DATA VISUALIZATION PROJECT

DATASET - HOTEL BOOKINGS

GROUP MEMBERS

1. SHALINI PRIYA.B
2. SHOIEB ALI KHAN
3. TANNU GIRI
4. OWK MADHU

1. B.Hemanth Kumar

IMPORTING THE LIBRARIES

NUMPY - It is used for working with arrays . It stands for Numerical Array

PANDAS - This library is for Data Analysis.

SEABORN - It provides a high - level interface for drawing attractive informative statistical graphics.

MATPLOTLIB - This library is built on the top of NumPy arrays and consists of several plots like bar plot , pie plot etc..

PLOTLY - This library makes iteractive ,publication quality graphs .

plotly.graph_objects- contains objects that are responsible for creating plots.

In [ ]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
from wordcloud import WordCloud, STOPWORDS
import warnings
warnings.filterwarnings("ignore")
```

IMPORTING FILES FROM GOOGLE DRIVE

In [ ]:

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

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

ABSTRACT OF THE DATASET

This data article describes two datasets with hotel demand data. One of the hotels (H1) is a resort hotel and the other is a city hotel (H2). Both datasets share the same structure, with 31 variables describing the 40,060 observations of H1 and 79,330 observations of H2. Each observation represents a hotel booking. Both datasets comprehend bookings due to arrive between the 1st of July of 2015 and the 31st of August 2017, including bookings that effectively arrived and bookings that were canceled. Since this is hotel real data, all data elements pertaining hotel or costumer identification were deleted. Due to the scarcity of real business data for scientific and educational purposes, these datasets can have an important role for research and education in revenue management, machine learning, or data mining, as well as in other fields.

DESCRIPTION OF COLUMNS

0 hotel (H1 = Resort Hotel or H2 = City Hotel)

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

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

3 arrival_date_year Year of arrival date

4 arrival_date_month Month of arrival date

5 arrival_date_week_number Week number of year for arrival date

6 arrival_date_day_of_month Day of arrival date

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

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

9 adults Number of adults

10 children Number of children

11 babies Number of babies

12 meal Type of meal booked. Categories are presented in standard hospitality meal packages: Undefined/SC – no meal

13 country Country of origin. Categories are represented in the ISO 3155–3:2013 format

14 market_segment Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”

15 distribution_channel Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”

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

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

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

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

20 assigned_room_typeCode for the type of room assigned to the booking.Code is presented instead of designation for anonymity reasons.

21 booking_changes Number of changes made to the booking from the moment the booking was entered on the PMS until the moment of check-in or out

22 deposit_type Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories: No

23 agent ID of the travel agency that made the booking

24 company ID of the company that made the booking or responsible for paying the booking.

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

26 customer_type Type of booking, assuming one of four categories:Transient - Transient-Party - Contract - Group

Group

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

28 required_car_parking_spaces Number of car parking spaces required by the customer

29 total_of_special_requestsNumber of special requests made by the customer (e.g. twin bed or high floor)

30 reservation_status Reservation last status, assuming one of three categories: Canceled – booking was canceled by the customer; Check-Out

31 reservation_status_date Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to

Extracting data from the csv and storing in data (dataframe)

In [ ]:

```
df=pd.read_csv("/content/drive/MyDrive/hotel_bookings.csv",encoding='latin-1')
df30=pd.read_csv("/content/drive/MyDrive/hotel_bookings.csv",encoding='latin-1',nrows=100
)
df100=pd.read_csv("/content/drive/MyDrive/hotel_bookings.csv",encoding='latin-1',nrows=10
00)
df500=pd.read_csv("/content/drive/MyDrive/hotel_bookings.csv",encoding='latin-1',nrows=60
00)
df.head()
```

Out[ ]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month
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

The info() function is used to print a concise summary of a DataFrame.

In [ ]:

```
print("The information of the hotel dataframe is")
print(df.info())
```

```
The information of the hotel dataframe is
<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
None
```

The head() returns the first n rows for the object based on position.

In [ ]:

```
df.head()
```

Out[ ]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month
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

The tail() function is used to get the last n rows. This function returns last n rows from the object based on position.

In [ ]:

```
df.tail()
```

Out[ ]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_mc
119385	City Hotel	0	23	2017	August	35	
119386	City Hotel	0	102	2017	August	35	
119387	City Hotel	0	34	2017	August	35	

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_mo
119388	City Hotel	0	109	2017	August	35	
119389	City Hotel	0	205	2017	August	35	

In [ ]:

```
print("The number of rows and columns are")
print(df.shape)
```

```
The number of rows and columns are
(119390, 32)
```

In [ ]:

```
print ( "Number of rows are", df.shape[0])
print ( "Number of columns are", df.shape[1])
```

```
Number of rows are 119390
Number of columns are 32
```

In [ ]:

```
print("The columns present in dataframe is")
print(df.columns)
```

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

In [ ]:

```
df.isnull()
```

Out[ ]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_mo
0	False	False	False	False	False	False	Fa
1	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	Fa
3	False	False	False	False	False	False	Fa
4	False	False	False	False	False	False	Fa
...	...	...	...	...	...	...	
119385	False	False	False	False	False	False	Fa
119386	False	False	False	False	False	False	Fa
119387	False	False	False	False	False	False	Fa
119388	False	False	False	False	False	False	Fa
119389	False	False	False	False	False	False	Fa

119390 rows × 32 columns

BAR PLOT

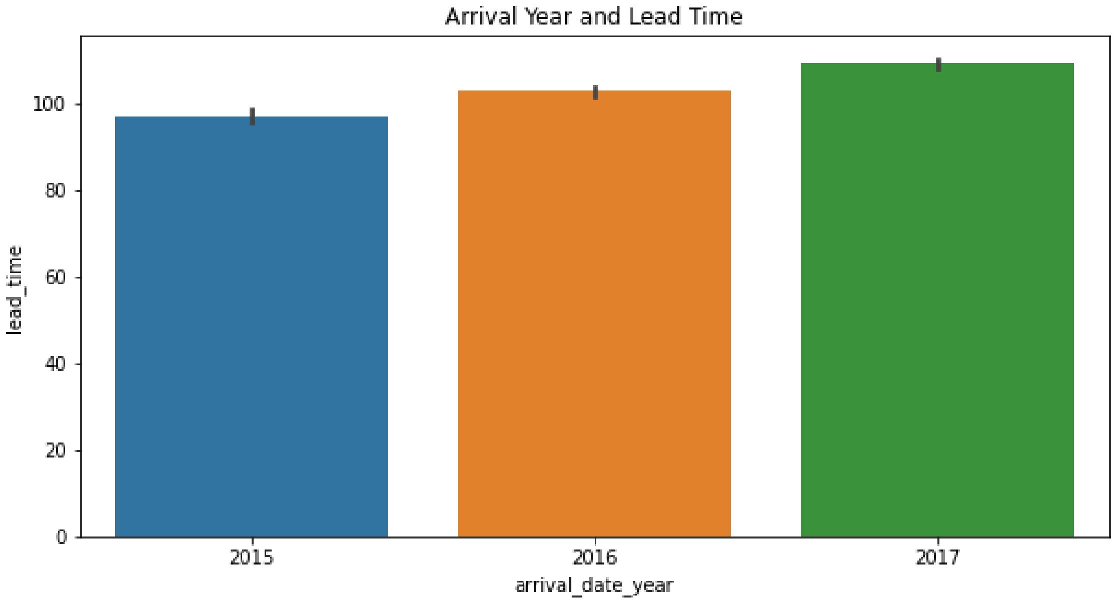
Bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically.

In [ ]:

```
plt.figure(figsize = (10, 5))
sns.barplot(x = 'arrival_date_year',y = 'lead_time', data = df).set_title('Arrival Year
and Lead Time')
```

Out[ ]:

```
Text(0.5, 1.0, 'Arrival Year and Lead Time')
```

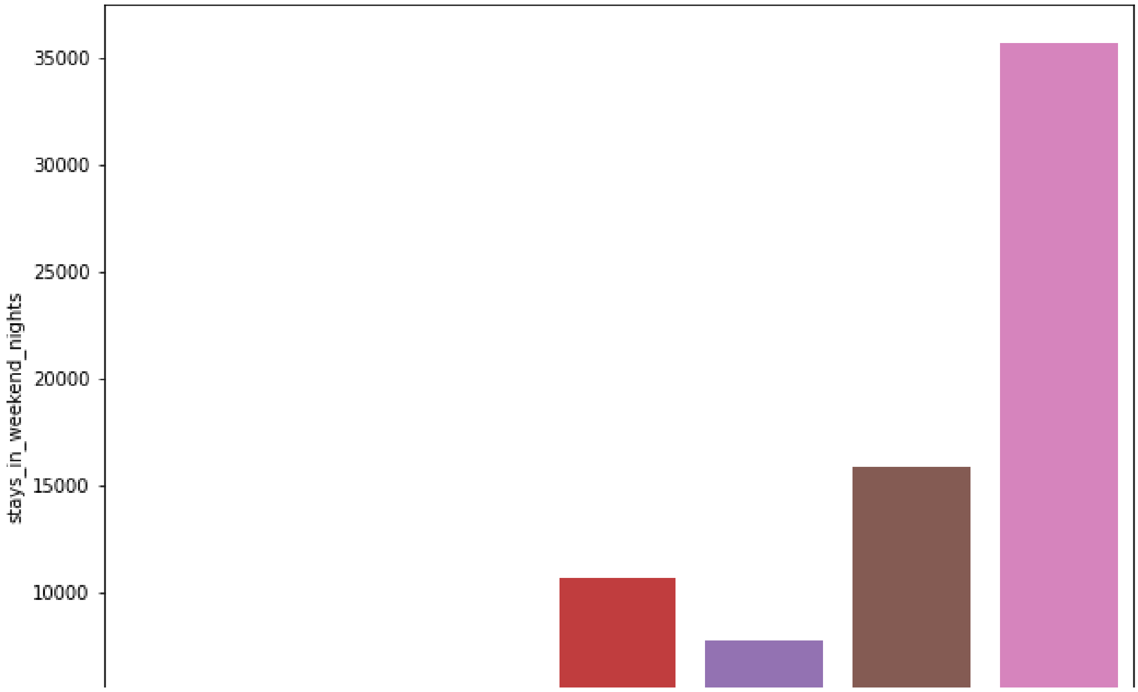


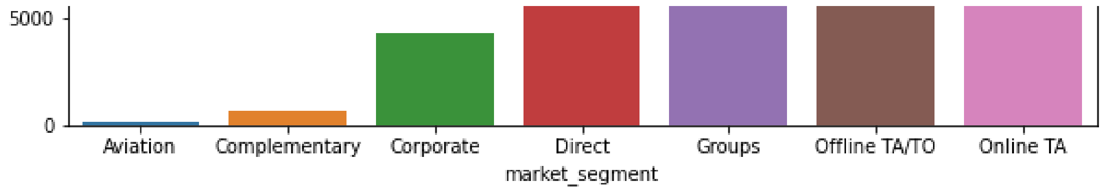
In [ ]:

```
plt.figure(figsize=(10,8))
sns.barplot(x=df[df['is_canceled']==0].groupby('market_segment')['stays_in_weekend_night
s'].count().index,
            y=df[df['is_canceled']==0].groupby('market_segment')['stays_in_weekend_night
s'].count())
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1378893f90>
```



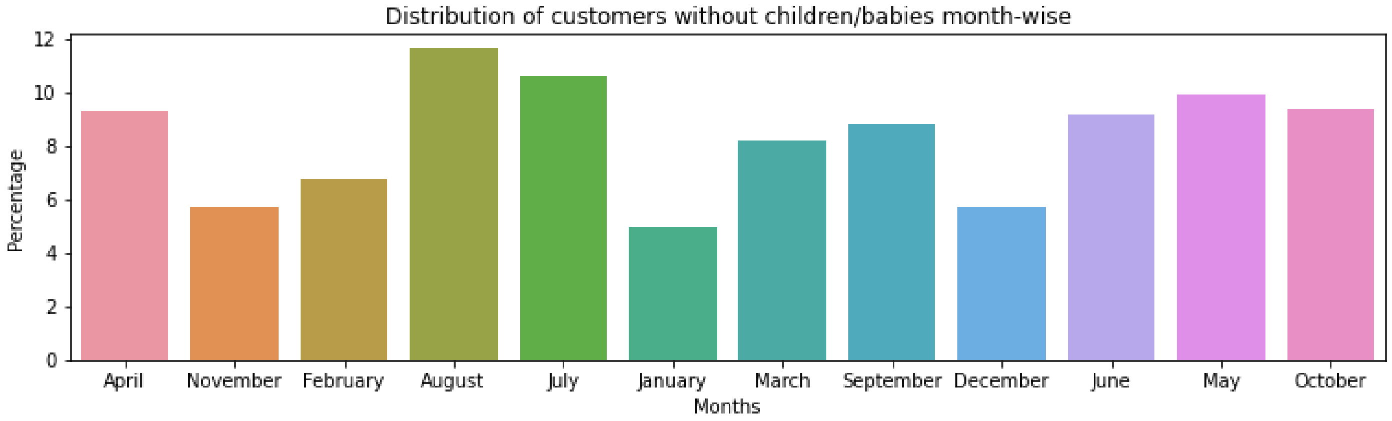


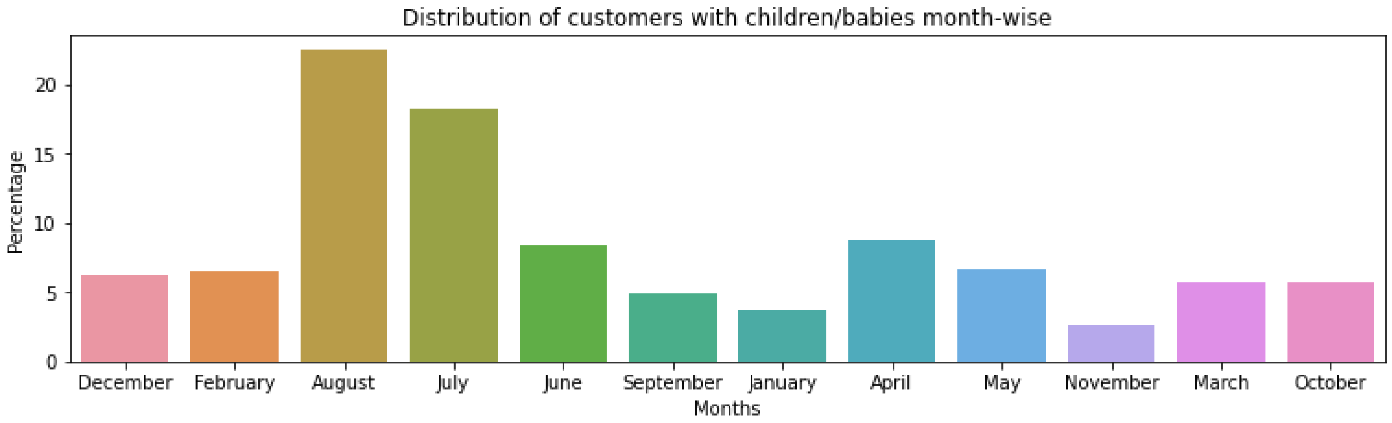
In [ ]:

```
without_children = df[(df['children'] == 0) | (df['babies'] == 0)]
with_children = df[(df['children'] != 0) | (df['babies'] != 0)]
without_children_perc = without_children['arrival_date_month'].value_counts(sort = False
)/sum(without_children['arrival_date_month'].value_counts(sort = False)) * 100
with_children_perc = with_children['arrival_date_month'].value_counts(sort = False)/sum(w
ith_children['arrival_date_month'].value_counts(sort = False)) * 100
fig, ax = plt.subplots(2, 1, figsize = (12, 8))

plt.rcParams.update({'text.color': "black",
                     'axes.labelcolor': "black"})
fig.tight_layout(pad = 6.0)

sns.barplot(x = without_children['arrival_date_month'].value_counts(sort = False).index,
y = without_children_perc, ax = ax[0]).set(title= "Distribution of customers without chi
ldren/babies month-wise", xlabel = "Months", ylabel = "Percentage")
sns.barplot(x = with_children['arrival_date_month'].value_counts(sort = False).index, y
= with_children_perc, ax = ax[1]).set(title= "Distribution of customers with children/bab
ies month-wise", xlabel = "Months", ylabel = "Percentage")
```

Out[ ]:

```
[Text(81.125, 0.5, 'Percentage'),
 Text(0.5, 51.00000000000006, 'Months'),
 Text(0.5, 1.0, 'Distribution of customers with children/babies month-wise')]
```



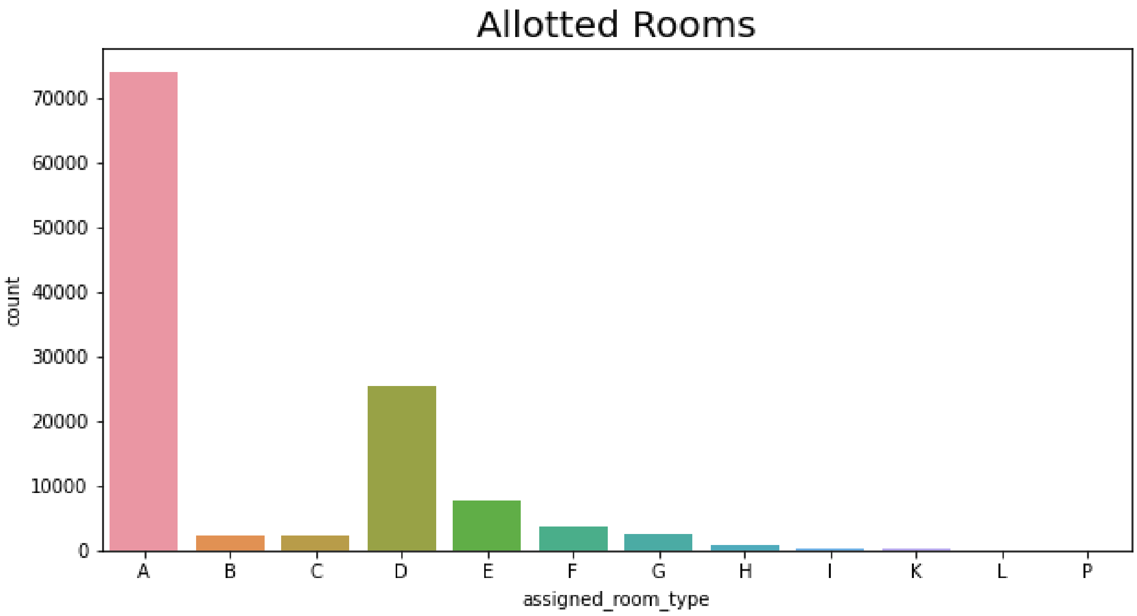


COUNT PLOT

A count plot can be thought of as a histogram across a categorical, instead of quantitative, variable.

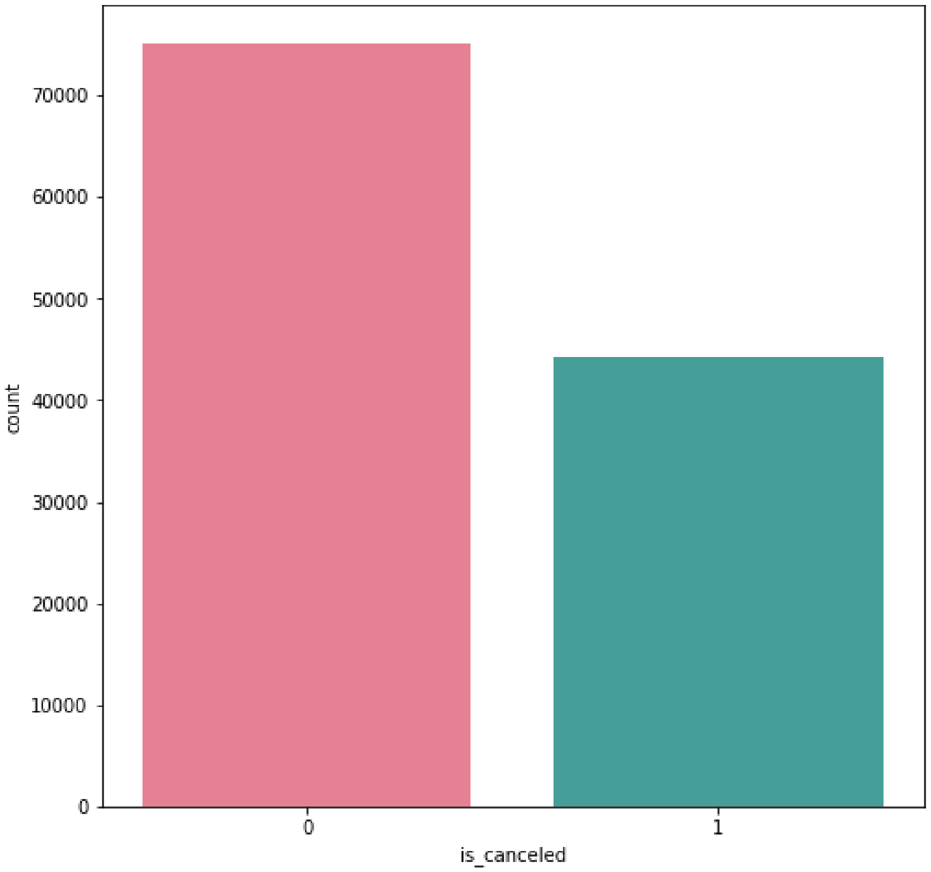
In [ ]:

```
plt.rcParams['figure.figsize'] = (10, 5)
sns.countplot(df['assigned_room_type'].sort_values(), )
plt.title('Allotted Rooms', fontsize = 20)
plt.show()
```



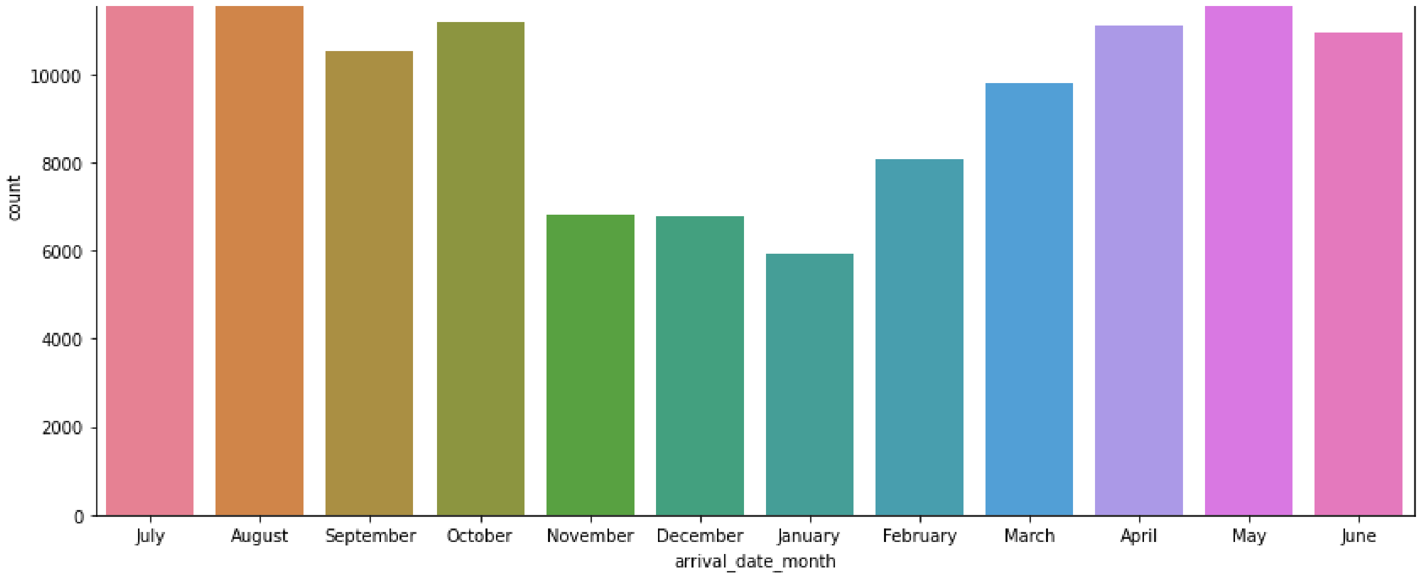
In [ ]:

```
plt.figure(figsize=(8,8))
sns.countplot(df['is_canceled'], palette='husl')
plt.show()
```



In [ ]:

```
plt.figure(figsize=(14,7))
sns.countplot(df['arrival_date_month'], palette='husl')
plt.show()
```

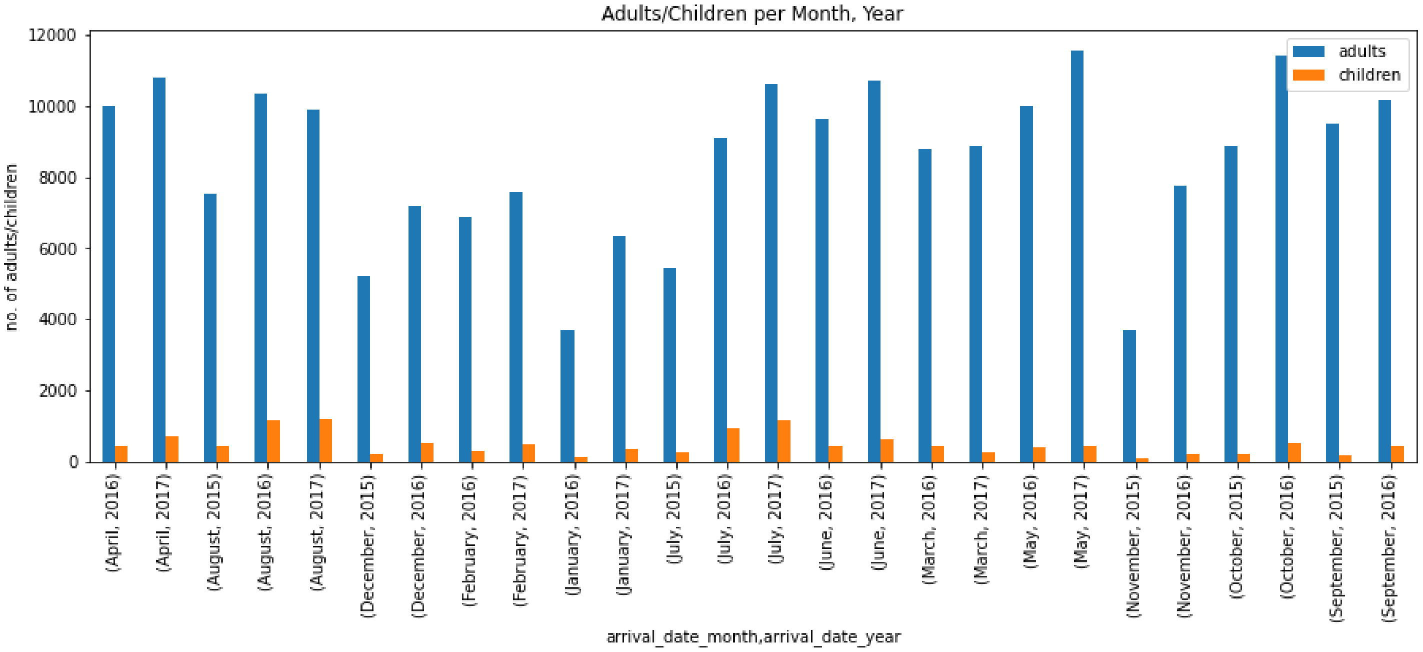


In [ ]:

```
df.groupby(['arrival_date_month', 'arrival_date_year'])[['adults', 'children']].sum().pl
ot.bar(figsize = (15, 5))

plt.ylabel('no. of adults/children')
plt.title('Adults/Children per Month, Year')
```

Out[ ]:

```
Text(0.5, 1.0, 'Adults/Children per Month, Year')
```

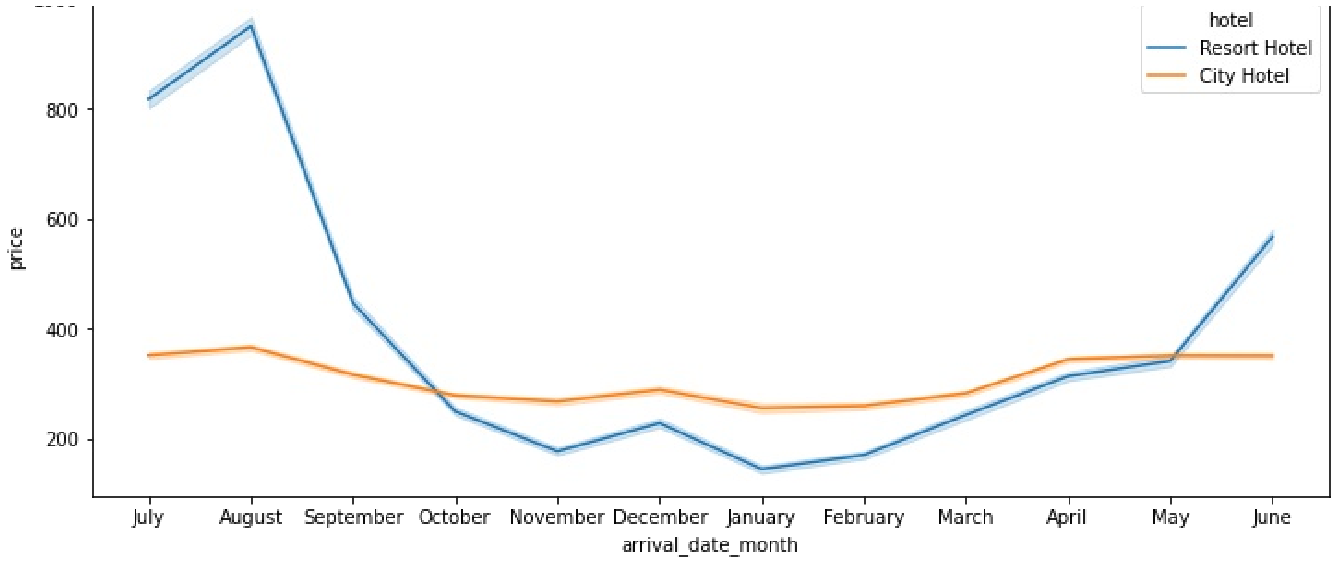


LINE GRAPH

A line graph is a graphical display of information that changes continuously over time. Within a line graph, there are various data points connected together by a straight line that reveals a continuous change in the values represented by the data points.

In [ ]:

```
plt.figure(figsize=(12,5))
df['adr_pp'] = df['adr'] / (df['adults'] + df['children'])
actual_guests = df.loc[df["is_canceled"] != '0']
actual_guests['price'] = actual_guests['adr'] * (actual_guests['stays_in_weekend_nights']
+ actual_guests['stays_in_week_nights'])
sns.lineplot(data = actual_guests, x = 'arrival_date_month', y = 'price', hue = 'hotel')
plt.show()
```

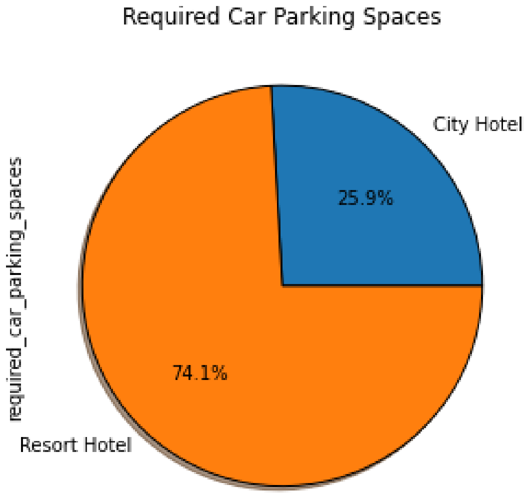


PIE GRAPH

A Pie Chart is a circular statistical plot that can display only one series of data. The area of the chart is the total percentage of the given data. The area of slices of the pie represents the percentage of the parts of the data. The slices of pie are called wedges.

In [ ]:

```
df.groupby(['hotel'])['required_car_parking_spaces'].sum().plot.pie(shadow=True,autopct=
"%0.1f%%",wedgeprops={'edgecolor':'black'},radius = 1)
plt.title('Required Car Parking Spaces')
explodes=[0.1,0]
```



In [ ]:

```
import plotly.graph_objects as go
```

In [ ]:

```
labels = df.groupby(['country']).size().sort_values(ascending = False).index
values = df.groupby(['country']).size().sort_values(ascending = False)


# Use `hole` to create a donut-like pie chart
fig = go.Figure(data=[go.Pie(labels=labels, values=values, hole=.3)])
fig.update_traces(textposition='inside')
fig.show()
```

LINEAR REGRESSION

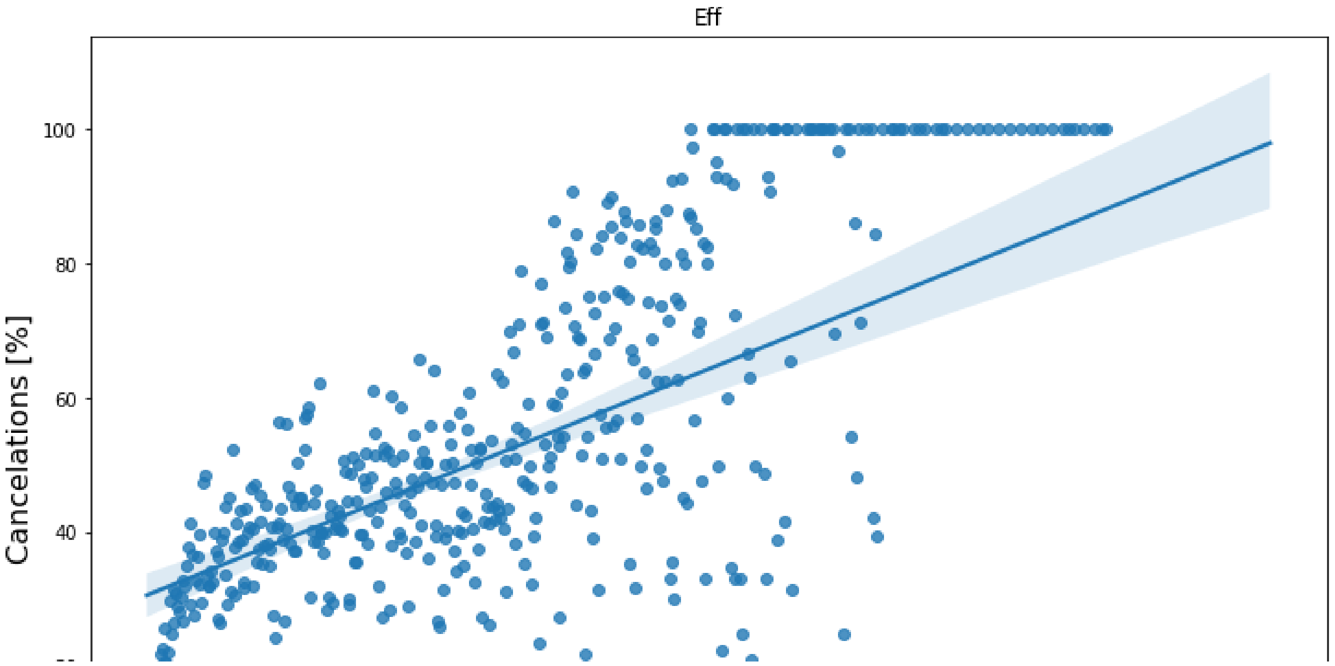
Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output).

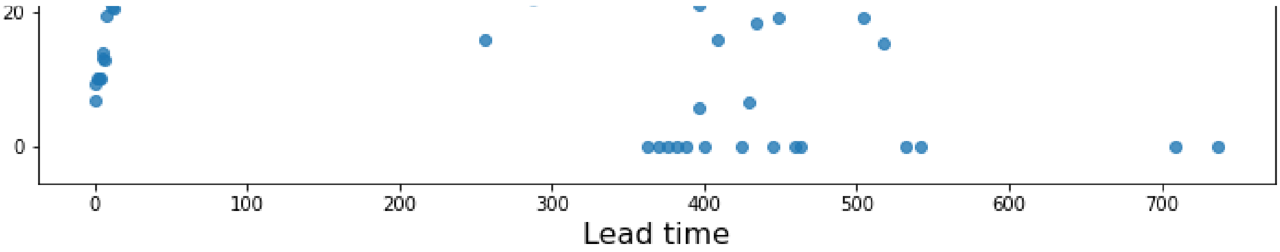
In [ ]:

```
lead_cancel_df_10 = df.groupby("lead_time")["is_canceled"].describe()

# show figure
plt.figure(figsize = (12, 8))
sns.regplot(x = lead_cancel_df_10.index, y = lead_cancel_df_10["mean"].values * 100)
plt.title("Effect of lead time on cancelation", fontsize=16)
plt.xlabel("Lead time", fontsize=16)
plt.ylabel("Cancelations [%]", fontsize=16)
plt.title("Eff")
```

Out[ ]:

```
Text(0.5, 1.0, 'Eff')
```





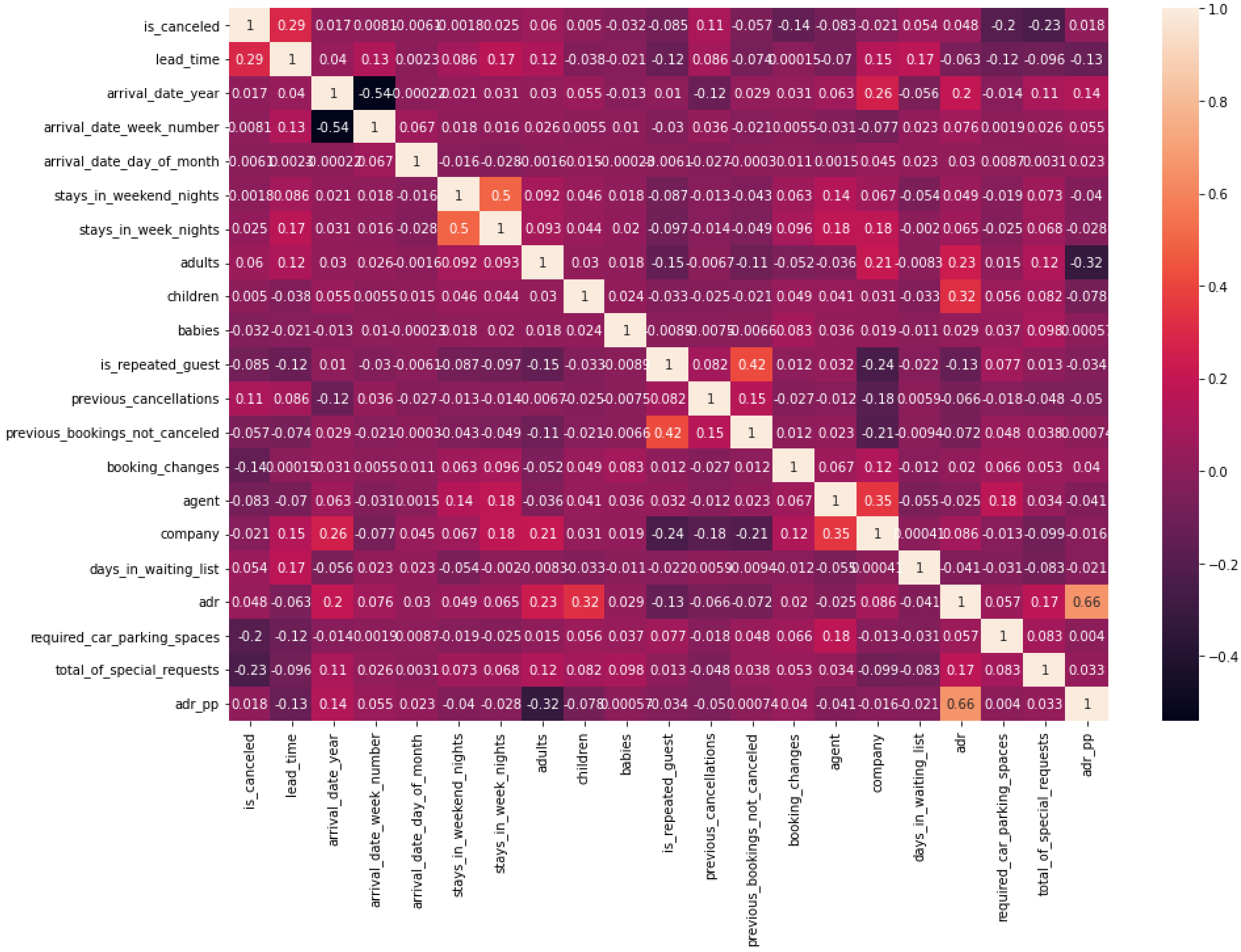
Bookings made a few days before the arrival date are rarely canceled, whereas bookings made over one year in advance are canceled very often.

HEAT MAPS

A heatmap contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. For a very different value a completely different colour can also be used.

In [ ]:

```
fig, ax = plt.subplots(figsize=(15,10))
sns.heatmap(df.corr(),annot=True);
```



CHOROPLETH MAP

A Choropleth Map is a map composed of colored polygons. It is used to represent spatial variations of a quantity.

In [ ]:

```
data_country = df[df['is_canceled']==0]['country'].value_counts().reset_index()
```

```
data_country.columns = ['Country','No.of Guests']
data_country
```

Out[ ]:

	Country	No.of Guests
0	PRT	21071
1	GBR	9676
2	FRA	8481
3	ESP	6391
4	DEU	6069
...	...	...
160	NPL	1
161	AIA	1
162	BHR	1
163	BHS	1
164	PLW	1

165 rows × 2 columns

In [ ]:

```
px.choropleth(data_country,
              locations=data_country['Country'],
              color=data_country['No.of Guests'],
              hover_name=data_country['Country'],
              title='Home Country of Guests')
```

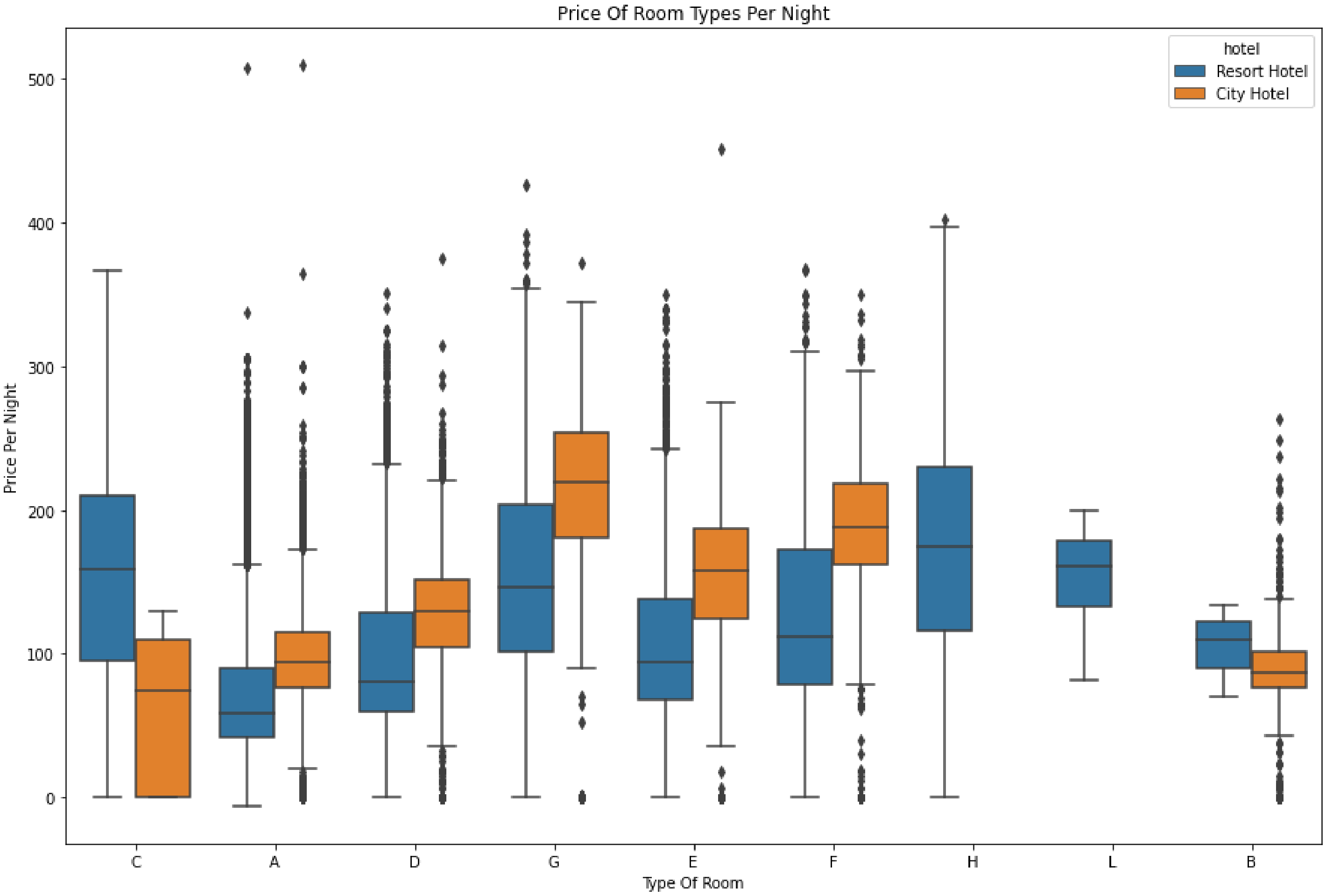
BOX PLOT

Boxplots are a measure of how well distributed the data in a data set is. It divides the data set into three

quartiles. This graph represents the minimum, maximum, median, first quartile and third quartile in the data set.

In [ ]:

```
plt.figure(figsize=(15,10))
sns.boxplot(data=df[df['is_canceled'] == 0],x='reserved_room_type',y='adr',hue='hotel')
plt.title('Price Of Room Types Per Night')
plt.xlabel('Type Of Room')
plt.ylabel('Price Per Night')
plt.show()
```

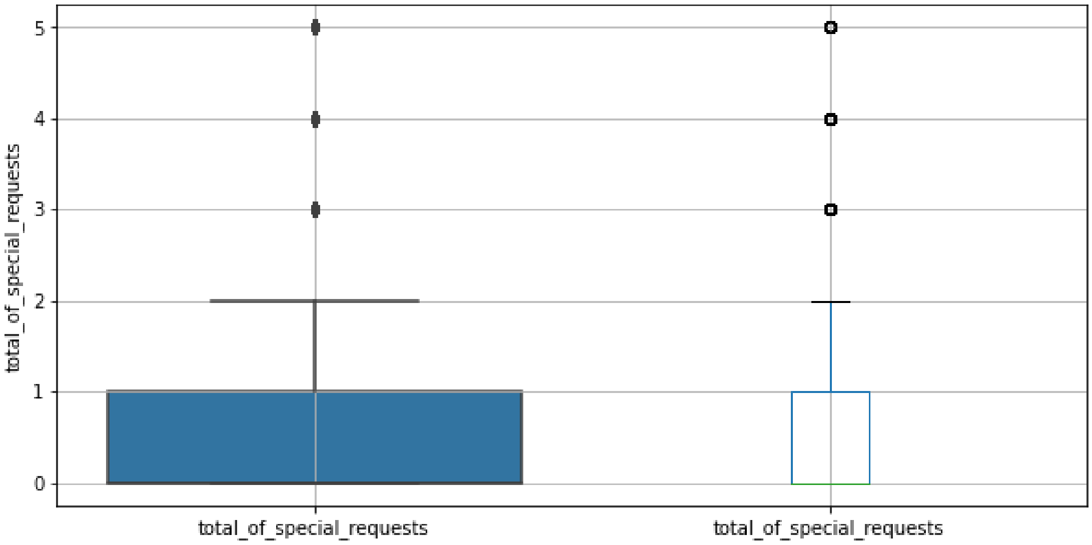


In [ ]:

```
sns.boxplot(y = df['total_of_special_requests']) # in pandas
df.boxplot(column = ['total_of_special_requests']) # in seaborn
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f136e116790>
```


total_of_special_requests
total_of_special_requests
total_of_special_requests