Importing libraries

In [23]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Loading the dataset

In [24]:
```
df = pd.read_csv('hotel_bookings 2.csv')
df.head()
```

Out[24]:

	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_
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 × 32 columns

Exploratory Data Analysis and Data Cleaning

In [25]:
```
df.shape  ## rows = 119390, columns = 32
```

Out[25]:
```
(119390, 32)
```

In [26]:
```
df.columns
```

Out[26]:
```
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 [27]:
```
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
```

In [28]:
```
# From above we can see that some columns have null values as non-null values are less than  number of rows present
# reservation_status_date data-type is object. So, need to first convert it to date time first.

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

In [29]:
```
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
```

In [30]:
```
df.describe(include = 'object')
```

Out[30]:

	hotel	arrival_date_month	meal	country	market_segment	distribution_channel	reserved_room_type	assigned_room_type	deposit_type	customer_
count	119390	119390	119390	118902	119390	119390	119390	119390	119390	11
unique	2	12	5	177	8	5	10	12	3	
top	City Hotel	August	BB	PRT	Online TA	TA/TO	A	A	No Deposit	Trar
freq	79330	13877	92310	48590	56477	97870	85994	74053	104641	8

In [31]:
```
# To see what categories are present in the object column
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']
--------------------------------------------------
```

In [32]:
```
df.isnull().sum()
```

Out[32]:
```
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
```

In [33]:
```
nan_df = df[df.isna().any(axis = 1)]
nan_df
```

Out[33]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	s
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	
...	...	...	...	...	...	...	...	...	
119385	City Hotel	0	23	2017	August	35	30	2	
119386	City Hotel	0	102	2017	August	35	31	2	
119387	City Hotel	0	34	2017	August	35	31	2	
119388	City Hotel	0	109	2017	August	35	31	2	
119389	City Hotel	0	205	2017	August	35	29	2	

119173 rows × 32 columns

In [34]:
```
df.drop(['agent', 'company'], axis =1, inplace = True)
df.dropna(inplace = True)
```

In [35]:
```
df.isnull().sum()
```

Out[35]:
```
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
```

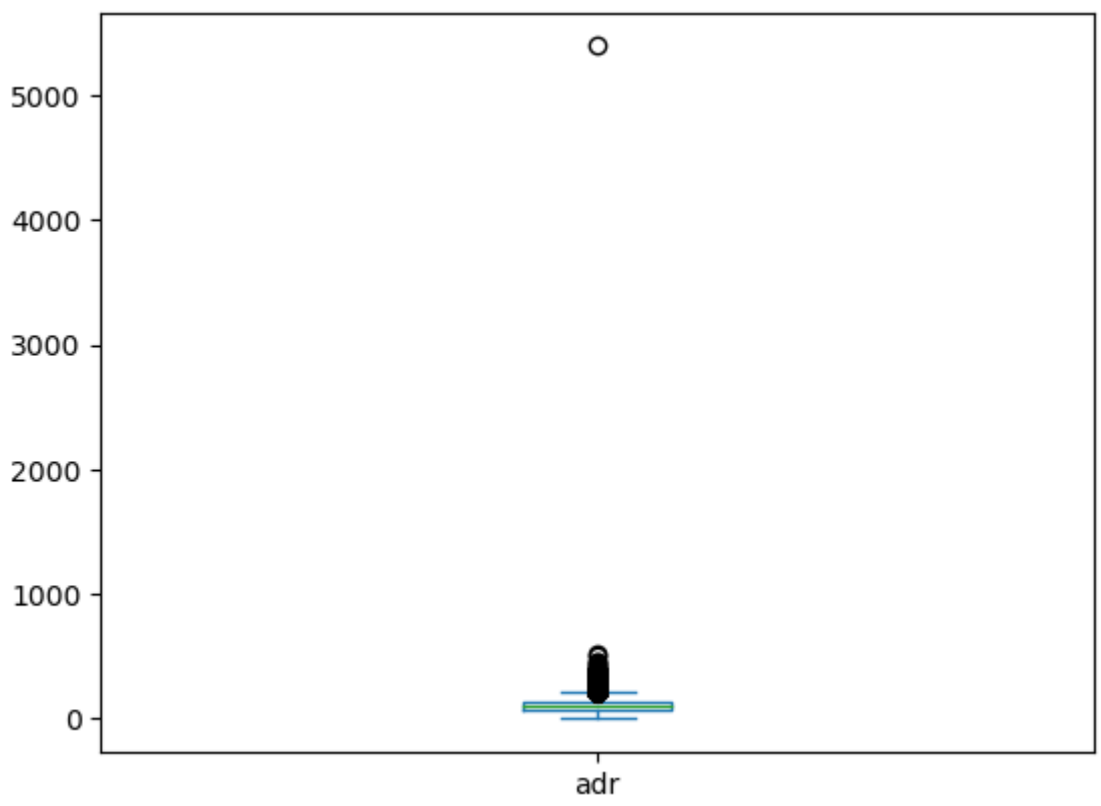
In [36]:
```
df.describe()
```

Out[36]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights
count	118898.000000	118898.000000	118898.000000	118898.000000	118898.000000	118898.000000	118898.000000
mean	0.371352	104.311435	2016.157656	27.166555	15.800880	0.928897	2.502145
std	0.483168	106.903309	0.707459	13.589971	8.780324	0.996216	1.900168
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000	0.000000
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	0.000000	1.000000
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	1.000000	2.000000
75%	1.000000	161.000000	2017.000000	38.000000	23.000000	2.000000	3.000000
max	1.000000	737.000000	2017.000000	53.000000	31.000000	16.000000	41.000000

In [37]:
```
df['adr'].plot(kind = 'box')
```

Out[37]: <AxesSubplot:>



In [38]:
```
# There are many outliers but adr i.e, the price column is necessary for us to find insights from the data.
# So need to remove that to get clean data

df = df[df['adr'] < 5000]
```

In [39]:
```
df.describe()
```

Out[39]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights
count	118897.000000	118897.000000	118897.000000	118897.000000	118897.000000	118897.000000	118897.000000
mean	0.371347	104.312018	2016.157657	27.166674	15.800802	0.928905	2.502157
std	0.483167	106.903570	0.707462	13.589966	8.780321	0.996217	1.900171
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000	0.000000
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	0.000000	1.000000
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	1.000000	2.000000
75%	1.000000	161.000000	2017.000000	38.000000	23.000000	2.000000	3.000000
max	1.000000	737.000000	2017.000000	53.000000	31.000000	16.000000	41.000000

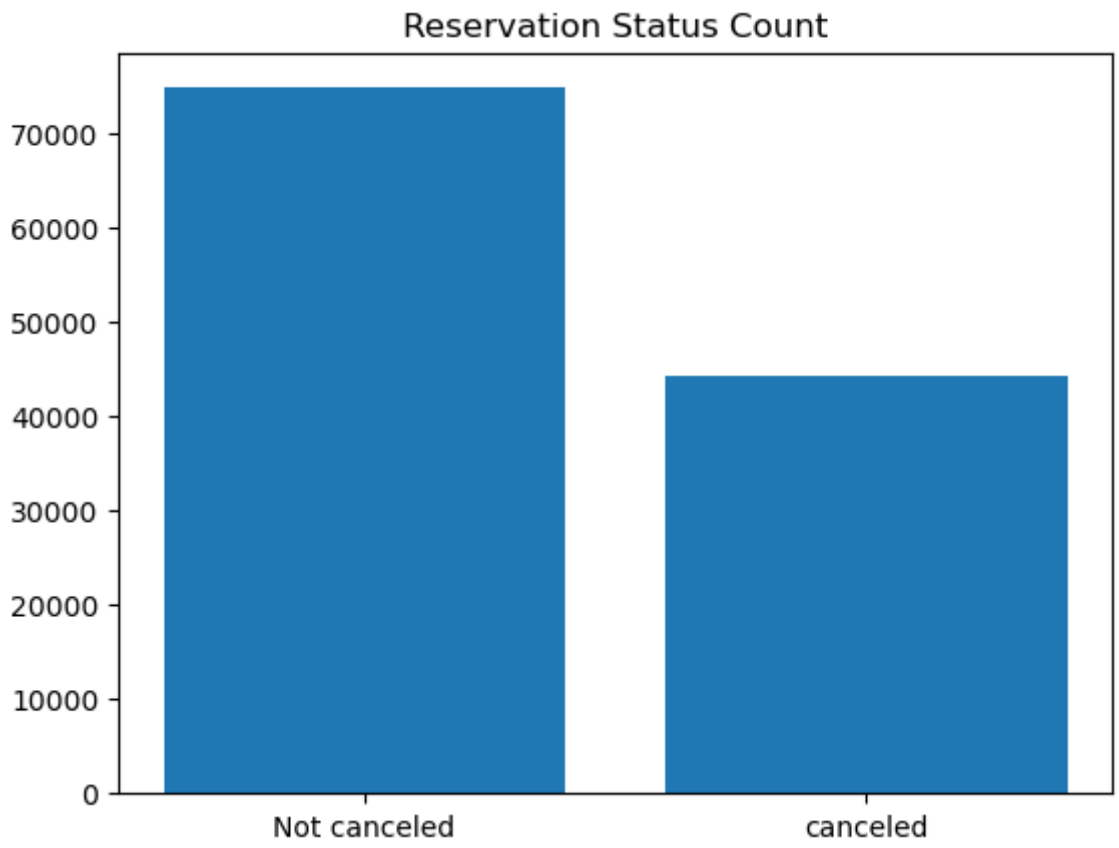
Data Analysis and Visualisations

In [47]:
```
#Cancelled vs not cancelled resrvation in the whole data set

cancelled = df['is_canceled'].value_counts()
print(cancelled)

plt.bar(['Not canceled', 'canceled'], cancelled)
plt.title('Reservation Status Count')
plt.show()
```

```
0    74745
1    44152
Name: is_canceled, dtype: int64
```



In [55]:
```
#Depending on hotels we see cancellation to not cancellation rate

sns.countplot(x='hotel', hue ='is_canceled', data = df)
plt.title('Reservation status in different hotels')
plt.xlabel('hotel')
plt.ylabel('number of reservations')
plt.legend(['not canceled','canceled'])
plt.show()
```

In [57]:
```
# Percentage calculation for resort hotel

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

Out[57]:
```
0    0.72025
1    0.27975
Name: is_canceled, dtype: float64
```

In [58]:
```
# Percentage calculation for city hotel

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

Out[58]:
```
0    0.582918
1    0.417082
Name: is_canceled, dtype: float64
```

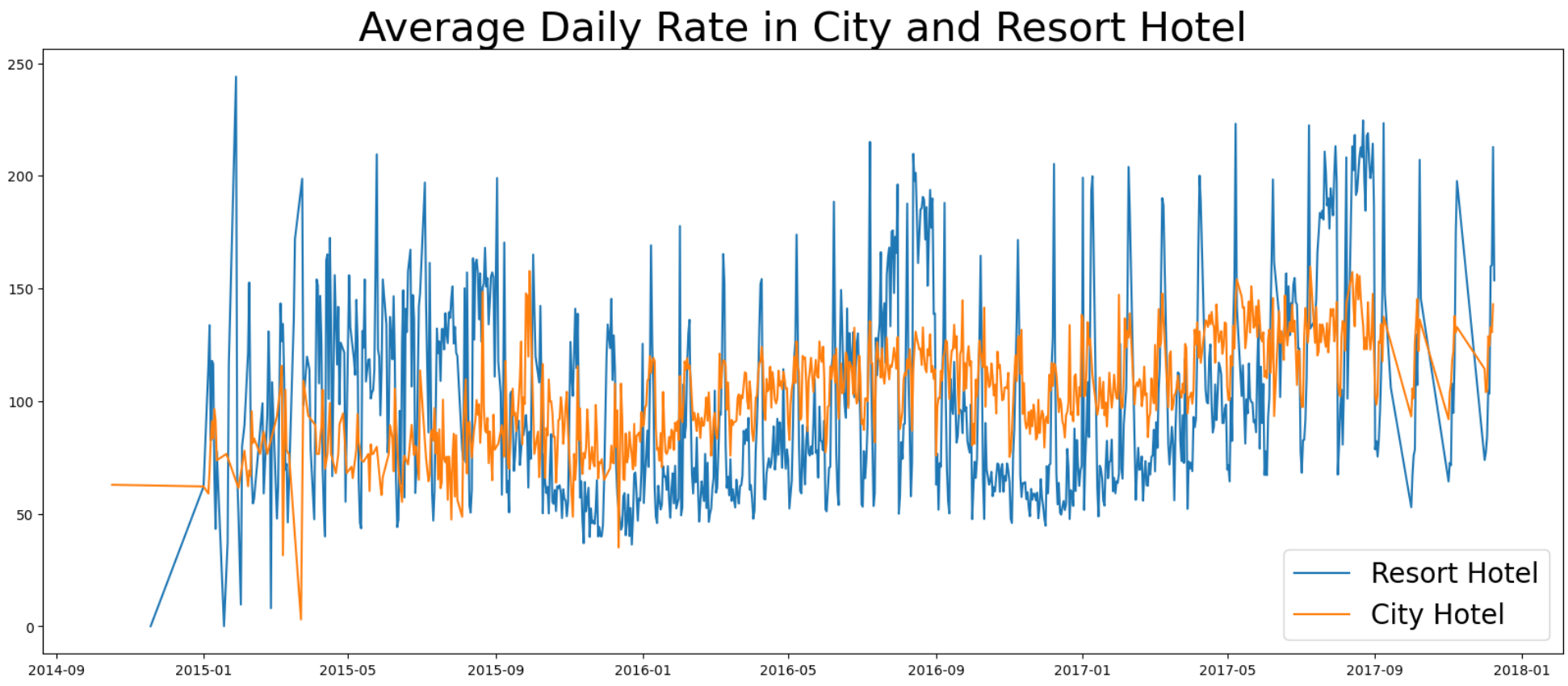
In [60]:
```
# To check if there is a effect of price in cancellation in case of both the hotels

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

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

```
reservation_status_date
2014-11-18      0.000000
2015-01-01     61.966667
2015-01-05    115.363333
2015-01-06    133.677143
2015-01-07     82.485455
                  ...
2017-12-05    103.287534
2017-12-06    159.808929
2017-12-07    160.306275
2017-12-08    212.767222
2017-12-09    153.570000
Name: adr, Length: 913, dtype: float64
reservation_status_date
2014-10-17     62.800000
2015-01-01     62.063158
2015-01-05     58.900000
2015-01-06     69.216667
2015-01-07     82.877500
                  ...
2017-12-04    128.755465
2017-12-05    124.544536
2017-12-06    132.725882
2017-12-07    130.473617
2017-12-08    142.949080
Name: adr, Length: 864, dtype: float64
```

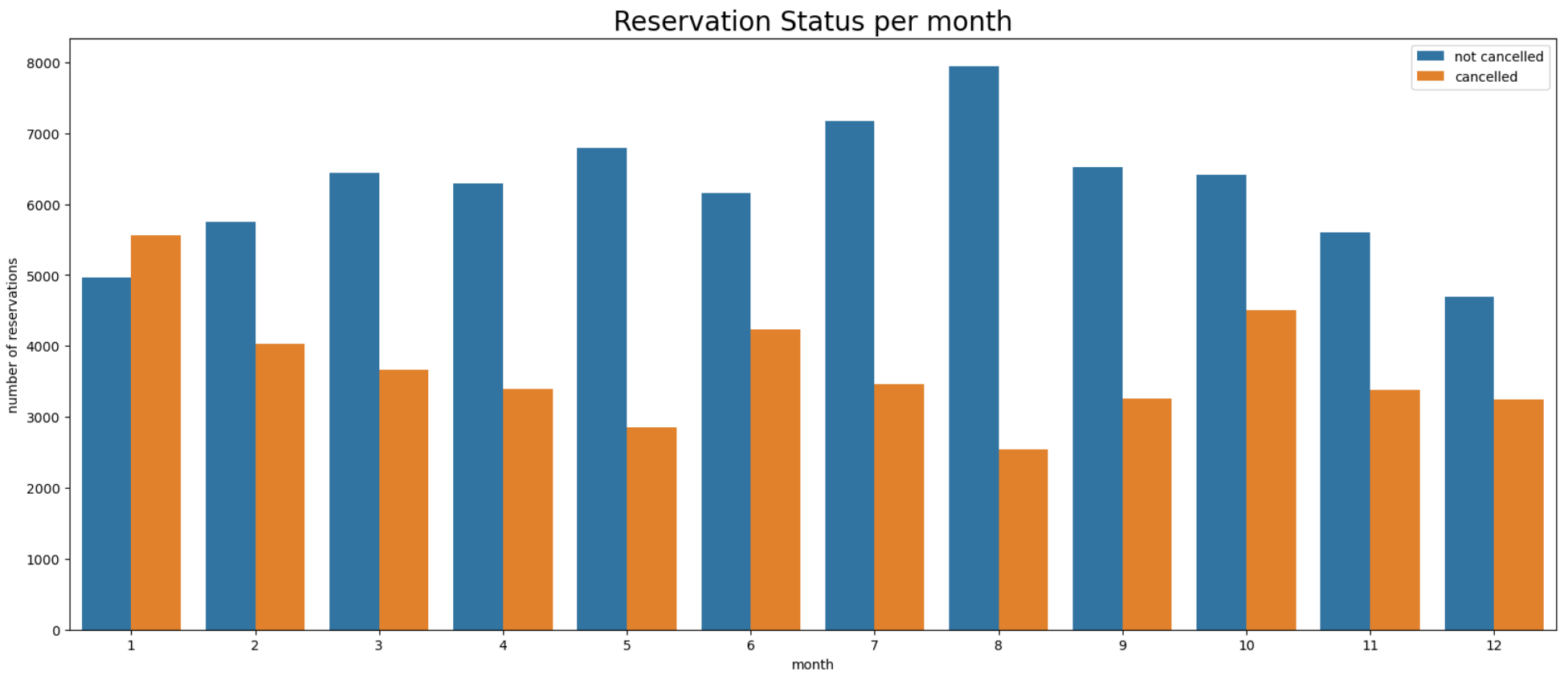
In [71]:
```
plt.figure(figsize=(20,8))
plt.title('Average Daily Rate in City and Resort Hotel', fontsize = 30)
plt.plot(resort_hotel_set.index, resort_hotel_set, label = 'Resort Hotel')
plt.plot(city_hotel_set.index, city_hotel_set, label = 'City Hotel')
plt.legend(fontsize = 20)
plt.show()
```



In [74]:
```
# Month-wise reservation and cancellations for the whole dataset

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

plt.figure(figsize=(20,8))
sns.countplot(x ='month', hue='is_canceled', data=df)
plt.title('Reservation Status per month',size = 20)
plt.ylabel('number of reservations')
plt.legend(['not cancelled', 'cancelled'])
plt.show()
```



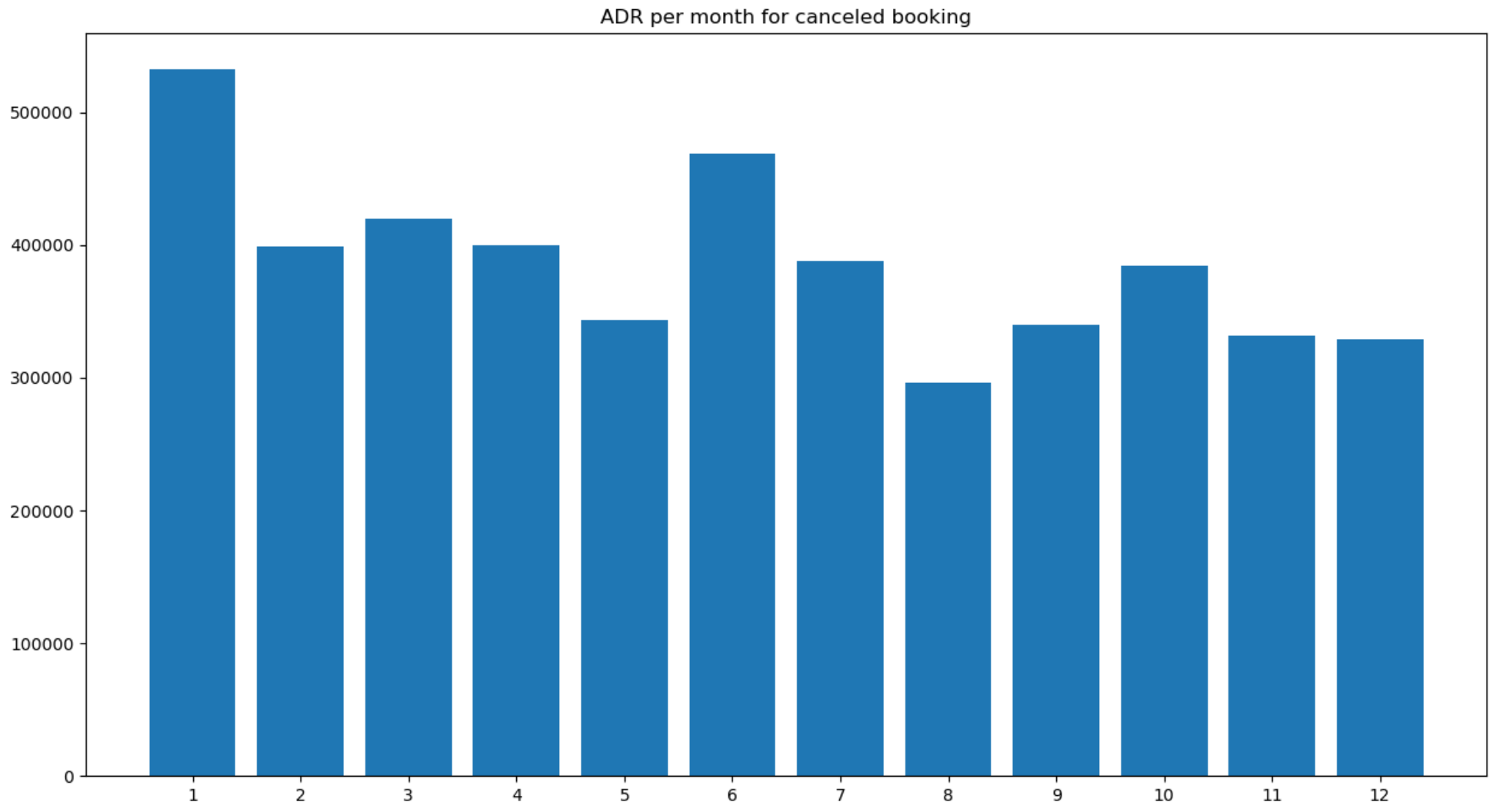
In [77]:
```
# Plotting the average daily rate for each month on cancellations

cancel_adr_df = df[df['is_canceled']==1]
result_set = cancel_adr_df.groupby('month').sum()['adr']
result_set
```

Out[77]:
```
month
1     532660.54
2     399081.98
3     419319.79
4     399361.20
5     343229.69
6     468827.36
7     387597.85
8     296665.40
9     340233.39
10    384742.47
11    331389.62
12    329229.60
Name: adr, dtype: float64
```

In [81]:
```
timeline = [month for month,cancel_adr_df in cancel_adr_df.groupby('month')]

plt.figure(figsize = (15,8))
plt.bar(timeline,result_set)
plt.xticks(timeline)
plt.title('ADR per month for canceled booking')
plt.show()
```

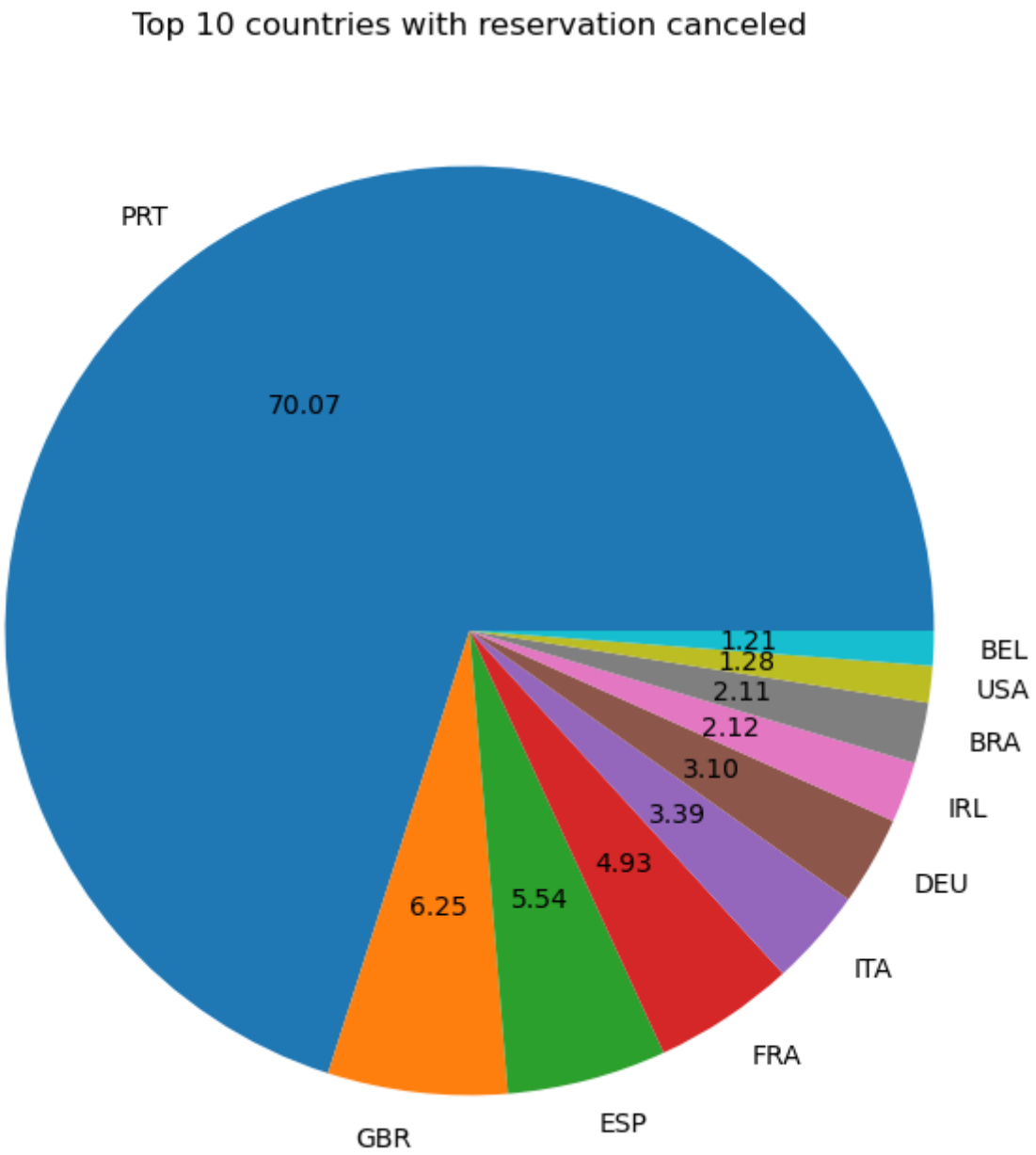


In [83]:
```
# Cancellations rate based on countries

cancelled_data = df[df['is_canceled']==1]
top_10_country = cancelled_data['country'].value_counts()[:10]
top_10_country
```

Out[83]:
```
PRT    27514
GBR     2453
ESP     2177
FRA     1934
ITA     1333
DEU     1218
IRL      832
BRA      830
USA      501
BEL      474
Name: country, dtype: int64
```

In [85]:
```
plt.figure(figsize=(10,8))
plt.pie(top_10_country, autopct = '%.2f', labels = top_10_country.index)
plt.title('Top 10 countries with reservation canceled')
plt.show()
```



In [86]:
```
# Let's see from where the clients are coming.... are they coming more from online TA or anything else

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

Out[86]:
```
Online TA        56402
Offline TA/TO    24159
Groups           19806
Direct           12448
Corporate         5111
Complementary      734
Aviation           237
Name: market_segment, dtype: int64
```

In [87]:
```
df['market_segment'].value_counts(normalize = True)
```

Out[87]:
```
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 [88]:
```
# Percentage cancellations from the cancelled df to see from whom cancellations are coming more

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

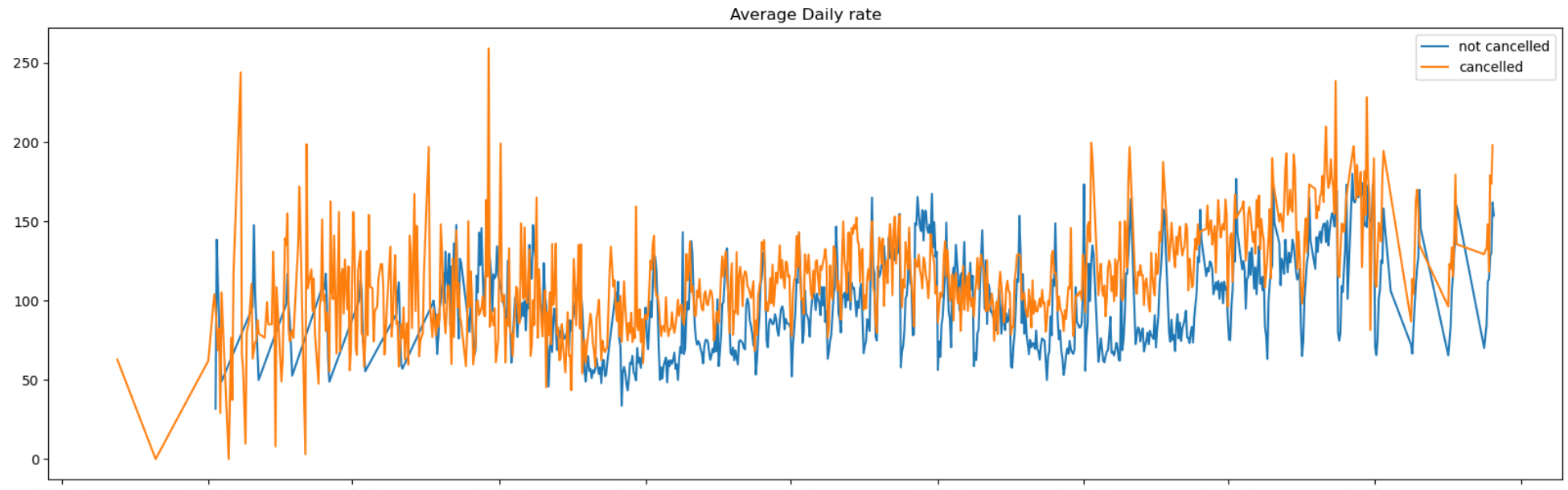
Out[88]:
```
Online TA        0.469696
Groups           0.273985
Offline TA/TO    0.187466
Direct           0.043486
Corporate        0.022151
Complementary    0.002038
Aviation         0.001178
Name: market_segment, dtype: float64
```

In [93]:
```
# canceled vs not cancelled comparison for 'adr'

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

not_cancelled_data = df[df['is_canceled']==0]
not_cancel_df = not_cancelled_data.groupby('reservation_status_date').mean()['adr']


plt.figure(figsize=(20,6))
plt.plot(not_cancel_df.index, not_cancel_df, label = 'not cancelled')
plt.plot(cancel_df.index, cancel_df, label = 'cancelled')
plt.title('Average Daily rate')
plt.legend()
plt.show()
```



In [102]:
```
# Filtering the data based on date to see more compact result

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

cancelled_data
```

Out[102]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights
3773	Resort Hotel	1	0	2016	January	2	4	1
3774	Resort Hotel	1	3	2016	January	2	4	1
3790	Resort Hotel	1	0	2016	January	2	7	0
3796	Resort Hotel	1	2	2016	January	2	8	0
3798	Resort Hotel	1	6	2016	January	2	8	0
...	...	...	...	...	...	...	...	...
110280	City Hotel	1	132	2017	April	17	25	0
111355	City Hotel	1	4	2017	June	23	5	1
111924	City Hotel	1	7	2017	May	22	31	0
111925	City Hotel	1	6	2017	July	29	17	1
117295	City Hotel	1	0	2017	August	31	2	0

30920 rows × 31 columns

In [103]:
```
not_cancelled_data = not_cancelled_data[(not_cancelled_data['reservation_status_date'] > '2016') & (not_cancelled_data
                                                                                                   ['reservation_status_date'] < '2017-09')]
not_cancelled_data
```

Out[103]:

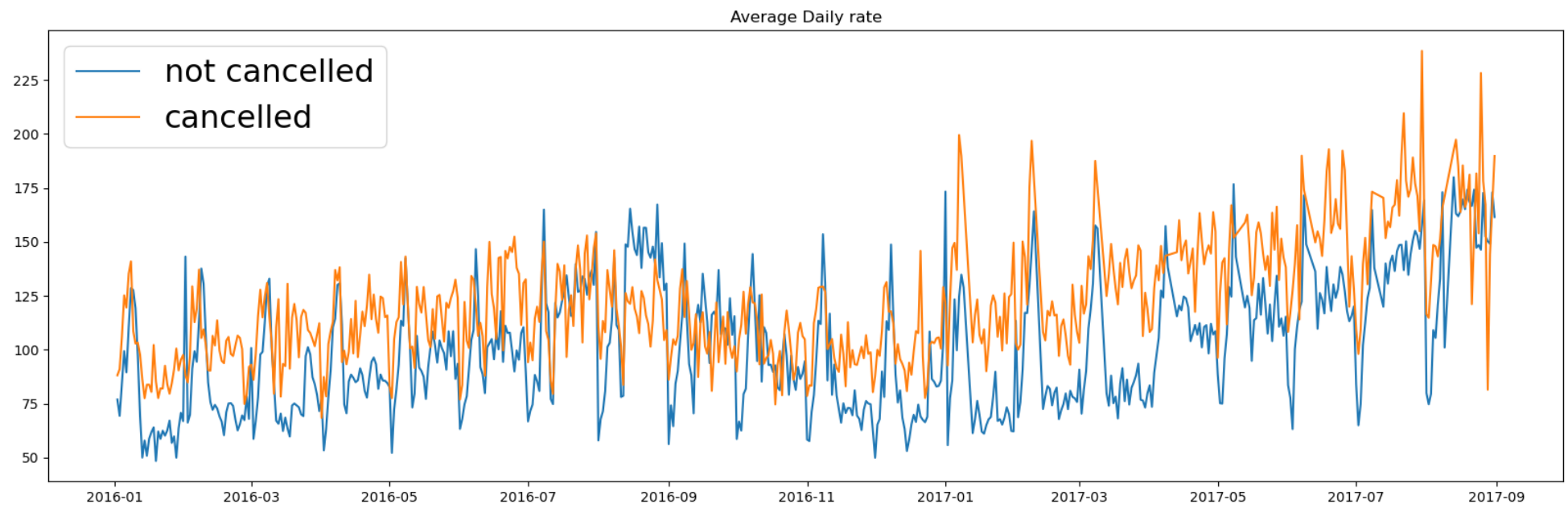
	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	s
3446	Resort Hotel	0	183	2015	December	50	8	8	
3469	Resort Hotel	0	49	2015	December	51	17	6	
3564	Resort Hotel	0	65	2015	December	52	23	2	
3621	Resort Hotel	0	129	2015	December	52	26	4	
3634	Resort Hotel	0	90	2015	December	53	28	1	
...	...	...	...	...	...	...	...	...	
119385	City Hotel	0	23	2017	August	35	30	2	
119386	City Hotel	0	102	2017	August	35	31	2	
119387	City Hotel	0	34	2017	August	35	31	2	
119388	City Hotel	0	109	2017	August	35	31	2	
119389	City Hotel	0	205	2017	August	35	29	2	

58030 rows × 31 columns

In [105]:
```
cancel_df = cancelled_data.groupby('reservation_status_date').mean()['adr']


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


plt.figure(figsize=(20,6))
plt.plot(not_cancel_df.index, not_cancel_df, label = 'not cancelled')
plt.plot(cancel_df.index, cancel_df, label = 'cancelled')
plt.title('Average Daily rate')
plt.legend(fontsize = 23)
plt.show()
```



In [ ]: