Hotel Booking Project

Using Python

Project Steps

- Create a problem Statement.
- Identify the data you want to analyze.
- Explore and clean the data.
- Analyze the data to get useful insights.
- Present the data in terms of reports or dashboards using visualization.

1st install the libraries

```
#pip install pandas
#pip install matplotlib
#pip install seaborn
#pip install numpy
```

import the libraries after installation

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as num
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Loading the dataset

```
df= pd.read_csv('hotel_bookings 2.csv')
```

Exploratory Data Analysis And Data Cleaning

```
# how data set looks like for that we run df.head()
```

```
df.head()
```

	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_nigh
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

```
df.tail()
```

	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
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	

5 rows × 32 columns

```
# how many rows and columns are there in dataset its showing
```

```
df.shape
```

```
(119390, 32)
```

```
df.columns
```

```
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')
```

```
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

 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  object
```

```
df['reservation_status_date'] = pd.to_datetime(df['reservation_status_date'], errors="coerce", dayfirst=True)
```

```
df.info()
```

```
30  reservation_status              119390 non-null  object
31  reservation_status_date         119390 non-null  datetime64[ns]
```

```
# to see in categorical column for how many unique values/object type are there.
```

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

	hotel	arrival_date_month	meal	country	market_segment	distribution_channel	reserved_room_type	assigned_room_type	deposit_type	customer_type	rese
count	119390	119390	119390	118902	119390	119390	119390	119390	119390	119390	
unique	2	12	5	177	8	5	10	12	3	4	
top	City Hotel	August	BB	PRT	Online TA	TA/TO	A	A	No Deposit	Transient	
freq	79330	13877	92310	48590	56477	97870	85994	74053	104641	89613	

```
# to get object datatype column unique value('TA-travel agent, TO-tour operator')
```

```
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']
--------------------------------------------------
```

```
# to check missimg values in columns
```

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

```
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
```

```
# remove unwanted column like which is not in use, for remove column axis = 1,if u want to change in data frame we have pass this parameter(inplace
```

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

```
# whatever missing values are there its removed already
```

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

```
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
```

```
df.describe()
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	1
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	
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	
std	0.483168	106.903309	0.707459	13.589971	8.780324	0.996216	1.900168	

```
# remove adr(average daily rate) outlier which is highest price
```

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

```
df.describe()
```

ions	previous_bookings_not_canceled	booking_changes	days_in_waiting_list	adr
0000	118897.000000	118897.000000	118897.000000	118897.000000
7143	0.131635	0.221175	2.330774	101.958683
0000	0.000000	0.000000	0.000000	-6.380000
0000	0.000000	0.000000	0.000000	70.000000
0000	0.000000	0.000000	0.000000	95.000000
0000	0.000000	0.000000	0.000000	126.000000
0000	72.000000	21.000000	391.000000	510.000000
5872	1.484678	0.652784	17.630525	48.091199

Data analysis and Visualizations

```
# here we are getting percentage of 'is_cancelled' column(62% are not cancelled, but 37% reservation are cancelled)
```

```
cancelled_perc = df['is_canceled'].value_counts(normalize=True)
cancelled_perc
plt.figure(figsize = (4,4))
plt.title('Reservation Status Count')
plt.bar(['Not canceled','Canceled'],df['is_canceled'].value_counts())
plt.show()
```

```
plt.figure(figsize=(8, 5))  # Use a tuple with width and height
ax1 = sns.countplot(x='hotel', hue='is_canceled', data=df, palette='Blues')
legend_labels, _ = ax1.get_legend_handles_labels()
ax1.legend(bbox_to_anchor=(1, 1), labels=['not_canceled', 'canceled'])  # Set custom labels here
plt.title('Reservation status in different hotels', size=20)
plt.xlabel('hotel')
plt.ylabel('Number of Reservations')

plt.show()
```

```
# resort hotel cancelled bookings = 28%, not canceled  = 72%
```

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

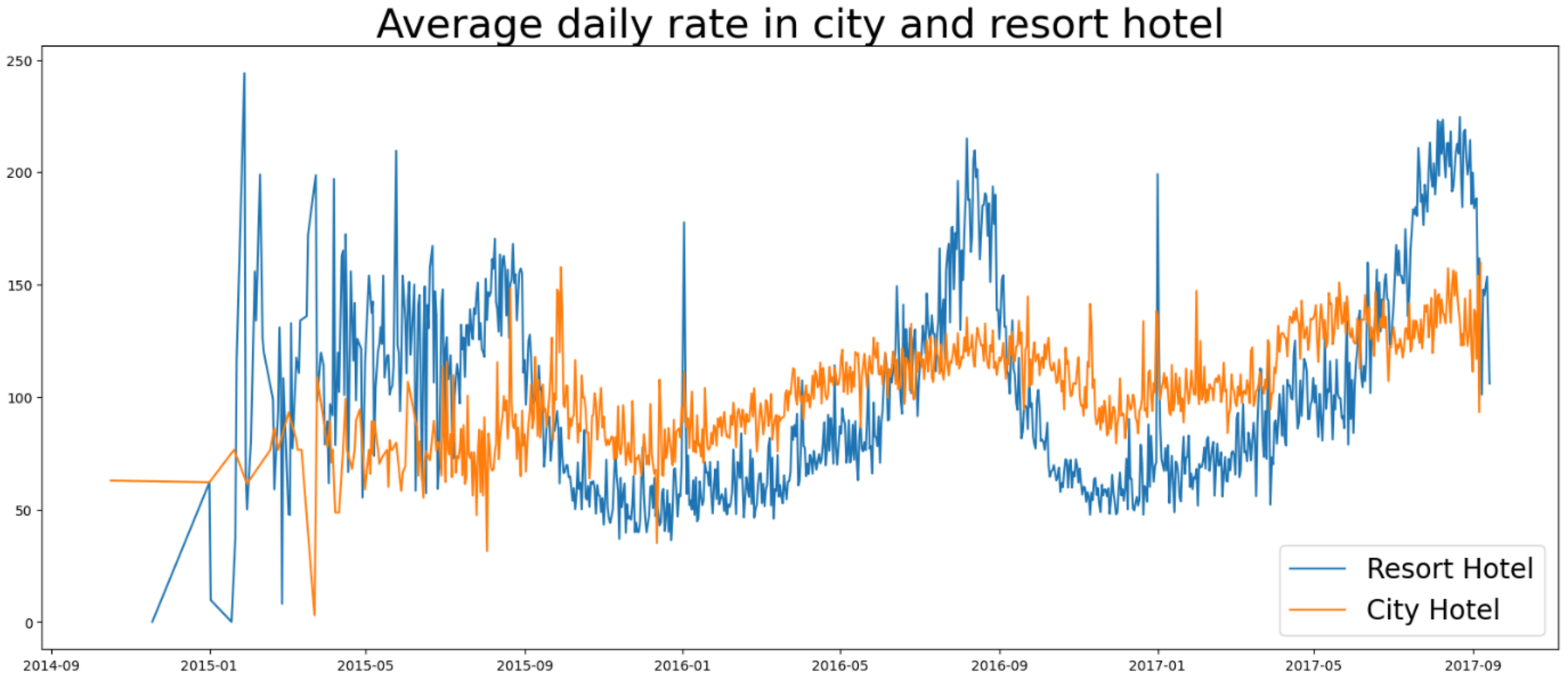
```
is_canceled
0    0.72025
1    0.27975
Name: proportion, dtype: float64
```

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

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

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

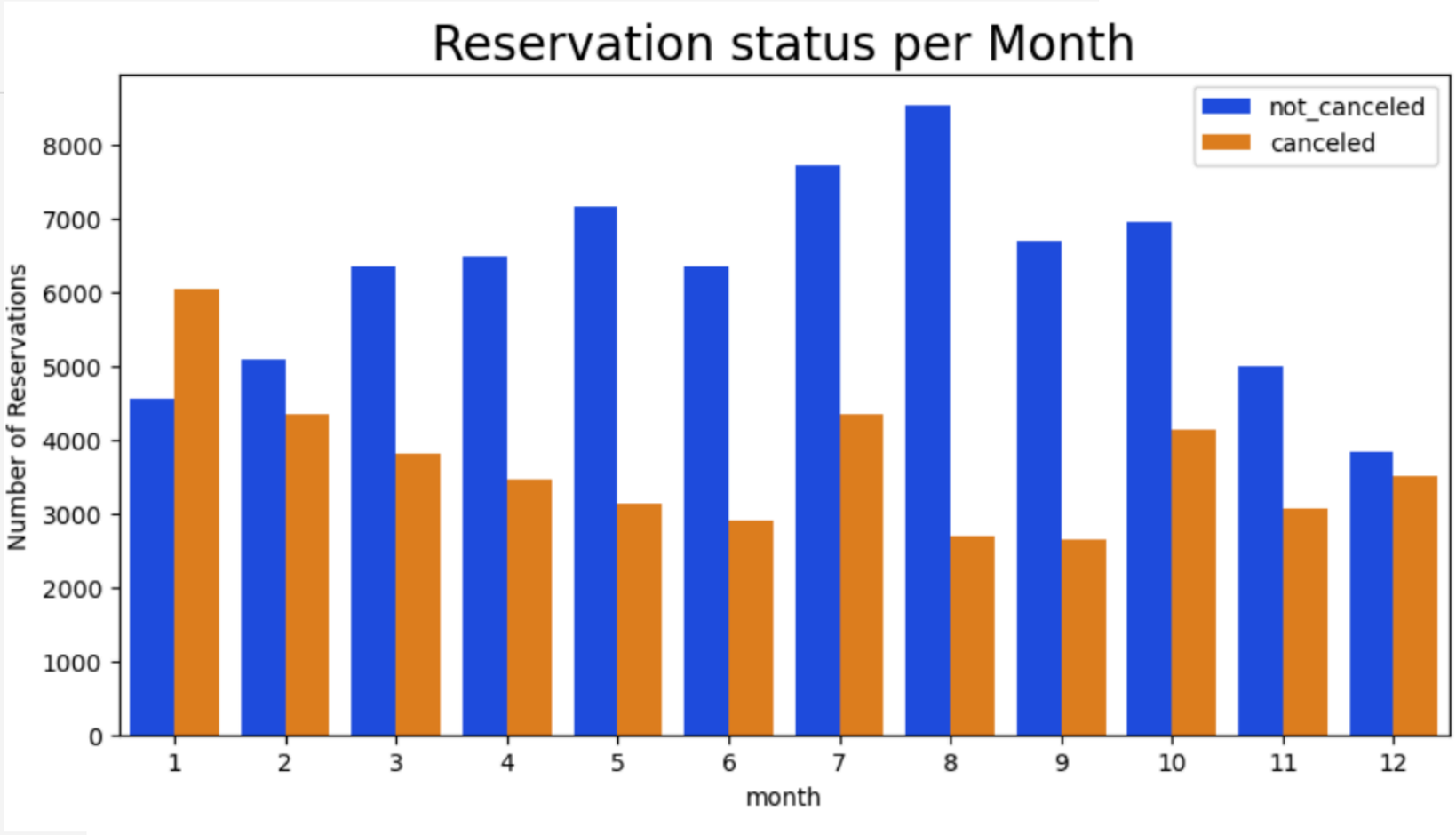
```
plt.figure(figsize = (20, 8))
plt.title('Average daily rate in city and resort hotel', fontsize = 30)
plt.plot(resort_hotel.index, resort_hotel['adr'], label = 'Resort Hotel')
plt.plot(city_hotel.index, city_hotel['adr'], label = 'City Hotel')
plt.legend(fontsize = 20)
plt.show()
```



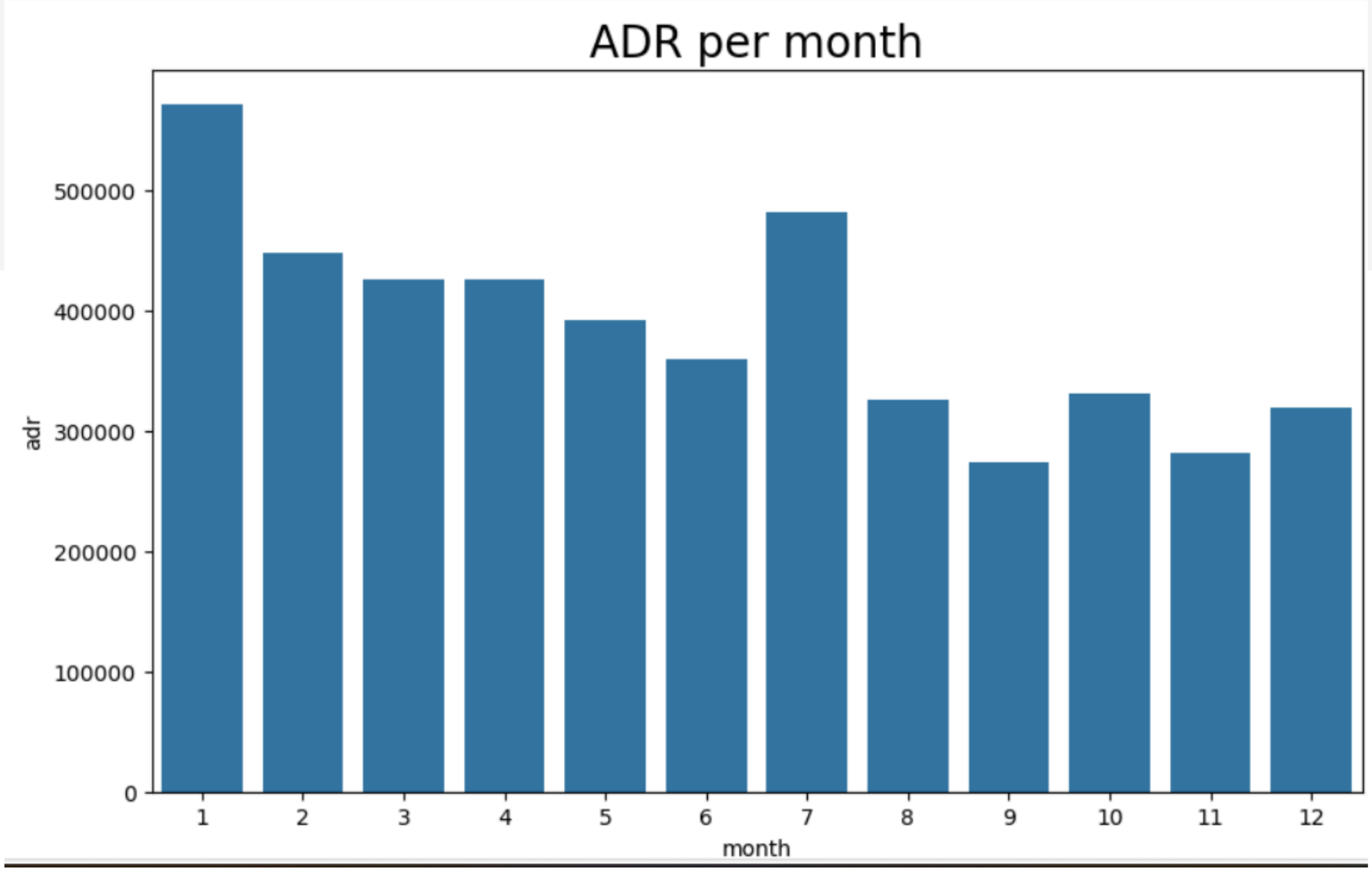
```python
df['month'] = df['reservation_status_date'].dt.month
plt.figure(figsize=(10, 5))  # Use a tuple with width and height
ax1= sns.countplot(x='month', hue='is_canceled', data=df, palette='bright')  # Assign countplot to 'ax'
legend_labels, _ = ax1.get_legend_handles_labels()
ax1.legend(bbox_to_anchor=(1, 1), labels=['not_canceled', 'canceled'])  # Set custom labels here

plt.title('Reservation status per Month', size=20)
plt.xlabel('month')
plt.ylabel('Number of Reservations')

plt.show()
```

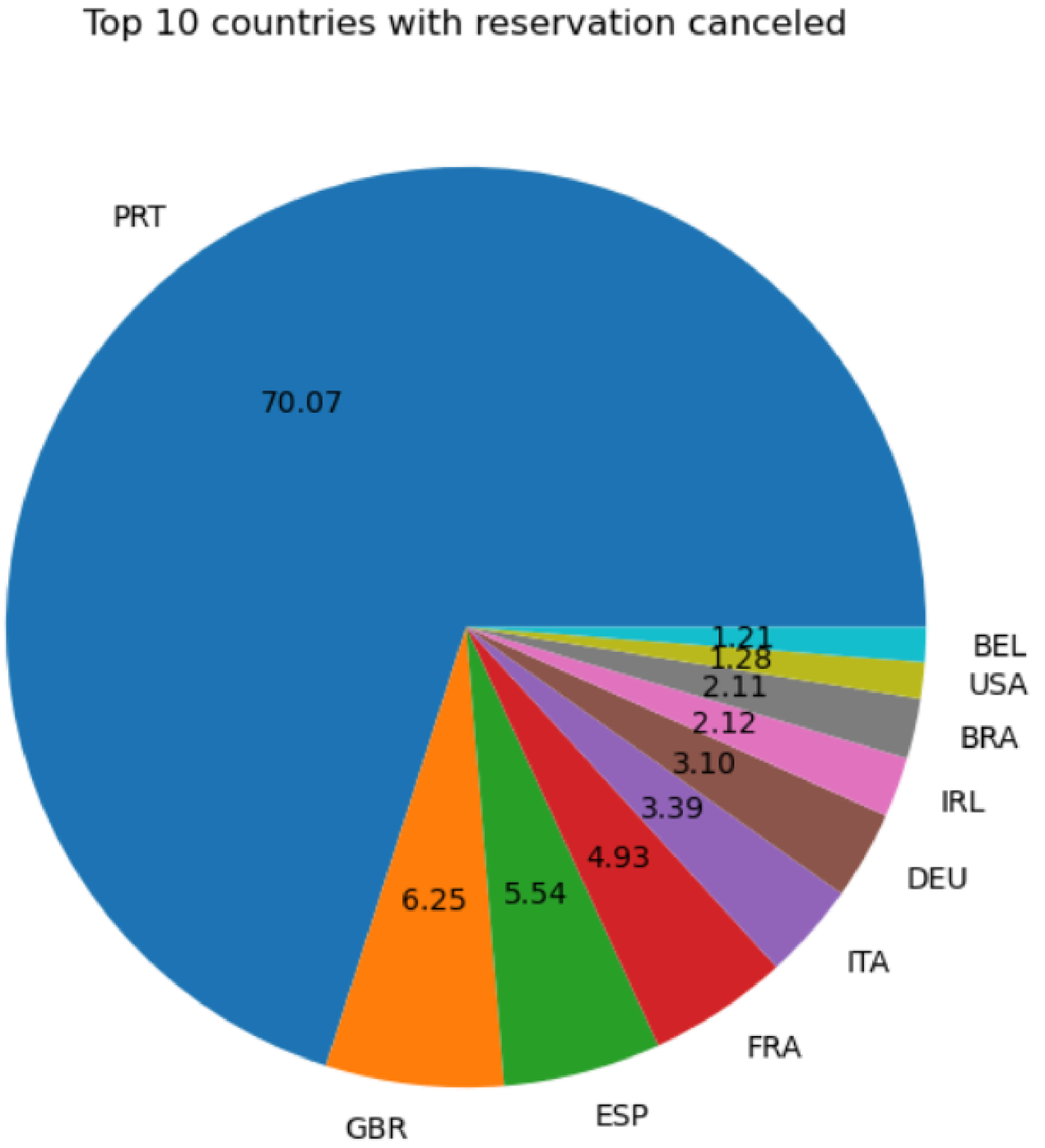


```
plt.figure(figsize = (10, 6))
plt.title('ADR per month', fontsize = 20)
sns.barplot(x = 'month',y = 'adr' , data = df[df['is_canceled'] == 1].groupby('month')[['adr']].sum().reset_index())
plt.show()
```



```
# top 10  country canceled data
```

```
cancelled_data = df[df['is_canceled'] == 1]
top_10_country = cancelled_data['country'].value_counts()[:10]
plt.figure(figsize = (7, 7))
plt.title('Top 10 countries with reservation canceled')
plt.pie(top_10_country, autopct = '%.2f', labels = top_10_country.index)
plt.show()
```



```
# from where people are booking the ticket most like online /offline
```

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

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

```
# to see percentage, online booking 47%
```

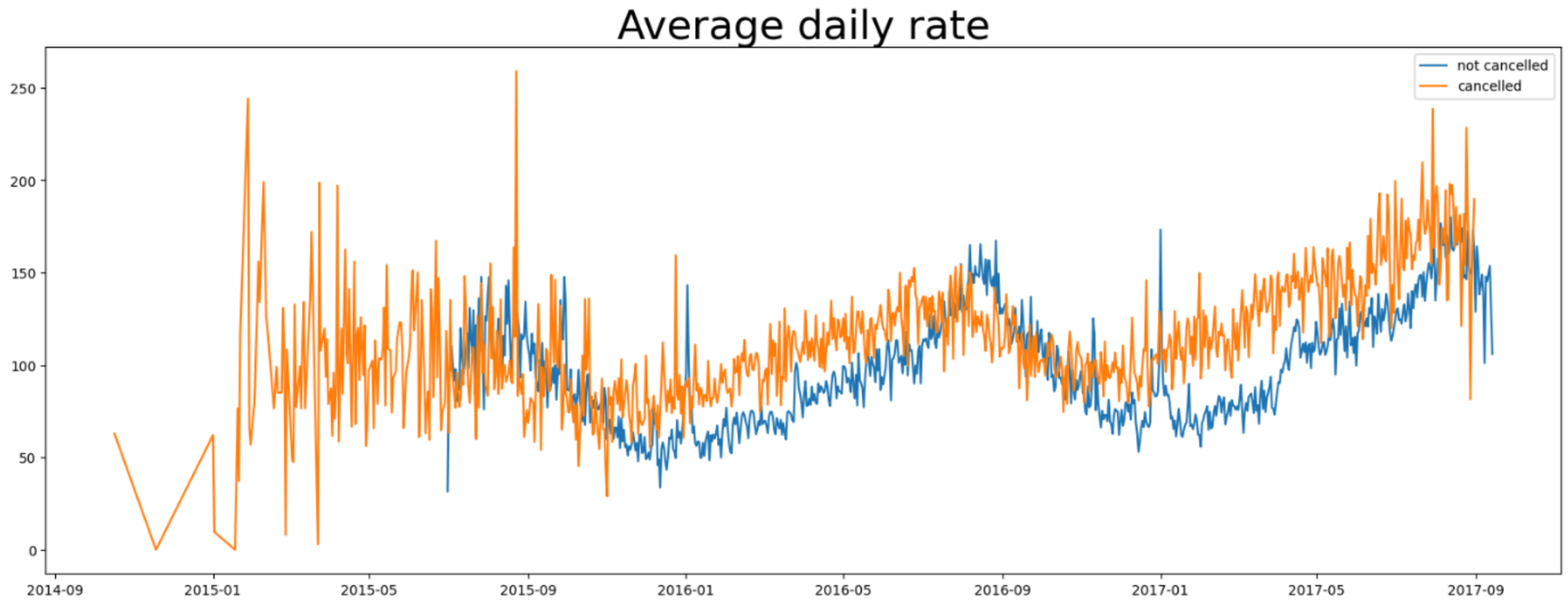
```
df['market_segment'].value_counts(normalize = True)
```

```
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
```

```python
# Group by reservation_status_date and calculate the mean ADR for canceled reservations
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)

# Filter for non-canceled data and calculate mean ADR
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)

# Plot the results
plt.figure(figsize=(20,7))
ax = plt.gca()  # Get the current axis
ax.set_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()
plt.show()
```



```
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')]
```

```
plt.figure(figsize = (20,7))
plt.title('Average daily rate', fontsize = 30)
plt.plot(not_cancelled_df_adr['reservation_status_date'],not_cancelled_df_adr['adr'], label = 'not canceled')
plt.plot(cancelled_df_adr['reservation_status_date'],cancelled_df_adr['adr'], label = 'canceled')
plt.legend(fontsize = 20)
```

```
<matplotlib.legend.Legend at 0x180d95a5be0>
```