Business Problem

1. **High Cancellation Rates**: Both City Hotel and Resort Hotel are experiencing high cancellation rates.

2. **Impact on Revenue**: Cancellations are leading to decreased revenue and inefficient room utilization.

3. **Primary Goal**: Reducing cancellation rates is a priority for both hotels to improve revenue generation and operational efficiency.

4. **Purpose of Analysis**: The report focuses on analyzing hotel booking cancellations and identifying factors affecting business performance and annual revenue.

5. **Business Recommendations**: The goal is to provide actionable insights and recommendations to address the high cancellation rates and improve hotel performance.

In [1]:

```
#Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv(r'hotel_bookings.csv') #I df.head() #Showing top 5 values
```

Out[2]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_nu
0	Resort Hotel	0	342	2015	July	
1	Resort Hotel	0	737	2015	July	
2	Resort Hotel	0	7	2015	July	
3	Resort Hotel	0	13	2015	July	
4	Resort Hotel	0	14	2015	July	

5 rows × 32 columns

In [3]:

```
df.shape #showing shape of the data, there are 119390 rows and 32 columns
```

```
Out[3]:  (119390, 32)
```

```
In [4]:  df.columns #Showing columns naa=me
```

```
Out[4]:  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 [5]:  df.info() #showing informatin about columns
```

```
<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
```

In [6]:
```
df.dtypes.value_counts() #counting data types
```

Out[6]:
```
int64      16
object     12
float64     4
Name: count, dtype: int64
```

In [7]:
```
#"reservation_status_date" data type should be datetime but is object
df['reservation_status_date'] = pd.to_datetime(df['reservation_status_date'], forma
df['Month_Name'] = df['reservation_status_date'].dt.month_name() #Extracting month
```

In [8]:
```
df.dtypes #Showing data types
```

Out[8]:

```
hotel                                     object
is_canceled                                int64
lead_time                                  int64
arrival_date_year                          int64
arrival_date_month                        object
arrival_date_week_number                   int64
arrival_date_day_of_month                  int64
stays_in_weekend_nights                    int64
stays_in_week_nights                       int64
adults                                     int64
children                                 float64
babies                                     int64
meal                                      object
country                                   object
market_segment                            object
distribution_channel                      object
is_repeated_guest                          int64
previous_cancellations                     int64
previous_bookings_not_canceled             int64
reserved_room_type                        object
assigned_room_type                        object
booking_changes                            int64
deposit_type                              object
agent                                    float64
company                                  float64
days_in_waiting_list                       int64
customer_type                             object
adr                                      float64
required_car_parking_spaces                int64
total_of_special_requests                  int64
reservation_status                        object
reservation_status_date           datetime64[ns]
Month_Name                                object
dtype: object
```

In [9]:

```
df.isnull().sum() #Showing null values
```

Out[9]:

```
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
Month_Name                             0
dtype: int64
```

In [10]:

```
df.drop(['company','agent'], inplace=True ,axis=1) #Dropping columns agent and comp
df.dropna(inplace=True) #Dropping null values
```

In [11]:

```
df.isnull().sum() #Showing null values
```

Out[11]:

```
hotel                             0
is_canceled                       0
lead_time                         0
arrival_date_year                 0
arrival_date_month                0
arrival_date_week_number          0
arrival_date_day_of_month         0
stays_in_weekend_nights           0
stays_in_week_nights              0
adults                            0
children                          0
babies                            0
meal                              0
country                           0
market_segment                    0
distribution_channel              0
is_repeated_guest                 0
previous_cancellations            0
previous_bookings_not_canceled    0
reserved_room_type                0
assigned_room_type                0
booking_changes                   0
deposit_type                      0
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
Month_Name                        0
dtype: int64
```

In [12]:

```
df.describe() #Showing descriptive statistic overview
```

Out[12]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_da
count	118898.000000	118898.000000	118898.000000	118898.000000	
mean	0.371352	104.311435	2016.157656	27.166555	
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	161.000000	2017.000000	38.000000	
max	1.000000	737.000000	2017.000000	53.000000	
std	0.483168	106.903309	0.707459	13.589971	

In [13]:

```
# Creating "total_stay" column
df['total_stay'] = df['stays_in_weekend_nights'] + df['stays_in_week_nights']
```

```
df.head()
```

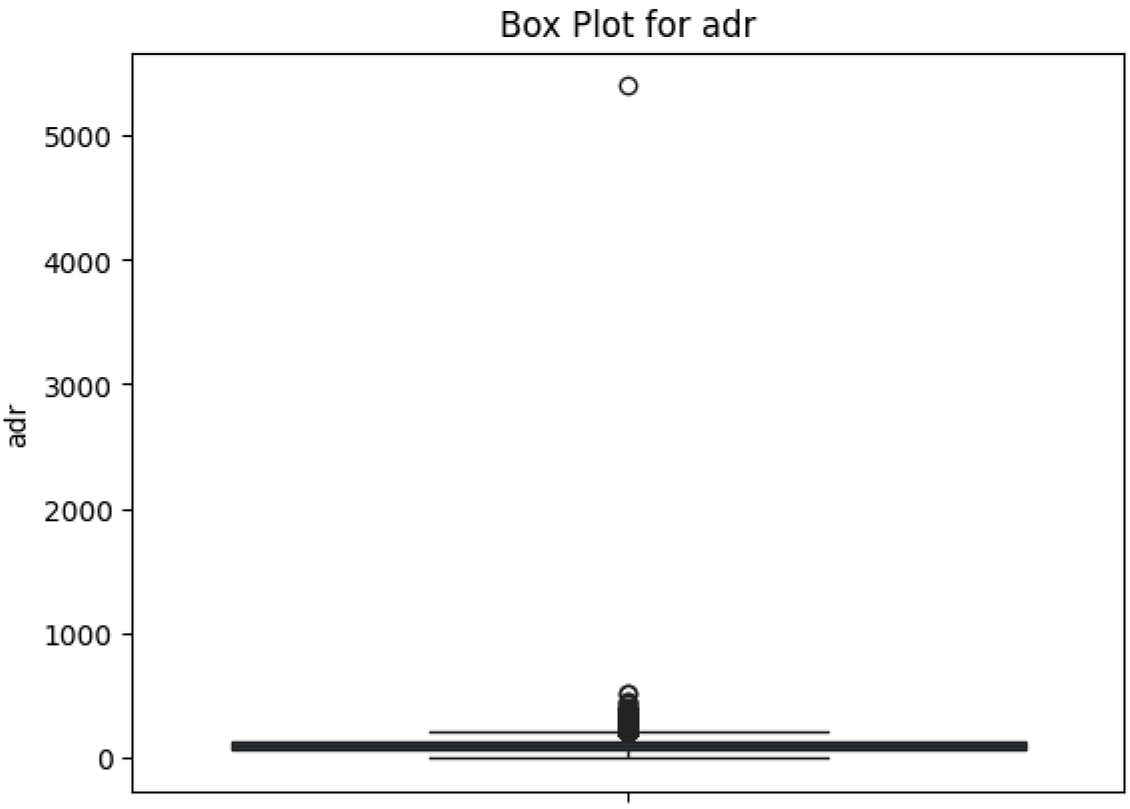
Out[13]:

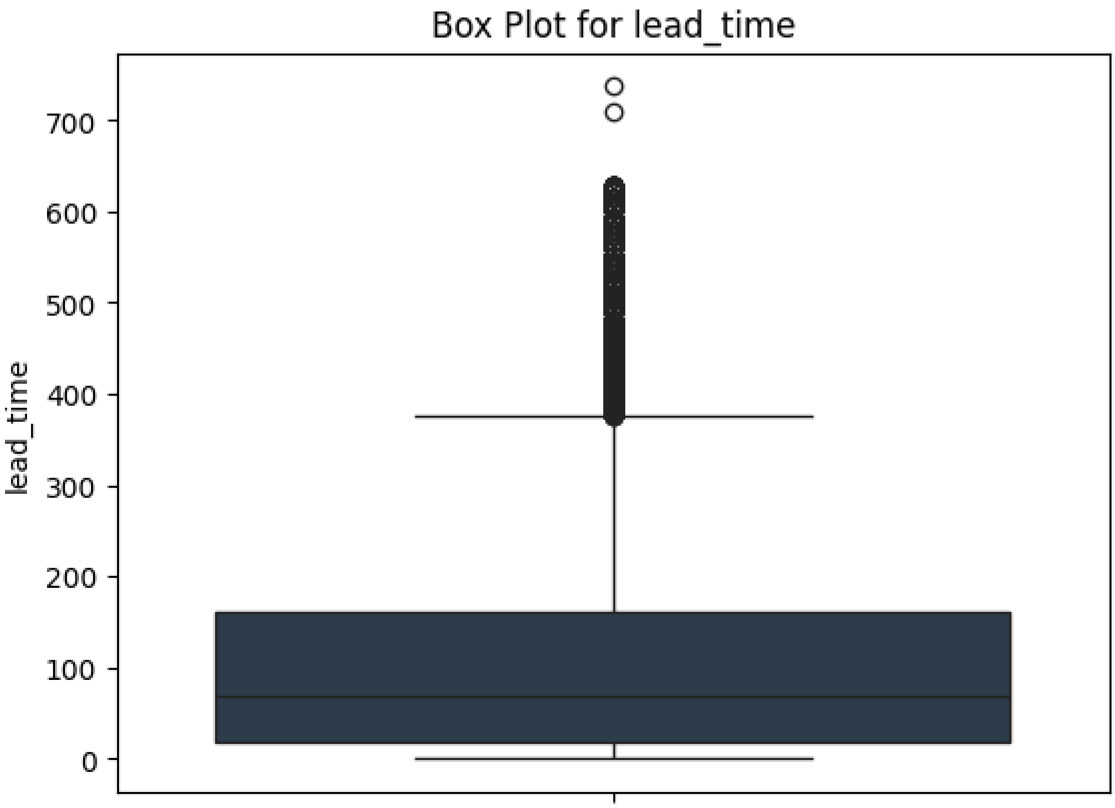
	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_nu
0	Resort Hotel	0	342	2015	July	
1	Resort Hotel	0	737	2015	July	
2	Resort Hotel	0	7	2015	July	
3	Resort Hotel	0	13	2015	July	
4	Resort Hotel	0	14	2015	July	

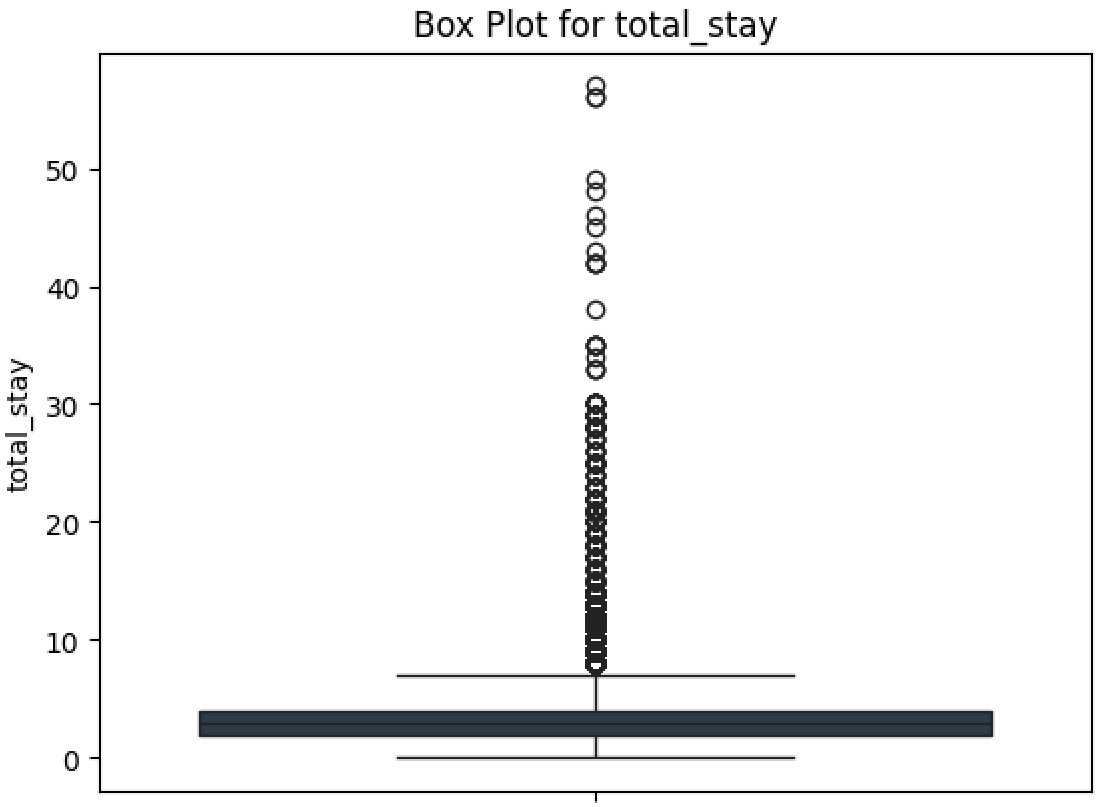
5 rows × 32 columns

In [14]:

```
#Checking outlier
for i in ['adr', 'lead_time','total_stay']:
    plt.title(f'Box Plot for {i}')
    sns.boxplot(df[i],
                color='#2c3e50'
)
    plt.show()
```





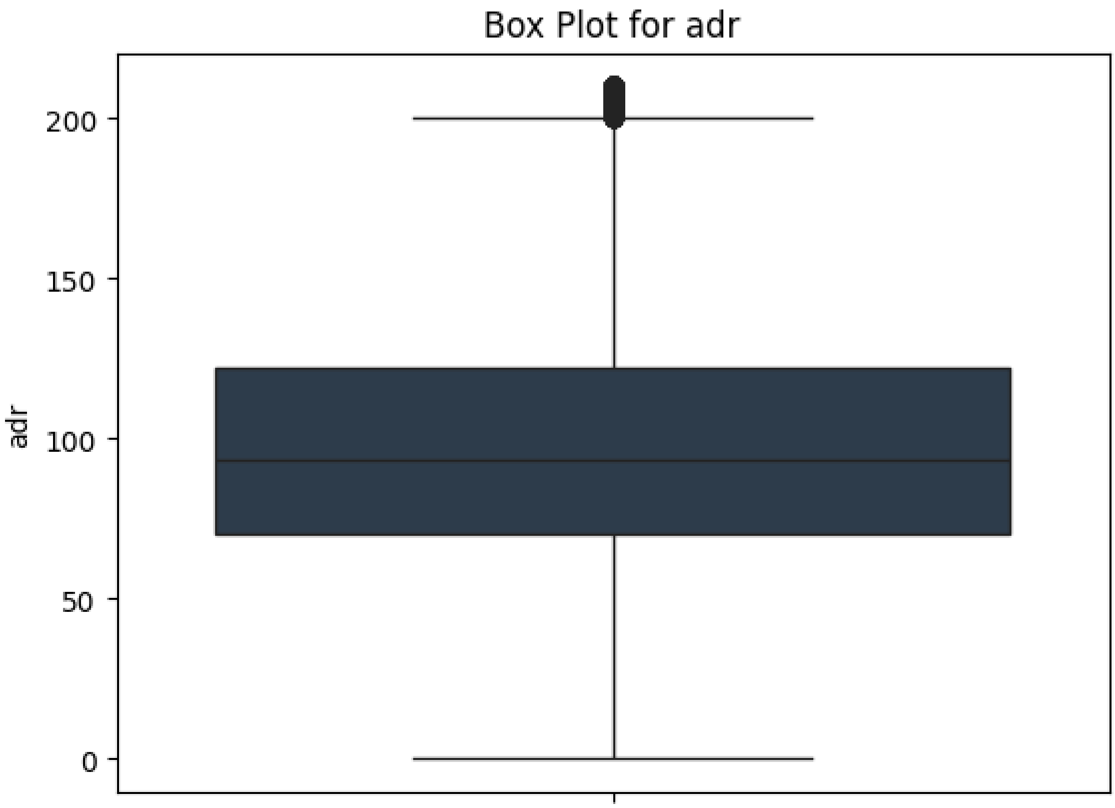


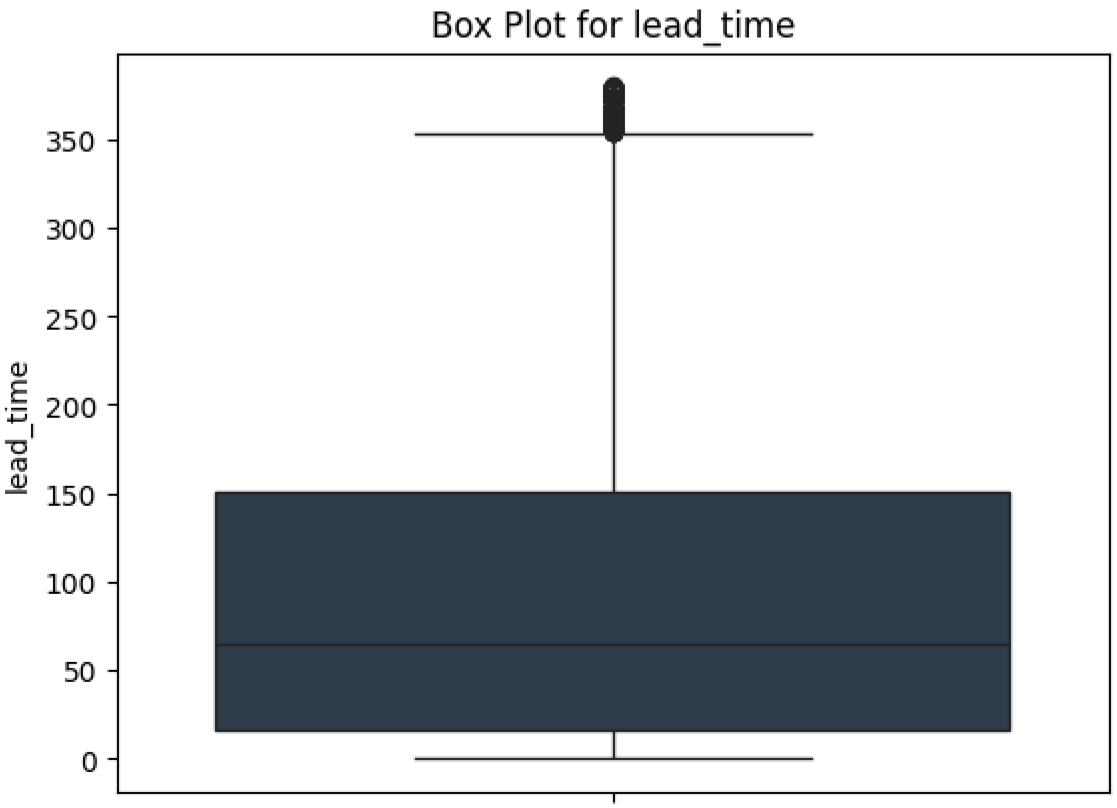
In [15]:

```
#Removing outliers by Interquartile Range method
for i in ['adr', 'lead_time','total_stay']: #For Loop Function
    q3 = np.percentile(df[i],75) #percentile of 75
    q1 = np.percentile(df[i],25) #percentile of 25
    IQR = q3-q1 #calculating interquartile range
    upper_bound = q3 + 1.5*IQR #calculating upper bound
    lower_bound = q1 - 1.5*IQR #calculating lower bound
    df = df[(df[i]<=upper_bound) & (df[i]>=lower_bound)]
```

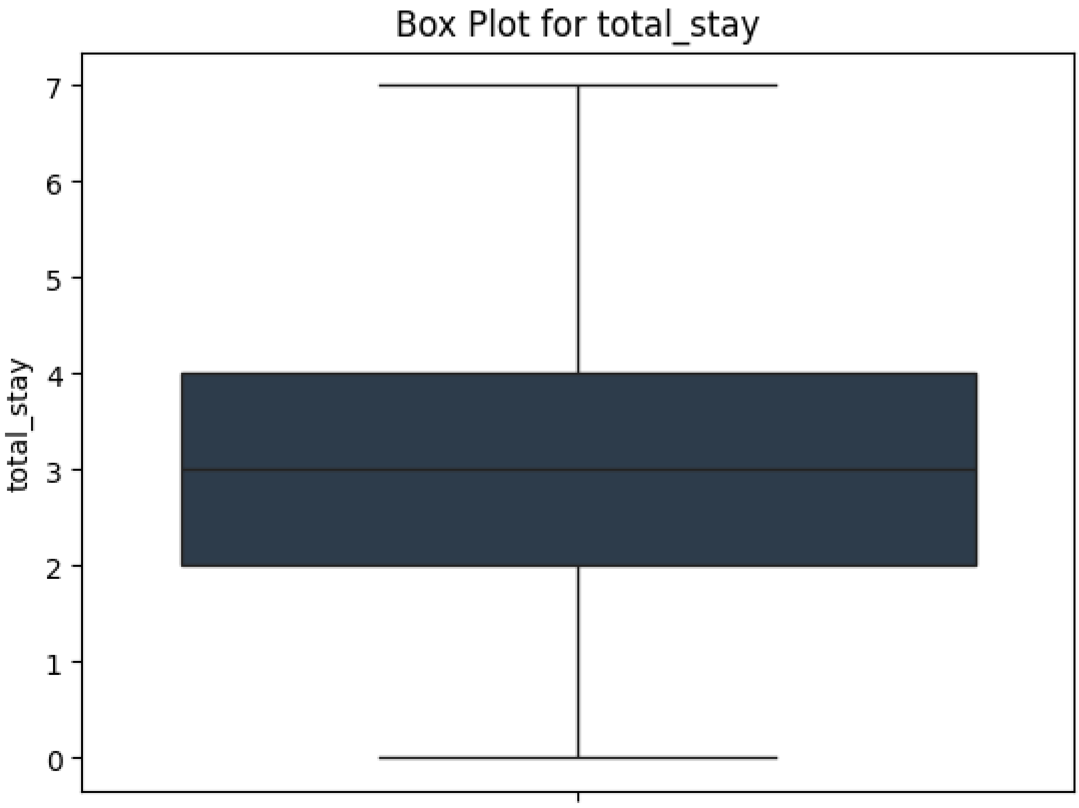
In [16]:

```
#Plotting boxplot to checking whether oulier removed or not
for i in ['adr', 'lead_time','total_stay']: #For Loop funxtion
    plt.title(f'Box Plot for {i}') #Title of the boxplot
    sns.boxplot(df[i],
                color='#2c3e50'
) #Visualizing boxplot
    plt.show()
```





Box Plot for lead_time



Box Plot for total_stay

In [17]:
```
df.describe() #Showing descriptive statistic overview
```

Out[17]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_da
count	107299.000000	107299.000000	107299.000000	107299.000000	
mean	0.363899	94.800511	2016.139218	26.912851	
min	0.000000	0.000000	2015.000000	1.000000	
25%	0.000000	16.000000	2016.000000	16.000000	
50%	0.000000	65.000000	2016.000000	27.000000	
75%	1.000000	151.000000	2017.000000	38.000000	
max	1.000000	380.000000	2017.000000	53.000000	
std	0.481122	93.312764	0.708196	13.835787	

In [18]:

```
df.describe(include='object') #Showing object data's info
```

Out[18]:

	hotel	arrival_date_month	meal	country	market_segment	distribution_channe
count	107299	107299	107299	107299	107299	10729
unique	2	12	5	175	7	5
top	City Hotel	May	BB	PRT	Online TA	TA/TC
freq	74379	10828	83788	43908	51388	8801

In [19]:

```
#Fuction for check unique values columns wise
for i in df.describe(include='object').columns: #For loop Fucntion
    print(i) #Column names
    print(df[i].unique()) #Unique values
    print('--------------------------------------')
```

```
hotel
['Resort Hotel' 'City Hotel']
--------------------------------------
arrival_date_month
['July' 'August' 'September' 'October' 'November' 'December' 'January'
 'February' 'March' 'April' 'May' 'June']
--------------------------------------
meal
['BB' 'FB' 'HB' 'SC' 'Undefined']
--------------------------------------
country
['PRT' 'GBR' 'USA' 'ESP' 'IRL' 'FRA' 'ROU' 'NOR' 'ARG' 'POL' 'DEU' 'BEL'
 'CHE' 'CN' 'GRC' 'NLD' 'RUS' 'SWE' 'AUS' 'EST' 'CZE' 'BRA' 'ITA' 'FIN'
 'DNK' 'MOZ' 'BWA' 'LUX' 'SVN' 'ALB' 'CHN' 'MEX' 'MAR' 'SMR' 'LVA' 'PRI'
 'SRB' 'IND' 'CHL' 'AUT' 'LTU' 'OMN' 'TUR' 'ZAF' 'AGO' 'ISR' 'CYM' 'ZMB'
 'CPV' 'ZWE' 'DZA' 'KOR' 'CRI' 'HUN' 'ARE' 'TUN' 'JAM' 'HRV' 'HKG' 'IRN'
 'AND' 'GIB' 'URY' 'BLR' 'JEY' 'CAF' 'CYP' 'COL' 'GGY' 'KWT' 'NGA' 'MDV'
 'VEN' 'FJI' 'SVK' 'LBN' 'PHL' 'SYC' 'BHR' 'NZL' 'KAZ' 'THA' 'DOM' 'MYS'
 'UKR' 'ARM' 'JPN' 'LKA' 'CUB' 'CMR' 'BIH' 'MUS' 'COM' 'SUR' 'UGA' 'BGR'
 'CIV' 'JOR' 'SYR' 'SGP' 'BDI' 'SAU' 'VNM' 'AZE' 'PLW' 'QAT' 'EGY' 'MLT'
 'MWI' 'ECU' 'MDG' 'IDN' 'ISL' 'UZB' 'NPL' 'BHS' 'PAK' 'MAC' 'TWN' 'STP'
 'SEN' 'PER' 'KNA' 'ETH' 'IRQ' 'HND' 'GEO' 'KHM' 'MCO' 'BGD' 'IMN' 'TJK'
 'NIC' 'BEN' 'VGB' 'TZA' 'GAB' 'MKD' 'TMP' 'GLP' 'LIE' 'GNB' 'KEN' 'MNE'
 'UMI' 'MYT' 'MMR' 'PAN' 'BFA' 'LBY' 'MLI' 'NAM' 'BOL' 'PRY' 'BRB' 'ABW'
 'SLV' 'DMA' 'PYF' 'GUY' 'LCA' 'ATA' 'RWA' 'GTM' 'GHA' 'ASM' 'TGO' 'MRT'
 'NCL' 'KIR' 'SDN' 'ATF' 'SLE' 'LAO' 'FRO']
--------------------------------------
market_segment
['Direct' 'Corporate' 'Online TA' 'Offline TA/TO' 'Complementary' 'Groups'
 'Aviation']
--------------------------------------
distribution_channel
['Direct' 'Corporate' 'TA/TO' 'Undefined' 'GDS']
--------------------------------------
reserved_room_type
['C' 'A' 'D' 'E' 'G' 'F' 'H' 'L' 'B' 'P']
--------------------------------------
assigned_room_type
['C' 'A' 'D' 'E' 'G' 'F' 'I' 'B' 'H' 'L' 'K' 'P']
--------------------------------------
deposit_type
['No Deposit' 'Refundable' 'Non Refund']
--------------------------------------
customer_type
['Transient' 'Contract' 'Transient-Party' 'Group']
--------------------------------------
reservation_status
['Check-Out' 'Canceled' 'No-Show']
--------------------------------------
Month_Name
['January' 'February' 'March' 'June' 'April' 'May' 'July' 'August'
 'November' 'September' 'December' 'October']
--------------------------------------
```

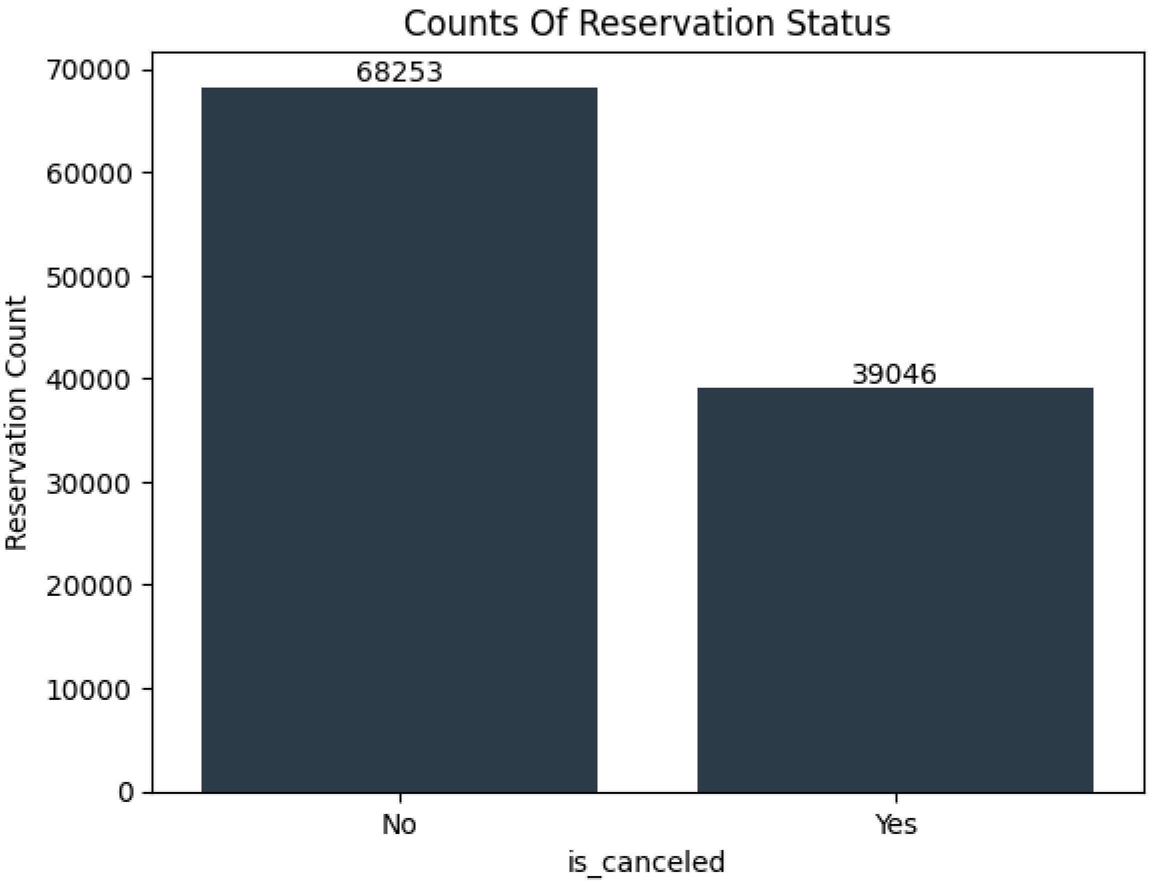
In [20]:

```
df['is_canceled'] = df['is_canceled'].replace([0,1],['No','Yes']) #Replacing values
```

Exploratory Data Analysis

In [21]:

```
#Visualize count plot for reservation rate
plt.title('Counts Of Reservation Status') #Title of the graph
ax = sns.countplot(data = df,
                   x = 'is_canceled',
                   color='#2c3e50'
)
ax.bar_label(ax.containers[0])
plt.ylabel('Reservation Count') #Change Y labels
plt.show()
```



-Non-canceled Reservations (No): There are 68,253 reservations that were not canceled.

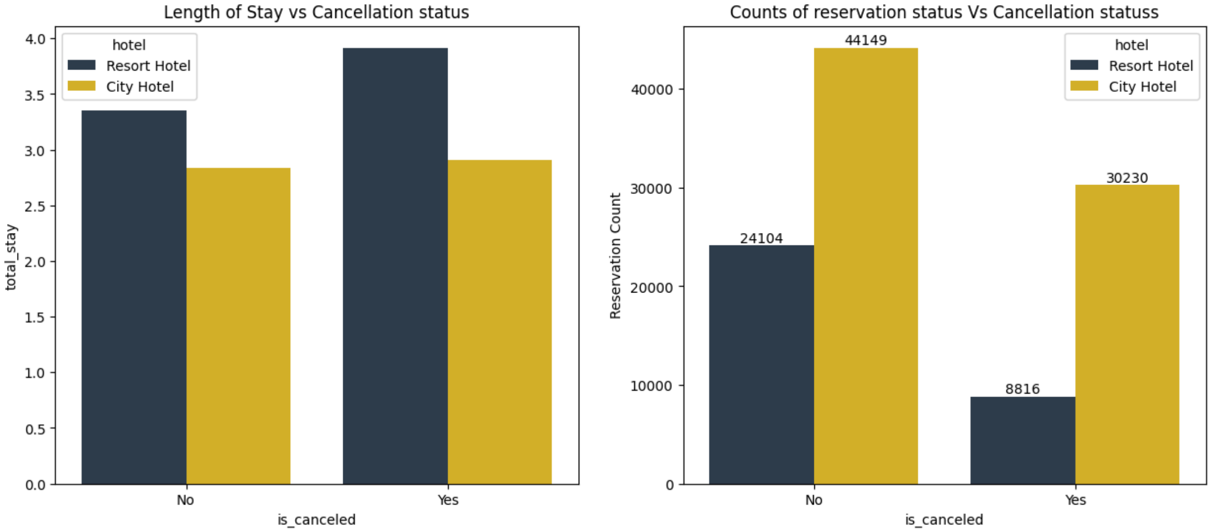
-Canceled Reservations (Yes): There are 39,046 reservations that were canceled.

In [22]:

```
# Visualize length of stay vs cancellation
plt.figure(figsize=(15,6))
plt.subplot(1,2,1)
plt.title('Length of Stay vs Cancellation status')
sns.barplot(data=df,
            x='is_canceled',
            y='total_stay',
            hue = 'hotel' ,
            ci = None,
            palette=['#2c3e50', '#f1c40f']
)



# Visualize Count of reservation status vs cancellation
plt.subplot(1,2,2)
plt.title('Counts of reservation status Vs Cancellation statuss') #Title of the gra
ax = sns.countplot(data = df,
                   x='is_canceled',
                   hue= 'hotel',
                   palette=['#2c3e50', '#f1c40f']
)
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.ylabel('Reservation Count') #Changes Y labels
plt.show()
```



-For Resort Hotel the average length of stay is higher for non-canceled reservations compared to canceled ones.

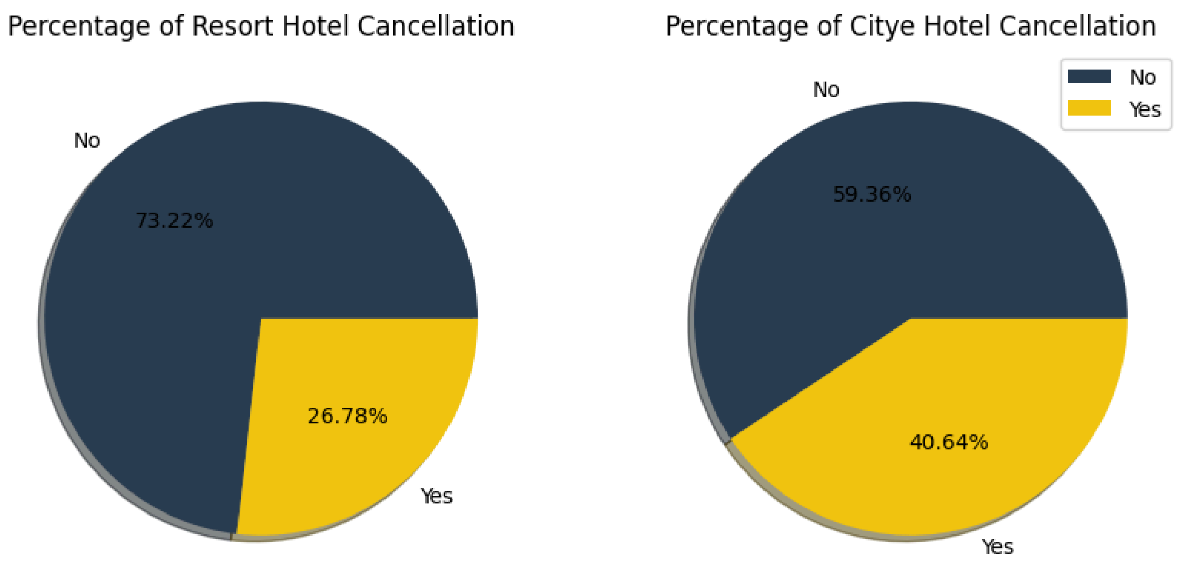
-For City Hotel the average length of stay is slightly higher for non-canceled reservations, but the difference is less pronounced compared to the Resort Hotel.

-For Resort Hotel the number of non-canceled reservations is significantly higher than canceled ones.

-For City Hotel the number of non-canceled reservations is also higher than canceled ones, but the difference is less pronounced compared to the Resort Hotel.

In [23]:

```
#Visualize percenge cancellation of city hotels and resort hotels
plt.figure(figsize = (10,10))
plt.subplot(1,2,1)
plt.title('Percentage of Resort Hotel Cancellation')
plt.pie(df[df['hotel']=='Resort Hotel']['is_canceled'].value_counts().values,
        labels=df[df['hotel']=='Resort Hotel']['is_canceled'].value_counts().index,
        shadow=True, autopct='%.2f%%',
       colors = ['#2c3e50', '#f1c40f']
)
plt.legend
plt.subplot(1,2,2)
plt.title('Percentage of Citye Hotel Cancellation')
plt.pie(df[df['hotel']=='City Hotel']['is_canceled'].value_counts().values,
        labels=df[df['hotel']=='City Hotel']['is_canceled'].value_counts().index,
        shadow=True, autopct='%.2f%%',
       colors=['#2c3e50', '#f1c40f']
)
plt.legend()
plt.show()
```

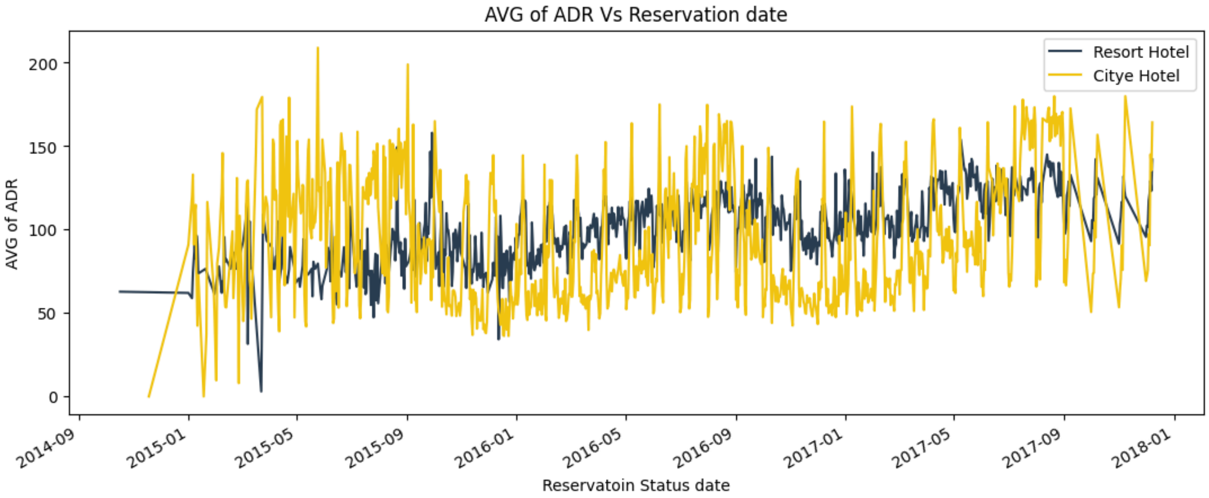


In [24]:

```
resortHotel_price = df[df['hotel']=='City Hotel'].groupby('reservation_status_date'
#grouping date and calculating adr mean value for each date
cityHotel_price = df[df['hotel']=='Resort Hotel'].groupby('reservation_status_date'
#grouping date and calculating adr mean value for each date
```

In [25]:

```
#Visualize Mean of ADR Vs Reservation date
plt.figure(figsize=(13,5))
plt.title('AVG of ADR Vs Reservation date') #Title of the graph
resortHotel_price['adr'].plot(label = 'Resort Hotel',
                              color = '#2c3e50'
) #creates line graph
cityHotel_price['adr'].plot(label = 'Citye Hotel',
                            color = '#f1c40f'
) #creates line graph
plt.ylabel('AVG of ADR') #Changes Y labels
plt.xlabel('Reservatoin Status date') #Changes X labels
plt.legend()
plt.show()
```

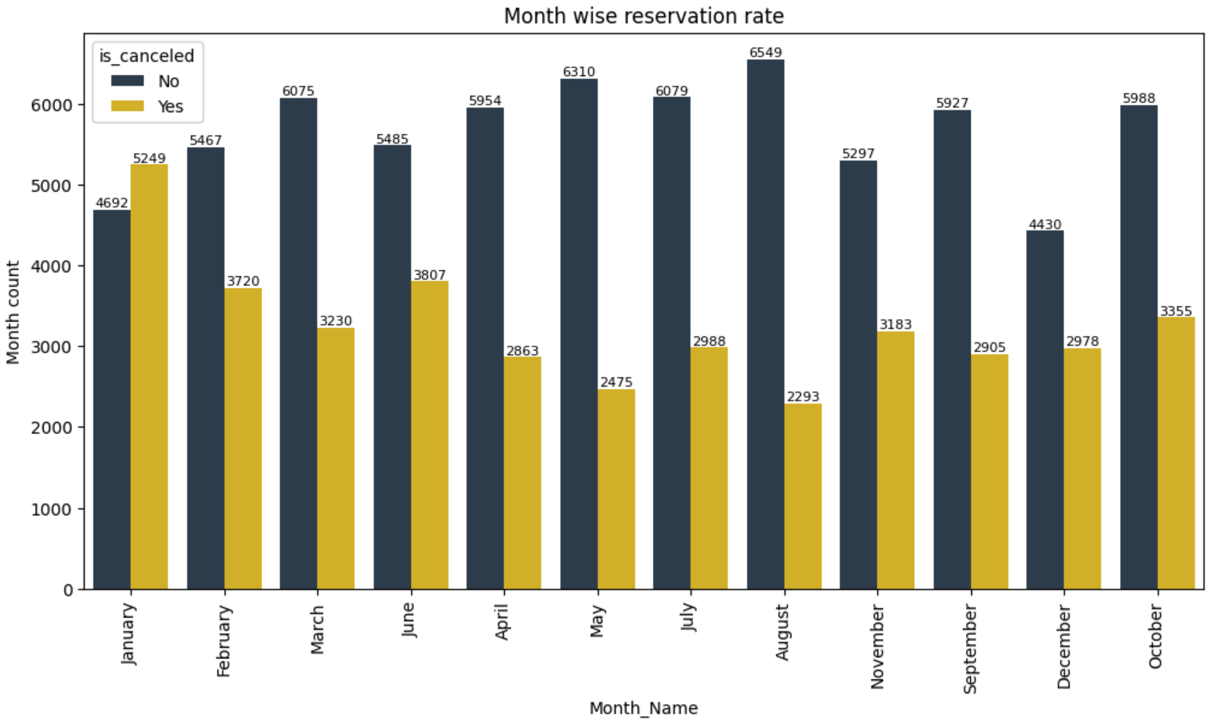


-City Hotels (orange line) show significantly more price volatility compared to Resort Hotels (blue line)

-City Hotels generally command higher peak prices than Resort Hotels

-Resort Hotels demonstrate more consistent pricing strategy

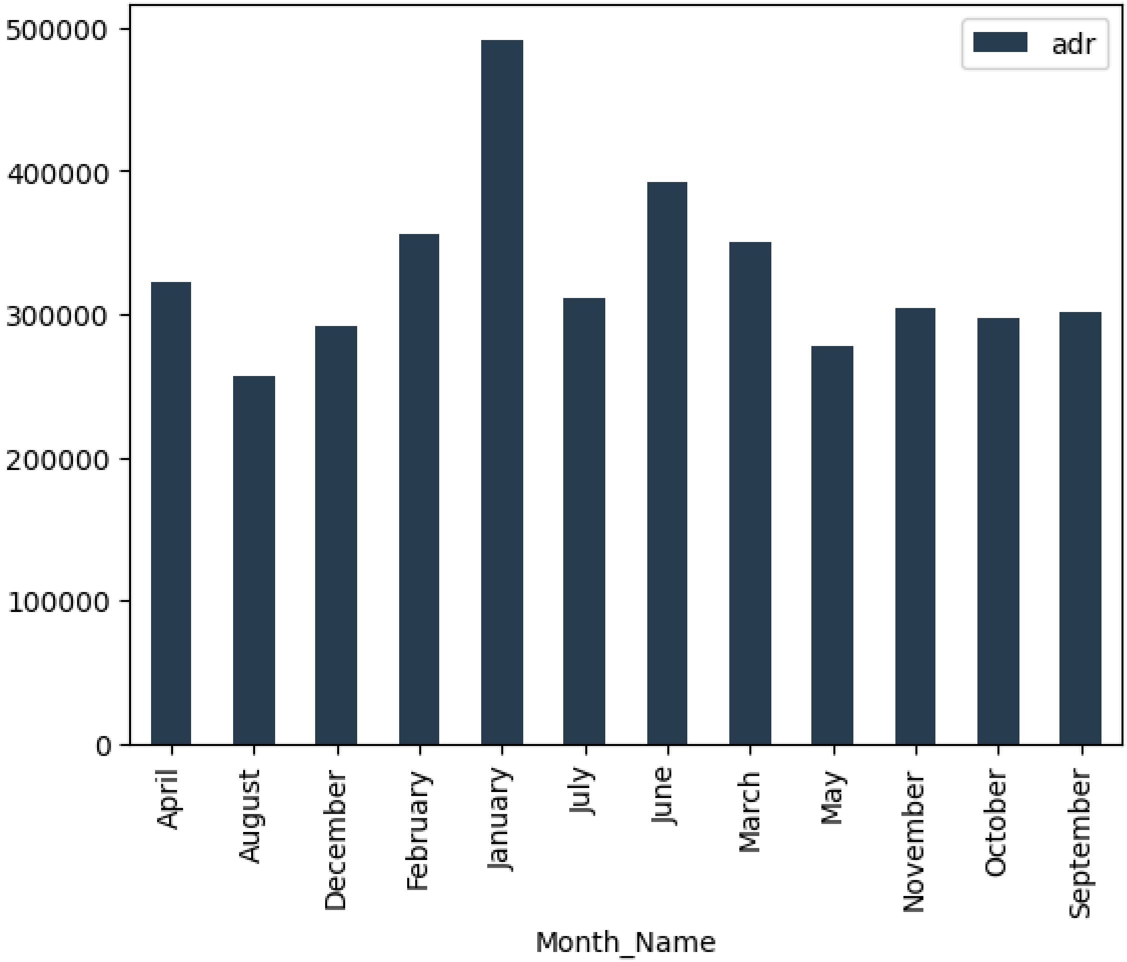
```
In [26]: #Visualize barplot to check mounth wise reservation status
         plt.figure(figsize=(12,6))
         plt.title('Month wise reservation rate') #Title of the graph
         ax = sns.countplot(data = df, x = 'Month_Name', hue = 'is_canceled', palette=['#2c3
         ax.bar_label(ax.containers[0], fontsize=8)
         ax.bar_label(ax.containers[1], fontsize=8)
         plt.xticks(rotation = 90)
         plt.ylabel('Month count') #Changes y labels
         plt.show()
```



-August is the month where reservation and cancellation

```
In [27]: # Visualize sum of Adr Vs Month where the reservation is cancelled,
         cancel_adr = df[df['is_canceled']=='Yes'].groupby('Month_Name')[['adr']].sum()
         cancel_adr.plot(kind = 'bar',
                         color = '#2c3e50')
```

```
Out[27]: <Axes: xlabel='Month_Name'>
```



-January is the month where the ADR value is the higher which is the main cause of hotel cancellation

```
In [28]: df.sample() #Showing one random row
```

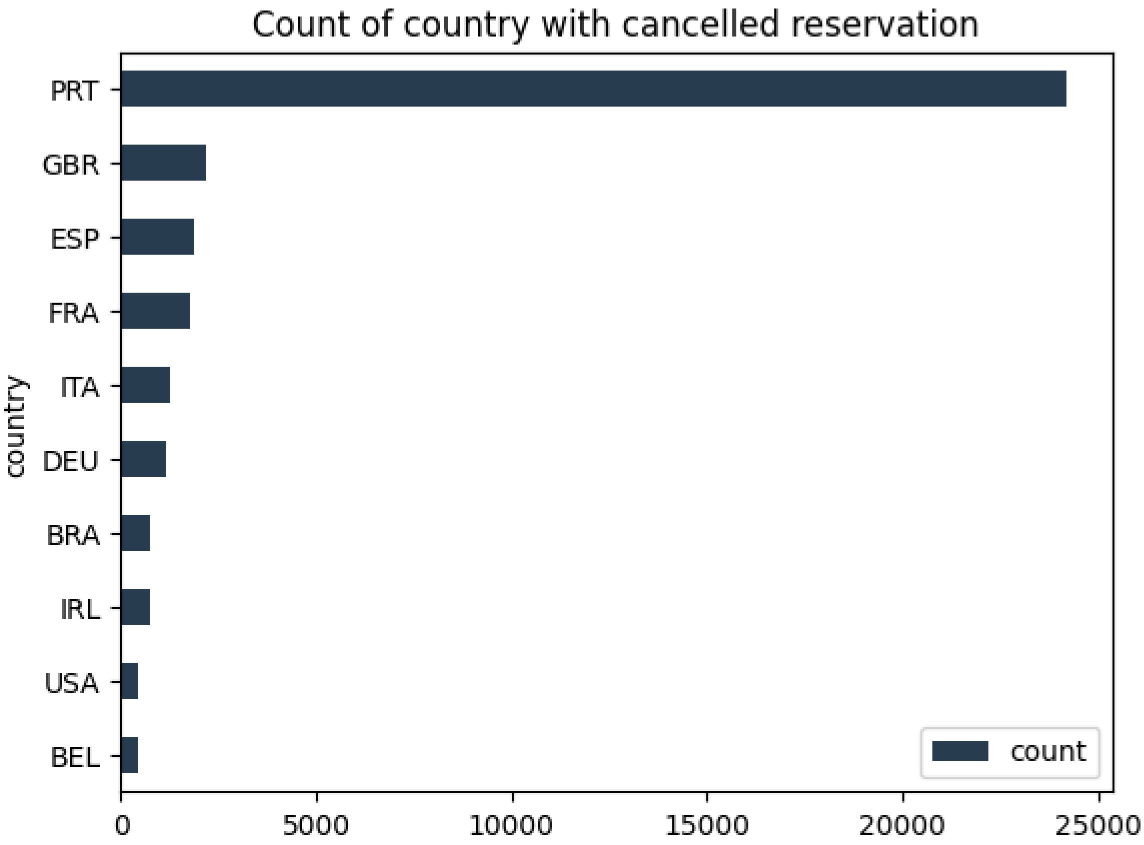
Out[28]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_wee
80744	City Hotel	Yes	25	2015	November	

1 rows × 32 columns

```
In [29]: #Visualize Country wise cancelled reservation
         countryWiseCancellation = df[df['is_canceled']=='Yes'] #Selecing data set where can
```

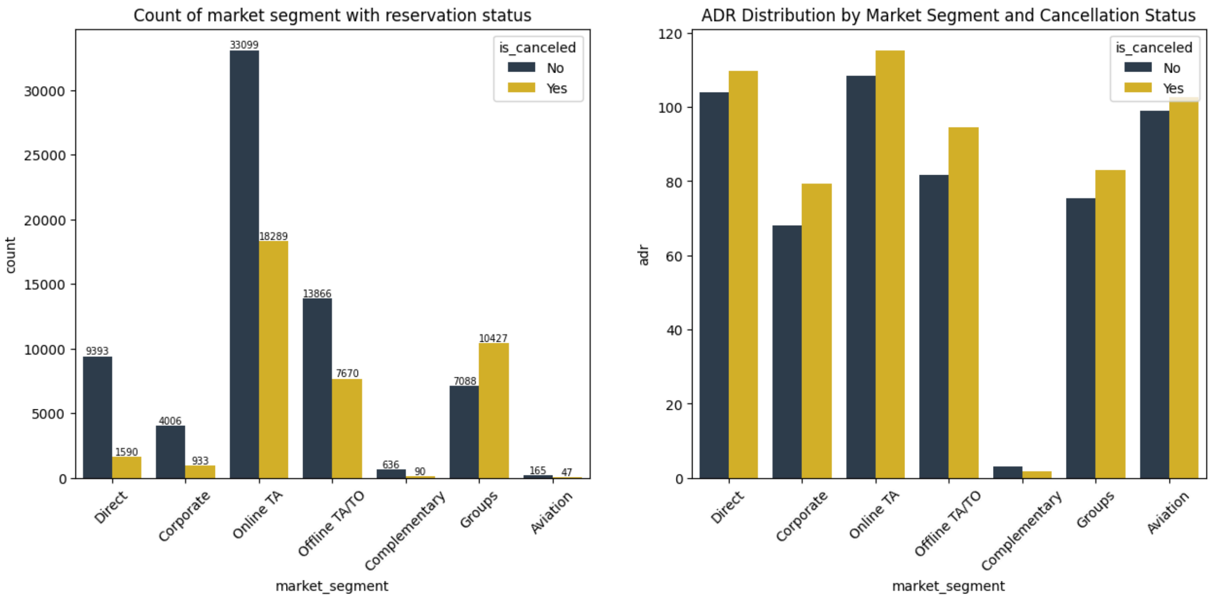
```
countryWiseCancellation['country'].value_counts().head(10).sort_values().plot(kind
                                                                                color
#Plotting top 10 country reservation cancelled
plt.title('Count of country with cancelled reservation') #Gives title of the graph
plt.legend(['count'])
plt.show()
```



-Country wise portugal is the country where reservation cancellation is highest

```
In [30]: plt.figure(figsize=(15,6))
#Visualize count of market segment with reservation rate
plt.subplot(1,2,1)
plt.title('Count of market segment with reservation status') #Shows title of the gr
ax = sns.countplot(data = df,
                   x ='market_segment',
                   hue = 'is_canceled',
                   palette=['#2c3e50', '#f1c40f']
)
ax.bar_label(ax.containers[0],fontsize=7)
ax.bar_label(ax.containers[1],fontsize=7)
plt.xticks(rotation = 45)


#Visualize ADR by market segment
plt.subplot(1,2,2)
sns.barplot(data=df,
            x='market_segment',
            y='adr', hue='is_canceled',
            ci = None, palette=['#2c3e50', '#f1c40f']
)
plt.title('ADR Distribution by Market Segment and Cancellation Status') #Shows titl
plt.xticks(rotation=45)
plt.show()
```



-Online TA (Online Travel Agents) has the highest number of reservations overall, with a high proportion of both canceled and non-canceled bookings (33,099 non-canceled and 18,289 canceled).

-Online TA segments has highest ADRs, with canceled bookings showing slightly higher ADRs than non-canceled ones.

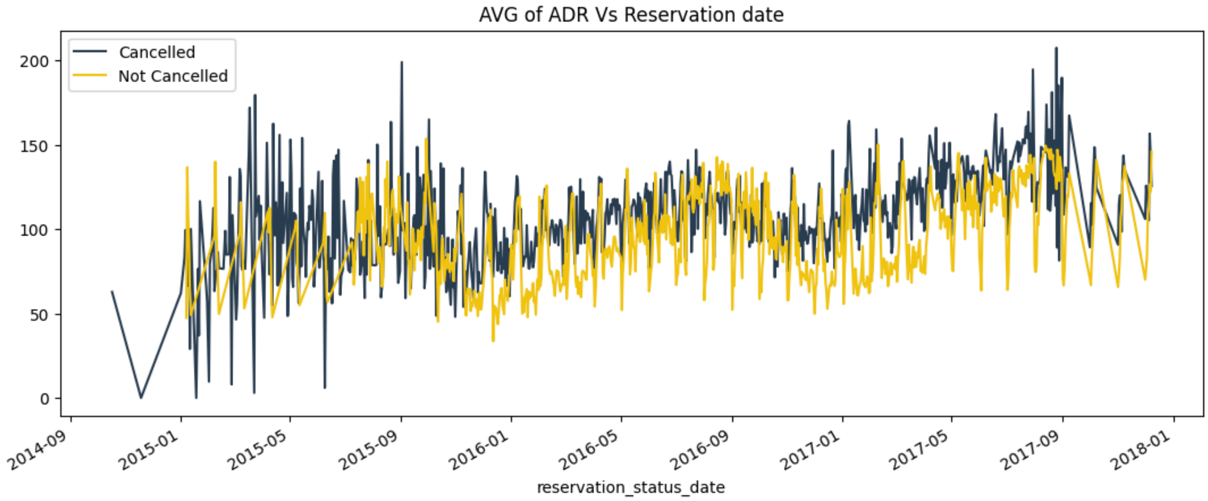
-Direct and Aviation segment have similar ADRs, and canceled bookings tend to have a higher ADR.

In [31]:

```
market_segment_canc_yes = df[df['is_canceled'] == 'Yes'].groupby('reservation_statu
#grouping reservation date  where reservation cancelled and calculating mean of ADR
market_segment_canc_no = df[df['is_canceled'] == 'No'].groupby('reservation_status_
#grouping reservation date  where reservation not cancelled and calculating mean of
```

In [32]:

```
#Visualize AVG of ADR Vs Reservation date
plt.figure(figsize=(13,5))
plt.title('AVG of ADR Vs Reservation date')
market_segment_canc_yes['adr'].plot(label = 'Cancelled', color = '#2c3e50')
market_segment_canc_no['adr'].plot(label = 'Not Cancelled', color = '#f1c40f')
plt.legend()
plt.show()
```



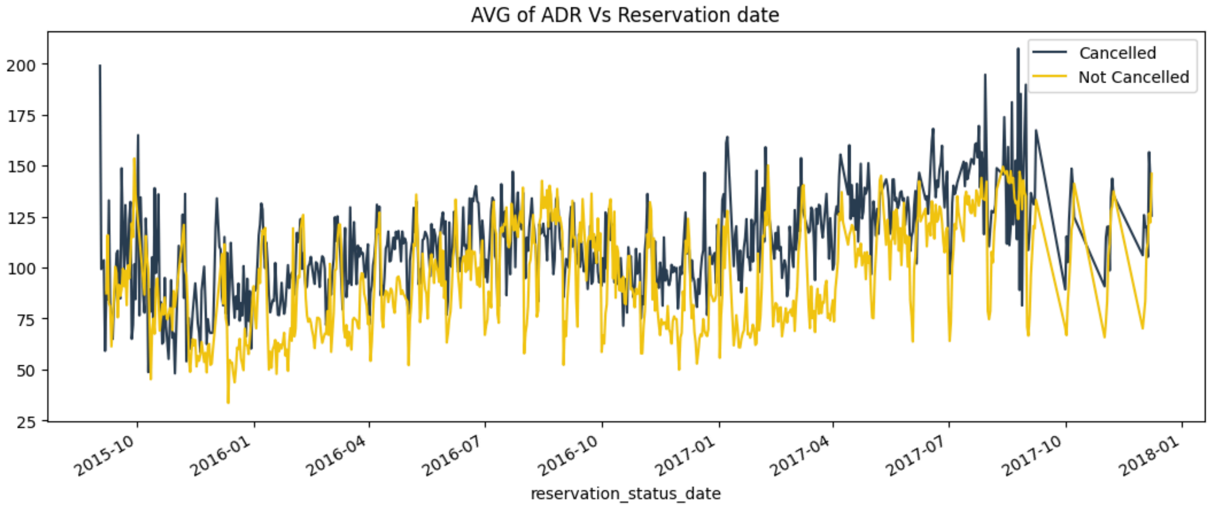
from 2015/9 can be observed that as the mean of ADR is higher the cancellation rate is also higher and vice versa

In [33]:
```
mark = df[(df['reservation_status_date']>'2015-09') & (df['reservation_status_date'
```

In [34]:
```
market_segment_canc_yes1 = mark[mark['is_canceled'] == 'Yes'].groupby('reservation_
#grouping reservation date  where reservation cancelled and calculating mean of ADR
market_segment_canc_no1 = mark[mark['is_canceled'] == 'No'].groupby('reservation_st
#grouping reservation date  where reservation not cancelled and calculating mean of
```

In [35]:
```
#Avg ADR Vs reservation date with reservation status
plt.figure(figsize=(13,5))
plt.title('AVG of ADR Vs Reservation date')
market_segment_canc_yes1['adr'].plot(label = 'Cancelled',
                                     color = '#2c3e50'
)


market_segment_canc_no1['adr'].plot(label = 'Not Cancelled',
                                    color = '#f1c40f'
)
plt.legend()
plt.show()
```



-Both cancelled and non-cancelled bookings show significant price volatility over time

-The ADR generally ranges between $50-150$, with some peaks reaching around $200

-Cancelled bookings (blue line) often show higher ADR values than non-cancelled bookings (orange line)

-There are noticeable seasonal patterns, with higher rates appearing in peak travel periods

In [36]:

```
df.head()
```

Out[36]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_nu
0	Resort Hotel	No	342	2015	July	
2	Resort Hotel	No	7	2015	July	
3	Resort Hotel	No	13	2015	July	
4	Resort Hotel	No	14	2015	July	
5	Resort Hotel	No	14	2015	July	

5 rows × 32 columns

In [37]:

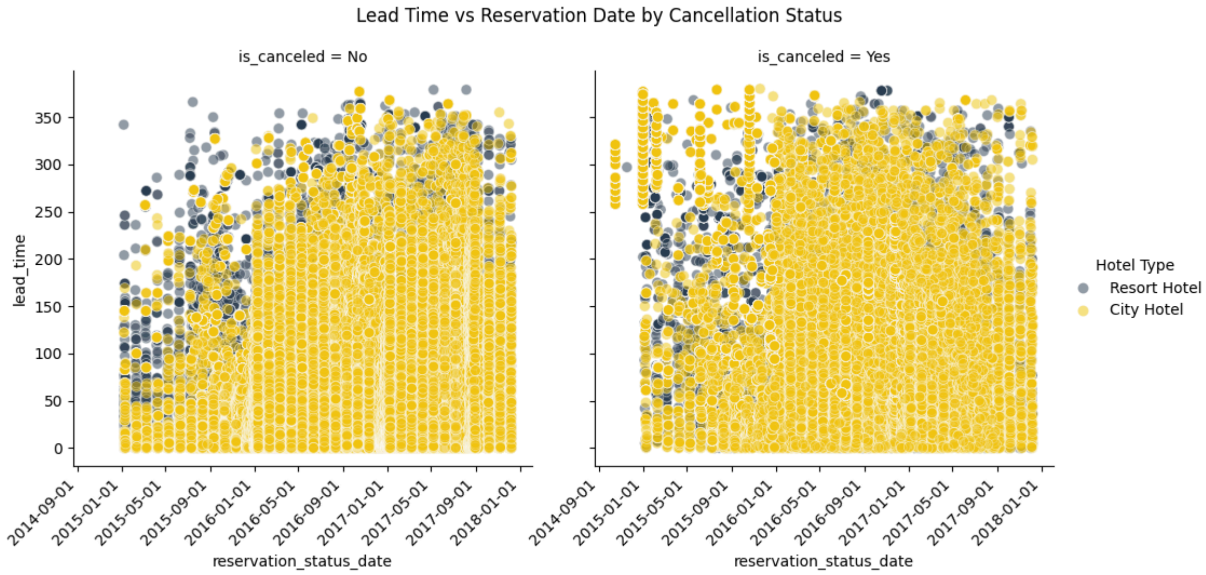
```
# Create the relplot with custom palette and other styling parameters
g = sns.relplot(data=df,
                x='reservation_status_date',
                y='lead_time',
                col='is_canceled',
                hue='hotel',
                kind='scatter',
                palette=['#2c3e50', '#f1c40f'],  # Set distinct colors for better v
                alpha=0.5,  # Add transparency to see overlapping points
                s=50)      # Adjust point size
```

```
# Rotate x-axis labels for all subplots
g.set_xticklabels(rotation=45, ha='right')

# Adjust the layout
g.fig.subplots_adjust(bottom=0.2)

# Optional: Customize the plot further
g.fig.suptitle('Lead Time vs Reservation Date by Cancellation Status', y=1.05)
g._legend.set_title('Hotel Type')

plt.show()
```



-The right panel (is_canceled = Yes) shows more bookings overall, suggesting a high cancellation rate

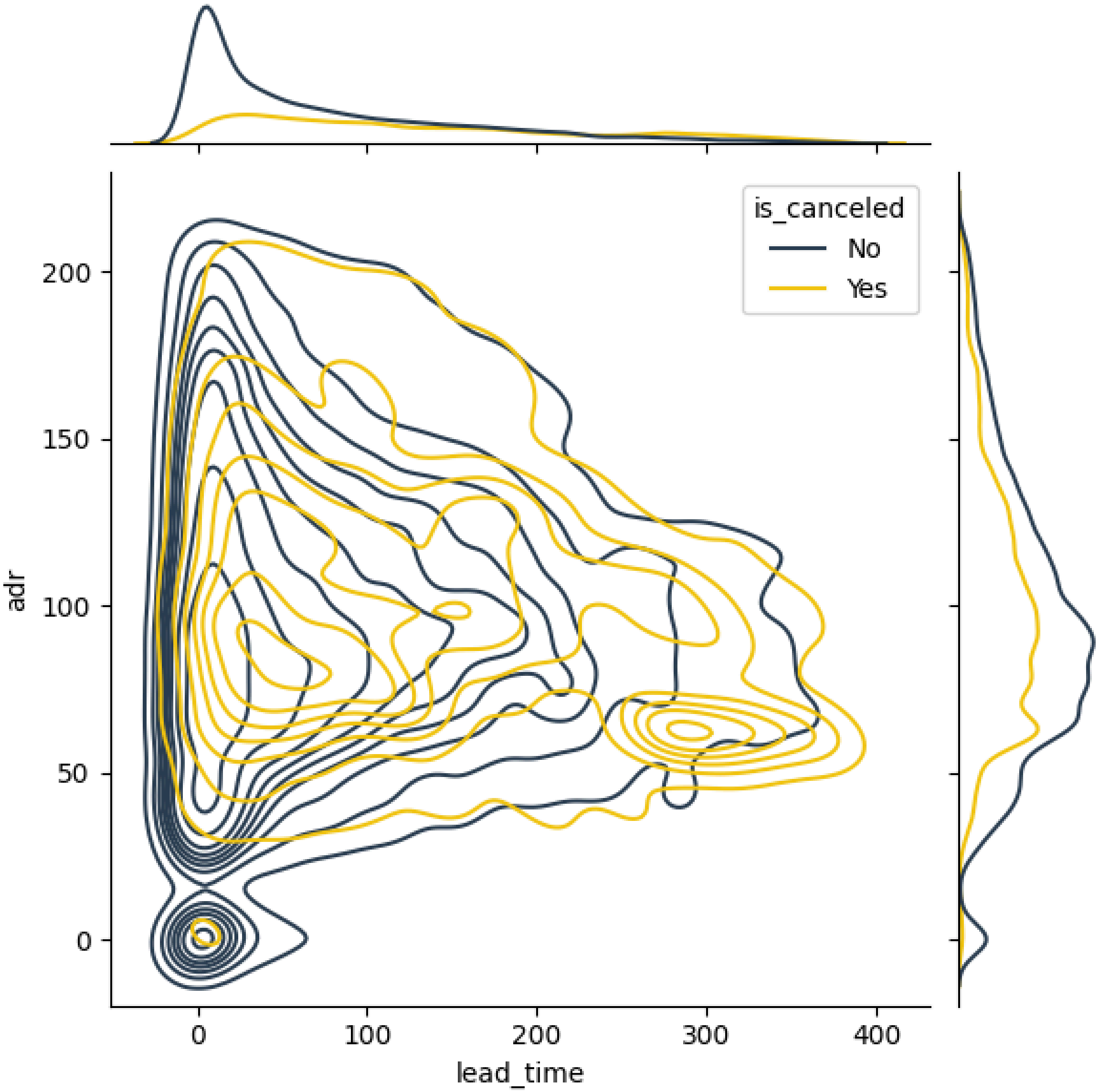
-City Hotels (orange) have significantly more cancellations than Resort Hotels (blue)

-Most bookings are made between 0-200 days in advance

-There's a noticeable increase in longer lead times (200+ days) from 2016 onwards

In [38]:
```
cityHotelleadtime = df[df['hotel']=='City Hotel']
resortHotelleadtime = df[df['hotel']=='Resort Hotel']
```

In [39]:
```
sns.jointplot(
    data=df,
    x='lead_time',
    y='adr',
    hue='is_canceled',
    kind='kde' ,
    palette=['#2c3e50', '#f1c40f']
)
plt.show()
```



-Most bookings are made with shorter lead times (0-100 days)

-Non-cancelled bookings (blue) show a higher peak at very short lead timesx x

-Cancelled bookings (orange) have a flatter distribution

-ADR mostly ranges from about 50 to 200.

-There's a peak in the middle range