Project Name - Hotel Booking Analysis

Project Type - EDA

Contribution - Individual

Member - Saif U llah

Project Summary -

The dataset we are dealing with here, presents a birdeye view of hotel bookings, preferences made during booking in terms of meal,parking slots,distribution channels invoved, repeated customers and much more.

The Hotel industry is an ever-growing industry and various kinds of trends demand to be noticed and exercised regularly in order to minimize retention rates and expand businesses.The average hotel occupancy rate is around 66% in India and the revenue generated is over $3.952 trillion world-wide , so we can assuredly say that the industry is here to stay and proliferate for higher returns.

In this project, I will try to highlight the significant business-impacting trends oberved over the period of time like when the best time of year to book a hotel room is and which amenities or practices result in higher booking rates and make some useful analysis to facilitate the interest of stake holders and business owners in terms of reducing retentions rates, identifying the indicators as to why the guests are leaving or not re-booking and measures to amplify the bookings.

GitHub Link -

Double-click (or enter) to edit

https://github.com/saif90834/Hotel_Bookings

https://github.com/saif90834

Problem Statement

Exploration of dataset to derive useful insights that govern the bookings.

Define Your Business Objective?

Analysing important factors that govern hotel bookings.

General Guidelines : -

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

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

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

3. Each and every logic should have proper comments.

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

```
# Chart visualization code
```

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

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

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

U - Univariate Analysis,

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

M - Multivariate Analysis]

Let's Begin !

1. Know Your Data

Import Libraries

```
# Import Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly.express as px
```

Dataset Loading

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

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

```
# Load Dataset
path = '/content/drive/MyDrive/Data Machine Learning/Hotel Bookings.csv'
df = pd.read_csv(path)
```

Dataset First View

```
# Dataset First Look
df.head(2)
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_w
0	Resort Hotel	0	342	2015	July	27	1	
1	Resort Hotel	0	737	2015	July	27	1	

2 rows × 32 columns

Dataset Rows & Columns count

```
# Dataset Rows & Columns count
df.shape
```

```
(119390, 32)
```

Dataset Information

```
# Dataset Info
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
```

```
df.describe()
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	sta
count	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	
mean	0.370416	104.011416	2016.156554	27.165173	15.798241	0.927599	
std	0.482918	106.863097	0.707476	13.605138	8.780829	0.998613	
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000	
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	0.000000	
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	1.000000	
75%	1.000000	160.000000	2017.000000	38.000000	23.000000	2.000000	
max	1.000000	737.000000	2017.000000	53.000000	31.000000	19.000000	

Duplicate Values

```
# Dataset Duplicate Value Count
len(df[df.duplicated()])
```

```
31994
```

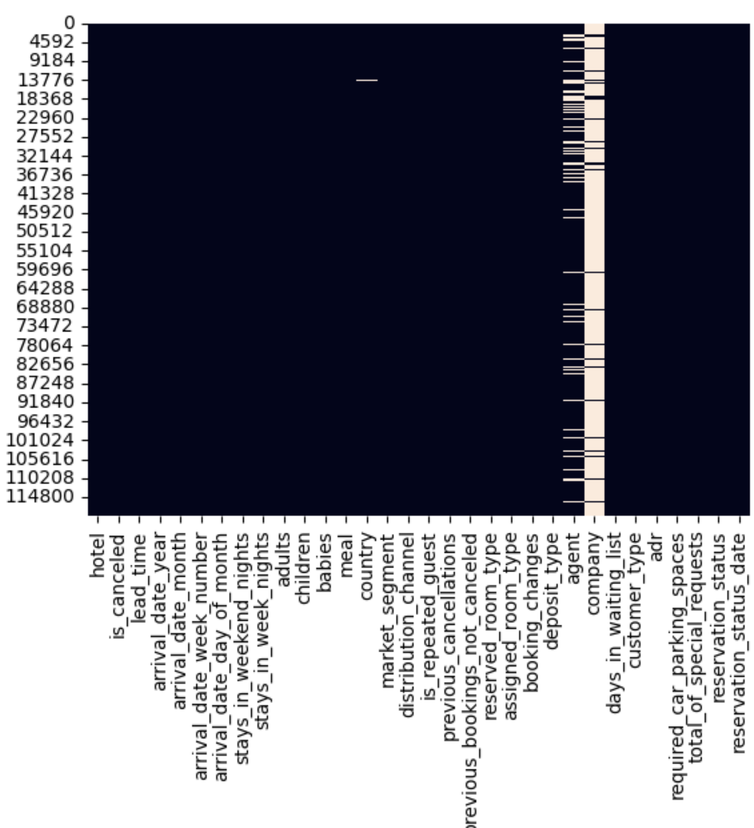
Missing Values/Null Values

```
# Missing Values/Null Values Count
df.isna().sum()
```

```
hotel                                  0
is_canceled                            0
lead_time                              0
arrival_date_year                      0
arrival_date_month                     0
arrival_date_week_number               0
arrival_date_day_of_month              0
stays_in_weekend_nights                0
stays_in_week_nights                   0
adults                                 0
children                               4
babies                                 0
meal                                   0
country                              488
market_segment                         0
distribution_channel                   0
is_repeated_guest                      0
previous_cancellations                 0
previous_bookings_not_canceled         0
reserved_room_type                     0
assigned_room_type                     0
booking_changes                        0
deposit_type                           0
agent                              16340
company                           112593
days_in_waiting_list                   0
customer_type                          0
adr                                    0
required_car_parking_spaces            0
total_of_special_requests              0
reservation_status                     0
reservation_status_date                0
dtype: int64
```

```
# Visualizing the missing values
sns.heatmap(df.isnull(), cbar = False)
```

```
<Axes: >
```



What did you know about your dataset?

The given dataset describes various aspects of Hotel Bookings like meal,country,weekly vs weekend stays,agent etc. Our objective is to analyze various factors that govern bookings and highlight the causes that result in poor business profitability.

- The dataset consist of 119390 rows and 32 columns.
- There are 31994 duplicate values in the dataset.
- There are 4 columns having missing values:children,country,agent and company.
- The dataset consist of 2 columns having Binary data(0 and 1): is_cancelled, is_repeated_guest.

2. Understanding Your Variables

```
# Dataset Columns
df.columns
```

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

```
# Dataset Describe
df.describe()
```

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

Variables Description

- **hotel** : there are two types of hotels,city hotel and resort hotel.
- **is_cancelled** : this indicates whether booking was cancelled(1) or not(0)
- **lead_time** : Time lapse between reservation and arrival date.
- **arrival_date_year** : Year of arrival date.
- **arrival_date_month** : Month of arrival date.
- **arrival_date_week_number** : Week number of arrival date.
- **arrival_date_day_of_month** : Day of arrival date.
- **stays_in_weekend_nights** : No of weekend night the guest stayed or booked the stay.
- **stays_in_week_nights** : No of week nights the guest stayed or booked the stay.
- **adults** : No of adults.
- **children** : No of children.
- **babies** : No of babies.
- **meal** : Kind of meal opted for.
- **country** : Country code.

- **market_segment** : Which segment of market the customer belong to.
- **distribution_channel** : How the customer accessed the stay-corporate booking/direct/TA.TO.
- **is_repeated_guest** : Guest coming for first time(0) or not (1).
- **previous_cancellations** : Was there a cancellation before.
- **previous_bookings_not_canceled** : Number of previous bookings not cancelled.
- **reserved_room_type** : Type of room reserved.
- **assigned_room_type** : Type of room assigned.
- **booking_changes** : Count of changes made to the booking.
- **deposit_type** : Deposit Type.
- **agent** : Booked through agent.
- **company** : ID of the company that made the booking.
- **days_in_waiting_list** : Number of days in waiting list.
- **customer_type** : Type of customer-Contract,Group,Transient,Transient Party.
- **adr** : Average Daily Rate
- **required_car_parking_spaces** : If car parking is required.
- **total_of_special_requests** : Number of additional special requirements.
- **reservation_status** : Last status of reservation like checked out,cancelled or no show.
- **reservation_status_date** : Date of specified status.

Check Unique Values for each variable.

```
# Check Unique Values for each variable.
print("No. of unique values:")
for i in df.columns.tolist():
  print(i,"= ",df[i].nunique())
```

```
No. of unique values:
hotel =  2
is_canceled =  2
lead_time =  479
arrival_date_year =  3
arrival_date_month =  12
arrival_date_week_number =  53
arrival_date_day_of_month =  31
stays_in_weekend_nights =  17
stays_in_week_nights =  35
adults =  14
children =  5
babies =  5
meal =  5
country =  177
market_segment =  8
distribution_channel =  5
is_repeated_guest =  2
previous_cancellations =  15
previous_bookings_not_canceled =  73
reserved_room_type =  10
assigned_room_type =  12
booking_changes =  21
deposit_type =  3
agent =  333
company =  352
days_in_waiting_list =  128
customer_type =  4
adr =  8879
required_car_parking_spaces =  5
total_of_special_requests =  6
reservation_status =  3
reservation_status_date =  926
```

3. ***Data Wrangling***

Data Wrangling Code

```
# Write your code to make your dataset analysis ready.
df_new = df.copy()
df_new.head(3)
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_w
0	Resort Hotel	0	342	2015	July	
1	Resort Hotel	0	737	2015	July	
2	Resort Hotel	0	7	2015	July	

3 rows × 32 columns

```
#dropping dulicates
df_new.drop_duplicates(inplace = True)
```

```
#merging date,month and year column as arrival timestamp.
df_new['arrival_timestamp'] = df_new['arrival_date_day_of_month'].astype(str) + "-" + df_new['arrival_date_month'].astype(str) + "-" + df_ne

#merging stays in weekend nights and stays in week nights as total stay
df_new['total_stay'] = df_new['stays_in_week_nights'] + df_new['stays_in_weekend_nights']

#merging adults,children and babies column as total guests.
df_new['total_guests'] = df_new['adults']+df_new['children']+df_new['babies']
```

```
#pd.DataFrame(df_new.groupby('hotel')['hotel'].value_counts().reset_index(name='total bookings'))
```

```
#Examining hotel type preference in terms of total stay
pd.DataFrame(df_new.groupby('hotel')['total_stay'].sum().reset_index(name='total stay'))
```

	hotel	total stay
0	City Hotel	168117
1	Resort Hotel	149188

```
#Examining average daily rate of City vs Resort Hotel
pd.DataFrame(df_new.groupby('hotel')['adr'].sum().reset_index(name='ADR'))
```

	hotel	ADR
0	City Hotel	5929757.03
1	Resort Hotel	3363692.96

```
pd.DataFrame(df_new.groupby('hotel')['previous_cancellations'].sum().reset_index(name='No of cancellations'))
```

	hotel	No of cancellations
0	City Hotel	1911
1	Resort Hotel	747

```
#Analyzing how the deposit type parameter affects the total_stay/bookings made by customers.
pd.DataFrame(df_new.groupby('deposit_type')['total_stay'].sum().reset_index(name='total stay'))
```

	deposit_type	total stay
0	No Deposit	314029
1	Non Refund	2874
2	Refundable	402

```
#Understanding the relationship between deposit type and nature of cancellations.
pd.DataFrame(df_new.groupby('deposit_type')['previous_cancellations'].sum().reset_index(name='No of cancellations'))
```

	deposit_type	No of cancellations

```
#Understanding the affect of deposit type on days in waiting list.
pd.DataFrame(df_new.groupby('deposit_type')['days_in_waiting_list'].sum().reset_index(name='No of days in waiting list'))
```

	deposit_type	No of days in waiting list
0	No Deposit	53280
1	Non Refund	11426
2	Refundable	803

```
pd.DataFrame(df_new.groupby('customer_type')['total_stay'].sum().reset_index(name='total stay'))
```

	customer_type	total stay
0	Contract	18942
1	Group	1589
2	Transient	257609
3	Transient-Party	39165

```
#Understanding if there is a relation between parking spaces and type of hotel preferred based on parking spaces.
pd.DataFrame(df_new.groupby('hotel')['required_car_parking_spaces'].value_counts())
```

		required_car_parking_spaces
hotel	**required_car_parking_spaces**	
City Hotel	**0**	51532
	1	1891
	2	3
	3	2
Resort Hotel	**0**	28551
	1	5389
	2	25
	8	2
	3	1

```
#Understanding the type of meal preference.
pd.DataFrame(df_new.groupby('meal')['meal'].value_counts())
```

		meal
meal	**meal**	
BB	**BB**	67978
FB	**FB**	360
HB	**HB**	9085
SC	**SC**	9481
Undefined	**Undefined**	492

```
#Analyzing which country customers are making maximum reservations.
pd.DataFrame(df_new['country'].value_counts().reset_index(name='No of bookings')).head()
```

index	No of bookings

```
#Understanding nature of bookings based on market segment
pd.DataFrame(df_new.groupby('market_segment')['total_stay'].sum().reset_index(name='total stay'))
```

	market_segment	total stay
0	Aviation	811
1	Complementary	1172
2	Corporate	8609
3	Direct	38139
4	Groups	17052
5	Offline TA/TO	65305
6	Online TA	186214
7	Undefined	3

```
#Understanding the ADR based on market segment to target the right category of customer
pd.DataFrame(df_new.groupby('market_segment')['adr'].mean().round(2).reset_index(name='ADR')).sort_values( by = 'ADR',ascending = False)
```

	market_segment	ADR
6	Online TA	118.17
3	Direct	116.58
0	Aviation	100.17
5	Offline TA/TO	81.76
4	Groups	74.86
2	Corporate	68.15
7	Undefined	15.00
1	Complementary	3.05

```
#Understanding the average daily rates of both types of hotels.
pd.DataFrame(df_new.groupby('hotel')['adr'].mean().reset_index(name='ADR'))
```

	hotel	ADR
0	City Hotel	110.985944
1	Resort Hotel	99.025346

```
#Understanding the nature of cancellations made by different market segments
pd.DataFrame(df_new.groupby('market_segment')['previous_cancellations'].sum().reset_index(name='No. of previous cancellations'))
```

	market_segment	No. of previous cancellations
0	Aviation	13
1	Complementary	158
2	Corporate	769
3	Direct	197
4	Groups	420
5	Offline TA/TO	312
6	Online TA	789
7	Undefined	0

```
#Understanding which market segments are making more repeated bookings.
pd.DataFrame(df_new.groupby('market_segment')['is_repeated_guest'].sum().reset_index(name='No. of repeated guests'))
```

	market_segment	No. of repeated guests
0	Aviation	63
1	Complementary	224
2	Corporate	1445
3	Direct	778
4	Groups	57
5	Offline TA/TO	269

```
#Understanding the relation between total_stay and distribution channel
pd.DataFrame(df_new.groupby('distribution_channel')['total_stay'].sum().reset_index(name='Total stay'))
```

	distribution_channel	Total stay
0	Corporate	11775
1	Direct	41473
2	GDS	361
3	TA/TO	263679
4	Undefined	17

```
#Understanding the ADR achived through different distribution channels.
pd.DataFrame(df_new.groupby('distribution_channel')['adr'].sum().reset_index(name='ADR'))
```

	distribution_channel	ADR
0	Corporate	348128.18
1	Direct	1417427.25
2	GDS	21777.53
3	TA/TO	7505885.83
4	Undefined	231.20

```
#Understanding the nature of cancellations based on distribution channels.
pd.DataFrame(df_new.groupby('distribution_channel')['previous_cancellations'].sum().reset_index(name='No. of previous cancellations'))
```

	distribution_channel	No. of previous cancellations
0	Corporate	893
1	Direct	344
2	GDS	0
3	TA/TO	1421
4	Undefined	0

```
#Understanding which distribution channels are benefecial for the business.
pd.DataFrame(df_new.groupby('distribution_channel')['is_repeated_guest'].sum().reset_index(name='No. of repeated guests'))
```

	distribution_channel	No. of repeated guests
0	Corporate	1539
1	Direct	917
2	GDS	3
3	TA/TO	956
4	Undefined	0

```
#Understanding how the adr is affected by the number of guests.
pd.DataFrame(df_new.groupby('total_guests')['adr'].sum().reset_index(name='ADR'))
```

	total_guests	ADR
0	0.0	1882.17
1	1.0	1269662.18
2	2.0	5818265.99
3	3.0	1452081.79
4	4.0	722298.26
5	5.0	28828.49
6	6.0	0.00
7	10.0	95.00
8	12.0	217.61
9	20.0	0.00
10	26.0	0.00
11	27.0	0.00
12	40.0	0.00
13	50.0	0.00

```
pd.DataFrame(df_new.groupby('total_guests')['is_repeated_guest'].sum().reset_index(name='No of repeated bookings'))
```

	total_guests	No of repeated bookings
0	0.0	51
1	1.0	2188
2	2.0	1041
3	3.0	96
4	4.0	36
5	5.0	3
6	6.0	0
7	10.0	0
8	12.0	0
9	20.0	0
10	26.0	0
11	27.0	0
12	40.0	0
13	50.0	0
14	55.0	0

```
#Understanding which customer type are making most booking changes.
pd.DataFrame(df_new.groupby('customer_type')['booking_changes'].sum().reset_index(name='Total Changes made'))
```

	customer_type	Total Changes made
0	Contract	471
1	Group	165
2	Transient	16142
3	Transient-Party	6959

```
#Understanding nature of special requests made by different customer types.
pd.DataFrame(df_new.groupby('customer_type')['total_of_special_requests'].sum().reset_index(name='total of special requests'))
```

	customer_type	total of special requests
0	Contract	2632

```
#Understanding nature of special requests made by different customer types.
pd.DataFrame(df_new.groupby('children')['total_of_special_requests'].sum().reset_index(name='total of special requests'))
```

	children	total of special requests
0	0.0	53735
1	1.0	4759
2	2.0	2483
3	3.0	69
4	10.0	1

```
#Understanding nature of special requests made by different customer types.
pd.DataFrame(df_new.groupby('babies')['total_of_special_requests'].sum().reset_index(name='total of special requests'))
```

	babies	total of special requests
0	0	59645
1	1	1378
2	2	28
3	9	0
4	10	1

```
#Creating df copy grouped by hotel type to better understand business driving factors.
hotel_type_city =  df_new[df_new['hotel']=='City Hotel'].reset_index()
```

```
hotel_type_resort = df_new[df_new['hotel']=='Resort Hotel'].reset_index()
```

```
#Understanding total cancellations made and factors based on.
total_cancellings = df_new['is_canceled'].sum()
print(f'Total Cancellations : {total_cancellings}')
no_of_cancellings_city_hotel = hotel_type_city['is_canceled'].sum()
print(f'Total Cancellations in City Hotels : {no_of_cancellings_city_hotel}')
no_of_cancellings_resort_hotel = hotel_type_resort['is_canceled'].sum()
print(f'Total Cancellations in Resort Hotels : {no_of_cancellings_resort_hotel}')
```

```
Total Cancellations : 24025
Total Cancellations in City Hotels : 16049
Total Cancellations in Resort Hotels : 7976
```

What all manipulations have you done and insights you found?

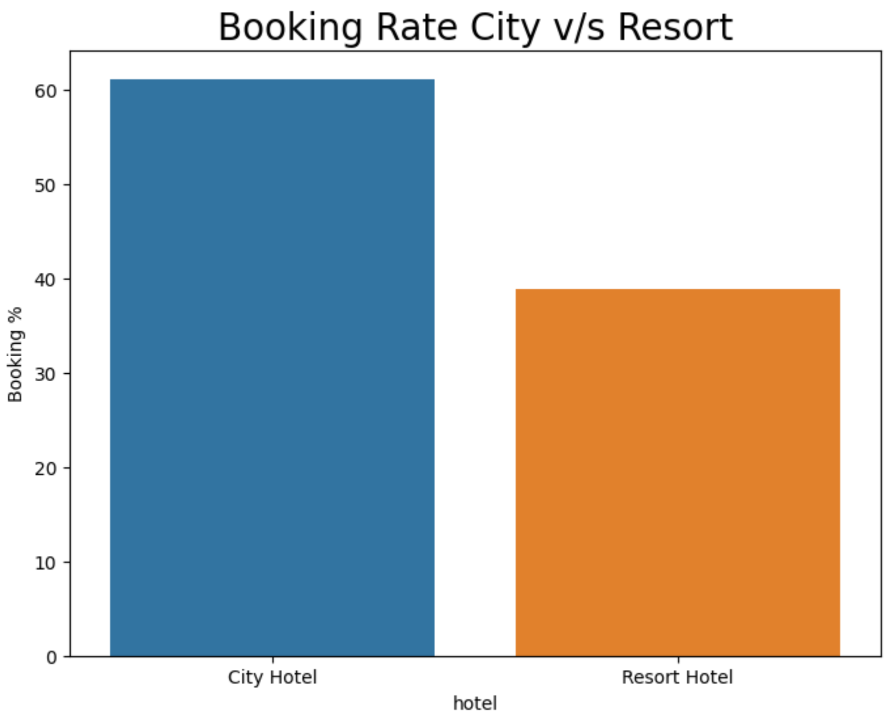
After the cleaning and wrangling of data, I found various insights and patterns governing the number of bookings made. We can get a clear view through visualization of the patterns, but deep diving through the dataset is required to examine the behavior of customers and business trends. I tried to understand the patterns of booking and cancellations made in city hotel type vs. resort hotel type. We also analyzed the Average Daily Rate of both types of hotel. Further we examined the patterns in no of bookings, no of cancellations ,days of waiting based on categorical variables like deposit type, type of customer, market segment, distribution type. We also analyzed the demand of parking spaces in different types of hotels and by different categories of customers. Further, we examined the meal preferences and business promulgation through different countries. We also saw, which segment of market shows tendency for re-booking so that we can target the right customers for business flourishment. By keenly observing these patterns, the stakeholders and business owners can drive better decision making in terms of market segmentation and facility offerings.

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

Chart - 1 : Booking Rate City v/s Resort

```
# Chart - 1 visualization code
print("Booking Rate City v/s Resort")
#Plotting bar plot for percentage of booking in hotel
hotel = df_new.groupby('hotel')
d1 = pd.DataFrame((hotel.size()/df_new.shape[0])*100).reset_index().rename(columns = {0:'Booking %'}) #Calculating percentage
plt.figure(figsize = (8,6))
sns.barplot(x = d1['hotel'], y = d1['Booking %'] )
plt.title('Booking Rate City v/s Resort',fontsize =20)
plt.show()
```

```
Booking Rate City v/s Resort
```



Bar Charts are a good way to represent categorical data with rectangular bars to give a steady comparison between the booking trends of both types of Hotels. Booking Rate is more in City Hotels (60%) as compared to Resort Hotels (40%)

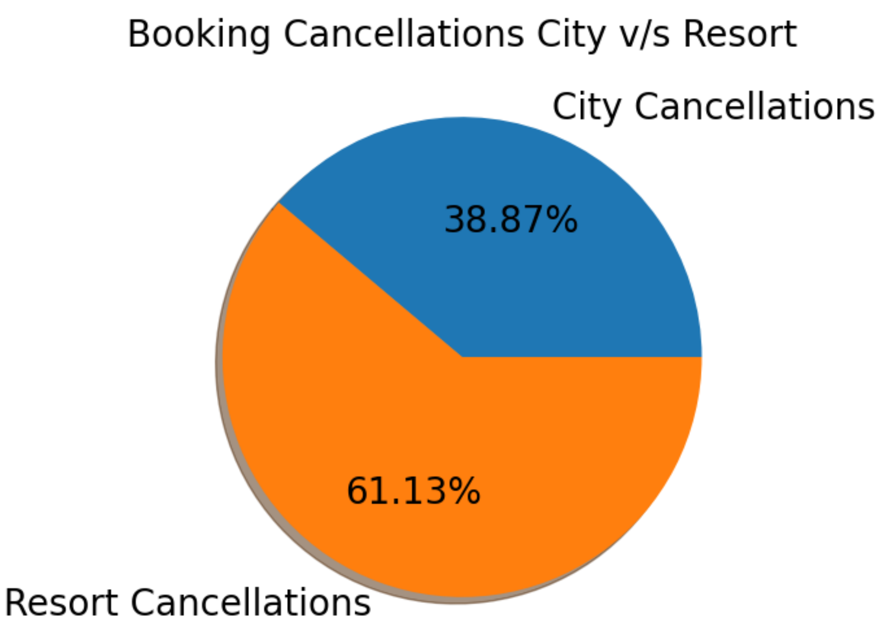
Chart - 2 : Booking Cancellation Trends

```
#Total Cancellations made
plt.figure(figsize=(8,6))
labels = ['Not canceled','Canceled', ]
df_new['is_canceled'].value_counts().plot.pie(labels = labels ,autopct='%0.2f%%', shadow=True,fontsize=20)
plt.title('Total Cancellation Percentage',fontsize = 20)
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7b19b3aaf700>
```

Chart-3 : Booking Cancellation Trends Hotel-Wise

```
# Booking Cancellation Trends Hotel-Wise

hotel = pd.DataFrame(df_new.groupby('hotel')['hotel'].value_counts())
hotel_count = hotel['hotel'].tolist()
city_cancellation_percentage = (no_of_cancellings_city_hotel/hotel_count[0])*100
resort_cancellation_percentage = (no_of_cancellings_city_hotel/hotel_count[1])*100
cancelations_consolidated = [city_cancellation_percentage,resort_cancellation_percentage]
label =['City Cancellations','Resort Cancellations']
plt.figure(figsize=(8,6))
plt.pie(cancelations_consolidated,labels=label,shadow=True,autopct='%0.2f%%',textprops={'fontsize': 20})
plt.title('Booking Cancellations City v/s Resort',fontsize =20)
plt.show()
```



A pie chart is a circular statistical graphic, which is divided into slices to illustrate numerical proportion.Approximately 27.5% bookings are cancelled out of which there are more cancellations observed in Resort Hotels (60%) as compared to City Hotels(40%).

Chart - 4 : Booking Trends Month-Wise

```
# Chart - 3 visualization code
#Understanding nature of bookings month-wise
print(df_new['arrival_date_month'].value_counts())

#Plotting seaborn bar chart to visualize no of bookings hotel wise as well as month wise.
plt.figure(figsize=(8,6))
graph=sns.countplot(data= df_new, x='arrival_date_month', hue = 'hotel',order = ['January', 'February', 'March', 'April', 'May', 'June', 'Jul
          'August', 'September', 'October', 'November', 'December'],linewidth=1,
    edgecolor='black')
plt.title('No of Bookings Month-wise',fontsize=20)
plt.xlabel('Months',fontsize=15)
plt.ylabel('No of Reservations',fontsize=15)
plt.xticks(rotation=45)
```

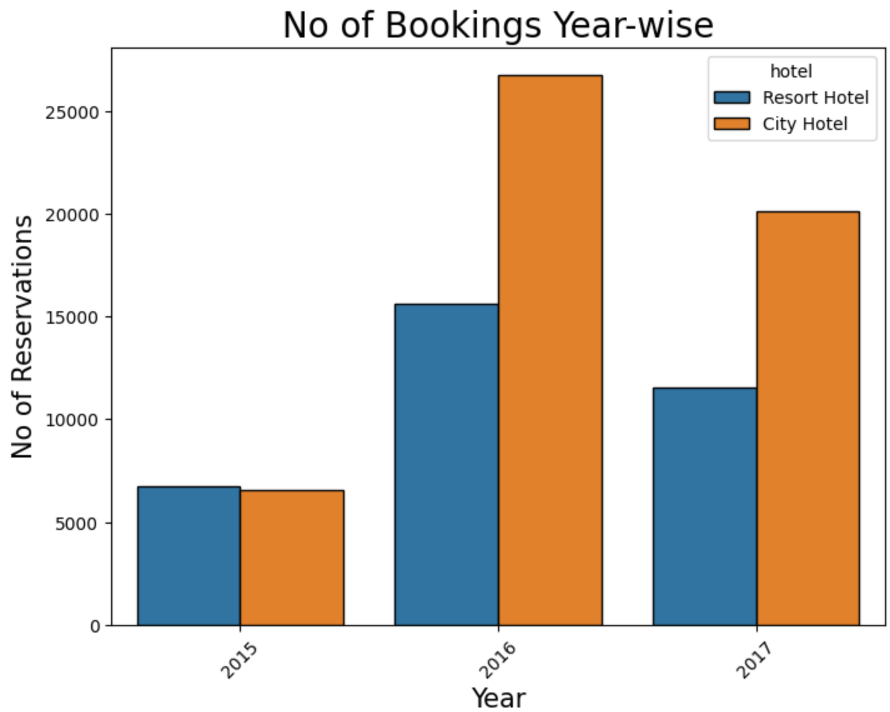
```
July         10057
May           8355
April         7908
June          7765
March         7513
October       6934
September     6690
February      6098
December      5131
November      4995
January       4693
Name: arrival_date_month, dtype: int64
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]),
 [Text(0, 0, 'January'),
  Text(1, 0, 'February'),
  Text(2, 0, 'March'),
  Text(3, 0, 'April'),
  Text(4, 0, 'May'),
  Text(5, 0, 'June'),
  Text(6, 0, 'July'),
  Text(7, 0, 'August'),
  Text(8, 0, 'September'),
  Text(9, 0, 'October'),
  Text(10, 0, 'November'),
  Text(11, 0, 'December')])
```

A double bar graph is a graphical display of information using two bars besides each other at various heights. The bars can be arranged vertically or horizontally. We can use a double bar graph to compare two data groups. Maximum bookings are made in the months of May , July and August. Minimum bookings are made in January , February , November and December . Marketing strategies can be improved to increase booking rates in these months

Chart - 5 : Booking Trends Year-Wise

```
# Chart - 4 visualization code
# Understanding the impact on Hotel business Year-Wise
print(df_new['arrival_date_year'].value_counts())

#Plotting seaborn bar chart to visualize no of bookings hotel wise as well as month wise.
plt.figure(figsize=(8,6))
graph=sns.countplot(data= df_new, x='arrival_date_year', hue = 'hotel',linewidth=1,edgecolor='black')
plt.title('No of Bookings Year-wise',fontsize = 20)
plt.xlabel('Year',fontsize = 15)
plt.ylabel('No of Reservations', fontsize = 15)
plt.xticks(rotation=45)
```

```
2016    42391
2017    31692
2015    13313
Name: arrival_date_year, dtype: int64
(array([0, 1, 2]),
 [Text(0, 0, '2015'), Text(1, 0, '2016'), Text(2, 0, '2017')])
```

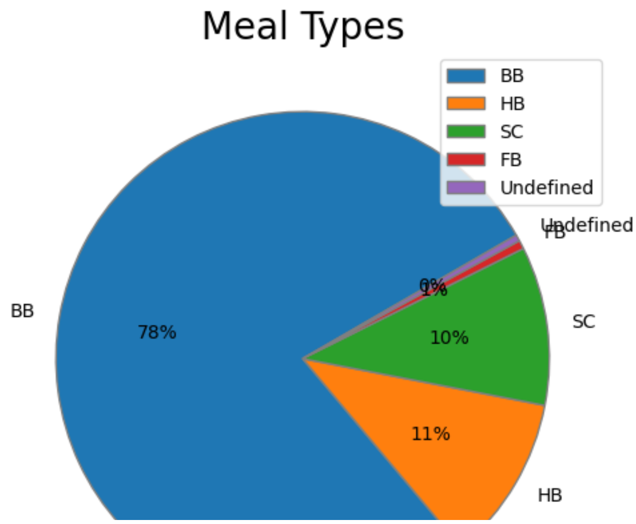


Double Bar Chart to understand the booking trends in both hotels year-wise.Maximum bookings were made in the year 2016 in both types of hotels, followed by year 2017 and least bookinngs were made in 2015.

Chart - 6 : Meal Type Preference

```
# Chart - 5 visualization code
# Understanding the meal type preference among customers.
meal=df_new['meal'].value_counts()
print(meal)
meal_labels= ['BB','HB','SC','FB', 'Undefined']

# Plotting pie-chart
plt.figure(figsize=(8,6))
plt.pie(meal, labels=meal_labels,  autopct='%0.0f%%',startangle=30,wedgeprops = { 'linewidth' : 1, 'edgecolor' : 'gray' })
plt.title('Meal Types', fontsize = 20)
plt.legend(meal_labels)
plt.show()
```

```
BB           67978
SC            9481
HB            9085
Undefined      492
FB             360
Name: meal, dtype: int64
```



A pie chart helps organize and show data as a percentage of a whole. From the chart we derive that most customers prefer BB Meal followed by HB and SC meal-type.Pnly 0.4% customers opt for FB meals. FB meals can be promoted by providing complimentary dessert.

Chart - 7 : Top 10 Countries for Bookings

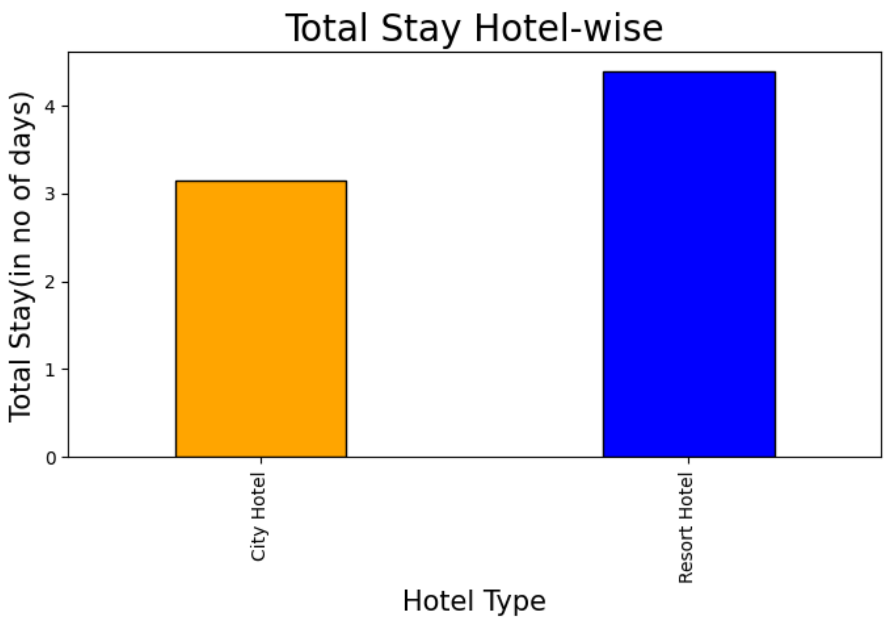
```
# Chart - 6 visualization code
#Understanding maximum business is happening through which country customers
country_wise_bookings = df_new[df_new['is_canceled'] == 0]['country'].value_counts().reset_index()
country_wise_bookings.columns = ['Country','No_of_bookings']
top_10_country_bookings = country_wise_bookings.head(10)
top_10_country_bookings

#Plotting seaborn bar chart to visualize the trend of bookings vs country
plt.figure(figsize=(8,6))
sns.barplot(x=top_10_country_bookings['Country'],y=top_10_country_bookings['No_of_bookings'])
plt.title('Top 10 Countries for Bookings', fontsize = 20)
plt.legend(top_10_country_bookings['Country'])
plt.xticks(rotation=45)
plt.show()
```

Top countries contributing in the hotel business are PRT,GBR,FRA,ESP,DEU,IRL,ITA,BEL,NLD,USA in the same order,PRT being the highest accounting for 17500 bookings.

Chart - 8 : Total Stay Hotel-wise

```
# Chart - 8 visualization code
# Understanding the stay trends in both types of hotels.
l1 = df_new['total_stay'].unique()
l2=l1.tolist()
ax = df_new.groupby(['hotel'])['total_stay'].mean().plot.bar(color=['#FFA500','blue'],width =0.4,linewidth=1,
    edgecolor='black',
    figsize = (8,4),
    fontsize = 10,)
ax.set_title('Total Stay Hotel-wise',fontsize=20)
ax.set_xlabel('Hotel Type',fontsize=15)
ax.set_ylabel('Total Stay(in no of days)',fontsize=15)
```

```
Text(0, 0.5, 'Total Stay(in no of days)')
```

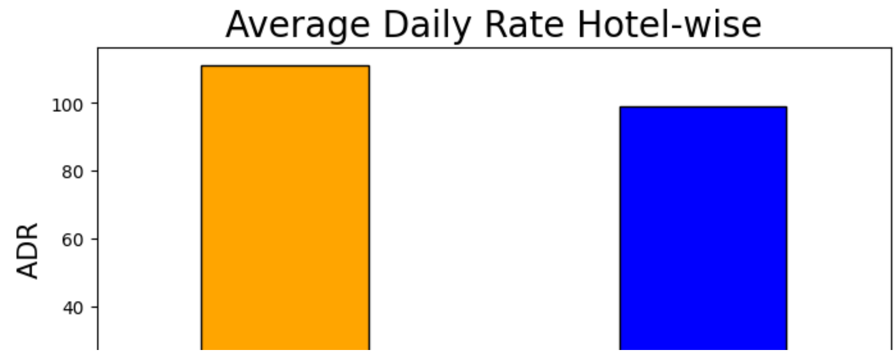


Total stay is more in Resort Hotels as compared to city hotels.On an average,customers prefer to stay for 4 days in Resort Hotels and 3 days in City hotels.

Chart - 9 : Average Daily Rate Hotel-Wise

```
#Understanding the Average Daily Rates of City Hotel vs Resort Hotel.
ax = df_new.groupby(['hotel'])['adr'].mean().plot.bar(color=['#FFA500','blue'],width =0.4,linewidth=1,
    edgecolor='black',
    figsize = (8,4),
    fontsize = 10,)
ax.set_title('Average Daily Rate Hotel-wise',fontsize = 20)
ax.set_xlabel('Hotel Type',fontsize=15)
ax.set_ylabel('ADR ',fontsize=15)
plt.xticks(rotation =0)
```

```
(array([0, 1]), [Text(0, 0, 'City Hotel'), Text(1, 0, 'Resort Hotel')])
```

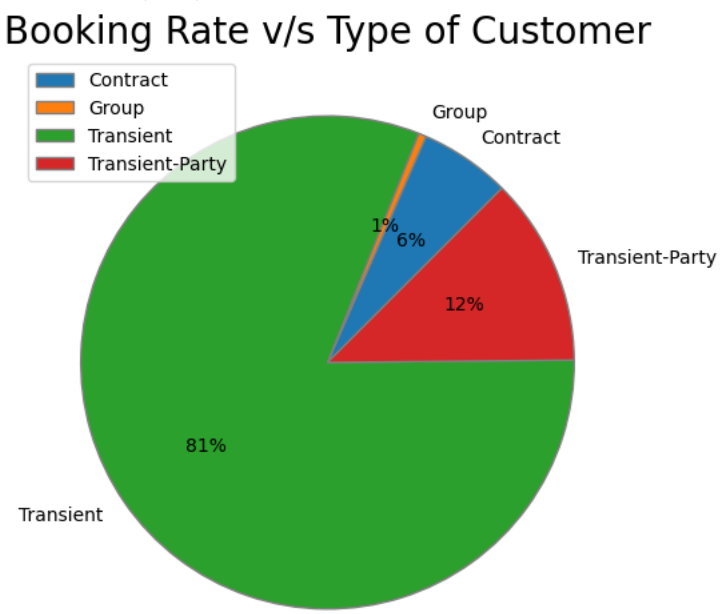


ADR for City Hotels is 110.98 whereas for Resort Hotels it is 99.02% . Daily Rates are high for City hotels in comparison to Resort Hotels.

Chart - 10 : Booking Rate v/s Type of Customer

```
#Booking Rate v/s Type of Customers.

total_stay_customer_wise=df_new.groupby('customer_type')['total_stay'].sum()
print(total_stay_customer_wise)
label= ['Contract','Group','Transient','Transient-Party']

plt.figure(figsize=(8,6))
plt.pie(total_stay_customer_wise, labels=label,  autopct='%0.0f%%',startangle=45,wedgeprops = { 'linewidth' : 1, 'edgecolor' : 'gray' })
plt.title('Booking Rate v/s Type of Customer', fontsize = 20)
plt.legend(label)
plt.show()
```

```
customer_type
Contract            18942
Group                1589
Transient          257609
Transient-Party     39165
Name: total_stay, dtype: int64
```

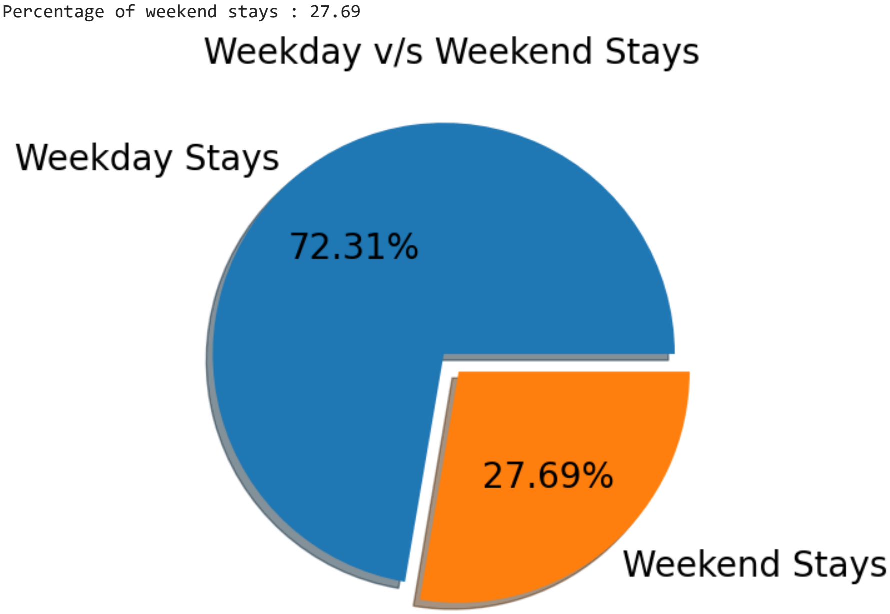


Transient : 81 % Transient-Party : 12% Contract : 6% Group : 1% Maximum bookings made by transient customers and minimum by Group-Type Customers.

Chart - 11 : Weekday v/s Weekend Stay Trends

```
# Chart - 8 visualization code
#Observing weekday vs weekend stay patterns to understand whether more bookings are made in weekdays or weekends.
count_of_weekday_stays =df_new['stays_in_week_nights'].sum()
count_of_weekend_stays =df_new['stays_in_weekend_nights'].sum()
total_stay_in_hotels = df_new['total_stay'].sum()
print('Total stay in weekdays :', count_of_weekday_stays)
print('Total stay in weekends :', count_of_weekend_stays)
print('Total stay : ',total_stay_in_hotels)

#Calculating weekday and weekend stay percentages.
weekday_stay_percent = (count_of_weekday_stays/total_stay_in_hotels)*100
weekend_stay_percent = (count_of_weekend_stays/total_stay_in_hotels)*100
print('Percentage of weekday stays :',round(weekday_stay_percent,2))
print('Percentage of weekend stays :',round(weekend_stay_percent,2))

#Plotting weekend vs weekday stay data
stay_data = [weekday_stay_percent,weekend_stay_percent]
label =['Weekday Stays','Weekend Stays']
plt.figure(figsize=(8,6))
plt.pie(stay_data,labels=label,explode=[0.05,0.05],shadow=True,autopct='%0.2f%%',textprops={'fontsize': 20})
plt.title('Weekday v/s Weekend Stays',fontsize = 20)
plt.show()
```

```
Total stay in weekdays : 229449
Total stay in weekends : 87856
Total stay :  317305
Percentage of weekday stays : 72.31
Percentage of weekend stays : 27.69
```

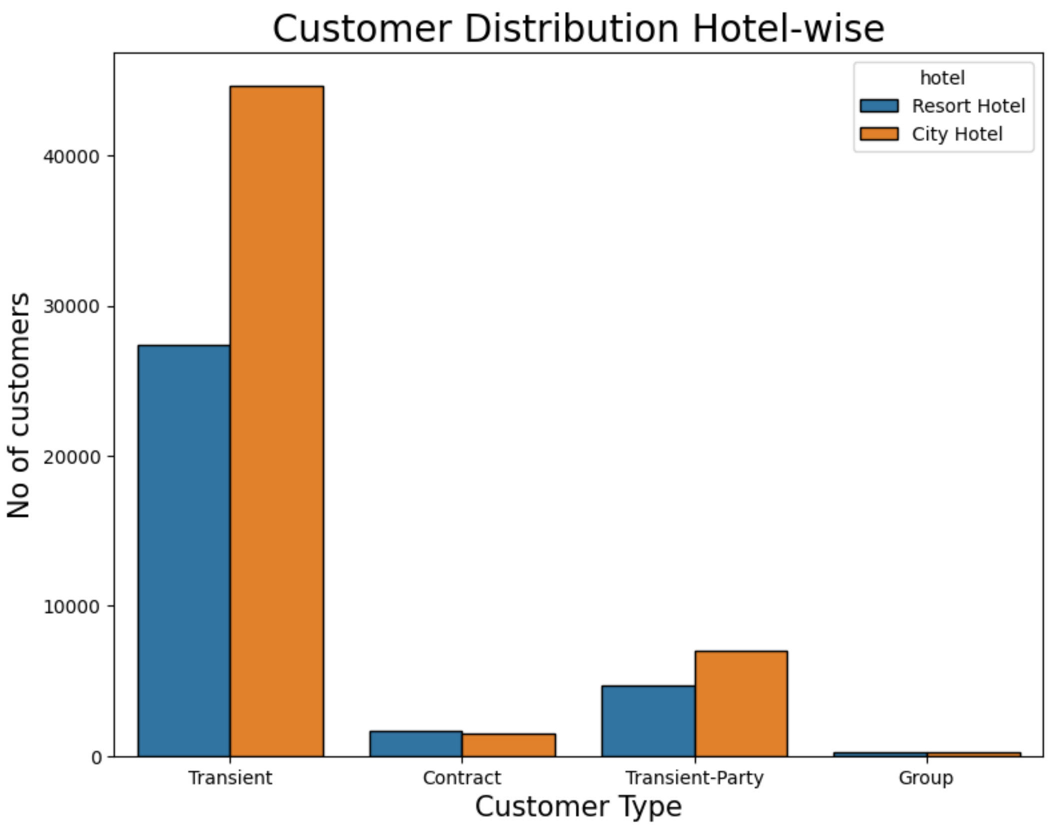


Customers prefer weekday stay to weekend stay. Almost 72% customers book during weekdays which may be due to corporate bookings ,other 28% prefer weekend stays.

Chart - 12 : Customer Distribution Hotel-wise

```
# Chart - 9 visualization code
#Customer Distribution based on hotel type
df = df_new.groupby(['hotel','customer_type']).size()
print(df)
plt.figure(figsize=(8,6))
graph=sns.countplot(data= df_new, x='customer_type', hue = 'hotel',linewidth=1,edgecolor='black')
plt.tight_layout()
plt.title('Customer Distribution Hotel-wise',fontsize = 20)
plt.xlabel('Customer Type',fontsize =15)
plt.ylabel('No of customers', fontsize = 15)
plt.show()
```

```
hotel         customer_type
City Hotel    Contract           1471
              Group               271
              Transient         44641
              Transient-Party    7045
Resort Hotel  Contract           1668
              Group               273
              Transient         27345
              Transient-Party    4682
dtype: int64
```

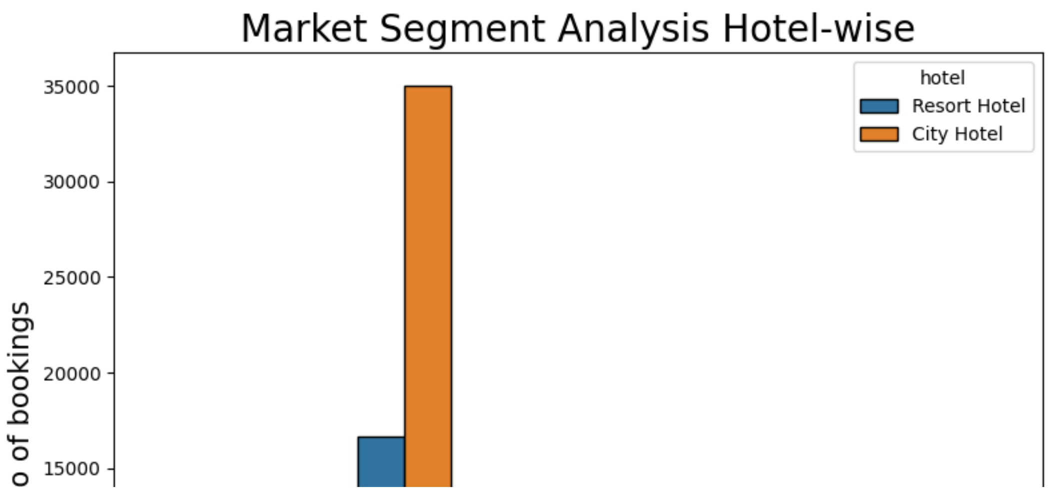


Maximum bookings are made by Transient customers in both types of hotels . Minimum bookings are made by Group customers. Hence, to attract group customers as they will result in better daily rates , hotels can provide redeemable points/ discounts to families or group bookings .

Chart-13 : Market Segment Analysis Hotel-wise

```
#Market Segment Analysis Hotel wise.
df = df_new.groupby(['hotel','market_segment']).size()
print(df)
plt.figure(figsize=(8,6))
graph=sns.countplot(data= df_new, x='market_segment', hue = 'hotel',linewidth=1,edgecolor='black')
plt.tight_layout()
plt.title('Market Segment Analysis Hotel-wise', fontsize = 20)
plt.xlabel('Market Segment',fontsize = 15)
plt.ylabel('No of bookings',fontsize = 15)
plt.show()
```

```
hotel         market_segment
City Hotel    Aviation            227
              Complementary       513
              Corporate          2227
              Direct             5559
              Groups             2635
              Offline TA/TO      7271
              Online TA         34994
              Undefined             2
Resort Hotel  Complementary       189
              Corporate          1985
              Direct             6245
              Groups             2307
              Offline TA/TO      6618
              Online TA         16624
dtype: int64
```

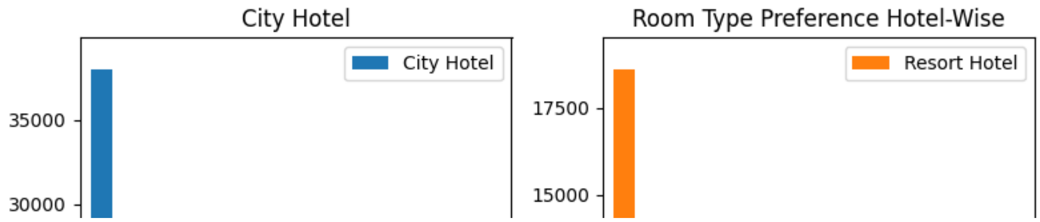


Maximum bookings are generated by Online TA and minimum through Aviation sector. Marketing strategies to promote business in declining segments is suggested.

Chart - 14 : Room Type Preference Hotel-Wise

```
# Chart - 10 visualization code
#Room Type Preferences in Both types of Hotels
df1 = df_new.groupby(['hotel','reserved_room_type']).size()
print(df1)

ax = df1.unstack(level=0).plot(kind='bar', subplots=True, rot=0, figsize=(8, 6), layout=(1, 2))
plt.tight_layout()
plt.title('Room Type Preference Hotel-Wise')
plt.xlabel('Reserved Room Type')
plt.ylabel('No of guests')
plt.show()
```

```
hotel         reserved_room_type
City Hotel    A                     37942
              B                       996
              C                        14
              D                     10766
              E                      1470
              F                      1757
              G                       479
              P                         4
Resort Hotel  A                     18610
              B                         3
              C                       901
              D                      6632
              E                      4579
              F                      1066
              G                      1573
              H                       596
              L                         6
              P                         2
dtype: int64
```



Subplotting is a technique for creating multiple plots that live side-by-side in one overall figure. We can use the subplots method to create a figure with multiple subplots. subplots takes two arguments. The first one controls the number of rows, the second one the number of columns. City Hotel : (Most Preferred : A,D, Least Preferred : G, Negligible Bookings : C,H ,L,P) Resort Hotel : (Most Preferred : A,D,E, Least Preferred : H, Negligible Bookings : B,L,P) The Hotels should inspect why customers are not preferring some room types.

Chart-15 : Parking Space Preference Hotel-Wise

```
#Parking Preference Hotel-Wise
df1 = df_new.groupby(['hotel','required_car_parking_spaces']).size()
print(df1)

ax = df1.unstack(level=0).plot(kind='bar', subplots=True, rot=0, figsize=(8, 5), layout=(1, 2))
plt.tight_layout()
plt.title('Parking Space Preference Hotel-Wise')
plt.xlabel('Parking Space ID')
plt.ylabel('No of parkings')
plt.show()
```

```
hotel          required_car_parking_spaces
City Hotel     0                              51532
               1                               1891
               2                                  3
               3                                  2
Resort Hotel   0                              28551
               1                               5389
               2                                 25
               3                                  1
```

Parking spaces are not usually required by customers in both hotels. More customers prefer parking spaces in Resort Hotels as compared to City Hotels. Mostly customers prefer 1 or 2 parking spaces.

Chart- 16 : No of children v/s Special Requests

```
# Understanding the relationship between Children and Total Special Requests made

plt.figure(figsize=(8,4))
ax = df_new.groupby(['children'])['total_of_special_requests'].mean().plot.bar()
plt.tight_layout()
plt.title('No of children v/s Special Requests', fontsize = 20)
plt.xlabel('No of children',fontsize = 15)
plt.ylabel('Avg No of Special Requests',fontsize = 15)
plt.show()
```

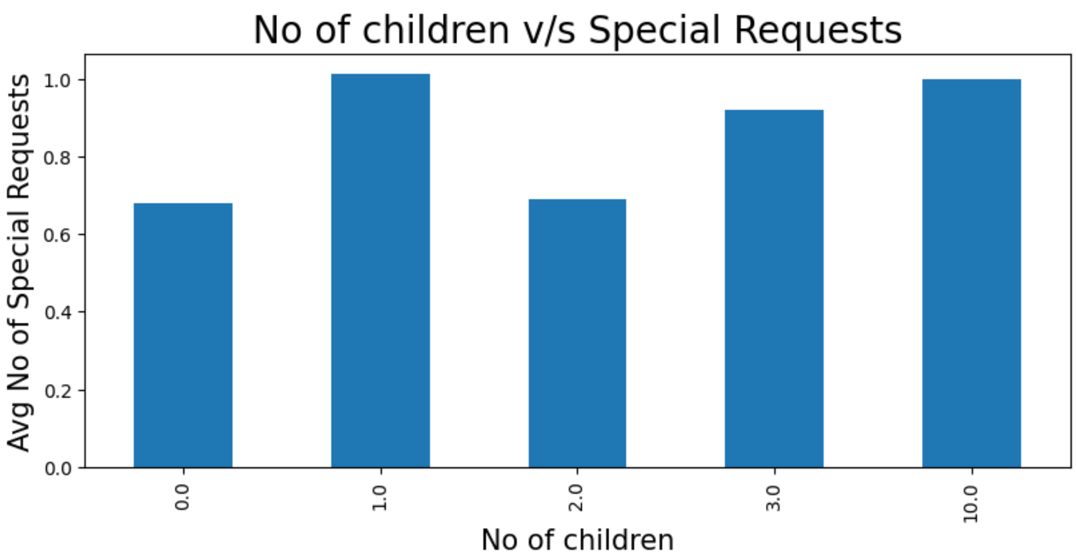


Chart-17 : No of babies v/s Special Requests

```
# Understanding the relationship between no of babies and Total Special Requests made
plt.figure(figsize=(8,4))
ax = df_new.groupby(['babies'])['total_of_special_requests'].mean().plot.bar()
plt.tight_layout()
plt.title('No of babies v/s Special Requests', fontsize = 20)
plt.xlabel('No of babies',fontsize = 15)
plt.ylabel('Avg No of Special Requests',fontsize = 15)
plt.show()
```

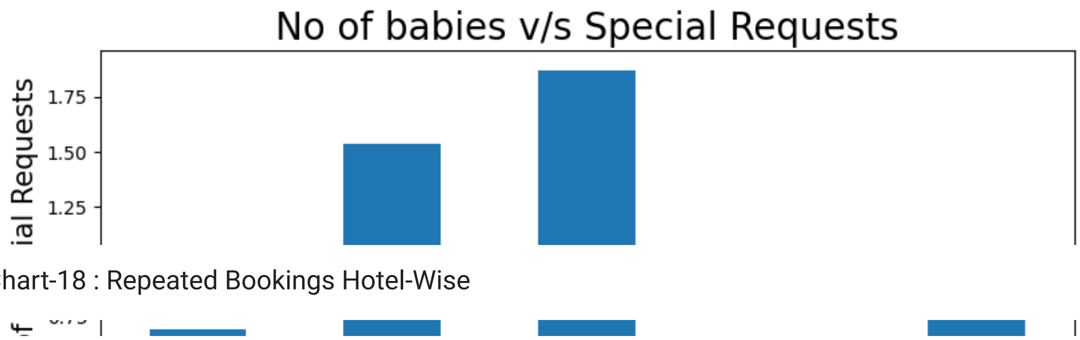
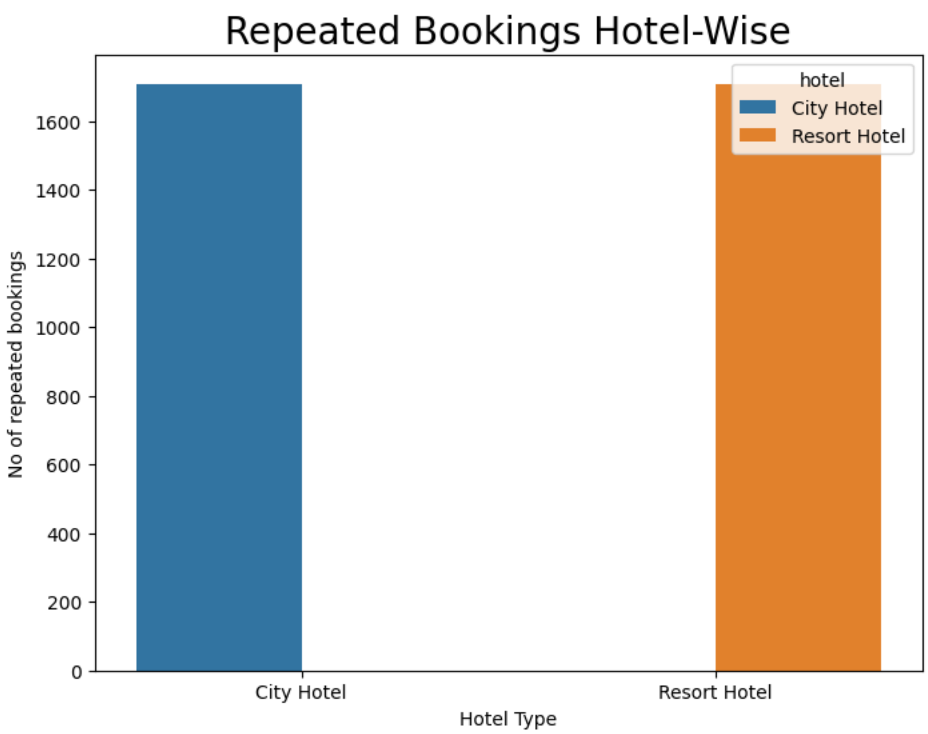


Chart-18 : Repeated Bookings Hotel-Wise

```
#Understanding the trends of repeated bookings hotel-wise.
repeated_guests_df=df_new[df_new['is_repeated_guest']==1]
repeated_guests_df=repeated_guests_df.groupby(['hotel']).size().reset_index().rename(columns={0:'counts'})
#set plotsize and call the barplot function
plt.figure(figsize=(8,6))
sns.barplot(x='hotel',y='counts',hue="hotel",data= repeated_guests_df)
plt.title('Repeated Bookings Hotel-Wise',fontsize = 20)
plt.xlabel('Hotel Type')
plt.ylabel('No of repeated bookings')
plt.show()
```

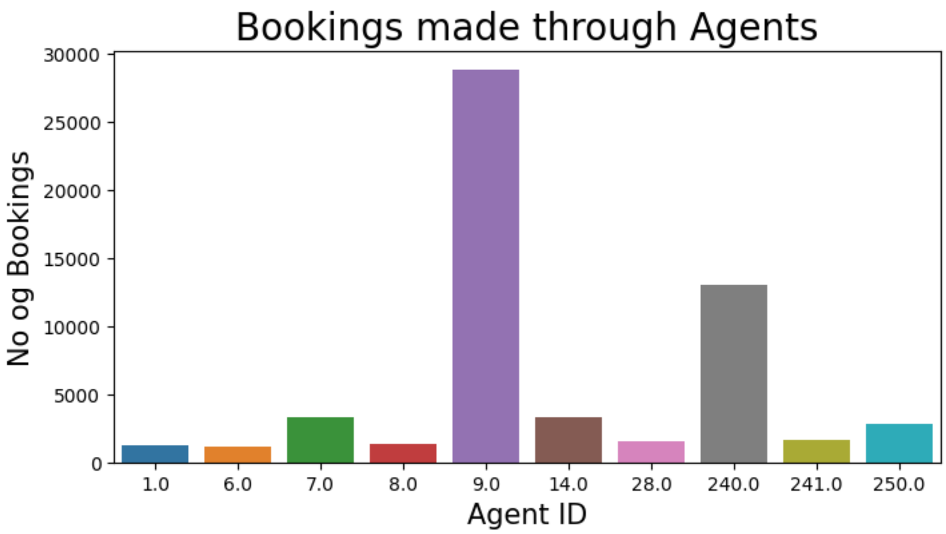


No of repeated guest are almost same in both types of hotels.

Chart - 19 : Bookings made through Agents

```
# Chart - 17 visualization code
#Understanding which agent made the maximum bookings.

bookings_made_through_agents = df_new.groupby(['agent'])['agent'].agg({'count'}).reset_index().rename(columns={'count':'No_of_Bookings'}).sor
top_10_bookings=bookings_made_through_agents[:10]
print(top_10_bookings)

plt.figure(figsize=(8,4))
sns.barplot(x=top_10_bookings['agent'],y=top_10_bookings['No_of_Bookings'])
plt.title('Bookings made through Agents',fontsize = 20)
plt.xlabel('Agent ID', fontsize = 15)
plt.ylabel('No og Bookings',fontsize = 15)
plt.show()
```

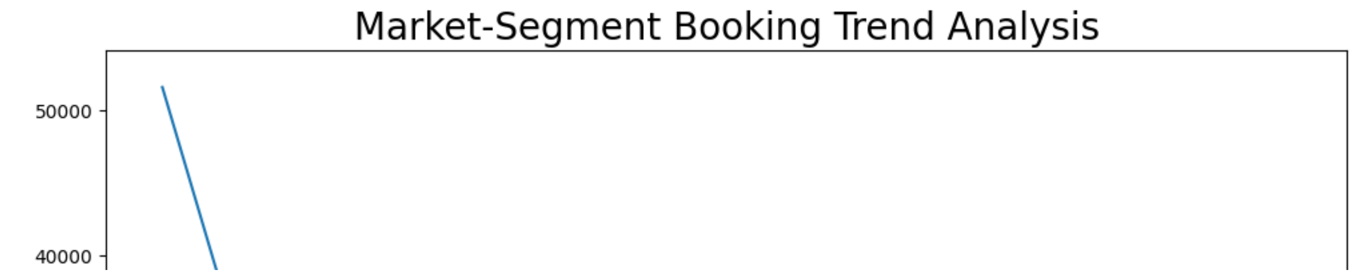
```
     agent  No_of_Bookings
8      9.0           28759
173  240.0           13028
13    14.0            3349
6      7.0            3300
182  250.0            2779
174  241.0            1644
26    28.0            1502
7      8.0            1383
0      1.0            1232
5      6.0            1117
```



Booking rates are improved when bookings are made through agents. Agent 9 has made maximum number of bookings. Agent 6 has made minimum bookings.

Chart - 20 : Market-Segment Booking Trend Analysis

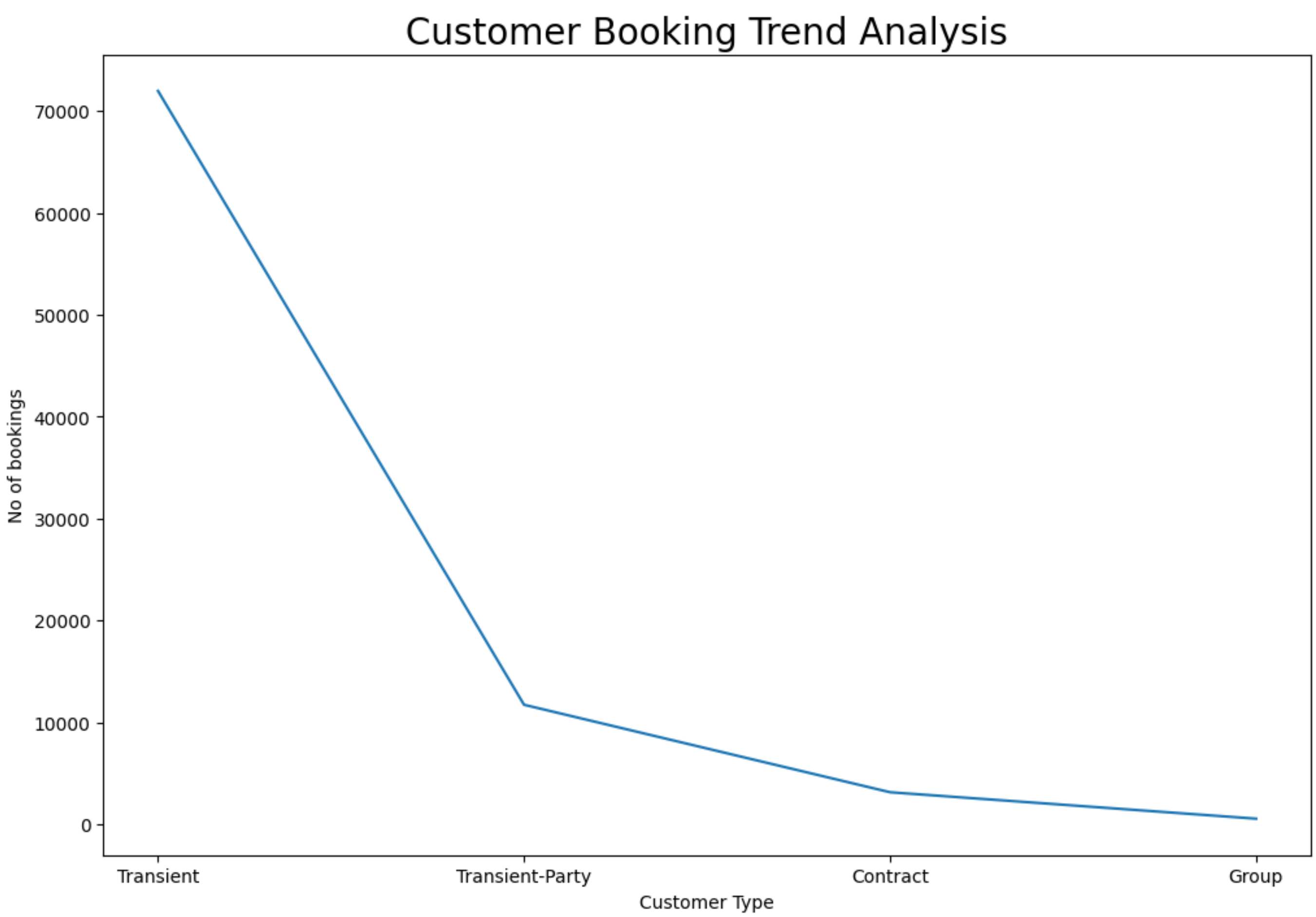
```
# Chart - 18 visualization code
#Understanding booking trends along different market segments.
plt.figure(figsize=(12,8))
plt.plot(df_new['market_segment'].value_counts())
plt.title('Market-Segment Booking Trend Analysis',fontsize = 20)
plt.xlabel('Market Segment')
plt.ylabel('No of bookings')
plt.show()
```



Maximum bookings are generated by Online TA and minimum through Aviation sector. Marketing strategies to promote business in declining segments is suggested.

Chart-21 : Customer-Type Booking Trend Analysis

```
#Customer Booking Trend Analysis
plt.figure(figsize=(12,8))
plt.plot(df_new['customer_type'].value_counts())
plt.title('Customer Booking Trend Analysis',fontsize = 20)
plt.xlabel('Customer Type')
plt.ylabel('No of bookings')
plt.show()
```



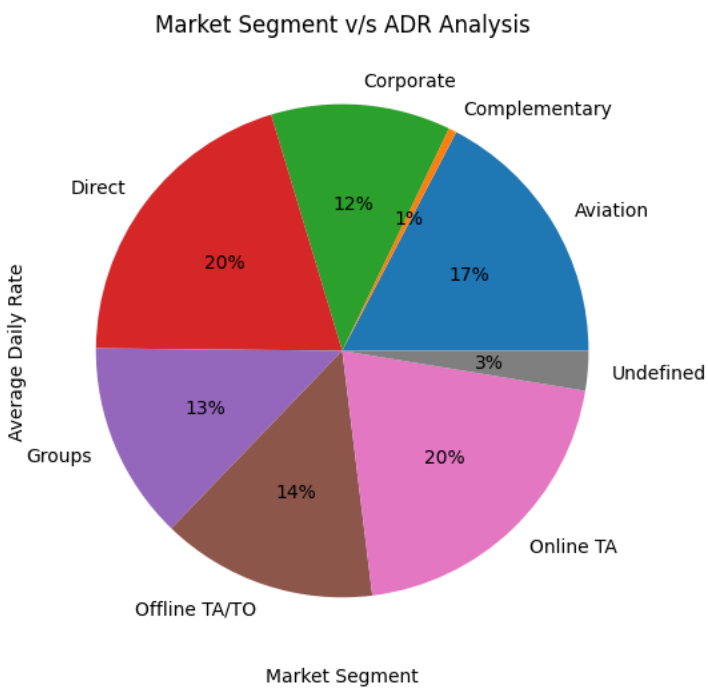
Maximum bookings are made by Transient customers in both types of hotels . Minimum bookings are made by Group customers. Hence, to attract group customers as they will result in better daily rates , hotels can provide redeemable points/ discounts to families or group bookings

```
type(df_new['market_segment'])
```

```
pandas.core.series.Series
```

Chart-22 : Market Segment v/s ADR Analysis

```
plt.figure(figsize=(6,6))
df_new.groupby(['market_segment'])['adr'].mean().plot(kind="pie",autopct='%0.0f%%')
plt.xticks(rotation=45)
plt.title('Market Segment v/s ADR Analysis')
plt.xlabel('Market Segment')
plt.ylabel('Average Daily Rate')
plt.show()
```

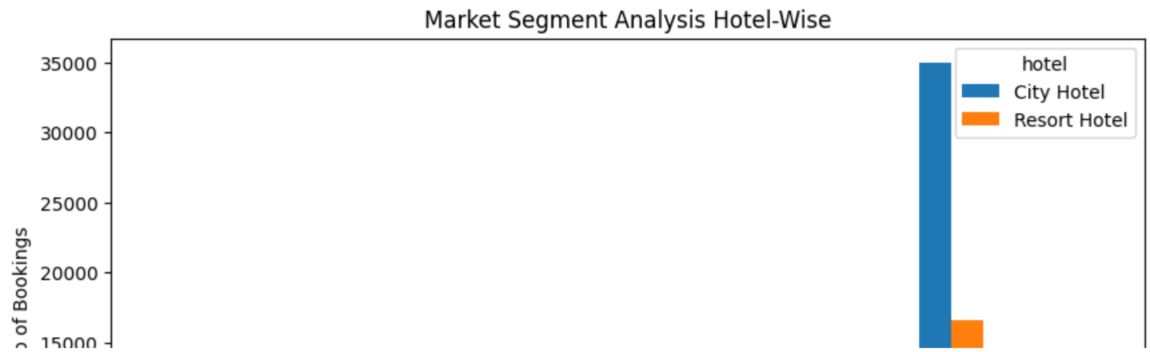


Online TA, Direct and Aviation market segments contribute to the highest ADR and Groups,Corporate and complementary has minimum daily rates

Chart - 23 : Market Segment Analysis Hotel-Wise

```
# Chart - 13 visualization code
#Understanding booking trends along different market segments hotel-wise.
df = df_new.groupby(['hotel','market_segment']).size()
df

plt.figure(figsize=(12,8))
ax = df.unstack(level=0).plot(kind='bar', rot=0, figsize=(10, 5), layout=(1, 2))
plt.xticks(rotation=45)
plt.title('Market Segment Analysis Hotel-Wise')
plt.xlabel('Market Segment')
plt.ylabel('No of Bookings')
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```

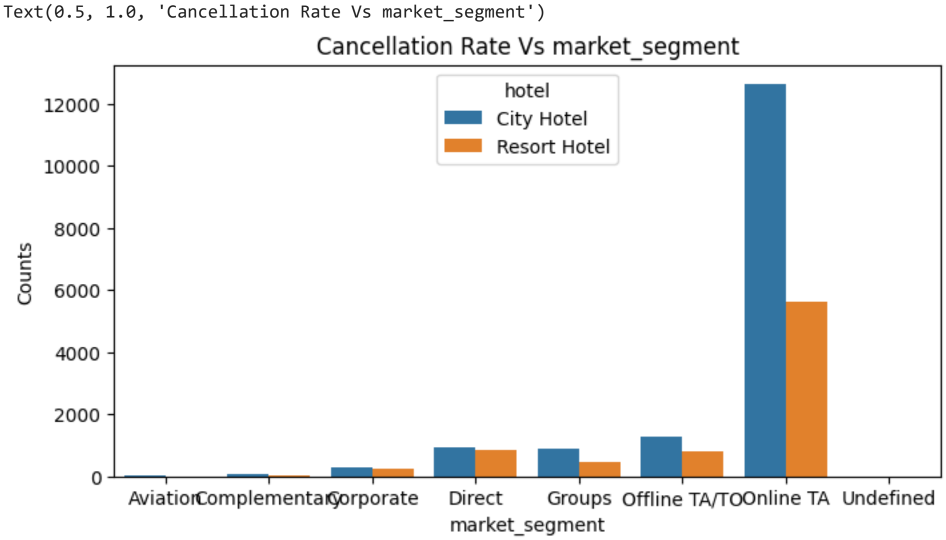


Maximum bookings are generated by Online TA and minimum through Aviation sector. Marketing strategies to promote business in declining segments is suggested.

Chart - 24 : Cancellation Rate Vs market_segment

```
#Cancellation rate in each market segment
market_segment_df=df_new[df_new['is_canceled']==1]
market_segment_df=market_segment_df.groupby(['market_segment','hotel']).size().reset_index().rename(columns={0:'counts'})   # group by
#set plotsize and call the barplot function
plt.figure(figsize=(8,4))
sns.barplot(x='market_segment',y='counts',hue="hotel",data= market_segment_df)
# set labels
plt.xlabel('market_segment')
plt.ylabel('Counts')
plt.title('Cancellation Rate Vs market_segment')
```

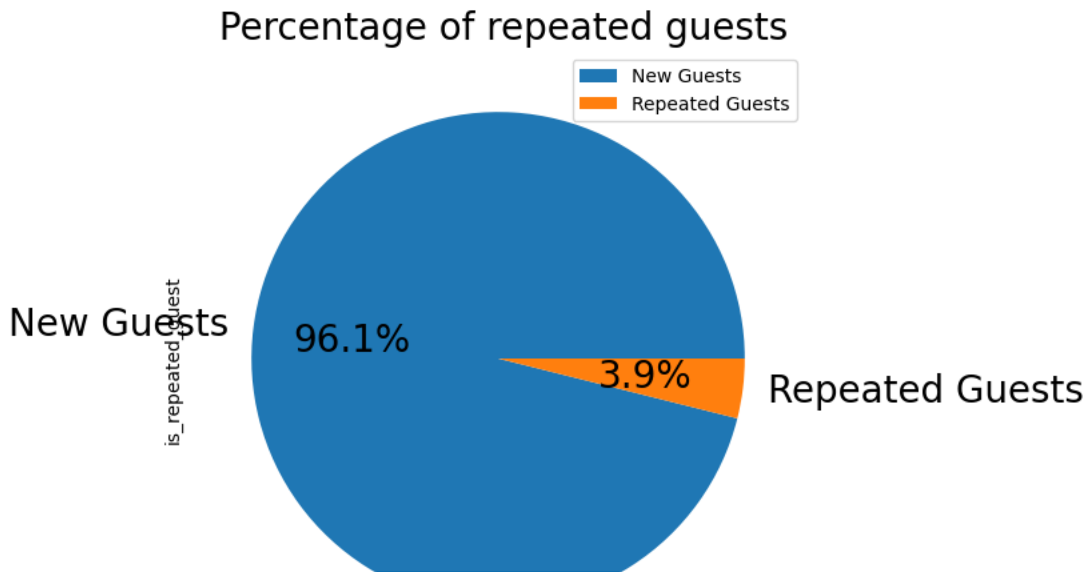
```
Text(0.5, 1.0, 'Cancellation Rate Vs market_segment')
```



Maximum cancellations are made through Online TA /TO (Travelling Allowance)market segment followed by Direct . Hotels can provide coupons and vouchers to these customers to minimize cancellations.

Chart-25 : Percentage of repeated guests

```
#plot a pie chart to see the percentage of repeated guests
label=['New Guests','Repeated Guests']
df_new['is_repeated_guest'].value_counts().plot.pie(labels=label,autopct='%1.1f%%',figsize=(8,6),fontsize=20)
plt.title(" Percentage of repeated guests",fontsize = 20 )
plt.legend(label)
plt.show()
```



Approximately 4% of customers have re booked the hotel ,96% are new guests which is an indicative of unsatisfactory stay experience. Hotels should prioritize stay experience of the guests.

Chart-26 : Cancellation Rate Vs deposit

```
#Understanding the relationship between Deposit Type and Booking Cancellation
df=df_new[df_new['is_canceled']==1]
df=df.groupby(['deposit_type','hotel']).size().reset_index().rename(columns={0:'counts'})   # group by
plt.figure(figsize=(8,4))
sns.barplot(x='deposit_type',y='counts',hue="hotel",data= df)
# set labels
plt.xlabel('deposit_type',fontsize=15)
plt.ylabel('No of cancellations',fontsize=15)
plt.title('Cancellation Rate Vs deposit',fontsize=20)
```

```
Text(0.5, 1.0, 'Cancellation Rate Vs deposit')
```

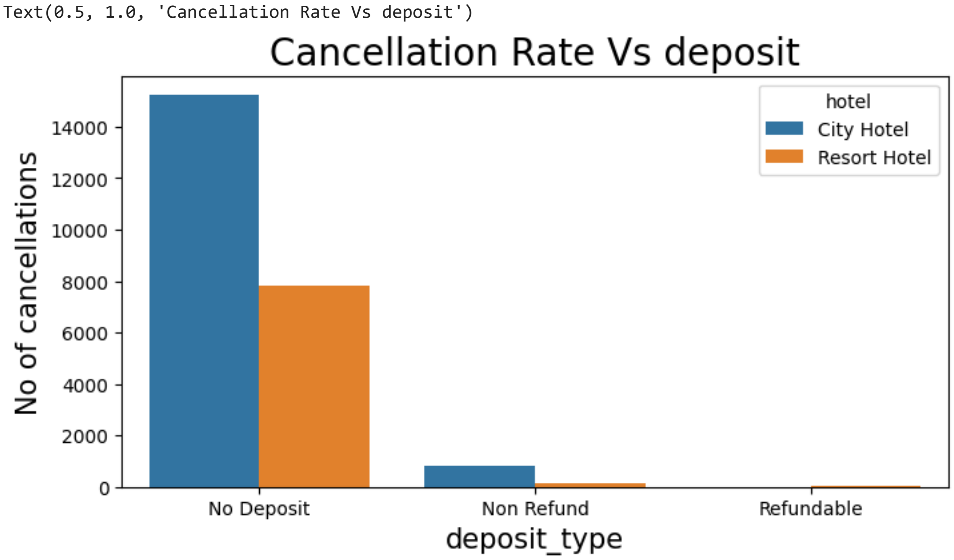


Chart - 27 - Correlation Heatmap

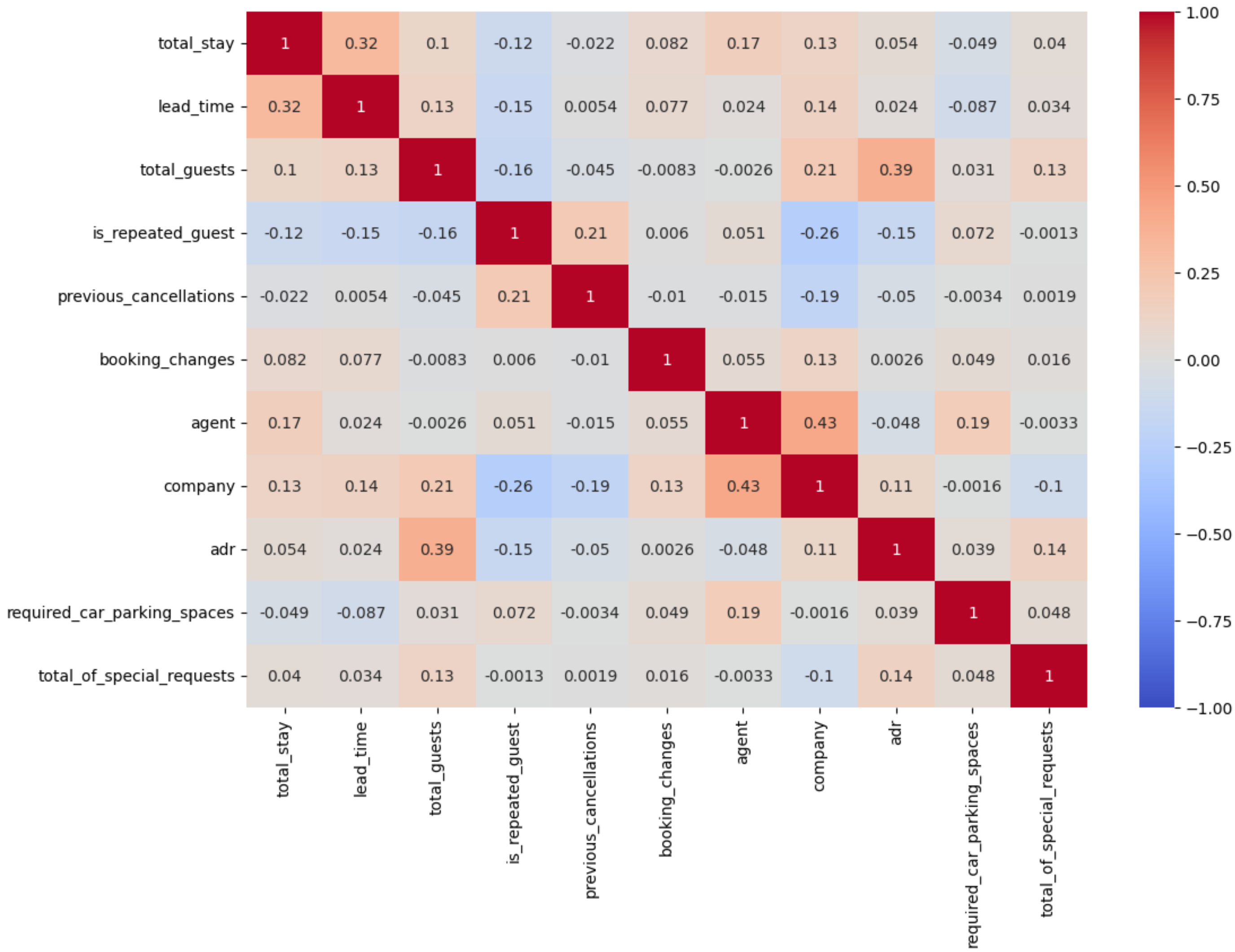
```
# Correlation Heatmap visualization code
hotel_booking_df = df_new[['hotel','total_stay','lead_time','arrival_date_month','total_guests','meal','market_segment','distribution_channe
                           'is_repeated_guest','previous_cancellations', 'reserved_room_type','assigned_room_type',
                           'booking_changes', 'deposit_type', 'agent','company', 'customer_type', 'adr',
                          'required_car_parking_spaces', 'total_of_special_requests']]
corr_df = hotel_booking_df.corr()
corr_df
```

```
<ipython-input-69-aa13f69ddf8c>:6: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version
  corr_df = hotel_booking_df.corr()
```

	total_stay	lead_time	total_guests	is_repeated_guest	previous_cancellations	booking_changes	agent	
total_stay	1.000000	0.318272	0.103584	-0.124313	-0.021663	0.081987	0.170870	(
lead_time	0.318272	1.000000	0.126751	-0.147003	0.005375	0.077028	0.023698	(
total_guests	0.103584	0.126751	1.000000	-0.163490	-0.044998	-0.008343	-0.002560	(
is_repeated_guest	-0.124313	-0.147003	-0.163490	1.000000	0.206374	0.006047	0.050939	-(
previous_cancellations	-0.021663	0.005375	-0.044998	0.206374	1.000000	-0.010269	-0.015146	-(
booking_changes	0.081987	0.077028	-0.008343	0.006047	-0.010269	1.000000	0.054959	(
agent	0.170870	0.023698	-0.002560	0.050939	-0.015146	0.054959	1.000000	(
company	0.130153	0.143630	0.205987	-0.259811	-0.190707	0.132329	0.425193	'
adr	0.054160	0.023564	0.387053	-0.153040	-0.050267	0.002552	-0.048062	(
required_car_parking_spaces	-0.048950	-0.086541	0.031426	0.072019	-0.003399	0.048718	0.186563	-(
total_of_special_requests	0.040204	0.034240	0.128083	-0.001321	0.001871	0.016115	-0.003295	-(

```
plt.figure(figsize=(12,8))
sns.heatmap(corr_df,vmin=-1, cmap='coolwarm', annot=True)
```

```
<Axes: >
```

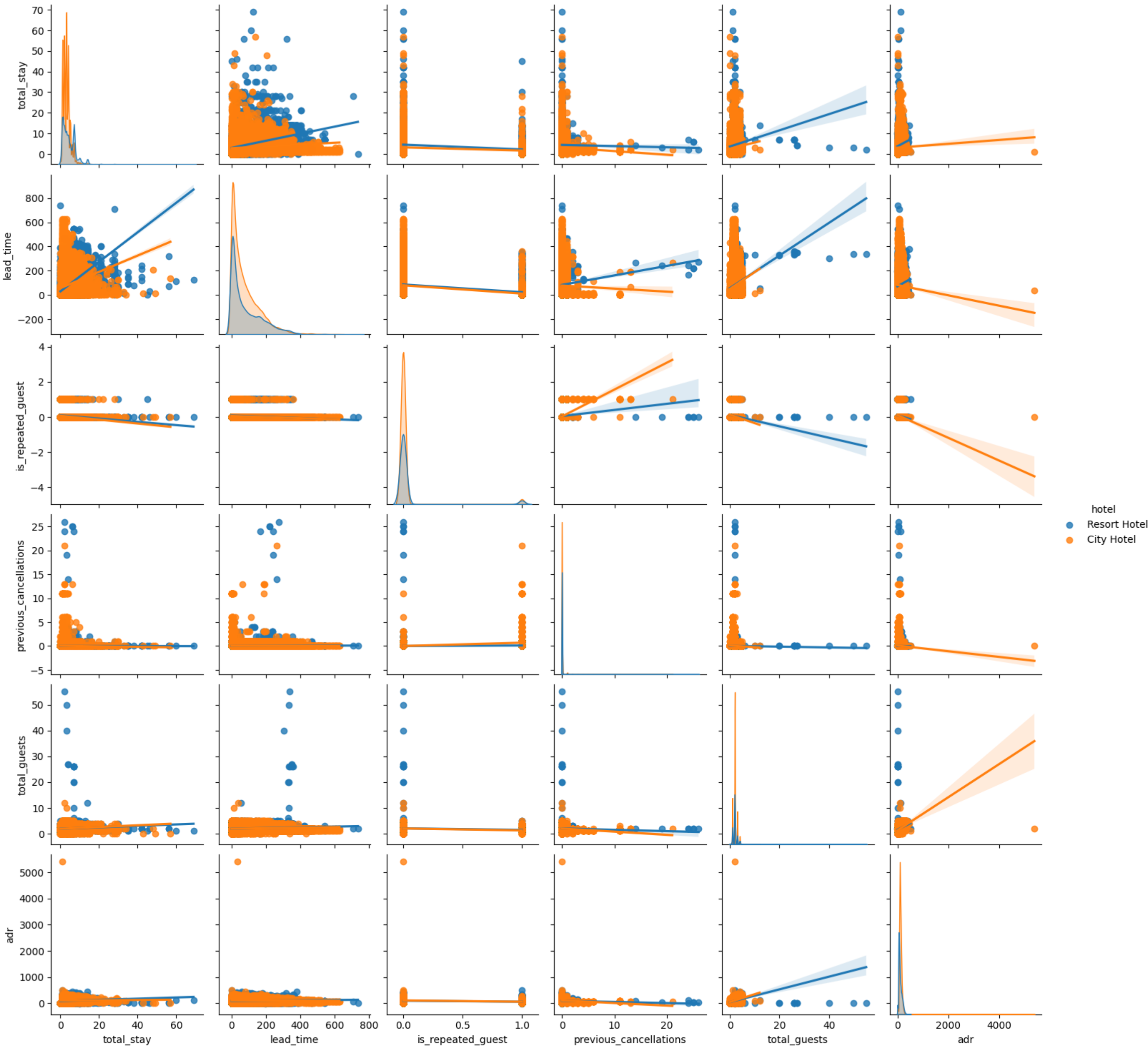


- Positive Correlation :totalstay-leadtime, totalguest-adr , agent-company

- Negative Correlation : is_repeated_guest-company, total_stay-is_repeated_guest

Chart - 28 : Pair Plot

```
# Pair Plot visualization code
#pair_plot_df = df_new[['hotel','total_stay','total_guests','meal','market_segment',
                          #'is_repeated_guest', 'reserved_room_type','assigned_room_type',
                          # 'company', 'customer_type', 'adr']]
sns.pairplot(df_new,vars =['total_stay','lead_time','is_repeated_guest','previous_cancellations','total_guests','adr'],hue ='hotel',kind='re
# to show
plt.figure(figsize=(8,6))
plt.show()
```




```
<Figure size 800x600 with 0 Axes>
```

5. Solution to Business Objective

What do you suggest the client to achieve Business Objective ?

Explain Briefly.

According to the analysis done above, following are the suggestions for the client in order to achieve business objective :

- More cancellations are occurring in City hotels, so need to put focus in retaining its customers.
- Deposit type has no relation with cancellation rates.
- Transient and contract type customers are booking for longer durations hence contributing more to the adr, hence client should try to retain these customers
- Group type customers are making minimum bookings hence these customers should be targeted by providing better offers and discounts.
- Car parking spaces are generally not required so less cost drainage on parking space maintenance .Customers ,if any ,generally opt for 1 or 2 parking spaces.
- Room type G in City Hotel and Room type H in Resort Hotel need better maintenance as least bookings are made in these rooms.
- BB: Bed & Breakfast. HB: Half Board (Breakfast and Dinner normally) FB: Full Board (Breakfast, Lunch and Dinner) SC: Self Catering(No meals are included) Mostly customers prefer BB type meal. Almost the same proportion opts for HB and SC and only very few opt for FB. More customers can be invited to opt for FB meal booking by providing complementary dessert or discount.
- Maximum bookings hence maximum business is getting generation through Online TA/TO, Direct and Group market segments .Hence market segment to be targeted should be aviation, complementary and corporate. There segments can be attracted by providing vouchers, discount for mass booking.
- No of cancellations are maximum in online TA and Corporates. So measures can be taken to retain these customers.
- No of repeated guests are least from Aviation sector, hence these customers can be given redeemable points for next bookings in order to retain .
- Maximum daily rates are for no of guests = 2
  Individual booking customers tend to repeat their booking. Groups of 3-4 people do not repeat bookings , so people who book as a family or group can be provided redeemable points or vouchers.
- Booking rates are least in the month of Jan, Feb, Nov, Dec. Hence marketing team should focus more on bookings in these months.

Conclusion

- City Hotels have more bookings in comparison to Resort Hotels.
- Maximum booking trends are observed in the months of July and August.January,February,November and December observes least booking.
- Maximum bookings were made in the year 2016 in both City and Resort Hotels.
- Almost 27.5% bookings are cancelled. Out of these, 39% are from City Hotels and 61% cancellations are made in Resort Hotels.
- Most preferred meal type is BB which accounts for almost 78% of meal bookings. Other meals types have bookings - HB : 11%, SC : 10%, FB-1%
- Top countries contributing in the hotel business are {PRT,GBR,FRA,ESP,DEU,IRL,ITA,BEL,NLD,USA} in the same order.
- Total stay patterns are observed as more no of days for stay in Resort Hotels than City Hotels.
- Weekday bookings are more in comparison to weekend bookings. Almost 72.31% of stays are made during the week and only 27.69% of stays are during weekends.
- Agent ID 9 has made maximum number of bookings and Agent ID 6 has made minimum number of bookings.
- Booking rate is higher when bookings are made through agents.
- Both types of Hotels have almost same no of repeated bookings.
- Room Preference Hotel-Wise
  City Hotel :

  ```
  { Most Preferred : A,D
  Least Preferred : G
  Negligible Bookings : C,H ,L,P}
  ```

  Resort Hotel :

  ```
  { Most Preferred : A,D,E
  Least Preferred : H
  ```

```
Negligible Bookings : B,L,P}
```

- Resort Hotel guests require more parking spaces as compared to City Hotel Guests.
- Transient Customer Type make most no of bookings in both hotels whereas customer type Group and Contract make least bookings.
- Maximum bookings are made through Online TA market segment and minimum bookings are made through Aviation Segment.
- Cancellations are more in Non-refundable deposit type.
- No of special requests are more when children or babies are accompanied.
- ADR is higher for City Hotel than Resort Hotel.

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