Hotel booking Data Analysis - Jupyter Notebook

localhost:8888/notebooks/Hotel booking Data Analysis.ipynb

Business Problem

In recent years, City Hotel and Resort Hotel have seen high cancellation rates. Each hotel is now dealing with a number of issues as a result, including fewer revenues and less than ideal hotel room use. Consequently, lowering cancellation rates is both hotels' primary goal in order to increase their efficiency in generating revenue, and for us to offer thorough business advice to address this problem.

The analysis of hotel booking cancellations as well as other factors that have no bearing on their business and yearly revenue generation are the main topics of this report.

Assumptions

1. No unusual occurrences between 2015 and 2017 will have a substantial impact on the data used.
2. The information is still current and can be used to analyze a hotel's possible plans in an efficient manner.
3. There are no unanticipated negatives to the hotel employing any advised technique.
4. The hotels are not currently using any of the suggested solutions.
5. The biggest factor affecting the effectiveness of earning income is booking cancellations.
6. Cancellations result in vacant rooms for the booked length of time.
7. Clients make hotel reservations the same year they make cancellations.

Research Question

1. What are the variables that affect hotel reservation cancellations?
2. How can we make hotel reservations cancellations better?
3. How will hotels be assisted in making pricing and promotional decisions?

Hypothesis

1. More cancellations occur when prices are higher.
2. When there is a longer waiting list, customers tend to cancel more frequently.
3. The majority of clients are coming from offline travel agents to make their reservations.

About Dataset

Context

This dataset contains 119390 observations for a City Hotel and a Resort Hotel. Each observation represents a hotel booking between the 1st of July 2015 and 31st of August 2017, including booking that effectively arrived and booking that were canceled.

Content

Since this is hotel real data, all data elements pertaining hotel or costumer identification were deleted. Four Columns, 'name', 'email', 'phone number' and 'credit_card' have been artificially created and added to the dataset.

Acknowledgements

The data is originally from the article Hotel Booking Demand Datasets, written by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, Volume 22, February 2019.

++

Importing Libraries

In [113]:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')
```

Loading The Dataset

In [114]:

```
df = pd.read_csv('hotel_booking.csv')
```

Exploratory Data Analysis and Data Cleaning

In [115]:

```
# first 5 rows of the dataset

df.head()
```

Out[115]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_r
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 × 36 columns

In [116]:

```
# last 5 rows of the dataset

df.tail()
```

Out[116]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day
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

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_
119388	City Hotel	0	109	2017	August	35	31
119389	City Hotel	0	205	2017	August	35	29

5 rows × 36 columns

In [117]:

```
#shape of the dataset(total rows * total columns)

df.shape
```

Out[117]:

```
(119390, 36)
```

In [118]:

```
#remove personal data to generalize the data

#drop column (name, email, phone-number, credit_card)


df.drop(columns=['name', 'email','phone-number','credit_card'], inplace= True)
```

In [119]:

```
df.shape
```

Out[119]:

```
(119390, 32)
```

In [120]:

```
df.columns
```

Out[120]:

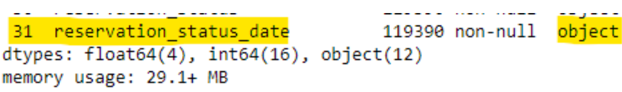
```
Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',
       'arrival_date_month', 'arrival_date_week_number',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'adults', 'children', 'babies', 'meal',
       'country', 'market_segment', 'distribution_channel',
       '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 [121]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   arrival_date_year               119390 non-null  int64
 4   arrival_date_month              119390 non-null  object
 5   arrival_date_week_number        119390 non-null  int64
 6   arrival_date_day_of_month       119390 non-null  int64
 7   stays_in_weekend_nights         119390 non-null  int64
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         118902 non-null  object
 14  market_segment                  119390 non-null  object
 15  distribution_channel            119390 non-null  object
 16  is_repeated_guest               119390 non-null  int64
 17  previous_cancellations          119390 non-null  int64
 18  previous_bookings_not_canceled  119390 non-null  int64
 19  reserved_room_type              119390 non-null  object
 20  assigned_room_type              119390 non-null  object
 21  booking_changes                 119390 non-null  int64
 22  deposit_type                    119390 non-null  object
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64
 26  customer_type                   119390 non-null  object
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  object
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB
```

Converting "reservation_status_date" column to date-time format



```
31  reservation_status_date         119390 non-null  object
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB
```

In [122]:

```
df['reservation_status_date'] = pd.to_datetime(df['reservation_status_date'])
```

In [123]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   arrival_date_year               119390 non-null  int64
 4   arrival_date_month              119390 non-null  object
 5   arrival_date_week_number        119390 non-null  int64
 6   arrival_date_day_of_month       119390 non-null  int64
 7   stays_in_weekend_nights         119390 non-null  int64
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         118902 non-null  object
 14  market_segment                  119390 non-null  object
 15  distribution_channel            119390 non-null  object
 16  is_repeated_guest               119390 non-null  int64
 17  previous_cancellations          119390 non-null  int64
 18  previous_bookings_not_canceled  119390 non-null  int64
 19  reserved_room_type              119390 non-null  object
 20  assigned_room_type              119390 non-null  object
 21  booking_changes                 119390 non-null  int64
 22  deposit_type                    119390 non-null  object
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64
 26  customer_type                   119390 non-null  object
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(4), int64(16), object(11)
memory usage: 29.1+ MB
```

checkout catagorical columns(Dtype = object)

In [124]:

```
df.describe(include= 'object')
```

Out[124]:

	hotel	arrival_date_month	meal	country	market_segment	distribution_channel	reserved_room_type	assig
count	119390	119390	119390	118902	119390	119390	119390	11939
unique	2	12	5	177	8	5	10	12
top	City Hotel	August	BB	PRT	Online TA	TA/TO	A	A
freq	79330	13877	92310	48590	56477	97870	85994	74053

checkout unique values for each catagorical column

	hotel	arrival_date_month	meal	country	market_segment	distribution_channel	reserved_room_type	assigned_room_type	deposit_type	customer_t
count	119390	119390	119390	118902	119390	119390	119390	119390	119390	119
unique	2	12	5	177	8	5	10	12	3	

In [125]:

```
for col in df.describe(include= 'object').columns:

    print(col)

    print(df[col].unique())

    print('-'*50)
```

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

check missing values

In [126]:

```
df.isnull().sum()[df.isnull().sum() > 0]
```

Out[126]:

```
children         4
country        488
agent        16340
company     112593
dtype: int64
```

['children', 'country'] = drop rows as missing record is very less we can drop the rows of these

['agent','company'] = drop columns as missing record is very high we can drop these columns(we have no use)

In [127]:

```
df.drop(['agent','company'], axis = 1, inplace= True) # axis = 1 (columns), inplace = True (changes on the same dataset)

df.dropna(inplace= True) # removes entire record which has null value
```

In [128]:

```
df.isnull().sum() # after all missing value records removed
```

Out[128]:

```
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
dtype: int64
```

In [129]:

```
df.describe()
```

Out[129]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_
count	118898.000000	118898.000000	118898.000000	118898.000000	118898.000000	118898.00
mean	0.371352	104.311435	2016.157656	27.166555	15.800880	0.928897
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	161.000000	2017.000000	38.000000	23.000000	2.000000
max	1.000000	737.000000	2017.000000	53.000000	31.000000	16.000000
std	0.483168	106.903309	0.707459	13.589971	8.780324	0.996216

finding outliers and removing it

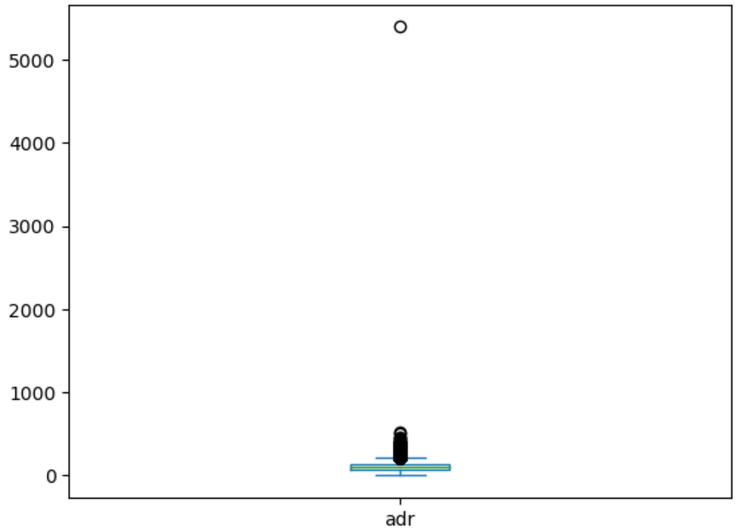
adr
118898.000000
102.003243
-6.380000
70.000000
95.000000
126.000000
5400.000000
50.485862

In [130]:

```
df['adr'].plot(kind = 'box')
```

Out[130]:

```
<Axes: >
```



In [131]:

```
df = df[df['adr'] < 5000] # removing records from dataset 'df' which has 'adr' >= 5000
```

In [132]:

```
df.describe()
```

Out[132]:

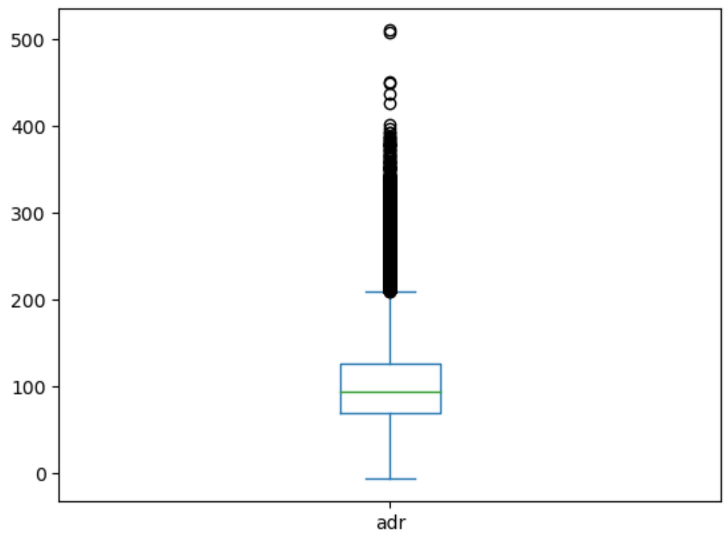
	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_\
count	118897.000000	118897.000000	118897.000000	118897.000000	118897.000000	118897.00
mean	0.371347	104.312018	2016.157657	27.166674	15.800802	0.928905
min	0.000000	0.000000	2015.000000	1.000000	1.000000	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	161.000000	2017.000000	38.000000	23.000000	2.000000
max	1.000000	737.000000	2017.000000	53.000000	31.000000	16.000000
std	0.483167	106.903570	0.707462	13.589966	8.780321	0.996217

In [133]:

```
df['adr'].plot(kind = 'box')
```

Out[133]:

```
<Axes: >
```



Here we have removed 1 outlier from our dataset and by this method we can remove any other outliers which can effect out data analysis

Data Analysis and Visualizations

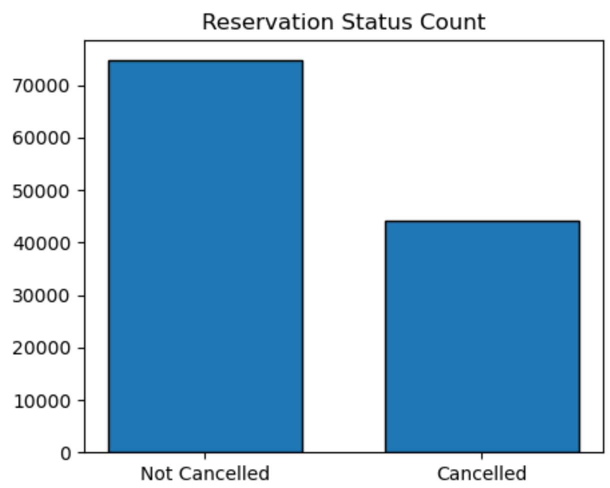
keeping in mind of our problem statement we need to analyse our data and do the visualization accordingly.

1. First thing to check how many reservation got cancelled and how many are not cancelled.

In [134]:

```
cancelled_percentage = df['is_canceled'].value_counts(normalize= True) #value_counts - returns count catagory wise

print(cancelled_percentage)    #normalize= True - returns in percentage



#creating bar plot to visualize the data



plt.figure(figsize=(5,4))

plt.title('Reservation Status Count')

plt.bar(['Not Cancelled', 'Cancelled'], df['is_canceled'].value_counts(), edgecolor = 'k', width = 0.7)

plt.show()
```

```
is_canceled
0    0.628653
1    0.371347
Name: proportion, dtype: float64
```



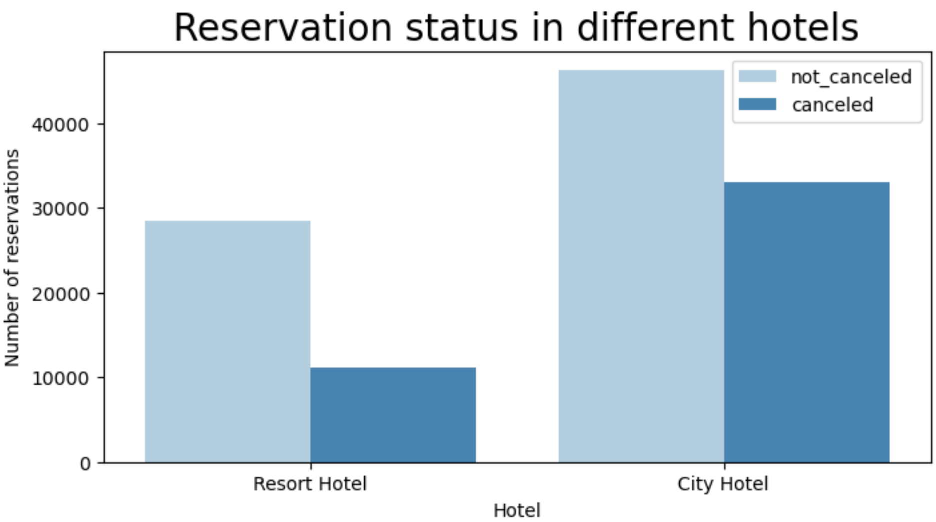
Insights: Here, we could see cancelled percentage is 37 % which is very high.

2. Depending on the Hotels checking whose cancelletion rate is higher

The accompanying bar graph shows the percentage of reservations that are canceled and those that are not. It is obvious that there are still a significant number of reservations that have not been canceled. There are still 37% of clients who canceled their reservation, which has a significant impact on the hotels' earnings.

In [135]:

```
plt.figure(figsize=(8, 4))

ax1 = sns.countplot(x='hotel', hue='is_canceled', data=df, palette='Blues')

legend_labels, _ = ax1.get_legend_handles_labels()

ax1.legend(['not_canceled', 'canceled'])

plt.title('Reservation status in different hotels', size=20)

plt.xlabel('Hotel')

plt.ylabel('Number of reservations')

plt.show()
```



In comparison to resort hotels, city hotels have more bookings. It's possible that resort hotels are more expensive than those in cities.

In [136]:

```
resort_hotel = df[df['hotel'] == 'Resort Hotel']

resort_hotel['is_canceled'].value_counts(normalize= True)
```

Out[136]:

```
is_canceled
0    0.72025
1    0.27975
Name: proportion, dtype: float64
```

In [137]:

```
city_hotel = df[df['hotel'] == 'City Hotel']

city_hotel['is_canceled'].value_counts(normalize= True)
```

Out[137]:

```
is_canceled
0    0.582918
1    0.417082
Name: proportion, dtype: float64
```

Here, could see for resort hotel calcelletion % is around 28 % whereas, for city hotel it's much higher than resort hotel which is around 42 %

Let's check if price is the factor for the higher cancelletion

---

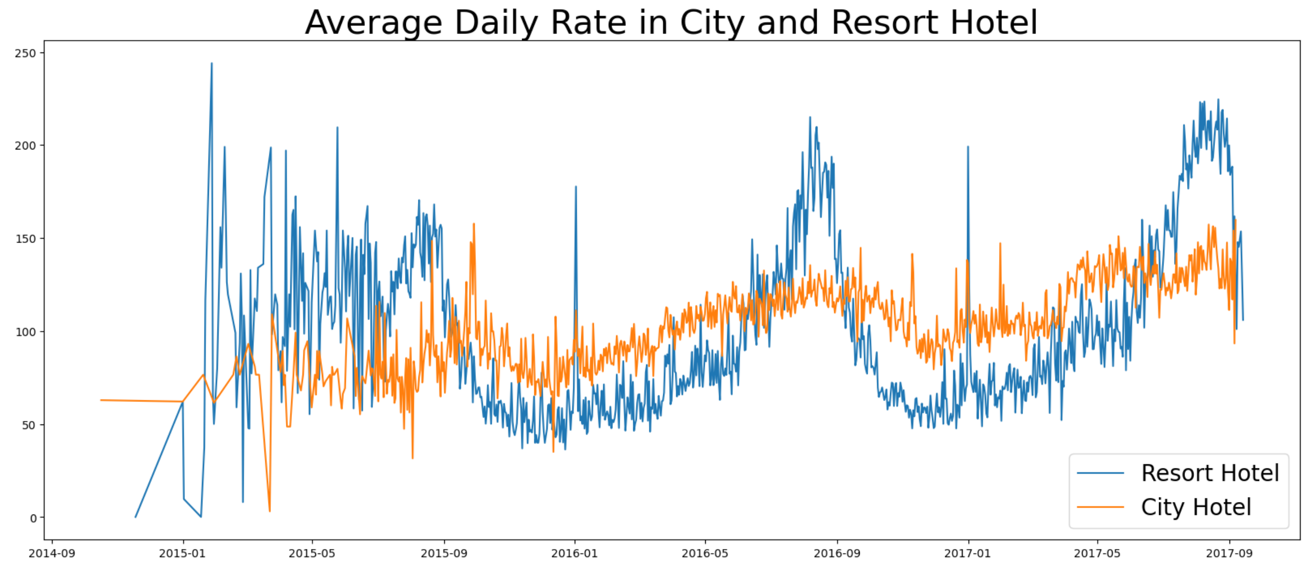
In [138]:

```
resort_hotel = resort_hotel.groupby('reservation_status_date')[['adr']].mean()
```

```
city_hotel = city_hotel.groupby('reservation_status_date')[['adr']].mean()
```

In [139]:

```
plt.figure(figsize= (20,8))

plt.title('Average Daily Rate in City and Resort Hotel', fontsize = 30)

plt.plot(resort_hotel.index, resort_hotel['adr'], label = 'Resort Hotel')

plt.plot(city_hotel.index, city_hotel['adr'], label = 'City Hotel')

plt.legend(fontsize = 20)

plt.show()
```



Here, We could see Resort Hotel's Average daily rate is much higher than the City Hotel.

We also can observe there is some sudden spike in the ADR value. That could be possible due to higher rate on the weekends.

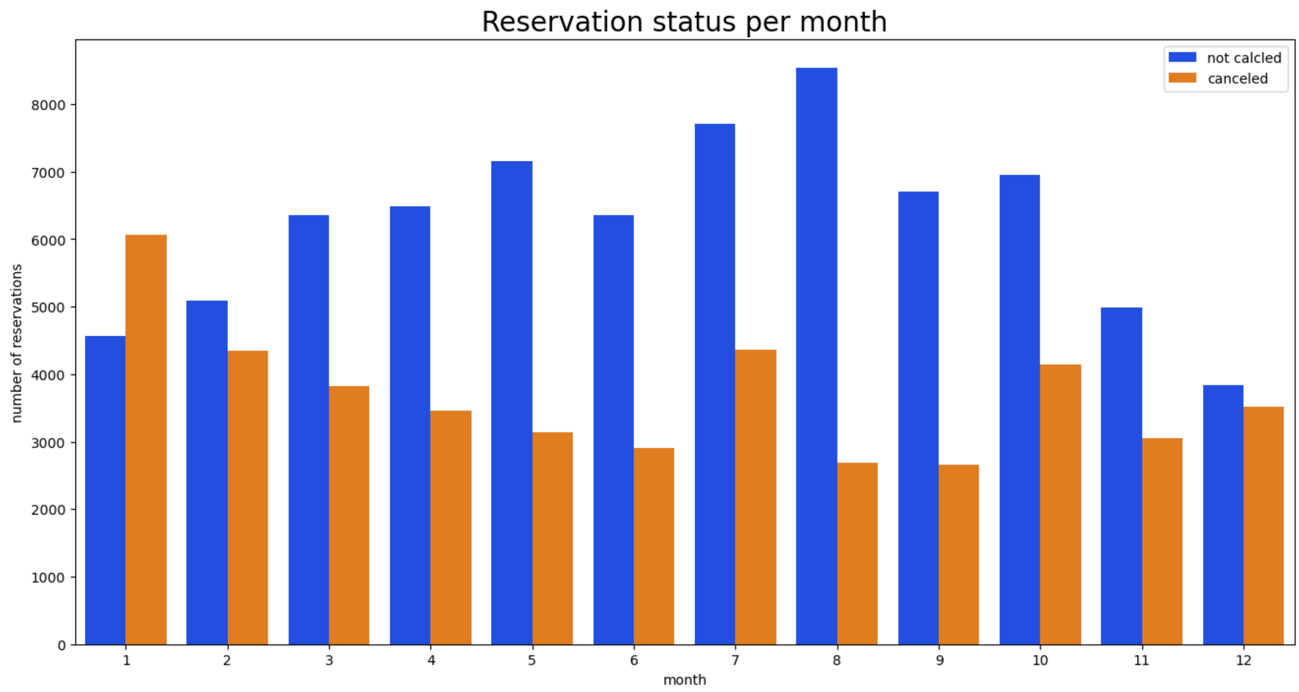
So, We can conclude that City Hotel's price is lower than the Resort Hotel's price

The line graph above shows that, on certain days, the average daily rate for a city hotel is less than that of a resort hotel, and on other days, it is even less. It goes without saying that weekends and holidays may see a rise in resort hotel rates.

Now, Let's find out on which month reservation and cancelletion is higher

In [140]:

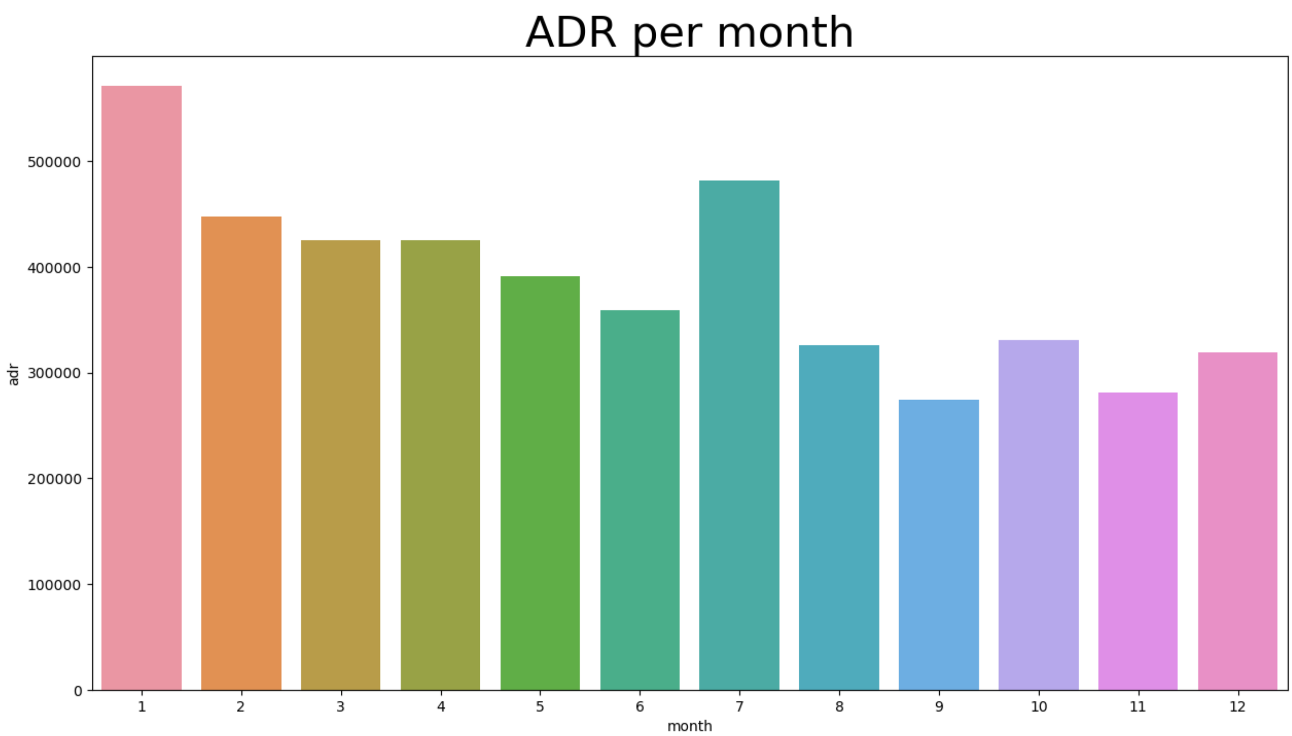
```
df['month'] = df['reservation_status_date'].dt.month

plt.figure(figsize = (16,8))

ax1 = sns.countplot(x = 'month', hue = 'is_canceled', data = df, palette = 'bright')

legend_labels,_ = ax1. get_legend_handles_labels()

ax1.legend(bbox_to_anchor=(1,1))

plt.title('Reservation status per month', size = 20)

plt.xlabel('month')

plt.ylabel('number of reservations')

plt.legend(['not calcled', 'canceled'])

plt.show()
```



We have developed the grouped bar graph to analyze the months with the highest and lowest reservation levels according to reservation status. As can be seen, both the number of confirmed reservations and the number of canceled reservations are largest in the month of August. whereas January is the month with the most canceled reservations.

In [141]:

```
plt.figure(figsize=(15, 8))

plt.title('ADR per month', fontsize=30)

sns.barplot(x = 'month',y = 'adr', data=df[df['is_canceled'] == 1].groupby('month')[['adr']].sum().reset_index())

plt.show()
```



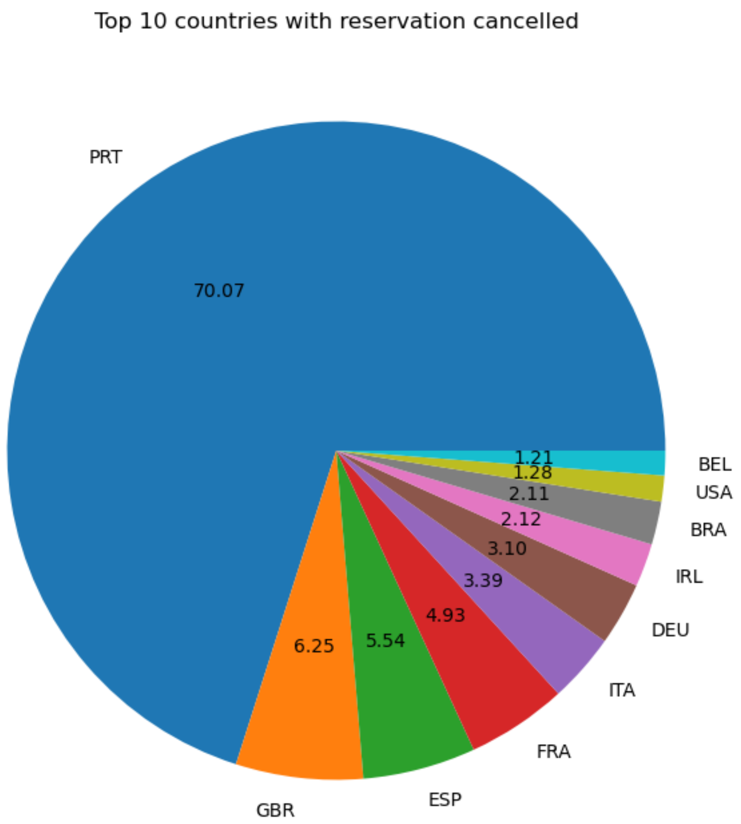
Here, we could observe than when price is higher cancelletion is also higher. for example for the month of January we could see that price is the highest compared to other months and we also could see same on the month of January calcelletion is also highest.

Now, let's see which country has the highest reservation canceled

---

In [142]:

```
cancelled_data = df[df['is_canceled'] == 1]

top_10_country = cancelled_data['country'].value_counts()[:10]


plt.figure(figsize= (8,8))

plt.title('Top 10 countries with reservation cancelled')

plt.pie(top_10_country, autopct= '%.2f', labels= top_10_country.index)

plt.show()
```



Here, we could see for the country "PRT" => Portugal cancelletion is the highest which is 70%. Hotels should try decreasing the cancelletion in the portugal country by decreasing price, providing more fecility and do more marketing to decrease the cancelletion.

In [143]:

```
df['market_segment'].value_counts()
```

Out[143]:

```
market_segment
Online TA        56402
Offline TA/TO    24159
Groups           19806
Direct           12448
Corporate         5111
Complementary      734
Aviation           237
Name: count, dtype: int64
```

Let's check the area from where guests are visiting the hotels and making reservations.Is it coming from Direct or Groups, Online or Offline Travel Agents? Around 47% of the clients come from online travel agencies, whereas 16% come from groups. Only 10% of clients book hotels directly by visiting them and making reservations.

In [145]:

```
df['market_segment'].value_counts(normalize= True)
```

Out[145]:

```
market_segment
Online TA        0.474377
Offline TA/TO    0.203193
Groups           0.166581
Direct           0.104696
Corporate        0.042987
Complementary    0.006173
Aviation         0.001993
Name: proportion, dtype: float64
```

Here, we could see Online reservations are highest which is contributing around 47% of the total reservation.

So, our hypothesis "The majority of clients are coming from offline travel agents to make their reservations." is proven wrong here.

In [147]:

```
cancelled_data['market_segment'].value_counts(normalize= True)
```

Out[147]:

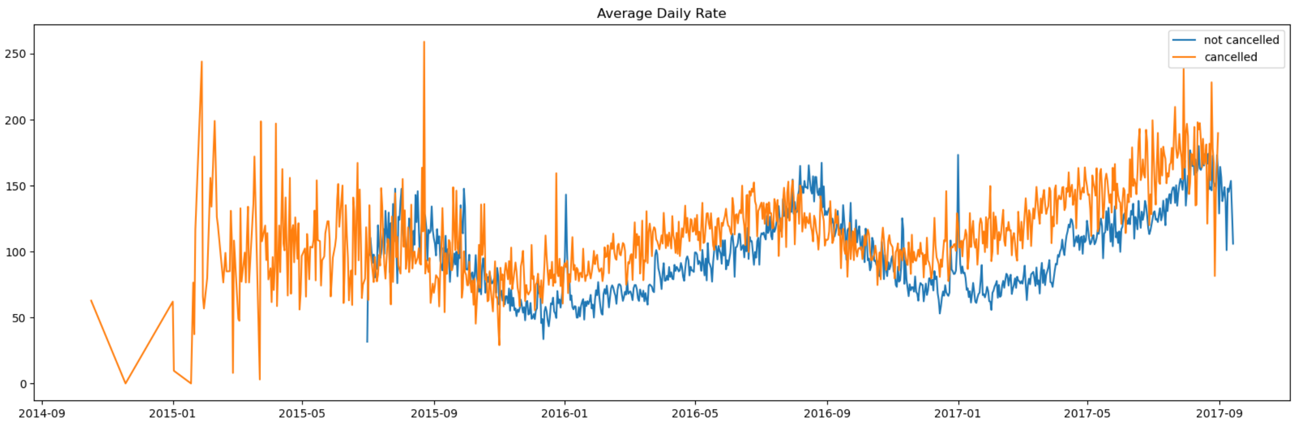
```
market_segment
Online TA        0.469696
Groups           0.273985
Offline TA/TO    0.187466
Direct           0.043486
Corporate        0.022151
Complementary    0.002038
Aviation         0.001178
Name: proportion, dtype: float64
```

Here, We also could see from the total cancelletions around 47% of the cancelletions are coming from online reservations.

Now, let's find out ADR of cancelled reservation and not cancelled reservations

In [152]:

```
cancelled_df_adr = cancelled_data.groupby('reservation_status_date')[['adr']].mean()

cancelled_df_adr. reset_index(inplace = True)

cancelled_df_adr.sort_values( 'reservation_status_date' ,inplace = True)


not_cancelled_data = df[df['is_canceled'] == 0]

not_cancelled_df_adr= not_cancelled_data.groupby('reservation_status_date')[['adr']].mean()

not_cancelled_df_adr.reset_index(inplace = True)

not_cancelled_df_adr.sort_values( 'reservation_status_date' ,inplace = True)


plt.figure(figsize = (20,6))

plt.title( 'Average Daily Rate')

plt.plot(not_cancelled_df_adr['reservation_status_date'], not_cancelled_df_adr['adr'], label = 'not cancelled')

plt.plot(cancelled_df_adr['reservation_status_date'], cancelled_df_adr['adr'], label = 'cancelled')

plt.legend()

plt.show()
```

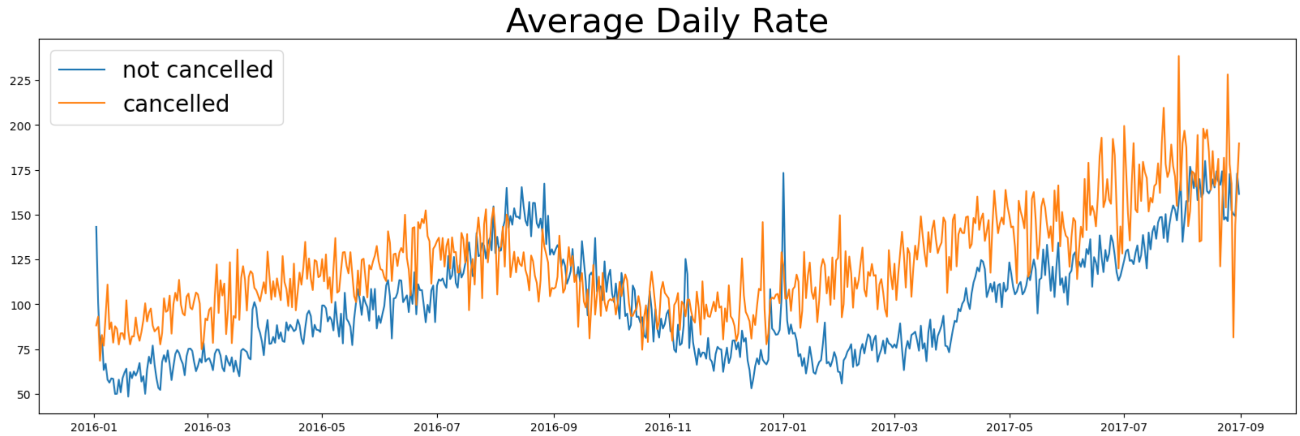


In [153]:

```
cancelled_df_adr = cancelled_df_adr[(cancelled_df_adr['reservation_status_date']>'2016') &
(cancelled_df_adr['reservation_status_date']<'2017-09')]


not_cancelled_df_adr= not_cancelled_df_adr[(not_cancelled_df_adr['reservation_status_date']>'2016') &
(not_cancelled_df_adr['reservation_status_date']<'2017-09')]
```

In [155]:

```
plt.figure(figsize=(20,6))

plt.title('Average Daily Rate', fontsize = 30)

plt.plot(not_cancelled_df_adr['reservation_status_date'], not_cancelled_df_adr['adr'], label = 'not cancelled')

plt.plot(cancelled_df_adr['reservation_status_date'], cancelled_df_adr['adr'], label = 'cancelled')

plt.legend(fontsize = 20)

plt.show()
```



As seen in the graph, reservations are canceled when the average daily rate is higher than when it is not canceled. It clearly proves all the above analysis, that the higher price leads to higher cancellation.

Suggestions

---

1. Cancellation rates rise as the price does. In order to prevent cancellations of reservations, hotels could work on their pricing strategies and try to lower the rates for specific hotels based on locations. They can also provide some discounts to the consumers.

---

2. As the ratio of the cancellation and not cancellation of the resort hotel is higher in the resort hotel than the city hotels. So the hotels should provide a reasonable discount on the room prices on weekends or on holidays.

---

3. In the month of January, hotels can start campaigns or marketing with a reasonable amount to increase their revenue as the cancellation is the highest in this month.

---

4. They can also increase the quality of their hotels and their services mainly in Portugal to reduce the cancellation rate.
