Project Name - Hotel Booking Analysis

Project Type - EDA

Contribution - Team

Team Member 1 - Aman verma

Team Member 2 - Kumar Ankit

Team Member 3 - Charmi Patel

Project Summary -

- The main objective behing this project is to explore and analyze data to discover important factors that govern the bookings and give insights to hotel management, which can perform various campaigns to boost the business and performance.
- For this project we'll be analyzing Hotel Booking data. This data set contains booking information for a city hotel and resort hotel, and includes information such as when the booking was made, length of stay, the number of adults, children, and the number of available parking spaces.

Data Description:

1. **hotel** : *Hotel(Resort Hotel or City Hotel)*
2. **is_canceled** : *Value indicating if the booking was canceled (1) or not (0)*
3. **lead_time** : *Number of days that elapsed between the entering date of the booking into the PMS and the arrival date*
4. **arrival_date_year** : *Year of arrival date*
5. **arrival_date_month** : *Month of 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) the guest stayed or booked to stay at the hotel*
9. **stays_in_week_nights** : *Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel*
10. **adults** : *Number of adults*
11. **children** : *Number of children*
12. **babies** : *Number of babies*
13. **meal** : *Type of meal booked. Categories are presented in standard hospitality meal packages:*
14. **country** : *Country of origin.`*
15. **market_segment** : *Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”*
16. **distribution_channel** : *Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”*
17. **is_repeated_guest** : *Value indicating if the booking name 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. Code is presented instead of designation for anonymity reasons.*
21. **assigned_room_type** : *Code for the type of room assigned to the booking.*
22. **booking_changes** : *Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation*

23. **deposit_type** : *Indication on if the customer made a deposit to guarantee the booking.*
24. **agent** : *ID of the travel agency that made the booking*
25. **company** : *ID of the company/entity that made the booking or responsible for paying the booking.*
26. **days_in_waiting_list** : *Number of days the booking was in the waiting list before it was confirmed to the customer*
27. **customer_type** : *Type of booking, assuming one of four categories*

1. **adr** : *Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights*
2. **required_car_parking_spaces** : *Number of car parking spaces required by the customer*
3. **total_of_special_requests** : *Number of special requests made by the customer (e.g. twin bed or high floor)*
4. **reservation_status** : *Reservation last status, assuming one of three categories*

- Canceled – booking was canceled by the customer
- Check-Out – customer has checked in but already departed
- No-Show – customer did not check-in and did inform the hotel of the reason why

1. **reservation_status_date** : *Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to understand when was the booking canceled or when did the customer checked-out of the hotel*

GitHub Link -

Problem Statement

This project contains the real world data record of hotel bookings of a city and a resort hotel containing details like bookings, cancellations, guest details etc. Main aim of the project is to understand and visualize dataset from hotel and customer point of view i.e.

- Reasons for booking cancellations across various parameters
- Best time to book hotel
- Peak season
- Dealing with missing values
- Plot the graphs and charts to get some insights about the dataset.

and give suggestions to reduce these cancellations and increase revenue of hotels.

Let's Begin !

Import Libraries

In [ ]:
```
#Importing libraries for the EDA.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn as sk
import math
%matplotlib inline
```

In [ ]:
```
#Making Connection with Google Drive
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", for
ce_remount=True).
```

Loading Dataset

In [ ]:
```
File_path = "/content/drive/MyDrive/Hotel Bookings.csv"
```

In [ ]:
```
df = pd.read_csv(File_path)
```

Dataset First View

In [ ]:
```
df.head()
```

Out[ ]:

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

5 rows × 32 columns

In [ ]:
```
df.tail()
```

Out[ ]:

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

5 rows × 32 columns

In [ ]:
```
df.describe()  #Checking the summary Statistics
```

Out[ ]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays
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	

Dataset Rows & Columns count

In [ ]:
```
df.shape  # Checking the shape of dataset
```

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

we have total 119390 rows and 32 Columns

Dataset Information

In [ ]:
```
df.info()  #fetching information about dataset
```

```
<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
```

Looking for and handling Null/ Missing Values and Outliers

In [ ]:
```
df.isna().sum().sort_values(ascending = False) #Check if our data contains any missing values in descending ord
```

Out[ ]:
```
company                         112593
agent                            16340
country                            488
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
```

Conclusion

As we can see from above column company agent, country and children contains missing values.

We can drop 'company' column as it contains too many missing values.

What did you know about your dataset?

So far we have come to know from this dataset that we have 119390 rows and 32 columns. And we have missing values also-

- in company we've 112593, in agent we've 16340, in country we've 488 and in children we've only 4 missing values.

Drop missing values

In [ ]:
```
df.drop( axis=1, columns=['company'] ,inplace=True) #dropped company column because it has too many missing val
```

In Children column we have 4 missing values, so we are filling those missing values with average value of children column and we are type-casting float values to integer value.

In [ ]:
```
df['children'].fillna(round(df['children'].mean()),inplace=True) #replacing all the missing values with the rou
df['children']= df['children'].apply(lambda x : int(x))
```

In agent column, we have 16340 missing values, so we are replacing those values with 'unknown'. Also we are typecasting the ID of Agent to integer values.

In [ ]:
```
#replacing all the missing values with 'unknown'.
df['agent'].fillna(0, inplace=True)
df['agent']= df['agent'].apply(lambda x : int(x))
df['agent']= df['agent'].apply(lambda x : 'unknown' if x==0 else x)
```

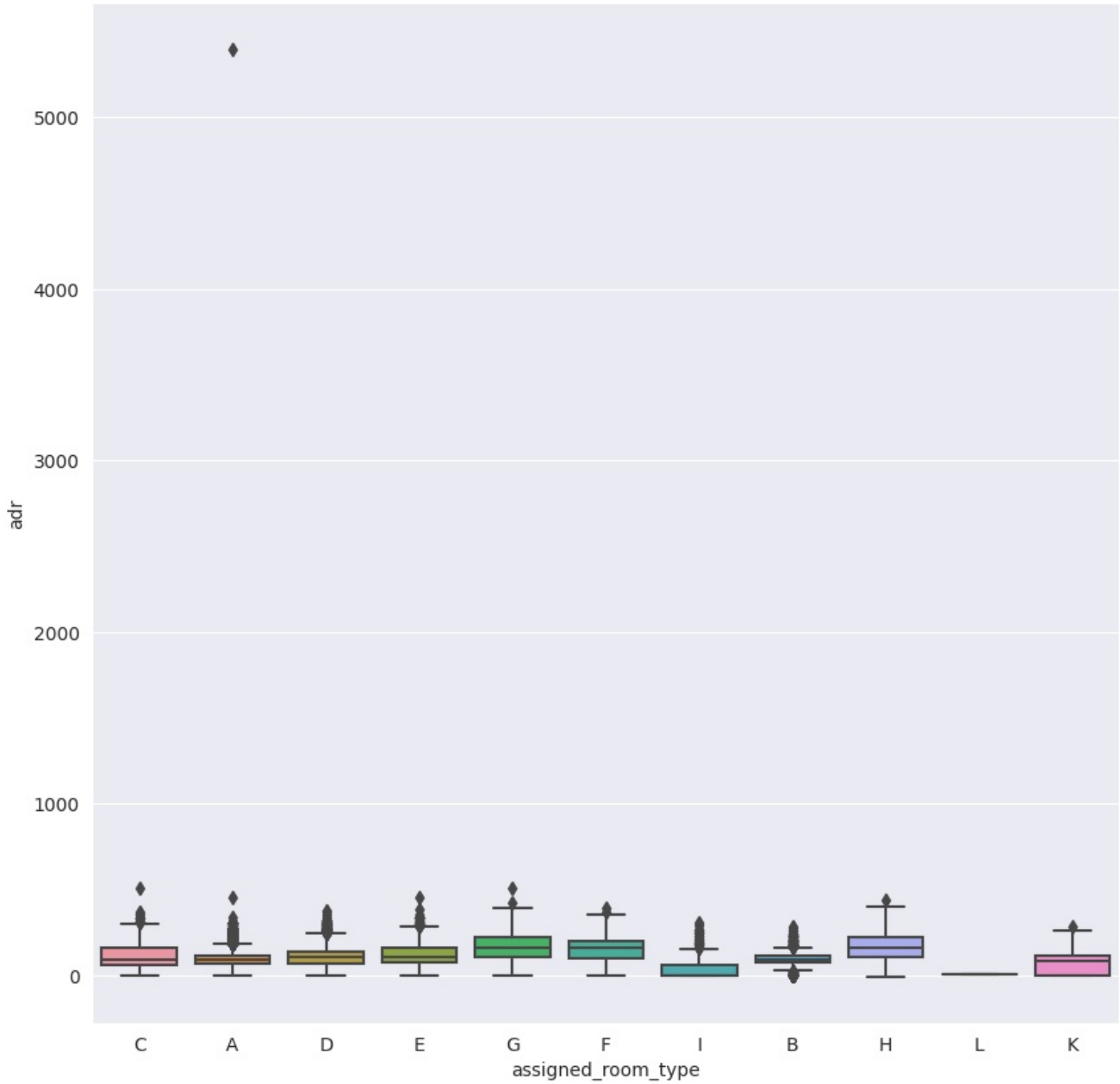
We are dropping the rows where the sum of adults, children and babies are equal to 0.

In [ ]:
```
#dropping the rows where the sum of adult,children and babies is 0.
df = df.drop(df[(df['adults']+df['children']+df['babies'])==0].index)
```

Checking for outliers in average daily rate (adr) column.

In [ ]:
```
plt.figure(figsize=(10,10))
sns.boxplot(y=df['adr'],x=df['assigned_room_type'])
```

Out[ ]:
```
<Axes: xlabel='assigned_room_type', ylabel='adr'>
```



Removing outlier data from datset where adr is greater than 1000.

In [ ]:
```
#dropping the row where the average daily rate per person is more than 1000.
df=df.drop(df[df['adr']>1000].index)
```

In [ ]:
```
#ropping the row where the average daily rate per person is less than 0.
df=df.drop(df[df['adr']<0].index)
```

Things we have done till now is:-

1.Checked Head and Tail of the dataset.

2.Analysed summary statistics of the dataset.

3.Looked for NaN/ Null/ Missing Values.

4.Filled missing values.

5.Analysed for Outliers and removed Outliers.

Now Let's try to find out the solutions for the below question.

Ques1.What is the percentage of booking done in different hotels?

Ques2.How many total bookings done in different Years?

Ques3.How many total bookings done in different months?

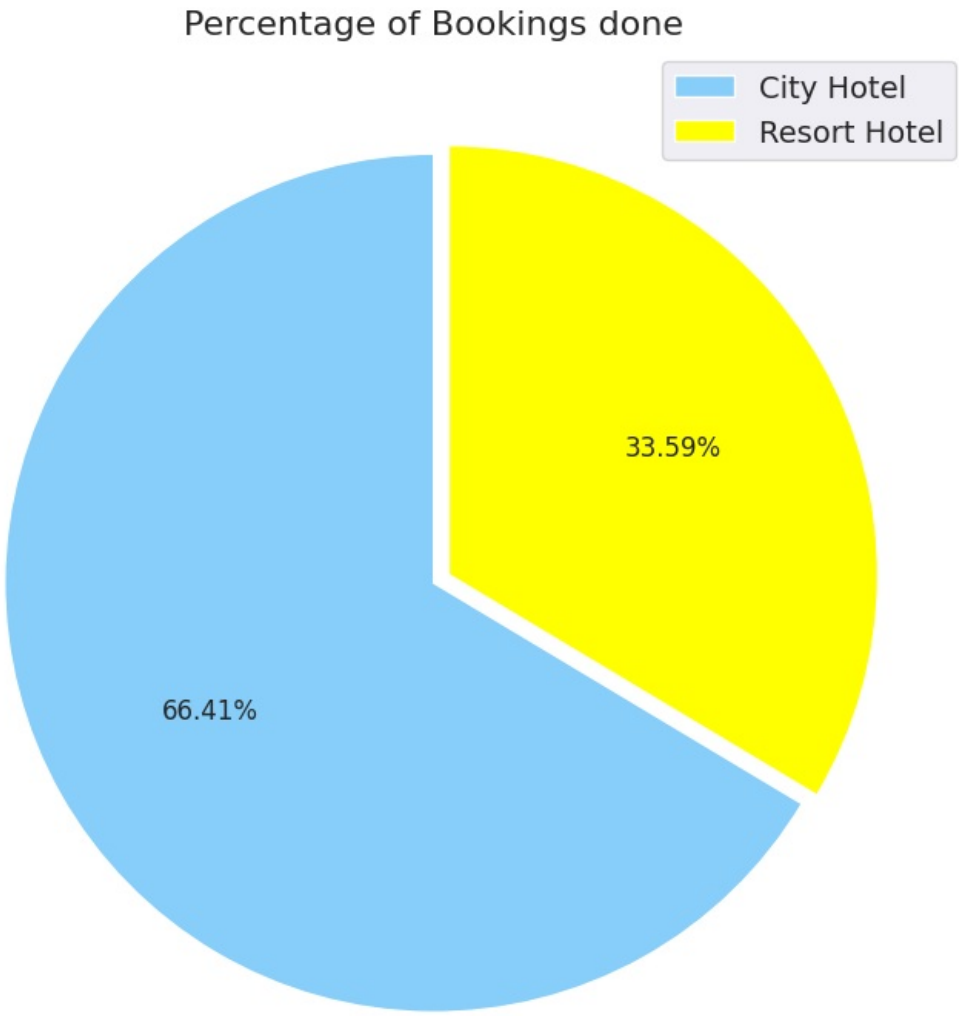
Ques4.Total Number of Booking Cancelled in different months?

Ques5.Total Number of Non-Cancelled Bookings in different months?

1. Plots of type of hotels and their booking

In [ ]:
```
#Pie-Chart for the percentage of bookings done in different types of Hotels.
plt.figure(figsize=(9,10))
labels = df['hotel'].value_counts().index.tolist()
sizes = df['hotel'].value_counts().tolist()
explode = (0, 0.04)
colors = ['lightskyblue','yellow']
plt.pie(sizes, explode=explode, colors=colors,autopct='%1.2f%%', startangle=90, textprops={'fontsize': 12})
plt.title('Percentage of Bookings done', fontsize=16)
plt.legend(labels,loc=1, prop = {'size' : 14})
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7f8f05cc39a0>
```



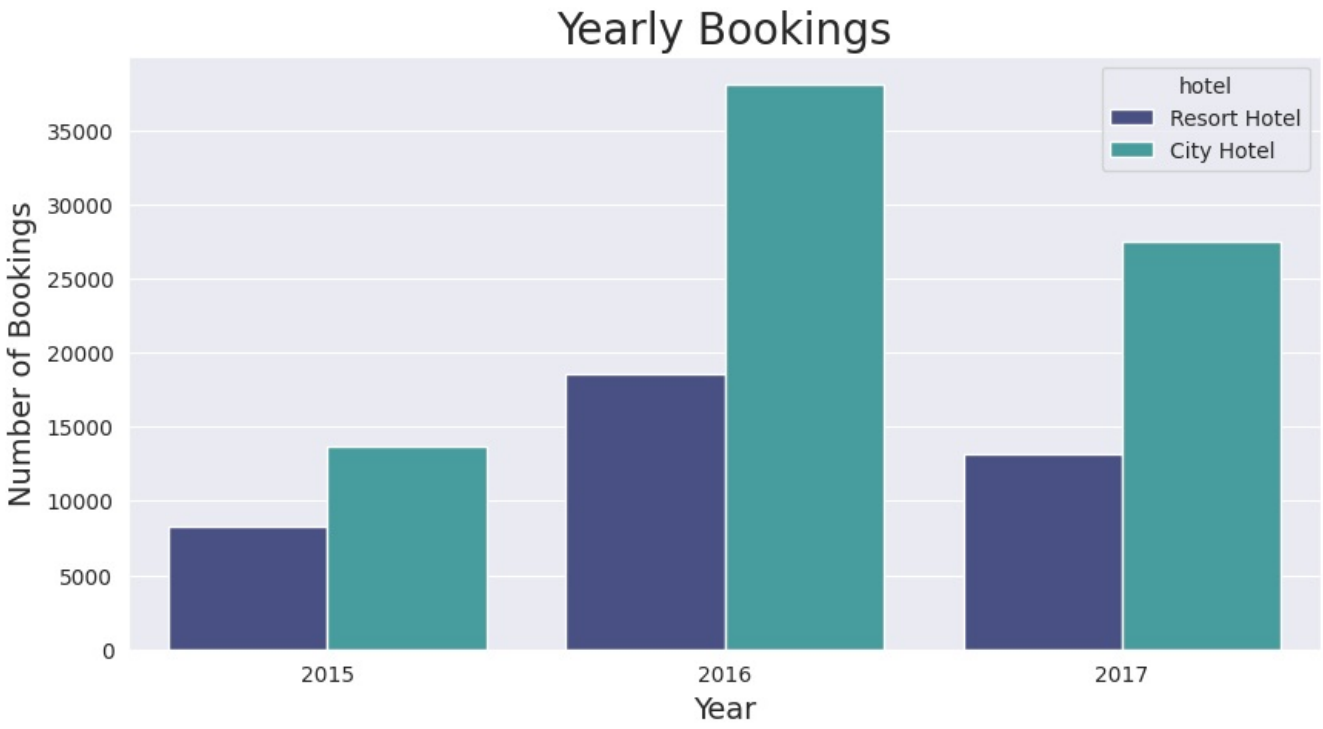
Conclusion:

From the above Pie-Chart, we can conclude that 66.41 % of booking done in City Hotel and 33.59 % of booking done in Resort Hotel.

2.Plot Year wise bookings in Hotel

In [ ]:
```
#Countplot for Number of bookings done in 2015, 2016 and 2017 in different type of hotels
plt.figure(figsize=(10,5))
sns.countplot (x= 'arrival_date_year', data= df, hue= 'hotel', palette='mako').set_title ('Yearly Bookings', fo
plt.xlabel('Year', fontsize=14)
plt.ylabel('Number of Bookings', fontsize=14)
```

Out[ ]:
```
Text(0, 0.5, 'Number of Bookings')
```



Conclusion :

We can see that, In year 2016 most bookings have been made in City Hotel and Resort Hotel.

3. Plot of Month wise Hotel Booking

In [ ]:
```
#countplot for total monthly booking in different type of hotels
plt.figure(figsize = (15, 10))
sns.countplot(x=df['arrival_date_month'], hue=df['hotel'], order=['January', 'February', 'March', 'April', 'May
          'August', 'September', 'October', 'November', 'December'],palette='crest').set_title ('Monthly Bookin
plt.xlabel('Months', fontsize=14)
plt.ylabel('Number of Bookings', fontsize=14)
plt.legend(prop={'size':15})
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7f8f060e1a20>
```

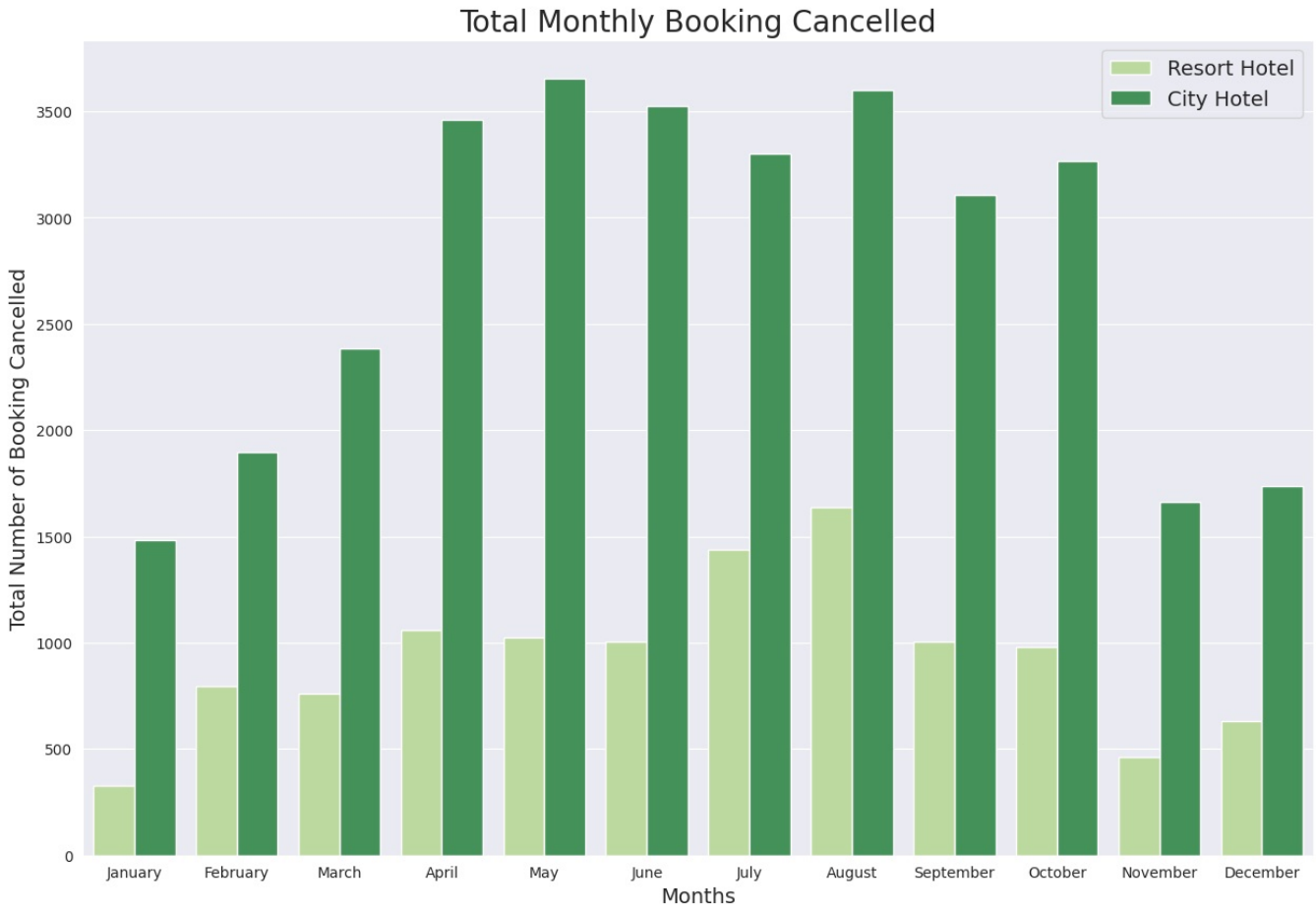
Conclusion:

Most of the bookings done in the Month of August in Resort Hotel and City Hotel.

4. Plot of Monthly cancelled.

In [ ]:
```
#Countplot of Total number of booking Cancelled month wise
df1=df[df['is_canceled']==1]
plt.figure(figsize = (15, 10))
sns.countplot(x=df1['arrival_date_month'], hue=df1['hotel'], order=['January', 'February', 'March', 'April', 'M
          'August', 'September', 'October', 'November', 'December'],palette='YlGn').set_title ('Total Monthly B
plt.xlabel('Months',fontsize=14)
plt.ylabel('Total Number of Booking Cancelled',fontsize=14)
plt.legend(prop={'size':14})
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7f8f03ea7bb0>
```



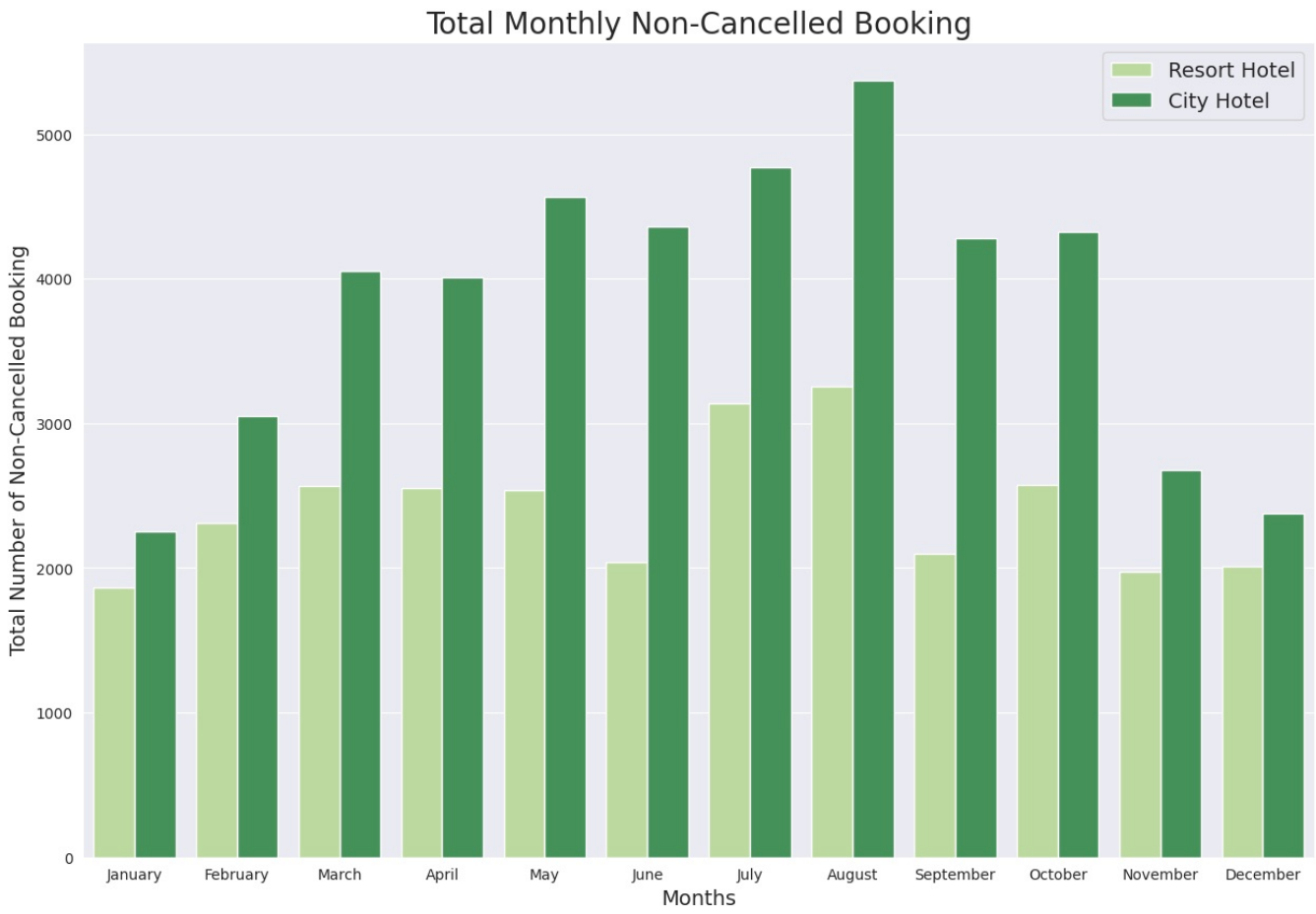
Conclusion :

In May, most of the bookings are cancelled in City Hotel and In August most of the cancellation done in Resort Hotel.

5. Plot of Monthly Actual Booking.

In [ ]:
```
#Countplot for Monthly Non-Cancelled Booking
df2=df[df['is_canceled']==0]
plt.figure(figsize = (15, 10))
sns.countplot(x=df2['arrival_date_month'], hue=df2['hotel'], order=['January', 'February', 'March', 'April', 'M
          'August', 'September', 'October', 'November', 'December'],palette='YlGn').set_title ('Total Monthly N
plt.xlabel('Months',fontsize=14)
plt.ylabel('Total Number of Non-Cancelled Booking',fontsize=14)
plt.legend(prop={'size':14})
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7f8f03c92b60>
```



Conclusion:

Maximum number of Booking done in August in City Hotel and Resort Hotel that are Not-Cancelled.

Questions are:-

6.How many days customers prefer to stay in week night?

7.How many days customers prefer to stay in weekend night?

8.what is the most preferred meal type by customers?

9.What is Percentage distribution of Deposite type ?

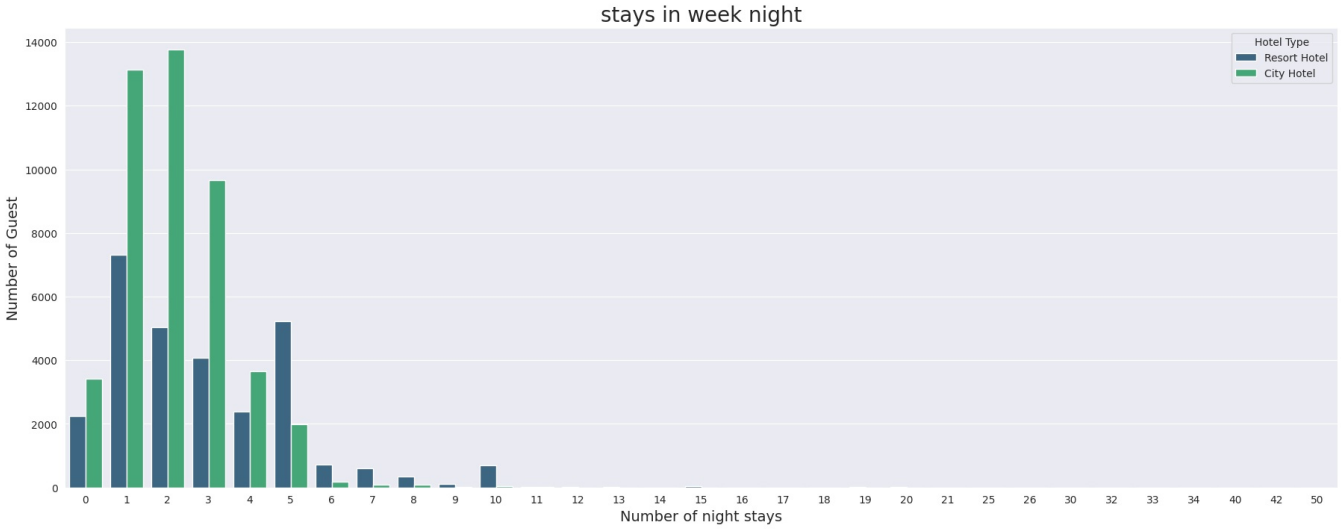
10.Which one is most preferred room type?

2. Customer type in each hotel

6. Customers stays in weekdays

In [ ]:
```
#Countplot of customers that stays in week night
sns.set_style('darkgrid')
plt.figure(figsize=(22,8))
sns.countplot(data = df2, x = 'stays_in_week_nights', hue='hotel', palette="viridis").set_title('stays in week
plt.xlabel("Number of night stays", fontsize=14)
plt.ylabel("Number of Guest", fontsize=14)
plt.legend(title = "Hotel Type",loc = 1)
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7f8f06527b20>
```



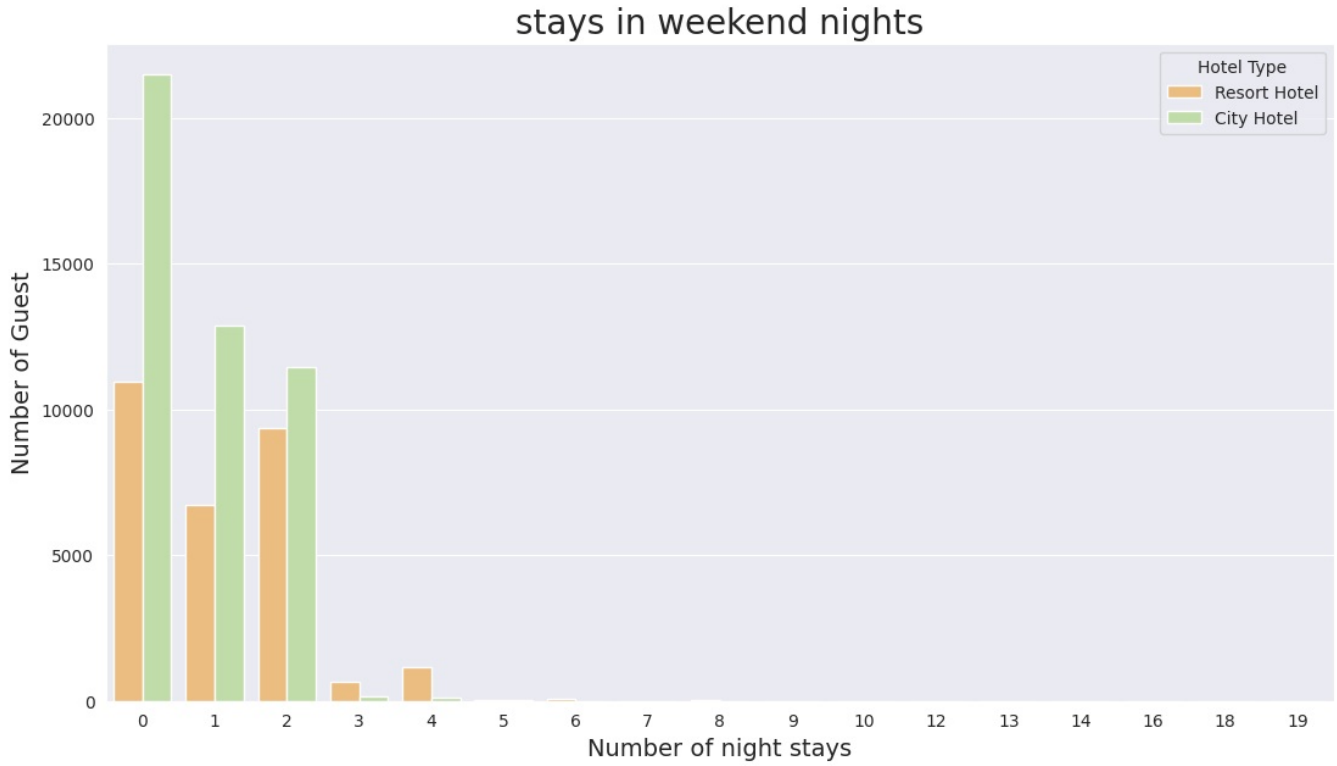
Conclusion :

Most of the customers preferred to stay for 2 days in City Hotel and 1 day in Resort Hotel.

7.Customers Stays in Weekend

In [ ]:
```
#Countplot for customer stays in weekend
sns.set_style('darkgrid')
plt.figure(figsize=(13,7))
sns.countplot(data = df2, x = 'stays_in_weekend_nights', hue='hotel', palette="Spectral").set_title('stays in w
plt.xlabel("Number of night stays", fontsize=14)
plt.ylabel("Number of Guest", fontsize=14)
plt.legend(title = "Hotel Type",loc = 1)
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7f8f059012a0>
```



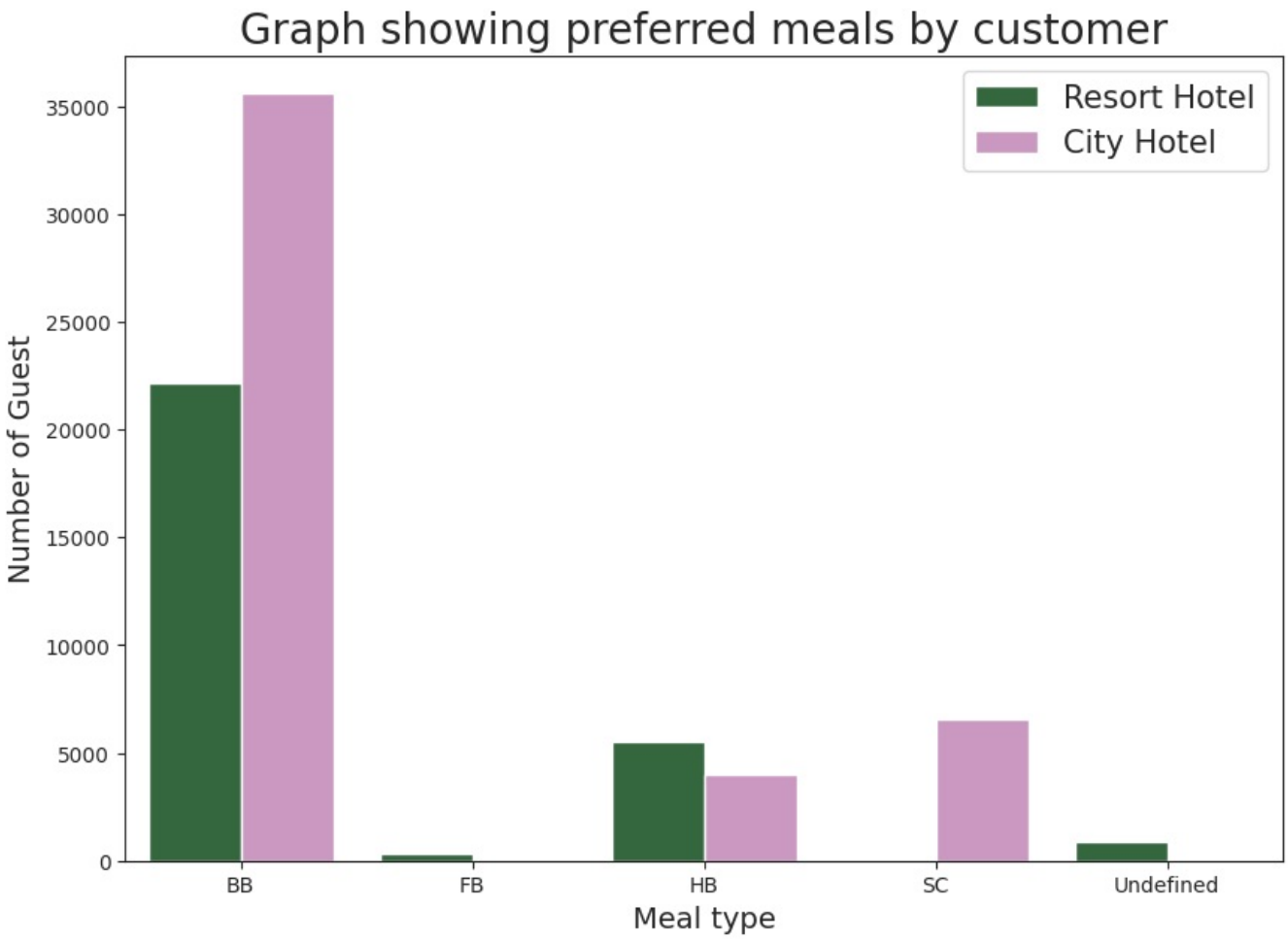
Conclusion:

Most customer stays for 0 nights in Resort and City hotels.

8. Preferred Meal type

In [ ]:
```
# for meal type.
sns.set_style('ticks')
plt.figure(figsize=(10,7))
sns.countplot(data = df2, x = 'meal', hue='hotel', palette="cubehelix").set_title('Graph showing preferred meal
plt.xlabel("Meal type", fontsize=14)
plt.ylabel("Number of Guest", fontsize=14)
plt.legend(loc=1, prop={'size':15})
```

Out[ ]:
```
<matplotlib.legend.Legend at 0x7f8f003781f0>
```



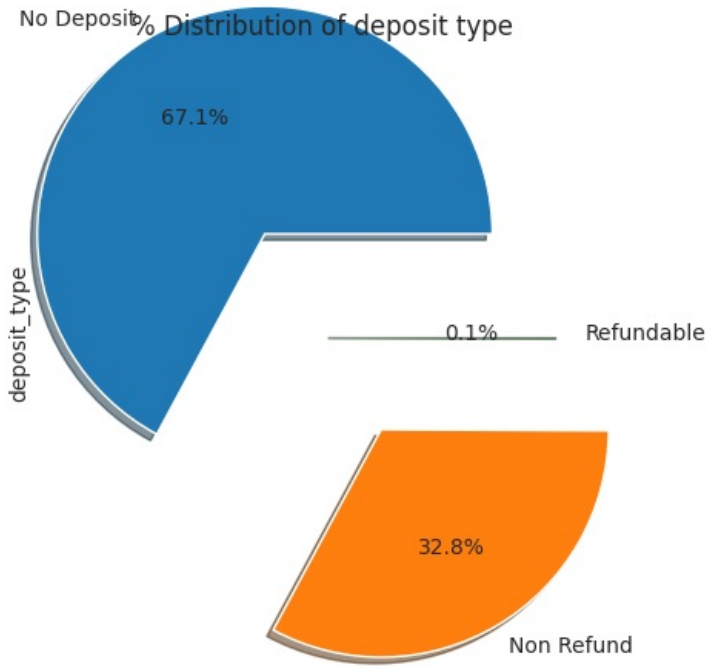
Conclusion:

Most preferred meal by customer is BB for Resort and City Hotel

9.What is Percentage distribution of Deposite type ?

In [ ]:
```
df1['deposit_type'].value_counts().plot(kind='pie',explode=(0.5,0.5,0.05),autopct='%1.1f%%',shadow=True)
plt.title("% Distribution of deposit type")
```

Out[ ]:
```
Text(0.5, 1.0, '% Distribution of deposit type')
```



Most of the guests are coming from portugal that is more 25000 guests are from portugal

abbreevations for countries-

PRT- Portugal

GBR- United Kingdom

FRA- France

ESP- Spain

DEU - Germany

ITA -Itlay

IRL - Ireland

BEL -Belgium

BRA -Brazil

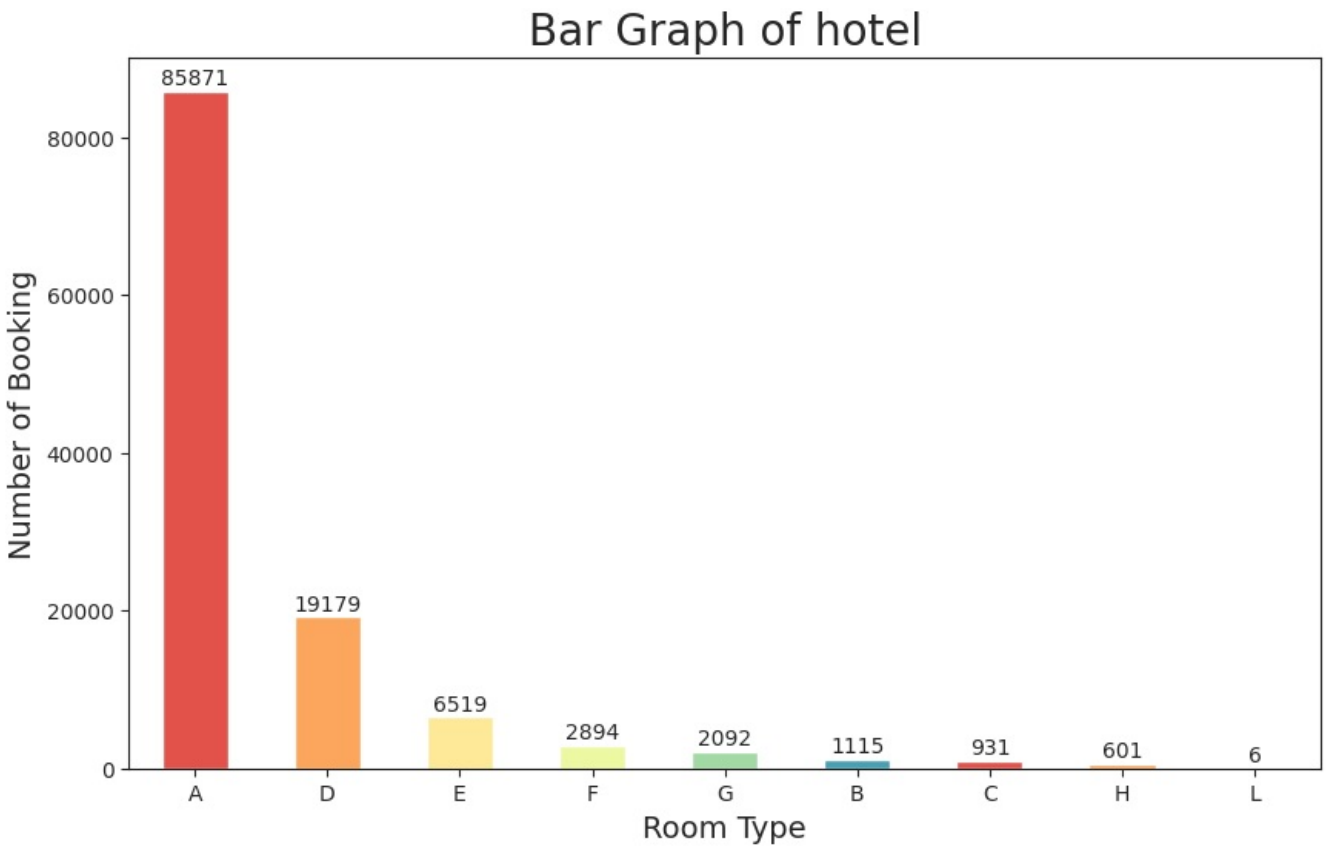
NLD-Netherlands

10. Most preferred Room-Type

In [ ]:
```
#Bar plot for number of customers and Room type
room_type = df['reserved_room_type'].value_counts()
plt.figure(figsize=(10, 6))
ax = room_type.plot(kind='bar', rot=0, color=sns.color_palette('Spectral'))
ax.set_title("Bar Graph of hotel", y = 1, fontsize=20)
plt.xlabel('Room Type', fontsize=14)
```

```
plt.ylabel('Number of Booking', fontsize=14)

for rect in ax.patches:
    y_value = rect.get_height()
    x_value = rect.get_x() + rect.get_width() / 2
    space = 1
    label = "{:.0f}".format(y_value)
    ax.annotate(label, (x_value, y_value), xytext=(0, space), textcoords="offset points", ha='center', va='bott
plt.show()
```



Conclusion:

Most Preferred room type by customer is Room A.

Questions are:-

11.What are the top 20 countries from where we are getting more customers?

12.What Deposit Type most customer choose?

13.From which market segment we are getting more number of Booking Cancellation?

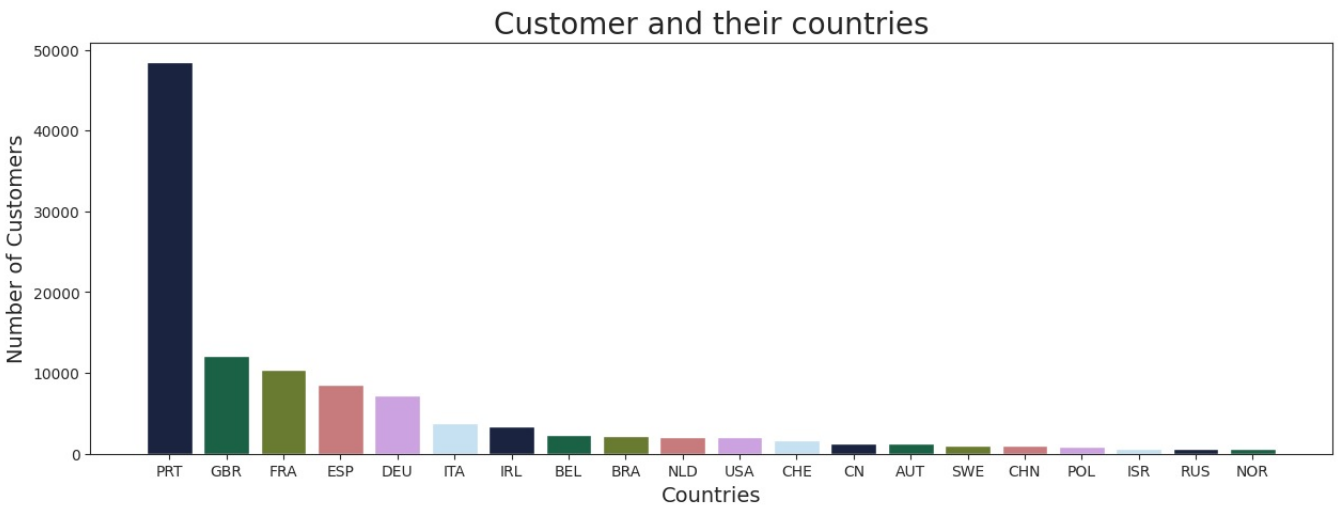
14.From which market segment we are getting more customers who are not cancelling their booking?

15.Which Agent(id) is booking the most number of hotels?

11.Top 20 Countries from where most number of customers are booking

In [ ]:
```
#barplot of customer and their origin countries
country = dict(df['country'].value_counts())
plt.figure(figsize=(15,5))
plt.bar((list(country.keys()))[:20],(list(country.values()))[:20] ,width=0.8, color=sns.color_palette('cubeheli
plt.title('Customer and their countries',fontsize = 20)
plt.xlabel('Countries' ,fontsize = 14)
plt.ylabel('Number of Customers', fontsize = 14)
```

Out[ ]:
```
Text(0, 0.5, 'Number of Customers')
```



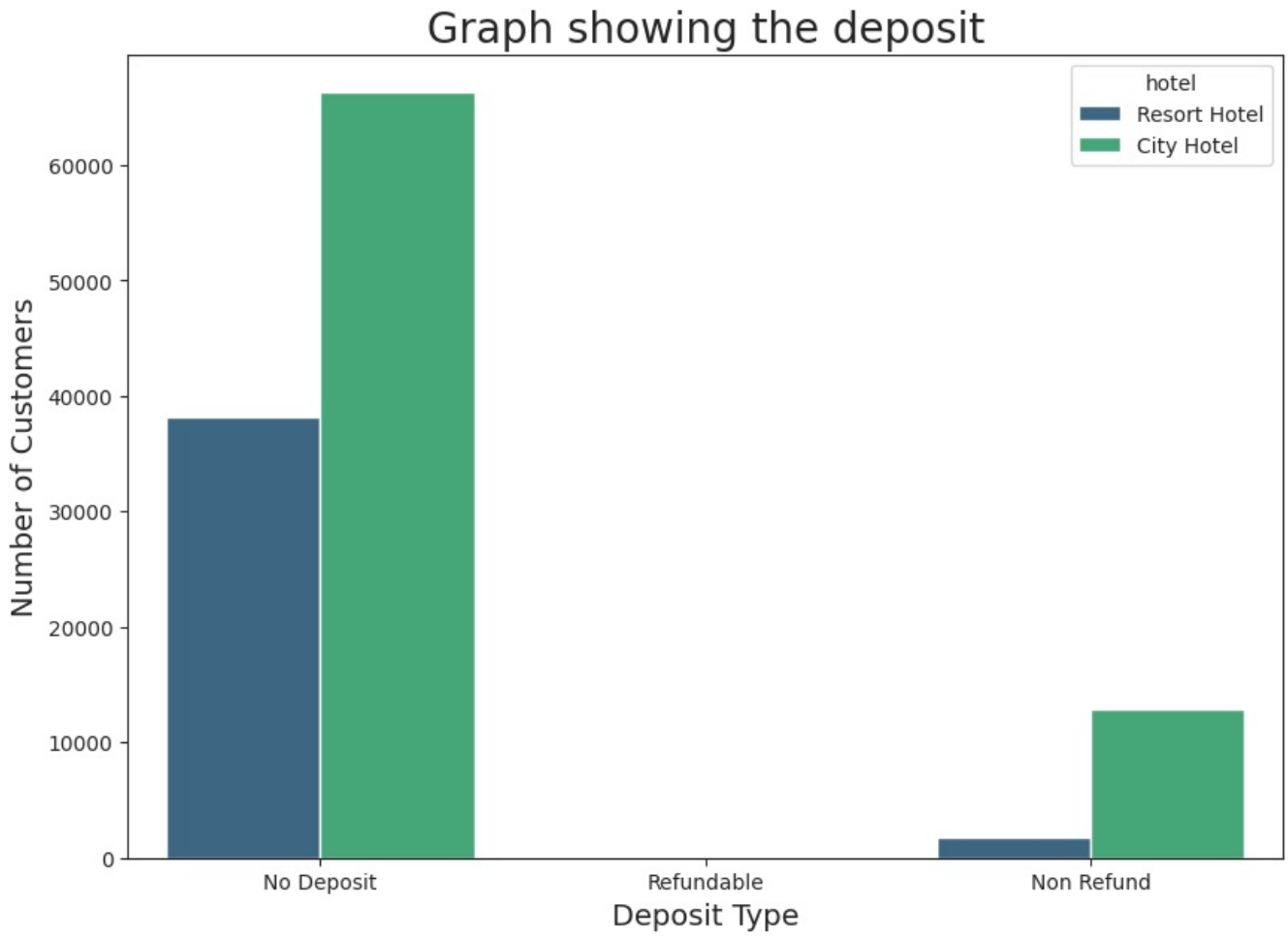
Conclusion:

Most of the customers do not make any special request.

12. Deposit Type

In [ ]:
```
#Countplot for deposit
sns.set_style('ticks')
plt.figure(figsize=(10,7))
ax=sns.countplot(data = df, x = 'deposit_type',hue='hotel', palette="viridis").set_title('Graph showing the dep
plt.xlabel("Deposit Type", fontsize=14)
plt.ylabel("Number of Customers", fontsize=14)
```

Out[ ]:
```
Text(0, 0.5, 'Number of Customers')
```

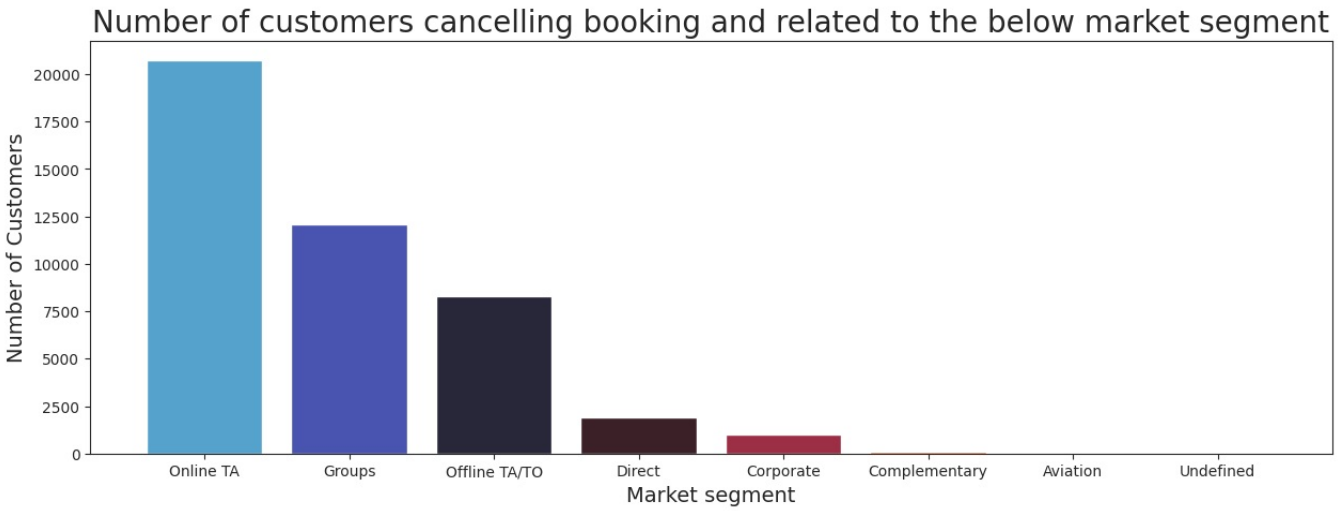


Conclusion:

Most of the customers prefer No Deposit for booking.

13. Plot of cancelling booking from given market segment.

In [ ]:
```
#barplot for cancelled booking and the market segment
counttt = dict(df1['market_segment'].value_counts())
aa=list(counttt.keys())
bb=list(counttt.values())
plt.figure(figsize = (15, 5))
plt.bar(aa,bb ,width=0.8, color=sns.color_palette('icefire'))
plt.title('Number of customers cancelling booking and related to the below market segment',fontsize = 20)
plt.xlabel('Market segment' ,fontsize = 14)
plt.ylabel('Number of Customers', fontsize = 14)
plt.show()
```



Conclusion:

Most of the customers from Online TA cancel their booking.

14. Plot of not cancelling booking from given market segment.

In [ ]:
```
#barplot for booking not cancelled and the market segment
counttt = dict(df2['market_segment'].value_counts())
aa=list(counttt.keys())
bb=list(counttt.values())
plt.figure(figsize = (15, 5))
plt.bar(aa,bb ,width=0.8, color=sns.color_palette('icefire'))
plt.title('Number of customers not cancelled booking from below market segments',fontsize = 20)
plt.xlabel('Market segment' ,fontsize = 14)
plt.ylabel('Number of Customers', fontsize = 14)
plt.show()
```

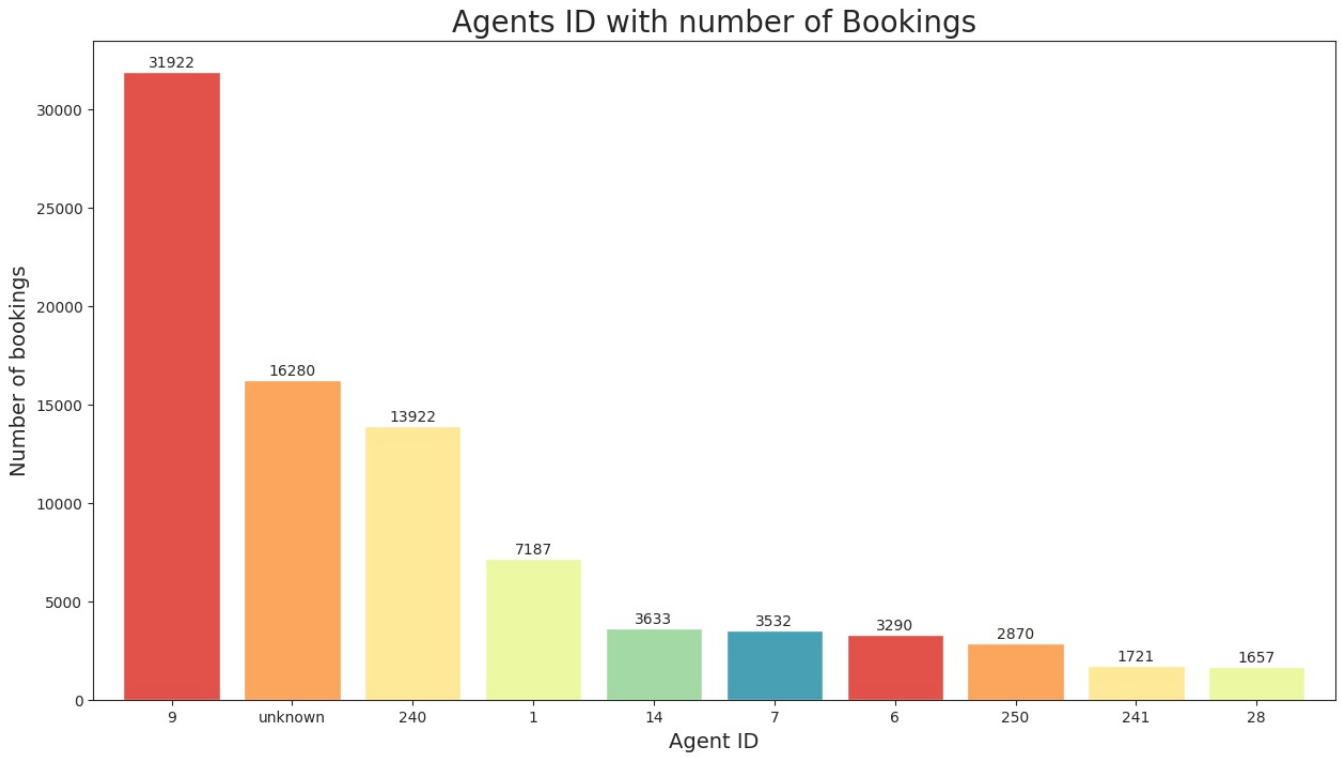
Conclusion:

Most of the customer from Online TA are not Cancelling their bookings.

15. Which Agent(id) is booking the most number of hotels

In [ ]:
```
#barplot for agents and their bookings
agent = df['agent'].value_counts().head(10)
plt.figure(figsize=(15,8))
ax = agent.plot(kind='bar', rot=0, color=sns.color_palette('Spectral'), width=0.8)
ax.set_title("Agents ID with number of Bookings", y = 1, fontsize = 20)
ax.set_xlabel('Agent ID', fontsize = 14)
ax.set_ylabel('Number of bookings', fontsize = 14)

for rect in ax.patches:
    y_value = rect.get_height()
    x_value = rect.get_x() + rect.get_width() / 2
    space = 1
    label = "{:.0f}".format(y_value)
    ax.annotate(label, (x_value, y_value), xytext=(0, space), textcoords="offset points", ha='center', va='bott
plt.show()
```



Conclusion:

Agent with ID 9, books most number of hotels.

Questions are:-

16.Which Room type has high average price?

17.In Which month most revenue are generated?

18.What is the optimal length to stay?

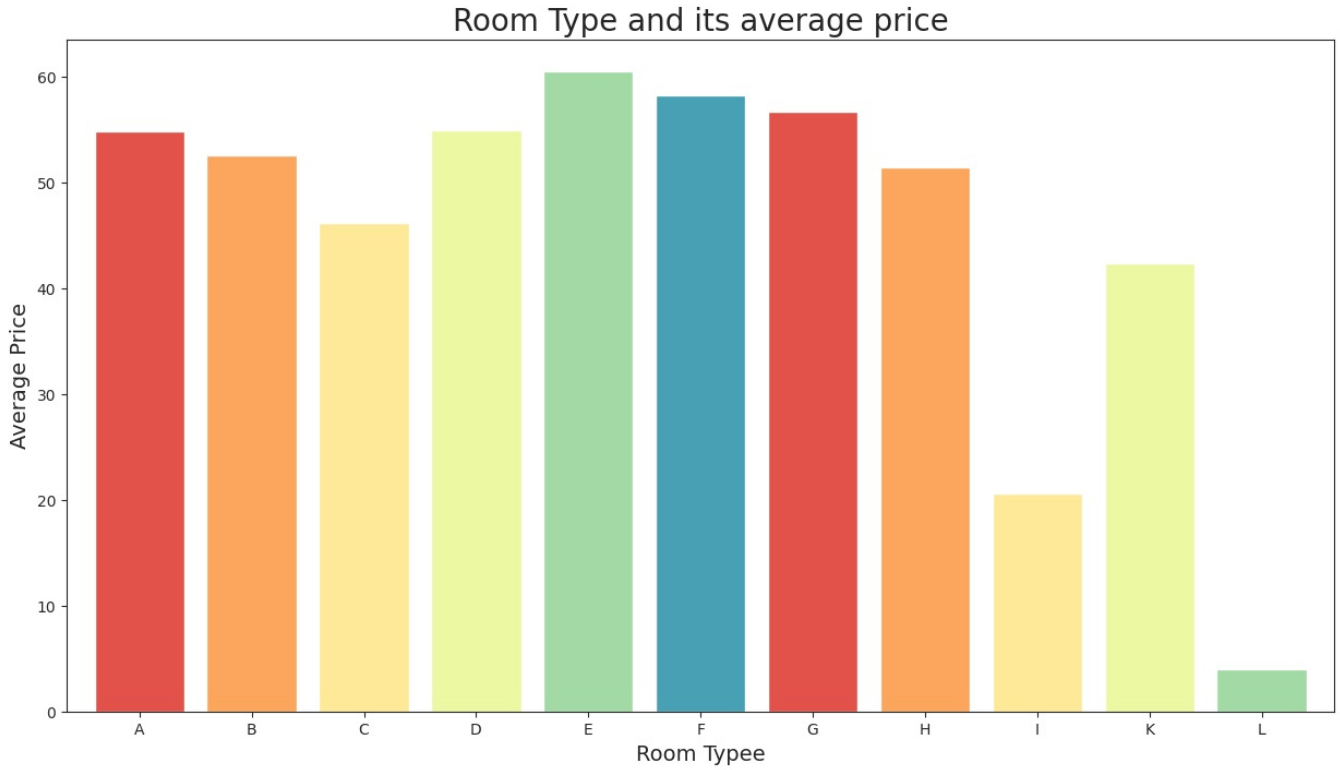
19.How many repeated guests we have?

20.Correlation between features.

16. Room type and average price

In [ ]:
```
#barplot for Room type and their average price
df['adr_pp'] = df['adr'] / (df['adults'] + df['children']+ df['babies'])
a=df.groupby('assigned_room_type')['adr_pp'].mean()
plt.figure(figsize=(15,8))
ax = a.plot(kind='bar', rot=0, color=sns.color_palette('Spectral'), width=0.8)
ax.set_title("Room Type and its average price", y = 1, fontsize = 20)
ax.set_xlabel('Room Typee', fontsize = 14)
ax.set_ylabel('Average Price', fontsize = 14)
```

Out[ ]:
```
Text(0, 0.5, 'Average Price')
```



Conclusion:

The average price of Room E is the maximum and the average price of Room L is minimum.

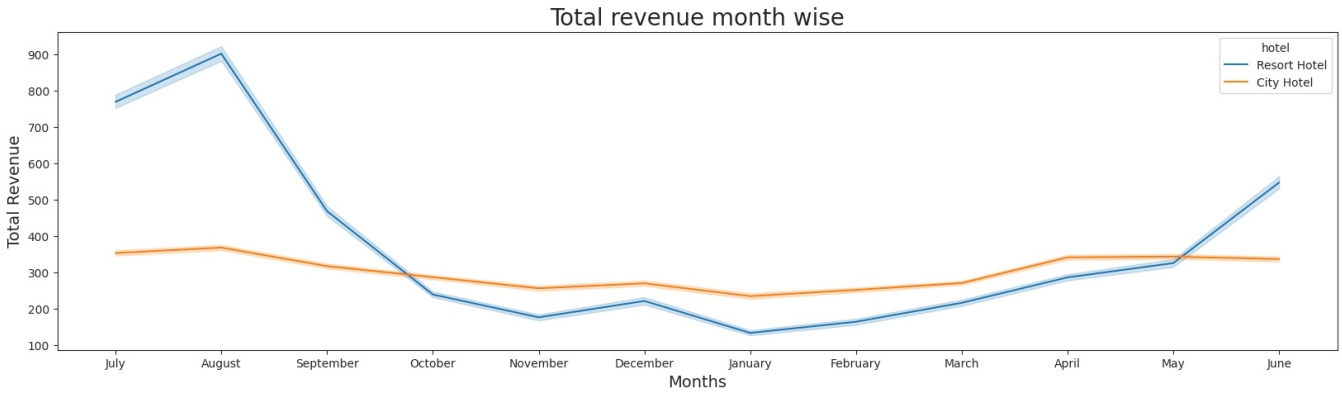
17. Revenue per month per hotel

In [ ]:
```
#lineplot for Revenue per hotel month-wise
plt.figure(figsize=(20,5))
df2['price'] = df2['adr'] * (df2['stays_in_weekend_nights'] + df2['stays_in_week_nights'])
sns.lineplot(data = df2, x = 'arrival_date_month', y = 'price', hue = 'hotel')
plt.title('Total revenue month wise', fontsize=20)
plt.xlabel('Months',fontsize=14)
plt.ylabel('Total Revenue',fontsize=14)
```

```
<ipython-input-76-b1f39e834c94>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df2['price'] = df2['adr'] * (df2['stays_in_weekend_nights'] + df2['stays_in_week_nights'])
```

Out[ ]:
```
Text(0, 0.5, 'Total Revenue')
```

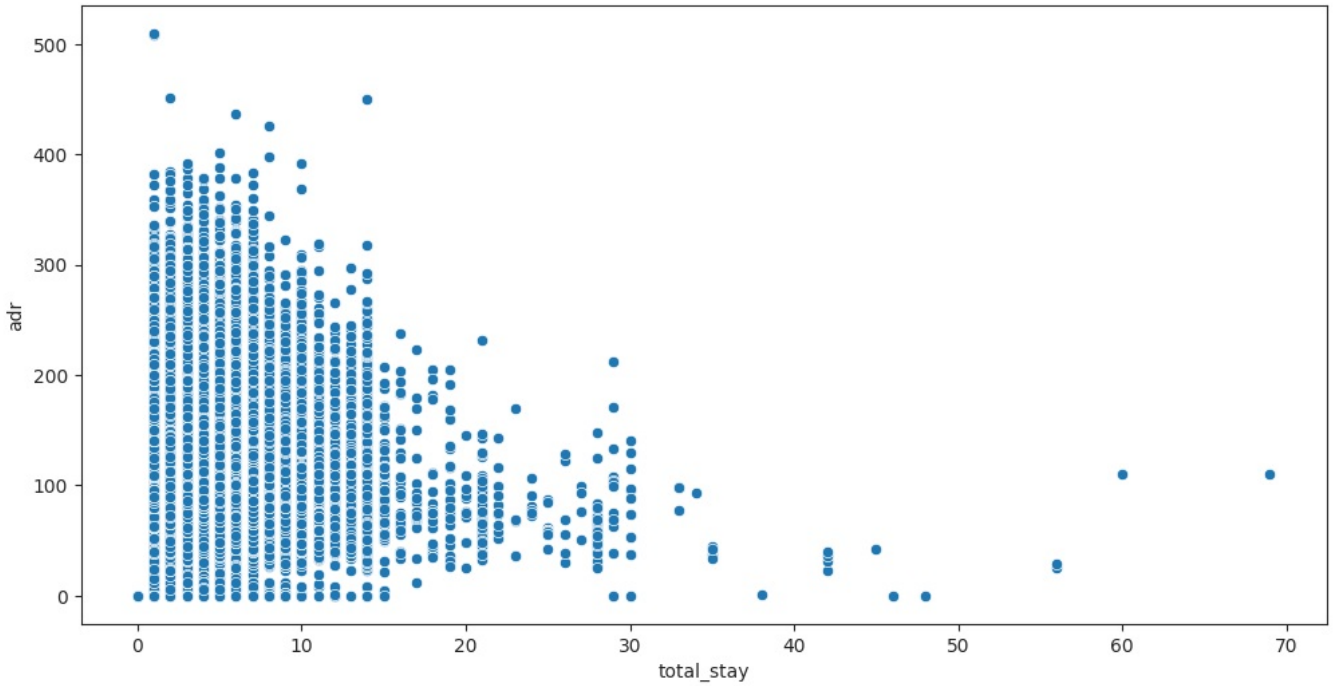


Conclusion:

Revenue of Resort Hotel are maximum in the month of August and least in the month of January, and there is no high fluctuation in the price of city Hotels throughout the Year.

18. Plotting Scatter plot to find the optimal length of stay

In [ ]:
```
#scatterplot for optimal stay length
df['total_stay']=df['stays_in_week_nights']+df['stays_in_weekend_nights']
plt.figure(figsize = (12,6))
sns.scatterplot(y = 'adr', x = 'total_stay', data = df)
plt.show()
```



Conclusion:

The longer the stay length, the best price customer will get.

19.Number of repeated guests

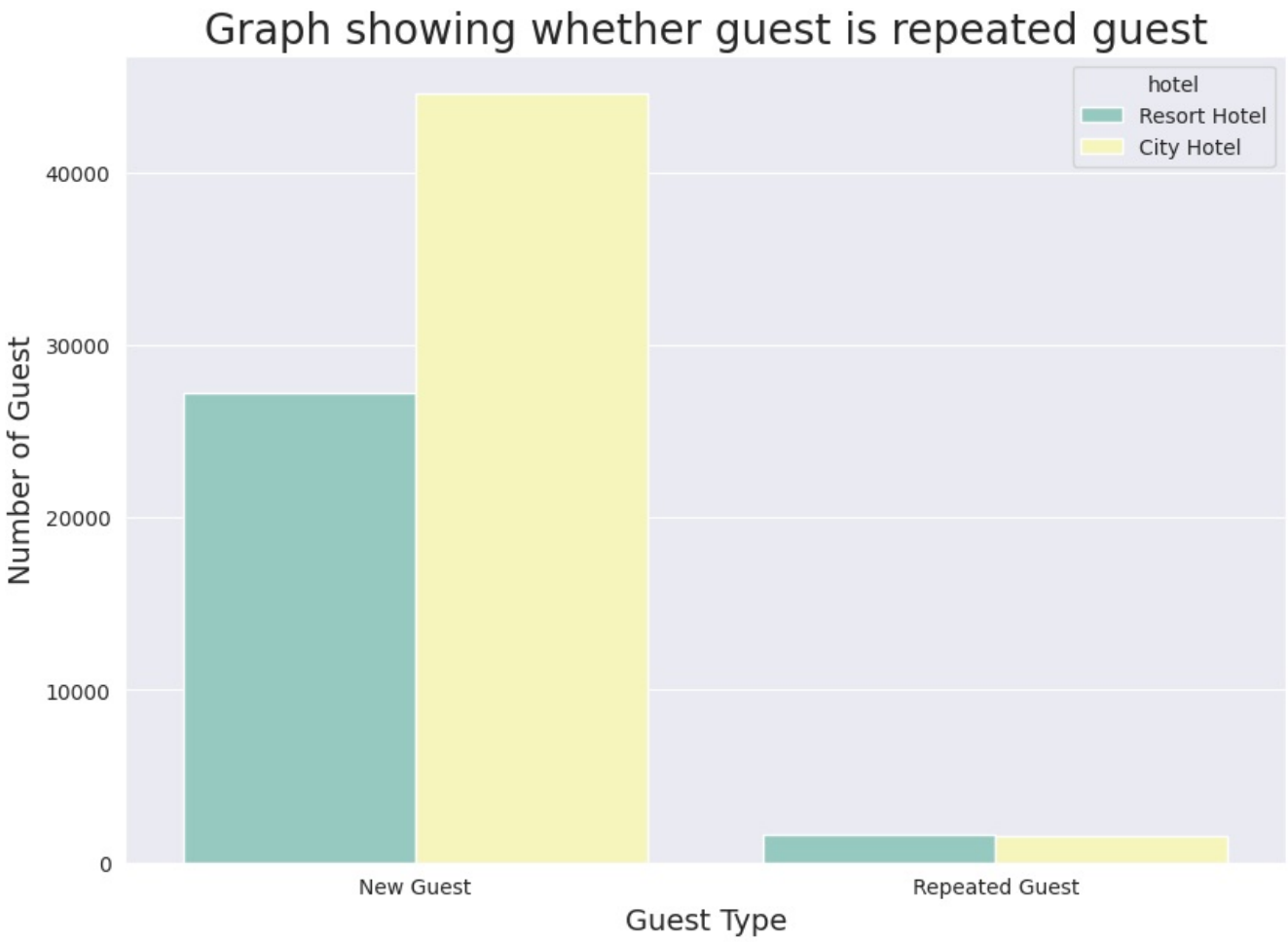
In [ ]:
```
df2['is_repeated_guest'] = df2['is_repeated_guest'].apply(lambda x : 'New Guest' if x==0 else 'Repeated Guest')
```

```
<ipython-input-78-49c0b1e6c5e3>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df2['is_repeated_guest'] = df2['is_repeated_guest'].apply(lambda x : 'New Guest' if x==0 else 'Repeated Guest
')
```

In [ ]:
```
#countplot for guests [Repeated or New]
sns.set_style('darkgrid')
plt.figure(figsize=(10,7))
sns.countplot(data = df2, x = 'is_repeated_guest', hue='hotel', palette="Set3").set_title('Graph showing whethe
plt.xlabel("Guest Type", fontsize=14)
plt.ylabel("Number of Guest", fontsize=14)
```

Out[ ]:
```
Text(0, 0.5, 'Number of Guest')
```



Conclusion:

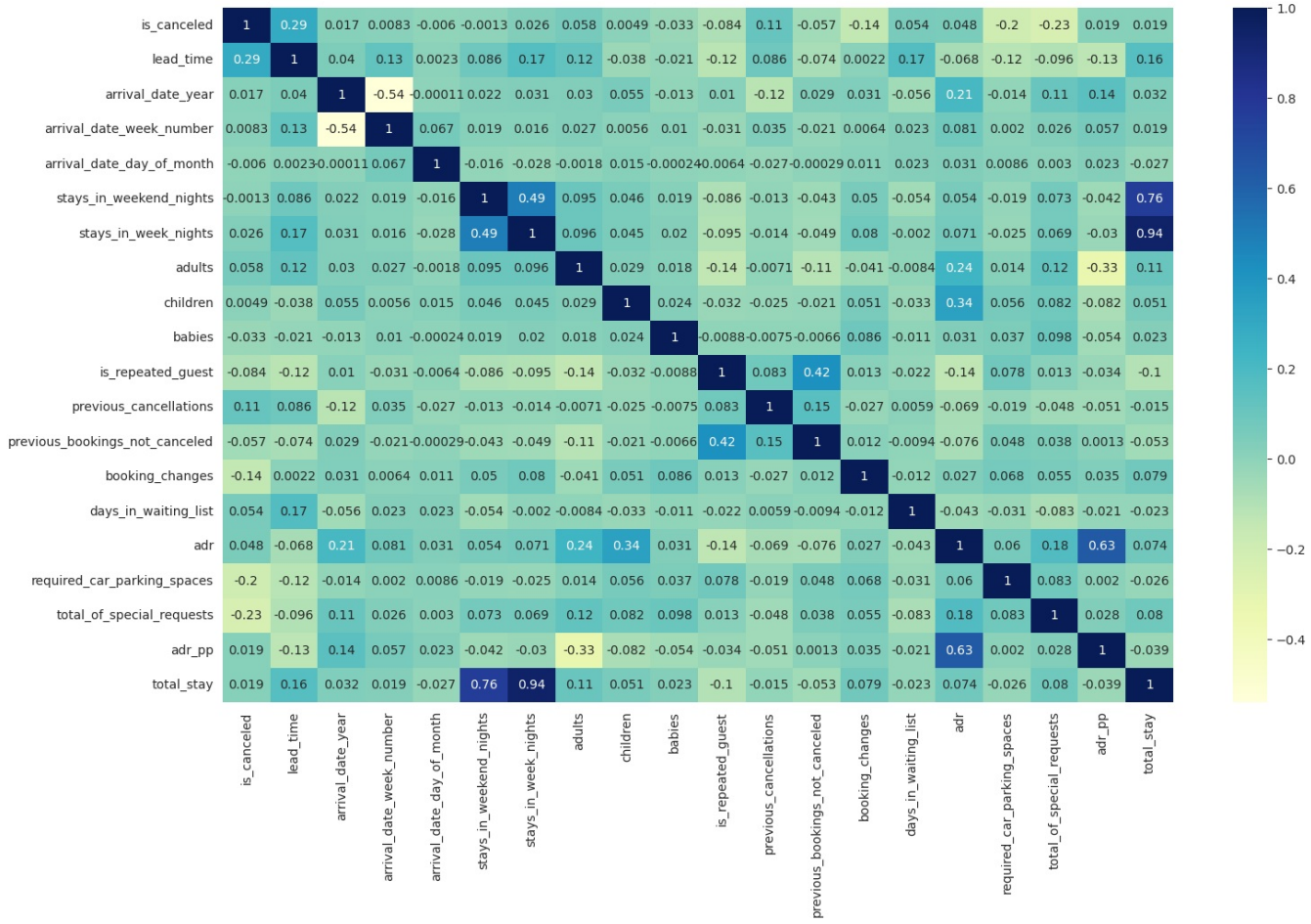
We have more number of New Guest in both type of hotels.

20. Correlation between features

In [ ]:
```
#Correlation
correlation = df.corr()
plt.figure(figsize=(17,10))
sns.heatmap(correlation, annot=True ,cmap="YlGnBu")
```

```
<ipython-input-80-0013ff83ce84>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprec
ated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_
only to silence this warning.
  correlation = df.corr()
```

Out[ ]:
```
<Axes: >
```



Conclusion:

Total stay is highly correlated with stays_in_weekend_nighrs and stays_in_week_nights.

In [ ]: