# Hotel Booking EDA Project

February 20, 2025

## 1 Hotel Booking EDA Project

[94]:
```
from PIL import Image
image_path = ("C:\\Users\\User\\Downloads\\hotel booking.jpg")
image = Image.open(image_path)
image
```

[94]:

## 2 Introduction

The hospitality industry is a dynamic and highly competitive sector that thrives on understanding customer behavior and optimizing operations. My exploratory data analysis (EDA) project on hotel bookings aims to uncover insights that can drive strategic decision-making and enhance customer experiences. By analyzing booking data, I seek to identify key trends, patterns, and anomalies that can inform hotel management practices and marketing strategies.

In this project, I will delve into various aspects of hotel bookings, such as booking lead times, cancellation rates, customer demographics, and seasonal trends. Through detailed visualizations and feature engineering, I aim to transform raw data into actionable insights. My goal is to provide

a comprehensive overview of the factors influencing hotel bookings and to highlight opportunities for improving occupancy rates and customer satisfaction.

[75]:
```
# Importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

### 2.1 Data Loading

[2]:
```
# Step 1: Load the Dataset
# Load the dataset into a Pandas DataFrame
file_path = 'C:\\Users\\User\\Downloads\\Hotel Bookings (1).csv'
df = pd.read_csv(file_path)
```

### 2.2 Data Inspection

[3]:
```
# Step 2: Data Inspection
# Checking the first few rows
display(df.head())
```

```
          hotel  is_canceled  lead_time  arrival_date_year arrival_date_month  \
0  Resort Hotel            0        342               2015               July
1  Resort Hotel            0        737               2015               July
2  Resort Hotel            0          7               2015               July
3  Resort Hotel            0         13               2015               July
4  Resort Hotel            0         14               2015               July

   arrival_date_week_number  arrival_date_day_of_month  \
0                        27                          1
1                        27                          1
2                        27                          1
3                        27                          1
4                        27                          1

   stays_in_weekend_nights  stays_in_week_nights  adults  …  deposit_type  \
0                        0                     0       2  …    No Deposit
1                        0                     0       2  …    No Deposit
2                        0                     1       1  …    No Deposit
3                        0                     1       1  …    No Deposit
4                        0                     2       2  …    No Deposit

   agent company days_in_waiting_list customer_type   adr  \
0    NaN     NaN                    0     Transient   0.0
1    NaN     NaN                    0     Transient   0.0
2    NaN     NaN                    0     Transient  75.0
3  304.0     NaN                    0     Transient  75.0
```

```
4  240.0     NaN                    0     Transient  98.0

   required_car_parking_spaces  total_of_special_requests  reservation_status  \
0                            0                          0           Check-Out
1                            0                          0           Check-Out
2                            0                          0           Check-Out
3                            0                          0           Check-Out
4                            0                          1           Check-Out

  reservation_status_date
0              2015-07-01
1              2015-07-01
2              2015-07-02
3              2015-07-02
4              2015-07-03

[5 rows x 32 columns]
```

[4]:
```
# Checking basic information about the dataset
df.info()
```

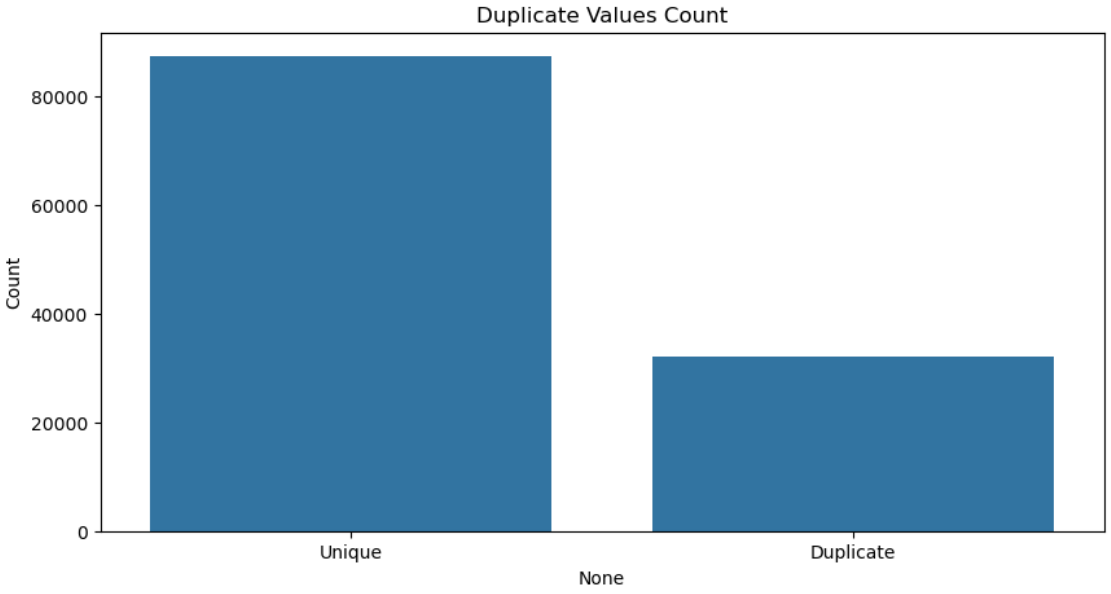
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   arrival_date_year               119390 non-null  int64
 4   arrival_date_month              119390 non-null  object
 5   arrival_date_week_number        119390 non-null  int64
 6   arrival_date_day_of_month       119390 non-null  int64
 7   stays_in_weekend_nights         119390 non-null  int64
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         118902 non-null  object
 14  market_segment                  119390 non-null  object
 15  distribution_channel            119390 non-null  object
 16  is_repeated_guest               119390 non-null  int64
 17  previous_cancellations          119390 non-null  int64
 18  previous_bookings_not_canceled  119390 non-null  int64
 19  reserved_room_type              119390 non-null  object
 20  assigned_room_type              119390 non-null  object
 21  booking_changes                 119390 non-null  int64
```

```
 22  deposit_type                    119390 non-null  object
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64
 26  customer_type                   119390 non-null  object
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  object
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB
```

[5]:
```
# Checking for duplicate values
duplicate_count = df.duplicated().sum()
print("Number of duplicate rows:", duplicate_count)
```

```
Number of duplicate rows: 31994
```

[6]:
```
# Visualizing duplicate values
plt.figure(figsize=(10, 5))
duplicate_counts = df.duplicated().value_counts()
sns.barplot(x=duplicate_counts.index, y=duplicate_counts.values)
plt.xticks(ticks=[0, 1], labels=['Unique', 'Duplicate'])
plt.ylabel('Count')
plt.title('Duplicate Values Count')
plt.show()
```



### 2.3 Data Wrangling

[7]:
```
# Remove duplicate values
df.drop_duplicates(inplace=True)
```

[8]:
```
# Checking for missing values
missing_values = df.isnull().sum()
print("Missing Values in Each Column:")
print(missing_values[missing_values > 0])
```

```
Missing Values in Each Column:
children        4
country       452
agent       12193
company     82137
dtype: int64
```

[9]:
```
# Dropping 'company' column due to many missing values
df.drop(columns=['company'], inplace=True)
```

[10]:
```
df.isna().sum()
```

[10]:
```
hotel                           0
is_canceled                     0
lead_time                       0
arrival_date_year               0
arrival_date_month              0
arrival_date_week_number        0
arrival_date_day_of_month       0
stays_in_weekend_nights         0
stays_in_week_nights            0
adults                          0
children                        4
babies                          0
meal                            0
country                       452
market_segment                  0
distribution_channel            0
is_repeated_guest               0
previous_cancellations          0
previous_bookings_not_canceled  0
reserved_room_type              0
assigned_room_type              0
booking_changes                 0
deposit_type                    0
agent                       12193
days_in_waiting_list            0
customer_type                   0
```

```
adr                                 0
required_car_parking_spaces         0
total_of_special_requests           0
reservation_status                  0
reservation_status_date             0
dtype: int64
```

[11]:
```
# Handle missing values
df.dropna(inplace=True)
```

### 2.4 Data Transformation

[12]:
```
# Creating a new column for total guests
df['total_guests'] = df['adults'] + df['children'] + df['babies']
```

[17]:
```
# Convert 'arrival_date' columns into a single datetime column
df['arrival_date'] = pd.to_datetime(df['arrival_date_year'].astype(str) + '-' +
                                    df['arrival_date_month'] + '-' +
                                    df['arrival_date_day_of_month'].astype(str),
                                    format='%Y-%B-%d')
```

[14]:
```
# Drop unnecessary columns
df.drop(columns=['arrival_date_week_number'], inplace=True)
```

### 2.5 Feature Engineering

Feature Engineering is the process of creating new features or modifying existing ones to improve model performance or gain better insights.

[18]:
```
# Booking Window:
# Create a feature for the number of days between booking and arrival.
df['booking_window'] = (pd.to_datetime(df['arrival_date']) - pd.
 ↪to_datetime(df['reservation_status_date'])).dt.days
```

[19]:
```
# Total Nights: Sum of weekend and weekday nights.
df['total_nights'] = df['stays_in_weekend_nights'] + df['stays_in_week_nights']
```

[21]:
```
# Has Kids: Create a binary feature
# indicating whether the booking includes children or babies
df['has_kids'] = df[['children', 'babies']].sum(axis=1).apply(lambda x: 1 if x␣
 ↪> 0 else 0)
```

[22]:
```
# Average Daily Rate per Guest: Create a feature for the average daily rate per␣
 ↪guest.
df['adr_per_guest'] = df['adr'] / df['total_guests']
```

[23]:
```
# Is High Season: Create a binary feature
# indicating high or low season based on the arrival month.
```

```
high_season = ['July', 'August', 'September']
df['is_high_season'] = df['arrival_date_month'].apply(lambda x: 1 if x in␣
 ↪high_season else 0)
```

[24]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 75074 entries, 3 to 119389
Data columns (total 37 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   hotel                           75074 non-null  object
 1   is_canceled                     75074 non-null  int64
 2   lead_time                       75074 non-null  int64
 3   arrival_date_year               75074 non-null  int64
 4   arrival_date_month              75074 non-null  object
 5   arrival_date_day_of_month       75074 non-null  int64
 6   stays_in_weekend_nights         75074 non-null  int64
 7   stays_in_week_nights            75074 non-null  int64
 8   adults                          75074 non-null  int64
 9   children                        75074 non-null  float64
 10  babies                          75074 non-null  int64
 11  meal                            75074 non-null  object
 12  country                         75074 non-null  object
 13  market_segment                  75074 non-null  object
 14  distribution_channel            75074 non-null  object
 15  is_repeated_guest               75074 non-null  int64
 16  previous_cancellations          75074 non-null  int64
 17  previous_bookings_not_canceled  75074 non-null  int64
 18  reserved_room_type              75074 non-null  object
 19  assigned_room_type              75074 non-null  object
 20  booking_changes                 75074 non-null  int64
 21  deposit_type                    75074 non-null  object
 22  agent                           75074 non-null  float64
 23  days_in_waiting_list            75074 non-null  int64
 24  customer_type                   75074 non-null  object
 25  adr                             75074 non-null  float64
 26  required_car_parking_spaces     75074 non-null  int64
 27  total_of_special_requests       75074 non-null  int64
 28  reservation_status              75074 non-null  object
 29  reservation_status_date         75074 non-null  object
 30  total_guests                    75074 non-null  float64
 31  arrival_date                    75074 non-null  datetime64[ns]
 32  booking_window                  75074 non-null  int64
 33  total_nights                    75074 non-null  int64
 34  has_kids                        75074 non-null  int64
 35  adr_per_guest                   74987 non-null  float64
```

```
 36  is_high_season                  75074 non-null  int64
dtypes: datetime64[ns](1), float64(5), int64(19), object(12)
memory usage: 21.8+ MB
```

#### 2.5.1 Data Overview: Before & After Cleaning and tranformation

`Before Cleaning` 1. Total Entries: 119,390 2. Total Columns: 32 3. Contained redundant and unstructured date columns (arrival_date_year, arrival_date_month, arrival_date_day_of_month). 4. Some columns had missing values (children, country, agent, company). 5. Had unnecessary columns like arrival_date_week_number.

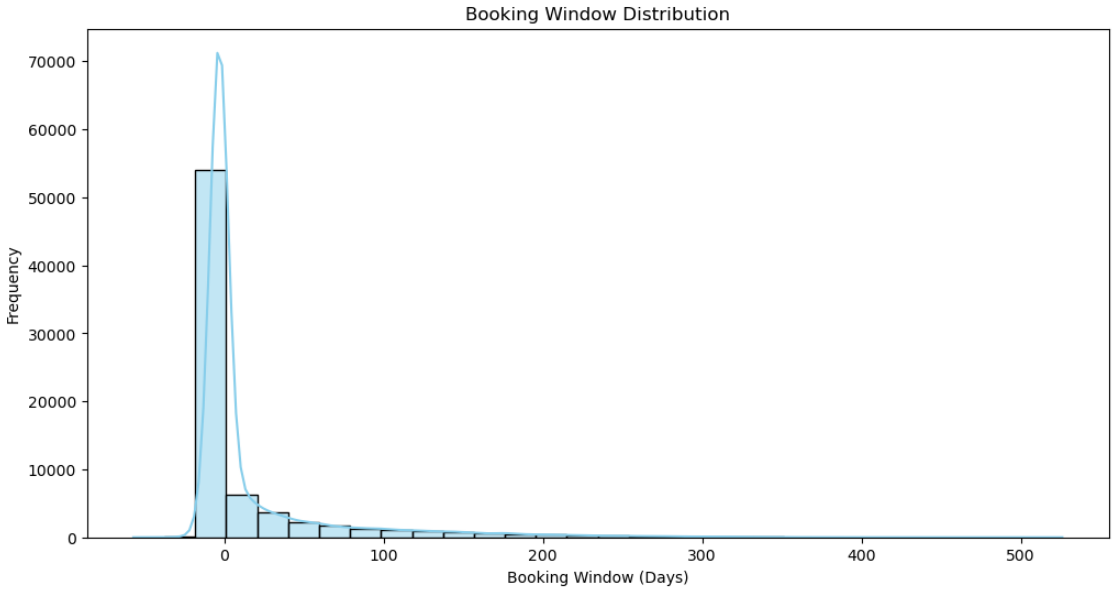
`After Cleaning` 1. Total Entries: 75,074 (after filtering/reducing irrelevant data). 2. Total Columns: 31 (removal of redundant ones). 3. Added further 6 columns (total 37) via FE. 5. Removed arrival_date_week_number as it was redundant. 6. Ensured missing values were handled appropriately.

Final dataset is now well-structured, making it easier for analysis and visualization.

### 2.6 Step 5: Data Visualization

#### 2.6.1 1. Booking Window Distribution

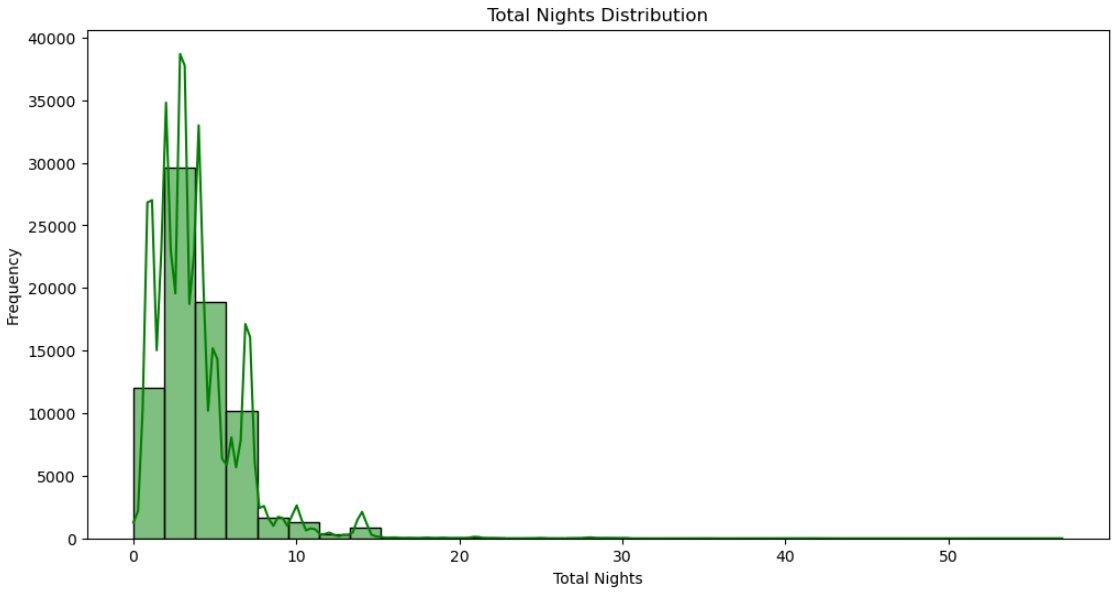
[25]:
```
plt.figure(figsize=(12, 6))
sns.histplot(df['booking_window'], bins=30, kde=True, color='skyblue')
plt.title('Booking Window Distribution')
plt.xlabel('Booking Window (Days)')
plt.ylabel('Frequency')
plt.show()
```



Reason: A histogram with a KDE plot is ideal for understanding the distribution of continuous numerical data. This will show how far in advance most bookings are made.

#### 2.6.2 2. Total Nights Distribution

[83]:
```
plt.figure(figsize=(12, 6))
sns.histplot(df['total_nights'], bins=30, kde=True, color='green')
plt.title('Total Nights Distribution')
plt.xlabel('Total Nights')
plt.ylabel('Frequency')
plt.show()
```




- Visualization:
  –

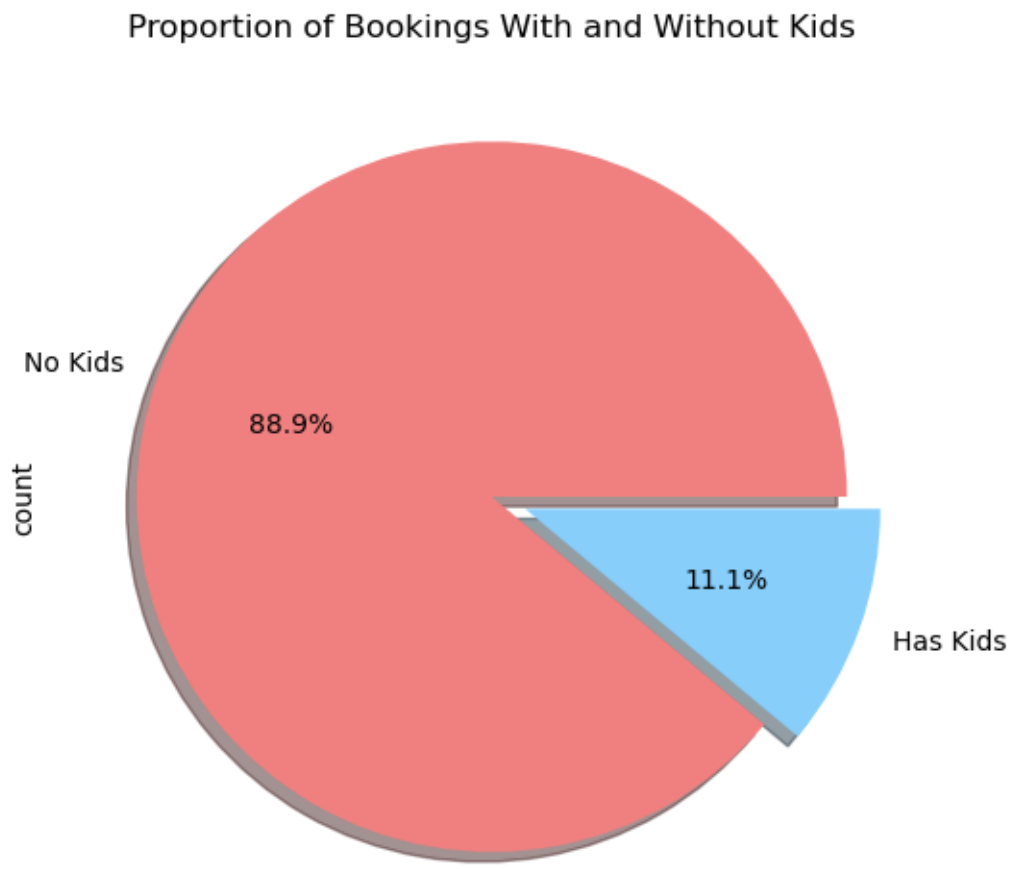
**2.7 Histograms are excellent for visualizing the distribution of a single continuous variable. They allow us to see the frequency of different ranges of values.**

- Insights:
  – The histogram shows that the majority of bookings are for shorter stays, with most bookings falling between 0 and 10 nights.
  –

**2.8 There are a few bookings with longer stays, but they are less frequent.**

### 2.8.1 3. Has Kids Proportion

[28]:
```
plt.figure(figsize=(8, 6))
explode = [0,0.1]
df['has_kids'].value_counts().plot(kind='pie', autopct='%1.1f%%',␣
 ↪colors=['lightcoral', 'lightskyblue'],shadow= True, explode = explode,␣
 ↪labels=['No Kids', 'Has Kids'])
plt.title('Proportion of Bookings With and Without Kids')
plt.show()
```

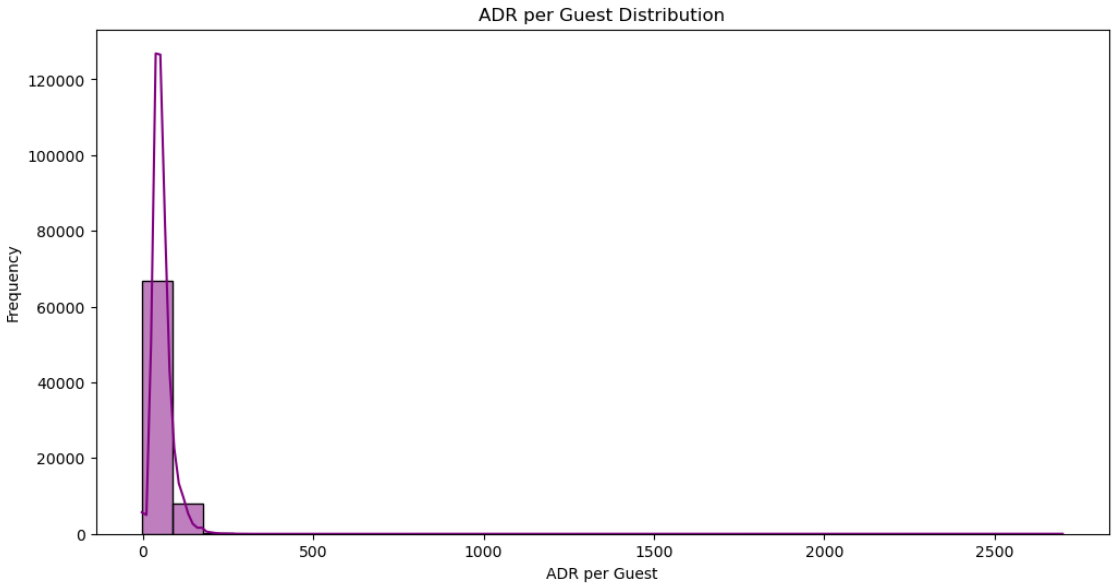



- Visualization:
  - A pie chart is suitable for showing proportions within a categorical variable, making it easy to compare bookings with and without kids.
- Insights:
  - The pie chart shows that a significant majority, 88.9% of bookings, are made without

kids.
  - This insight suggests that most customers prefer to book alone or with adult companions.
  - This information can help in tailoring services and amenities specifically for families, such as family rooms or child-friendly activities.

### 2.8.2 4. ADR per Guest Distribution

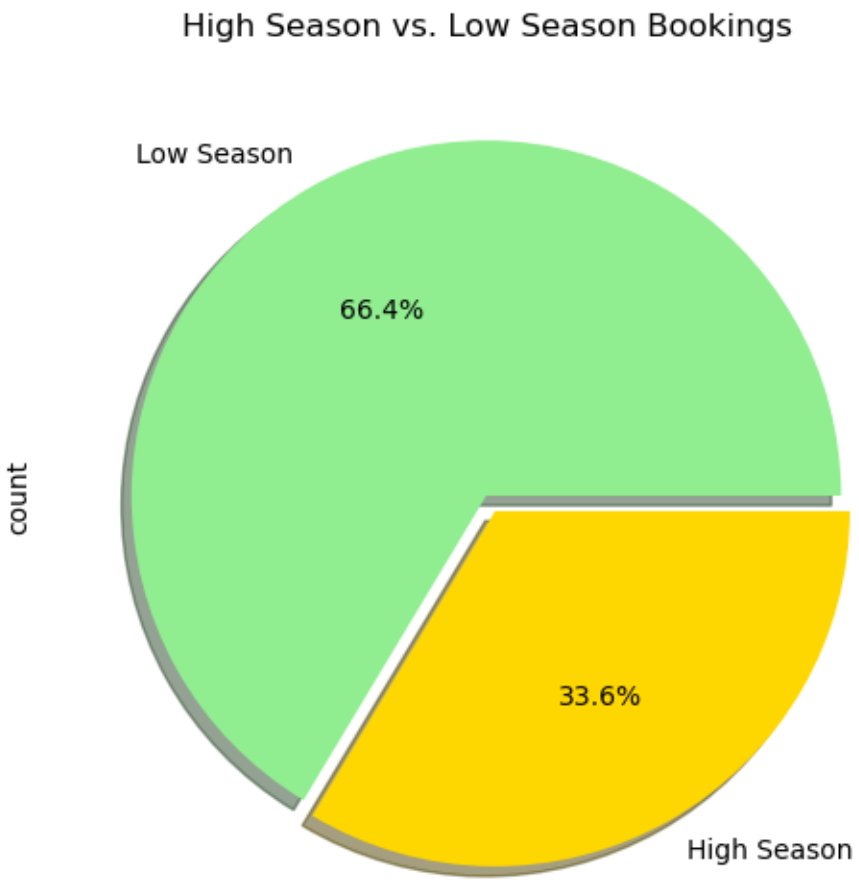
[33]:
```
plt.figure(figsize=(12, 6))
sns.histplot(df['adr_per_guest'], bins=30, kde=True, color='purple')
plt.title('ADR per Guest Distribution')
plt.xlabel('ADR per Guest')
plt.ylabel('Frequency')
plt.show()
```




- Visualization:
  - A histogram with KDE helps visualize the distribution of average daily rate per guest, highlighting common price points.
- Insights:
  - The histogram shows that the majority of ADR (Average Daily Rate) per guest values are concentrated near the lower end of the range. This indicates that most bookings have a relatively low ADR per guest.
  - There is a long tail extending towards higher ADR per guest values, suggesting that while higher rates are less common, they do exist in the dataset.

### 2.8.3 5. High Season Booking Proportion

[37]:
```
plt.figure(figsize=(8, 6))
explode = [0,0.05]
df['is_high_season'].value_counts().plot(kind='pie', autopct='%1.1f%%',␣
↪colors=['lightgreen', 'gold'], shadow= True, explode = explode, labels=['Low␣
↪Season', 'High Season'])
plt.title('High Season vs. Low Season Bookings')
plt.show()
```

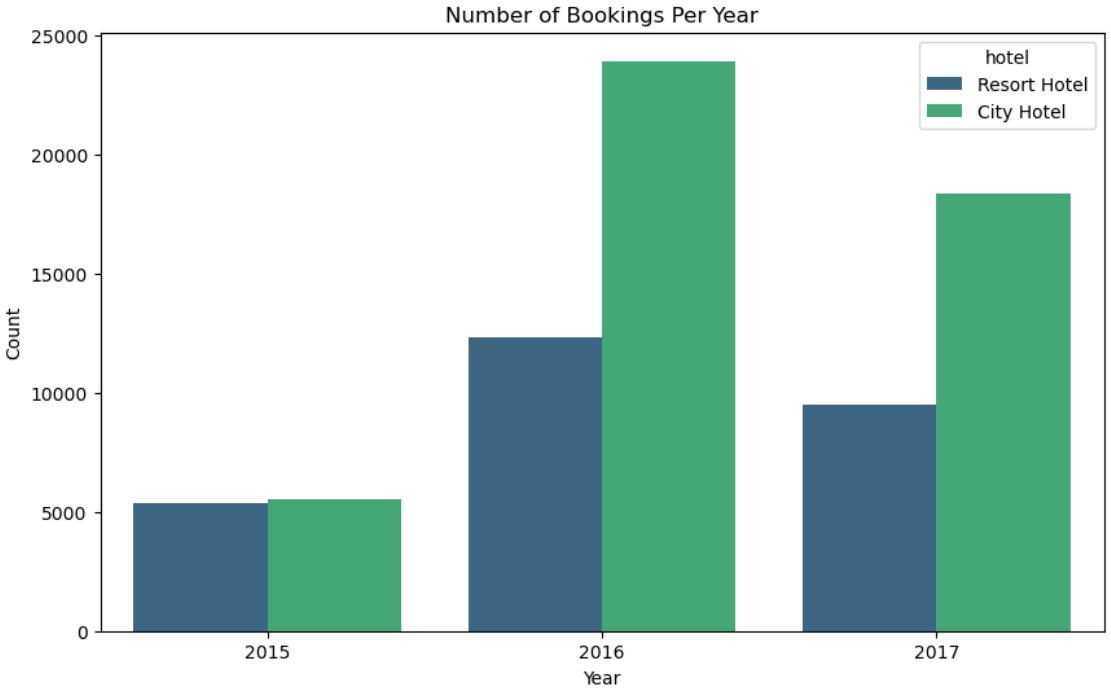



- Visualization:
  - Pie charts help in understanding the proportion of bookings during high and low seasons, providing insights into booking trends. Pie chart makes it easy to understand at a glance. ***
- Insights:
  - This insight indicates that the majority of customers prefer to book during the low season, possibly due to lower rates, fewer crowds, or more availability.
  - Understanding this distribution can help in planning marketing strategies, special offers,

and resource allocation. ***

### 2.8.4 6. Number of Bookings Per Year

[41]:
```
plt.figure(figsize=(10, 6))
sns.countplot(x='arrival_date_year', data=df, hue= 'hotel', palette='viridis')
plt.title('Number of Bookings Per Year')
plt.xlabel('Year')
plt.ylabel('Count')
plt.show()
```

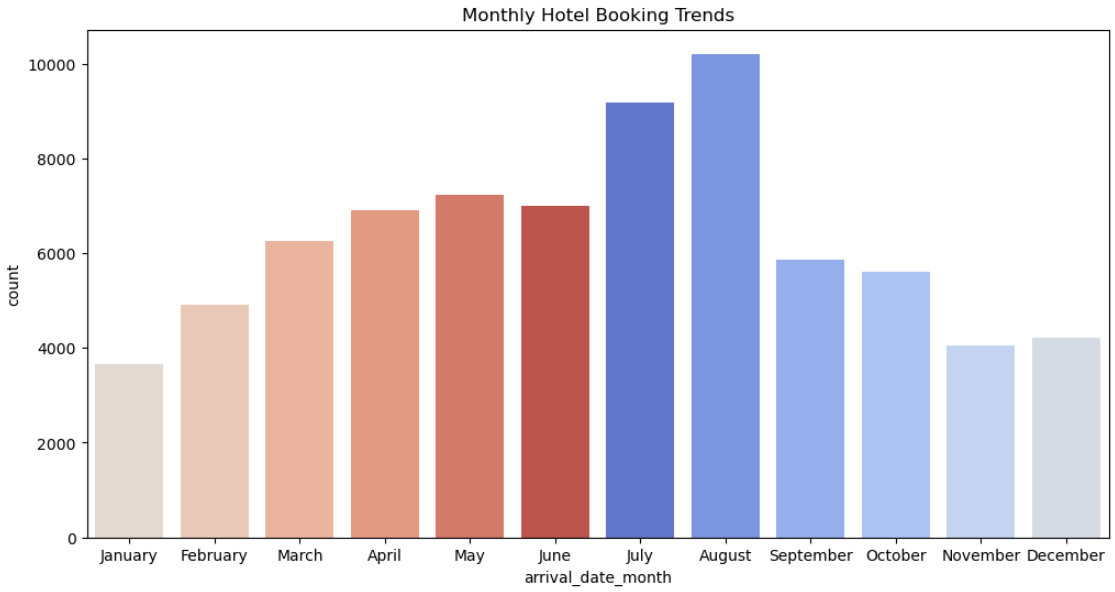



Visualization Used : Bar Chart * Reason: A bar chart is used to compare categorical data, making it ideal for analyzing trends over time. **Insights: 1. For example, if 2016 has more bookings than 2015 and 2017, it could indicate business growth 2. city hotels have more booking than resort hotel.** *

### 2.8.5 7. What is the monthly trend of hotel bookings?

[17]:
```
plt.figure(figsize=(12, 6))
sns.countplot(x='arrival_date_month', data=df, hue='arrival_date_month',␣
 ↪palette='coolwarm', order=[
    'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',␣
 ↪'September', 'October', 'November', 'December'
])
plt.title('Monthly Hotel Booking Trends')
```

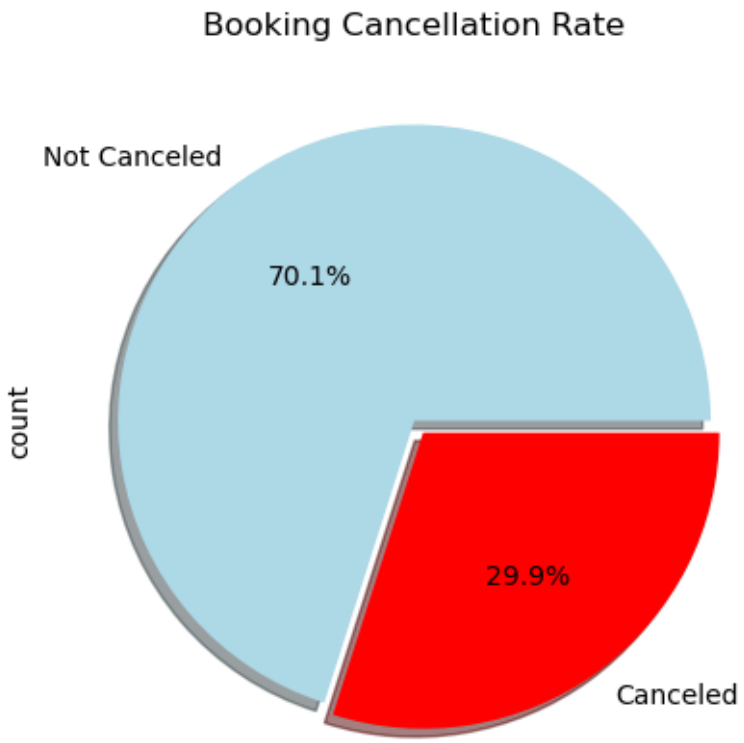
```
plt.show()
```




Visualization: Bar Chart * Reason: The bar chart helps in identifying seasonal trends in hotel bookings. Insights**: * July and August show higher bookings, it indicates these are peak months. * Hotels can increase room rates during these months.** *

### 2.8.6 8. What is the proportion of canceled bookings?

[32]:
```
plt.figure(figsize=(10, 5))
explode = [0,0.05]
df['is_canceled'].value_counts().plot(kind='pie', autopct='%1.1f%%',␣
↪colors=['lightblue', 'red'],shadow= True, explode = explode, labels=['Not␣
↪Canceled', 'Canceled'])
plt.title('Booking Cancellation Rate')
plt.show()
```




- Visualization: Pie Chart
  - A pie chart is used to display proportions ***
- **Insights**:
  - If the cancellation rate is 29.9% i.e. nearly 30%.
  - To reduce the Bookig Cancellation, hotels should consider stricter cancellation policies such as penalty for cancelling the last moment or incentives for non-cancelled bookings. ***

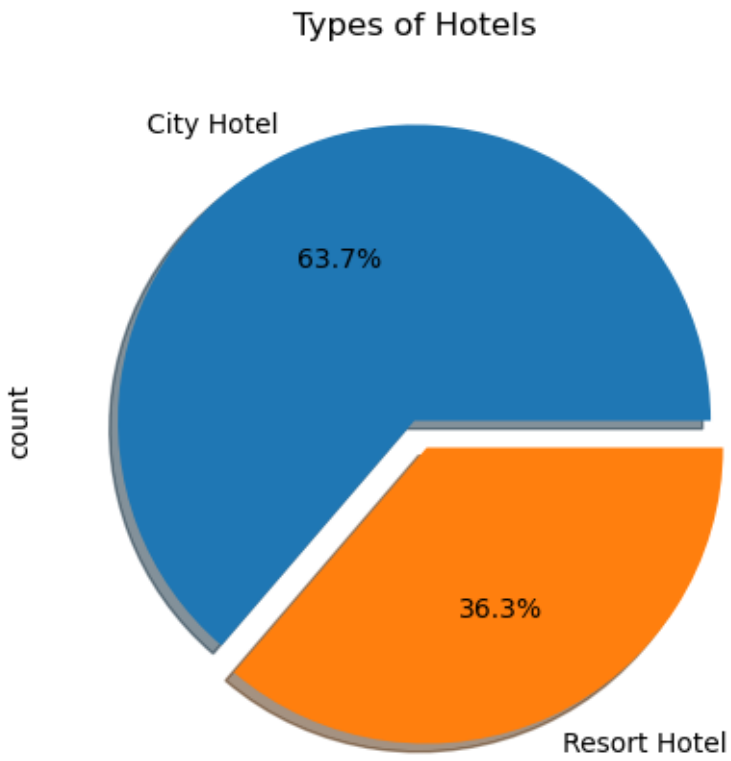
### 2.8.7 9. Which type of hotels are more prefered?

[19]:
```
plt.figure(figsize=(10, 5))
explode = [0, 0.1]  # Exploding only the second category (Resort Hotel)

df['hotel'].value_counts().plot(
    kind='pie',
    autopct='%1.1f%%',
    labels=['City Hotel', 'Resort Hotel'],
    explode=explode,
    shadow=True
)

plt.title('Types of Hotels')
```

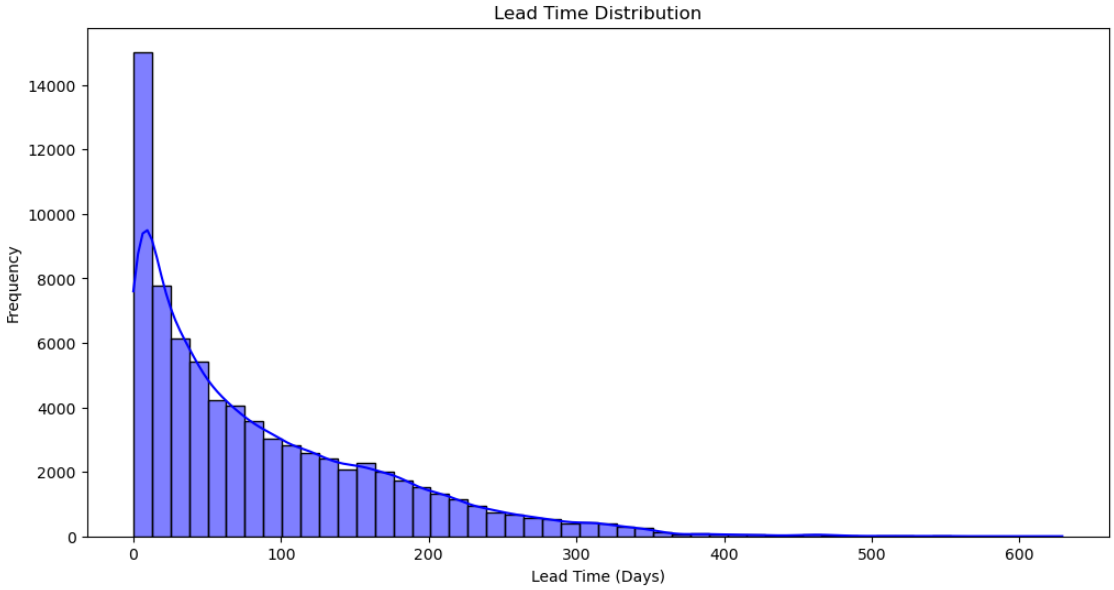
```
plt.show()
```




- Visualization: Pie Chart
  - Reason: A pie chart is used to display proportions, helping us understand cancellation rates. ***
- **Insights**:
  - The bookings of City Hotels are more than Resort Hotels. ***

### 2.9 10. What is the distribution of lead time (time before arrival)?

[22]:
```
plt.figure(figsize=(12, 6))
sns.histplot(df['lead_time'], bins=50, kde=True,color='blue')
plt.title('Lead Time Distribution')
plt.xlabel('Lead Time (Days)')
plt.ylabel('Frequency')
plt.show()
```

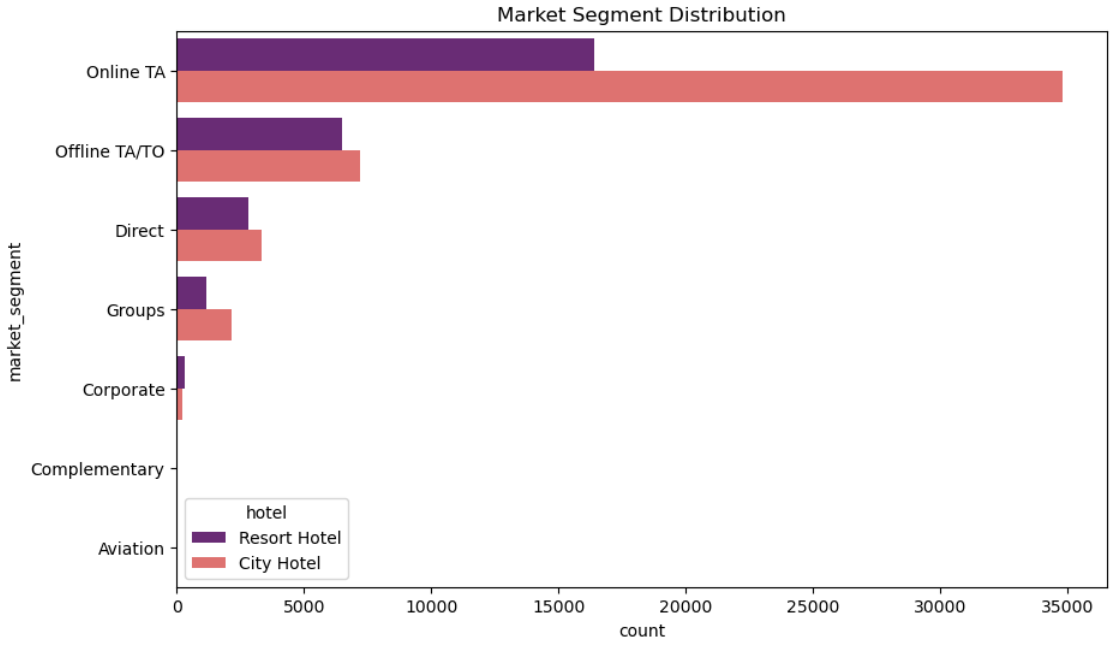



- Visualization: Histogram
  - Reason: A histogram is ideal for showing the frequency distribution of continuous numerical variables. ***
- **Insight**:
  - Understand booking behavior. Most of bookings are made within 50 days, hotels can `create last-minute deals` to attract more customers.

---

### 2.9.1 11. Which market segment brings in the most guests?

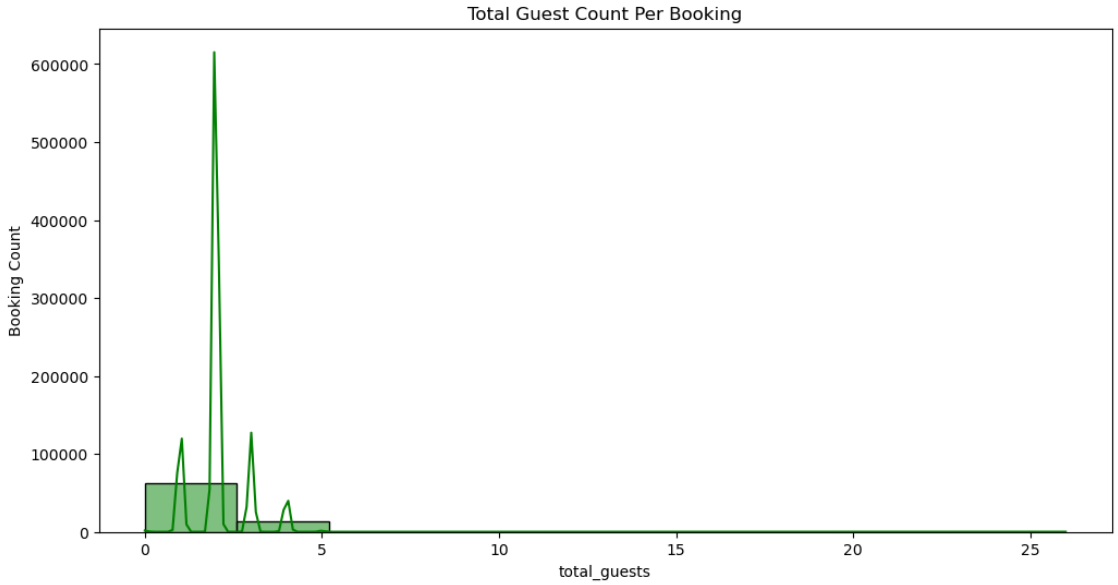
[45]:
```
plt.figure(figsize=(10, 6))
sns.countplot(y='market_segment', data=df, hue='hotel', palette='magma',␣
↪order=df['market_segment'].value_counts().index)
plt.title('Market Segment Distribution')
plt.show()
```




- Visualization: Bar Chart
  - Reason: A bar chart effectively represents categorical data for easy comparison. ***
- **Insight**:
  - The "Online TA" (Travel Agent) segment appears to be the most dominant for both types of hotels, with the highest count of bookings. This suggests that most customers prefer booking through online travel agents.
  - The City Hotel seems to have a higher number of bookings across all market segments compared to the Resort Hotel. This could indicate that the City Hotel is more popular or has a larger capacity.
  - After "Online TA," the "Offline TA/TO" (Travel Agent/Tour Operator) segment is the next most popular, followed by "Direct" bookings. This indicates that traditional travel agents and direct bookings still hold significant market share. ***

### 2.9.2 12. What is the distribution of total guests per booking?

[46]:
```
plt.figure(figsize=(12, 6))
df['total_guests'] = df['adults'] + df['children'] + df['babies']
sns.histplot(df['total_guests'], bins=10, kde=True, color='green')
plt.title('Total Guest Count Per Booking')
plt.ylabel('Booking Count')
plt.show()
```

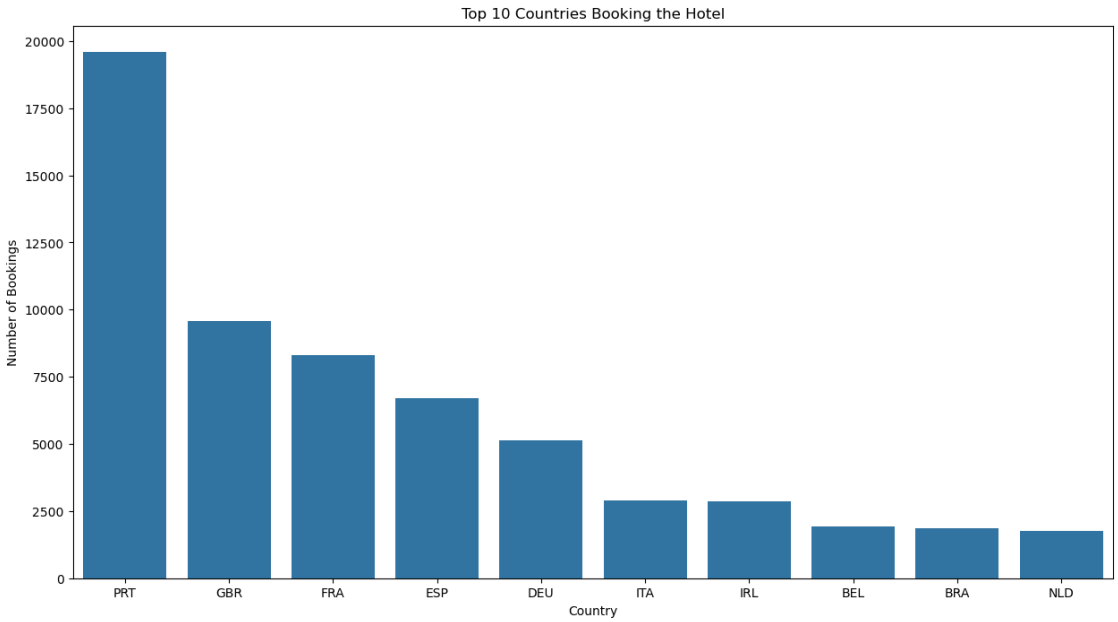



- Histogram with KDE:
  - A histogram is used to show the frequency distribution of a continuous variable (total guests per booking).
  - The KDE curve is added to provide a smooth estimate of the distribution, making it easier to identify patterns and skewness ***

- **Insights**:
  - Guest Count Distribution:
    * The histogram shows the distribution of the total number of guests per booking, which is calculated by summing adults, children, and babies.
    * The majority of bookings have a total guest count between 1 and 4, indicating that most bookings are for small groups or families.
  - The highest frequency of bookings is for 2 guests, which is common for couples or single travelers with one child. This suggests that solo travelers or small families are the primary customer base.
  - ookings with a higher number of guests (e.g., more than 4) are less frequent. This could indicate that larger groups or extended families are less common or that the hotel may not be as accommodating for large groups. ***

### 2.9.3 13. Which are the top 10 countries making hotel bookings?

[85]:
```
plt.figure(figsize=(15, 8))
top_countries = df['country'].value_counts().head(10)
sns.barplot(x=top_countries.index, y=top_countries.values)
plt.title('Top 10 Countries Booking the Hotel')
plt.xlabel('Country')
plt.ylabel('Number of Bookings')
plt.show()
```

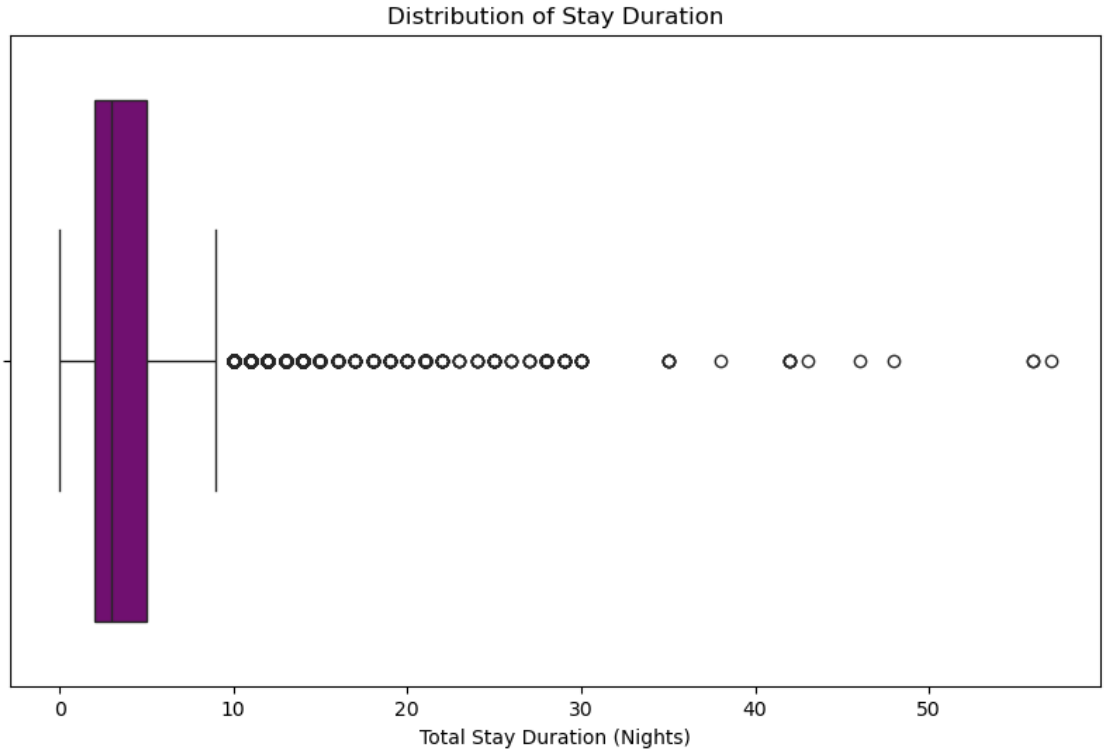



Visualization: Bar Chart/ Count Plot: * Count Plot: A count plot is used to show the frequency of each category (room type in this case) in a dataset. It is effective for visualizing the distribution of categorical data and identifying the most and least common categories. Insights**: * Portugal (PRT) has the highest number of hotel bookings, with approximately 19,000 bookings. This indicates a strong demand for hotel accommodations in Portugal. * After Portugal, the United Kingdom (GBR) comes in second with around 10,000 bookings, followed by France (FRA) with 8,000, Spain (ESP) with 6,000, and Germany (DEU) with 5,000 bookings. These countries represent major markets for the hotel industry.

---

### 2.9.4 14. What is the distribution of stay duration?

[49]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x=df['stays_in_week_nights'] +␣
 ↪df['stays_in_weekend_nights'], color='purple')
plt.title('Distribution of Stay Duration')
plt.xlabel('Total Stay Duration (Nights)')
plt.show()
```

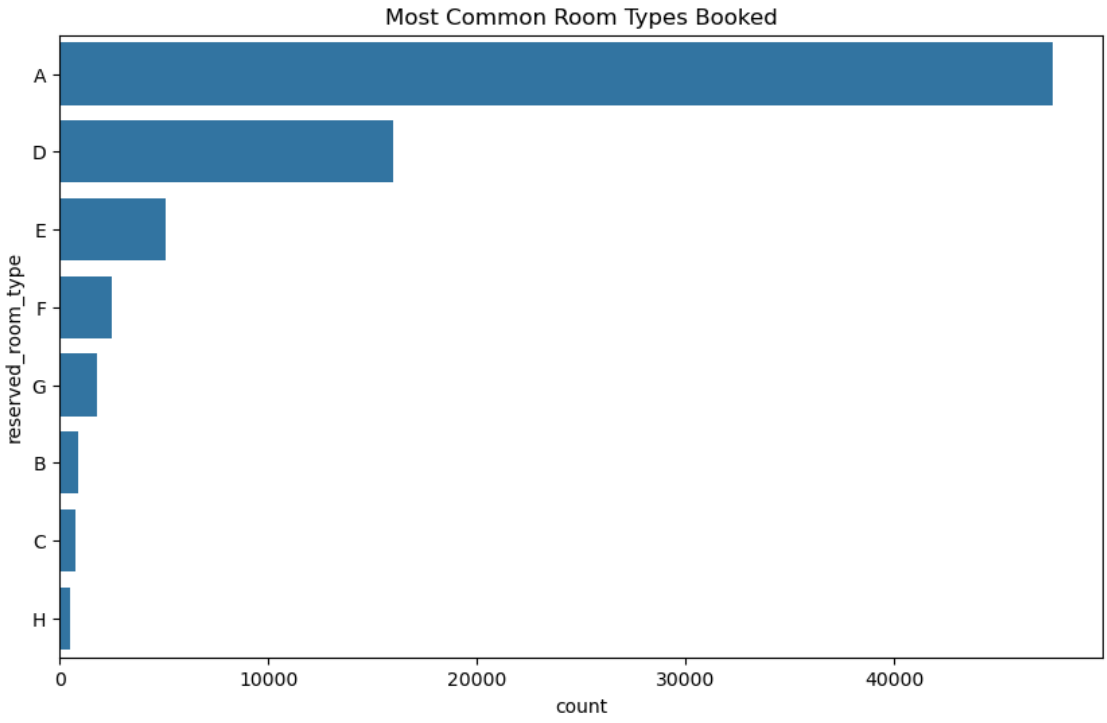



- Visualizations Used:
  - A boxplot is used in this context because it effectively summarizes the distribution of a continuous variable (stay duration) by showing the central tendency, spread, and potential outliers. It provides a clear visual representation of the data's variability and highlights any anomalies that may need further investigation. ***
- **Insights from the Graph**:
  - The boxplot shows that the median stay duration is around 3 nights.
  - The box represents the interquartile range (IQR), which is the middle 50% of the data. The IQR for stay duration appears to be between approximately 2 and 5 nights.
  - There are several outliers in the data, as indicated by the circles beyond the whiskers. These outliers represent unusually long stay durations, with some extending up to 50 nights.
  - The whiskers extend from the box to the smallest and largest values within 1.5 times the IQR from the quartiles, indicating the spread of the majority of the data. The data is skewed to the right, as there are more outliers on the higher end of the stay duration. ***

### 2.9.5 15. What is the most common room type booked?

[27]:
```
plt.figure(figsize=(10, 6))
sns.countplot(y='reserved_room_type', data=df,
              order=df['reserved_room_type'].value_counts().index)
plt.title('Most Common Room Types Booked')
```

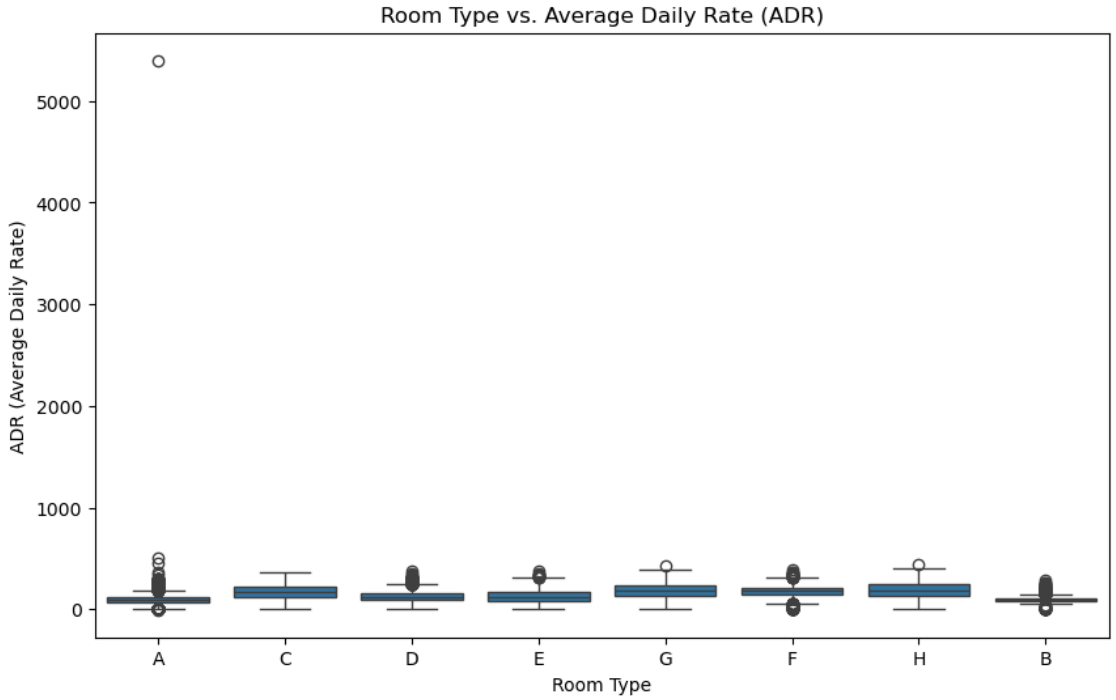
```
plt.show()
```




Visualization: Bar Chart * Reason: A bar chart makes it easy to see which room type is most popular. *** * **Insights**: * Room type “A” is the most frequently booked, indicating that it is the most popular choice among guests. This could be due to factors such as price, availability, or amenities. * Room types “D” and “E” also show significant booking counts, suggesting they are also preferred by guests, though not as much as type “A”. * Room types like “L”, “P”, and “F” have much lower booking counts, indicating they are less popular or possibly more specialized or expensive. ***

### 2.9.6 16. How do prices vary across different room types?

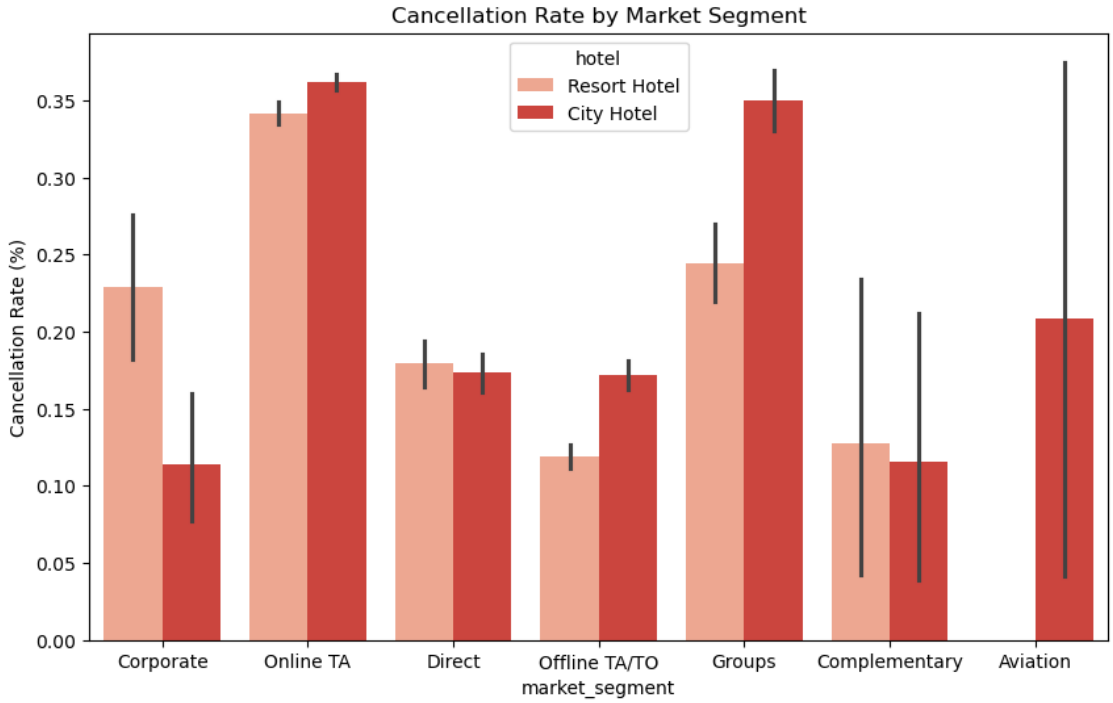
[28]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='reserved_room_type', y='adr', data=df)
plt.title('Room Type vs. Average Daily Rate (ADR)')
plt.xlabel('Room Type')
plt.ylabel('ADR (Average Daily Rate)')
plt.show()
```




Visualization: Box Plot * The boxplot is used in this context because it effectively shows the distribution of ADR values for each room type. It includes the median, quartiles, and potential outliers, providing a comprehensive view of the central tendency, spread, and variability of prices across different room types. Insights**: * Room Type A: This room type has a wide range of ADR (Average Daily Rate) values. This room type has a higher average price compared to the others. Some prices for Room Type A are much higher than the rest, which means there are expensive rooms in this category. * Room Types B to H: These room types generally have lower average prices compared to Room Type A. There isn't as much variation in prices, so most rooms in these categories are priced similarly.** *

### 2.10 17. Do online bookings have a higher cancellation rate?

[53]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x='market_segment', y='is_canceled', data=df, hue=␣
↪'hotel',palette='Reds')
plt.title('Cancellation Rate by Market Segment')
plt.ylabel('Cancellation Rate (%)')
plt.show()
```

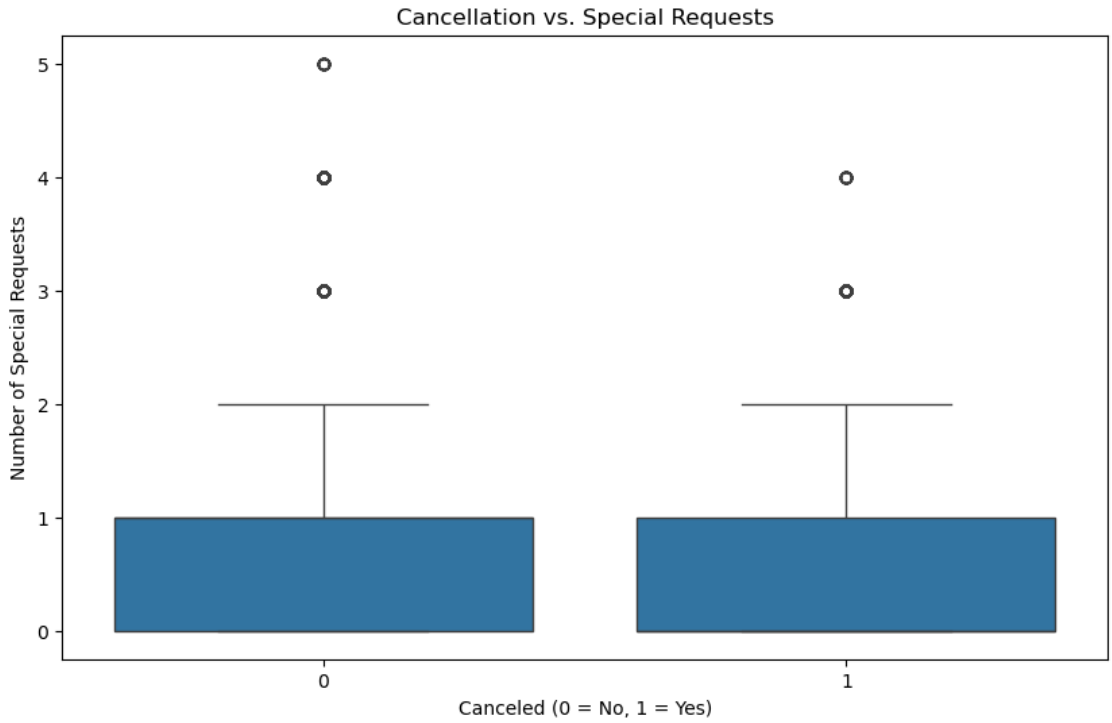



- Visualization: Bar Chart
  - Reason: A bar chart effectively compares categorical variables like booking type and cancellation.
  - A bar plot is used here because it effectively compares the cancellation rates across different market segments. It makes it easy to identify which segments have higher or lower cancellation rates. rate. ***
- **Insights**:
  - Complimentary bookings have the lowest cancellation rate, around 5-10%.
  - Bookings made through offline travel agencies or tour operators have a lower cancellation rate, about 10-15%, making them more reliable than online bookings.
  - Bookings made through online travel agencies have the highest cancellation rate, around 35%. This means that more than one-third of bookings from this source get canceled.
  - Group bookings have a relatively high cancellation rate, approximately 25%, indicating that these bookings are also frequently canceled.
  - Both corporate and direct bookings have a moderate cancellation rate, ranging between 15-20%. This suggests that these bookings are somewhat reliable but still have a fair amount of cancellations.

### 2.10.1 18. What is the impact of special requests on cancellations?

[58]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='is_canceled', y='total_of_special_requests', data=df)
plt.title('Cancellation vs. Special Requests')
plt.xlabel('Canceled (0 = No, 1 = Yes)')
```

```
plt.ylabel('Number of Special Requests')
plt.show()
```

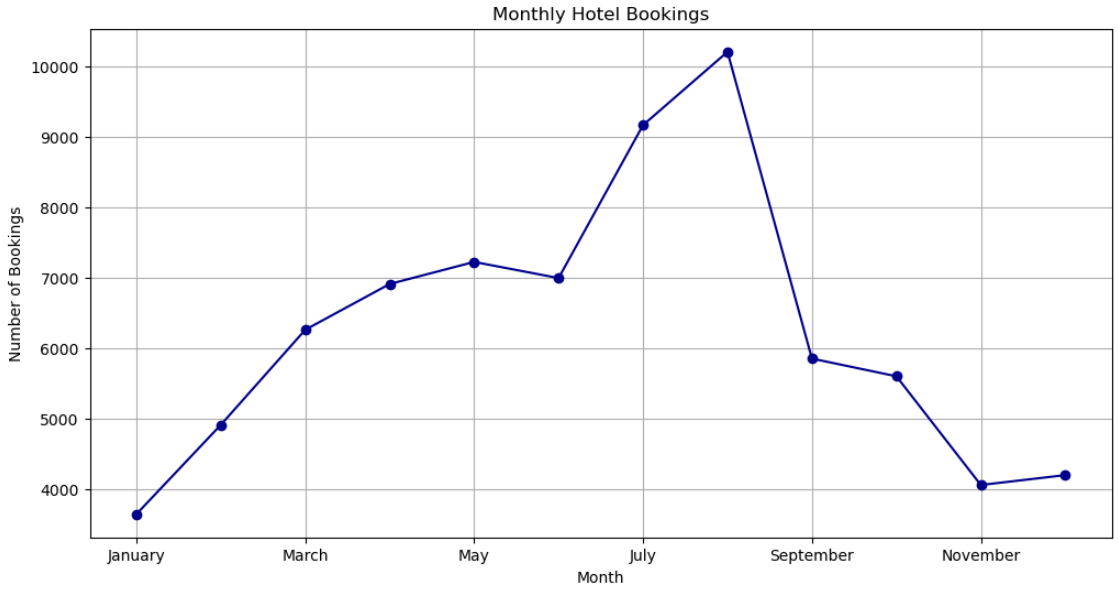



- Visualization: Box Plot
  - Reason: A boxplot is used here because it effectively summarizes the distribution of the number of special requests for both canceled and non-canceled bookings. It shows the central tendency (median), spread (IQR), and potential outliers in a clear and concise manner. ***
- **Insights**:
  - Both canceled and non-canceled bookings have a similar average number of special requests. This means that whether a booking gets canceled or not doesn't seem to depend much on the number of special requests.
  - Most bookings, whether canceled or not, have only a few special requests. Very few bookings have a large number of requests.
  - Outliers with a high number of special requests are rare, but they exist in both canceled and non-canceled categories ***

### 2.10.2 19. What are the busiest months for hotel bookings?

[60]:
```
monthly_bookings = df.groupby('arrival_date_month')['hotel'].count().reindex([
    'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',␣
 ↪'September', 'October', 'November', 'December'
])
```

```
plt.figure(figsize=(12, 6))
monthly_bookings.plot(kind='line', marker='o', color='darkblue')
plt.title('Monthly Hotel Bookings')
plt.xlabel('Month')
plt.ylabel('Number of Bookings')
plt.grid()
plt.show()
```




- Visualization: Line Chart
  - Reason: A line chart is used because it effectively shows trends over time, making it easy to identify patterns and changes in the number of bookings across different months

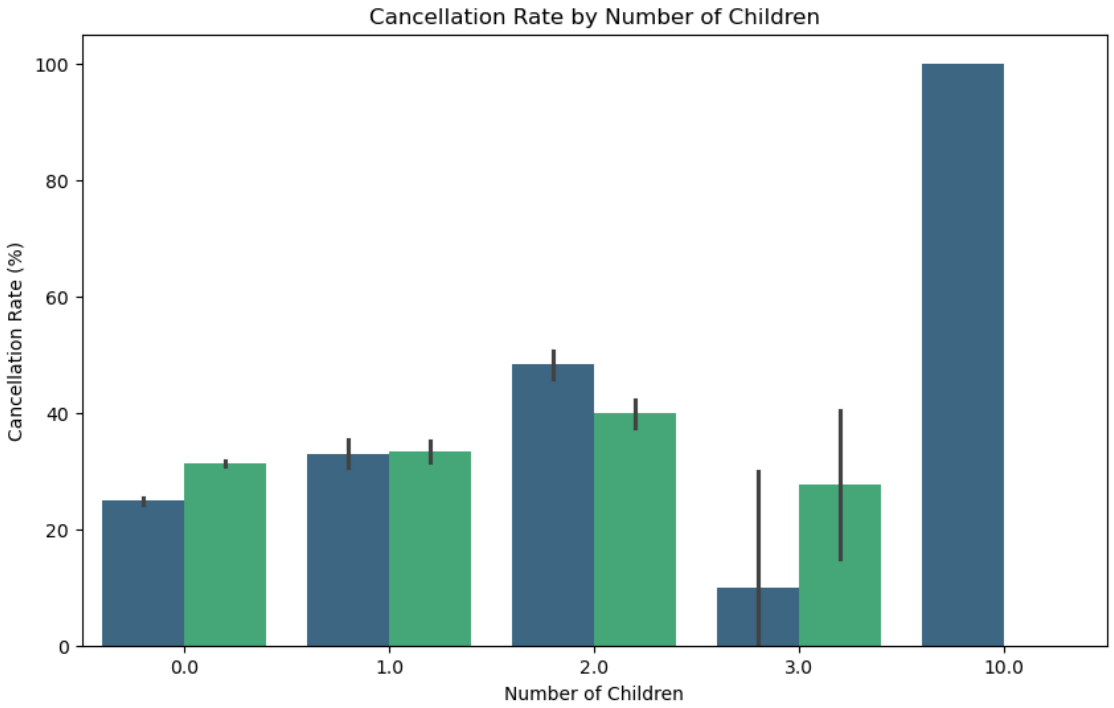
---

- **Insights**:
  - July and August are the busiest months
  - The number of hotel bookings steadily increases from January to August, with the highest number of bookings in August. This indicates that August is the peak season for hotel bookings.
  - There is a significant drop in bookings after August, reaching the lowest point in November. This suggests that November is the off-peak season. ***

### 2.10.3 20. How does the number of children affect cancellation rates?

[63]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x='children', y='is_canceled', data=df, hue='hotel', legend=False,␣
 ↪palette='viridis', estimator=lambda x: np.mean(x) * 100)
plt.title('Cancellation Rate by Number of Children')
plt.xlabel('Number of Children')
```

```
plt.ylabel('Cancellation Rate (%)')
plt.show()
```



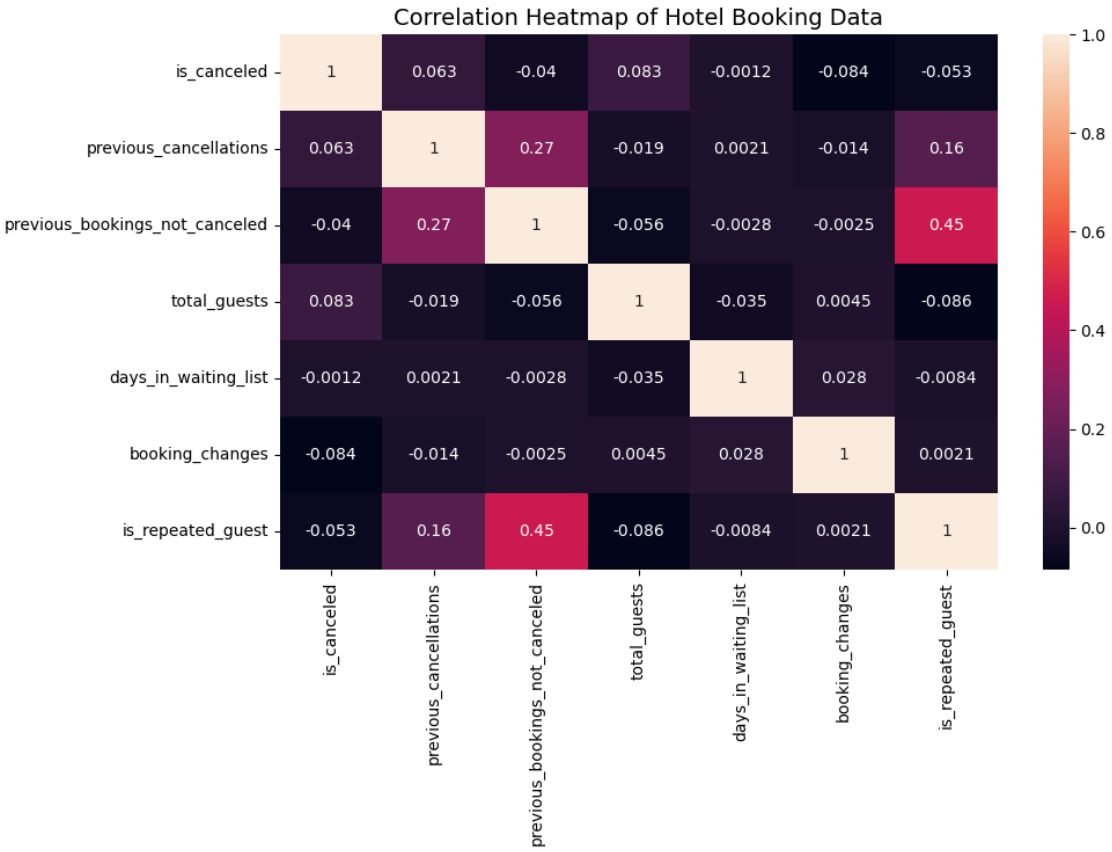

- Visualization: Bar Chart
  - A bar plot is used because it effectively compares the cancellation rates across different categories (number of children). It provides a clear visual representation of the data, making it easy to see how cancellation rates vary with the number of children. ***
- **Insights from the Graph**:
  - For bookings with no children, the cancellation rate is between 20-30%. This suggests that a moderate number of bookings without children are canceled.
  - The cancellation rate for bookings with one and two children is higher than with no childen, around 0-40% and 40-50% respectively.
  - nterestingly, bookings with three children have a lower cancellation rate of around 10-20%. This suggests that families with three children tend to cancel less often.
  - The cancellation rate spikes to 100% for bookings with ten children, indicating that all such bookings are canceled. * ***

### 2.10.4 21.Correlation Heatmap

[66]:
```
plt.figure(figsize=(10, 6))
a = df[['is_canceled', 'previous_cancellations',␣
 ↪'previous_bookings_not_canceled', 'total_guests',
       'days_in_waiting_list', 'booking_changes', 'is_repeated_guest' ]].corr()
```

```
# corrilation heatmap
sns.heatmap(a, annot = True)
# Adding a title
plt.title('Correlation Heatmap of Hotel Booking Data', fontsize=14)

# Displaying the heatmap
plt.show()
```



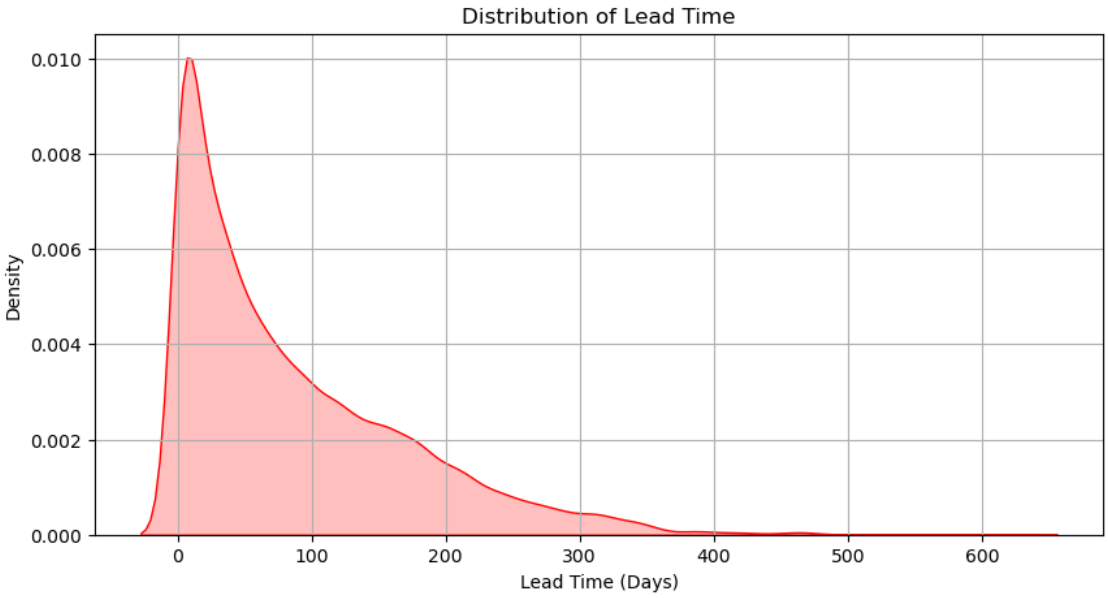

- Visualization:
  - A correlation heatmap is used here because it provides a clear and concise way to visualize the relationships between multiple variables simultaneously. It helps quickly identify which variables are correlated and the strength of those correlations. The color scale aids in interpreting the correlations easily. _____
- **Insights**:
  - There's a moderate positive correlation (0.45) between "previous bookings not canceled" and "is repeated guest." This means that guests who haven't canceled previous bookings are more likely to return.
  - The variable "is_canceled" has weak correlations with other factors. The highest correlation is with "previous cancellations" (0.063) and "total guests" (0.083), indicating that cancellations are not strongly influenced by these factors.

- Most of the variables in the dataset show very weak or negligible correlations with each other, suggesting that they are relatively independent.
- Since most variables are independent, businesses can address each factor separately to optimize operations and improve customer satisfaction. ***

### 2.10.5 22. What is the distribution of lead time? (KDE Plot)

KDE (Kernel Density Estimate) plots show the probability distribution of numerical data

[74]:
```
plt.figure(figsize=(10, 5))
sns.kdeplot(df['lead_time'], fill=True, color='red')
plt.title("Distribution of Lead Time")
plt.xlabel("Lead Time (Days)")
plt.ylabel("Density")
plt.grid(True)
plt.show()
```




- Visualizations:
  - A Kernel Density Estimate (KDE) plot is used because it provides a smooth curve that represents the distribution of the data. ***
- **Insight**:
  - Most bookings have short lead times. This means customers typically book their stays not too far in advance.
  - There's a long tail on the right side, indicating some bookings are made well in advance, but these are less common.
  - The highest peak is around a low number of days, suggesting that the majority of bookings are made shortly before the stay date.

### 2.10.6 23. Comparison of ADR for different types of bookings? (Subplots: Direct vs Online Travel Agent)

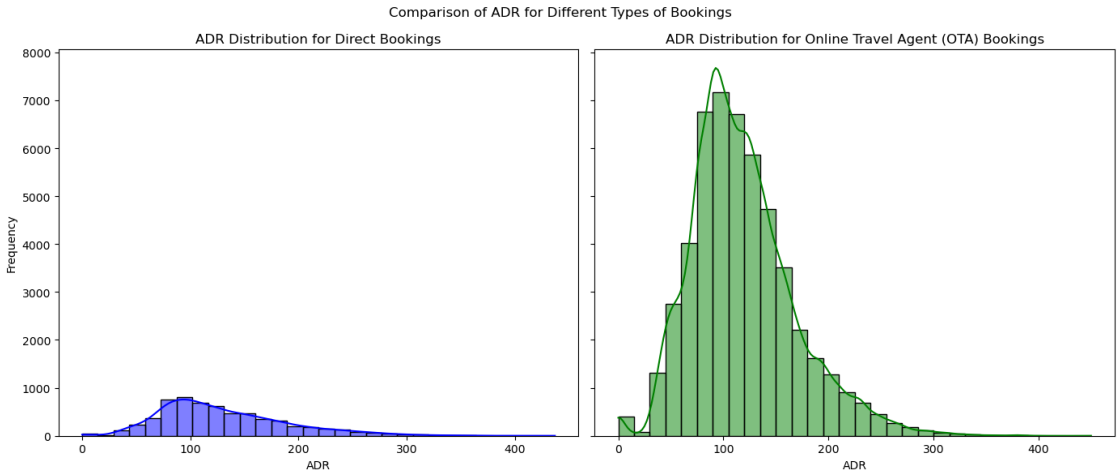
[36]:
```
# Create subplots
fig, axes = plt.subplots(1, 2, figsize=(14, 6), sharey=True)

# Direct Booking ADR Distribution
sns.histplot(df[df['market_segment'] == 'Direct']['adr'], bins=30, kde=True,␣
 ↪color='blue', ax=axes[0])
axes[0].set_title("ADR Distribution for Direct Bookings")
axes[0].set_xlabel("ADR")
axes[0].set_ylabel("Frequency")

# Online Travel Agent (OTA) ADR Distribution
sns.histplot(df[df['market_segment'] == 'Online TA']['adr'], bins=30, kde=True,␣
 ↪color='green', ax=axes[1])
axes[1].set_title("ADR Distribution for Online Travel Agent (OTA) Bookings")
axes[1].set_xlabel("ADR")
axes[1].set_ylabel("")

# Overall title
plt.suptitle("Comparison of ADR for Different Types of Bookings")

plt.tight_layout()
plt.show()
```




- Visualization:
  - Histograms are effective for visualizing the distribution of a single numerical variable. They allow you to see how frequently different values occur. ***

- **Insights**:

1. First Plot: Direct Bookings
    - The average daily rate (ADR) for direct bookings is mostly concentrated between 0 and 100.
    - There is a peak around 50, indicating that most direct bookings have an ADR close to this value.
    - There are some outliers with ADR values extending up to 400, but these are less common. ***
2. Online Travel Agent (OTA) Bookings:
    - The ADR values for OTA bookings are more spread out compared to direct bookings.
    - The peak is around 100, and many bookings have ADR values between 100 and 200.
    - Similar to direct bookings, there are some outliers with high ADR values, but they are more spread out. ***

## 2.11 Key Insights

1. Cancellation Rates by Market Segment:
    - Online Travel Agencies (OTAs) have highest cancellation rate at around 35%.
    - Followed by Groups(25%), Corporate and Direct Bookings (15-20%), ect
2. Cancellation Rates by Room Type:
    - Room Type A has higher average daily rate (ADR) and several outliers.
    - Room Types B to H has lower ADRs with narrower ranges and fewer outliers.
3. Special Requests and Cancellations:
    - Similar average number of special requests for both canceled and non-canceled bookings.
    - Most bookings have a few requests, with rare high-request bookings.
4. Monthly Booking Trends:
    - Highest bookings in August. July and August being the busiest months
5. Lead Time and Cancellations:
    - Majority of bookings have short lead times(50 days).
    - Long tail indicating some bookings made well in advance.
6. Customer Types and ADR:
    - Transient Customers have Highest ADR around 115 followed by Contract Customers having Lower ADR around 95.

## 2.12 `Business Solutions to Grow the Business:`

1. Targeted Marketing and Promotions:
    - Introduce flexible booking policies to reduce cancellations.
    - Offer group discounts and special packages to encourage bookings.
    - Develop targeted marketing campaigns for July and August to capitalize on high demand.
    - Offer special deals and discounts for bookings in January and November to attract bookings during off-peak months.
2. Customer Retention Strategies:

- Create loyalty programs targeting repeated guests who haven't canceled previous bookings.
- Streamline the process for handling common special requests to ensure customer satisfaction.

3. Pricing Strategies:
   - Implement dynamic pricing strategies to adjust ADR based on demand and lead time.
   - Tailor pricing strategies for different customer types (transient, contract, group) to maximize revenue.
4. Booking Policies:
   - Encourage non-refundable deposit options to reduce cancellations.
   - Offer flexible booking options for direct and OTA bookings to attract more customers.
   - Use automated email and messaging systems to confirm bookings, send reminders, and provide pre-arrival information.
5. Operational Improvements:
   - Optimize staffing and resources for peak seasons to handle increased demand efficiently.
   - Continuously collect and analyze guest feedback to identify areas for improvement and enhance the overall booking experience.

## 3 Thank You!

[ ]: