

MY INTERNSHIP PROJECT



Project Name - Hotel Booking Analysis

Project By - Aditya Kumar Singh

Submitted To - Vijay Bljwan Sir

Project Summary -

This EDA project on Hotel Booking Analysis investigates cancellations, and their underlying patterns; and suggests measures that can be implemented to reduce cancellations and secure revenue1:

The project covers booking information for a city hotel and a resort hotel including information such as when the booking was made, length of stay, the number of adults, children. The project went through the basic idea of the EDA and visualization process.

In this project I will do Exploratory Data Analysis on the given dataset. The project suggests measures that can be implemented to reduce cancellations and secure revenue. For example, hotels can offer discounts or promotions to customers who book early or who book for longer stays. Hotels can also offer incentives such as free parking or free breakfast to customers who book directly with them instead of through third-party websites.

This EDA involves following steps where in first step involves exploration and inspection over raw data, and second in second step I have dealt with data impurities and cleaned the data by handling null values and dropping irrelevant data from the dataset.

The project concludes that by analyzing hotel bookings data and understanding cancellations patterns, hotels can take steps to reduce cancellations and increase revenue.

Problem Statement

Have you ever wondered when the best time of year to book a hotel room is? Or the optimal length of stay in order to get the best daily rate? What if you wanted to predict whether or not a hotel was likely to receive a disproportionately high number of special requests? This hotel booking dataset can help you explore those questions! This data set contains booking information for a city hotel and a resort hotel, and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things. All personally identifying information has been removed from the data..

Define Your Business Objective?

The project aims to gain interesting insight into customers' behavior when booking a hotel. The demand for different segment of customer may differ and forecasting become harder as it may requires different model for different segment.These insights can guide hotels to adjust their customer strategies and make preparation for unknown.

1. Know Your Data

Import Libraries

```
In [2]: # Importing necessary libraries needed in EDA
import numpy as np
import pandas as pd
# for visualisation
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
# will be used for plotting
import plotly.express as px
```

Dataset Loading

```
In [4]: #Loading the dataset
hb_df = pd.read_csv('Hotel Bookings.csv')
```

```
In [5]: hb_df.shape
```

```
Out[5]: (119390, 32)
```

Dataset First View

```
In [6]: # Dataset First Look
hb_df
```

```
Out[6]:
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weeks
0	Resort Hotel	0	342	2015	July	27		1
1	Resort Hotel	0	737	2015	July	27		1
2	Resort Hotel	0	7	2015	July	27		1
3	Resort Hotel	0	13	2015	July	27		1
4	Resort Hotel	0	14	2015	July	27		1
...
119385	City Hotel	0	23	2017	August	35		30
119386	City Hotel	0	102	2017	August	35		31
119387	City Hotel	0	34	2017	August	35		31
119388	City Hotel	0	109	2017	August	35		31
119389	City Hotel	0	205	2017	August	35		29

119390 rows × 32 columns

```
In [7]: #Looking first 5 rows of the dataset
hb_df.head()
```

```
Out[7]:
```

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

5 rows × 32 columns

```
In [8]: #Looking the last 5 rows of the dataset
hb_df.tail()
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights
119385	City Hotel	0	23	2017	August	35	30	
119386	City Hotel	0	102	2017	August	35	31	
119387	City Hotel	0	34	2017	August	35	31	
119388	City Hotel	0	109	2017	August	35	31	
119389	City Hotel	0	205	2017	August	35	29	

5 rows × 32 columns

Dataset Rows & Columns count

```
In [9]: print(f'Number of rows : {len(hb_df.axes[0])}')
print(f'Number of rows : {len(hb_df.axes[1])}'')
```

Number of rows : 119390
Number of rows : 32

Dataset Information

```
In [10]: hb_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
```

Duplicate Values

```
In [10]: # Dataset Duplicate Value Count
hb_df.duplicated().sum()
```

Out[10]: 31994

There are 31994 duplicate values in the dataset

```
In [11]: #Dropping the duplicate values
hb_df.drop_duplicates(inplace = True)
```

```
In [12]: hb_df.shape
```

```
Out[12]: (87396, 32)
```

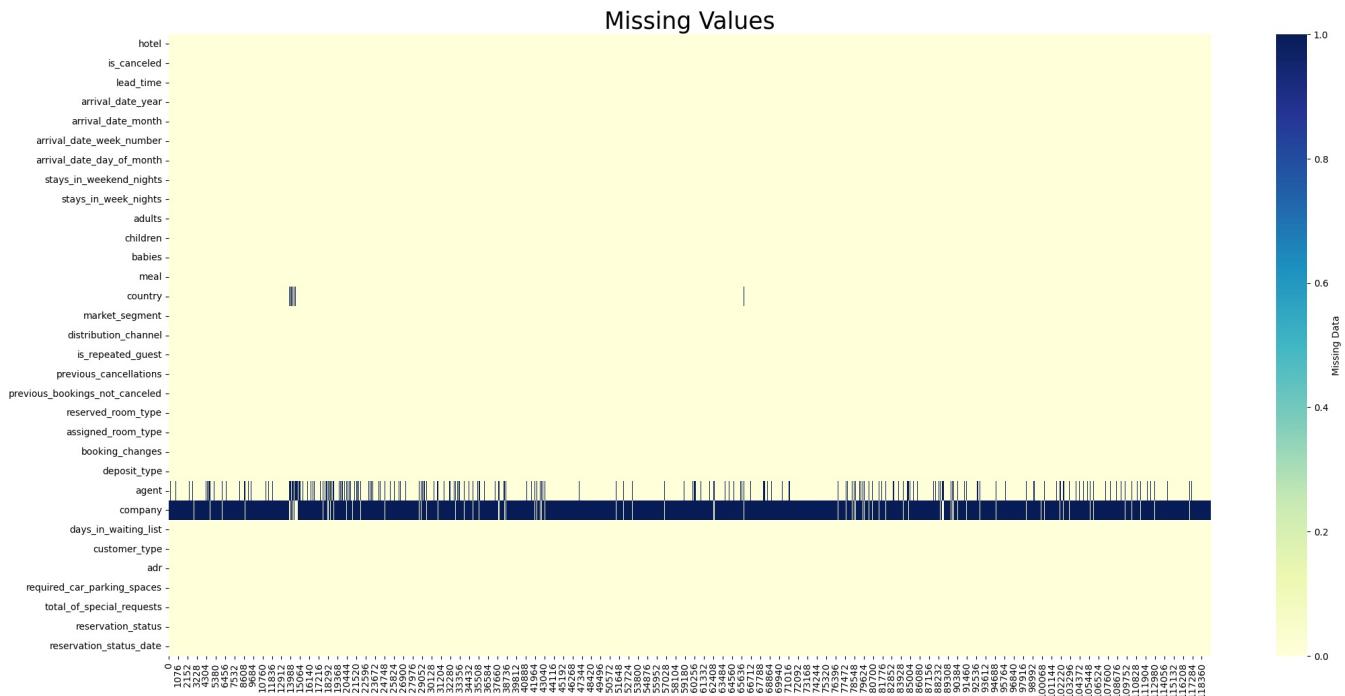
Missing Values/Null Values

```
In [13]: # Missing Values/Null Values Count  
hb_df.isnull().sum()
```

```
Out[13]: 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_notCanceled 0  
reserved_room_type 0  
assigned_room_type 0  
booking_changes 0  
deposit_type 0  
agent 12193  
company 82137  
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
```

```
In [14]: # Visualizing the missing values using Seaborn heatmap
```

```
plt.figure(figsize=(25,12))  
sns.heatmap(hb_df.isna().transpose(),  
            cmap="YlGnBu",  
            cbar_kws={'label': 'Missing Data'})  
  
plt.title('Missing Values', fontsize=25)  
plt.show()
```



What did you know about your dataset?

We can see that there are total four columns with missing/null values : company, agent, country, children.

1. In children column, I will replace null values with 0 assuming that customer did not have any children.

2. Column country has null values. I will replace null values in this column with 'Others' assuming customer's country was not mentioned while booking.
3. In company and agent column it might be a case when customers did not book hotel through them so these columns might have null values in it. As these 2 columns have numeric data in it, I will replace them with 0.

2. Understanding Your Variables

```
In [15]: # Dataset Columns
hb_df.columns
```

```
Out[15]: 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')
```

```
In [16]: # Dataset Describe
hb_df.describe()
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights
count	87396.000000	87396.000000	87396.000000	87396.000000	87396.000000	87396.000000	87396.000000
mean	0.274898	79.891368	2016.210296	26.838334	15.815541	1.005263	
std	0.446466	86.052325	0.686102	13.674572	8.835146	1.031921	
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000	
25%	0.000000	11.000000	2016.000000	16.000000	8.000000	0.000000	
50%	0.000000	49.000000	2016.000000	27.000000	16.000000	1.000000	
75%	1.000000	125.000000	2017.000000	37.000000	23.000000	2.000000	
max	1.000000	737.000000	2017.000000	53.000000	31.000000	19.000000	

Variables Description

Hotel : (Resort Hotel or City Hotel)

is_canceled: Value indicating if the booking was canceled (1) or not (0)

lead_time : Number of days that elapsed between the entering date of the booking into the PMS and the arrival date

arrival_date_year : Year of arrival date

arrival_date_month : Month of arrival date

arrival_date_week_number : Week number of year for arrival date

arrival_date_day_of_month : Day of arrival date

stays_in_weekend_nights : Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel

stays_in_week_nights : Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel

adults : Number of adults

children : Number of children

babies : Number of babies

meal : Type of meal booked. Categories are presented in standard hospitality meal packages

country : Country of origin. `market_segment : Market segment designation. In categories, the term "TA" means "Travel Agents" and "TO" means "Tour Operators"

distribution_channel : Booking distribution channel. The term "TA" means "Travel Agents" and "TO" means "Tour Operators"

is_repeated_guest : Value indicating if the booking name was from a repeated guest (1) or not (0)

previous_cancellations : Number of previous bookings that were cancelled by the customer prior to the current booking

previous_bookings_not_canceled : Number of previous bookings not cancelled by the customer prior to the current booking

reserved_room_type : Code of room type reserved. Code is presented instead of designation for anonymity reasons.

assigned_room_type : Code for the type of room assigned to the booking.

booking_changes : Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation

deposit_type : Indication on if the customer made a deposit to guarantee the booking.

agent : ID of the travel agency that made the booking

company : ID of the company/entity that made the booking or responsible for paying the booking.

days_in_waiting_list : Number of days the booking was in the waiting list before it was confirmed to the customer

customer_type : Type of booking, assuming one of four categories

adr : Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights

required_car_parking_spaces : Number of car parking spaces required by the customer

total_of_special_requests : Number of special requests made by the customer (e.g. twin bed or high floor)

reservation_status : Reservation last status, assuming one of three categories

Canceled – booking was canceled by the customer **Check-Out** – customer has checked in but already departed **No-Show** – customer did not check-in and did inform the hotel of the reason why **reservation_status_date** - Date at which the last status was set

Check Unique Values for each variable.

```
In [17]: # Check Unique Values for each variable.  
pd.Series({col:hb_df[col].unique() for col in hb_df})
```

```
Out[17]:
```

Variable	Unique Values
hotel	[Resort Hotel, City Hotel]
is_canceled	[0, 1]
lead_time	[342, 737, 7, 13, 14, 0, 9, 85, 75, 23, 35, 68...]
arrival_date_year	[2015, 2016, 2017]
arrival_date_month	[July, August, September, October, November, D...]
arrival_date_week_number	[27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 3...]
arrival_date_day_of_month	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...]
stays_in_weekend_nights	[0, 1, 2, 4, 3, 6, 13, 8, 5, 7, 12, 9, 16, 18,...]
stays_in_week_nights	[0, 1, 2, 3, 4, 5, 10, 11, 8, 6, 7, 15, 9, 12,...]
adults	[2, 1, 3, 4, 40, 26, 50, 27, 55, 0, 20, 6, 5, 10]
children	[0.0, 1.0, 2.0, 10.0, 3.0, nan]
babies	[0, 1, 2, 10, 9]
meal	[BB, FB, HB, SC, Undefined]
country	[PRT, GBR, USA, ESP, IRL, FRA, nan, ROU, NOR, ...]
market_segment	[Direct, Corporate, Online TA, Offline TA/T0, ...]
distribution_channel	[Direct, Corporate, TA/T0, Undefined, GDS]
is_repeated_guest	[0, 1]
previous_cancellations	[0, 1, 2, 3, 26, 25, 14, 4, 24, 19, 5, 21, 6, ...]
previous_bookings_not_canceled	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,...]
reserved_room_type	[C, A, D, E, G, F, H, L, P, B]
assigned_room_type	[C, A, D, E, G, F, I, B, H, P, L, K]
booking_changes	[3, 4, 0, 1, 2, 5, 17, 6, 8, 7, 10, 16, 9, 13,...]
deposit_type	[No Deposit, Refundable, Non Refund]
agent	[nan, 304.0, 240.0, 303.0, 15.0, 241.0, 8.0, 2...]
company	[nan, 110.0, 113.0, 270.0, 178.0, 240.0, 154.0...]
days_in_waiting_list	[0, 50, 47, 65, 122, 75, 101, 150, 125, 14, 60...]
customer_type	[Transient, Contract, Transient-Party, Group]
adr	[0.0, 75.0, 98.0, 107.0, 103.0, 82.0, 105.5, 1...]
required_car_parking_spaces	[0, 1, 2, 8, 3]
total_of_special_requests	[0, 1, 3, 2, 4, 5]
reservation_status	[Check-Out, Canceled, No-Show]
reservation_status_date	[2015-07-01, 2015-07-02, 2015-07-03, 2015-05-0...]
dtype: object	

```
In [15]: # creating a duplicate of the original dataset before making any changes in it  
hb_df1 = hb_df.copy()
```

```
In [19]: hb_df1.columns
```

```
Out[19]: 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')
```

```
In [20]: # replacing null values in children column with 0 assuming that family had 0 children
# replacing null values in company and agent columns with 0 assuming those rooms were booked without company/agent

hb_df1['children'].fillna(0, inplace = True)
hb_df1['company'].fillna(0, inplace = True)
hb_df1['agent'].fillna(0, inplace = True)

# replacing null values in country column as 'Others'

hb_df1['country'].fillna('Others', inplace = True)
```

```
In [21]: # checking for null values after replacing them
hb_df1.isnull().sum()
```

```
Out[21]: 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          0
babies             0
meal               0
country            0
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              0
company             0
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
```

```
In [23]: # dropping the 'company' column as it contains a lot of null values in comparison to other columns
hb_df1.drop(['company'], axis =1 , inplace = True)           # dropping the values vertically at axis 1 (columns)
```

```
In [24]: # dropping rows where no adults , children and babies are available because no bookings were made that day

no_guest=hb_df1[hb_df1['adults']+hb_df1['babies']+hb_df1['children']==0]
hb_df1.drop(no_guest.index, inplace=True)
```

```
In [25]: # adding some new columns to make our data analysis ready
hb_df1['total_people'] = hb_df1['adults'] + hb_df1['babies'] + hb_df1['children']           # creating total people

hb_df1['total_stay'] = hb_df1['stays_in_weekend_nights'] + hb_df1['stays_in_week_nights']    # creating a column for total stay
```

```
In [26]: # having a final look to check if our dataset is ready to analyse
hb_df1.head()
```

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

5 rows × 33 columns

In [27]: `hb_df1.tail()`

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights
119385	City Hotel	0	23	2017	August	35		30
119386	City Hotel	0	102	2017	August	35		31
119387	City Hotel	0	34	2017	August	35		31
119388	City Hotel	0	109	2017	August	35		31
119389	City Hotel	0	205	2017	August	35		29

5 rows × 33 columns

In [28]: `# checking the final shape of the dataset
print(f' final shape of the dataset is {hb_df1.shape}')`

final shape of the dataset is (87230, 33)

In [29]: `# checking the unique values which is to be analysed
pd.Series({col:hb_df1[col].unique() for col in hb_df1})`

```
Out[29]: hotel                               [Resort Hotel, City Hotel]
is_canceled                           [0, 1]
lead_time                             [342, 737, 7, 13, 14, 0, 9, 85, 75, 23, 35, 68...]
arrival_date_year                     [2015, 2016, 2017]
arrival_date_month                    [July, August, September, October, November, D...
arrival_date_week_number              [27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 3...
arrival_date_day_of_month             [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...
stays_in_weekend_nights               [0, 1, 2, 4, 3, 6, 13, 8, 5, 7, 12, 9, 16, 18, ...
stays_in_week_nights                  [0, 1, 2, 3, 4, 5, 10, 11, 8, 6, 7, 15, 9, 12, ...
adults                                [2, 1, 3, 4, 40, 26, 50, 27, 55, 20, 6, 5, 10, 0]
children                             [0.0, 1.0, 2.0, 10.0, 3.0]
babies                                [0, 1, 2, 10, 9]
meal                                   [BB, FB, HB, SC, Undefined]
country                                [PRT, GBR, USA, ESP, IRL, FRA, Others, ROU, NO...
market_segment                         [Direct, Corporate, Online TA, Offline TA/TO, ...
distribution_channel                   [Direct, Corporate, TA/TO, Undefined, GDS]
is_repeated_guest                     [0, 1]
previous_cancellations                [0, 1, 2, 3, 26, 25, 14, 4, 24, 19, 5, 21, 6, ...]
previous_bookings_notCanceled         [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...]
reserved_room_type                    [C, A, D, E, G, F, H, L, B]
assigned_room_type                    [C, A, D, E, G, F, I, B, H, L, K]
booking_changes                       [3, 4, 0, 1, 2, 5, 17, 6, 8, 7, 10, 16, 9, 13, ...]
deposit_type                           [No Deposit, Refundable, Non Refund]
agent                                    [0.0, 304.0, 240.0, 303.0, 15.0, 241.0, 8.0, 2...
days_in_waiting_list                  [0, 50, 47, 65, 122, 75, 101, 150, 125, 14, 60...]
customer_type                          [Transient, Contract, Transient-Party, Group]
adr                                     [0.0, 75.0, 98.0, 107.0, 103.0, 82.0, 105.5, 1...
required_car_parking_spaces           [0, 1, 2, 8, 3]
total_of_special_requests             [0, 1, 3, 2, 4, 5]
reservation_status                    [Check-Out, Canceled, No-Show]
reservation_status_date                 [2015-07-01, 2015-07-02, 2015-07-03, 2015-05-0...
total_people                            [2.0, 1.0, 3.0, 4.0, 5.0, 12.0, 40.0, 26.0, 50...
total_stay                             [0, 1, 2, 3, 4, 5, 6, 7, 14, 15, 10, 11, 8, 9, ...]
dtype: object
```

We can see that we have dealt with all the null values and added some new columns and now our dataset is ready to analysed.

What all manipulations have you done and insights you found?

Created a copy of the dataset before doing any manipulation then filled missing values with 0 in children , company and agent columns as those columns had numerical values and in column country filled missing values with 'others'. after dealing with missing values I dropped the country column as this had 96% missing values and was of no use in our analysis. In next step I created 2 new columns named 'total_people' and 'total_stay' for further analysis. In total people column I added all the babies, children and adults. similarly in second new column I added weekend stay and week stay column.

After doing all the manipulation I checked new manipulated dataset to check if this is ready to be analyzed.

After manipulating the dataset these were the **insights I found**:

1. There are 2 types of hotel which guests could book so I can find which type of hotel was booked most.
2. There are different types of guests and they come from different countries.
3. Guests can choose different foods from the menu.
4. Guests can book hotel directly or through different channels that are available.
5. Guests can cancel their booking and there are repeated guests also.
6. Guests can choose rooms of their liking while booking.
7. There is column available in the dataset named 'adr' which could be used to analyze hotel's performance on the basis of revenue.

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

Chart - 1

Which type of hotel is most preffered by the guests?

In [47]:

```
# Chart - 1 visualization code
# Storing unique hotel names in a variable
hotel_name = hb_df1['hotel'].unique()

# Checking the number of unique booking in each hotel type
unique_booking = hb_df1.hotel.value_counts().sort_values(ascending=True)

# Creating a donut chart using plotly.express
fig1 = px.pie(names = hotel_name, values = unique_booking, hole = 0.5, color = hotel_name,
               color_discrete_map={
                   'Resort Hotel': 'teal' , 'City Hotel' : 'nude'})

# Giving it a title and updating the text info
fig1.update_traces(textinfo = 'percent + value')
fig1.update_layout(title_text = 'Hotel Booking Percentage', title_x = 0.5)

# Setting the legend at center
fig1.update_layout(legend=dict(
    orientation = 'h',
    yanchor = 'bottom',
    xanchor = 'center',
    x = 0.5
))

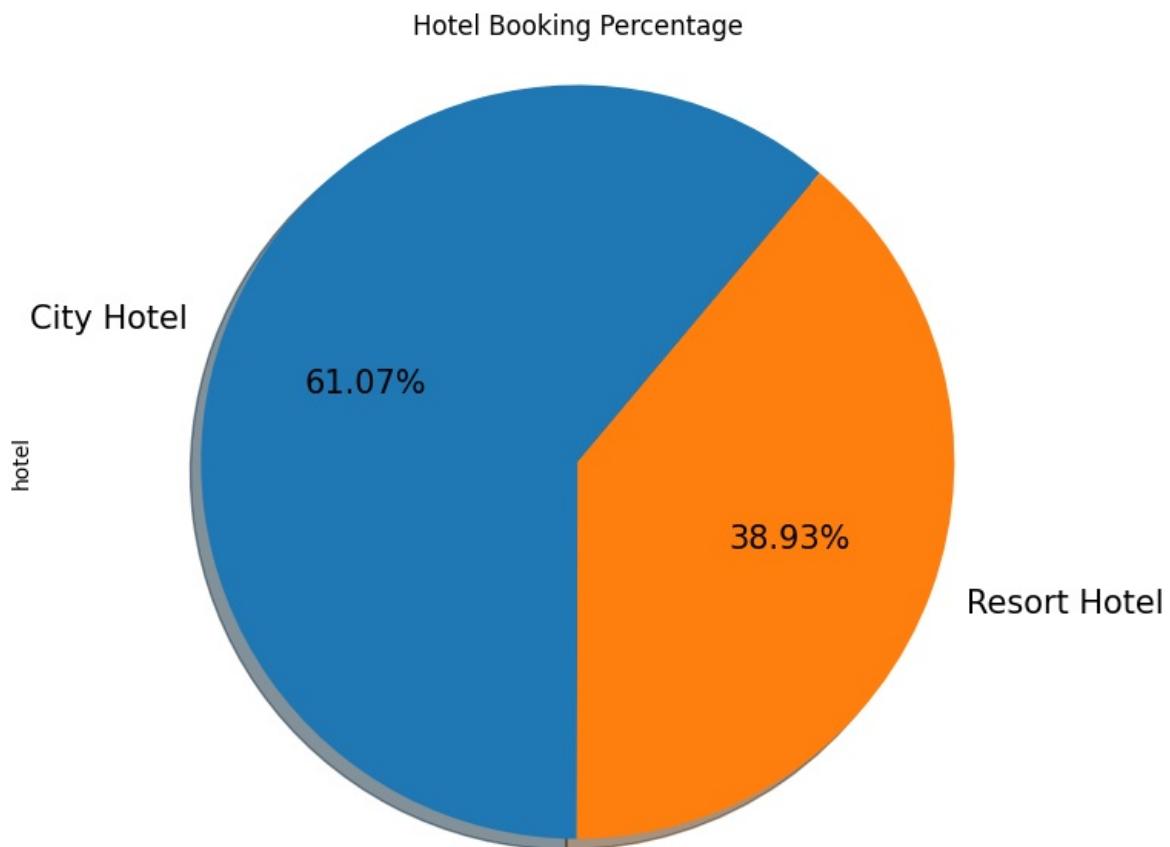
# Display the figure
fig1.show()
```

Creating a Pie chart also for the above problem statement as
Donut chart is not exported to github.

In [48]:

```
# Count Hotel
hotel_count = hb_df1.hotel.value_counts()

# Plotting Values in a simple pie chart
hotel_count.plot.pie(figsize=(9,7), autopct='%.2f%%', shadow=True, fontsize=15, startangle=50)
# Setting the title
plt.title('Hotel Booking Percentage')
plt.axis('equal')
plt.show()
```



I used Donut chart here because it is used to show the proportions of categorical data, with the size of each piece representing the proportion of each category.

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

I found out that guests prefer Resort Hotel most over City Hotel.

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

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

This insight is useful for the stakeholder to check which hotel is performing best and they can invest more capitals in that. There is no such negative growth but stakeholders can focus more on City Hotel to get more booking and increase the overall revenue.

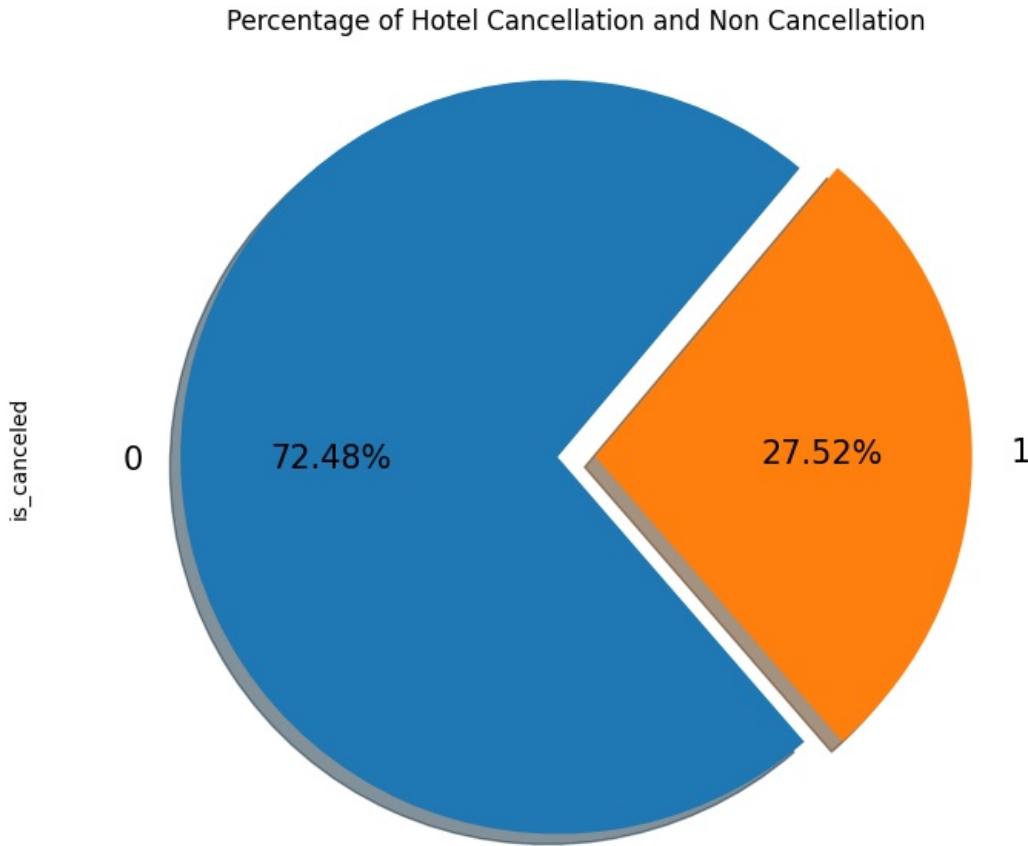
Chart - 2

What is percentage of hotel booking cancellation?

```
In [ ]: # Chart - 2 visualization code
# Extracting and storing unique values of hotel cancelation
cancelled_hotel = hb_df1.is_canceled.value_counts()

# Creating a pie chart
cancelled_hotel.plot.pie(figsize=(9,7), explode=(0.05,0.05), autopct='%1.2f%%', shadow=True, fontsize=15,startangle=90)

# Giving our pie chart a title
plt.title('Percentage of Hotel Cancellation and Non Cancellation')
plt.axis('equal')
plt.show()
```



1. Why did you pick the specific chart?

I had to show a part-to-a-whole relationship and percentage of both the values and here pie chart was a good option to show segmented values.

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

Here we can see that around 72.48% bookings are not canceled by guests but around 27.52% bookings are canceled by 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.

This insight will help stakeholders in comparing the cancellation and non cancellation of bookings. With the help of this insight stakeholders can offer rescheduling the bookings instead of cancellation and set a flexible cancellation policy to reduce booking cancellation.

Chart - 3

Which type of meal is most preffered by guests?

In []:

```
# Chart - 3 visualization code

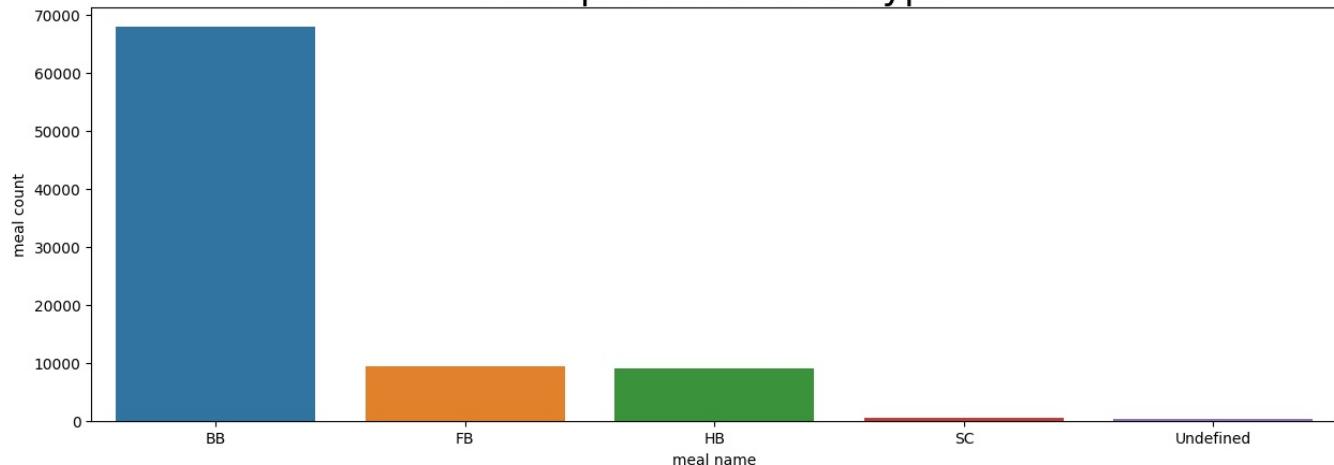
# Counting each meal type
meal_count = hb_df1.meal.value_counts()

# Extracting each meal type and storing in a variable
meal_name = hb_df1['meal'].unique()

# Creating a dataset of each meal type and count
meal_df = pd.DataFrame(zip(meal_name,meal_count), columns = ['meal name', 'meal count'])

# Visualising the values on a bar chart
plt.figure(figsize=(15,5))
g = sns.barplot(data=meal_df, x='meal name', y ='meal count')
g.set_xticklabels(meal_df['meal name'])
plt.title('Most preffered meal type', fontsize=25)
plt.show()
```

Most preffered meal type



Meal type variable description:

BB - (Bed and Breakfast)

HB- (Half Board)

FB- (Full Board)

SC- (Self Catering)

1. Why did you pick the specific chart?

There were 4 values to compare and Bar graphs are used to compare things between different groups that is why I used this chart.

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

After visualizing the above chart we can see that BB - (Bed and Breakfast) is the most preffered meal type by 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.

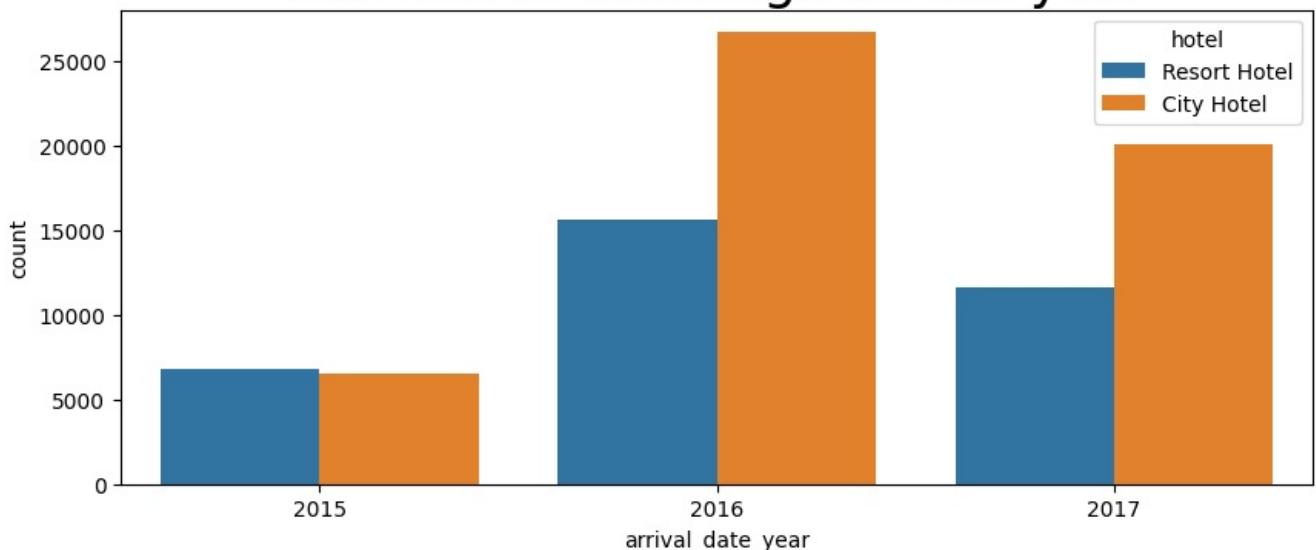
Yes, from the gained insight above now stakeholders know that **BB(Bed and Breakfast)** is most preferred meal type so they can arrange raw material for this meal in advance and deliver the meal without any delay.

Chart - 4

Which year has the most bookings ?

```
In [ ]: # Chart - 4 visualization code
# Plotting with countplot
plt.figure(figsize=(10,4))
sns.countplot(x=hb_df1['arrival_date_year'],hue=hb_df1['hotel'])
plt.title("Number of bookings across year", fontsize = 25)
plt.show()
```

Number of bookings across year



1. Why did you pick the specific chart?

Bar graphs are used to compare things between different groups that is why I used this chart.

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

From above insight I found out that hotel was booked most times in year 2016.

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

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

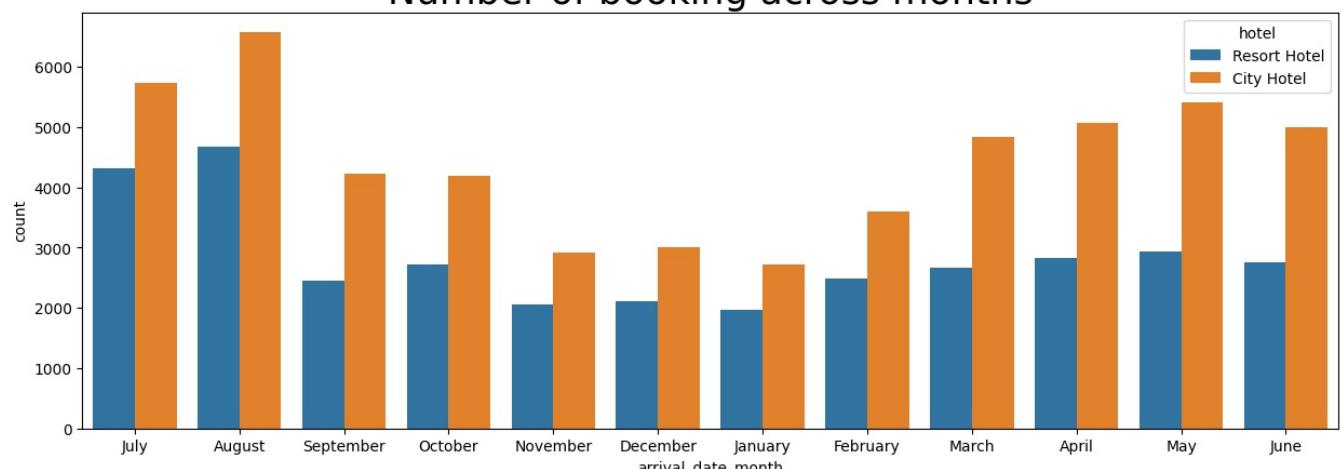
Above insight shows that number of booking was declined after year 2016. Stakeholders can now what went wrong after 2016 and fix that problem to increase the umber of bookings. One way to do this is ask for feedbacks from guests and have a meeting with old employees who else were serving int the year 2016.

Chart - 5

Which month has the most bookings in each hotel type?

```
In [ ]: # Chart - 5 visualization code
plt.figure(figsize=(15,5))
sns.countplot(x=hb_df1['arrival_date_month'],hue=hb_df1['hotel'])
plt.title("Number of booking across months", fontsize = 25)
plt.show()
```

Number of booking across months



1. Why did you pick the specific chart?

I had to compare values across the months and for that bar chart was one of the best choice.

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

Above insight shows that August and July were 2 most busy months in compare to others.

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

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

There is negative insight but hotel can use this insight to arrange everything in advance and welcome their guest in the best way possible and hotel can also run some promotional offer in these 2 months to attract more guests.

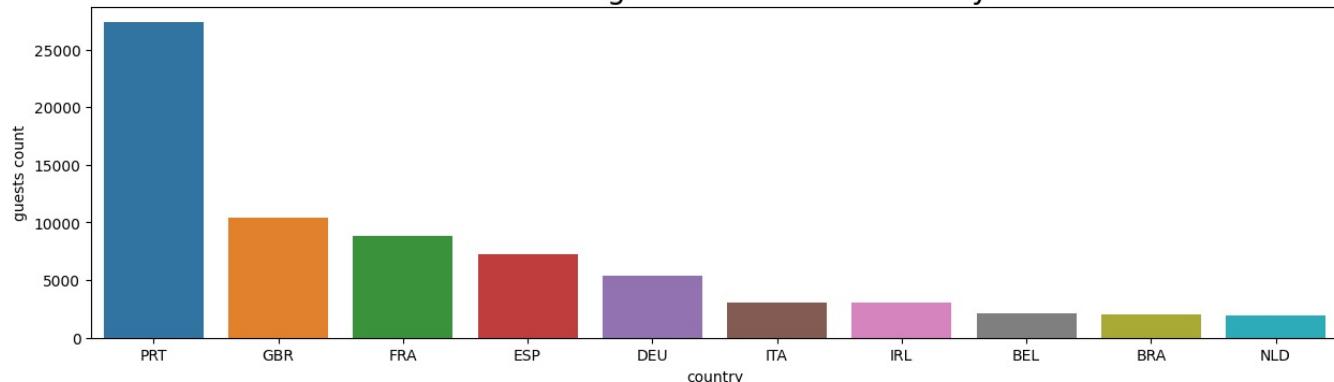
Chart - 6

From which country most guests come?

```
In [ ]: # Chart - 6 visualization code
# Counting number of guests from various countries and changing column names
country_df = hb_df1['country'].value_counts().reset_index().rename(columns={'index': 'country', 'country': 'guests count'})

# Visualising the values on a bar chart
# setting the graph size
plt.figure(figsize=(15,4))
sns.barplot(x=country_df['country'], y=country_df['guests count'])
plt.title('Number of guests from each country', fontsize=20)
plt.show()
```

Number of guests from each country



1. Why did you pick the specific chart?

Here I compared different values that's why I used bar chart.

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

From the above chart I found out that most guests come from PRT(Portugal).

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

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

There is no negative insight. After knowing that most of the guests come from Portugal Hotels can add more Portugal dishes in their menu to make guests order more food.

Chart - 7

Which distribution channel is most used in booking?

```
In [ ]: # Visualization code
# Creating a dataset of distribution channel name and count
dist_df = hb_df1['distribution_channel'].value_counts().reset_index()

# Renaming the columns to appropriate names
dist_df = dist_df.rename(columns={'index': 'Channel name', 'distribution_channel': 'channel count'})

# Creating a explode data
my_explode = (0.05,0.05,0.05,0.05,0.05)
```

```

#adding percentage columns to the distribution_channel_df
dist_df['percentage']=round(dist_df['channel count']*100/hb_df1.shape[0],1)

# Deciding the figure size
plt.figure(figsize=(15,6))

# Plotting the chart values
plt.pie(dist_df['channel count'], labels=None, explode = my_explode, startangle = 50)

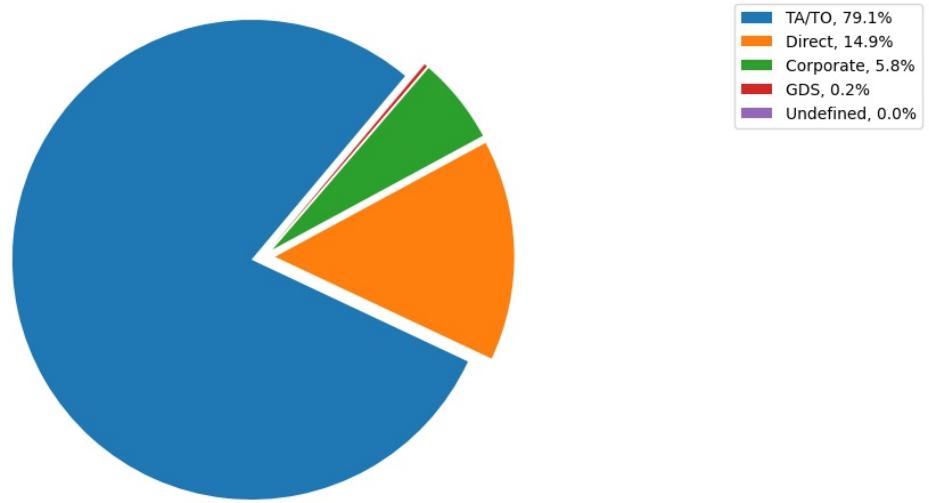
# Adding legends with percenatge using list comprehension
labels = [f'{l}, {s}%' for l, s in zip(dist_df['Channel name'].value_counts().index.tolist(), dist_df['percentage'])]
plt.legend(bbox_to_anchor=(0.85, 1), loc='upper left', labels=labels)

# Setting the chart title
plt.title('Most Used Booking Distribution Channels by Guests' ,fontsize = 16)

# Show the chart
plt.axis('equal')
plt.show()

```

Most Used Booking Distribution Channels by Guests



1. Why did you pick the specific chart?

Pie chart is one of the best chart to visualize categoriacal data.

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

From the above insight it is clear that TA/TO (travel agents/Tour operators) is most used distribution channel by 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.

There is no negative insight. Hotels can run promotional offers to motivate other channels to contribute more in bookings.

Chart - 8

Which room type is most preffered by guests?

```

In [ ]:
# Chart - 8 visualization code
# Setting the figure size
plt.figure(figsize=(15,5))

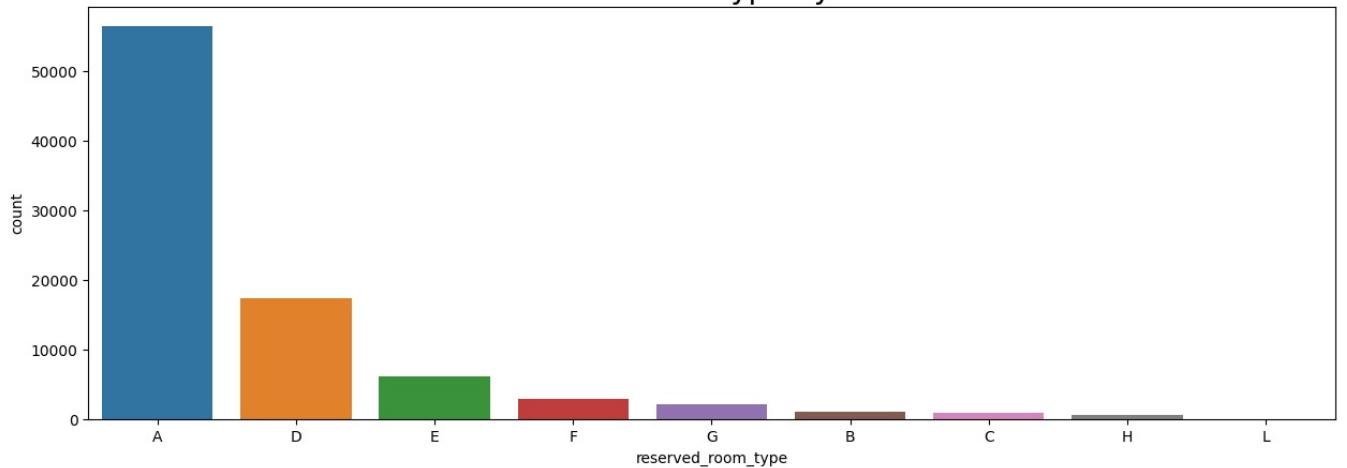
# Plotting the values in chart
sns.countplot(x=hb_df1['reserved_room_type'],order=hb_df1['reserved_room_type'].value_counts().index)

# Setting the title
plt.title('Preffered Room Type by Guests' , fontsize = 20)

# Show the chart
plt.show()

```

PREFERRED ROOM TYPE BY GUESTS



1. Why did you pick the specific chart?

A bar plot shows categorical data as rectangular bars with the height of bars proportional to the value they represent. It is often used to compare between values of different categories in the data.

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

By observing the above chart we can understand that the room type A most preferred (almost 55,000) by the guests while booking the hotel.

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

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

As it is clear that room type A is most used hotel should increase the number of A type room to maximize the revenue.

Chart - 9

Which room type is most assigned?

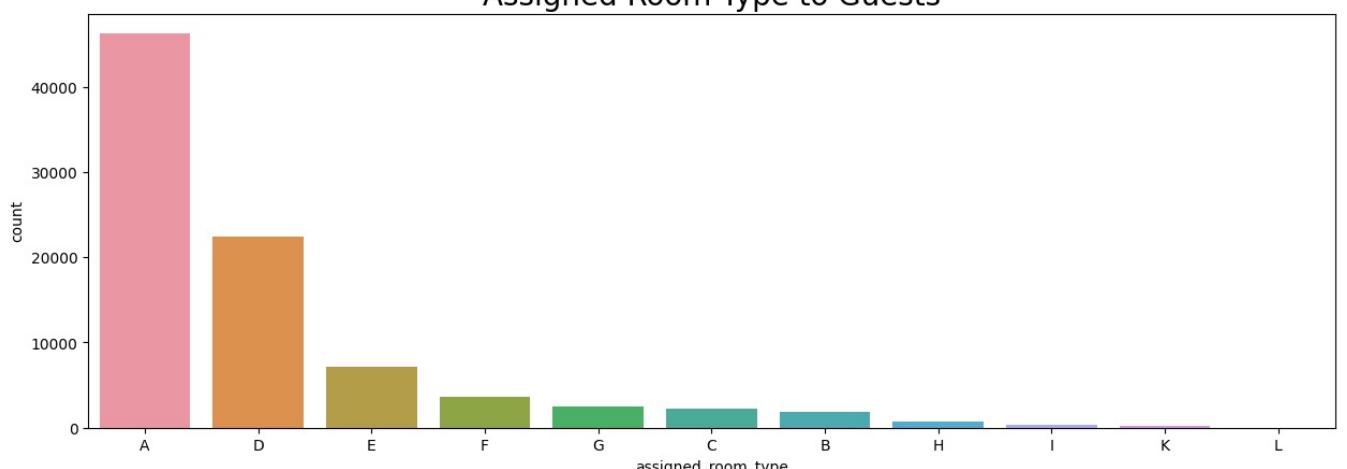
```
In [ ]: # Chart - 9 visualization code
# Setting the figure size
plt.figure(figsize=(15,5))

# Plotting the values
sns.countplot(x=hb_df1['assigned_room_type'], order = hb_df1['assigned_room_type'].value_counts().index)

# Setting the title
plt.title('Assigned Room Type to Guests', fontsize = 20)

# show the chart
plt.show()
```

Assigned Room Type to Guests



1. Why did you pick the specific chart?

A bar plot shows categorical data as rectangular bars with the height of bars proportional to the value they represent.

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

From the above chart it is clear that room type A is most assigned to 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.

In the 8th chart we saw that around 55,000 guests preferred room type A but 45,000 people were assigned A type room. This could be a reason to cancel the bookings. Hotel could increase A type room to decrease cancellation.

Chart - 10

Top 5 agents in terms of most bookings?

```
In [ ]: # Chart - 10 visualization code
# Creating a dataset by grouping by agent column and its count
agents = hb_df1.groupby(['agent'])['agent'].agg({'count'}).reset_index().rename(columns={'count':'Booking Count'})
        .sort_values(by = 'Booking Count')

# Extracting top 5 agents by booking count
top_5 = agents[:5]

# Explosion
explode = (0.02,0.02,0.02,0.02,0.02)

# Colors
colors = ( "orange", "cyan", "brown", "indigo", "beige")

# Wedge properties
wp = { 'linewidth' : 1, 'edgecolor' : "green" }

# Creating autopct arguments
def func(pct, allvalues):
    absolute = int(pct / 100.*np.sum(allvalues))
    return "{:.1f}%\n({:d} g)".format(pct, absolute)

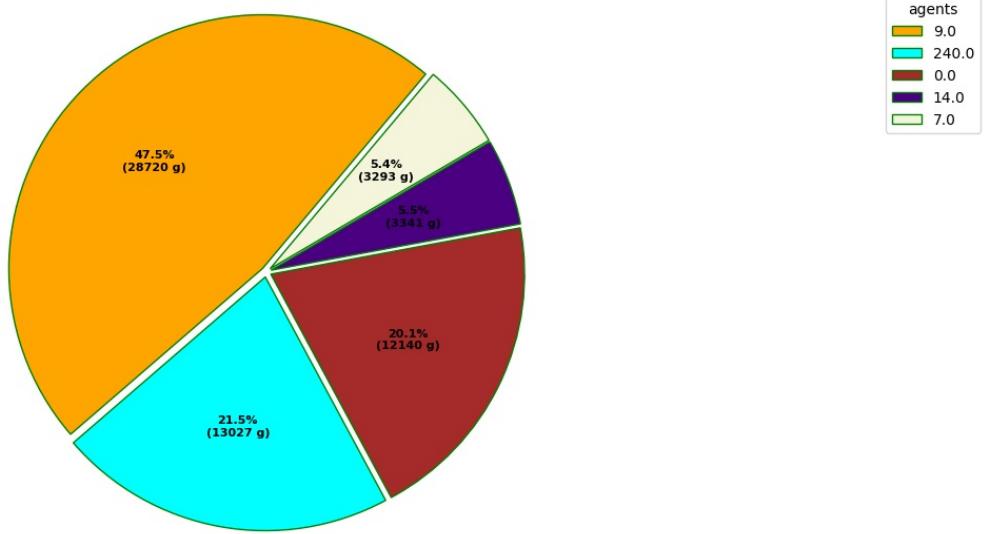
# Plotting the values
fig, ax = plt.subplots(figsize =(15, 7))
wedges, texts, autotexts = ax.pie(top_5['Booking Count'],
                                   autopct = lambda pct: func(pct, top_5['Booking Count']),
                                   explode = explode,
                                   shadow = False,
                                   colors = colors,
                                   startangle = 50,
                                   wedgeprops = wp)

# Adding legend
ax.legend(wedges, top_5['agent'],
          title ="agents",
          loc ="upper left",
          bbox_to_anchor =(1, 0, 0.5, 1))

plt.setp(autotexts, size = 8, weight ="bold")
ax.set_title("Top 5 agents in terms of booking", fontsize = 17)

# Show chart
plt.axis('equal')
plt.show()
```

Top 5 agents in terms of booking



1. Why did you pick the specific chart?

A pie chart helps organize and show data as a percentage of a whole

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

We can see that agent number 9 has made the most number of bookings followed by agent number 240, 0, 14 and 7.

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

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

Hotel can offer them bonus for their incredible work and to motivate them. This will help to increase the revenue.

Chart - 11

What is the percentage of repeated guests?

```
In [ ]: # Chart - 11 visualization code
# Creating a variable containing guests with their repeated counts
rep_guests = hb_df1['is_repeated_guest'].value_counts()

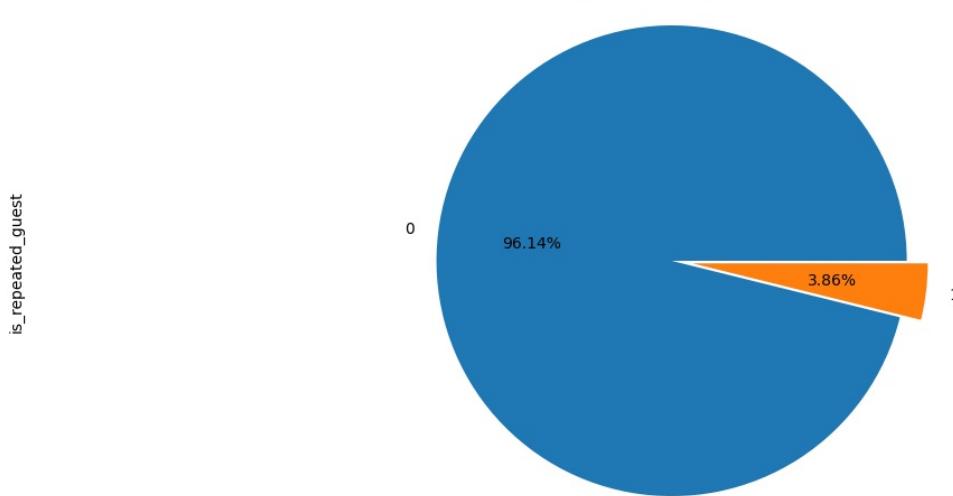
# Plotting the values in a pie chart
rep_guests.plot.pie(autopct='%.2f%%', explode=(0.00,0.09), figsize=(15,6), shadow=False)

# Setting the title
plt.title('Percentage of Repeated Guests', fontsize=20)

# Setting the chart in centre
plt.axis('equal')

# Show the chart
plt.show()
```

Percentage of Repeated Guests



1. Why did you pick the specific chart?

A pie chart helps organize and show data as a percentage of a whole

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

From the above insight we can see that 3.86% guests are repeated 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.

We can see that number of repeated guests is very low and it shows negative growth of the hotel. Hotel can offer loyalty discount to their guests to increase repeated guests.

Chart - 12

Which customer type has the most booking?

```
In [ ]: # Chart - 12 visualization code
cust_type = hb_df1['customer_type'].value_counts()

# Plotting the values in a line chart
cust_type.plot(figsize=(15,5))

# Setting the x label , y label and title
plt.xlabel('Count', fontsize=8)
plt.ylabel('Customer Type', fontsize=10)
plt.title('Customer Type and their booking count', fontsize=20)

# Show the chart
plt.show()
```



1. Why did you pick the specific chart?

Line graphs are used to track changes over different categories.

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

We can see that Transient customer type has most number of bookings.

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

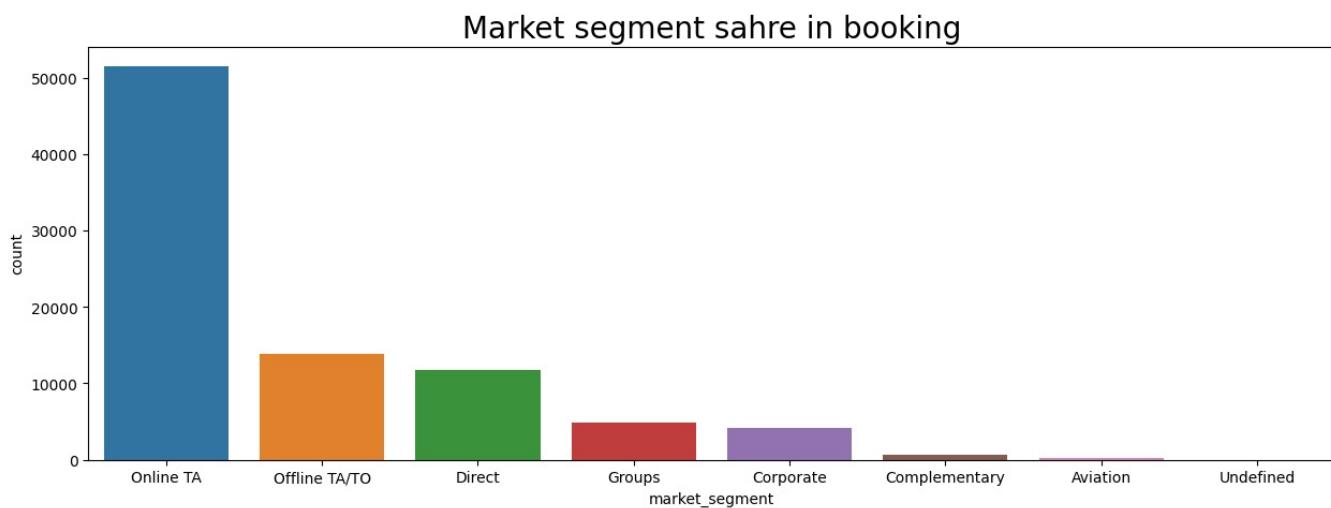
Are there any insights that lead to negative growth? Justify with specific reason.

Hotel can run promotional offers to increase the number of bookings over other categories. such as hotel could offer discounts for groups.

Chart - 13

Which Market Segment has the most booking?

```
In [ ]: # Chart - 13 visualization code
plt.figure(figsize=(15,5))
sns.countplot(x=hb_df1['market_segment'], order = hb_df1['market_segment'].value_counts().index)
plt.title('Market segment sahre in booking', fontsize=20)
plt.show()
```



1. Why did you pick the specific chart?

A bar plot shows categorical data as rectangular bars with the height of bars proportional to the value they represent.

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

Above insight shows that Online TA (Travel Agent) has the most bookings.

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

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

There is no negative growth. Hotel should come up with some great idea to increase sahre among other market segments to increase the revenue.

Chart -14

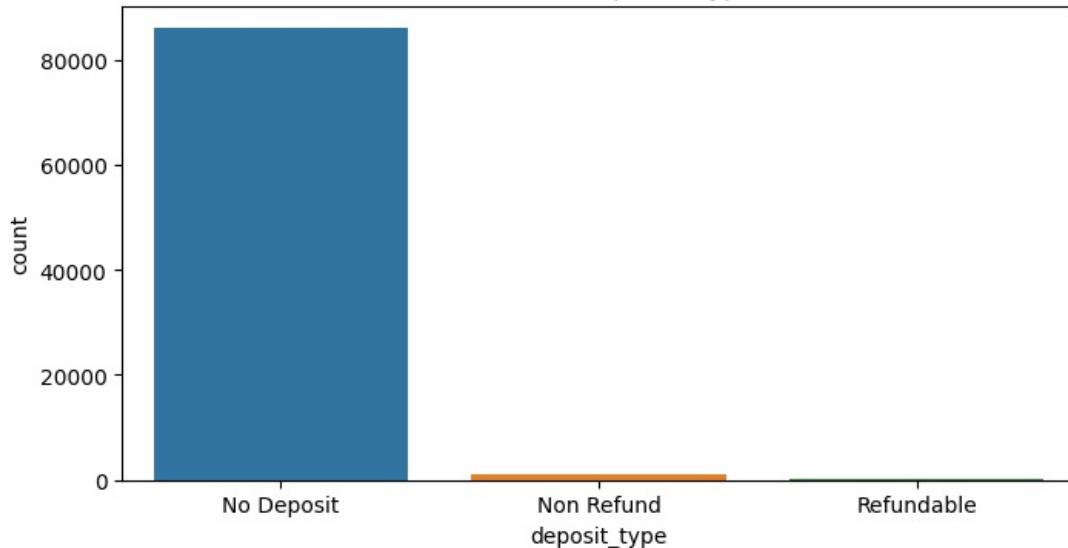
Which deposite type is most preffered?

```
In [ ]: # Visualization Code
# Counting each deposite type
deposite = hb_df1['deposit_type'].value_counts().index

# Setting the chart size
plt.figure(figsize=(8,4))

# plotting the values
sns.countplot(x=hb_df1['deposit_type'], order= deposite)
plt.title('Most used deposite type')
plt.show()
```

Most used deposite type



Bivariate and Multivariate Analysis

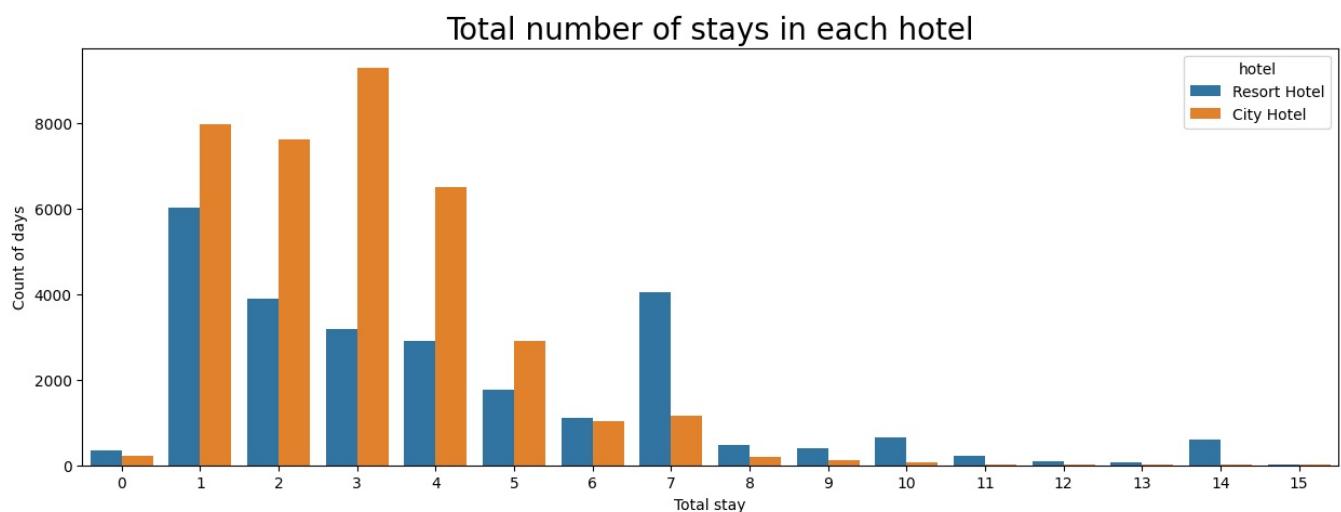
Chart - 15

How long people stay in the hotel?

```
In [ ]: # Chart - 11 visualization code
# Creating a not cancelled dataframe
not_cancelled_df = hb_df1[hb_df1['is_canceled'] == 0]
# Creating a hotel stay dataframe
hotel_stay = not_cancelled_df[not_cancelled_df['total_stay'] <= 15] #Visualizing pattern till 15days stay

# Setting plot size and plotting barchart
plt.figure(figsize = (15,5))
sns.countplot(x = hotel_stay['total_stay'], hue = hotel_stay['hotel'])

# Adding the label of the chart
plt.title('Total number of stays in each hotel', fontsize = 20)
plt.xlabel('Total stay')
plt.ylabel("Count of days")
plt.show()
```



From the above chart we can see that in City hotel most people stay for 3 days and in Resort hotel most people stay for only 1 day.

Hotel should work on to increase total stay in Resort hotel to increase revenue.

Chart-16

Which hotel makes most revenue?

```
In [ ]: # Counting the revenue for each hotel type using groupby function
most_rev = hb_df1.groupby('hotel')['adr'].count()
```

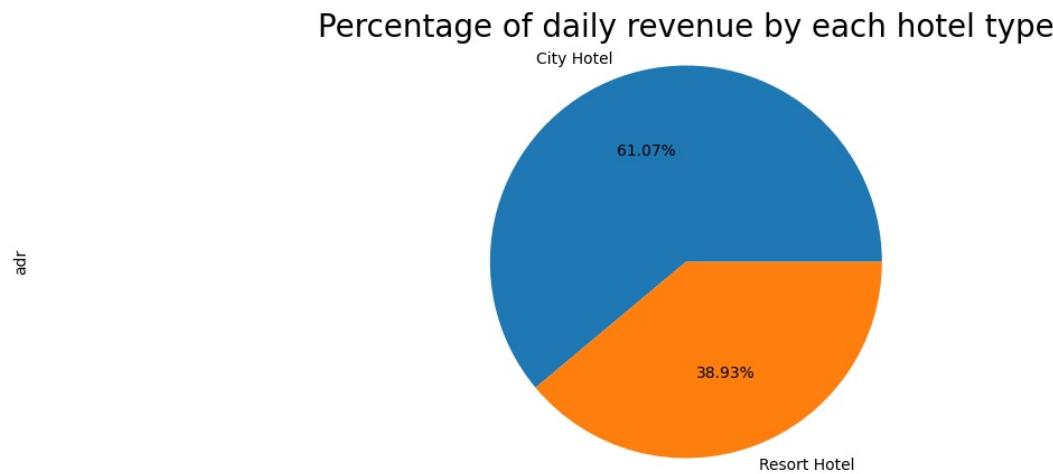
```

# Plotting the values in a pie chart
most_rev.plot.pie(autopct='%.1f%%', figsize=(15,5))

# Setting the title
plt.title('Percentage of daily revenue by each hotel type', fontsize=20)
plt.axis('equal')

# Show the chart
plt.show()

```



From the above insight it is clear that City hotel has more share in revenue generation over Resort Hotel.

Stake holders could improve the service of Resort hotel so that people stay more in resort hotel and increase the revenue.

Chart - 17

Which hotel has the longer waiting time?

```

In [ ]: # Grouping by hotel and taking the mean of days in waiting list
waiting_time_df = hb_df1.groupby('hotel')['days_in_waiting_list'].mean().reset_index()

# Waiting_time_df

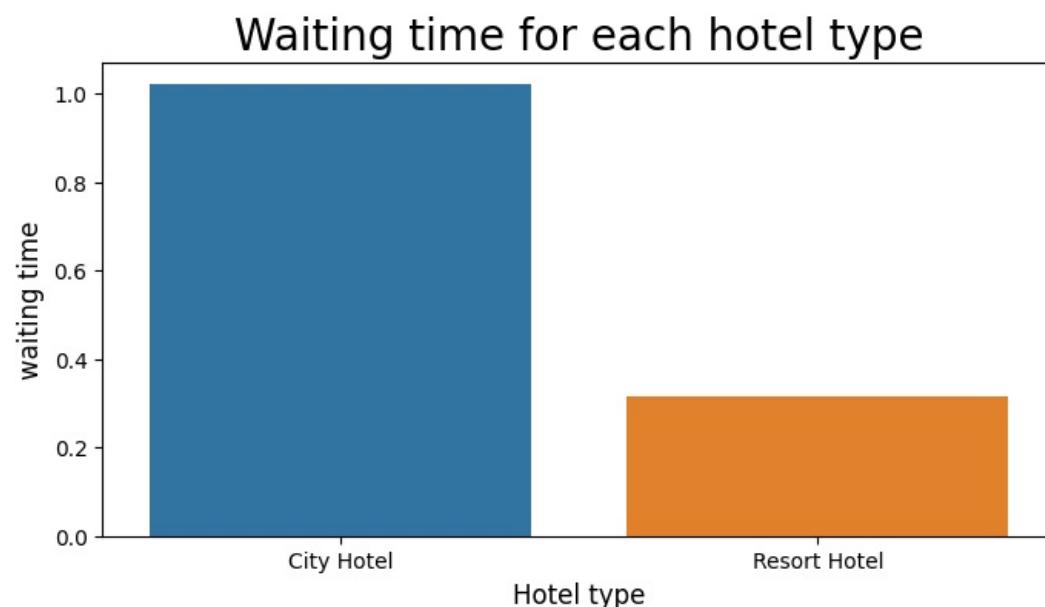
# Setting the plot size
plt.figure(figsize=(8,4))

# Plotting the barchart
sns.barplot(x=waiting_time_df['hotel'],y=waiting_time_df['days_in_waiting_list'])

# Setting the labels
plt.xlabel('Hotel type',fontsize=12)
plt.ylabel('waiting time',fontsize=12)
plt.title("Waiting time for each hotel type",fontsize=20)

# Show chart
plt.show()

```



Above chart shows that City hotel has more waiting period. This could be because people stay more in City hotel as we saw in previous

insight.

Stakeholders should increase rooms in City hotel or convert some of rooms of Resort hotel into City Hotel to decrease the waiting time.

Chart - 18

Hotel with most repeated guests.

```
In [ ]: # Grouping hotel types on repeated guests
rep_guest = hb_df1[hb_df1['is_repeated_guest']==1].groupby('hotel').size().reset_index()

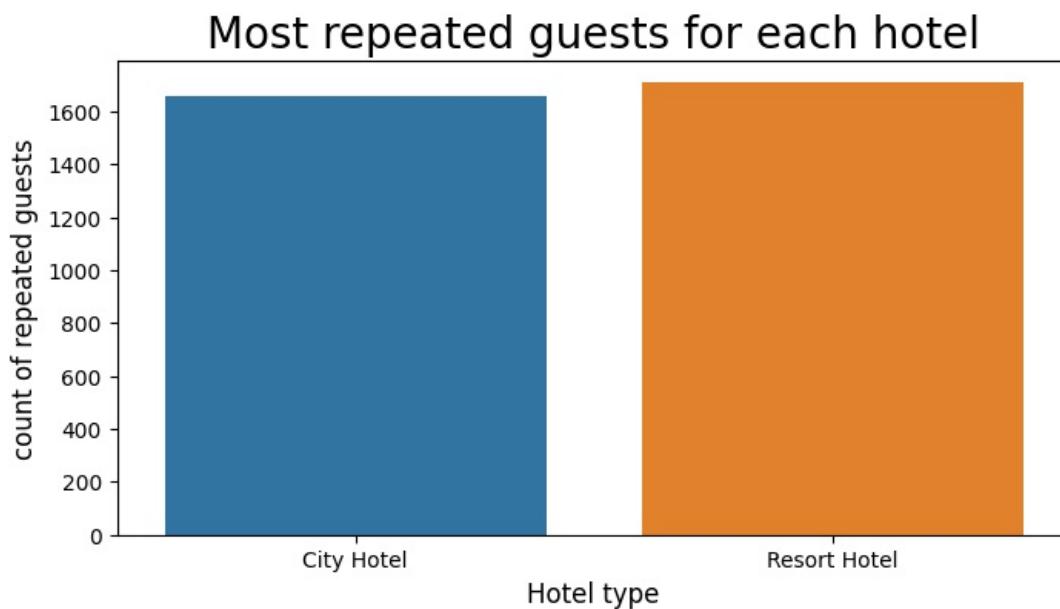
# Renaming the column
rep_guest = rep_guest.rename(columns={0:'number_of_repeated_guests'})

# Setting the chart size
plt.figure(figsize=(8,4))

# Plotting the values in a bar chart
sns.barplot(x=rep_guest['hotel'],y=rep_guest['number_of_repeated_guests'])

# Setting the labels and title
plt.xlabel('Hotel type', fontsize=12)
plt.ylabel('count of repeated guests', fontsize=12)
plt.title('Most repeated guests for each hotel', fontsize=20)

# Show Chart
plt.show()
```



We can see that Resort Hotel has slightly more repeated guests over City Hotel this could be because of less waiting time in Resort Hotel and better service there because of less rush.

Chart - 19

What is the adr across different months?

```
In [ ]: # Grouping arrival_month and hotel on mean of adr
bookings_months=hb_df1.groupby(['arrival_date_month','hotel'])['adr'].mean().reset_index()

# Creating a month list to order the months in ascending
months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']

# Creating a dataset of months, hotel and their adr
bookings_months['arrival_date_month']=pd.Categorical(bookings_months['arrival_date_month'],categories=months,ordered=True)

# Sorting the months
bookings_months=bookings_months.sort_values('arrival_date_month')
```

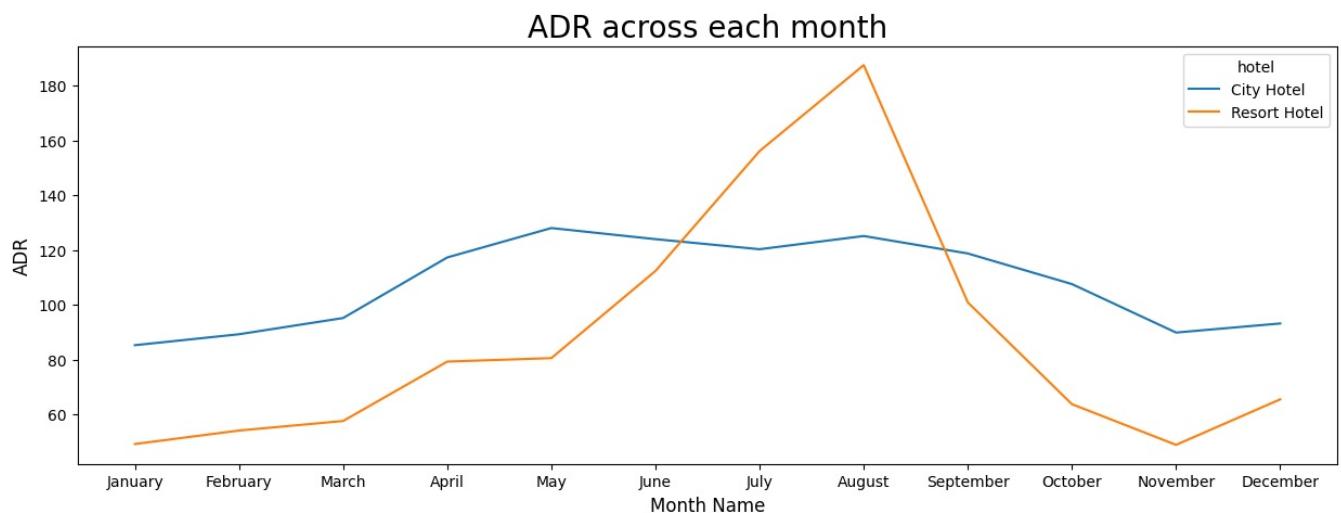
	arrival_date_month	hotel	adr
8	January	City Hotel	85.269875
9	January	Resort Hotel	49.181693
6	February	City Hotel	89.266427
7	February	Resort Hotel	54.102809
15	March	Resort Hotel	57.590889
14	March	City Hotel	95.193911
0	April	City Hotel	117.314134
1	April	Resort Hotel	79.283805
17	May	Resort Hotel	80.551101
16	May	City Hotel	128.055724
13	June	Resort Hotel	112.380859
12	June	City Hotel	123.996416
11	July	Resort Hotel	156.166914
10	July	City Hotel	120.318314
3	August	Resort Hotel	187.566659
2	August	City Hotel	125.148662
22	September	City Hotel	118.764693
23	September	Resort Hotel	100.892331
20	October	City Hotel	107.585401
21	October	Resort Hotel	63.723065
18	November	City Hotel	89.882912
19	November	Resort Hotel	48.871043
5	December	Resort Hotel	65.488671
4	December	City Hotel	93.204767

```
In [ ]: # Setting the chart size
plt.figure(figsize=(15,5))

# Plotting the values in a line chart
sns.lineplot(x=bookings_months['arrival_date_month'],y=bookings_months['adr'],hue=bookings_months['hotel'])

# Setting the labels and title
plt.title('ADR across each month', fontsize=20)
plt.xlabel('Month Name', fontsize=12)
plt.ylabel('ADR', fontsize=12)

# Show chart
plt.show()
```



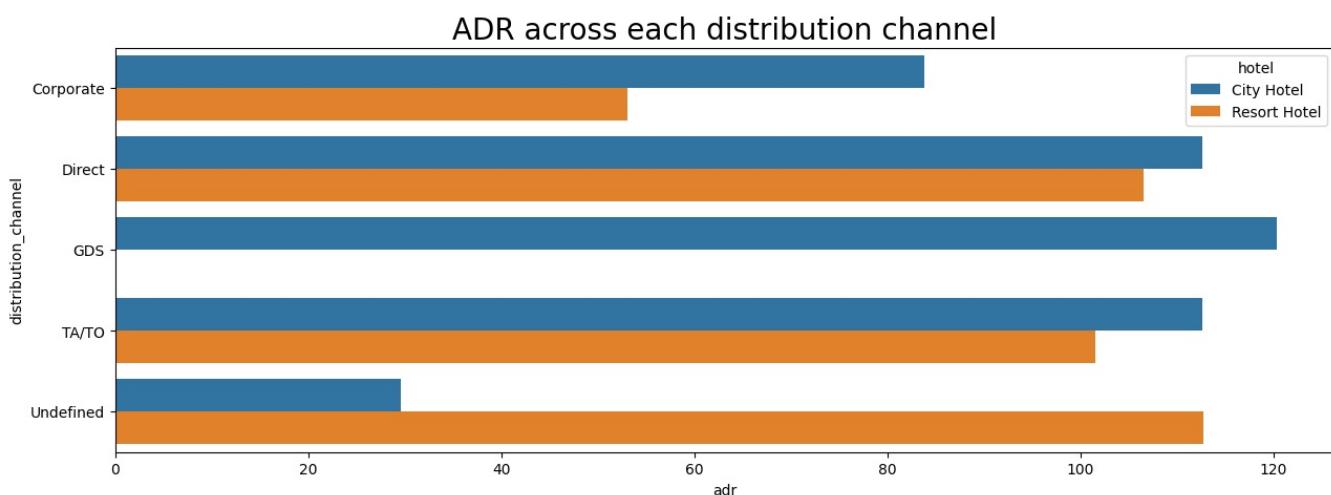
City Hotel : It is clear that City Hotel generates more revenue in May months in comparison to other months.

Resort Hotel : Resort Hotel generates more revenue in between July and August months.

Stakeholders could prepare in advance for these 2 months as these 2 months generate more revenue.

Which distribution channel has highest adr?

```
In [ ]: # Grouping dist_channel and hotels on their adr  
dist_channel_adr = hb_df1.groupby(['distribution_channel','hotel'])['adr'].mean().reset_index()  
  
# Setting the figure size  
plt.figure(figsize=(15,5))  
  
# Creating a horizontal bar chart  
sns.barplot(x='adr', y='distribution_channel', data=dist_channel_adr, hue='hotel')  
  
# Setting the title  
plt.title('ADR across each distribution channel', fontsize=20)  
  
# Show chart  
plt.show()
```



GDS has contributed more in generating the ADR. GDS is a worldwide conduit between travel bookers and suppliers, such as hotels and other accommodation providers. It communicates live product, price and availability data to travel agents and online booking engines, and allows for automated transactions.

Direct- means that bookings are directly made with the respective hotels

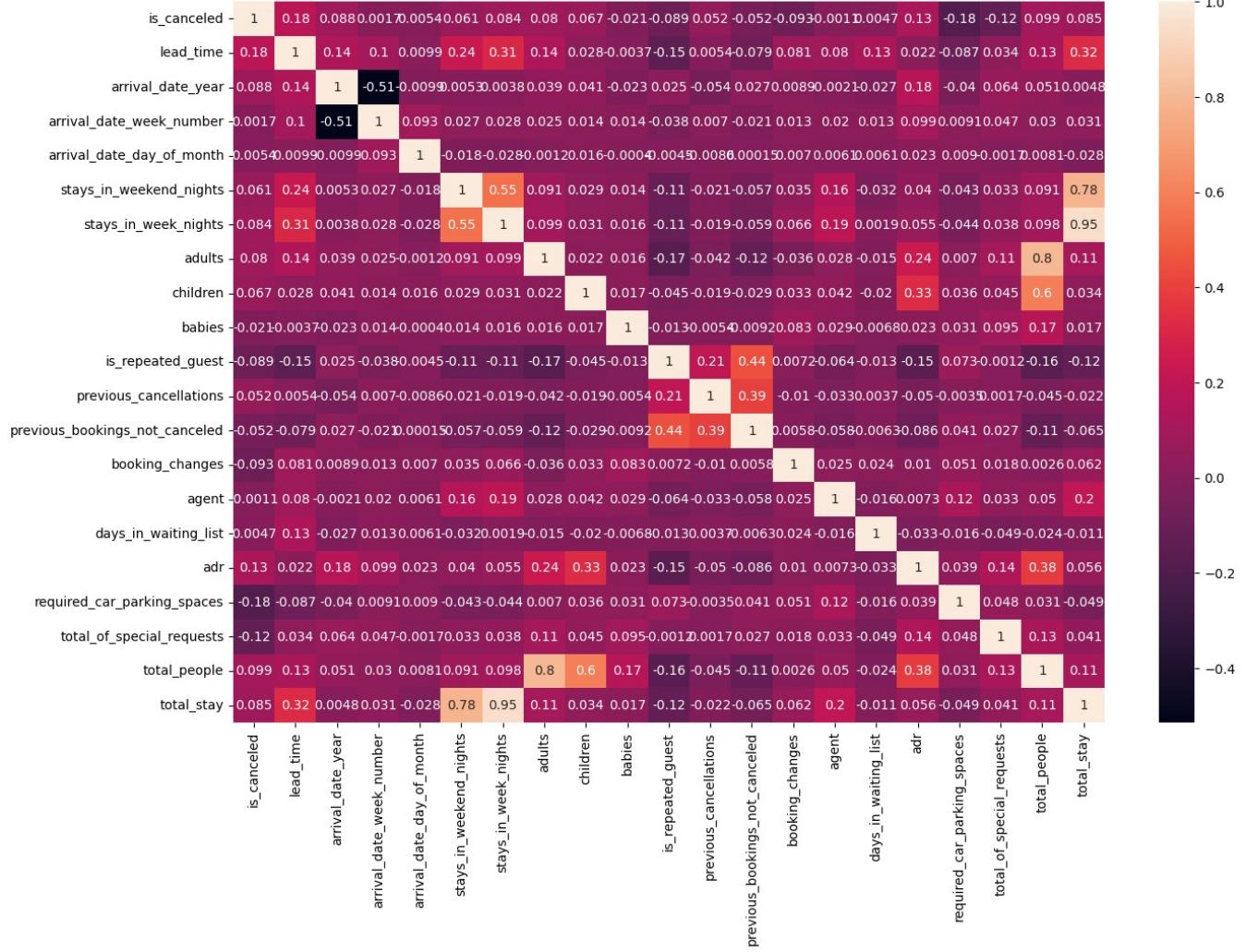
TA/TO- means that bookings are made through travel agents or travel operators.

Undefined- Bookings are undefined. may be customers made their bookings on arrival.

Chart - 21 - Correlation Heatmap

```
In [ ]: # Correlation Heatmap visualization code  
# Setting the chart size  
plt.figure(figsize=(15,10))  
  
# Creating heatmap to see correlation of each columns  
sns.heatmap(hb_df1.corr(numeric_only=True), annot=True) # Setting the numeric only column to True to avoid  
  
# Setting the title  
plt.title('Correlation of the columns', fontsize=20)  
  
# Show heatmap  
plt.show()
```

Correlation of the columns



1. Why did you pick the specific chart?

Correlation heatmaps was used to find potential relationships between variables and to understand the strength of these relationships.

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

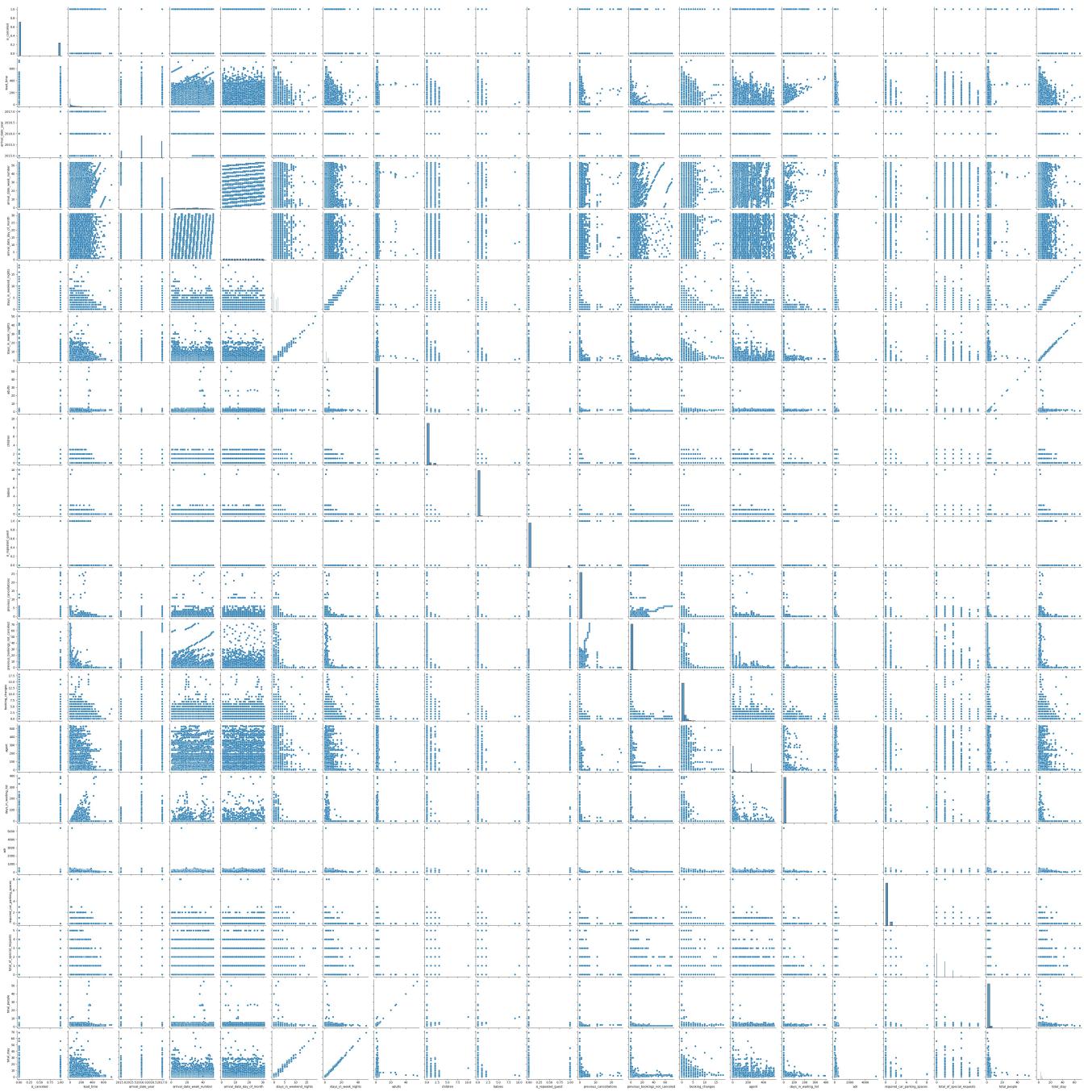
1) lead_time and total_stay is positively corelated. that means if customers stay more then the lead time increases.

2)adults,childrens and babies are corelated to each other. That means more the people more will be adr.

3) is_repeated guest and previous bookings not canceled has strong corelation. That means repeated guests don't cancel their bookings.**

Chart - 22 - Pair Plot

```
In [ ]: # Pair Plot visualization code
sns.pairplot(hb_df1)
plt.show()
```



1. Why did you pick the specific chart?

A pairs plot allows us to see both distribution of single variables and relationships between two variables .

We can see the realtionship between all the columns with each other in above chart.

1. From the above pair plot we can see that if cancellation increases then total stay also decreases.
2. As the total number of people increases adr also increases. Thus adr and total people are directly proportional to each other.

5. Solution to Business Objective

What do you suggest the client to achieve Business Objective ?

Explain Briefly.

1. Resort Hotel is most preffered so Stakeholders can offer discounts on City Hotel to increase bookings.
2. Around 27.52% of bookings are cancelled so Hotel can offer layality discount if guests don't cnacel their booking. 3.Hotel can maintain raw materials for BB type meal in advance to avoid delay as BB(Bead and Breakfast) is the most preffered meal.
3. Hotel should increase number of rooms in City Hotels to decrease the waiting time.
4. TA has the most number of bookings over other MArket segments so Hotel could run some offer to get more bookings from otehr segment.
5. Room type A is most preffered by guests so Hotel should increase the number of A type room.
6. Number of repeated guests is low that indicates that there is something they don't like about Hotel and that needs to be fixed to increase number of repeated guests. 8.Waiting time period for City hotel is high as compared to resort hotels. That means city hotels

are much busier than Resort hotels.

7. Optimal stay in both the type hotel is less than 7 days. Usually people stay for a week so Hotel need to take some actions to improve their performance.
8. Maximum number of guests were from Portugal.

Conclusion

Inorder to achieve the business objective, i would suggest the client to make the price dynamic, introduce offers and packages to attract new customers. To retain the existing customers and ensure their repetition the client must introduce loyalty points program which can be redeemed by the customers in their next bookings. Amenities such as parking spaces, kids corner, free internet connection can be provided to increase the number of bookings.

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