Project Name - Hospitality By the Numbers: Hotel Booking Study

Project Type - EDA

Contribution - Individual

Member - Nishant Sharma

Project Summary -

The aim of this project is to explore and analyze a comprehensive hotel booking dataset, comprising information from both a city hotel and a resort hotel. The dataset includes diverse variables such as booking date, length of stay, the count of adults, children, and babies, and the availability of parking spaces, among other relevant factors.

1. The primary focus of this analysis is to answer crucial questions that can provide valuable insights to both hotel management and potential guests. The main research areas are as follows:
2. Optimal Booking Time: Determine the best time of the year to book a hotel room to secure the most competitive rates. Analyzing historical booking patterns can unveil seasonal trends, allowing businesses to optimize pricing strategies.
3. Length of Stay Optimization: Investigate the relationship between the length of stay and the daily room rate. Identify patterns indicating optimal durations for the best value-for-money options.
4. Special Request Predictions: Develop a predictive model to forecast whether a hotel is likely to receive an unusually high number of special requests based on specific factors like booking time, hotel type, and guest demographics. This information can assist hotels in enhancing their services and resource planning.

The dataset's richness enables a comprehensive exploration of factors influencing hotel bookings and can contribute significantly to decision-making processes within the hospitality industry. Employing advanced data analysis and machine learning techniques, this project seeks to unveil patterns, correlations, and potential predictors that drive hotel booking behaviors.

By delving into this dataset, we aspire to deliver actionable insights that empower hotels to optimize their offerings, enhance guest experiences, and improve overall business efficiency. Additionally, travelers can benefit from understanding the most opportune times to book, leading to enhanced satisfaction and better-informed travel planning.

GitHub Link -

Provide your GitHub Link here.

https://github.com/Ast0n1sh/hotelbooking121

Problem Statement

The hospitality industry faces challenges in understanding guest behavior and optimizing hotel operations. To address these challenges, we aim to analyze a comprehensive hotel booking dataset comprising information from both city and resort hotels. This dataset includes variables such as booking date, length of stay, the number of adults, children, and babies, and the availability of parking spaces, among others.

The primary goal of this project is to answer essential questions that can provide valuable insights to hotel management and potential guests. The specific problems to be addressed are as follows:

- Optimal Booking Time
- Length of Stay Optimization
- Special Request Predictions
- Additional Ideas

To tackle these challenges, we will employ advanced data analysis and machine learning techniques. The project's outcomes will provide valuable insights into factors influencing hotel bookings, enabling data-driven decision-making for hotels and empowering travelers to make more informed choices while planning their stays.

The success of this analysis will contribute significantly to the growth and efficiency of the hospitality industry by facilitating better resource allocation, enhancing guest experiences, and enabling hotels to offer competitive services.

Define Your Business Objective?

The primary business objective of this project is to leverage the hotel booking dataset to enhance the efficiency and competitiveness of the hospitality industry. By conducting a comprehensive analysis of booking information from city and resort hotels, the project aims to achieve the following key objectives:

1. Data-Driven Insights: Utilize data analysis and machine learning techniques to derive valuable insights and patterns from the dataset. These insights will provide actionable information to hotel management, allowing them to make informed decisions and optimize their operations.
2. Pricing Strategy Optimization: Identify the optimal time of the year for guests to book hotel rooms and recommend pricing strategies based on historical booking patterns. This objective aims to attract more customers during specific seasons and enhance revenue generation.
3. Enhanced Guest Experiences: Develop a predictive model to anticipate special requests and personalized requirements from guests. By doing so, hotels can provide tailored services, improving overall guest satisfaction and loyalty.
4. Competitive Advantage: Provide hotels with a competitive advantage by enabling them to offer attractive packages based on the relationship between length of stay and daily room rates. This objective aims to attract more guests and increase market share.
5. Traveler Empowerment: Empower travelers with insights into the best times to book hotels, leading to better-informed travel planning and enhanced satisfaction during their stays.

By achieving these business objectives, this project will contribute to the growth and success of the hospitality industry by fostering data-driven decision-making, improved resource allocation, and elevated guest experiences. Ultimately, the project aims to create a win-win situation for both hotels and travelers, optimizing business operations while offering enhanced value to customers.

General Guidelines : -

1. Well-structured, formatted, and commented code is required.
2. Exception Handling, Production Grade Code & Deployment Ready Code will be a plus. Those students will be awarded some additional credits.

   The additional credits will have advantages over other students during Star Student selection.

   ```
   [ Note: - Deployment Ready Code is defined as, the whole .ipynb notebook should be executable in one go
             without a single error logged. ]
   ```

3. Each and every logic should have proper comments.
4. You may add as many number of charts you want. Make Sure for each and every chart the following format should be answered.

```
# Chart visualization code
```

- Why did you pick the specific chart?
- What is/are the insight(s) found from the chart?
- Will the gained insights help creating a positive business impact? Are there any insights that lead to negative growth? Justify with specific reason.

5. You have to create at least 20 logical & meaningful charts having important insights.

[Hints : - Do the Vizualization in a structured way while following "UBM" Rule.

U - Univariate Analysis,

B - Bivariate Analysis (Numerical - Categorical, Numerical - Numerical, Categorical - Categorical)

M - Multivariate Analysis]

Let's Begin !

1. Know Your Data

Import Libraries

```
# Import Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Setting to display all columns of the dataframe in outputs
pd.set_option('display.max_columns', None)
```

Dataset Loading

```
# Load Dataset
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Dataset First View

```
# Dataset First Look
dataset = pd.read_csv('/content/drive/MyDrive/Project X Raw Data/Hotel Bookings.csv')
display(dataset.head())
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arr
0	Resort Hotel	0	342	2015	July	27	
1	Resort Hotel	0	737	2015	July	27	
2	Resort Hotel	0	7	2015	July	27	
3	Resort Hotel	0	13	2015	July	27	
4	Resort Hotel	0	14	2015	July	27	

Dataset Rows & Columns count

```
# Dataset Rows & Columns count
# Getting the shape of the dataset
dataset_shape = dataset.shape
dataset_shape
```

```
(119390, 32)
```

The Data Set has 119390 Rows and 32 Columns

Dataset Information

```
# Dataset Info
# Getting the information about the dataset
dataset_info = dataset.info()
dataset_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
```

Duplicate Values

```
# Dataset Duplicate Value Count
duplicate_count = dataset.duplicated().sum()
duplicate_count
```

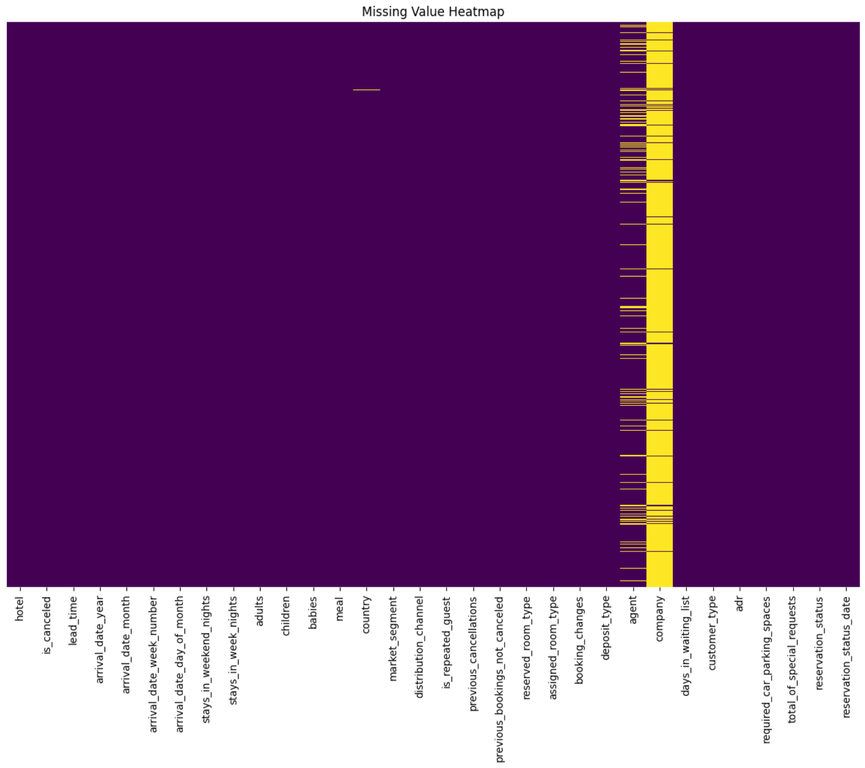
```
31994
```

Missing Values/Null Values

```
# Missing Values/Null Values Count
missing_values = dataset.isnull().sum()
missing_values
```

```
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                              488
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                              16340
company                           112593
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
```

```
# Visualizing the missing values
plt.figure(figsize=(15, 10))
sns.heatmap(dataset.isnull(), cbar=False, cmap='viridis', yticklabels=False)
plt.title('Missing Value Heatmap')
plt.show()
```



What did you know about your dataset?

From our analysis till now, here are the observations:

- The dataset comprises 119,390 rows and 32 columns.
- There are several columns with missing values, with the 'company' column having the most missing values.
- The dataset contains 31,994 duplicate rows.
- A mix of data types is present, including integers, floats, and objects (strings).
- Some columns like 'agent' and 'company' have a significant number of missing values, as visualized in the heatmap.

2. Understanding Your Variables

```
# Dataset Columns
dataset_columns = dataset.columns
dataset_columns
```

```
Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',
       'arrival_date_month', 'arrival_date_week_number',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'adults', 'children', 'babies', 'meal',
       'country', 'market_segment', 'distribution_channel',
       'is_repeated_guest', 'previous_cancellations',
       'previous_bookings_not_canceled', 'reserved_room_type',
       'assigned_room_type', 'booking_changes', 'deposit_type', 'agent',
       'company', 'days_in_waiting_list', 'customer_type', 'adr',
       'required_car_parking_spaces', 'total_of_special_requests',
       'reservation_status', 'reservation_status_date'],
      dtype='object')
```

```
# Dataset Describe
dataset_description = dataset.describe()
dataset_description
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arr
count	119390.000000	119390.000000	119390.000000	119390.000000	
mean	0.370416	104.011416	2016.156554	27.165173	
std	0.482918	106.863097	0.707476	13.605138	
min	0.000000	0.000000	2015.000000	1.000000	
25%	0.000000	18.000000	2016.000000	16.000000	
50%	0.000000	69.000000	2016.000000	28.000000	
75%	1.000000	160.000000	2017.000000	38.000000	
max	1.000000	737.000000	2017.000000	53.000000	

Variables Description

1. hotel: Type of hotel (Resort Hotel or City Hotel).
2. is_canceled: Indicates if the booking was canceled (1 if canceled, 0 otherwise).
3. lead_time: Number of days between the booking date and the arrival date.
4. arrival_date_year: Year of the arrival date.
5. arrival_date_month: Month of the arrival date.
6. arrival_date_week_number: Week number of the year for the arrival date.
7. arrival_date_day_of_month: Day of the month of the arrival date.
8. stays_in_weekend_nights: Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay.
9. stays_in_week_nights: Number of week nights (Monday to Friday) the guest stayed or booked to stay.
10. adults: Number of adults.
11. children: Number of children.
12. babies: Number of babies.
13. meal: Type of meal booked.
14. country: Country of origin of the booking.
15. market_segment: Market segment designation.
16. distribution_channel: Booking distribution channel.
17. is_repeated_guest: Indicates if the booking name is a repeated guest (1 if true, 0 otherwise).
18. previous_cancellations: Number of previous bookings that were canceled by the customer.
19. previous_bookings_not_canceled: Number of previous bookings not canceled by the customer.
20. reserved_room_type: Code of the room type reserved.
21. assigned_room_type: Code for the type of room assigned to the booking.
22. booking_changes: Number of changes made to the booking.
23. deposit_type: Type of deposit made.
24. agent: ID of the travel agency that made the booking.
25. company: ID of the company/entity that made the booking or responsible for paying the booking.
26. days_in_waiting_list: Number of days the booking was in the waiting list before it was confirmed.
27. customer_type: Type of booking (e.g., Contract, Group, Transient).
28. adr: Average Daily Rate.
29. required_car_parking_spaces: Number of parking spaces required by the customer.
30. total_of_special_requests: Number of special requests made by the customer.
31. reservation_status: Last reservation status (e.g., Canceled, Check-Out, No-Show).
32. reservation_status_date: Date when the last status was set.

Check Unique Values for each variable.

```
# Check Unique Values for each variable
unique_values = dataset.nunique()
```

```
unique_values

     hotel                                2
     is_canceled                          2
     lead_time                          479
     arrival_date_year                    3
     arrival_date_month                  12
     arrival_date_week_number            53
     arrival_date_day_of_month           31
     stays_in_weekend_nights             17
     stays_in_week_nights                35
     adults                              14
     children                             5
     babies                               5
     meal                                 5
     country                            177
     market_segment                       8
     distribution_channel                 5
     is_repeated_guest                    2
     previous_cancellations              15
     previous_bookings_not_canceled      73
     reserved_room_type                  10
     assigned_room_type                  12
     booking_changes                     21
     deposit_type                         3
     agent                              333
     company                            352
     days_in_waiting_list               128
     customer_type                        4
     adr                               8879
     required_car_parking_spaces          5
     total_of_special_requests            6
     reservation_status                   3
     reservation_status_date            926
     dtype: int64
```

3. ***Data Wrangling***

Data Wrangling Code

```
# Write your code to make your dataset analysis ready.
# 1. Removing duplicate rows
dataset_cleaned = dataset.drop_duplicates()

# 2. Handling missing values
# For 'company' and 'agent', replacing NaN values with a placeholder '0' (indicating no company or agent)
dataset_cleaned['company'].fillna(0, inplace=True)
dataset_cleaned['agent'].fillna(0, inplace=True)

# For 'country' and 'children', replacing NaN values with the mode of the respective columns
dataset_cleaned['country'].fillna(dataset_cleaned['country'].mode()[0], inplace=True)
dataset_cleaned['children'].fillna(dataset_cleaned['children'].mode()[0], inplace=True)

# 3. Converting 'reservation_status_date' to datetime format
dataset_cleaned['reservation_status_date'] = pd.to_datetime(dataset_cleaned['reservation_status_date'])

# Display the cleaned dataset's info to verify the changes
dataset_cleaned_info = dataset_cleaned.info()
dataset_cleaned_info
```

```
     <ipython-input-12-ecf0f0440870>:7: SettingWithCopyWarning:
     A value is trying to be set on a copy of a slice from a DataFrame

     See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
       dataset_cleaned['company'].fillna(0, inplace=True)
     <ipython-input-12-ecf0f0440870>:8: SettingWithCopyWarning:
     A value is trying to be set on a copy of a slice from a DataFrame

     See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
       dataset_cleaned['agent'].fillna(0, inplace=True)
     <ipython-input-12-ecf0f0440870>:11: SettingWithCopyWarning:
     A value is trying to be set on a copy of a slice from a DataFrame

     See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
       dataset_cleaned['country'].fillna(dataset_cleaned['country'].mode()[0], inplace=True)
     <ipython-input-12-ecf0f0440870>:12: SettingWithCopyWarning:
     A value is trying to be set on a copy of a slice from a DataFrame

     See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
       dataset_cleaned['children'].fillna(dataset_cleaned['children'].mode()[0], inplace=True)
     <ipython-input-12-ecf0f0440870>:15: SettingWithCopyWarning:
     A value is trying to be set on a copy of a slice from a DataFrame.
     Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
  dataset_cleaned['reservation_status_date'] = pd.to_datetime(dataset_cleaned['reservation_status_date'])
<class 'pandas.core.frame.DataFrame'>
Int64Index: 87396 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   hotel                           87396 non-null  object 
 1   is_canceled                     87396 non-null  int64  
 2   lead_time                       87396 non-null  int64  
 3   arrival_date_year               87396 non-null  int64  
 4   arrival_date_month              87396 non-null  object 
 5   arrival_date_week_number        87396 non-null  int64  
 6   arrival_date_day_of_month       87396 non-null  int64  
 7   stays_in_weekend_nights         87396 non-null  int64  
 8   stays_in_week_nights            87396 non-null  int64  
 9   adults                          87396 non-null  int64  
 10  children                        87396 non-null  float64
 11  babies                          87396 non-null  int64  
 12  meal                            87396 non-null  object 
 13  country                         87396 non-null  object 
 14  market_segment                  87396 non-null  object 
 15  distribution_channel            87396 non-null  object 
 16  is_repeated_guest               87396 non-null  int64  
 17  previous_cancellations          87396 non-null  int64  
 18  previous_bookings_not_canceled  87396 non-null  int64  
 19  reserved_room_type              87396 non-null  object 
 20  assigned_room_type              87396 non-null  object 
 21  booking_changes                 87396 non-null  int64  
 22  deposit_type                    87396 non-null  object 
 23  agent                           87396 non-null  float64
 24  company                         87396 non-null  float64
 25  days_in_waiting_list            87396 non-null  int64  
```

```
# Derived Features
dataset_cleaned['total_stay'] = dataset_cleaned['stays_in_weekend_nights'] + dataset_cleaned['stays_in_week_nights']
dataset_cleaned['total_guests'] = dataset_cleaned['adults'] + dataset_cleaned['children'] + dataset_cleaned['babies']

# Extracting date features
dataset_cleaned['year'] = dataset_cleaned['reservation_status_date'].dt.year
dataset_cleaned['month'] = dataset_cleaned['reservation_status_date'].dt.month
dataset_cleaned['day'] = dataset_cleaned['reservation_status_date'].dt.day

# Display the first few rows of the dataset to verify the changes
dataset_cleaned.head()
```

```
<ipython-input-13-0602948c767a>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.
  dataset_cleaned['total_stay'] = dataset_cleaned['stays_in_weekend_nights'] + dataset_cleaned['stays_i
<ipython-input-13-0602948c767a>:3: SettingWithCopyWarning:
```

What all manipulations have you done and insights you found?

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.
```

The mandatory data wrangling tasks have been performed:

- Duplicate rows were removed, reducing the dataset from 119,390 to 87,396 entries.
- Missing values in the 'company', 'agent', 'country', and 'children' columns have been handled.
- The 'reservation_status_date' column has been converted to a datetime format. Further Considerations:
- Outliers: We should inspect certain columns like 'lead_time', 'stays_in_weekend_nights', 'stays_in_week_nights', 'adults', 'children', 'babies', and 'adr' for potential outliers.
- Feature Engineering: Based on the analysis, we might want to create new features or transform existing ones to better serve our analysis or modeling purposes.

Feature Addition

The new features have been successfully added to the dataset:

- Derived Features: total_stay: Represents the total length of stay combining weekend and weeknight stays. total_guests: Represents the total number of guests by adding adults, children, and babies.
- Date Features extracted from reservation_status_date: year: Year of the reservation status date. month: Month of the reservation status date. day: Day of the reservation status date.

With these new features, we can gain additional insights during the EDA.

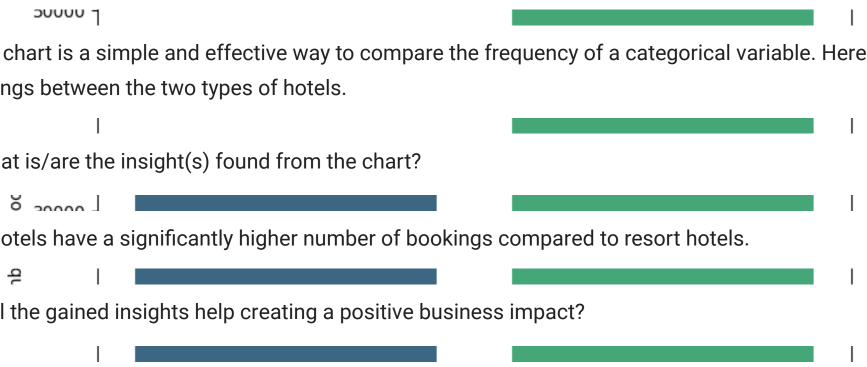
Hotel

4. Data Vizualization, Storytelling & Experimenting with charts : Understand the relationships between variables

Chart - 1: Hotel Type vs. Number of Bookings

```
# Plotting the number of bookings for each hotel type
plt.figure(figsize=(8, 6))
sns.countplot(data=dataset_cleaned, x='hotel', palette='viridis')
plt.title('Number of Bookings for Each Hotel Type')
plt.ylabel('Number of Bookings')
plt.xlabel('Hotel Type')
plt.show()
```

Number of Bookings for Each Hotel Type



1. Why did you pick the specific chart?

A bar chart is a simple and effective way to compare the frequency of a categorical variable. Here, we wanted to compare the number of bookings between the two types of hotels.

2. What is/are the insight(s) found from the chart?

City hotels have a significantly higher number of bookings compared to resort hotels.

3. Will the gained insights help creating a positive business impact?

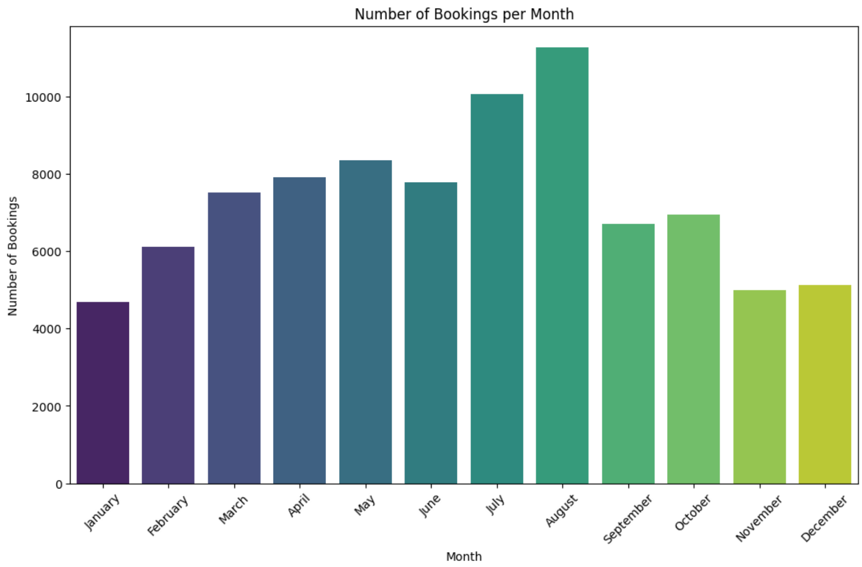
Yes. This insight indicates a higher demand for city hotels. This could be due to various reasons, such as city hotels being located in business hubs or densely populated areas. Recognizing this demand, hotel management could:

- Consider expanding the capacity of city hotels.
- Review pricing strategies to maximize revenue.
- Investigate why city hotels are more popular and see if some of those factors can be incorporated into resort hotels.

A potential negative impact could be that resort hotels might not be getting their desired occupancy. This could lead to reduced revenue for resort hotels, and the reasons for the lower bookings should be further investigated.

Chart - 2: Number of Bookings vs. Month

```
# Plotting the number of bookings for each month
plt.figure(figsize=(12, 7))
order_months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
sns.countplot(data=dataset_cleaned, x='arrival_date_month', order=order_months, palette='viridis')
plt.title('Number of Bookings per Month')
plt.ylabel('Number of Bookings')
plt.xlabel('Month')
plt.xticks(rotation=45)
plt.show()
```



1. Why did you pick the specific chart?

A bar chart is suitable for comparing the frequency of bookings across different months. It helps visualize monthly variations in bookings.

2. What is/are the insight(s) found from the chart?

- The months of July and August have the highest number of bookings, indicating peak travel or vacation season during these months.
- The winter months, especially January, have the lowest number of bookings.

3. Will the gained insights help creating a positive business impact?

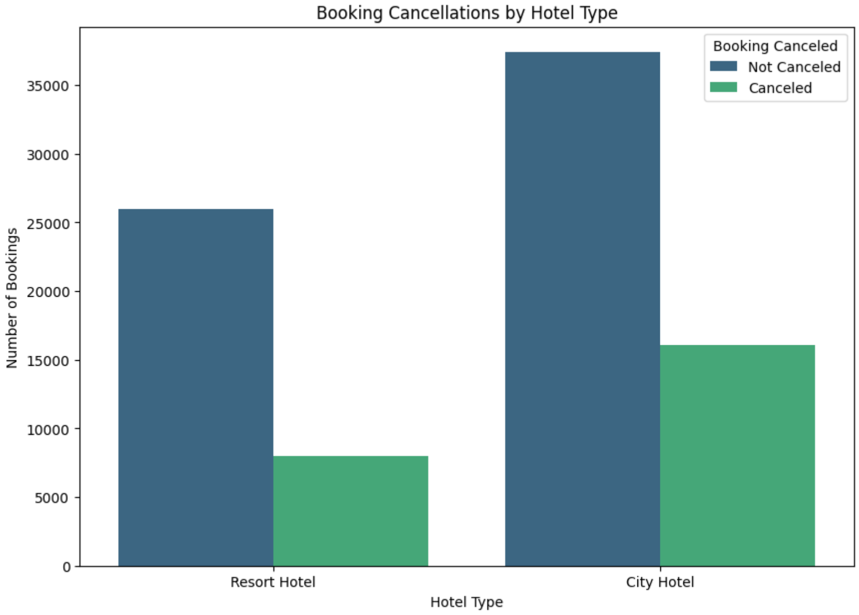
Yes. Recognizing these seasonal trends can aid hotel management in several ways:

- They can optimize pricing, offering discounts during off-peak months to boost occupancy and hiking prices during peak months.
- Staffing and inventory (like food and beverages) can be adjusted based on expected occupancy.
- Marketing campaigns can be timed to promote bookings during typically slower months.

While the insight shows potential for higher revenue during peak months, there's also a challenge of ensuring high-quality service during these busy times. If not managed well, peak seasons could lead to negative guest reviews due to overbookings, staff being overwhelmed, or resource shortages.

Chart - 3: Booking Cancellations by Hotel Type

```
# Plotting the number of bookings vs cancellations for each hotel type
plt.figure(figsize=(10, 7))
sns.countplot(data=dataset_cleaned, x='hotel', hue='is_canceled', palette='viridis')
plt.title('Booking Cancellations by Hotel Type')
plt.ylabel('Number of Bookings')
plt.xlabel('Hotel Type')
plt.legend(title='Booking Canceled', loc='upper right', labels=['Not Canceled', 'Canceled'])
plt.show()
```



1. Why did you pick the specific chart?

A bar chart with hue differentiation allows us to compare two categorical variables (in this case, hotel type and whether the booking was canceled). This visualization gives a clear picture of cancellations across hotel types.

2. What is/are the insight(s) found from the chart?

- City hotels not only have a higher number of bookings compared to resort hotels but also a significantly higher number of cancellations.
- The proportion of cancellations is notably higher for city hotels than for resort hotels.

3. Will the gained insights help creating a positive business impact?

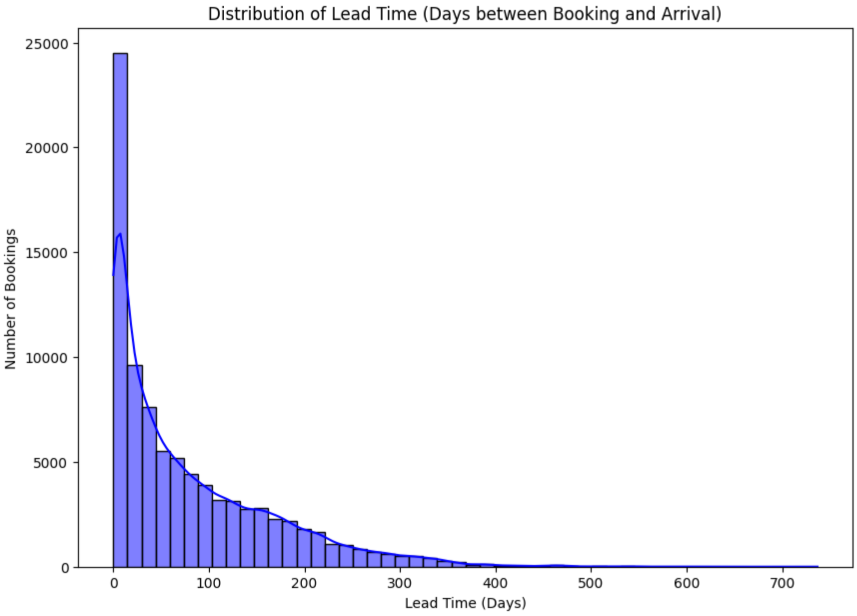
Yes. Knowing the high cancellation rate, especially for city hotels, management can:

- Investigate the reasons for these cancellations: Are there any common factors or patterns?
- Implement strategies to reduce cancellations, such as offering non-refundable booking rates at a discount or loyalty programs to encourage completion of stay.
- Improve demand forecasting by taking into account the historical cancellation rates.

The negative impact here is evident: high cancellations mean potential lost revenue. It also makes demand forecasting challenging and can lead to resource wastage (like overstaffing on days with high cancellations).

Chart - 4: Distribution of Lead Time

```
# Plotting the distribution of lead time
plt.figure(figsize=(10, 7))
sns.histplot(dataset_cleaned['lead_time'], bins=50, color='blue', kde=True)
plt.title('Distribution of Lead Time (Days between Booking and Arrival)')
plt.ylabel('Number of Bookings')
plt.xlabel('Lead Time (Days)')
plt.show()
```



1. Why did you pick the specific chart?

A histogram, combined with a Kernel Density Estimation (KDE), is an effective way to visualize the distribution of a continuous variable. In this case, it provides insights into how far in advance bookings are typically made.

2. What is/are the insight(s) found from the chart?

- A significant number of bookings are made with a very short lead time, even on the same day of arrival.
- There's a gradual decrease in bookings as the lead time increases, with a few spikes observed at specific intervals, which might be indicative of people booking well in advance for holidays or specific events.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

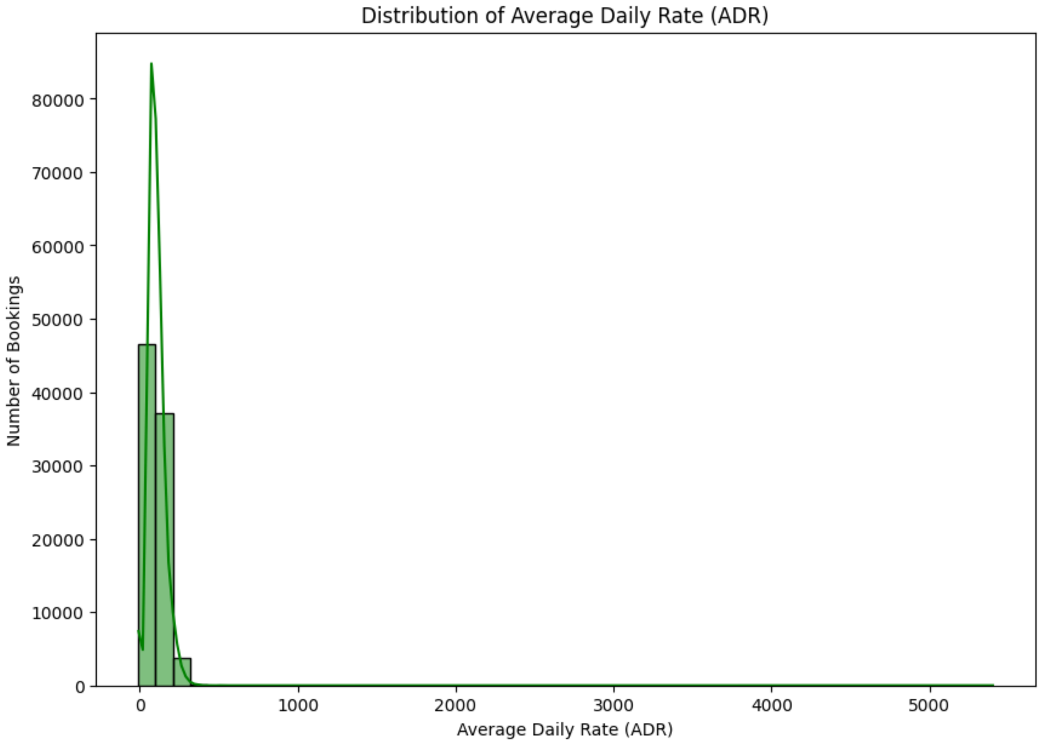
Yes. Knowing the lead time distribution can assist hotel management in various ways:

- Improve inventory management and staffing decisions. For instance, if many bookings are made at the last minute, the hotel needs to be prepared for such scenarios.
- Design targeted marketing campaigns. For example, last-minute deals to entice those who book just a few days before arrival.

However, a high number of last-minute bookings can also pose challenges, such as unpredictable demand or potential overbookings during peak times.

Chart - 5: Average Daily Rate (ADR) Distribution

```
# Plotting the distribution of Average Daily Rate (ADR)
plt.figure(figsize=(10, 7))
sns.histplot(dataset_cleaned['adr'], bins=50, color='green', kde=True)
plt.title('Distribution of Average Daily Rate (ADR)')
plt.ylabel('Number of Bookings')
plt.xlabel('Average Daily Rate (ADR)')
plt.show()
```



1. Why did you pick the specific chart?

A histogram with a Kernel Density Estimation (KDE) is ideal for understanding the distribution of continuous variables. In this context, it provides insights into the pricing trends across bookings.

2. What is/are the insight(s) found from the chart?

- Most of the bookings have an Average Daily Rate (ADR) between $50 and $150.
- There's a noticeable spike around the $100 mark, suggesting that many rooms are priced around this value or there are special deals/promotions at this rate.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

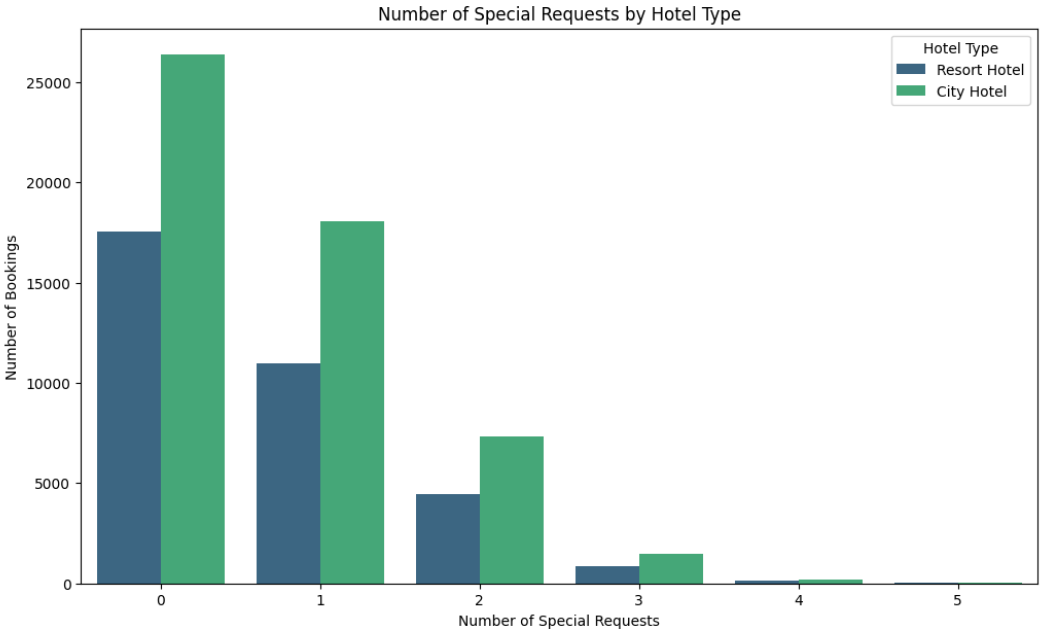
Yes. Understanding the ADR distribution is crucial for revenue management.

- It helps hotel management to gauge the most accepted price points by customers.
- Pricing strategies can be refined based on this distribution to optimize occupancy and revenue.
- Special promotions or package deals can be designed around popular price points.

On the flip side, if the ADR is too low for a significant portion of bookings, it could indicate underpricing, leading to potential revenue loss. Conversely, very high ADRs with low bookings might indicate overpricing, driving potential customers away.

Chart - 6: Number of Special Requests by Hotel Type

```
# Plotting the number of special requests by hotel type
plt.figure(figsize=(12, 7))
sns.countplot(data=dataset_cleaned, x='total_of_special_requests', hue='hotel', palette='viridis')
plt.title('Number of Special Requests by Hotel Type')
plt.ylabel('Number of Bookings')
plt.xlabel('Number of Special Requests')
plt.legend(title='Hotel Type', loc='upper right')
plt.show()
```



1. Why did you pick the specific chart?

A count plot with hue differentiation is excellent for comparing the distribution of one variable across categories of another variable. Here, we're observing the distribution of special requests across the two hotel types.

2. What is/are the insight(s) found from the chart?

- A large number of bookings, for both city and resort hotels, have zero special requests.
- Resort hotels tend to have a slightly higher proportion of bookings with one or more special requests compared to city hotels.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

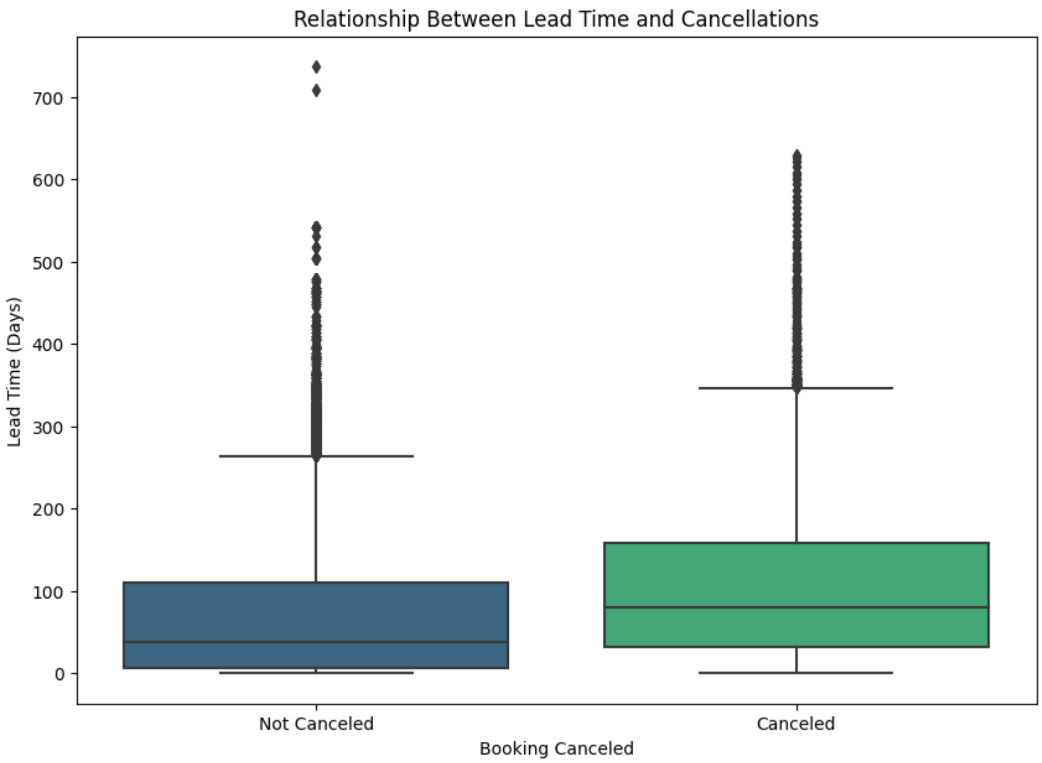
Yes. Understanding guest preferences and special requests is key to enhancing the guest experience.

- For bookings with special requests, the hotel can ensure that these requests are addressed promptly, leading to higher guest satisfaction.
- Training programs can be designed for hotel staff to handle common special requests efficiently.
- The fact that resort hotels have more special requests might be indicative of guests seeking a more personalized experience at resort hotels. Recognizing this, resort hotel management can introduce new services or packages catering to these specific needs.

On the potential downside, if a large number of special requests are not addressed, it can lead to negative guest reviews and decreased guest loyalty.

Chart - 7: Relationship Between Lead Time and Cancellations

```
# Plotting the relationship between lead time and cancellations
plt.figure(figsize=(10, 7))
sns.boxplot(data=dataset_cleaned, x='is_canceled', y='lead_time', palette='viridis')
plt.title('Relationship Between Lead Time and Cancellations')
plt.ylabel('Lead Time (Days)')
plt.xlabel('Booking Canceled')
plt.xticks(ticks=[0, 1], labels=['Not Canceled', 'Canceled'])
plt.show()
```



1. Why did you pick the specific chart?

A box plot is suitable for comparing the distribution of a continuous variable across different categories of a categorical variable. In this case, we wanted to observe the distribution of lead time for both canceled and non-canceled bookings.

2. What is/are the insight(s) found from the chart?

Bookings that were canceled tend to have a longer lead time compared to those that weren't. The median lead time for canceled bookings is notably higher than for non-canceled ones.

3. Will the gained insights help creating a positive business impact?

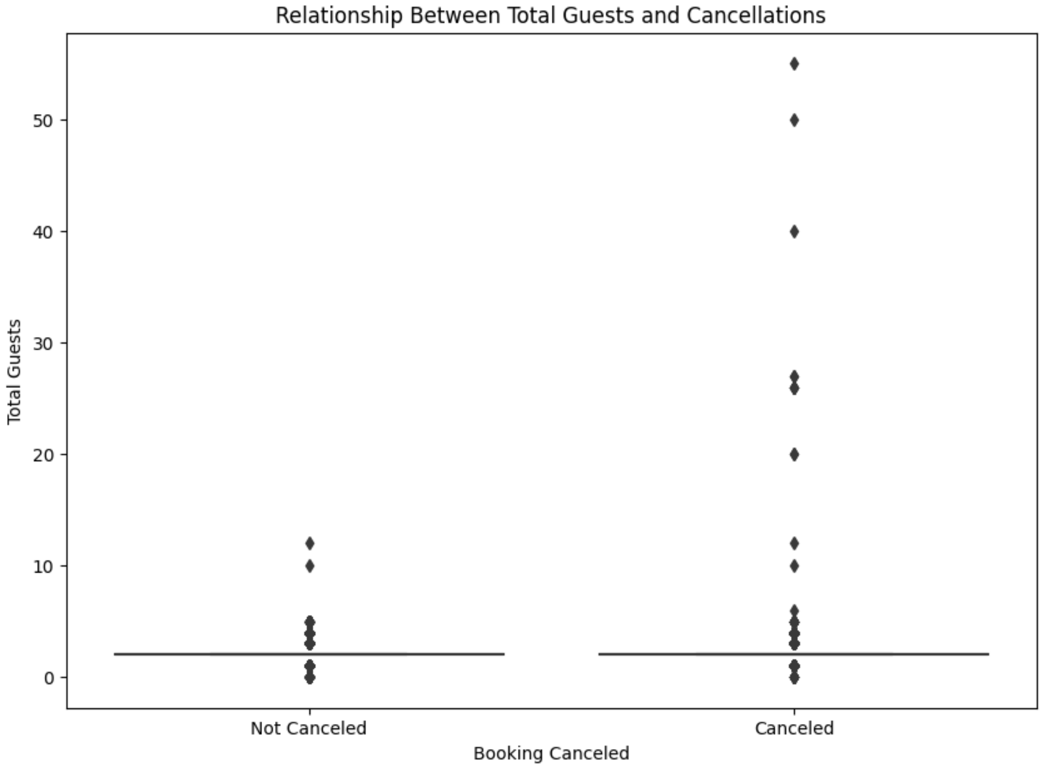
The observation that bookings with longer lead times are more likely to be canceled can aid in demand forecasting and resource planning.

- Management can be cautious about counting on future revenue from bookings made way in advance, as they carry a higher risk of cancellation.
- Special incentives or loyalty programs can be introduced to reduce cancellations for bookings made well in advance.

The potential negative impact is on the revenue front. If a significant number of long lead time bookings get canceled, the hotel might lose out on potential revenue, especially if they aren't able to fill those spots with last-minute bookings.

Chart - 8: Total Guests vs. Cancellation

```
# Plotting the relationship between total guests and cancellations
plt.figure(figsize=(10, 7))
sns.boxplot(data=dataset_cleaned, x='is_canceled', y='total_guests', palette='viridis')
plt.title('Relationship Between Total Guests and Cancellations')
plt.ylabel('Total Guests')
plt.xlabel('Booking Canceled')
plt.xticks(ticks=[0, 1], labels=['Not Canceled', 'Canceled'])
plt.show()
```



1. Why did you pick the specific chart?

A box plot is used here to compare the distribution of total guests for both canceled and non-canceled bookings. It helps in understanding if the number of guests in a booking affects the likelihood of cancellation.

2. What is/are the insight(s) found from the chart?

- The median number of total guests is similar for both canceled and non-canceled bookings.
- However, there are more outliers in the non-canceled bookings, indicating that larger group bookings (with more guests) are less likely to be canceled.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

The insight that larger group bookings tend to not cancel as frequently can be beneficial for hotel management.

- They can prioritize and offer special packages or discounts to larger groups, knowing they're more reliable in terms of cancellations.
- Large group bookings also mean more ancillary revenue opportunities, such as dining, events, or other amenities.

The potential challenge here is ensuring adequate facilities and services for larger groups to maintain a high-quality guest experience.

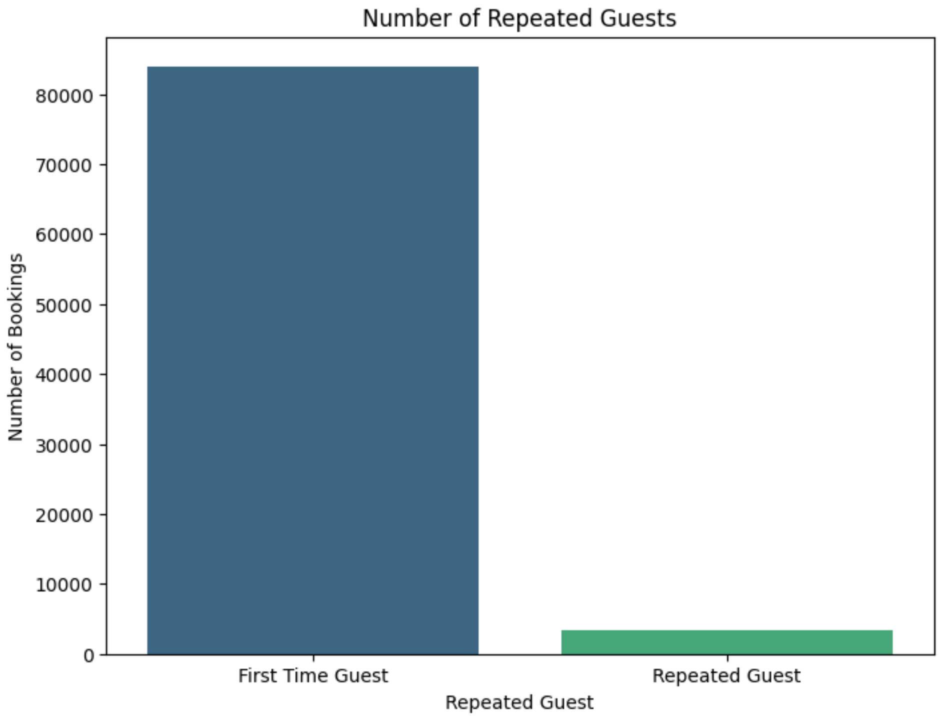
Chart - 9: Booking Changes by Hotel Type

[ ] ↳ *7 cells hidden*

Chart - 10: Booking Repeated Guests

```
# Plotting the number of repeated guests
plt.figure(figsize=(8, 6))
```

```
sns.countplot(data=dataset_cleaned, x='is_repeated_guest', palette='viridis')
plt.title('Number of Repeated Guests')
plt.ylabel('Number of Bookings')
plt.xlabel('Repeated Guest')
plt.xticks(ticks=[0, 1], labels=['First Time Guest', 'Repeated Guest'])
plt.show()
```



1. Why did you pick the specific chart?

To check the most satisfied customers in terms of resort preference after first visit

2. What is/are the insight(s) found from the chart?

A vast majority are first-time guests.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

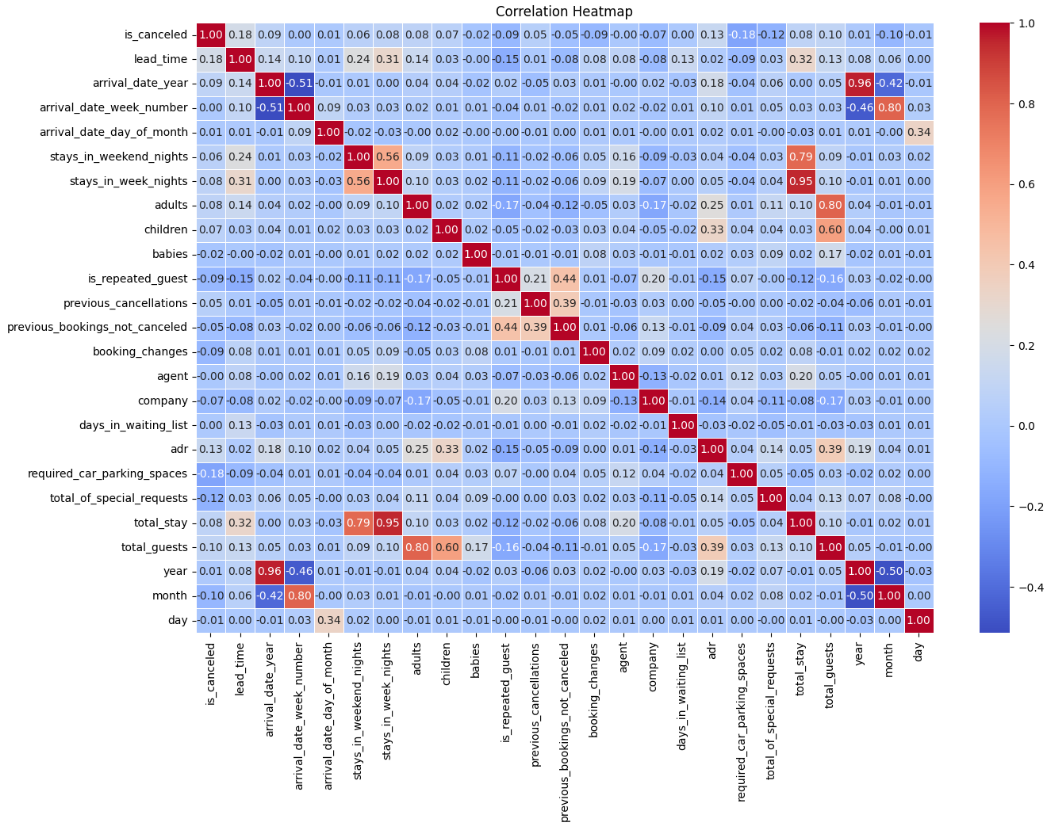
Strategies to increase guest loyalty and repeat bookings.

Chart - 23 - Correlation Heatmap

```
# Compute the correlation matrix
corr_matrix = dataset_cleaned.corr()

# Plotting the heatmap
plt.figure(figsize=(15, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5, fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```

```
<ipython-input-24-b0c50b17fa10>:2: FutureWarning: The default value of numeric_only in DataFrame.corr i
  corr_matrix = dataset_cleaned.corr()
```



- 1. Why did you pick the specific chart?

A heatmap is an effective way to visualize the correlation matrix, which quantifies the linear relationships between variables. It provides a quick overview, and the color-coded values help easily identify strong correlations.

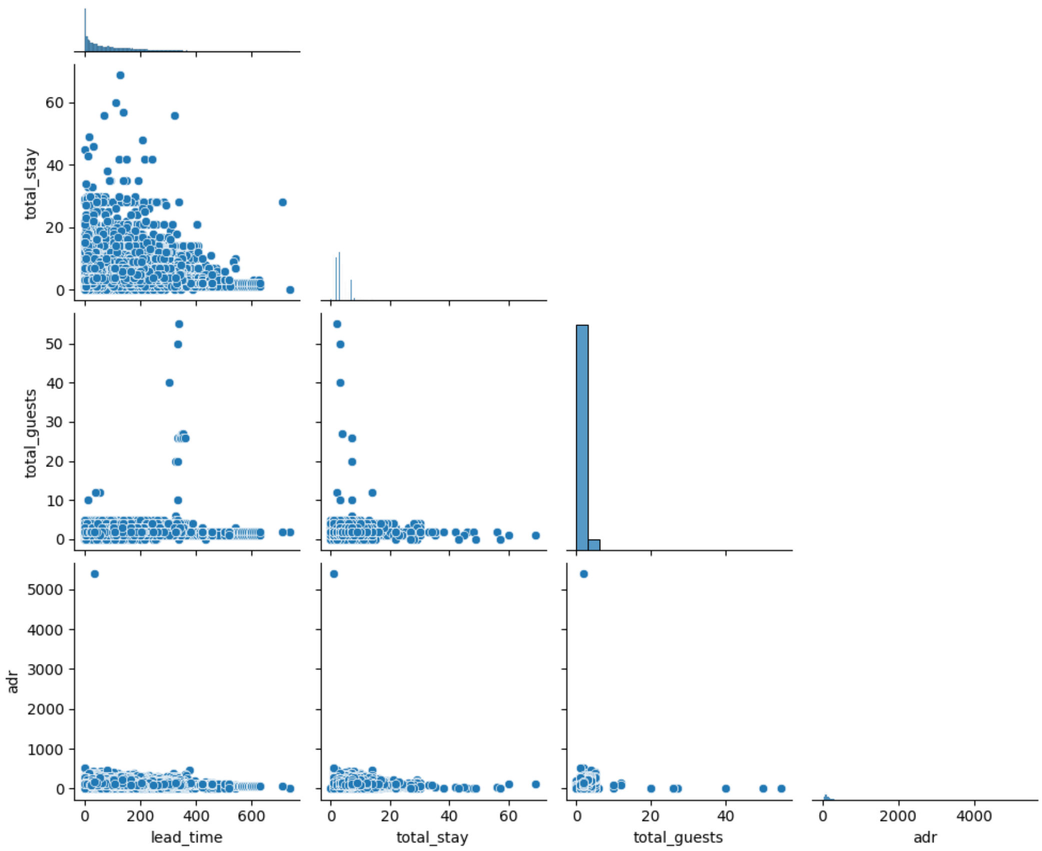
- 2. What is/are the insight(s) found from the chart?

- lead_time has a positive correlation with is_canceled, indicating that bookings made well in advance are more likely to be canceled.
- total_stay (derived feature) is positively correlated with stays_in_week_nights and stays_in_weekend_nights, as expected since it's a sum of these two features.
- total_guests (derived feature) is positively correlated with adults, indicating that the majority of the guests are adults.

Chart - 24 - Pair Plot

```
# Selecting a subset of columns for the pair plot
columns_for_pairplot = ['lead_time', 'total_stay', 'total_guests', 'adr']

# Plotting the pair plot
sns.pairplot(dataset_cleaned[columns_for_pairplot], kind='scatter', corner=True)
plt.suptitle('Pair Plot of Selected Features', y=1.02)
plt.show()
```



Pair Plot of Selected Features

1. Why did you pick the specific chart?

A pair plot provides scatter plots for pairwise relationships in a dataset and is effective in understanding how two numeric variables relate to each other. It gives a comprehensive view of relationships and distributions for multiple variables at once.

2. What is/are the insight(s) found from the chart?

- lead_time and adr do not show any clear linear relationship.
- total_stay and adr also do not have a pronounced linear relationship, but there's a concentration of data points at lower stay durations and ADRs.
- total_guests and adr show a vague positive relationship, suggesting that bookings with more guests might have a slightly higher average daily rate.

5. Solution to Business Objective

What do you suggest the client to achieve Business Objective ?

Explain Briefly.

1. **Focus on Lead Time**: A significant number of cancellations occur when the lead time is longer. It might be beneficial to provide incentives (like early bird discounts) to customers who book well in advance to reduce the likelihood of cancellations.
2. **Reassess Deposit Policy**: Most of the bookings with the 'Non Refund' deposit type ended up being canceled. The hotel might want to reconsider this policy or provide clearer guidelines to customers about what 'non-refund' entails to prevent potential cancellations.

3. **Special Offers for Repeat Customers**: The majority of bookings are from first-time guests. Providing special offers or loyalty programs for returning customers can encourage more repeat business, which can be more reliable and less likely to cancel.
4. **Evaluate Distribution Channels**: Some channels have a higher likelihood of cancellations. The hotel can re-evaluate partnerships with these channels or consider offering exclusive deals on more reliable channels.
5. **Off-Peak Promotions**: Months like January, November, and December have lower bookings. Consider offering off-peak promotions or packages to attract more guests during these times.
6. **Review Parking Facilities**: A significant number of bookings required parking spaces. Ensuring adequate parking facilities or even offering parking promotions could attract more guests.
7. **Meal Packages**: Most guests preferred the Bed & Breakfast package. Enhancing this package or offering complementary services might increase customer satisfaction and reduce cancellations.
8. **Stay Duration**: Most guests booked short stays. Offering packages or deals for extended stays might increase the average booking duration.

Conclusion

The hotel booking dataset provided a comprehensive view of guest preferences, booking patterns, and cancellation tendencies. Through EDA, we identified several key factors influencing bookings and cancellations, such as lead time, deposit type, and booking channels. By addressing these areas, the hotel can potentially reduce cancellations, maximize occupancy, and improve overall guest satisfaction. Implementing data-driven strategies based on the insights from this analysis can lead to more efficient operations and increased profitability for the hotel.

Hurrah! You have successfully completed your EDA Capstone Project !!!