# Data Analysis project Steps

1. create a problem statement
2. identify the data you want to analyze
3. explore and clean the data
4. analyze the data to get useful insights
5. present the data interms of report or dashboard using visualization

# Business Problem :-

In recent years ,city hotel and resort have seen high cancellation rates.Each hotel is now dealing with number issues as a result, including fewer revenues and less than ideal hotel room use. Consequently lower cancellation rates is both the hotels primary goal in order to increase their efficiency in generating revenue, and for us to offer thorough bussines advice to adrees 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 the report.

# Research :-

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

In [1]:

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

In [3]:

```
#Loading the dataset
df=pd.read_csv('hotel_booking.csv')
```

In [4]:

```
#Exploratory data analysis and data cleaning
df.head()
```

Out[4]:

| | hotel | is_canceled | lead_time | arrival_date_year | arrival_date_month | arrival_date_week_number | arrival_date_day_of_month | stays_in_weekend_nights | st |
|---|---|---|---|---|---|---|---|---|---|
| **0** | Resort Hotel | 0 | 342 | 2015 | July | 27 | 1 | 0 | |
| **1** | Resort Hotel | 0 | 737 | 2015 | July | 27 | 1 | 0 | |
| **2** | Resort Hotel | 0 | 7 | 2015 | July | 27 | 1 | 0 | |
| **3** | Resort Hotel | 0 | 13 | 2015 | July | 27 | 1 | 0 | |
| **4** | Resort Hotel | 0 | 14 | 2015 | July | 27 | 1 | 0 | |

5 rows × 36 columns

In [5]:

```
df.tail()
#last 5 datas
```

Out[5]:

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

5 rows × 36 columns

In [6]:

```
df.shape
```

Out[6]:

```
(119390, 36)
```

In [7]:

```
df.columns
```

```
Out[7]: 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', 'name', 'email',
               'phone-number', 'credit_card'],
              dtype='object')
```

```
In [8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 36 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 
 32  name                            119390 non-null  object 
 33  email                           119390 non-null  object 
 34  phone-number                    119390 non-null  object 
 35  credit_card                     119390 non-null  object 
dtypes: float64(4), int64(16), object(16)
memory usage: 32.8+ MB
```

In [9]:
```
df['reservation_status_date']=pd.to_datetime(df['reservation_status_date'])
```

In [10]:
```
df.describe(include='object')
```

Out[10]:

| | hotel | arrival_date_month | meal | country | market_segment | distribution_channel | reserved_room_type | assigned_room_type | deposit_type | cust |
|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 119390 | 119390 | 119390 | 118902 | 119390 | 119390 | 119390 | 119390 | 119390 | |
| **unique** | 2 | 12 | 5 | 177 | 8 | 5 | 10 | 12 | 3 | |
| **top** | City Hotel | August | BB | PRT | Online TA | TA/TO | A | A | No Deposit | |
| **freq** | 79330 | 13877 | 92310 | 48590 | 56477 | 97870 | 85994 | 74053 | 104641 | |

In [11]:
```
#define the all object datatype
for col in df.describe(include='object').columns:
  print(col)
  print(df[col].unique())
```

```
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']
name
['Ernest Barnes' 'Andrea Baker' 'Rebecca Parker' ... 'Wesley Aguilar'
 'Caroline Conley MD' 'Ariana Michael']
email
['Ernest.Barnes31@outlook.com' 'Andrea_Baker94@aol.com'
 'Rebecca_Parker@comcast.net' ... 'Mary_Morales@hotmail.com'
 'MD_Caroline@comcast.net' 'Ariana_M@xfinity.com']
phone-number
['669-792-1661' '858-637-6955' '652-885-2745' ... '395-518-4100'
 '531-528-1017' '422-804-6403']
credit_card
['************4322' '************9157' '************3734' ...
 '************9170' '************6349' '************7959']
```

In [12]:

```
#checking the null values
df.isnull().sum()
```

Out[12]:

```
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
name                                   0
email                                  0
phone-number                           0
credit_card                            0
dtype: int64
```

In [13]:

```
df.drop(['company','agent'],axis=1,inplace=True)
df.dropna(inplace=True)
#dropping the values because of large no. of datas.
```

In [14]:

```
df.isnull().sum()
```

Out[14]:

```
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
name                              0
email                             0
phone-number                      0
credit_card                       0
dtype: int64
```

In [15]:

```
df.describe()
#describe is using for fetching the numerical columns.
```

Out[15]:

| | is_canceled | lead_time | arrival_date_year | arrival_date_week_number | arrival_date_day_of_month | stays_in_weekend_nights | stays_in_week_nigh |
|---|---|---|---|---|---|---|---|
| **count** | 118898.000000 | 118898.000000 | 118898.000000 | 118898.000000 | 118898.000000 | 118898.000000 | 118898.00000 |
| **mean** | 0.371352 | 104.311435 | 2016.157656 | 27.166555 | 15.800880 | 0.928897 | 2.5021 |
| **std** | 0.483168 | 106.903309 | 0.707459 | 13.589971 | 8.780324 | 0.996216 | 1.9001 |
| **min** | 0.000000 | 0.000000 | 2015.000000 | 1.000000 | 1.000000 | 0.000000 | 0.00000 |
| **25%** | 0.000000 | 18.000000 | 2016.000000 | 16.000000 | 8.000000 | 0.000000 | 1.00000 |
| **50%** | 0.000000 | 69.000000 | 2016.000000 | 28.000000 | 16.000000 | 1.000000 | 2.00000 |
| **75%** | 1.000000 | 161.000000 | 2017.000000 | 38.000000 | 23.000000 | 2.000000 | 3.00000 |
| **max** | 1.000000 | 737.000000 | 2017.000000 | 53.000000 | 31.000000 | 16.000000 | 41.00000 |

In [16]:

```
#reducing adr size because large no of datas.
df=df[df['adr']<5000]
```

In [17]:

```
df.describe()
```

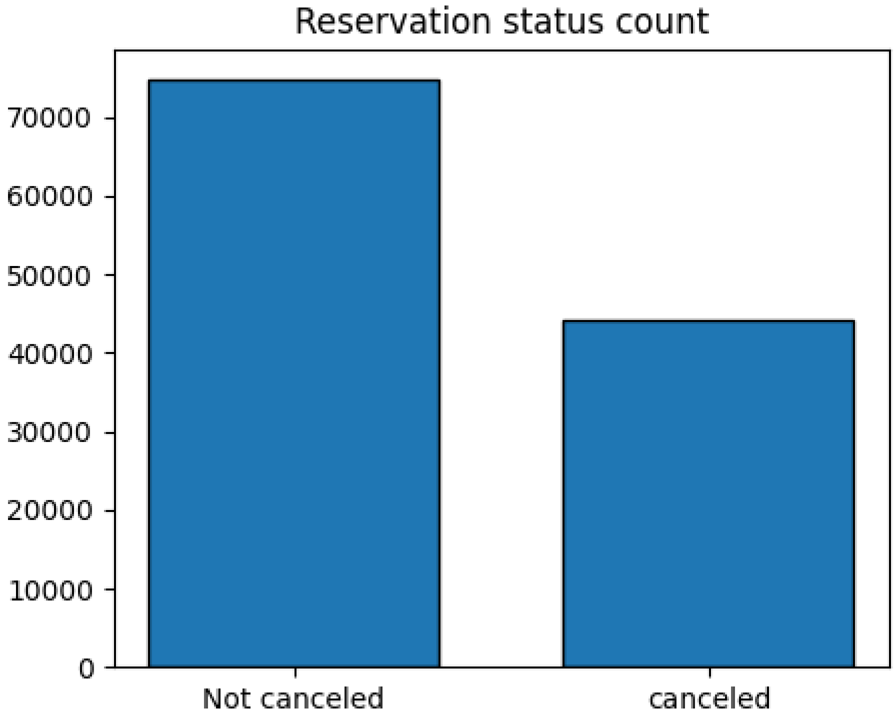
Out[17]:

| | is_canceled | lead_time | arrival_date_year | arrival_date_week_number | arrival_date_day_of_month | stays_in_weekend_nights | stays_in_week_nigh |
|---|---|---|---|---|---|---|---|
| **count** | 118897.000000 | 118897.000000 | 118897.000000 | 118897.000000 | 118897.000000 | 118897.000000 | 118897.00000 |
| **mean** | 0.371347 | 104.312018 | 2016.157657 | 27.166674 | 15.800802 | 0.928905 | 2.5021 |
| **std** | 0.483167 | 106.903570 | 0.707462 | 13.589966 | 8.780321 | 0.996217 | 1.9001 |
| **min** | 0.000000 | 0.000000 | 2015.000000 | 1.000000 | 1.000000 | 0.000000 | 0.00000 |
| **25%** | 0.000000 | 18.000000 | 2016.000000 | 16.000000 | 8.000000 | 0.000000 | 1.00000 |
| **50%** | 0.000000 | 69.000000 | 2016.000000 | 28.000000 | 16.000000 | 1.000000 | 2.00000 |
| **75%** | 1.000000 | 161.000000 | 2017.000000 | 38.000000 | 23.000000 | 2.000000 | 3.00000 |
| **max** | 1.000000 | 737.000000 | 2017.000000 | 53.000000 | 31.000000 | 16.000000 | 41.00000 |

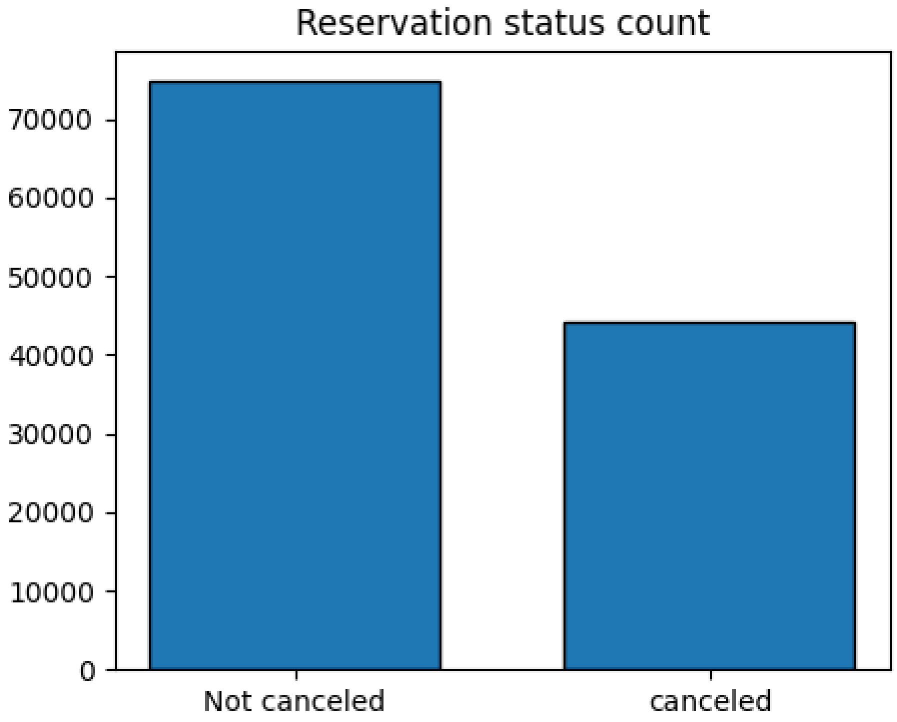
```
In [18]: #Data analysis and visualization
#here i am defining percentage of hotel cancellation
cancelled_pred=df['is_canceled'].value_counts(normalize=True)
print(cancelled_pred)

plt.figure(figsize = (5,4))
plt.title('Reservation status count')
plt.bar(['Not canceled','canceled'],df['is_canceled'].value_counts(), edgecolor = 'k',width = 0.7)
plt.show()
```

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

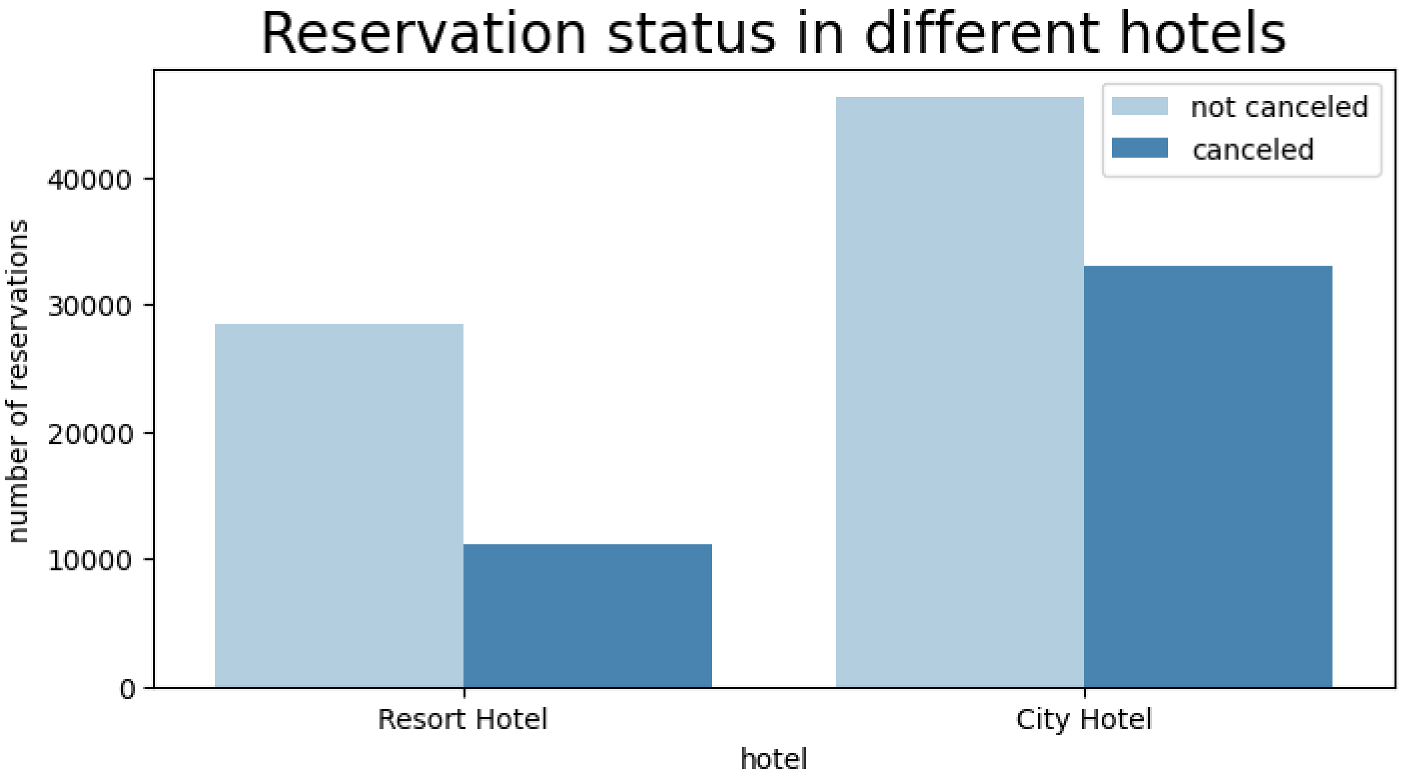


```
In [19]: plt.figure(figsize=(5,4))
plt.title('Reservation status count')
plt.bar(['Not canceled' , 'canceled'],df['is_canceled'].value_counts() , edgecolor='k' , width=0.7)
plt.show()
#fetching the data in bar graph
```



In [20]:

```
#checking whose ratio is more = canceled or not-canceled in graph.
plt.figure(figsize=(8,4))
ax1=sns.countplot(x='hotel',hue='is_canceled',data=df , palette= 'Blues')
plt.title('Reservation status in different hotels',size=20)
plt.xlabel('hotel')
plt.ylabel('number of reservations')
plt.legend(['not canceled' , 'canceled'])
plt.show()
```



In [21]:
```
resort_hotel=df[df['hotel']=='Resort Hotel']
resort_hotel['is_canceled'].value_counts(normalize=True)
```

Out[21]:
```
0    0.72025
1    0.27975
Name: is_canceled, dtype: float64
```

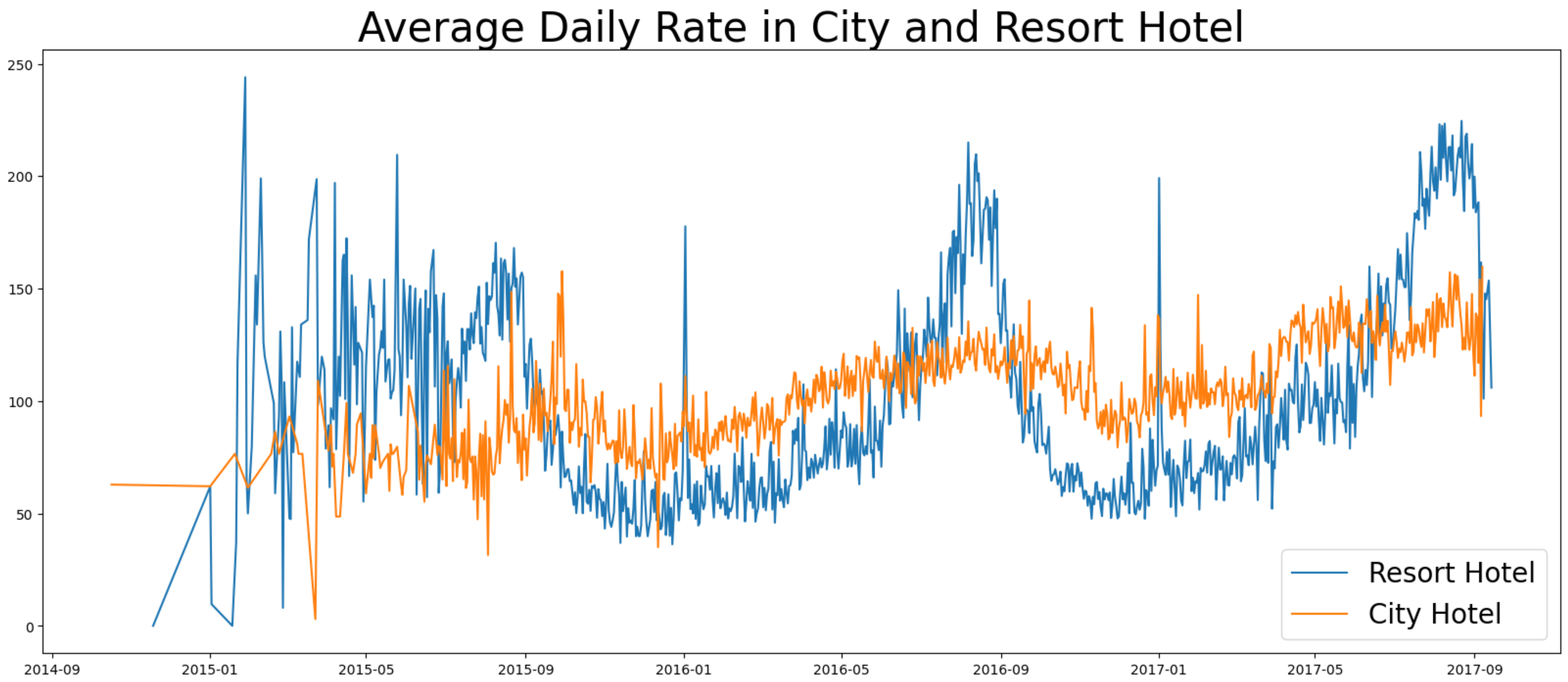
In [22]:
```
city_hotel=df[df['hotel']=='City Hotel']
city_hotel['is_canceled'].value_counts(normalize=True)
```

Out[22]:
```
0    0.582918
1    0.417082
Name: is_canceled, dtype: float64
```

In [23]:
```
#checking price part for cause for more hotel cancellation
resort_hotel=resort_hotel.groupby('reservation_status_date')[['adr']].mean()
city_hotel=city_hotel.groupby('reservation_status_date')[['adr']].mean()
```

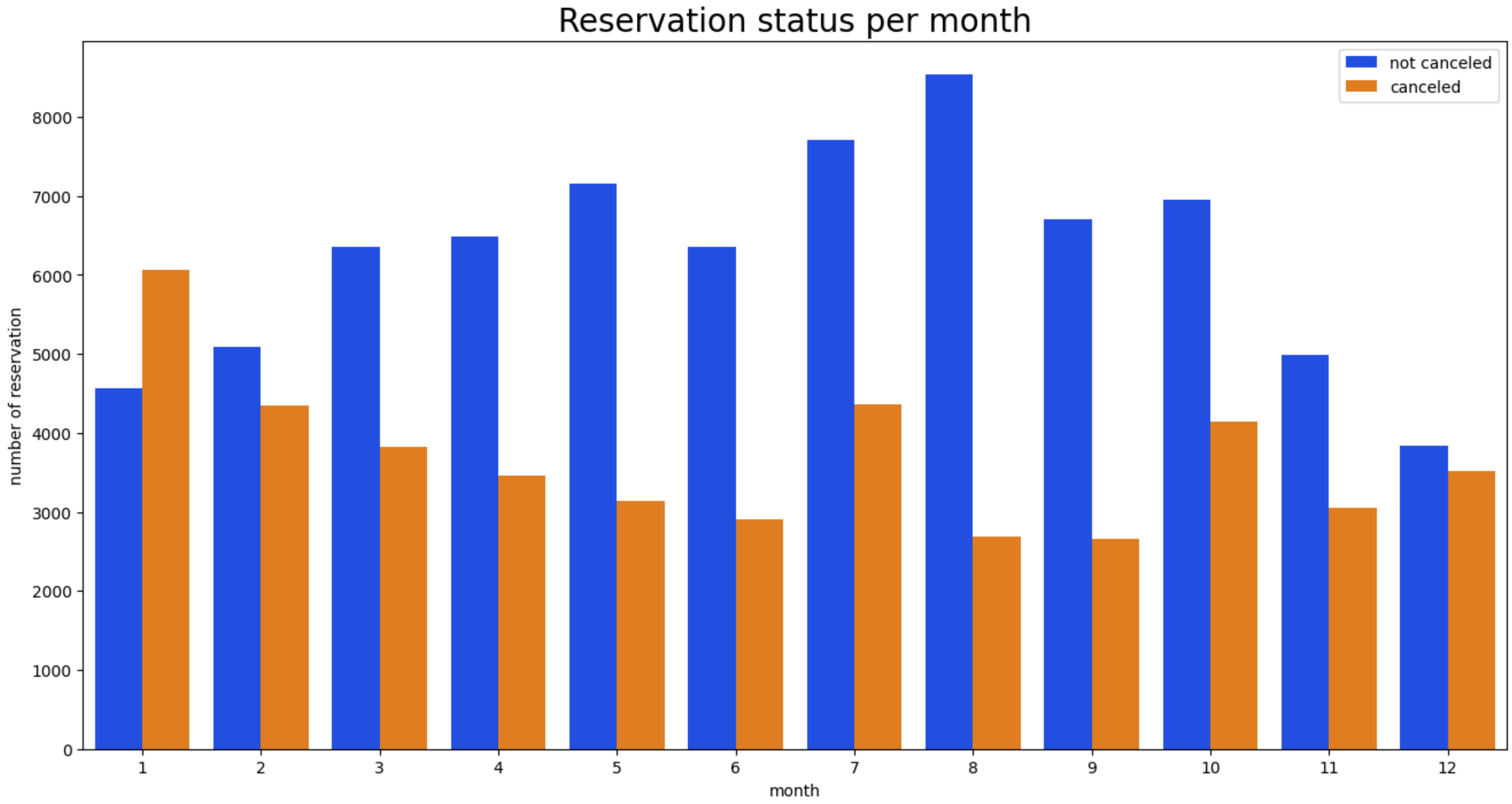
In [24]:

```
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()
```



In [25]:

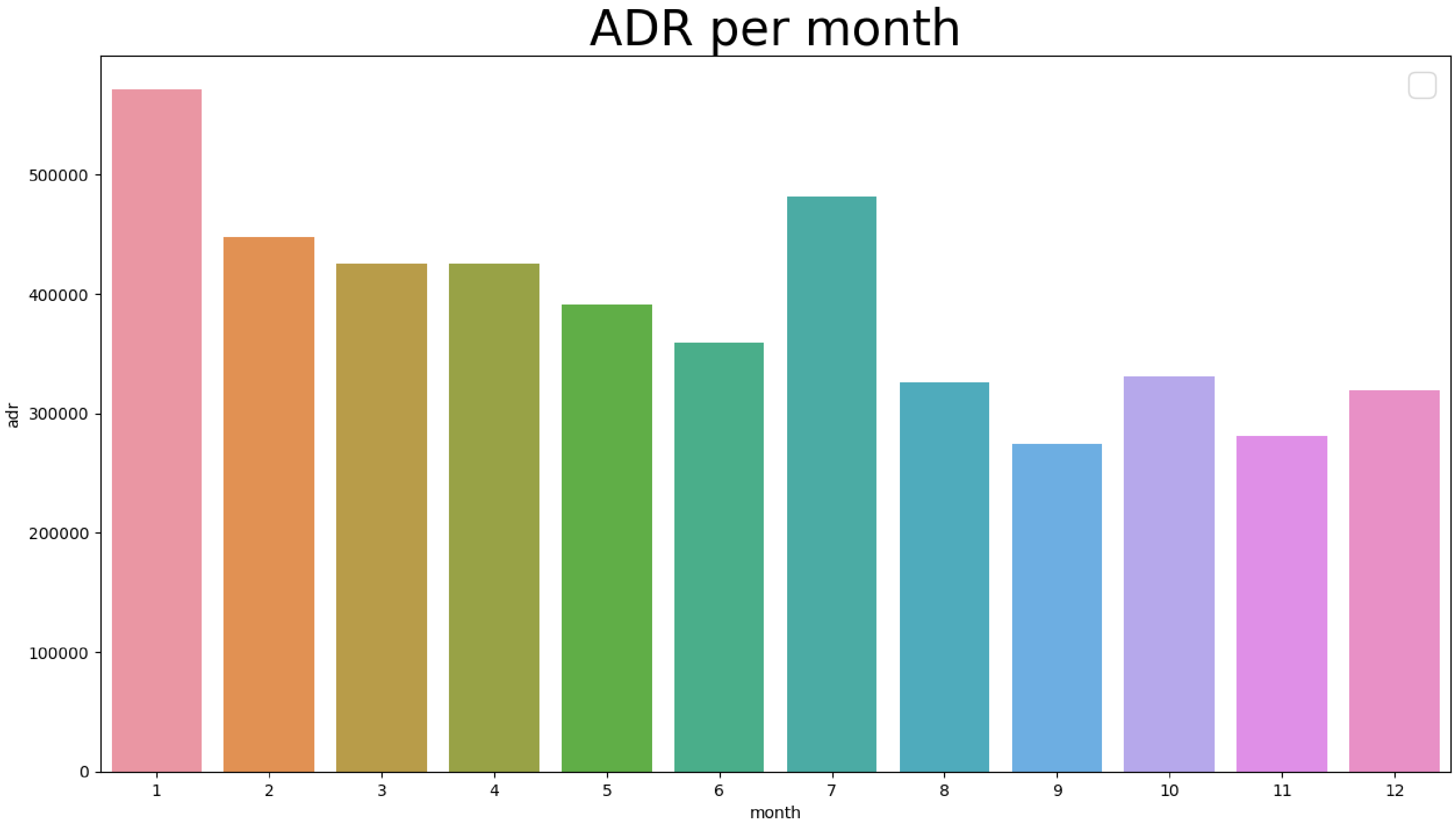
```
# To find more resevation and less resevation of month.
df['month']=df['reservation_status_date'].dt.month
plt.figure(figsize=(16,8))
ax1=sns.countplot(x='month' , hue='is_canceled',data=df , palette='bright')
plt.title('Reservation status per month',size=20)
plt.xlabel('month')
plt.ylabel('number of reservation')
plt.legend(['not canceled','canceled'])
plt.show()
```



In [26]:

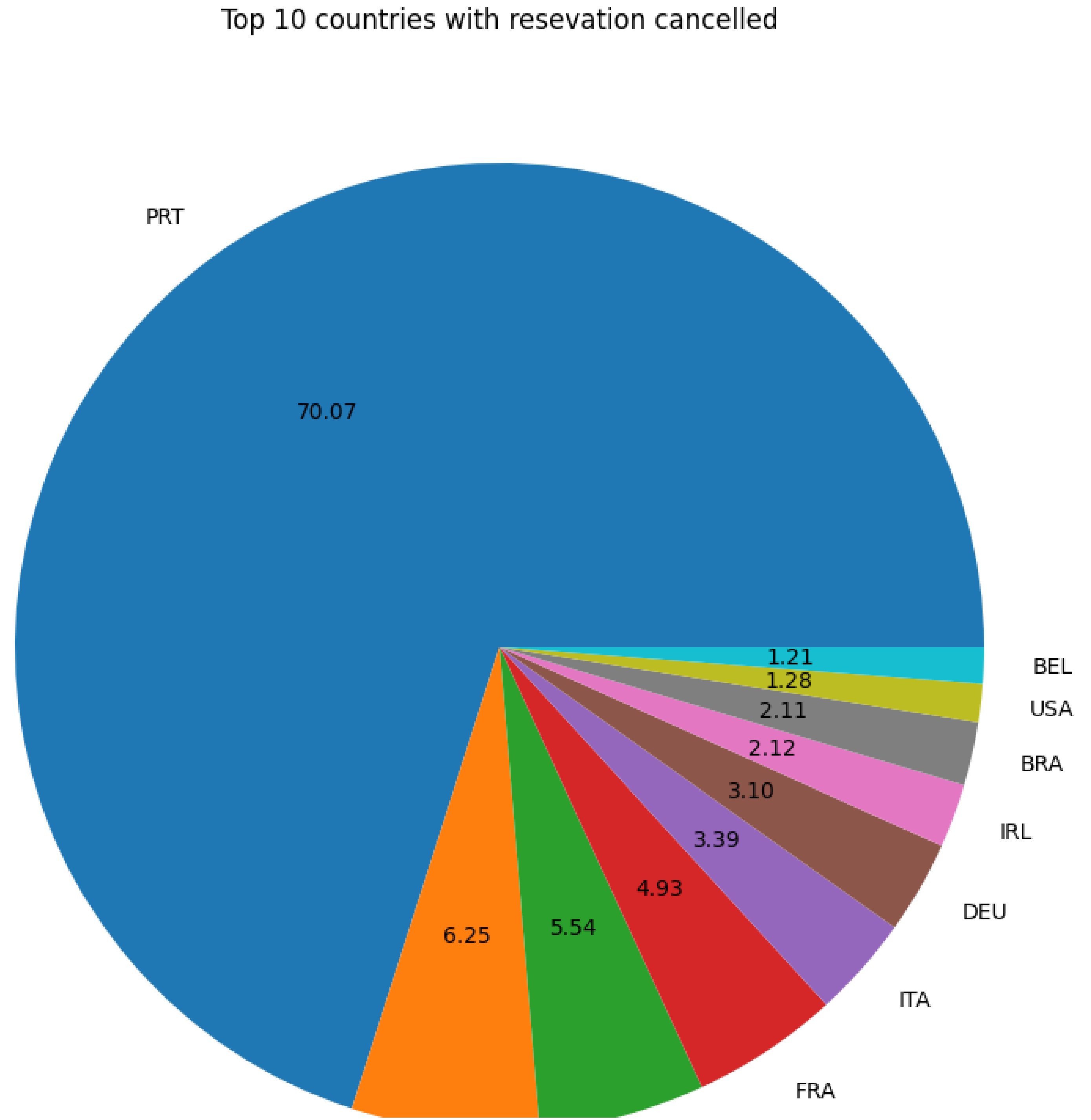
```
#cancel effect on data or not.
plt.figure(figsize=(15, 8))
plt.title('ADR per month', fontsize=30)
sns.barplot(x='month', y='adr', data=df[df['is_canceled'] == 1].groupby('month')[['adr']].sum().reset_index())
plt.legend(fontsize=20)
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend()
is called with no argument.
```



In [27]:

```
#top 10 country with reservation cancelled
cancelled_data=df[df['is_canceled']==1]
top_10_country=cancelled_data['country'].value_counts()[:10]
plt.figure(figsize=(10,10))
plt.title('Top 10 countries with resevation cancelled')
plt.pie(top_10_country,autopct= '%.2f' , labels=top_10_country.index)
plt.show()
```


Top 10 countries with resevation cancelled
PRT
70.07
BEL
1.21
USA
1.28
BRA
2.11
IRL
2.12
DEU
3.10
ITA
3.39
FRA
4.93
5.54
6.25

In [28]:

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

Out[28]:

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

In [29]:

```
df['market_segment'].value_counts(normalize = True)
#online resevation ratio
```

Out[29]:

```
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: market_segment, dtype: float64
```

In [30]:

```
cancelled_data['market_segment'].value_counts(normalize = True)
#cancelation ratio after booking online
```

Out[30]:

```
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: market_segment, dtype: float64
```

In [31]:

```
#Average of daily data
cancelled_df_adr=cancelled_data.groupby('reservation_status_date')[['adr']].mean()
```

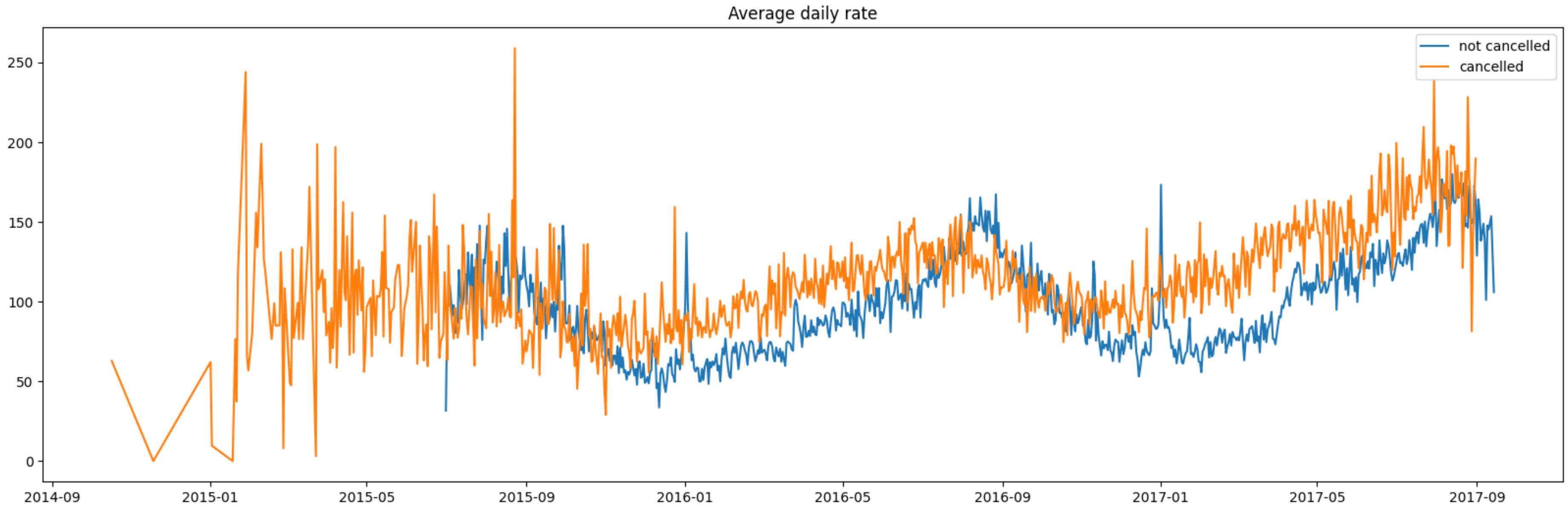
```
cancelled_df_adr.reset_index(inplace=True)

not_cancelled_data=df[df['is_canceled']==0]
not_cancelled_data_adr=not_cancelled_data.groupby('reservation_status_date')[['adr']].mean()
not_cancelled_data_adr.reset_index(inplace=True)
not_cancelled_data_adr.sort_values('reservation_status_date',inplace=True)

plt.figure(figsize=(20,6))
plt.title('Average daily rate')
plt.plot(not_cancelled_data_adr['reservation_status_date'],not_cancelled_data_adr['adr'],label='not cancelled')
plt.plot(cancelled_df_adr['reservation_status_date'],cancelled_df_adr['adr'],label='cancelled')
plt.legend()
```

Out[31]:

```
<matplotlib.legend.Legend at 0x26c4e2af160>
```



In [ ]: