HOTEL
BOOKING
HOTEL ONLINE
★★★★★
BOOK NOW

Project Name - Hotel Booking Analysis

Project Type - Exploratory Data Analysis

Contribution - Rahul Kumar Bala

Project Summary -

- This project is related to Hotel Booking having two hotel description i.e City Hotel and Resort Hotel. In this dataset contains total rows 119390 and 32 columns.In this we divide data manipulation workflow in three category Data Collection ,Data cleaning and manipulation and EDA(Exploratory Data Analysis).As Further moved i.e Data collections first step to find different columns which is done by coding Head(), tail(), info(), describe(), columns() and some others method used for data collections, some of the columns name is updated here i.e hotel,is_canceled,lead_time,arrival_date_year,arrival_date_month,arrival_date_week_number,arrival_date_day_of_month,stays_in_weekend_nights.As we further moved we find unique value of each columns and generate a list in tabular form and also check the dataset type of each columns' find some columns not in accurate data types which correct it later done in Data cleaning part and as well as duplicates data items must be removed as we find duplicates items equal to 87396 which is dropped from dataset later.
- Before visualize any data from the data set we have to do data wrangling. For that, we are checked the null value of all the columns. After checking, when we are getting a column which has more number of null values, dropped that column by using the 'drop' method. In this way, we are dropped the 'company' column. When we are find minimal number of null values, filling thse null values with necesary values as per requirement by using .fillna()
- Different charts are used for data visualization so that better insights and Business objective is attained.

Define Your Business Objective?

- Analyse the data on bookings of City Hotel and Resort Hotel to gain insights on the different factors that affect the booking. This is undertaken as an individual project.

GitHub Link -

Github Link - https://github.com/rahulkumarbala/Hotel-Bookng-EDA

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.

Explore and analyze the data to discover important factors that govern the bookings.

Let's Begin !

1. Know Your Data

Firstly we will import all the imortant libraries which helps us in our Analysis process

---

Import Libraries

```
import pandas as pd
import numpy as np
from datetime import datetime
from datetime import date
from datetime import timedelta
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Now we will mount our google drive and import the data into a variable from CSV file.

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

```
    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Dataset Loading

```
# Load Dataset
hotel_data_df = pd.read_csv('/content/drive/MyDrive/EDA/Hotel Booking Analysis - Rahul Kumar Bala/Hotel Bookings.csv')
```

Now we will check whether our data is loaded successfull or not and then we will do some basic analysis of our data

---

Dataset First View

```
# Dataset First Look
hotel_data_df
```

	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	...	deposit_
0	Resort Hotel	0	342	2015	July	27	1	0	0	2	...	No D
1	Resort Hotel	0	737	2015	July	27	1	0	0	2	...	No D
2	Resort Hotel	0	7	2015	July	27	1	0	1	1	...	No D
3	Resort Hotel	0	13	2015	July	27	1	0	1	1	...	No D
4	Resort Hotel	0	14	2015	July	27	1	0	2	2	...	No D
...	...	...	...	...	...	...	...	...	...	...	...	
119385	City Hotel	0	23	2017	August	35	30	2	5	2	...	No D
119386	City Hotel	0	102	2017	August	35	31	2	5	3	...	No D
119387	City Hotel	0	34	2017	August	35	31	2	5	2	...	No D
119388	City Hotel	0	109	2017	August	35	31	2	5	2	...	No D
119389	City Hotel	0	205	2017	August	35	29	2	7	2	...	No D

119390 rows × 32 columns

Dataset Rows & Columns count

```
# Dataset Rows & Columns count
print(hotel_data_df.index)
print('\n')
print(hotel_data_df.columns)
```

```
RangeIndex(start=0, stop=119390, step=1)


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 Information

```
# Dataset Info
hotel_data_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

```
# Dataset Duplicate Value Count, to remove these values, we use function drop.duplicate to delete duplicate rows.
hotel_data_df.drop_duplicates(inplace = True)

# total rows = 119390, Duplicate Rows = 31994
uni_num_of_rows = hotel_data_df.shape[0]
uni_num_of_rows # now unique rows = 87396
```

```
87396
```

```
# View unique data
hotel_data_df.reset_index()
```

	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	...	deposit_t
0	0	Resort Hotel	0	342	2015	July	27	1	0	0	...	No De
1	1	Resort Hotel	0	737	2015	July	27	1	0	0	...	No De
2	2	Resort Hotel	0	7	2015	July	27	1	0	1	...	No De
3	3	Resort Hotel	0	13	2015	July	27	1	0	1	...	No De
4	4	Resort Hotel	0	14	2015	July	27	1	0	2	...	No De
...	...	...	...	...	...	...	...	...	...	...	...	
87391	119385	City Hotel	0	23	2017	August	35	30	2	5	...	No De
87392	119386	City Hotel	0	102	2017	August	35	31	2	5	...	No De
87393	119387	City Hotel	0	34	2017	August	35	31	2	5	...	No De
87394	119388	City Hotel	0	109	2017	August	35	31	2	5	...	No De
87395	119389	City Hotel	0	205	2017	August	35	29	2	7	...	No De

87396 rows × 33 columns

Cleaning the data and Handling the null values.

Missing Values/Null Values

```
# Missing Values/Null Values Count
null_value = hotel_data_df.isnull() == True
hotel_data_df.fillna(np.nan, inplace = True)

hotel_data_df # we replace all the null value as NaN.
```

	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	...	deposit_
0	Resort Hotel	0	342	2015	July	27	1	0	0	2	...	No De
1	Resort Hotel	0	737	2015	July	27	1	0	0	2	...	No De
2	Resort Hotel	0	7	2015	July	27	1	0	1	1	...	No De
3	Resort Hotel	0	13	2015	July	27	1	0	1	1	...	No De
4	Resort Hotel	0	14	2015	July	27	1	0	2	2	...	No De
...	...	...	...	...	...	...	...	...	...	...	...	
119385	City Hotel	0	23	2017	August	35	30	2	5	2	...	No De
119386	City Hotel	0	102	2017	August	35	31	2	5	3	...	No De
119387	City Hotel	0	34	2017	August	35	31	2	5	2	...	No De
119388	City Hotel	0	109	2017	August	35	31	2	5	2	...	No De
119389	City Hotel	0	205	2017	August	35	29	2	7	2	...	No De

87396 rows × 32 columns

```
# Visualizing the missing values
miss_values =hotel_data_df.isnull().sum().sort_values(ascending=False)
miss_values # We have check the count of null value in individual columns
```

```
company                           82137
agent                             12193
country                             452
children                              4
reserved_room_type                    0
assigned_room_type                    0
booking_changes                       0
deposit_type                          0
hotel                                 0
previous_cancellations                0
days_in_waiting_list                  0
customer_type                         0
adr                                   0
required_car_parking_spaces           0
total_of_special_requests             0
reservation_status                    0
previous_bookings_not_canceled        0
is_repeated_guest                     0
is_canceled                           0
distribution_channel                  0
market_segment                        0
```

```
meal                              0
babies                            0
adults                            0
stays_in_week_nights              0
stays_in_weekend_nights           0
arrival_date_day_of_month         0
arrival_date_week_number          0
arrival_date_month                0
arrival_date_year                 0
lead_time                         0
reservation_status_date           0
dtype: int64
```

What did you know about your dataset?

This data set contains a single file which compares various booking information between two hotels: a city hotel and a resort hotel.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. The dataset contains a total of 119390 rows and 32 columns.Dataset Contains duplicated items i.e 31944 which is removed later .In this dataset we find data types of every columns i.e (Int, float ,string) and observe that some columns data types is not accurate and remove later .We find unique value of every columns it means what actual values in every columns

2. Understanding Your Variables

let's get all columns

```
# Dataset Columns
df_column = hotel_data_df.columns
df_column
```

```
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')
```

Let's describe data for insights

```
# Dataset Describe
hotel_data_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	adults	children	babies	is_re
count	87396.000000	87396.000000	87396.000000	87396.000000	87396.000000	87396.000000	87396.000000	87396.000000	87392.000000	87396.000000	
mean	0.274898	79.891368	2016.210296	26.838334	15.815541	1.005263	2.625395	1.875795	0.138640	0.010824	
std	0.446466	86.052325	0.686102	13.674572	8.835146	1.031921	2.053584	0.626500	0.455881	0.113597	
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	11.000000	2016.000000	16.000000	8.000000	0.000000	1.000000	2.000000	0.000000	0.000000	
50%	0.000000	49.000000	2016.000000	27.000000	16.000000	1.000000	2.000000	2.000000	0.000000	0.000000	
75%	1.000000	125.000000	2017.000000	37.000000	23.000000	2.000000	4.000000	2.000000	0.000000	0.000000	
max	1.000000	737.000000	2017.000000	53.000000	31.000000	19.000000	50.000000	55.000000	10.000000	10.000000	

Variables Description

Description of individual Variable

The columns and the data it represents are listed below:

1. **hotel :** Name of the hotel (Resort Hotel or City Hotel)
2. **is_canceled :** If the booking was canceled (1) or not (0)
3. **lead_time:** Number of days before the actual arrival of the guests
4. **arrival_date_year :** Year of arrival date
5. **arrival_date_month :** Month of month arrival date
6. **arrival_date_week_number :** Week number of year for arrival date
7. **arrival_date_day_of_month :** Day of arrival date
8. **stays_in_weekend_nights :** Number of weekend nights (Saturday or Sunday) spent at the hotel by the guests.
9. **stays_in_week_nights :** Number of weeknights (Monday to Friday) spent at the hotel by the guests.
10. **adults :** Number of adults among guests
11. **children :** Number of children among guests
12. **babies :** Number of babies among guests
13. **meal :** Type of meal booked
14. **country :** Country of guests
15. **market_segment :** Designation of market segment
16. **distribution_channel :** Name of booking distribution channel
17. **is_repeated_guest :** If the booking was from a repeated guest (1) or not (0)
18. **previous_cancellations :** Number of previous bookings that were cancelled by the customer prior to the current booking

19. **previous_bookings_not_canceled :** Number of previous bookings not cancelled by the customer prior to the current booking
20. **reserved_room_type :** Code of room type reserved
21. **assigned_room_type :** Code of room type assigned
22. **booking_changes :** Number of changes/amendments made to the booking
23. **deposit_type :** Type of the deposit made by the guest
24. **agent :** ID of travel agent who made the booking
25. **company :** ID of the company that made the booking
26. **days_in_waiting_list :** Number of days the booking was in the waiting list
27. **customer_type :** Type of customer, assuming one of four categories
28. **adr :** Average Daily Rate, as defined by dividing the sum of all lodging transactions by the total number of staying nights
29. **required_car_parking_spaces :** Number of car parking spaces required by the customer
30. **total_of_special_requests :** Number of special requests made by the customer
31. **reservation_status :** Reservation status (Canceled, Check-Out or No-Show)
32. **reservation_status_date :** Date at which the last reservation status was updated

Check Unique Values for each variable.

```
# Check Unique Values for each variable.
print(hotel_data_df.apply(lambda col: col.unique())) # We have describes unique value in all individual column.
```

```
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/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_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
```

3. ***Data Wrangling***

Data Cleaning

```
#to fill the NaN value in the column, let's check which colomns has null value, we have already stored the same.
miss_values[:4]
```

```
company     82137
agent       12193
country       452
children        4
dtype: int64
```

```
#lets check, what is the percentage of null value in each column, starting from company

percentage_company_null = miss_values[0] / uni_num_of_rows*100
percentage_company_null
```

```
93.98256213098998
```

```
# It is better to drop the column 'company' altogether since the number of missing values is extremely high compared to the number of rows.

hotel_data_df.drop(['company'], axis=1, inplace=True)
```

```
# now let's check for agent

percentage_agent_null = miss_values[1] / uni_num_of_rows*100
percentage_agent_null
```

```
13.951439425145315
```

```
# As we have seen, there is minimul null values in agent, Lets fill these value by taking mode of the all values

hotel_data_df['agent'].fillna(value = 0, inplace = True)
hotel_data_df['agent'].isnull().sum() # we re-check that column has no null value
```

```
0
```

```
#Check the percentage null value in country col

percentage_country_null = miss_values[2] / uni_num_of_rows*100
percentage_country_null
```

```
0.5171861412421621
```

```
# We have less null vlues in country col, so we will replace null from 'other' as country name.

hotel_data_df['country'].fillna(value = 'others', inplace = True)
hotel_data_df['country'].isnull().sum() # we re-check that column has no null value
```

```
0
```

```
#Check the percentage null value in children col

percentage_children_null = miss_values[3] / uni_num_of_rows*100
percentage_children_null
```

```
0.004576868506567806
```

```
# We have less null vlues in country col, so we will replace null from 0 as country name.

hotel_data_df['children'].fillna(value = 0, inplace = True)
hotel_data_df['children'].isnull().sum() # we re-check that column has no null value
```

```
0
```

```
#let's check whether database having any other null value

hotel_data_df.isnull().sum() # As we have seen, no column has any null value
```

```
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
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
```

Change in datatype for required columns

```
#showing the info of the data to check datatype
hotel_data_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 87396 entries, 0 to 119389
Data columns (total 31 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  days_in_waiting_list            87396 non-null  int64
 25  customer_type                   87396 non-null  object
 26  adr                             87396 non-null  float64
 27  required_car_parking_spaces     87396 non-null  int64
 28  total_of_special_requests       87396 non-null  int64
 29  reservation_status              87396 non-null  object
 30  reservation_status_date         87396 non-null  object
dtypes: float64(3), int64(16), object(12)
memory usage: 21.3+ MB
```

```
# We have seen that childern & agent column as datatype as float whereas it contains only int value, lets change datatype as 'int64'
hotel_data_df[['children', 'agent']] = hotel_data_df[['children', 'agent']].astype('int64')
```

Addition of new column as per requirement

```
#total stay in nights
hotel_data_df['total_stay_in_nights'] = hotel_data_df ['stays_in_week_nights'] + hotel_data_df ['stays_in_weekend_nights']
```

```
hotel_data_df['total_stay_in_nights'] # We have created a col for total stays in nights by adding week night & weekend nights stay col.
```

```
0         0
1         0
2         1
3         1
4         2
         ..
119385    7
119386    7
119387    7
119388    7
119389    9
Name: total_stay_in_nights, Length: 87396, dtype: int64
```

```
# We have created a col for revenue using total stay * adr
hotel_data_df['revenue'] = hotel_data_df['total_stay_in_nights'] *hotel_data_df['adr']
hotel_data_df['revenue']
```

```
0            0.00
1            0.00
2           75.00
3           75.00
4          196.00
            ...
119385     672.98
119386    1578.01
119387    1103.97
119388     730.80
119389    1360.80
Name: revenue, Length: 87396, dtype: float64
```

```
# Also, for information, we will add a column with total guest coming for each booking
hotel_data_df['total_guest'] = hotel_data_df['adults'] + hotel_data_df['children'] + hotel_data_df['babies']
hotel_data_df['total_guest'].sum()
```

```
176999
```

```
# for understanding, from col 'is_canceled': we will replace the value from (0,1) to not_canceled, is canceled.

hotel_data_df['is_canceled'] = hotel_data_df['is_canceled'].replace([0,1], ['not canceled', 'is canceled'])
hotel_data_df['is_canceled']
```

```
0         not canceled
1         not canceled
2         not canceled
3         not canceled
4         not canceled
              ...
119385    not canceled
119386    not canceled
119387    not canceled
119388    not canceled
119389    not canceled
Name: is_canceled, Length: 87396, dtype: object
```

```
#Same for 'is_repeated_guest' col
hotel_data_df['is_repeated_guest'] = hotel_data_df['is_repeated_guest'].replace([0,1], ['not repeated', 'repeated'])
```

```
hotel_data_df['is_repeated_guest']
```

```
0         not repeated
1         not repeated
2         not repeated
3         not repeated
4         not repeated
              ...
119385    not repeated
119386    not repeated
119387    not repeated
119388    not repeated
119389    not repeated
Name: is_repeated_guest, Length: 87396, dtype: object
```

```
#Now, we will check overall revenue hotel wise
hotel_wise_total_revenue = hotel_data_df.groupby('hotel')['revenue'].sum()
hotel_wise_total_revenue
```

```
hotel
City Hotel      18774101.54
Resort Hotel    15686837.77
Name: revenue, dtype: float64
```

```
hotel_data_df[['hotel', "revenue"]]
```

	hotel	revenue
0	Resort Hotel	0.00
1	Resort Hotel	0.00
2	Resort Hotel	75.00
3	Resort Hotel	75.00
4	Resort Hotel	196.00
...	...	...
119385	City Hotel	672.98
119386	City Hotel	1578.01
119387	City Hotel	1103.97
119388	City Hotel	730.80
119389	City Hotel	1360.80

87396 rows × 2 columns

```
hotel_data_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 87396 entries, 0 to 119389
Data columns (total 34 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   hotel                           87396 non-null  object
 1   is_canceled                     87396 non-null  object
 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  int64
 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  object
 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  int64
 24  days_in_waiting_list            87396 non-null  int64
 25  customer_type                   87396 non-null  object
 26  adr                             87396 non-null  float64
 27  required_car_parking_spaces     87396 non-null  int64
 28  total_of_special_requests       87396 non-null  int64
 29  reservation_status              87396 non-null  object
 30  reservation_status_date         87396 non-null  object
 31  total_stay_in_nights            87396 non-null  int64
 32  revenue                         87396 non-null  float64
 33  total_guest                     87396 non-null  int64
dtypes: float64(2), int64(18), object(14)
memory usage: 23.3+ MB
```

What all manipulations have you done and insights you found?

We have done few manipulations in the Data.

----Addition of columns----

We have seen that there are few columns required in Data to analysis purpose which can be evaluated from the given columns.

a) **Total Guests:** This columns will help us to evaluate the volumes of total guest and revenue as well. We get this value by adding total no. of Adults, Children & babies.

b) **Revenue:** We find revenue by multiplying adr & total guest. This column will use to analyse the profit and growth of each hotel.

----Delete of columns----

a)**company:** As we have seen that this columns has almost Null data. so we have delete this column as this will not make any impact in the analysis.

----Replace of Values in columns----

a)**is_canceled, is_not_canceled & is_repeated_guest:** We have seen, that these columns contains only 0,1 as values which represent the status of booing cancellation. We replace these values (0,1) from 'Canceled' & 'Not canceled. In the same way for column 'is_repeated_guest', we replace 0,1 from 'Repeated' & 'Not repeated'. Now this values will help to make better understanding while visulization.

----Changes in data type of values in columns----

a)**Agent & Children:** We checked that these columns contains float values, which is not making any sense in data as this values repreasent the count of guest & ID of agent. So we have changed the data type of these columns from 'float' to 'Integer'.

----Removed is_null values & duplicate entries----

a)Before visualize any data from the data set we have to do data wrangling. For that, we have checked the null value in all the columns. After checking, when we are getting a column which has more number of null values, dropped that column by using the 'drop' method. In this way, we are dropped the 'company' column. When we are find minimal number of null values, filling thse null values with necesary values as per requirement by using .fillna().

b) In the same, we have checked if there is any duplicacy in data & we found that there are few rows have duplicate data. So we have removed those row from data set by using .drop_duplicates() method.

In this way, we have removed unneccesary data & make our data clean and ready to analyse.

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

Chart - 1

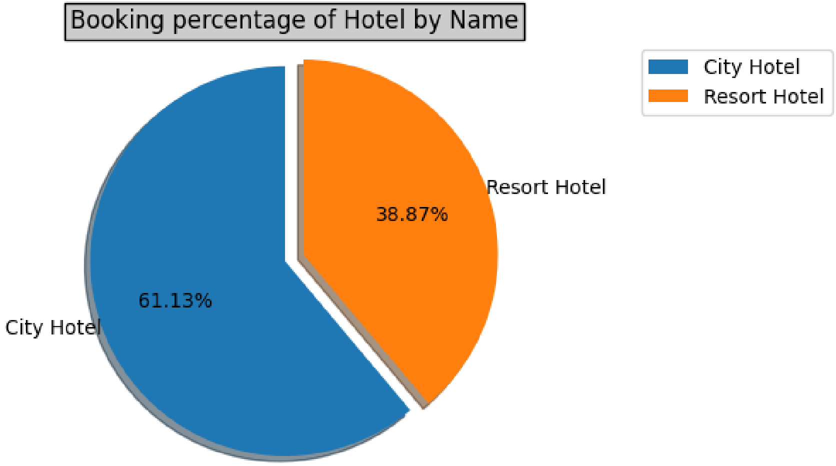
Which type of hotel genrally people prefer to book?

```
# Let's create a function which will give us bar chart of data respective with a col.
def get_count_from_column_bar(df, column_label):
  df_grpd = df[column_label].value_counts()
  df_grpd = pd.DataFrame({'index':df_grpd.index, 'count':df_grpd.values})
  return df_grpd


def plot_bar_chart_from_column(df, column_label, t1):
  df_grpd = get_count_from_column(df, column_label)
  fig, ax = plt.subplots(figsize=(14, 6))
  c= ['g','r','b','c','y']
  ax.bar(df_grpd['index'], df_grpd['count'], width = 0.4, align = 'edge', edgecolor = 'black', linewidth = 4, color = c, linestyle = ':', alpha = 0.5)
  plt.title(t1, bbox={'facecolor':'0.8', 'pad':3})
  plt.legend()
  plt.ylabel('Count')
  plt.xticks(rotation = 15) # use to format the lable of x-axis
  plt.xlabel(column_label)
  plt.show()
```

```
# Chart - 1 visualization code

def get_count_from_column(df, column_label):
  df_grpd = df[column_label].value_counts()
  df_grpd = pd.DataFrame({'index':df_grpd.index, 'count':df_grpd.values})
  return df_grpd
```

```
# plot a pie chart from grouped data
def plot_pie_chart_from_column(df, column_label, t1, exp):
  df_grpd = get_count_from_column(df, column_label)
  fig, ax = plt.subplots(figsize=(10,4))
  ax.pie(df_grpd.loc[:, 'count'], labels=df_grpd.loc[:, 'index'], autopct='%1.2f%%',startangle=90,shadow=True, labeldistance = 1, explode = exp)
  plt.title(t1, bbox={'facecolor':'0.8', 'pad':3})
  ax.axis('equal')
  plt.legend()
  plt.show()
```

```
exp1 = [0.05,0.05]
plot_pie_chart_from_column(hotel_data_df, 'hotel', 'Booking percentage of Hotel by Name', exp1)
```



1. Why did you pick the specific chart?

To present the data that in which hotel more booking have been done.

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

Here, we found that the booking number is Higher in City Hotel which is 61.13% than Resort Hotel which is 38.87%. Hence we can say that City hotel has more consumption

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, for both Hotels, this data making some positive business impact : -

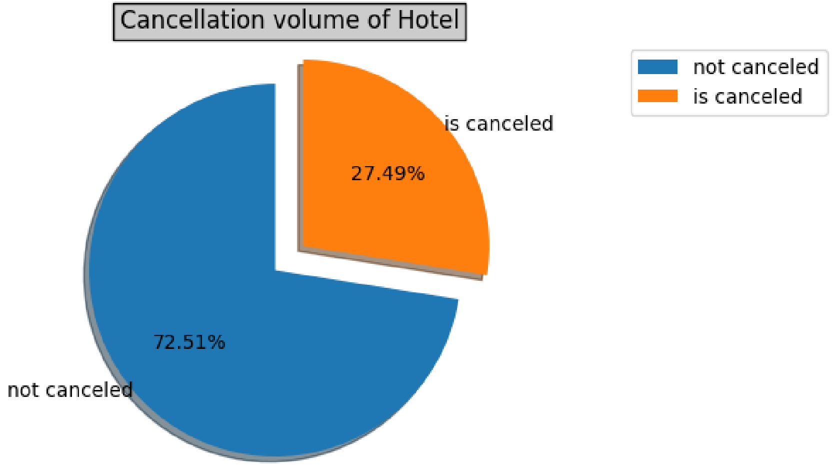
City Hotel :- Provided more services to attract more guest to increase more revenue.

Resort Hotel :- Find solution to attract guest and find what city hotel did to attract guest.

Chart - 2

What is the percentage of cancellation of Bookings?

```
# Chart - 2 visualization code
exp4 = [0,0.2]
plot_pie_chart_from_column(hotel_data_df, 'is_canceled', 'Cancellation volume of Hotel', exp4)
```



1. Why did you pick the specific chart?

In this chart, we presented the cancellation rate of the hotels booking

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

Here, we found that overall more than 25% of booking got cancelled

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

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

Here, we can see, that more than 27% booking getting cancelled.

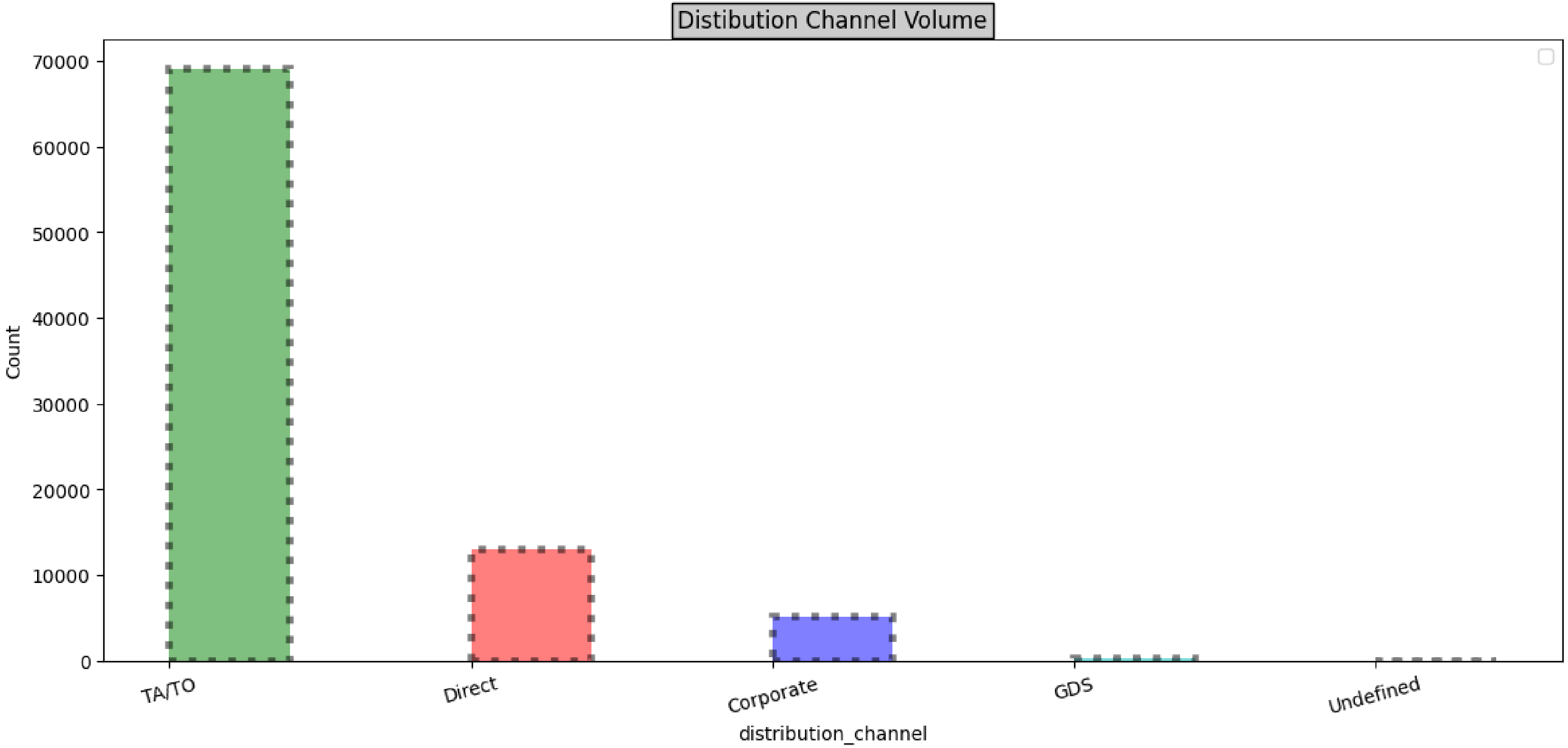
Solution: We can check the reason of cancellation of a booking & need to get this sort on business level

Chart - 3

Which type of customers do more bookings?

```
# Chart - 3 visualization code
plot_bar_chart_from_column(hotel_data_df,'distribution_channel', 'Distibution Channel Volume')
```

```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with no argument.
```



1. Why did you pick the specific chart?

The following chart represent maximum volume of booking done through which channel to represnt the numbers in descending order we chose bar graph

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

As clearly seen TA/TO(Tour of Agent & Tour of operator) is highest, recommending to continue booking through TA/TO

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

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

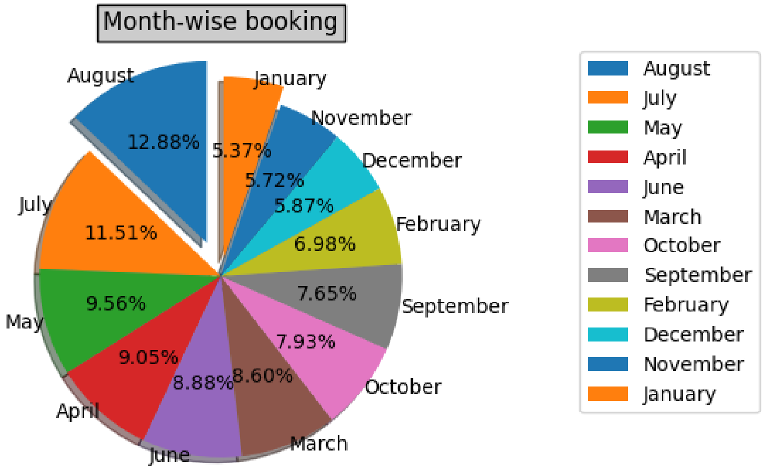
Yes this shows positive business impact.

Higher the number of TA/TO will help to increase the revenue generation of Hotel.

Chart - 4

What is the percentage share of booking in each month,on overall level ?

```
# Chart - 4 visualization code
exp2 = [0.2, 0,0,0,0,0,0,0,0,0,0,0.1]
plot_pie_chart_from_column(hotel_data_df, 'arrival_date_month', 'Month-wise booking', exp2)
```



1. Why did you pick the specific chart?

To show the percentage share of booking in each month,on overall level

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

The above percentage shows month May, July and Aug are the highest booking months due to holiday season. Recommending aggressive advertisement to lure more and more customers.

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

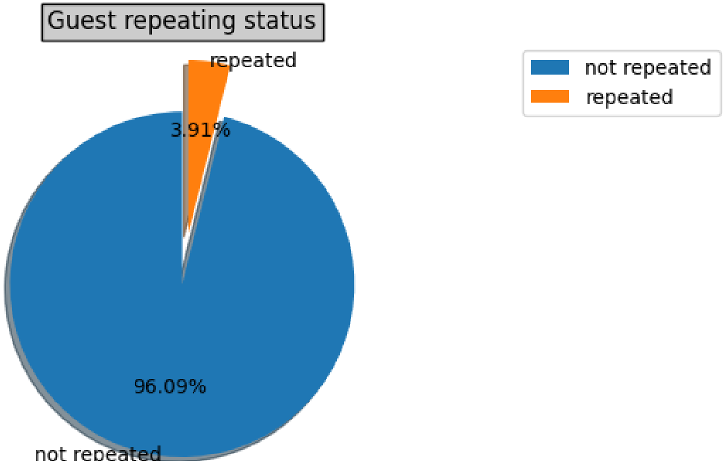
Are there any insights that lead to negative growth? Justify with specific reason.

Yes, with increased volume of visitors will help hotel to manage revenue in down time, will also help employee satisfaction and retention.

Chart - 5

What is the percentage of repeated guest?

```
# Chart - 5 visualization code
exp3 = [0,0.3]
plot_pie_chart_from_column(hotel_data_df, 'is_repeated_guest', 'Guest repeating status', exp3)
```



1. Why did you pick the specific chart?

To show the percentage share of repeated & non-repeated guests.

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

Here, we can see that the number of repeated guests is very less as compared to overall 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.

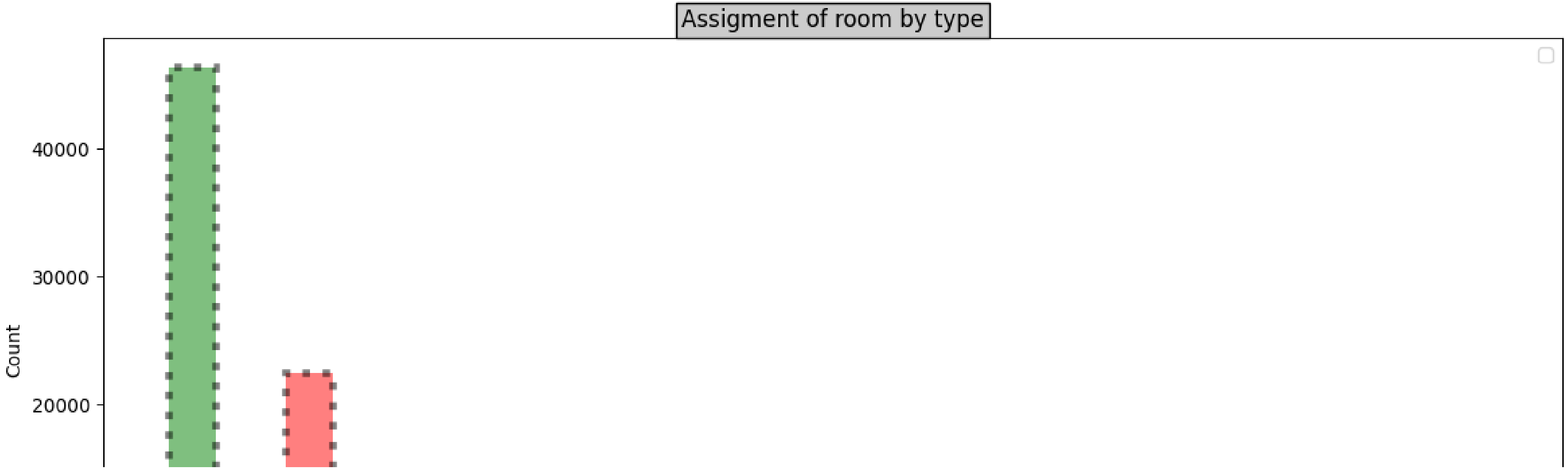
We can give alluring offers to non-repetitive customers during Off seasons to enhance revenue

Chart - 6

What is the most preferred room type?

```
# Chart - 6 visualization code
plt.figure(figsize=(0.5,0.5))
plot_bar_chart_from_column(hotel_data_df, 'assigned_room_type', 'Assigment of room by type')
plt.show()
```

```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with no argument.
<Figure size 50x50 with 0 Axes>
```



1. Why did you pick the specific chart?

To show distribution by volume, which room is alotted.

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

This chart shows room type 'A' is most prefered by guest.

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, Positive impact because 'A','D','E' is more prefered by guest due to better services offered in room type.

Chart - 7

Which type of hotel of market segment do more bookings?

```
# Chart - 7 visualization code
guest_month_wise = pd.DataFrame(hotel_data_df[['arrival_date_month', 'total_guest']])
guest_month_wise_df = guest_month_wise.groupby(['arrival_date_month'])['total_guest'].sum()
guest_month_wise_df.sort_values(ascending = False, inplace = True)
```

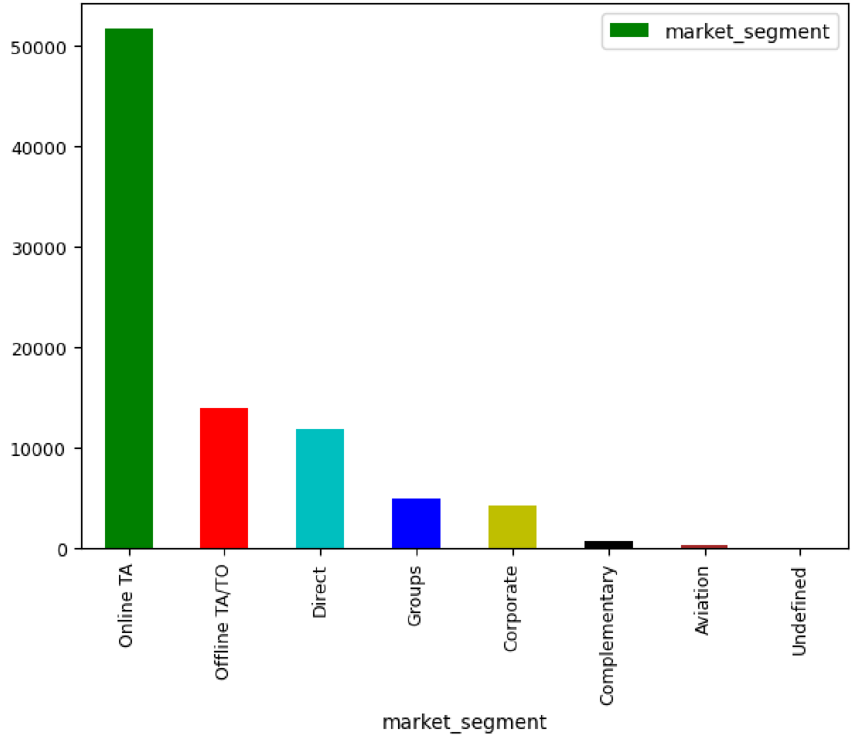
```
hotel_data_df['total_guest']
```

```
0         2
1         2
2         1
3         1
4         2
         ..
```

```
119385    2
119386    3
119387    2
119388    2
119389    2
Name: total_guest, Length: 87396, dtype: int64
```

```
market_segment_df = pd.DataFrame(hotel_data_df['market_segment'])
market_segment_df_data = market_segment_df.groupby('market_segment')['market_segment'].count()
market_segment_df_data.sort_values(ascending = False, inplace = True)
plt.figure(figsize=(7,5))
y = np.array([4,5,6])
market_segment_df_data.plot(kind = 'bar', color=['g', 'r', 'c', 'b', 'y', 'black', 'brown'], fontsize = 9,legend='True')
```

```
<Axes: xlabel='market_segment'>
```



1. Why did you pick the specific chart?

In this chart, we have seen market segment by which hotel has booked

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

Online TA has been used most frequently to book hotel by the guest.

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, it is creating positive business impact that guests are using Online TA market segment as most prefered to book hotels.

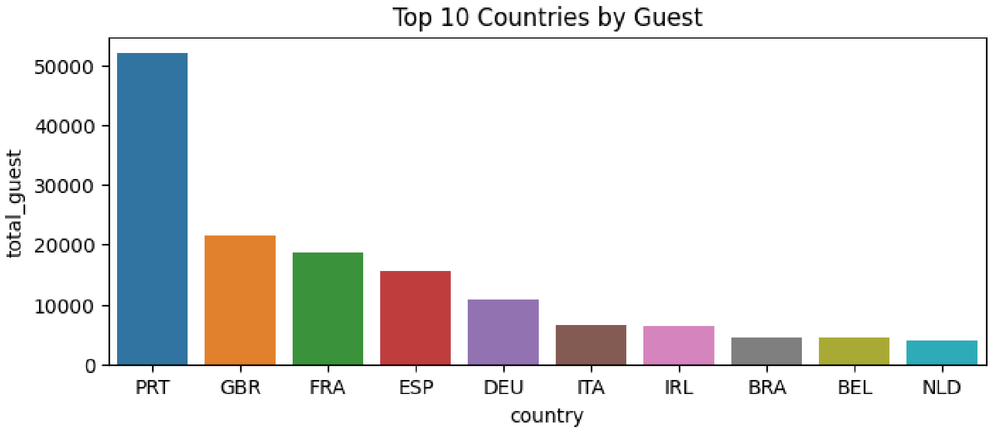
Chart - 8

From which country mostly guests are coming from?

```
# Chart - 8 visualization code
guest_country_wise = pd.DataFrame(hotel_data_df[['country', 'total_guest']])
guest_country_wise_df = guest_country_wise.groupby(['country'])['total_guest'].sum()
guest_country_wise_df.sort_values(ascending = False, inplace = True)
top_10_country_by_guest = guest_country_wise_df.head(10)
```

```
plt.figure(figsize=(8,3))
sns.barplot(x=top_10_country_by_guest.index,  y=top_10_country_by_guest).set(title='Top 10 Countries by Guest')
print("\n\nPRT = Portugal, GBR = Great Britain & Northern Ireland, FRA = France, ESP = Spain, DEU = Germany\nITA = Italy, IRL = Ireland, BRA = Brazil, BEL = Belgium, NLD = Nether
```

```
PRT = Portugal, GBR = Great Britain & Northern Ireland, FRA = France, ESP = Spain, DEU = Germany
ITA = Italy, IRL = Ireland, BRA = Brazil, BEL = Belgium, NLD = Netherland
```



1. Why did you pick the specific chart?

We have seen that mostly from which country Guests is coming

Chart is showing for top 10 country

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

As we can see, that maximum guest is coming from 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.

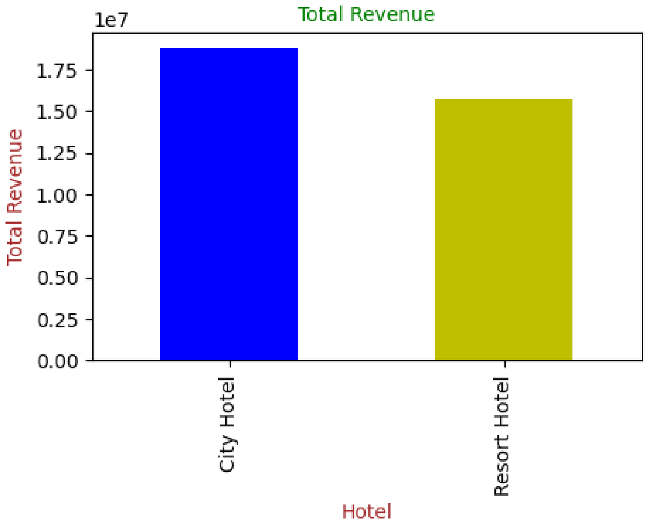
We can do more advertising & can provide attractive offers to Portugal guests to enhance the customer volume

Chart - 9

Which hotels generating more ADR?

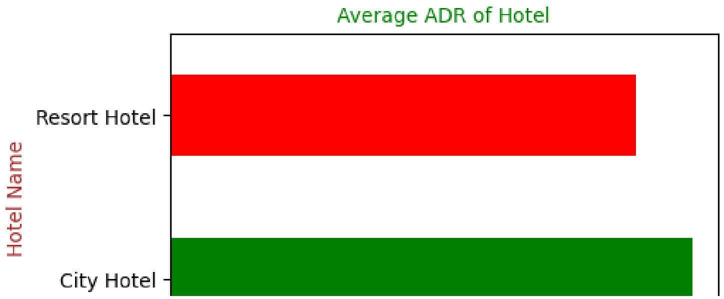
```
# Chart - 9 visualization code
plt.figure(figsize = (5,3))
hotel_wise_revenue = hotel_data_df.groupby('hotel')['revenue'].sum()
hotel_wise_revenue
ax = hotel_wise_revenue.plot(kind = 'bar', color = ('b', 'y'))
plt.xlabel("Hotel", fontdict={'fontsize': 10, 'fontweight' : 5, 'color' : 'Brown'})
plt.ylabel("Total Revenue", fontdict={'fontsize': 10, 'fontweight' : 5, 'color' : 'Brown'} )
plt.title("Total Revenue", fontdict={'fontsize': 10, 'fontweight' : 5, 'color' : 'Green'} )
```

```
Text(0.5, 1.0, 'Total Revenue')
```



```
average_adr = hotel_data_df.groupby('hotel')['adr'].mean()
average_adr
plt.subplots(figsize=(5, 3))
average_adr.plot(kind = 'barh', color = ('g', 'r'))
plt.xlabel("Average ADR", fontdict={'fontsize': 10, 'fontweight' : 5, 'color' : 'Brown'})
plt.ylabel("Hotel Name", fontdict={'fontsize': 10, 'fontweight' : 5, 'color' : 'Brown'} )
plt.title("Average ADR of Hotel", fontdict={'fontsize': 10, 'fontweight' : 5, 'color' : 'Green'} )
```

```
Text(0.5, 1.0, 'Average ADR of Hotel')
```



1. Why did you pick the specific chart?

To specify the average ADR for both hotels

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

As we can see the average ADR of City hotel is higher than Resort hotel, so the profit and revenue will be higher for city hotel

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

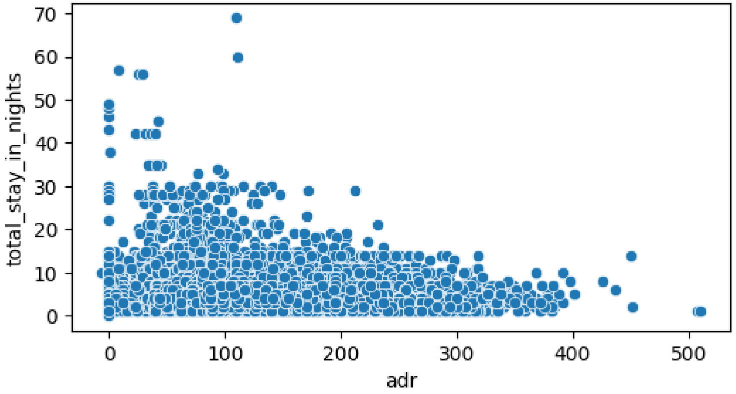
Are there any insights that lead to negative growth? Justify with specific reason.

Here, we can do more advertising for City hotel to get more customer, which result higher profit

Chart - 10

What is the comparision & affect of total stay days vs ADR?

```
# Chart - 10 visualization code
plt.figure(figsize = (6,3))
sns.scatterplot(y = 'total_stay_in_nights', x = 'adr', data = hotel_data_df[hotel_data_df['adr'] < 1000])
plt.show()
```



1. Why did you pick the specific chart?

To show comparision & affect of total stay days vs ADR

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

Here, we found that if guest's stay days is getting decreased, ADR is getting high

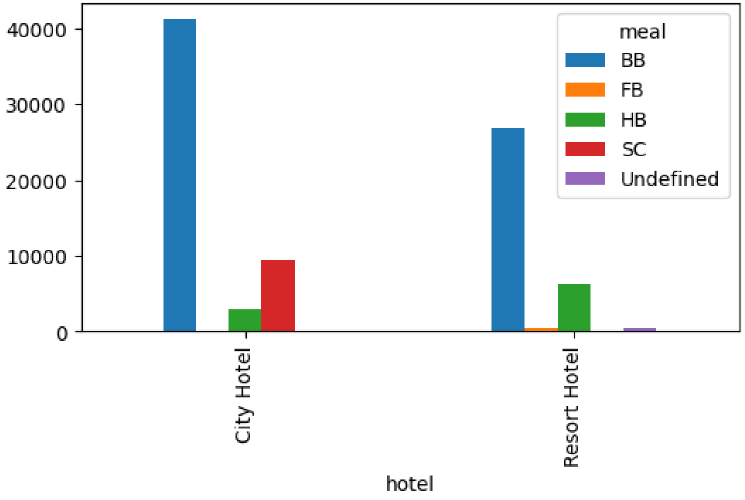
Chart - 11

Which kind of meal is mostly preffered by the guests?

```
# Chart - 11 visualization code
plt.figure(figsize = (6,6), dpi = 100)
hotel_wise_meal = hotel_data_df.groupby(['hotel', 'meal'])['meal'].count().unstack()
hotel_wise_meal.plot(kind ='bar', figsize = (6,3))
hotel_wise_meal
```

meal	BB	FB	HB	SC	Undefined
hotel					
City Hotel	41229.0	11.0	2791.0	9397.0	NaN
Resort Hotel	26749.0	349.0	6294.0	84.0	492.0

```
<Figure size 600x600 with 0 Axes>
```



1. Why did you pick the specific chart?

To show the meal preferance of the guest hotel-wise

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

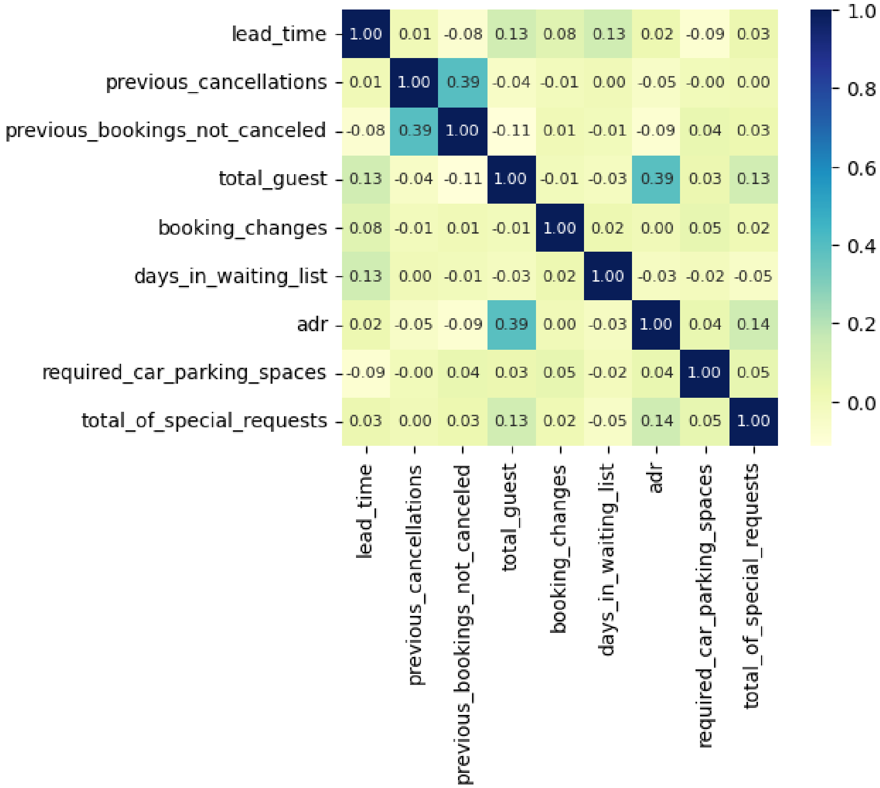
As we can see, BB (Bed & breakfast) meal is most prefered by guests in both the hotels. So Hotel can give more delisious dishes in this meal to get customer repeat & attaract new customer

Chart - 12

Correlation Heatmap

```
# Correlation Heatmap visualization code
corr_df = hotel_data_df[['lead_time','previous_cancellations', 'previous_bookings_not_canceled', 'total_guest',
                    'booking_changes', 'days_in_waiting_list', 'adr', 'required_car_parking_spaces', 'total_of_special_requests']].corr()
f, ax = plt.subplots(figsize=(6, 4))
sns.heatmap(corr_df, annot = True, fmt='.2f', annot_kws={'size': 8},  vmax=1, square=True, cmap="YlGnBu")
```

```
<Axes: >
```



	lead_time	previous_cancellations	previous_bookings_not_canceled	total_guest	booking_changes	days_in_waiting_list	adr	required_car_parking_spaces	total_of_special_requests
lead_time	1.00	0.01	-0.08	0.13	0.08	0.13	0.02	-0.09	0.03
previous_cancellations	0.01	1.00	0.39	-0.04	-0.01	0.00	-0.05	-0.00	0.00
previous_bookings_not_canceled	-0.08	0.39	1.00	-0.11	0.01	-0.01	-0.09	0.04	0.03
total_guest	0.13	-0.04	-0.11	1.00	-0.01	-0.03	0.39	0.03	0.13
booking_changes	0.08	-0.01	0.01	-0.01	1.00	0.02	0.00	0.05	0.02
days_in_waiting_list	0.13	0.00	-0.01	-0.03	0.02	1.00	-0.03	-0.02	-0.05
adr	0.02	-0.05	-0.09	0.39	0.00	-0.03	1.00	0.04	0.14
required_car_parking_spaces	-0.09	-0.00	0.04	0.03	0.05	-0.02	0.04	1.00	0.05
total_of_special_requests	0.03	0.00	0.03	0.13	0.02	-0.05	0.14	0.05	1.00

1. Why did you pick the specific chart?

To understand the relationsip between different numerical values

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

Highest corelation value between axis is 39% positive & lowest corelation value between the axis is -9% negative

5. Solution to Business Objective

Business objective attained as follows:

1. For hotel business to flourish few things which we need to consider is high revenue generation, customers satisfaction and employeee retention.
2. We are able achieve the same by showing the client which are the months which are high in revenue generation by pie chart distribution
3. Increasing the revenue achieved by bar chart distribution of which typre room are most reserved and what are the months likely for visitors
4. So for these the client can be well prepare in advance so that minimum grievances would be faced by clients in long run and would help in further enhancement of their hospitality.
5. Outliers like higher the visitor then adr has reduced drastically was shown in scattered plot so in off season client can engage with offices for bulk booking this will aslo help extra revenue generation
6. We are able to show the trend of arrivals of visitor at client locations through which client engaged visitos well advance for there entaertainment and leisure activities
7. We where also able to co relate the values showing the max and min percentage between them so that the percenytage lying those numbers can be enhanced by various medium

Conclusion

1. City Hotel seems to be more preferred among travellers and it also generates more revenue & profit.
2. Most number of bookings are made in July and August as compared rest of the months.
3. Room Type A is the most preferred room type among travellers.
4. Most number of bookings are made from Portugal & Great Britain.
5. Most of the guest stays for 1-4 days in the hotels.
6. City Hotel retains more number of guests.
7. Around one-fourth of the total bookings gets cancelled. More cancellations are from City Hotel.
8. New guest tends to cancel bookings more than repeated customers.
9. Lead time, number of days in waiting list or assignation of reserved room to customer does not affect cancellation of bookings.
10. Corporate has the most percentage of repeated guests while TA/TO has the least whereas in the case of cancelled bookings TA/TO has the most percentage while Corporate has the least.
11. The length of the stay decreases as ADR increases probably to reduce the cost.

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

✓ 3s completed at 4:03 PM