IMPORTING LIBRARIES

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import warnings
import matplotlib.pyplot as plt
warnings.filterwarnings('ignore')
```

LOADING THE DATASET

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

EXPLORATRY DATA ANALYSIS AND DATA CLEANING

In [5]:
```
df.head()
```

Out[5]:

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

5 rows × 32 columns

In [7]:
```
df.tail()
```

Out[7]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month
119385	City Hotel	0	23	2017	August
119386	City Hotel	0	102	2017	August
119387	City Hotel	0	34	2017	August
119388	City Hotel	0	109	2017	August
119389	City Hotel	0	205	2017	August

5 rows × 32 columns

In [9]:
```
df.shape
```

Out[9]:
```
(119390, 32)
```

In [11]:
```
df.columns
```

Out[11]:
```
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 [13]:
```
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 [15]:
```
df['reservation_status_date'] = pd.to_datetime(df['reservation_status_date']
```

In [17]:
```
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 [19]:
```
df.describe(include = 'object')
```

Out[19]:

	hotel	arrival_date_month	meal	country	market_segment	distri
count	119390	119390	119390	118902	119390	
unique	2	12	5	177	8	
top	City Hotel	August	BB	PRT	Online TA	
freq	79330	13877	92310	48590	56477	

In [21]:
```
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']
```

In [23]:
```
df.isnull().sum()
```

Out[23]:
```
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 [25]:
```
df.drop(['company', 'agent'],axis = 1,inplace = True)
df.dropna(inplace = True)
```

In [27]:
```
df.isnull().sum()
```

Out[27]:
```
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 [29]:
```
df.describe()
```

Out[29]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_nur
count	118898.000000	118898.000000	118898.000000	118898.00
mean	0.371352	104.311435	2016.157656	27.16
min	0.000000	0.000000	2015.000000	1.00
25%	0.000000	18.000000	2016.000000	16.00
50%	0.000000	69.000000	2016.000000	28.00
75%	1.000000	161.000000	2017.000000	38.00
max	1.000000	737.000000	2017.000000	53.00
std	0.483168	106.903309	0.707459	13.58

In [31]:
```
print("Missing values in each column:")
print(df.isnull().sum())
```

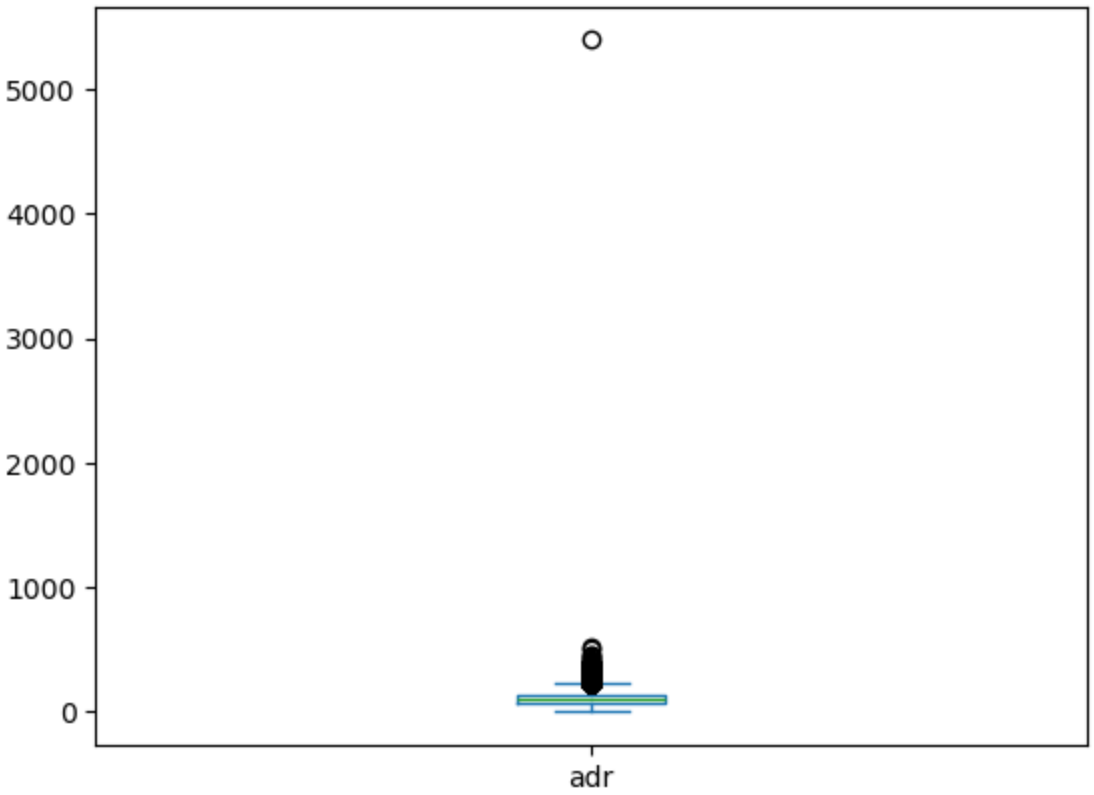
```
Missing values in each column:
hotel                             0
is_canceled                       0
lead_time                         0
arrival_date_year                 0
arrival_date_month                0
arrival_date_week_number          0
arrival_date_day_of_month         0
stays_in_weekend_nights           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 [33]:
```
duplicates = df.duplicated().sum()
if duplicates > 0:
    print(f"Found {duplicates} duplicate rows. Dropping them.")
    df = df.drop_duplicates()
```

```
Found 31984 duplicate rows. Dropping them.
```

In [35]:
```
df['adr'].plot(kind = 'box')
```

Out[35]:
```
<Axes: >
```



In [37]:
```
df = df[df['adr']<5000]
```

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

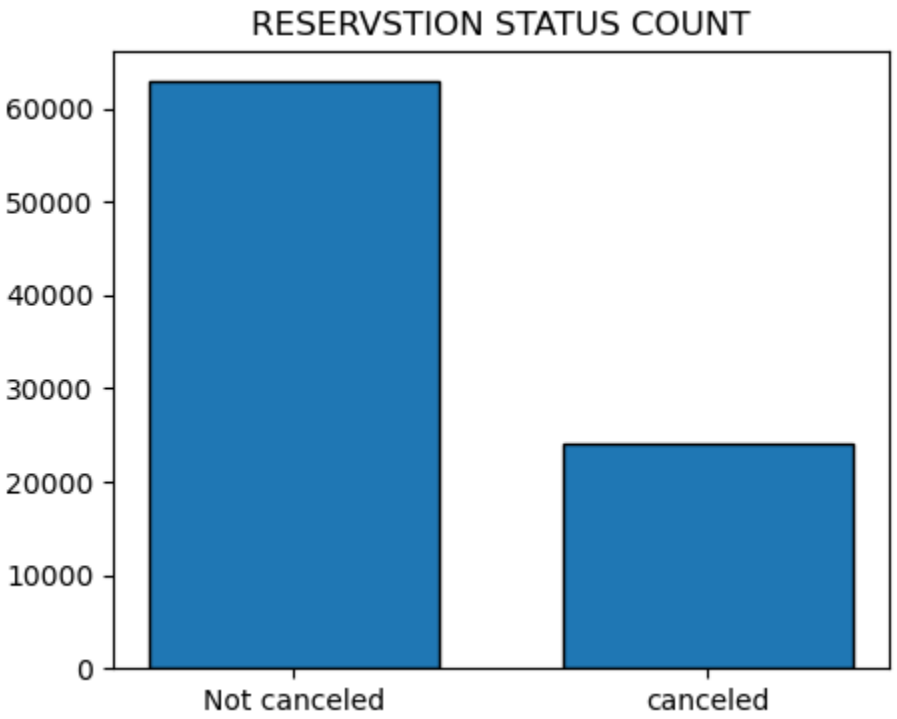
Out[39]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_numb
count	86913.000000	86913.000000	86913.000000	86913.0000
mean	0.275931	80.203261	2016.211844	26.8418
min	0.000000	0.000000	2015.000000	1.0000
25%	0.000000	12.000000	2016.000000	16.0000
50%	0.000000	50.000000	2016.000000	27.0000
75%	1.000000	125.000000	2017.000000	37.0000
max	1.000000	737.000000	2017.000000	53.0000
std	0.446985	86.103261	0.685992	13.6541

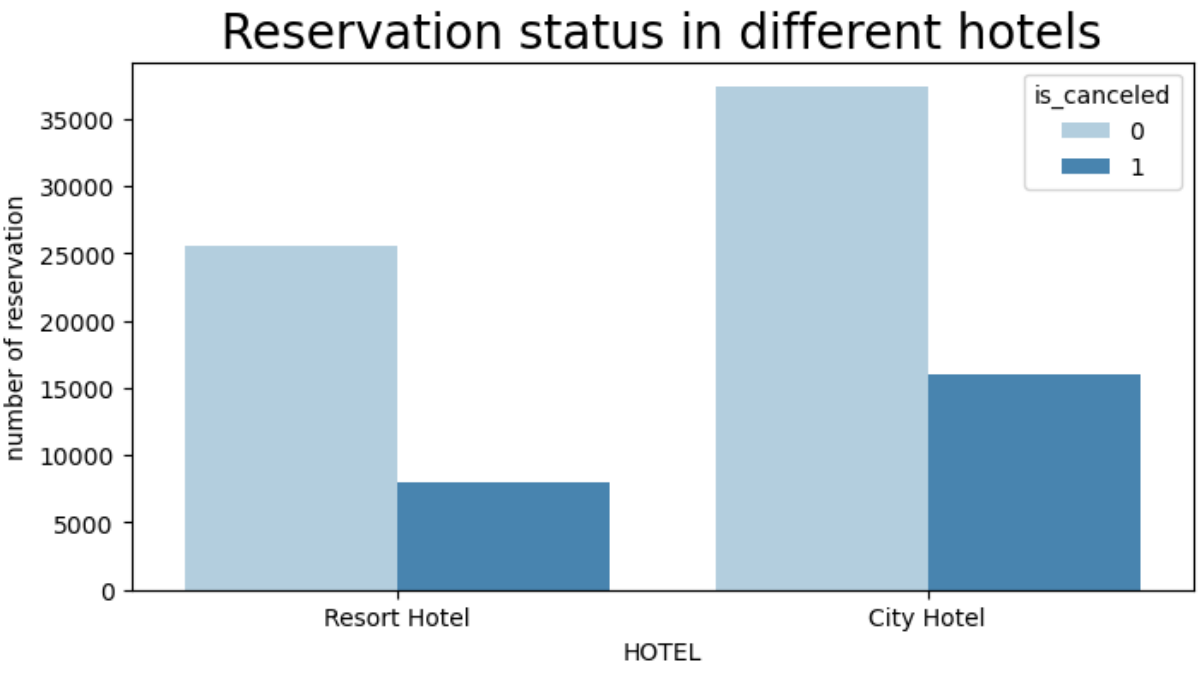
DATA ANALYSIS AND VISULIZATIONS

In [41]:
```
cancelled_perc = df['is_canceled'].value_counts(normalize = True)
print(cancelled_perc)
plt.figure(figsize = (5,4))
plt.title('RESERVSTION STATUS COUNT')
plt.bar(['Not canceled', 'canceled'],df['is_canceled'].value_counts(), edgec
plt.show()
```

```
is_canceled
0    0.724069
1    0.275931
Name: proportion, dtype: float64
```



In [43]:
```
plt.figure(figsize = (8,4))
ax1= sns.countplot(x= 'hotel' ,hue = 'is_canceled' , data = df, palette = 'B
legend_labels,_=ax1. get_legend_handles_labels()
plt.title('Reservation status in different hotels', size = 20)
plt.xlabel('HOTEL')
plt.ylabel('number of reservation')
plt.show()
```



In [45]:
```
resort_hotel = df[df['hotel'] =='Resort Hotel']
resort_hotel['is_canceled'].value_counts(normalize = True)
```

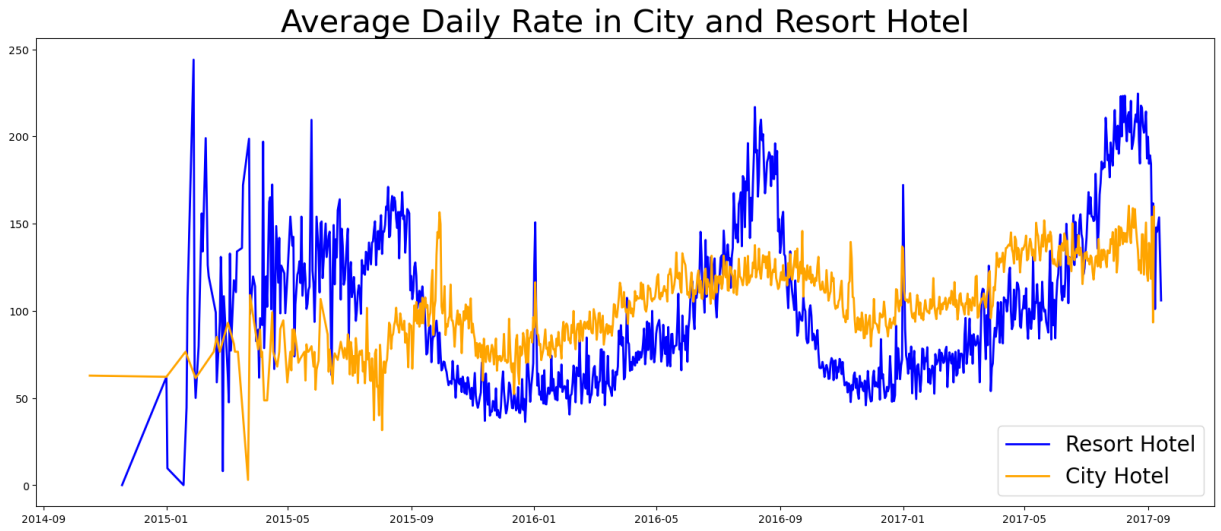
Out[45]:
```
is_canceled
0    0.762936
1    0.237064
Name: proportion, dtype: float64
```

In [47]:
```
city_hotel = df[df['hotel'] =='City Hotel']
city_hotel['is_canceled'].value_counts(normalize = True)
```

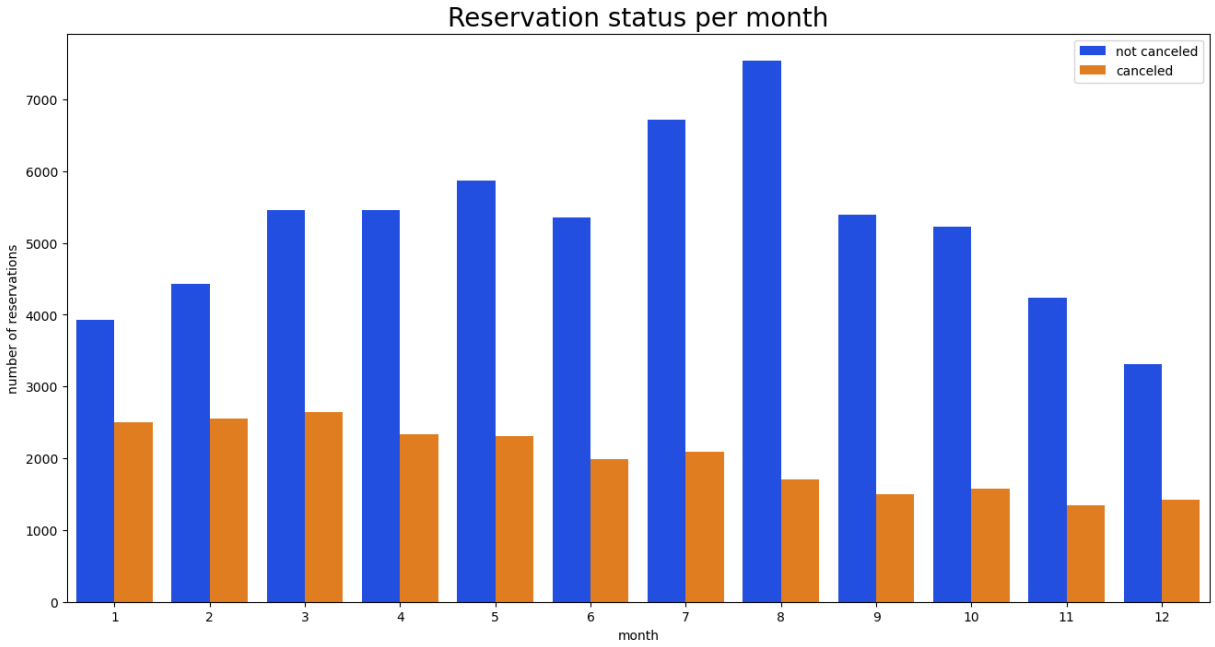
Out[47]:
```
is_canceled
0    0.69968
1    0.30032
Name: proportion, dtype: float64
```

In [49]:
```
resort_hotel = resort_hotel.groupby('reservation_status_date')[['adr']].mear
city_hotel = city_hotel.groupby('reservation_status_date')[['adr']].mean()
```

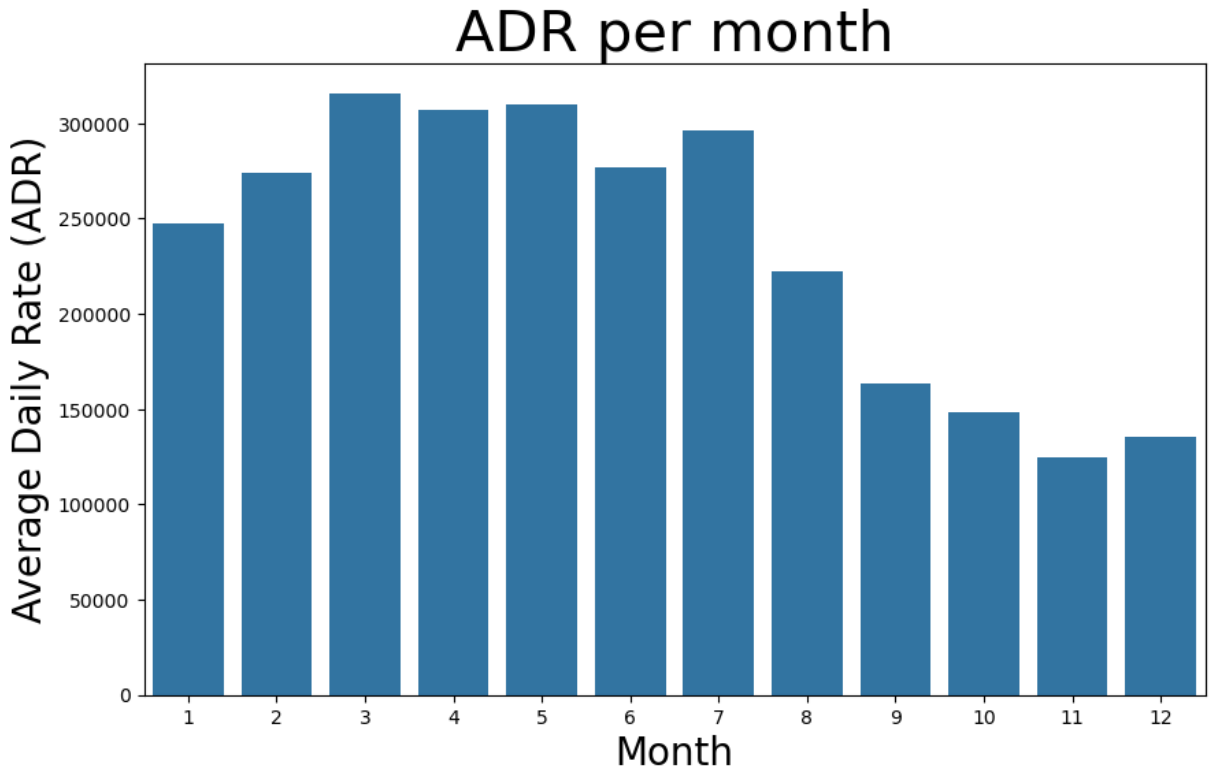
In [51]:
```
plt.figure(figsize=(20, 8))
plt.title('Average Daily Rate in City and Resort Hotel', fontsize=30)
plt.plot(resort_hotel.index, resort_hotel['adr'], label='Resort Hotel', colo
plt.plot(city_hotel.index, city_hotel['adr'], label='City Hotel', color='ora
plt.legend(fontsize=20)
plt.show()
```



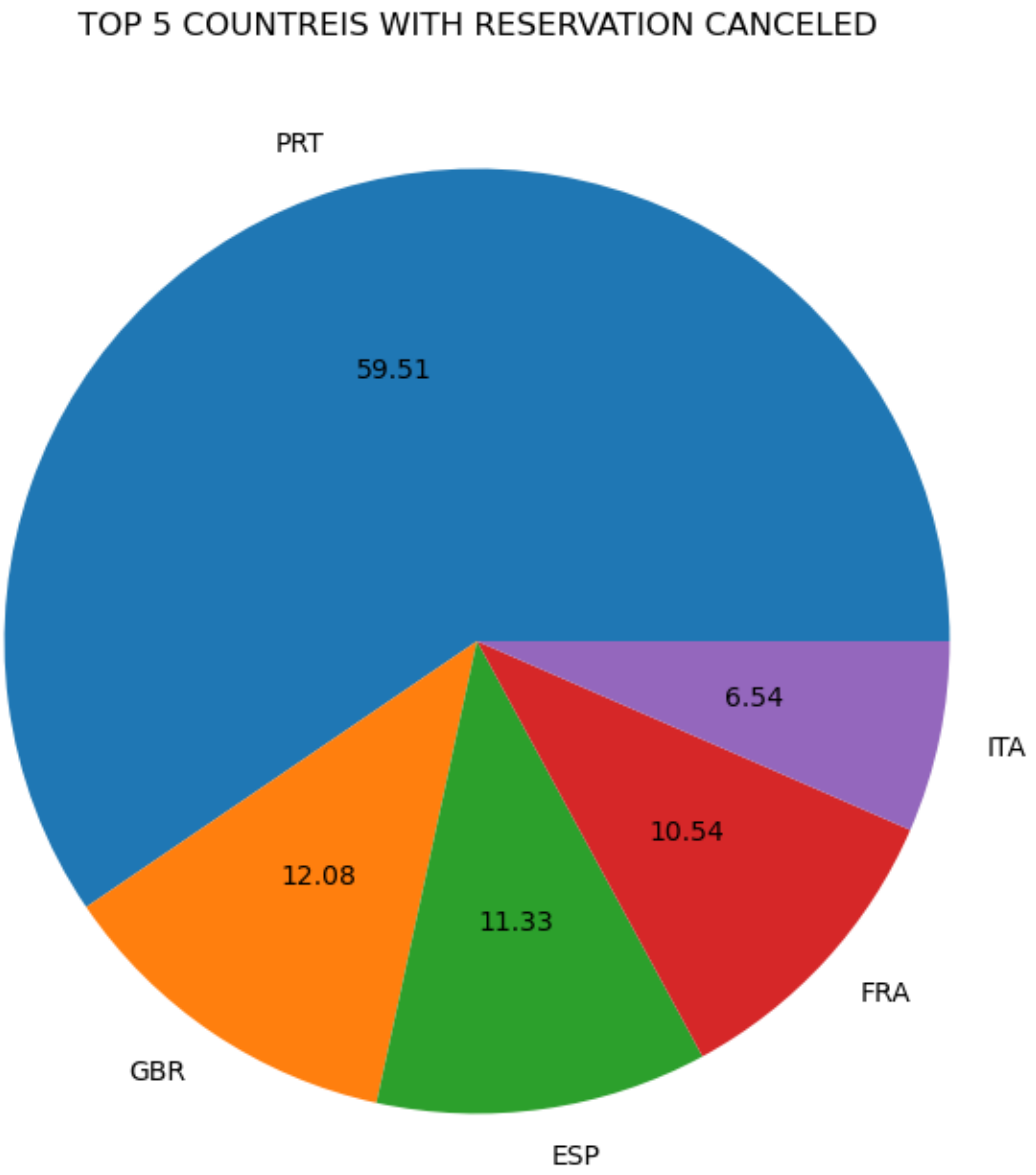
In [53]:
```
df['month'] = df['reservation_status_date'].dt.month
plt.figure(figsize = (16,8))
ax1 = sns.countplot(x = 'month',hue = 'is_canceled',data = df, palette = 'br
legend_labels,_ =ax1. get_legend_handles_labels()
ax1.legend(bbox_to_anchor=(1,1))
plt.title('Reservation status per month',size = 20)
plt.xlabel('month')
plt.ylabel('number of reservations')
plt.legend(['not canceled', 'canceled'])
plt.show()
```



In [55]:
```
plt.figure(figsize=(10, 6))  # Optional: Set the figure size
plt.title('ADR per month', fontsize=30)


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

plt.xlabel('Month', fontsize=20)  # Optional: Set x-axis label
plt.ylabel('Average Daily Rate (ADR)', fontsize=20)  # Optional: Set y-axis

plt.show()
```



In [57]:
```
cancelled_data = df[df['is_canceled'] == 1]
top_5_country = cancelled_data['country'].value_counts()[:5]
plt.figure(figsize = (8,8))
plt.title('TOP 5 COUNTREIS WITH RESERVATION CANCELED')
plt.pie(top_5_country,autopct = '%.2f',labels = top_5_country.index)
plt.show()
```



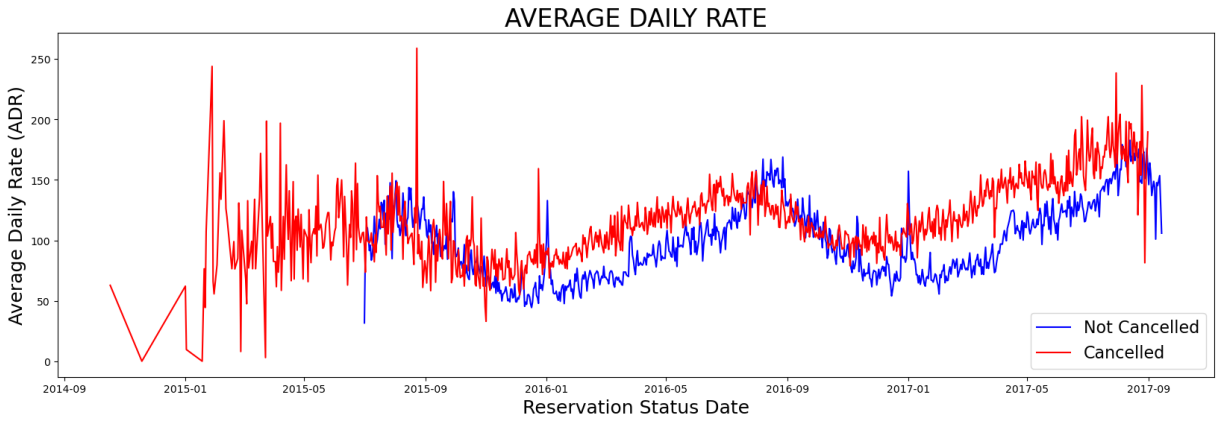
In [59]:
```
df['market_segment'].value_counts()
```

Out[59]:
```
market_segment
Online TA        51534
Offline TA/TO    13848
Direct           11645
Groups            4936
Corporate         4025
Complementary      698
Aviation           227
Name: count, dtype: int64
```

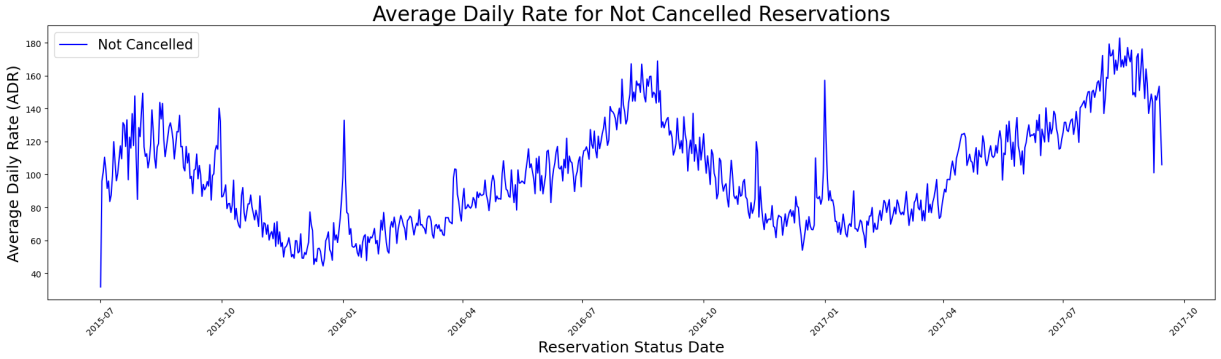
In [61]:
```
cancelled_data['market_segment'].value_counts(normalize = True)
```

Out[61]:
```
market_segment
Online TA        0.760696
Offline TA/TO    0.085606
Direct           0.071846
Groups           0.055667
Corporate        0.020724
Complementary    0.003586
Aviation         0.001876
Name: proportion, dtype: float64
```

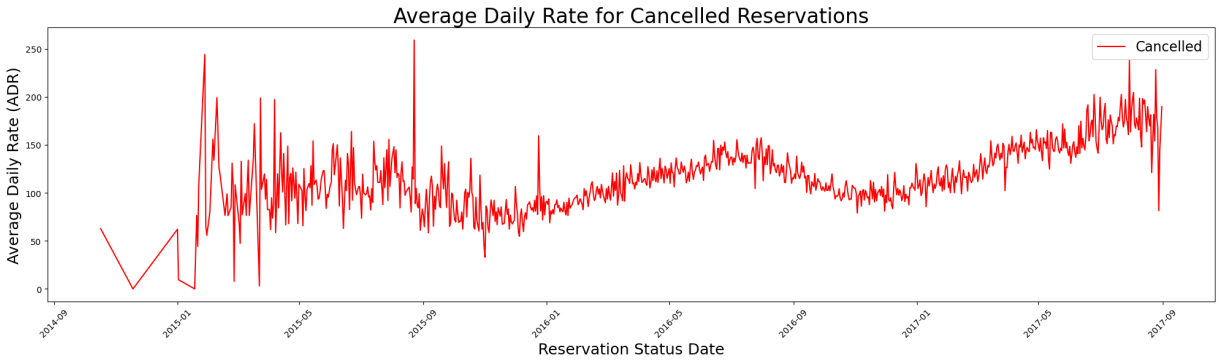
In [63]:
```
cancelled_df_adr = cancelled_data.groupby('reservation_status_date')[['adr']
cancelled_df_adr.reset_index(inplace=True)
cancelled_df_adr.sort_values('reservation_status_date', inplace=True)


not_cancelled_data = df[df['is_canceled'] == 0]
not_cancelled_df_adr = not_cancelled_data.groupby('reservation_status_date')
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', fontsize=24)
plt.plot(not_cancelled_df_adr['reservation_status_date'], not_cancelled_df_a
plt.plot(cancelled_df_adr['reservation_status_date'], cancelled_df_adr['adr'
plt.xlabel('Reservation Status Date', fontsize=18)
plt.ylabel('Average Daily Rate (ADR)', fontsize=18)
plt.legend(fontsize=16)

plt.show()
```



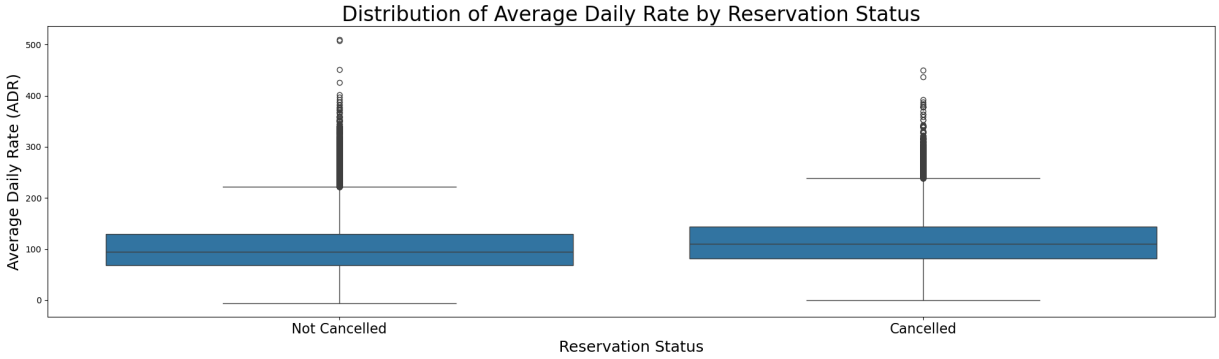
In [65]:
```
plt.figure(figsize=(20, 6))
plt.title('Average Daily Rate for Not Cancelled Reservations', fontsize=24)
plt.plot(not_cancelled_df_adr['reservation_status_date'], not_cancelled_df_a
plt.xlabel('Reservation Status Date', fontsize=18)
plt.ylabel('Average Daily Rate (ADR)', fontsize=18)
plt.legend(fontsize=16)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



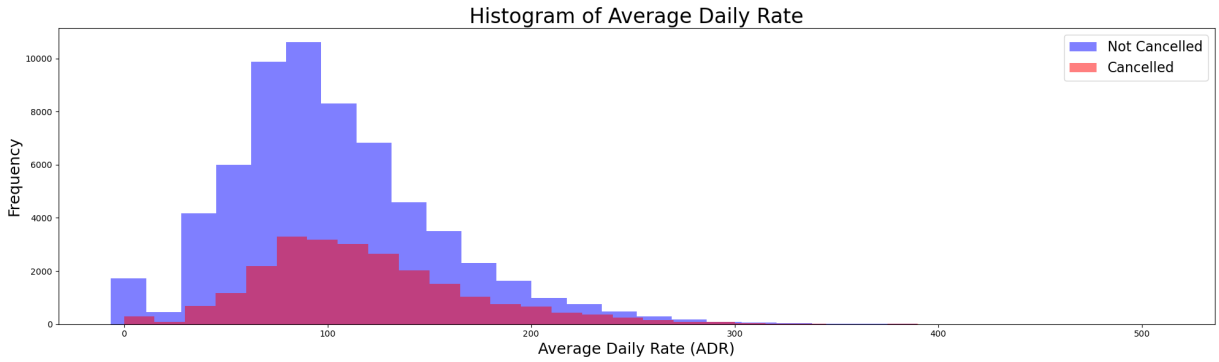
In [67]:
```
plt.figure(figsize=(20, 6))
plt.title('Average Daily Rate for Cancelled Reservations', fontsize=24)
plt.plot(cancelled_df_adr['reservation_status_date'], cancelled_df_adr['adr'
plt.xlabel('Reservation Status Date', fontsize=18)
plt.ylabel('Average Daily Rate (ADR)', fontsize=18)
plt.legend(fontsize=16)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



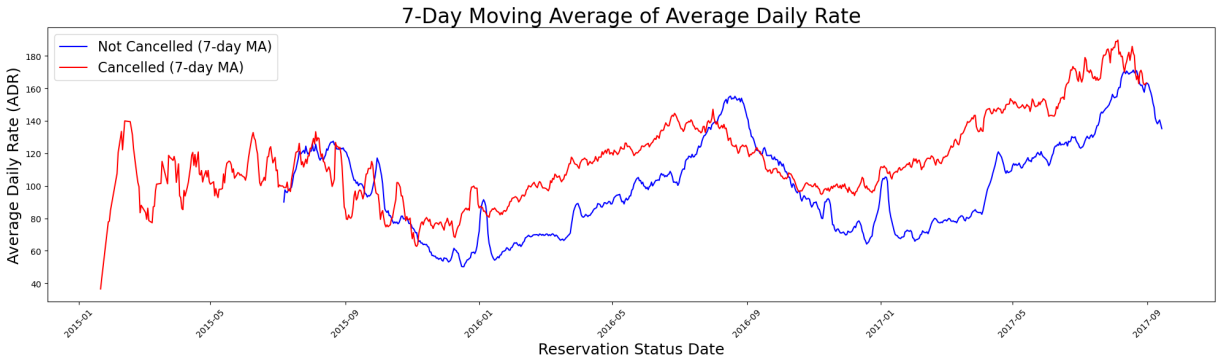
In [69]:
```
not_cancelled_data['Reservation Status'] = 'Not Cancelled'
cancelled_data['Reservation Status'] = 'Cancelled'

# Combine the data
combined_data = pd.concat([not_cancelled_data[['reservation_status_date', 'a
                           cancelled_data[['reservation_status_date', 'adr'

# Box Plot
plt.figure(figsize=(20, 6))
sns.boxplot(data=combined_data, x='Reservation Status', y='adr')
plt.title('Distribution of Average Daily Rate by Reservation Status', fontsi
plt.xlabel('Reservation Status', fontsize=18)
plt.ylabel('Average Daily Rate (ADR)', fontsize=18)
plt.xticks(fontsize=16)
plt.tight_layout()
plt.show()
```



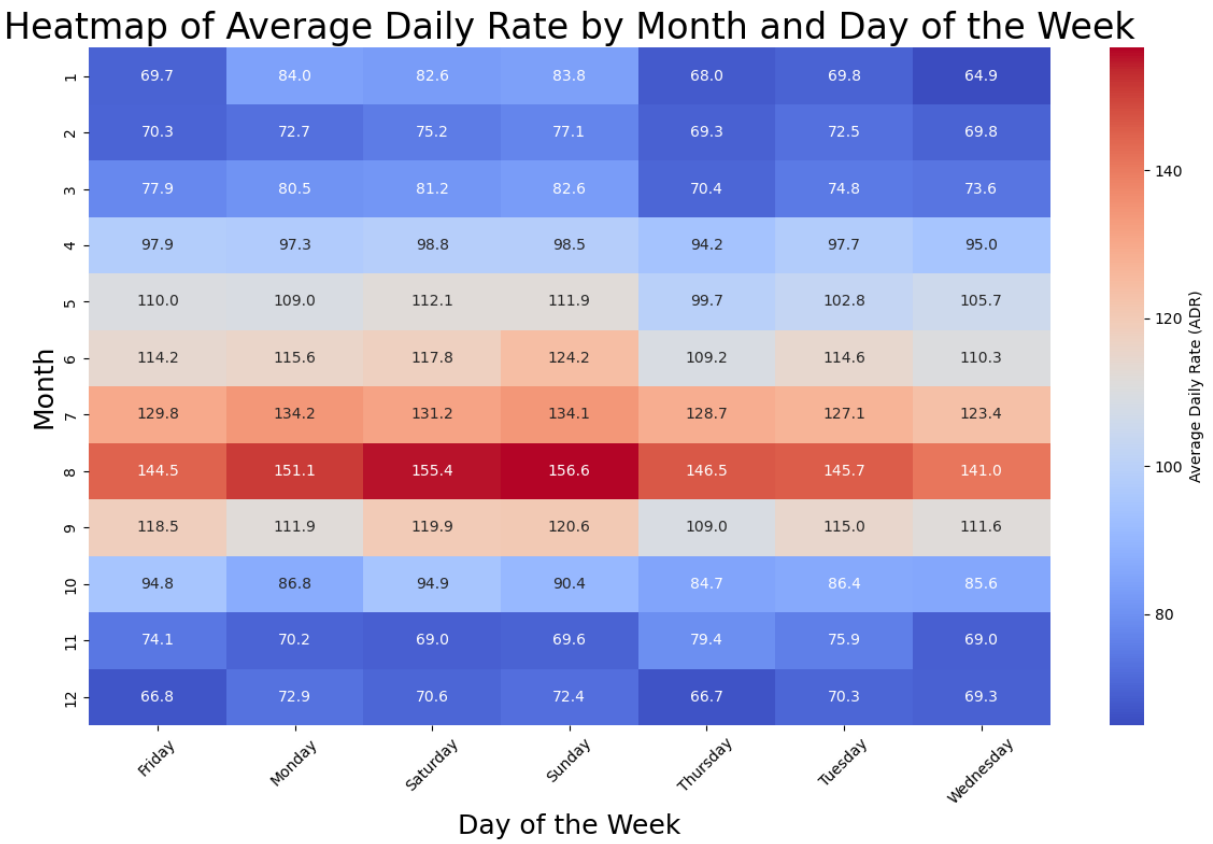
In [71]:
```
plt.figure(figsize=(20, 6))
plt.hist(not_cancelled_data['adr'], bins=30, alpha=0.5, label='Not Cancelled
plt.hist(cancelled_data['adr'], bins=30, alpha=0.5, label='Cancelled', color
plt.title('Histogram of Average Daily Rate', fontsize=24)
plt.xlabel('Average Daily Rate (ADR)', fontsize=18)
plt.ylabel('Frequency', fontsize=18)
plt.legend(fontsize=16)
plt.tight_layout()
plt.show()
```



In [73]:
```
# Calculate moving average
not_cancelled_df_adr['Moving Average'] = not_cancelled_df_adr['adr'].rolling
cancelled_df_adr['Moving Average'] = cancelled_df_adr['adr'].rolling(window=

plt.figure(figsize=(20, 6))
plt.plot(not_cancelled_df_adr['reservation_status_date'], not_cancelled_df_a
plt.plot(cancelled_df_adr['reservation_status_date'], cancelled_df_adr['Movi
plt.title('7-Day Moving Average of Average Daily Rate', fontsize=24)
plt.xlabel('Reservation Status Date', fontsize=18)
plt.ylabel('Average Daily Rate (ADR)', fontsize=18)
plt.legend(fontsize=16)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

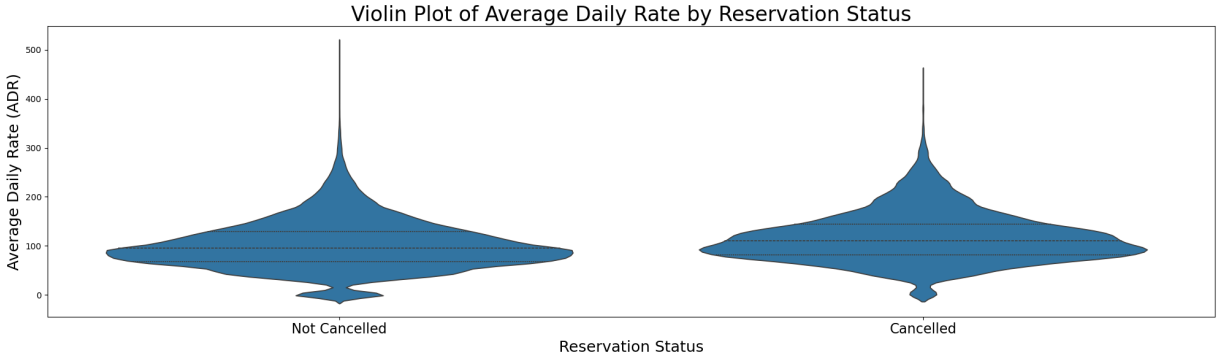


In [75]:
```
not_cancelled_data['month'] = not_cancelled_data['reservation_status_date'].
not_cancelled_data['day_of_week'] = not_cancelled_data['reservation_status_c

# Group by month and day of the week
heatmap_data = not_cancelled_data.groupby(['month', 'day_of_week'])['adr'].m

# Heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(heatmap_data, annot=True, fmt=".1f", cmap='coolwarm', cbar_kws={
plt.title('Heatmap of Average Daily Rate by Month and Day of the Week', font
plt.xlabel('Day of the Week', fontsize=18)
plt.ylabel('Month', fontsize=18)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

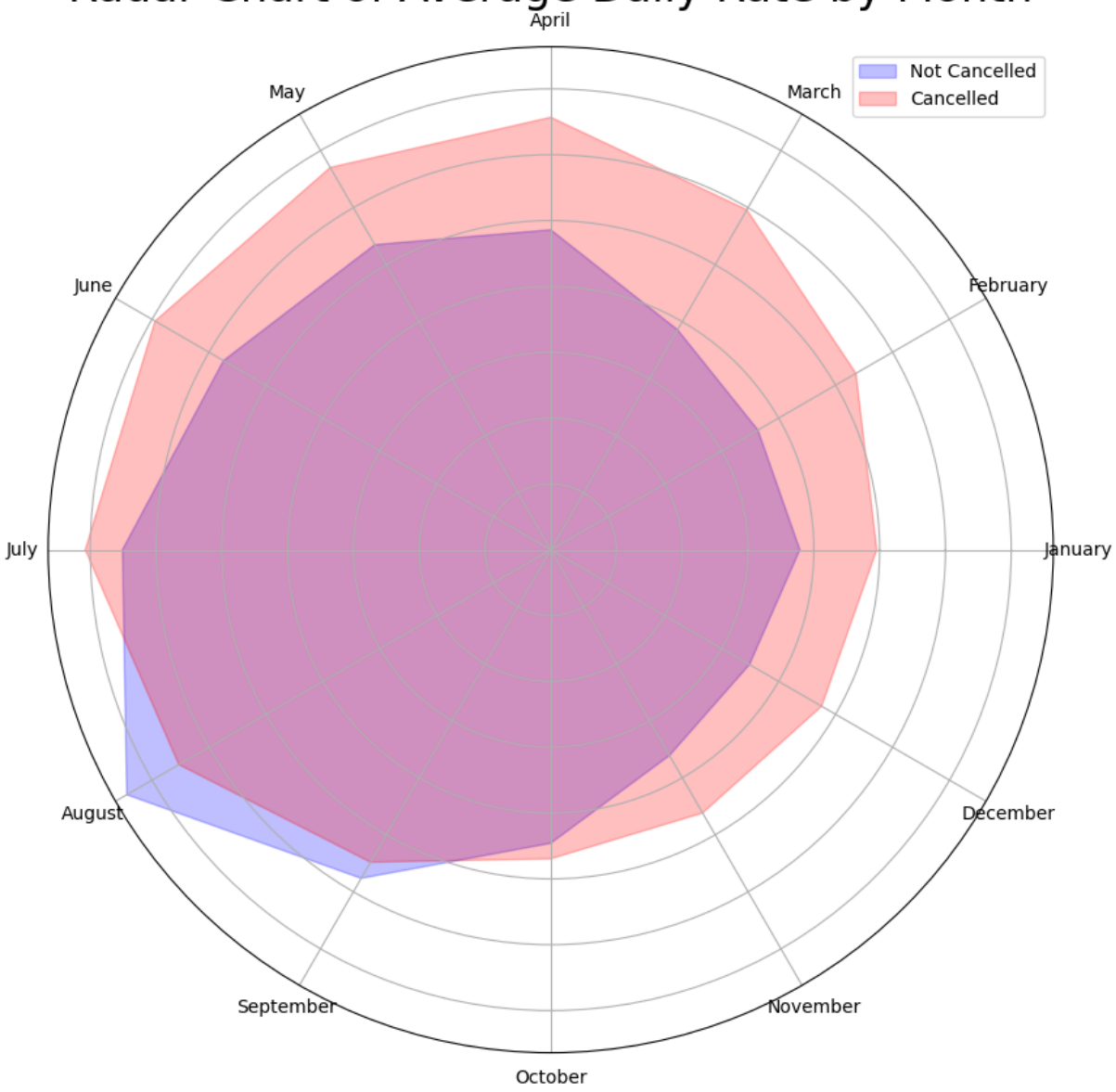


In [77]:
```
plt.figure(figsize=(20, 6))
sns.violinplot(data=combined_data, x='Reservation Status', y='adr', inner='c
```

```
plt.title('Violin Plot of Average Daily Rate by Reservation Status', fontsiz
plt.xlabel('Reservation Status', fontsize=18)
plt.ylabel('Average Daily Rate (ADR)', fontsize=18)
plt.xticks(fontsize=16)
plt.tight_layout()
plt.show()
```



In [79]:
```
labels = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'A
not_cancelled_avg = [not_cancelled_data[not_cancelled_data['month'] == i]['a
cancelled_avg = [cancelled_data[cancelled_data['month'] == i]['adr'].mean()

# radar chart
angles = np.linspace(0, 2 * np.pi, len(labels), endpoint=False).tolist()
not_cancelled_avg += not_cancelled_avg[:1]  # Repeat the first value to clos
cancelled_avg += cancelled_avg[:1]  # Repeat the first value to close the ci
angles += angles[:1]  # Repeat the first angle to close the circle

plt.figure(figsize=(10, 10))
ax = plt.subplot(111, polar=True)
ax.fill(angles, not_cancelled_avg, color='blue', alpha=0.25)
ax.fill(angles, cancelled_avg, color='red', alpha=0.25)
ax.set_yticklabels([])
ax.set_xticks(angles[:-1])
ax.set_xticklabels(labels)
plt.title('Radar Chart of Average Daily Rate by Month', fontsize=24)
plt.legend(['Not Cancelled', 'Cancelled'], loc='upper right')
plt.show()
```



In [81]:
```
labels = ['January', 'February', 'March', 'April', 'May', 'June',
          'July', 'August', 'September', 'October', 'November', 'December']

# Calculate average daily rates for each month
not_cancelled_avg = [not_cancelled_data[not_cancelled_data['month'] == i]['a
cancelled_avg = [cancelled_data[cancelled_data['month'] == i]['adr'].mean()

# Create a scatter plot
plt.figure(figsize=(12, 6))

# Scatter plot for not canceled
plt.scatter(labels, not_cancelled_avg, color='blue', label='Not Canceled', s

# Scatter plot for canceled
plt.scatter(labels, cancelled_avg, color='red', label='Canceled', s=100, alp

# Adding titles and labels
plt.title('Average Daily Rate by Month', fontsize=24)
plt.xlabel('Month', fontsize=16)
```

```
plt.ylabel('Average Daily Rate (ADR)', fontsize=16)
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```