In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
```

In [2]:
```
df = pd.read_excel('hotel_bookings 2.xlsx')
```

In [3]:
```
df.shape
```

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

In [4]:
```
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 [5]:
```
df['reservation_status_date'] = pd.to_datetime(df['reservation_status_date'])
```

```
C:\Users\archa\AppData\Local\Temp\ipykernel_31852\2355418346.py:1: UserWarning: Parsing dates in DD/MM/YYYY form
at when dayfirst=False (the default) was specified. This may lead to inconsistently parsed dates! Specify a form
at to ensure consistent parsing.
  df['reservation_status_date'] = pd.to_datetime(df['reservation_status_date'])
```

In [6]:
```
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 [7]:
```
df.describe(include = 'object')
```

Out[7]:

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

In [8]:
```
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 [9]:
```
df.isnull().sum()
```

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
dtype: int64
```

In [10]:
```
df.drop(['agent','company'], axis = 1, inplace = True)
```

In [11]:
```
df.dropna(inplace = True)
```

In [12]:
```
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                        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 [13]:
```
df.describe()
```

Out[13]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays
count	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	
std	0.483168	106.903309	0.707459	13.589971	8.780324	0.996216	
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	

In [14]:
```
df = df[df['adr'] < 5000]
```

In [15]:
```
Canceled_perc = df['is_canceled'].value_counts(normalize = True)
print(Canceled_perc)
```

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

In [16]:
```
plt.figure(figsize = (5,4))
plt.title('Reservation Status counts')
plt.bar(['Not canceled','canceled'],df['is_canceled'].value_counts(),edgecolor = 'k', width = 0.6)
plt.show()
```

In [17]:
```
plt.figure(figsize = (7,5))
ax1 = sns.countplot(x = 'hotel', hue = 'is_canceled', data = df, palette = 'dark')
plt.title('Reservation count hotels wise', size = 20)
plt.xlabel('hotel')
plt.ylabel('Number of reservation')
plt.legend(['not canceled','canceled'])
plt.show()
```

In [18]:
```
Resort_Hotel = df[df['hotel'] == 'Resort Hotel']
Resort_Hotel['is_canceled'].value_counts(normalize = True)
```

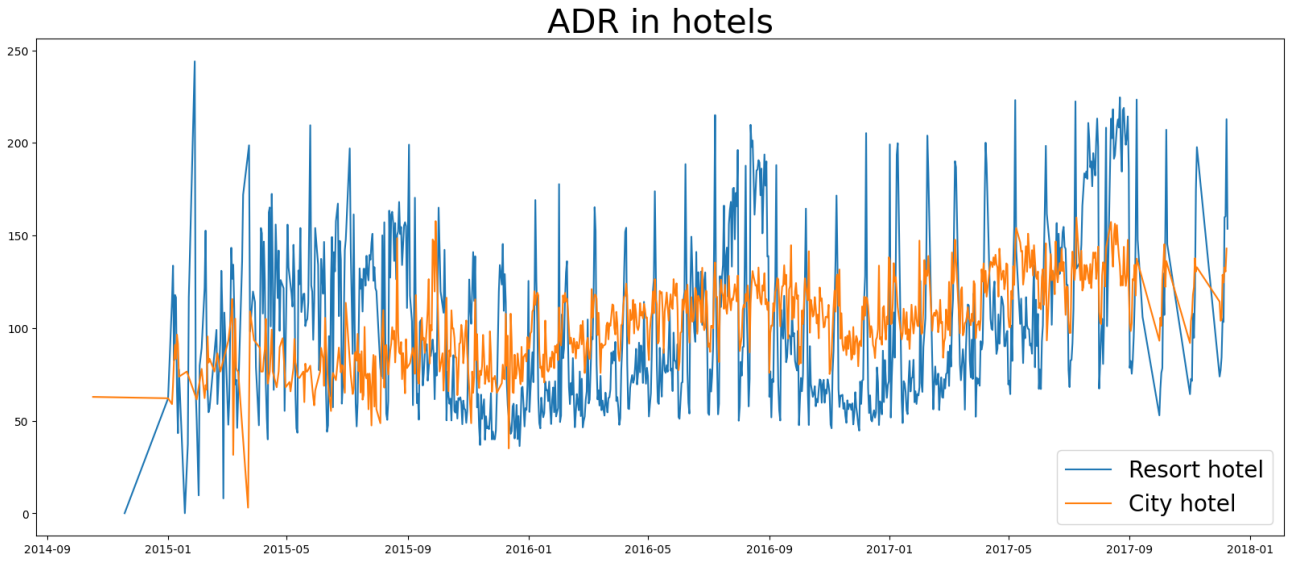
Out[18]:
```
0    0.72025
1    0.27975
Name: is_canceled, dtype: float64
```

In [19]:
```
City_Hotel = df[df['hotel'] == 'City Hotel']
City_Hotel['is_canceled'].value_counts(normalize = True)
```

Out[19]:
```
0    0.582918
1    0.417082
Name: is_canceled, dtype: float64
```

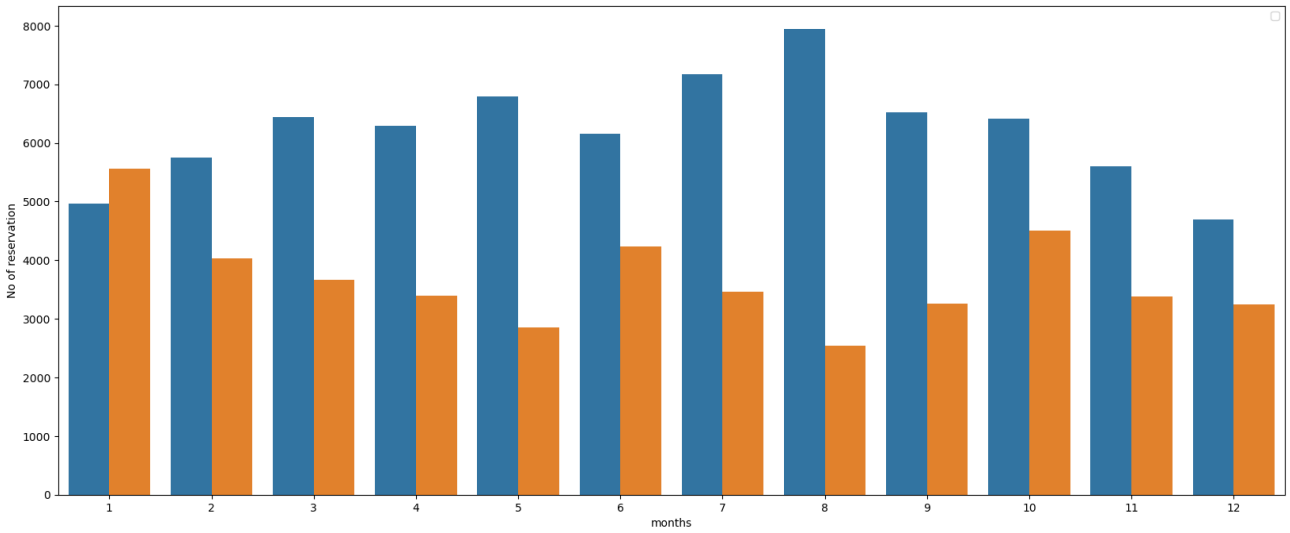
In [20]:
```
Resort_Hotel = Resort_Hotel.groupby('reservation_status_date')[['adr']].mean()
City_Hotel = City_Hotel.groupby('reservation_status_date')[['adr']].mean()
```

In [21]:
```
plt.figure(figsize = (20,8))
plt.title('ADR in hotels', size = 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()
```



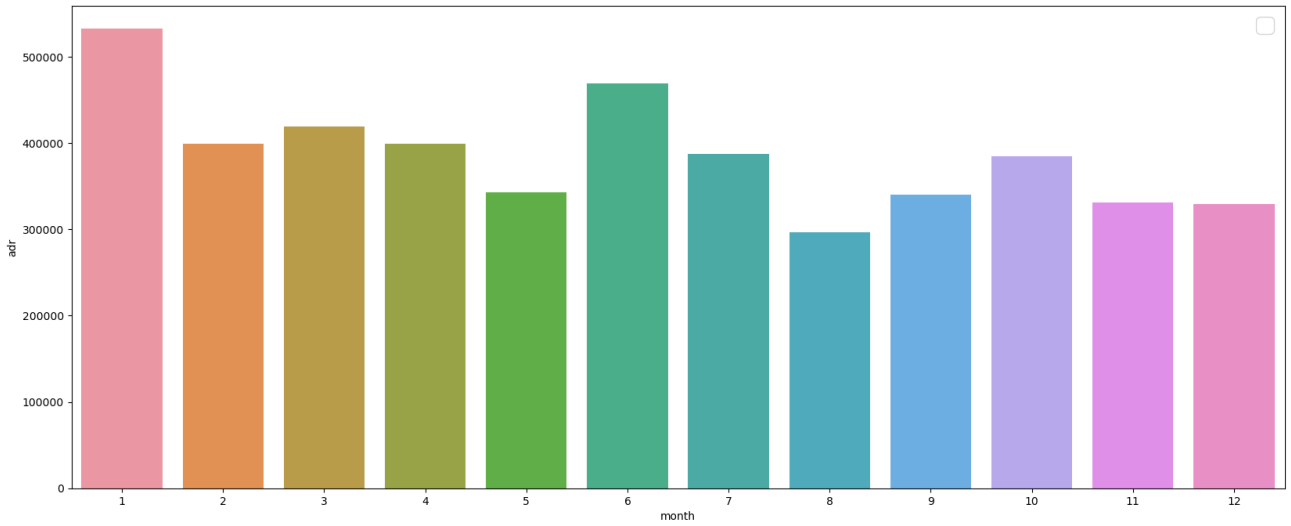
In [22]:
```
df['month'] = df['reservation_status_date'].dt.month
plt.figure(figsize = (20,8))
ax1 = sns.countplot(x = 'month' , hue = 'is_canceled', data = df)
plt.xlabel('months')
plt.ylabel('No of reservation')
plt.legend('Not canceled', 'canceled')
plt.show()
```

```
C:\Users\archa\AppData\Local\Temp\ipykernel_31852\3786250238.py:6: UserWarning: Legend does not support handles
for str instances.
A proxy artist may be used instead.
See: https://matplotlib.org/stable/tutorials/intermediate/legend_guide.html#controlling-the-legend-entries (http
s://matplotlib.org/stable/tutorials/intermediate/legend_guide.html#controlling-the-legend-entries)
  plt.legend('Not canceled', 'canceled')
```

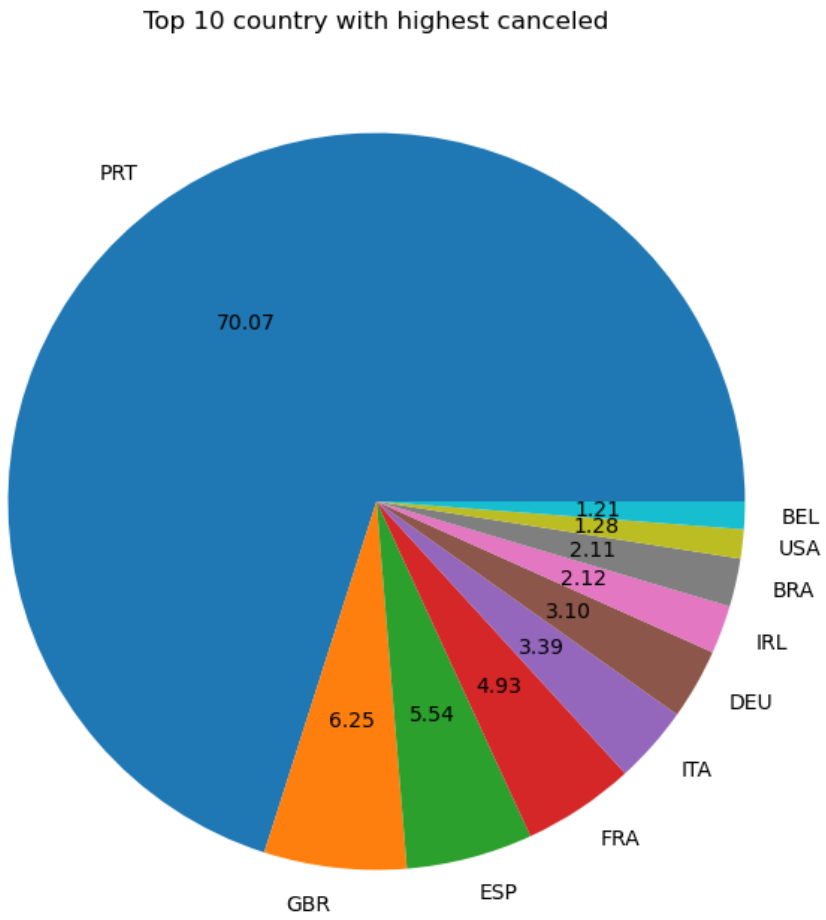


In [23]:
```
plt.figure(figsize = (20,8))
sns.barplot(x='month', y='adr', data=df[df['is_canceled'] == 1].groupby('month')[['adr']].sum().reset_index())
plt.legend(fontsize = 20)
plt.show()
```

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



In [24]:
```
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 country with highest canceled')
plt.pie(top_10_country, autopct ='%.2f' , labels = top_10_country.index)
plt.show()
```



In [25]:
```
df['market_segment'].value_counts(normalize = True)
```

Out[25]:
```
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 [26]:
```
cancelled_data['market_segment'].value_counts(normalize = True)
```

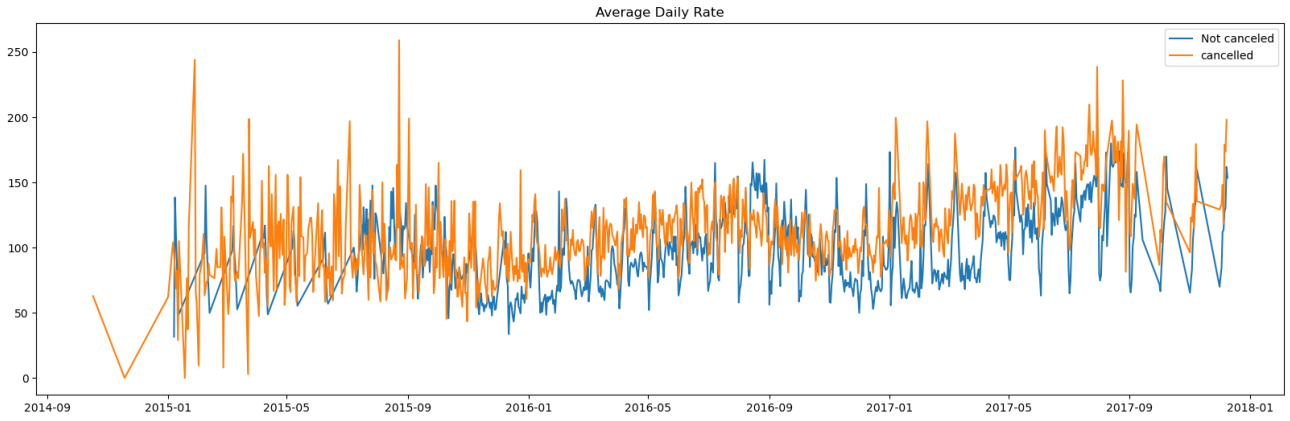
Out[26]:
```
Online TA        0.469696
Groups           0.273985
Offline TA/TO    0.187466
Direct           0.043486
Corporate        0.022151
Complementary    0.002038
Aviation         0.001178
Name: market_segment, dtype: float64
```

In [27]:
```
cancelled_adr = cancelled_data.groupby('reservation_status_date')[['adr']].mean()
cancelled_adr.reset_index(inplace = True)
cancelled_adr.sort_values('reservation_status_date', inplace = True)

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

plt.figure(figsize = (20,6))
plt.title('Average Daily Rate')
plt.plot(not_cancelled_adr['reservation_status_date'],not_cancelled_adr['adr'], label = 'Not canceled')
plt.plot(cancelled_adr['reservation_status_date'],cancelled_adr['adr'], label = 'cancelled')
plt.legend()
```

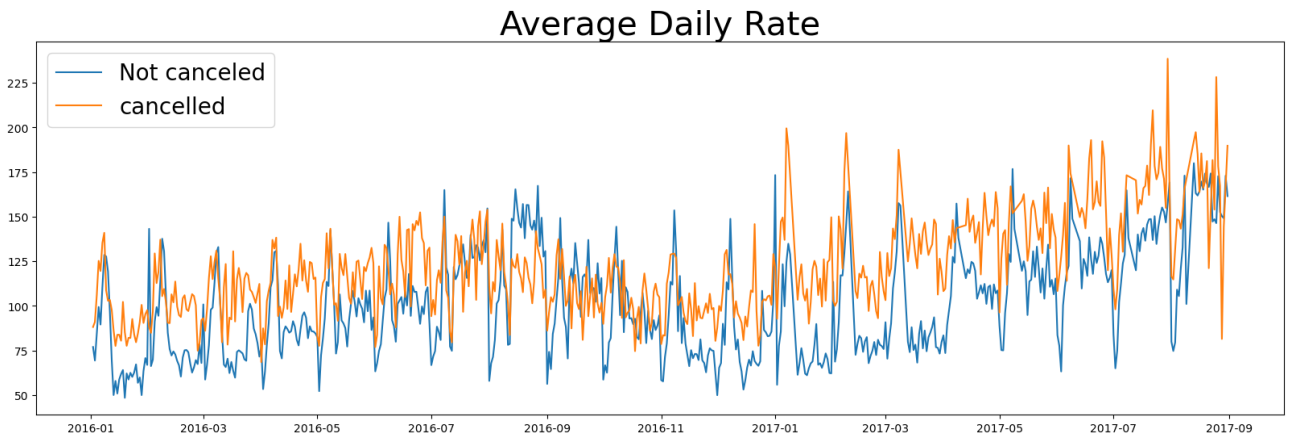
Out[27]:
```
<matplotlib.legend.Legend at 0x1c3c964f5e0>
```



In [28]:
```
cancelled_adr = cancelled_adr[(cancelled_adr['reservation_status_date'] > '2016') & (cancelled_adr['reservation_s
not_cancelled_adr = not_cancelled_adr[(not_cancelled_adr['reservation_status_date'] > '2016') & (not_cancelled_ad
```

In [29]:
```
plt.figure(figsize = (20,6))
plt.title('Average Daily Rate', fontsize = 30)
plt.plot(not_cancelled_adr['reservation_status_date'],not_cancelled_adr['adr'], label = 'Not canceled')
plt.plot(cancelled_adr['reservation_status_date'],cancelled_adr['adr'], label = 'cancelled')
plt.legend(fontsize = 20)
```

Out[29]: <matplotlib.legend.Legend at 0x1c3ccb20fa0>



In [ ]: