Project - Hotel Booking Cancellation Prediction

Problem Statement

Context

A significant number of hotel bookings are called off due to cancellations or no-shows. Typical reasons for cancellations include change of plans, scheduling conflicts, etc. This is often made easier by the option to do so free of charge or preferably at a low cost. This may be beneficial to hotel guests, but it is a less desirable and possibly revenue-diminishing factor for hotels to deal with. Such losses are particularly high on last-minute cancellations.

The new technologies involving online booking channels have dramatically changed customers' booking possibilities and behavior. This adds a further dimension to the challenge of how hotels handle cancellations, which are no longer limited to traditional booking and guest characteristics.

This pattern of cancellations of bookings impacts a hotel on various fronts:

1. **Loss of resources (revenue)** when the hotel cannot resell the room.
2. **Additional costs of distribution channels** by increasing commissions or paying for publicity to help sell these rooms.
3. **Lowering prices last minute**, so the hotel can resell a room, resulting in reducing the profit margin.
4. **Human resources to make arrangements** for the guests.

Objective

This increasing number of cancellations calls for a Machine Learning based solution that can help in predicting which booking is likely to be canceled. INN Hotels Group has a chain of hotels in Portugal - they are facing problems with this high number of booking cancellations and have reached out to your firm for data-driven solutions. You, as a Data Scientist, have to analyze the data provided to find which factors have a high influence on booking cancellations, build a predictive model that can predict which booking is going to be canceled in advance, and help in formulating profitable policies for cancellations and refunds.

Data Description

The data contains the different attributes of customers' booking details. The detailed data dictionary is given below:

Data Dictionary

- **Booking_ID:** Unique identifier of each booking
- **no_of_adults:** Number of adults
- **no_of_children:** Number of children
- **no_of_weekend_nights:** Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
- **no_of_week_nights:** Number of weekday nights (Monday to Friday) the guest stayed or booked to stay at the hotel
- **type_of_meal_plan:** Type of meal plan booked by the customer:
  - Not Selected – No meal plan selected
  - Meal Plan 1 – Breakfast
  - Meal Plan 2 – Half board (breakfast and one other meal)
  - Meal Plan 3 – Full board (breakfast, lunch, and dinner)
- **required_car_parking_space:** Does the customer require a car parking space? (0 - No, 1- Yes)
- **room_type_reserved:** Type of room reserved by the customer. The values are ciphered (encoded) by INN Hotels.
- **lead_time:** Number of days between the date of booking and the arrival date
- **arrival_year:** Year of arrival date
- **arrival_month:** Month of arrival date
- **arrival_date:** Date of the month
- **market_segment_type:** Market segment designation.
- **repeated_guest:** Is the customer a repeated guest? (0 - No, 1- Yes)
- **no_of_previous_cancellations:** Number of previous bookings that were canceled by the customer prior to the current booking
- **no_of_previous_bookings_not_canceled:** Number of previous bookings not canceled by the customer prior to the current booking
- **avg_price_per_room:** Average price per day of the reservation; prices of the rooms are dynamic. (in euros)
- **no_of_special_requests:** Total number of special requests made by the customer (e.g. high floor, view from the room, etc)
- **booking_status:** Flag indicating if the booking was canceled or not.

Importing the libraries required

In [1]:
```
# Importing the basic libraries we will require for the project
```

```
# Libraries to help with reading and manipulating data
import pandas as pd
import numpy as np

# Libaries to help with data visualization
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

warnings.filterwarnings('ignore')
```

Loading the dataset

In [2]:
```
hotel = pd.read_csv("INNHotelsGroup.csv")
```

In [3]:
```
# Copying data to another variable to avoid any changes to original data
data = hotel.copy()
```

Overview of the dataset

Viewing the first and last 5 rows of the dataset

Let's **view the first few rows and last few rows** of the dataset in order to understand its structure a little better.

We will use the head() and tail() methods from Pandas to do this.

In [4]:
```
data.head()
```

Out[4]:

	Booking_ID	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_meal_plan	required_car_parking_space	room_t
0	INN00001	2	0	1	2	Meal Plan 1	0	
1	INN00002	2	0	2	3	Not Selected	0	
2	INN00003	1	0	2	1	Meal Plan 1	0	
3	INN00004	2	0	0	2	Meal Plan 1	0	
4	INN00005	2	0	1	1	Not Selected	0	

In [5]:
```
data.tail()
```

Out[5]:

	Booking_ID	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_meal_plan	required_car_parking_space	ro
36270	INN36271	3	0	2	6	Meal Plan 1	0	
36271	INN36272	2	0	1	3	Meal Plan 1	0	
36272	INN36273	2	0	2	6	Meal Plan 1	0	
36273	INN36274	2	0	0	3	Not Selected	0	
36274	INN36275	2	0	1	2	Meal Plan 1	0	

Understanding the shape of the dataset

In [6]:
```
data.shape
```

Out[6]:
```
(36275, 19)
```

- The dataset has 36275 rows and 19 columns.

Checking the data types of the columns for the dataset

In [7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36275 entries, 0 to 36274
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   Booking_ID                            36275 non-null  object 
 1   no_of_adults                          36275 non-null  int64  
 2   no_of_children                        36275 non-null  int64  
 3   no_of_weekend_nights                  36275 non-null  int64  
 4   no_of_week_nights                     36275 non-null  int64  
 5   type_of_meal_plan                     36275 non-null  object 
 6   required_car_parking_space            36275 non-null  int64  
 7   room_type_reserved                    36275 non-null  object 
 8   lead_time                             36275 non-null  int64  
 9   arrival_year                          36275 non-null  int64  
 10  arrival_month                         36275 non-null  int64  
 11  arrival_date                          36275 non-null  int64  
 12  market_segment_type                   36275 non-null  object 
 13  repeated_guest                        36275 non-null  int64  
 14  no_of_previous_cancellations          36275 non-null  int64  
 15  no_of_previous_bookings_not_canceled  36275 non-null  int64  
 16  avg_price_per_room                    36275 non-null  float64
 17  no_of_special_requests                36275 non-null  int64  
 18  booking_status                        36275 non-null  object 
dtypes: float64(1), int64(13), object(5)
memory usage: 5.3+ MB
```

- `Booking_ID`, `type_of_meal_plan`, `room_type_reserved`, `market_segment_type`, and `booking_status` are of object type while rest columns are numeric in nature.
- There are no null values in the dataset.

Dropping duplicate valuesmif any

In [8]:
```
# checking for duplicate values
data.duplicated().sum()
```

Out[8]:
```
0
```

- There are **no duplicate values** in the data.

Dropping the unique values column

Let's drop the Booking_ID column first before we proceed forward, as a column with unique values will have almost no predictive power for the Machine Learning problem at hand.

In [9]:
```
data = data.drop(["Booking_ID"], axis=1)
```

In [10]:
```
data.head()
```

Out[10]:

	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_meal_plan	required_car_parking_space	room_type_reserved
0	2	0	1	2	Meal Plan 1	0	Room_Type 1
1	2	0	2	3	Not Selected	0	Room_Type 1
2	1	0	2	1	Meal Plan 1	0	Room_Type 1
3	2	0	0	2	Meal Plan 1	0	Room_Type 1
4	2	0	1	1	Not Selected	0	Room_Type 1

Checking the summary statistics of the dataset

In [11]:
```
data.describe().T
```

Out[11]:

	count	mean	std	min	25%	50%	75%	max
no_of_adults	36275.0	1.844962	0.518715	0.0	2.0	2.00	2.0	4.0
no_of_children	36275.0	0.105279	0.402648	0.0	0.0	0.00	0.0	10.0
no_of_weekend_nights	36275.0	0.810724	0.870644	0.0	0.0	1.00	2.0	7.0
no_of_week_nights	36275.0	2.204300	1.410905	0.0	1.0	2.00	3.0	17.0
required_car_parking_space	36275.0	0.030986	0.173281	0.0	0.0	0.00	0.0	1.0
lead_time	36275.0	85.232557	85.930817	0.0	17.0	57.00	126.0	443.0
arrival_year	36275.0	2017.820427	0.383836	2017.0	2018.0	2018.00	2018.0	2018.0
arrival_month	36275.0	7.423653	3.069894	1.0	5.0	8.00	10.0	12.0
arrival_date	36275.0	15.596995	8.740447	1.0	8.0	16.00	23.0	31.0
repeated_guest	36275.0	0.025637	0.158053	0.0	0.0	0.00	0.0	1.0
no_of_previous_cancellations	36275.0	0.023349	0.368331	0.0	0.0	0.00	0.0	13.0
no_of_previous_bookings_not_canceled	36275.0	0.153411	1.754171	0.0	0.0	0.00	0.0	58.0
avg_price_per_room	36275.0	103.423539	35.089424	0.0	80.3	99.45	120.0	540.0
no_of_special_requests	36275.0	0.619655	0.786236	0.0	0.0	0.00	1.0	5.0

Observations

- The number of adults per room is an average of 2.
- The number of children is maximum of 10. This is very unlikely.
- Some of the room don't have adults in them which is very unlikely.
- For the number of previous cancellation, the maximum is 13
- The average price per room is about 103.4. Some of the room has no price which is unlikely or are complimentary or promotional campaign by the hotel to their guests.
- The number of previous bookings not cancelled is more than those cancelled with maximum number of 58 as against 13 for the previously cancelled bookings

Exploratory Data Analysis

Univariate Analysis

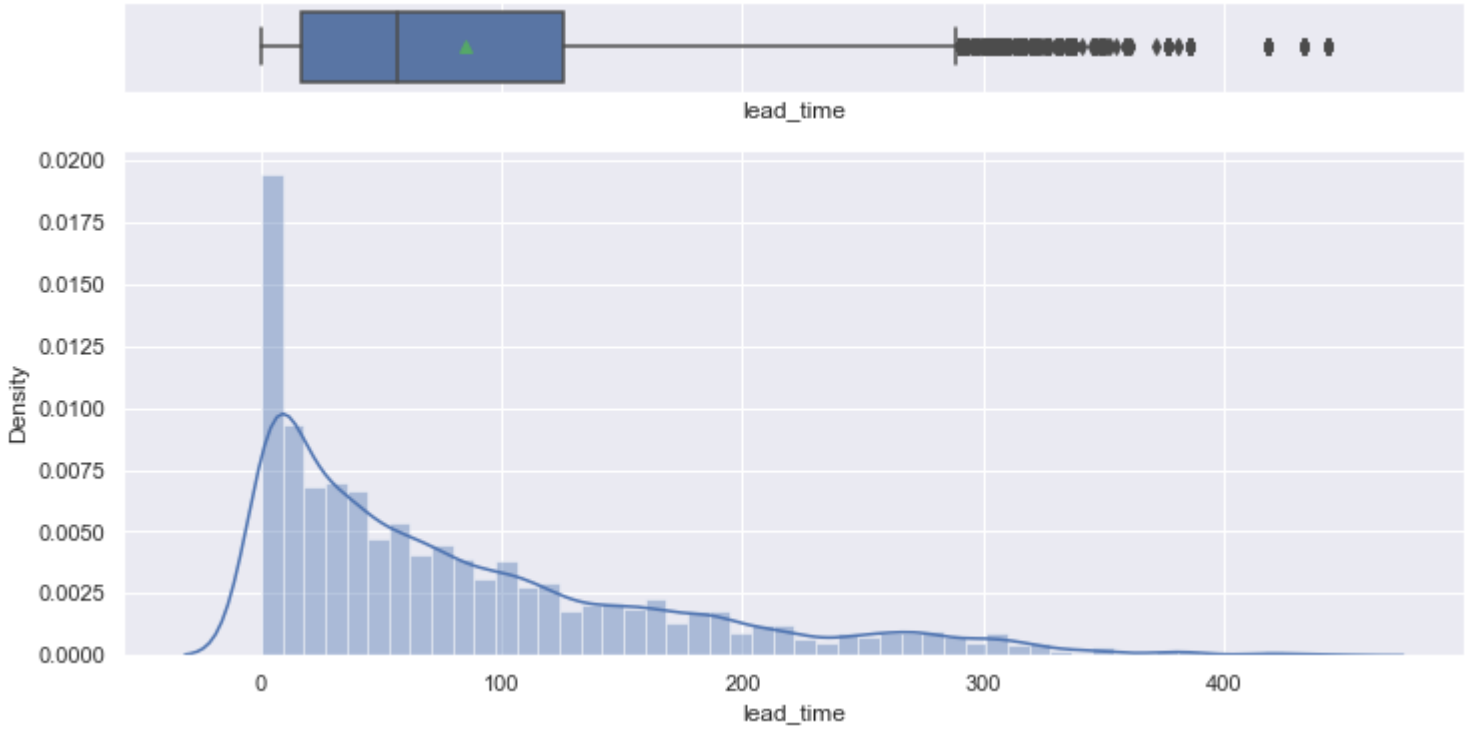
Let's explore these variables in some more depth by observing their distributions.

We will first define a **hist_box() function** that provides both a boxplot and a histogram in the same visual, with which we can perform univariate analysis on the columns of this dataset.

In [12]:
```
# Defining the hist_box() function
def hist_box(data,col):
  f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={'height_ratios': (0.15, 0.85)}, figsize=(12,6))
  # Adding a graph in each part
  sns.boxplot(data[col], ax=ax_box, showmeans=True)
  sns.distplot(data[col], ax=ax_hist)
  plt.show()
```

Plotting the histogram and box plot for the variable `Lead Time` using the hist_box function

In [13]:
```
# Remove _________ and complete the code
hist_box(data,'lead_time')
```



Observations

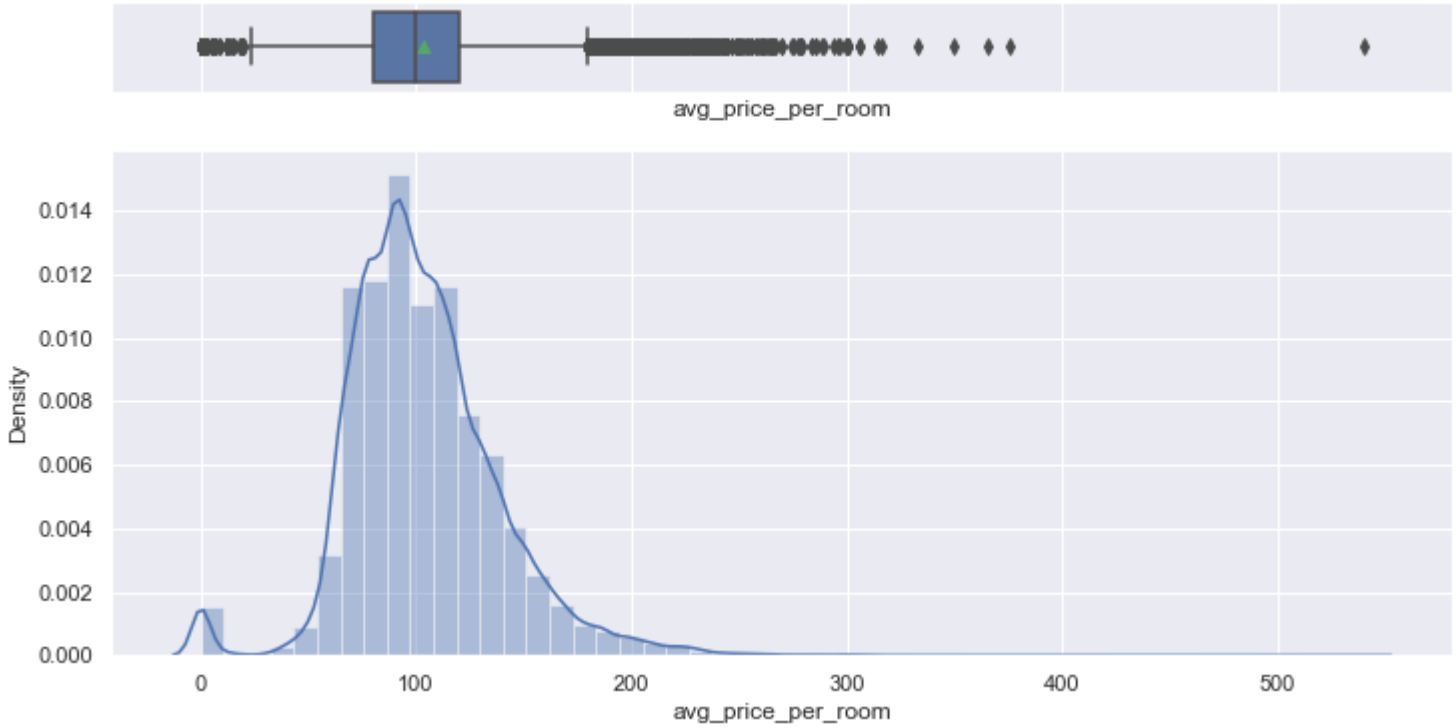
- The distribution of lead time is right skewed
- Number of days between the date of booking and the arrival date is in smaller increases.

Plotting the histogram and box plot for the variable `Average Price per Room` using the hist_box function.

In [14]:
```
data['avg_price_per_room'].skew()
```

Out[14]:
```
0.6671328746979995
```

In [15]:
```
hist_box(data,"avg_price_per_room")
```



Observation

- The average price per room is almost normally distributed although the box plot shows some outliers.

Interestingly some rooms have a price equal to 0. Let's check them.

In [16]:
```
data[data["avg_price_per_room"] == 0]
```

Out[16]:

	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_meal_plan	required_car_parking_space	room_type_rese
63	1	0	0	1	Meal Plan 1	0	Room_Ty
145	1	0	0	2	Meal Plan 1	0	Room_Ty
209	1	0	0	0	Meal Plan 1	0	Room_Ty
266	1	0	0	2	Meal Plan 1	0	Room_Ty
267	1	0	2	1	Meal Plan 1	0	Room_Ty
...	...	...	...	...	...	...	
35983	1	0	0	1	Meal Plan 1	0	Room_Ty
36080	1	0	1	1	Meal Plan 1	0	Room_Ty
36114	1	0	0	1	Meal Plan 1	0	Room_Ty
36217	2	0	2	1	Meal Plan 1	0	Room_Ty
36250	1	0	0	2	Meal Plan 2	0	Room_Ty

545 rows × 18 columns

- There are quite a few hotel rooms which have a price equal to 0.
- In the market segment column, it looks like many values are complementary.

In [17]:
```
data.loc[data["avg_price_per_room"] == 0, "market_segment_type"].value_counts()
```

Out[17]:
```
Complementary    354
Online           191
Name: market_segment_type, dtype: int64
```

- It makes sense that most values with room prices equal to 0 are the rooms given as complimentary service from the hotel.
- The rooms booked online must be a part of some promotional campaign done by the hotel.

In [18]:
```
# Calculating the 25th quantile
Q1 = data["avg_price_per_room"].quantile(0.25)

# Calculating the 75th quantile
```

```
Q3 = data["avg_price_per_room"].quantile(0.75)

# Calculating IQR
IQR = Q3 - Q1

# Calculating value of upper whisker
Upper_Whisker = Q3 + 1.5 * IQR
Upper_Whisker
```

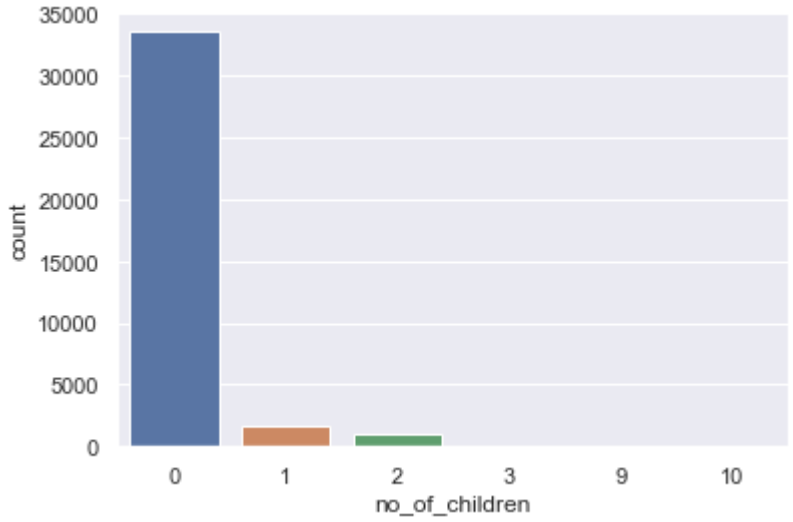
Out[18]: 179.55

In [19]:
```
# assigning the outliers the value of upper whisker
data.loc[data["avg_price_per_room"] >= 500, "avg_price_per_room"] = Upper_Whisker
```

Let's understand the distribution of the categorical variables

Number of Children

In [20]:
```
sns.countplot(data['no_of_children'])
plt.show()
```



In [21]:
```
data['no_of_children'].value_counts(normalize=True)
```

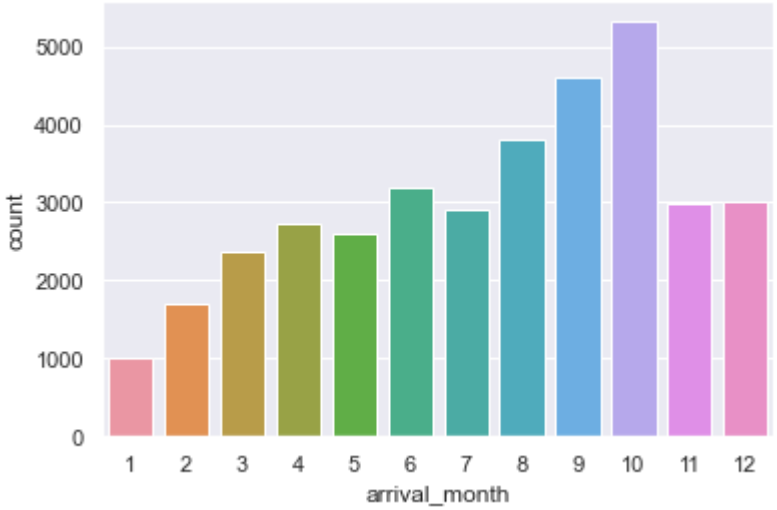
Out[21]:
```
0     0.925624
1     0.044604
2     0.029166
3     0.000524
9     0.000055
10    0.000028
Name: no_of_children, dtype: float64
```

- Customers were not travelling with children in 93% of cases.
- There are some values in the data where the number of children is 9 or 10, which is highly unlikely.
- We will replace these values with the maximum value of 3 children.

In [22]:
```
# replacing 9, and 10 children with 3
data["no_of_children"] = data["no_of_children"].replace([9, 10], 3)
```

Arrival Month

In [23]:
```
sns.countplot(data["arrival_month"])
plt.show()
```



In [24]:
```
data['arrival_month'].value_counts(normalize=True)
```

Out[24]:
```
10    0.146575
9     0.127112
8     0.105114
6     0.088298
12    0.083280
11    0.082150
7     0.080496
4     0.075424
5     0.071620
3     0.065003
2     0.046975
1     0.027953
Name: arrival_month, dtype: float64
```

- October is the busiest month for hotel arrivals followed by September and August. **Over 35% of all bookings**, as we see in the above table, were for one of these three months.
- Around 14.7% of the bookings were made for an October arrival.

Booking Status

In [25]:
```
sns.countplot(data["booking_status"])
plt.show()
```

In [26]:
```
data['booking_status'].value_counts(normalize=True)
```

Out[26]:
```
Not_Canceled    0.672364
Canceled        0.327636
Name: booking_status, dtype: float64
```

- 32.8% of the bookings were canceled by the customers.

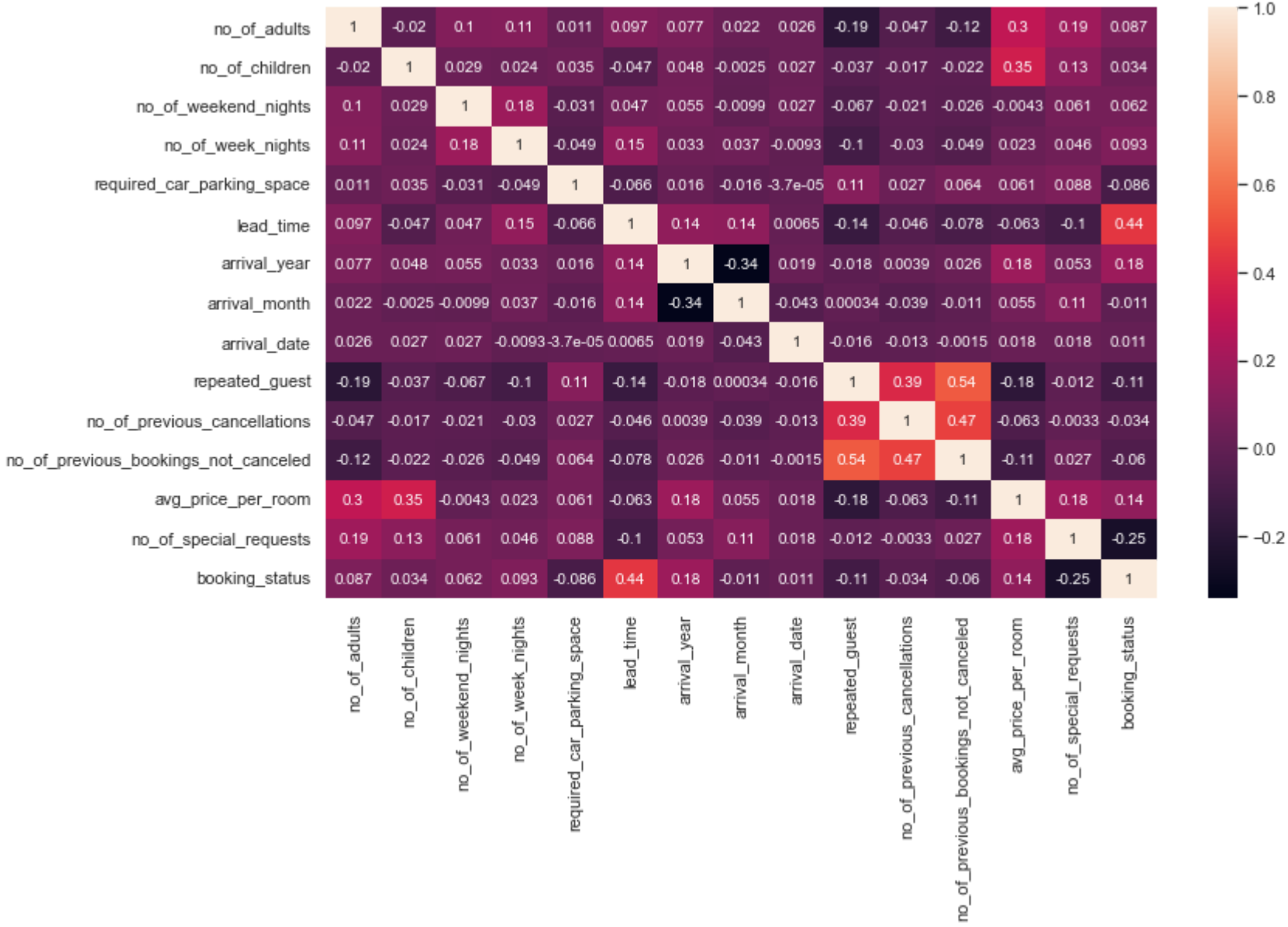
Let's encode Canceled bookings to 1 and Not_Canceled as 0 for further analysis

In [27]:
```
data["booking_status"] = data["booking_status"].apply(
    lambda x: 1 if x == "Canceled" else 0
)
```

Bivariate Analysis

Finding and visualizing the correlation matrix using a heatmap

In [28]:
```
cols_list = data.select_dtypes(include=np.number).columns.tolist()

plt.figure(figsize=(12, 7))
sns.heatmap(data.corr(), annot=True)
plt.show()
```



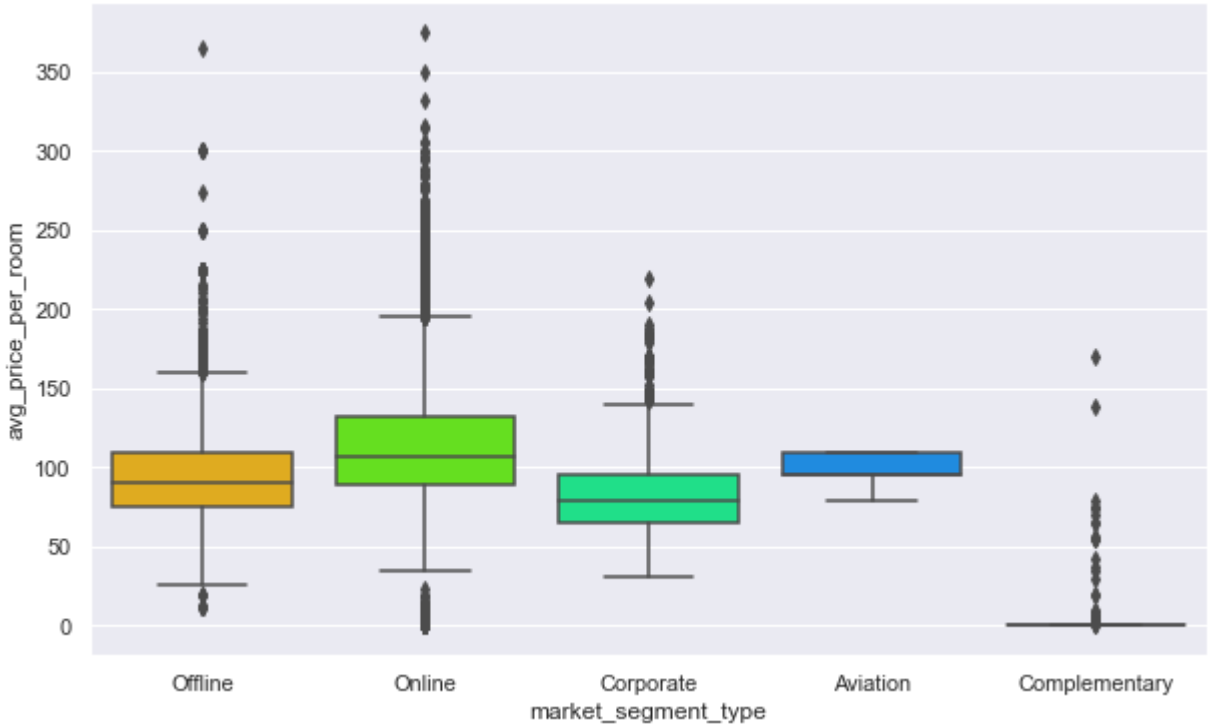
Observations

- There is some correlation between repeated guests and the number if previously booked not cancelled. This is possible as some guest that had stayed in the hotel previously had booked again and dont need to cancel maybe due to the great service they enjoyed and would like to stay at the hotel again.

- Most of the variables are independent.

Hotel rates are dynamic and change according to demand and customer demographics. Let's see how prices vary across different market segments

In [29]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(
    data=data, x="market_segment_type", y="avg_price_per_room", palette="gist_rainbow"
)
plt.show()
```



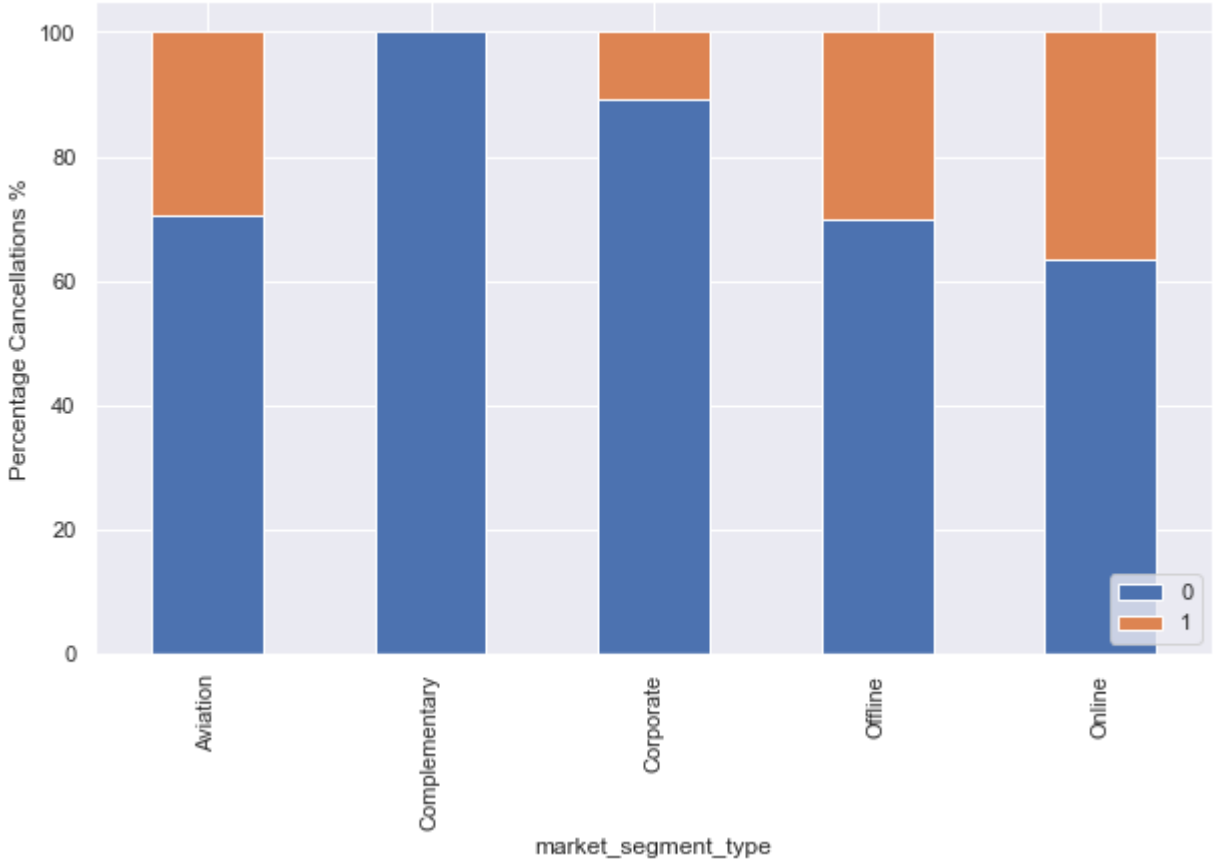
- Rooms booked online have high variations in prices.
- The offline and corporate room prices are almost similar.
- Complementary market segment gets the rooms at very low prices, which makes sense.

We will define a **stacked barplot()** function to help analyse how the target variable varies across predictor categories.

In [30]:
```
# Defining the stacked_barplot() function
def stacked_barplot(data,predictor,target,figsize=(10,6)):
  (pd.crosstab(data[predictor],data[target],normalize='index')*100).plot(kind='bar',figsize=figsize,stacked=True)
  plt.legend(loc="lower right")
  plt.ylabel('Percentage Cancellations %')
```

Plotting the stacked barplot for the variable `Market Segment Type` against the target variable `Booking Status` using the stacked_barplot function

In [31]:
```
stacked_barplot(data,'market_segment_type','booking_status' )
```



Observations

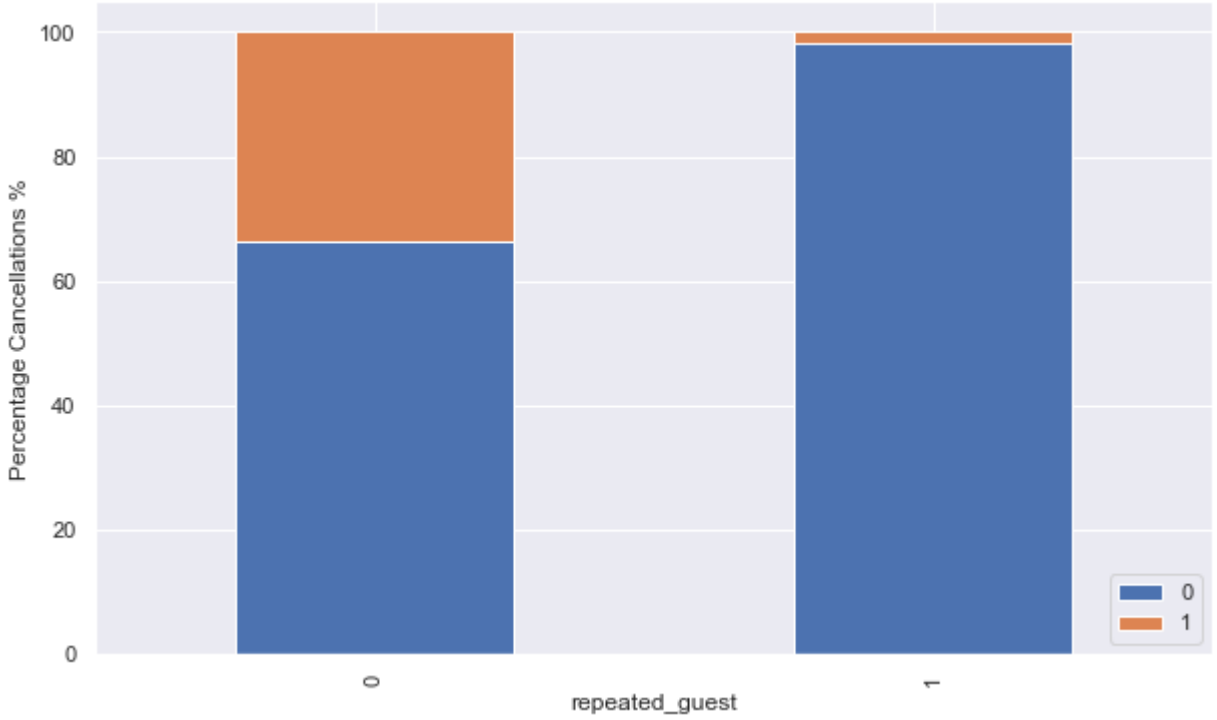
- There are no cancellation in the complimentary market segment type. This is a complimentary service which is free and so there will be no cancellation here

- Amongst the market segment with cancellation, the percentage of cancellation for the online market segment is the highest while corporate market segment is the lowest. Cancellation of booking in Corporate organizations are not frequent unless situations beyond their control such as change of dates of their events.

Question 3.3: Plot the stacked barplot for the variable `Repeated Guest` against the target variable `Booking Status` using the stacked_barplot function provided and write your insights. (1 Mark)

Repeating guests are the guests who stay in the hotel often and are important to brand equity.

In [32]:
```
stacked_barplot(data, 'repeated_guest','booking_status')
```

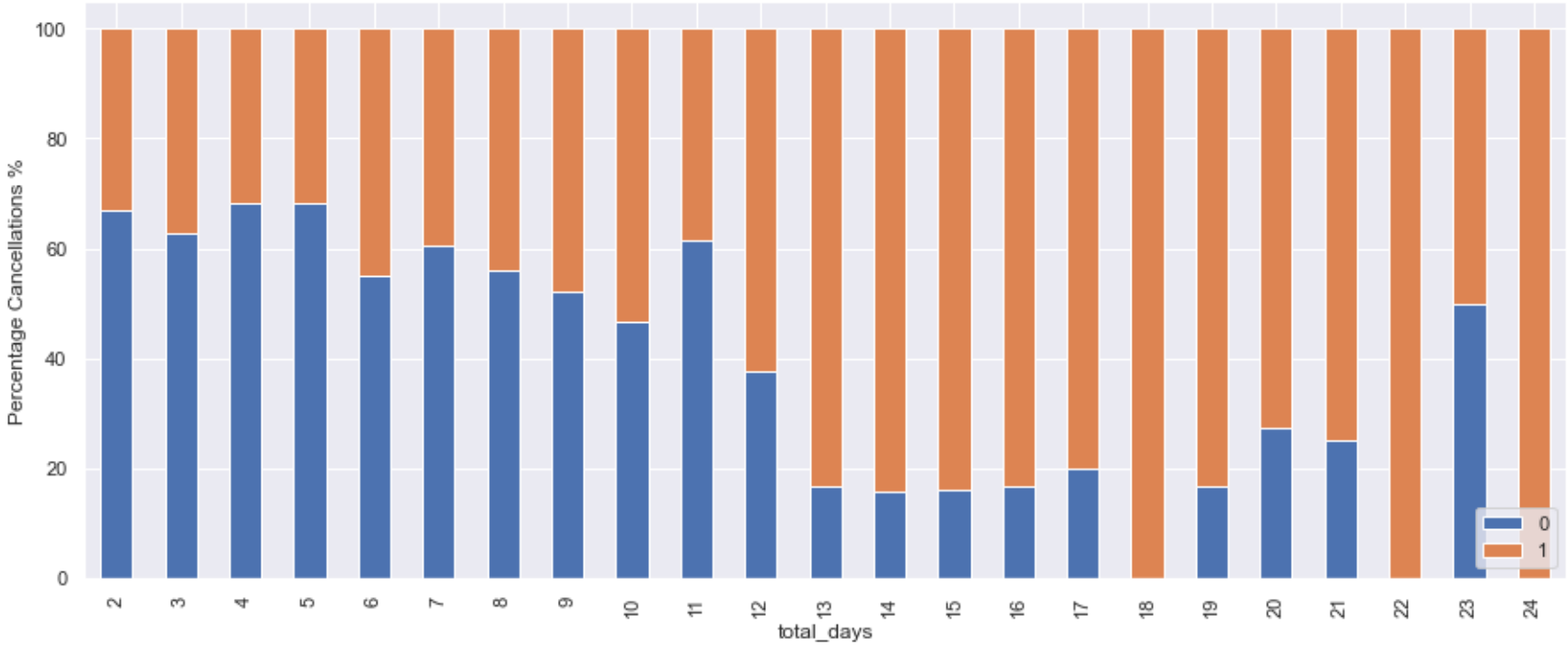


Observations

- Repeated guest rarely cancel their reservation and if they do, it may due to situations beyond their control.
- This is because guest they had stayed in the hotel previously and had enjoyed the service of the hotel. They are comfortable with the brand of INN Hotels Group
- First time guest tends to cancel more frequently as they have not experience the service of the hotel before.

Let's analyze the customer who stayed for at least a day at the hotel.

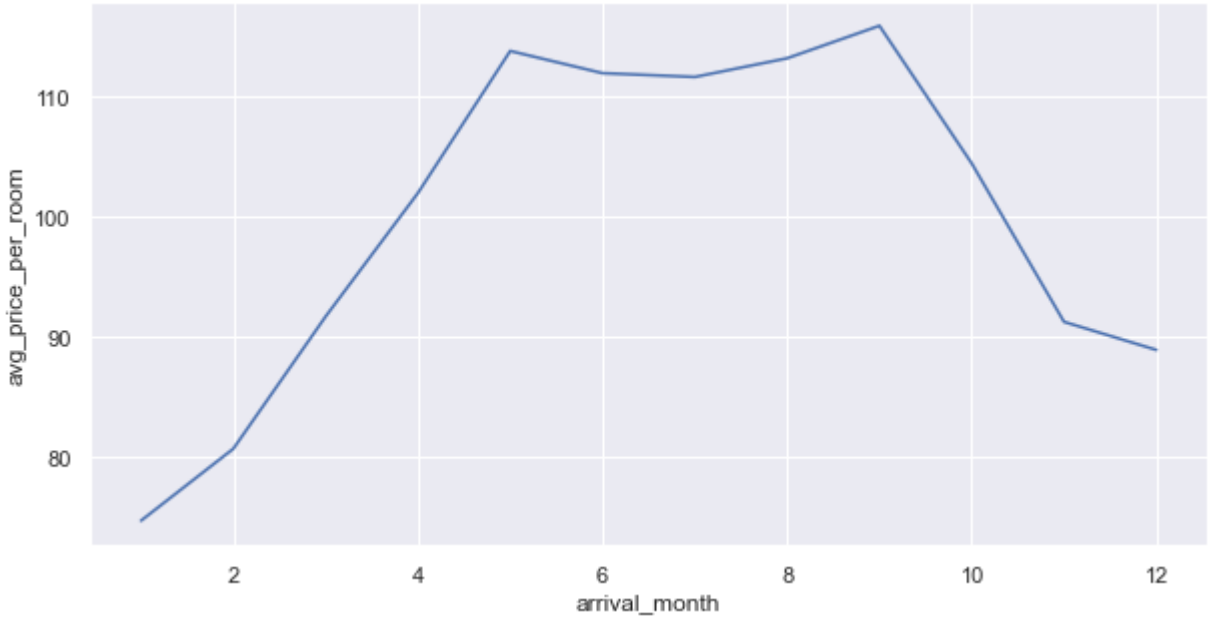
In [33]:
```
stay_data = data[(data["no_of_week_nights"] > 0) & (data["no_of_weekend_nights"] > 0)]
stay_data["total_days"] = (stay_data["no_of_week_nights"] + stay_data["no_of_weekend_nights"])

stacked_barplot(stay_data, "total_days", "booking_status",figsize=(15,6))
```



- The general trend is that the chances of cancellation increase as the number of days the customer planned to stay at the hotel increases.

As hotel room prices are dynamic, Let's see how the prices vary across different months

In [34]:
```
plt.figure(figsize=(10, 5))
sns.lineplot(y=data["avg_price_per_room"], x=data["arrival_month"], ci=None)
plt.show()
```



- The price of rooms is highest in May to September - around 115 euros per room.

Recommendations

- From our analysis, the average price per room is one of the reasons why hotels bookings are cancelled. The company should try to reduce the prices of the rooms when it is not the peak periods and make these prices during peak periods (between May and October) reasonably lower than their competitors to retain hotel bookings. If their prices are considerably lower with low impact on their profit margin, they tend to have more bookings that are not cancelled and have higher number of repeated guests.
- The company should carry out frequent campaigns and complimentary services to attract more clients in each market segment especially the online segment. Rooms booked online have high variations in prices and they also tend to cancel their bookings more frequently than other segments. Lower prices and frequent online campaigns can help retain the bookings of the online segments.
- To make first time guests a repeated guest, INN Hotel should make their customers feel amazing and special by frequently communicating with them. This may be in form of emails, paid questionnaires and surveys, discounts on hotel room for the client and their referrals. As the number of hotels booking cancellation from repeated guest is quite low, ensuring the first-time guest are converted to frequent guest will help reduce the booking cancellations and retain more customers.
- The general trend is that the chances of cancellation increase as the number of days the customer planned to stay at the hotel increases. For bookings with longer stay, INN Group should propose more services. The INN Group can propose complimentary

dinner for booking that tend to stay for more than 5 days while others enjoy only complimentary breakfast. They might also propose discount for longer stays which would tend to sway the client in INN Group's favor to not cancel their bookings and enjoy these complimentary services.