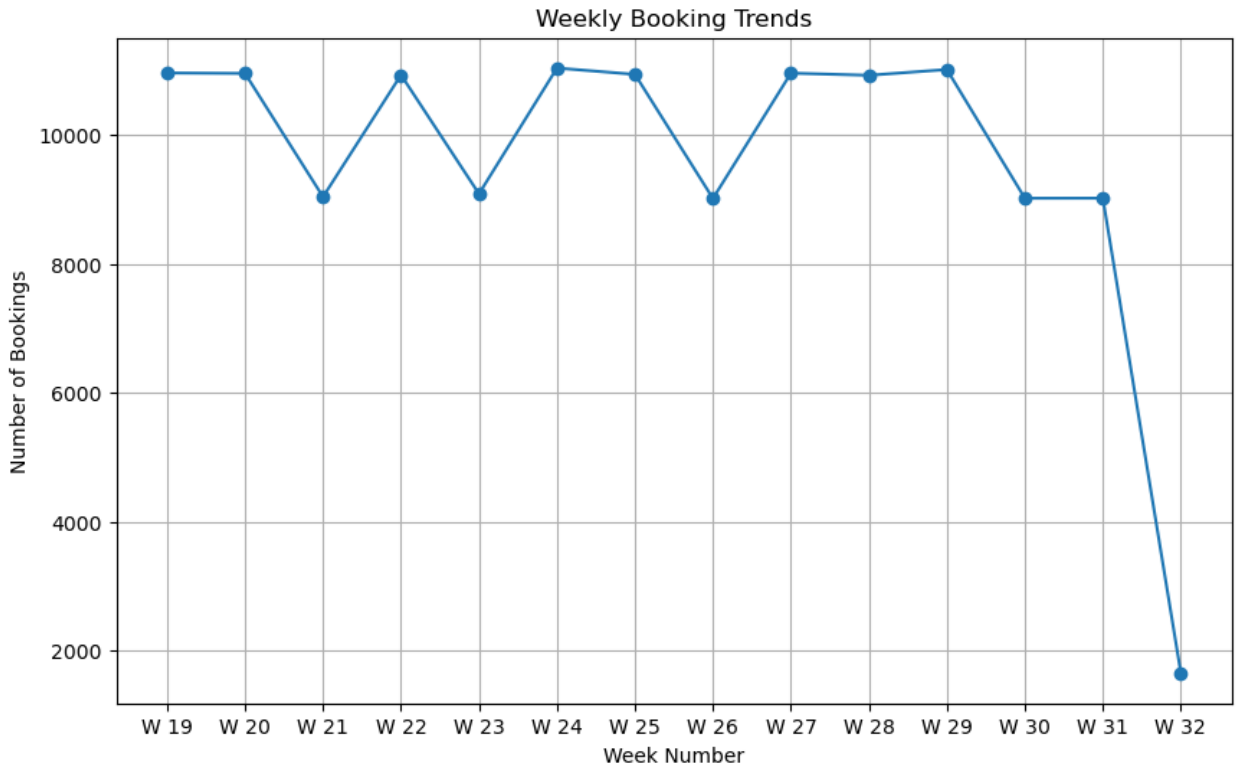
```
3    W 19  weekend
4    W 19  weekend
```

## Creating Visualizations

Weekly Booking Trends:

```
import matplotlib.pyplot as plt
# Group by week and sum the bookings
weekly_trends = complete_data.groupby('week no')
['successful_bookings'].sum().reset_index()

# Plotting the weekly trends
plt.figure(figsize=(10, 6))
plt.plot(weekly_trends['week no'],
weekly_trends['successful_bookings'], marker='o')
plt.title('Weekly Booking Trends')
plt.xlabel('Week Number')
plt.ylabel('Number of Bookings')
plt.grid(True)
plt.show()
```

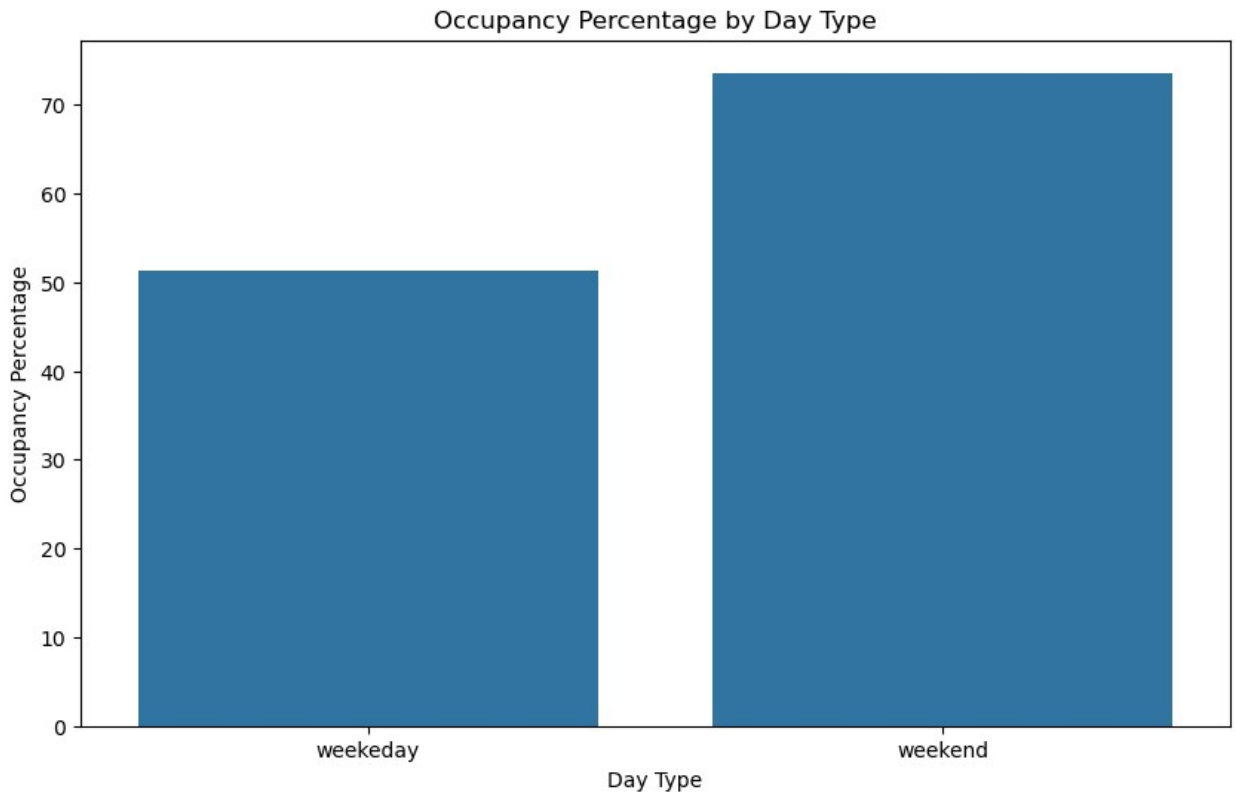



Occupancy by Day Type

```
import seaborn as sns
```

```
# Group by day type and calculate occupancy
occupancy_by_day_type = complete_data.groupby('day_type').agg(
    successful_bookings=('successful_bookings', 'sum'),
    capacity=('capacity', 'sum')
).reset_index()
occupancy_by_day_type['occupancy_percentage'] =
(occupancy_by_day_type['successful_bookings'] /
occupancy_by_day_type['capacity']) * 100

# Plotting occupancy by day type
plt.figure(figsize=(10, 6))
sns.barplot(x='day_type', y='occupancy_percentage',
data=occupancy_by_day_type)
plt.title('Occupancy Percentage by Day Type')
plt.xlabel('Day Type')
plt.ylabel('Occupancy Percentage')
plt.show()
```




Booking Percentage by Property Category:

```
# Group by property category and calculate percentage
booking_by_category =
complete_data['category'].value_counts(normalize=True) * 100

# Plotting booking percentage by property category
```

```
plt.figure(figsize=(10, 6))
booking_by_category.plot(kind='bar')
plt.title('Booking Percentage by Property Category')
plt.xlabel('Property Category')
plt.ylabel('Percentage')
plt.show()
```