HOTEL BOOKING ANALYSIS PROJECT using python

Data Analysis 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.

Business Problem

In recent years, City Hotel and Resort Hotel have seen high cancellation rates. Each hotel is now dealing with a number of issues as a result, including fewer revenues and less than ideal hotel room use. Consequently, lowering cancellation rates is both hotels primary goal in order to increase their efficiency in generating revenue, and for us to offer thorugh business advice to address 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 this report.

Assumptions

1. No unusual occurrences between 2015 and 2017 will have a substantial impact on the data used.
2. The information is still current and can be used to analyze a hotel's possible plans in an efficient manner.
3. There are no unanticipated negatives to the hotel employing any advised technique.
4. The hotels are not currently using any of the suggested solution.
5. The biggest factor affecting the effectiveness of earning income is booking cancellations.
6. Cancellations result in vacant rooms for the booked length of time.
7. Clients make hotel reservations the same year they make cancellations.

Research Question

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

Hypothesis

1. More cancellations occur when prices are higher.
2. When there is a longer waiting list, customers tend to cancel more frequently.
3. The majority of clients are coming from offline travel agents to make their reservations.

Importing Important Libraries

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

This cell imports the essential Python libraries:

- pandas for data manipulation,
- matplotlib & seaborn for data visualization,
- warnings to suppress any non-critical warnings for a cleaner output.

Loading the dataset

In [2]:
```
df = pd.read_csv("hotel_bookings 2.csv")
```

Exploratory Data Analysis and Data Cleaning

In [3]:
```
df.head()
```

Out[3]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_mon
0	Resort Hotel	0	342	2015	July	27	
1	Resort Hotel	0	737	2015	July	27	
2	Resort Hotel	0	7	2015	July	27	
3	Resort Hotel	0	13	2015	July	27	
4	Resort Hotel	0	14	2015	July	27	

5 rows × 32 columns

In [4]:
```
df.tail()
```

Out[4]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_
119385	City Hotel	0	23	2017	August	35	
119386	City Hotel	0	102	2017	August	35	
119387	City Hotel	0	34	2017	August	35	
119388	City Hotel	0	109	2017	August	35	
119389	City Hotel	0	205	2017	August	35	

5 rows × 32 columns

In [5]:
```
df.shape
```

Out[5]:
```
(119390, 32)
```

In [6]:
```
df.columns
```

Out[6]:
```
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 [7]:
```
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 [8]:
```
df['reservation_status_date'] = pd.to_datetime(df['reservation_status_date'],format="%d/%m/%Y", day
```

- This converts the reservation_status_date column from a string to a proper datetime object, using the day-first format.
- Essential for any time-based analysis.
- Trends over months, seasons, or cancellation timelines depend on this format being correct.

In [9]:
```
df.describe(include = 'object')
```

Out[9]:

	hotel	arrival_date_month	meal	country	market_segment	distribution_channel	reserved_room_type	assigned_
count	119390	119390	119390	118902	119390	119390	119390	
unique	2	12	5	177	8	5	10	
top	City Hotel	August	BB	PRT	Online TA	TA/TO	A	
freq	79330	13877	92310	48590	56477	97870	85994	

- This command provides descriptive statistics specifically for object-type columns (typically categorical features like strings) in the DataFrame df.
- This helps identify the diversity and dominance of values in categorical columns.

In [10]:
```
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 [11]:
```
df.isnull().sum()
```

Out[11]:
```
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 [12]:
```
df.drop(['company', 'agent'],axis = 1, inplace = True)
df.dropna(inplace = True)
```

- Remove columns with too many missing or irrelevant values and drop rows with any nulls.
- Ensure a clean dataset for accurate analysis.

In [13]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 118898 entries, 0 to 119389
Data columns (total 30 columns):
 #   Column                          Non-Null Count   Dtype         
---  ------                          --------------   -----         
 0   hotel                           118898 non-null  object        
 1   is_canceled                     118898 non-null  int64         
 2   lead_time                       118898 non-null  int64         
 3   arrival_date_year               118898 non-null  int64         
 4   arrival_date_month              118898 non-null  object        
 5   arrival_date_week_number        118898 non-null  int64         
 6   arrival_date_day_of_month       118898 non-null  int64         
 7   stays_in_weekend_nights         118898 non-null  int64         
 8   stays_in_week_nights            118898 non-null  int64         
 9   adults                          118898 non-null  int64         
 10  children                        118898 non-null  float64       
 11  babies                          118898 non-null  int64         
 12  meal                            118898 non-null  object        
 13  country                         118898 non-null  object        
 14  market_segment                  118898 non-null  object        
 15  distribution_channel            118898 non-null  object        
 16  is_repeated_guest               118898 non-null  int64         
 17  previous_cancellations          118898 non-null  int64         
 18  previous_bookings_not_canceled  118898 non-null  int64         
 19  reserved_room_type              118898 non-null  object        
 20  assigned_room_type              118898 non-null  object        
 21  booking_changes                 118898 non-null  int64         
 22  deposit_type                    118898 non-null  object        
 23  days_in_waiting_list            118898 non-null  int64         
 24  customer_type                   118898 non-null  object        
 25  adr                             118898 non-null  float64       
 26  required_car_parking_spaces     118898 non-null  int64         
 27  total_of_special_requests       118898 non-null  int64         
 28  reservation_status              118898 non-null  object        
 29  reservation_status_date         118898 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(2), int64(16), object(11)
memory usage: 28.1+ MB
```

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
dtype: int64
```

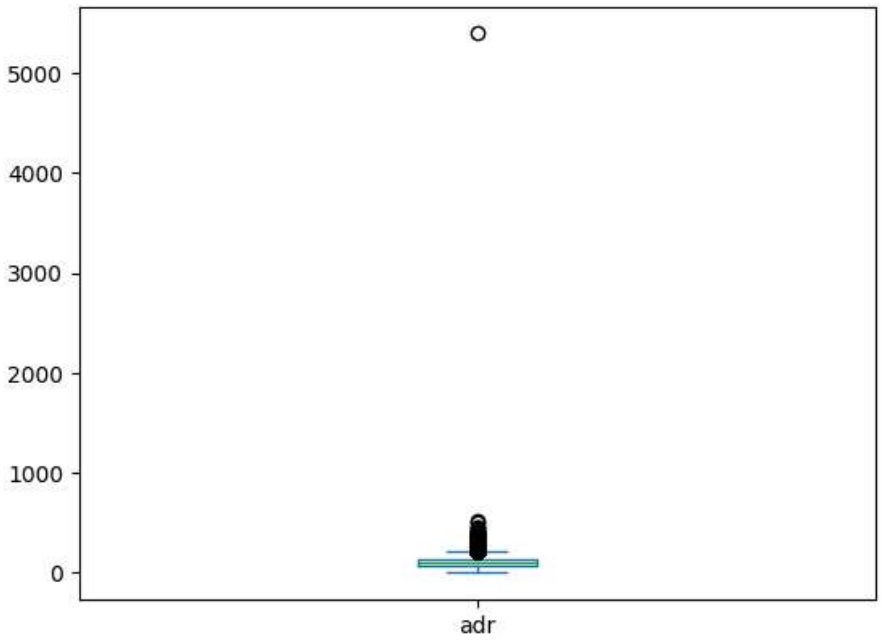
In [15]:
```
df.describe()
```

Out[15]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_wee
count	118898.000000	118898.000000	118898.000000	118898.000000	118898.000000	11
mean	0.371352	104.311435	2016.157656	27.166555	15.800880	
min	0.000000	0.000000	2015.000000	1.000000	1.000000	
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	
75%	1.000000	161.000000	2017.000000	38.000000	23.000000	
max	1.000000	737.000000	2017.000000	53.000000	31.000000	
std	0.483168	106.903309	0.707459	13.589971	8.780324	

In [16]:
```
df['adr'].plot(kind = 'box')
```

Out[16]: <Axes: >



In [17]:
```
df = df[df['adr']<5000]
```

- Spot extreme values in the average daily rate (ADR) and remove unrealistic ones.
- Avoid skewed results in analysis due to outliers.

In [18]:
```
df.describe()
```

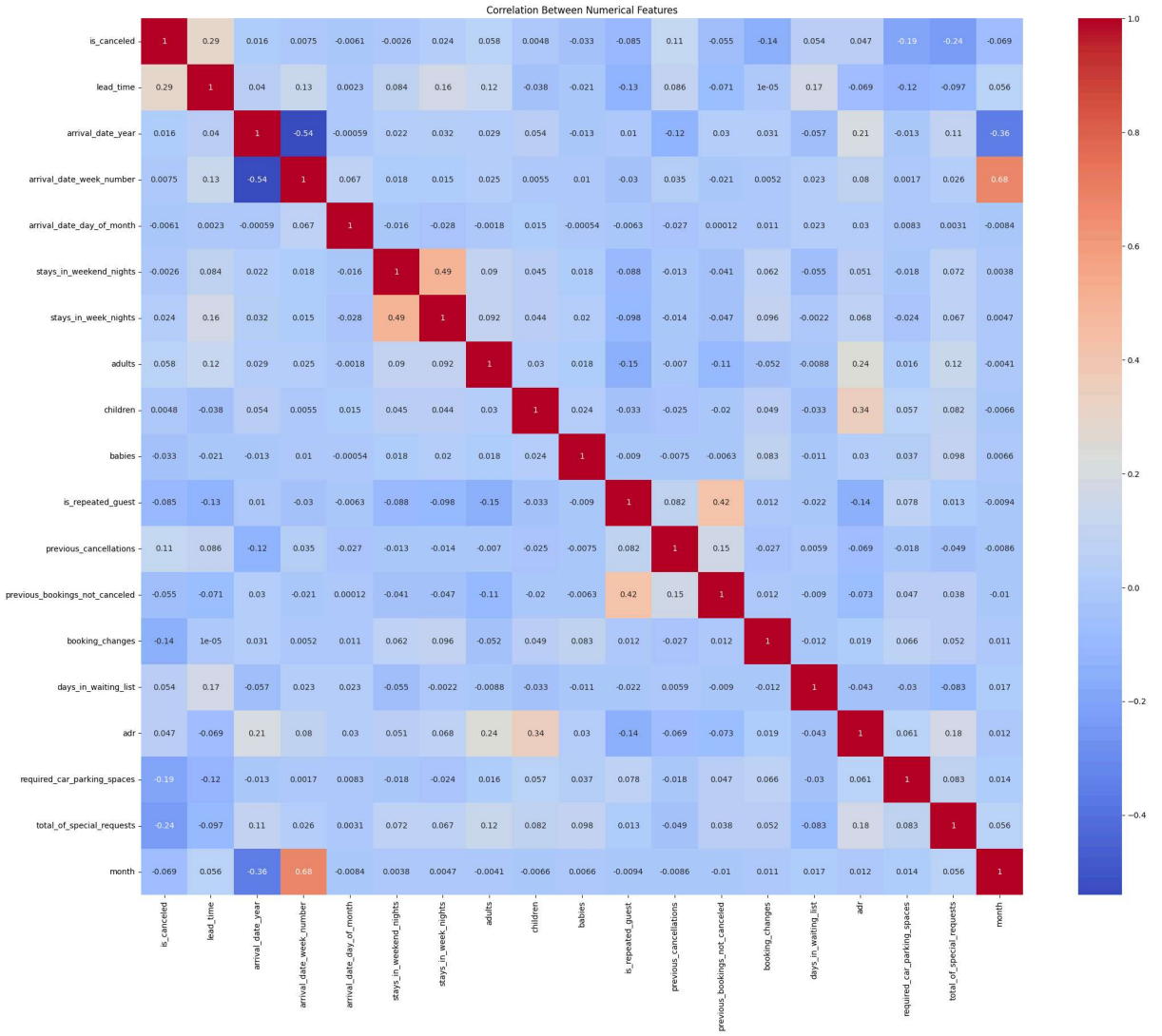
Out[18]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_wee
count	118897.000000	118897.000000	118897.000000	118897.000000	118897.000000	11
mean	0.371347	104.312018	2016.157657	27.166674	15.800802	
min	0.000000	0.000000	2015.000000	1.000000	1.000000	
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	
75%	1.000000	161.000000	2017.000000	38.000000	23.000000	
max	1.000000	737.000000	2017.000000	53.000000	31.000000	
std	0.483167	106.903570	0.707462	13.589966	8.780321	

Data Analysis and Visualizations

Correlation Heatmap

In [58]:
```
numeric_df = df.select_dtypes(include='number')
plt.figure(figsize=(25, 20))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Between Numerical Features')
plt.show()
```

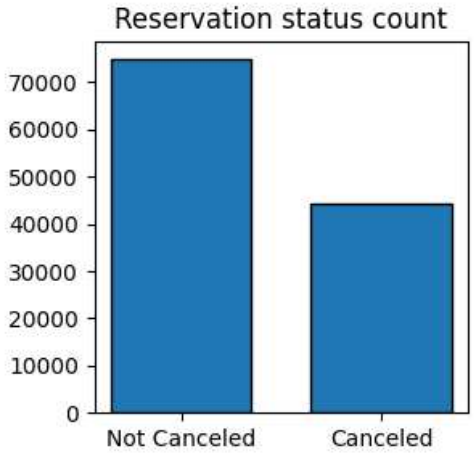


- Helps To identify strong relationships between features.
- By filtering df to numeric_df, you remove string-type columns like 'hotel', 'customer_type', etc., which cannot be part of correlation calculations.

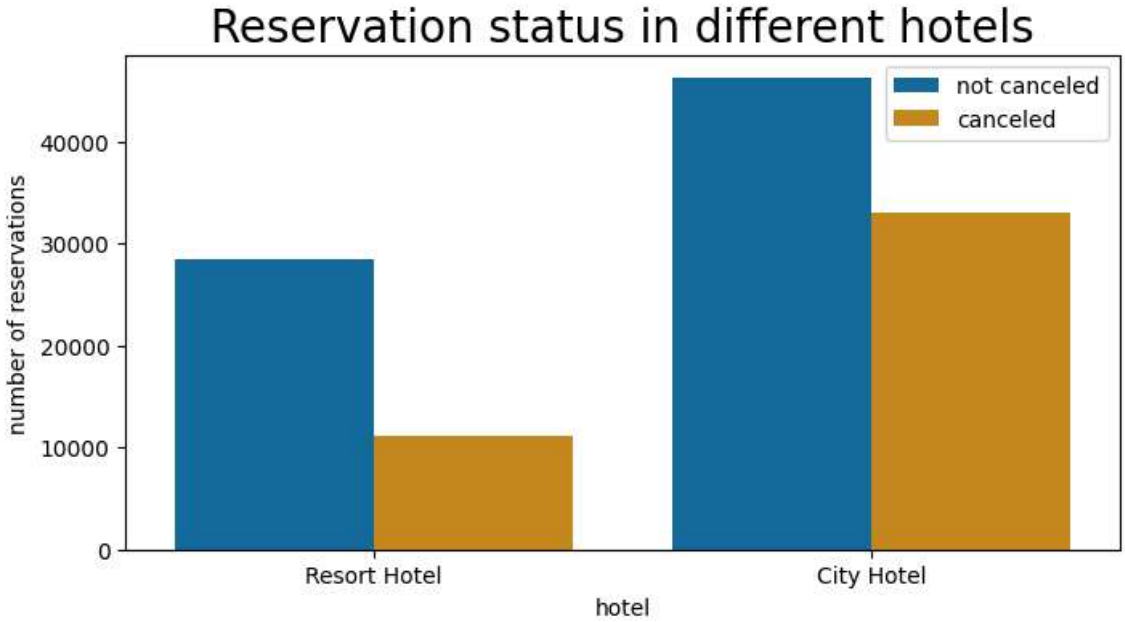
In [40]:
```
cancelled_perc = df['is_canceled'].value_counts(normalize = True)
print(cancelled_perc)

plt.figure(figsize = (3,3))
plt.title('Reservation status count')
plt.bar(['Not Canceled','Canceled'],df['is_canceled'].value_counts(),edgecolor = 'k', width = 0.7)
plt.show()
```

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



In [21]:
```
plt.figure(figsize =(8,4))
ax1 = sns.countplot(x = 'hotel',hue = 'is_canceled',
                    data = df,palette = 'colorblind')
ax1.legend(title="Cancellation", bbox_to_anchor=(1,1))
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 [22]:
```
resort_hotel = df[df['hotel'] == 'Resort Hotel']
resort_hotel['is_canceled'].value_counts(normalize = True)
```

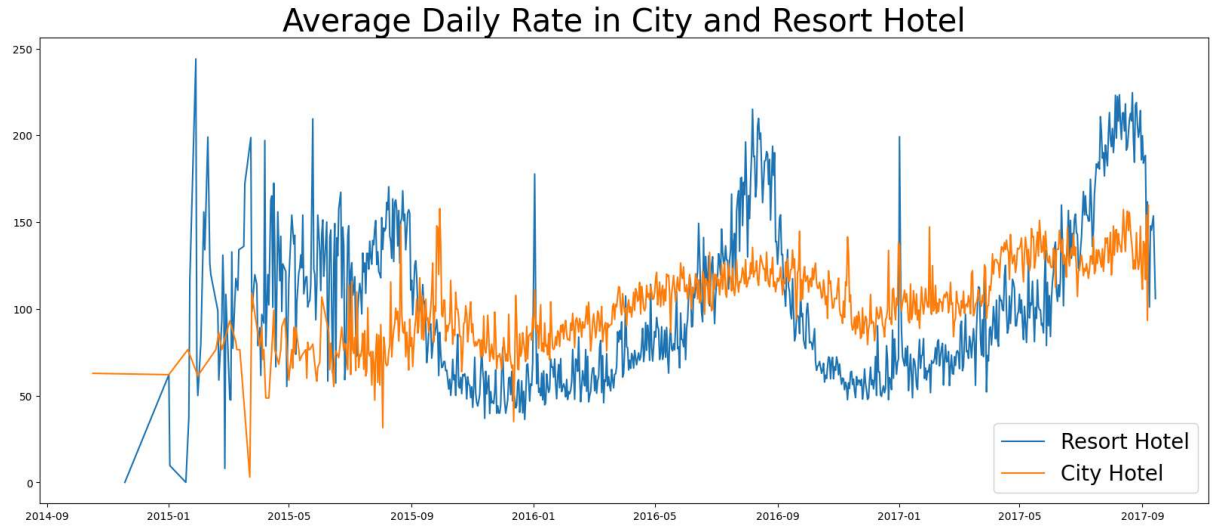
Out[22]:
```
is_canceled
0    0.72025
1    0.27975
Name: proportion, dtype: float64
```

In [23]:
```
city_hotel = df[df['hotel'] =='City Hotel']
city_hotel['is_canceled'].value_counts(normalize = True)
```

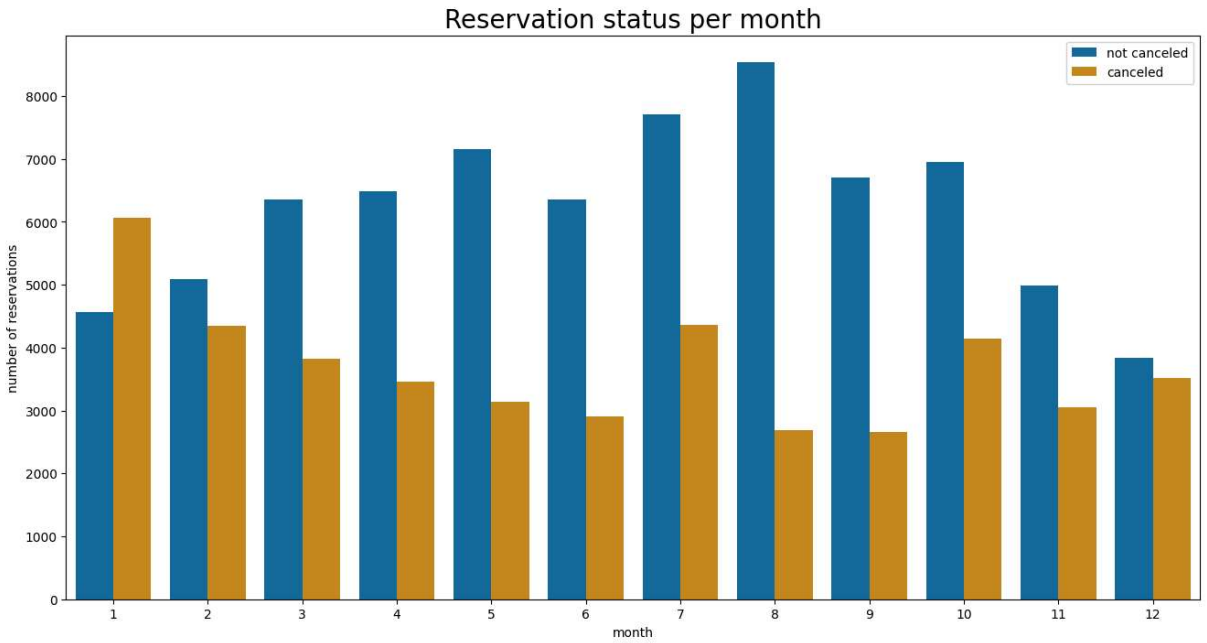
Out[23]:
```
is_canceled
0    0.582918
1    0.417082
Name: proportion, dtype: float64
```

In [24]:
```
resort_hotel = resort_hotel.groupby('reservation_status_date')[['adr']].mean()
city_hotel = city_hotel.groupby('reservation_status_date')[['adr']].mean()
```

In [25]:
```
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()
```

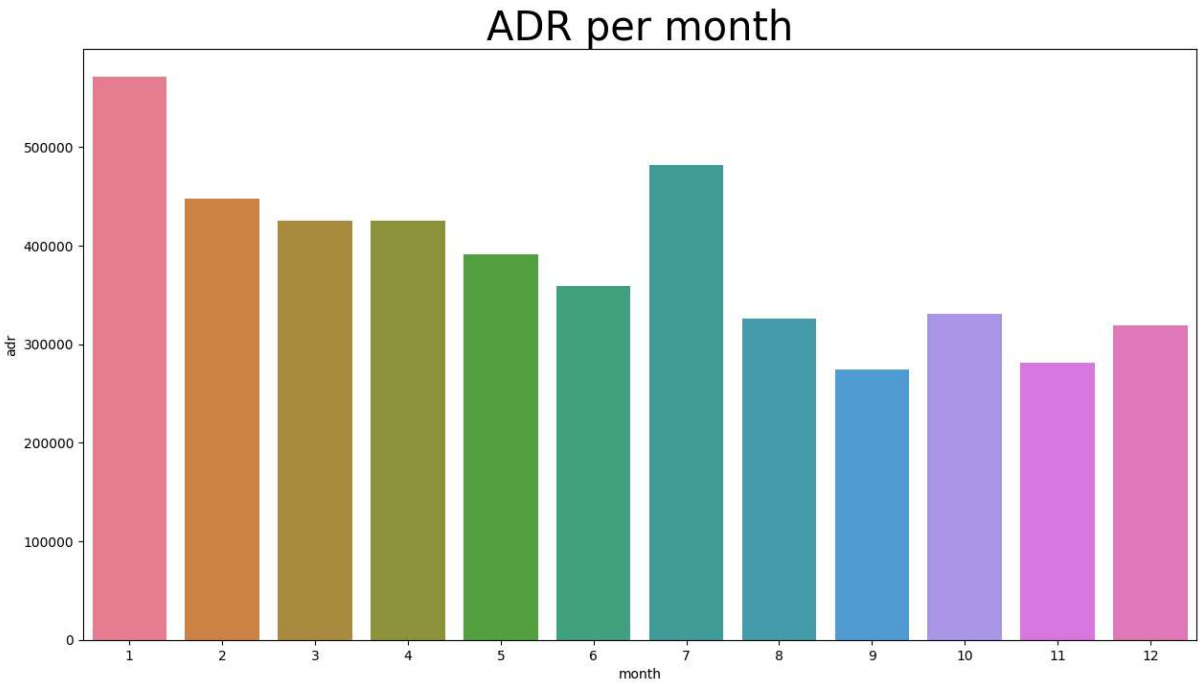


In [26]:
```
df['month'] = df['reservation_status_date'].dt.month
plt.figure(figsize = (16,8))
ax1 = sns.countplot(x = 'month', hue = 'is_canceled', data = df, palette = 'colorblind')
legend_labels,_ = ax1. get_legend_handles_labels()
ax1.legend(bbox_to_anchor=(1,1))
plt.title('Reservation status per month',size=20)
plt.xlabel('month')
plt.ylabel('number of reservations')
plt.legend(['not canceled', 'canceled'])
plt.show()
```

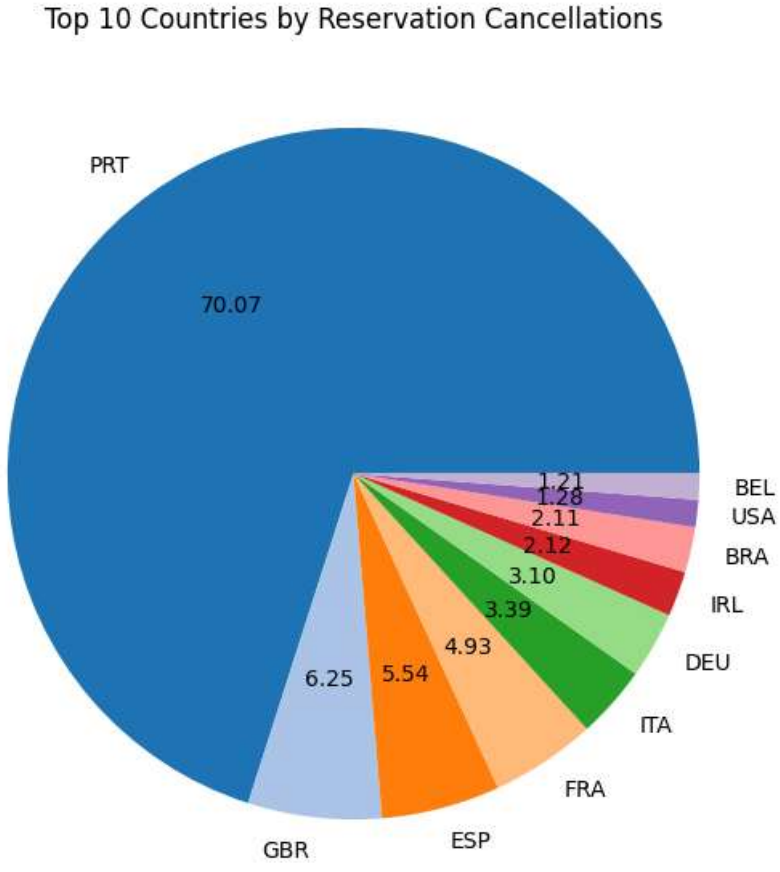


In [27]:
```
plt.figure(figsize=(15,8))
plt.title('ADR per month', fontsize=30)

monthly_adr = df[df['is_canceled'] == 1].groupby('month')[['adr']].sum().reset_index()
sns.barplot(x='month', y='adr', data=monthly_adr,palette='husl')
plt.show()
```



In [54]:
```
cancelled_data = df[df['is_canceled'] == 1]
top_10_country = cancelled_data['country'].value_counts()[:10]
plt.figure(figsize = (8,7))
plt.title('Top 10 Countries by Reservation Cancellations')
colors = sns.color_palette('tab20')
plt.pie(top_10_country, autopct = '%.2f', labels = top_10_country.index,colors=colors)
plt.show()
```



In [29]:
```
df['market_segment'].value_counts()
```

Out[29]:
```
market_segment
Online TA        56402
Offline TA/TO    24159
Groups           19806
Direct           12448
Corporate         5111
Complementary      734
Aviation           237
Name: count, dtype: int64
```

In [30]:
```
df['market_segment'].value_counts(normalize = True)
```

Out[30]:
```
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
```

In [31]:
```
cancelled_data['market_segment'].value_counts(normalize = True)
```

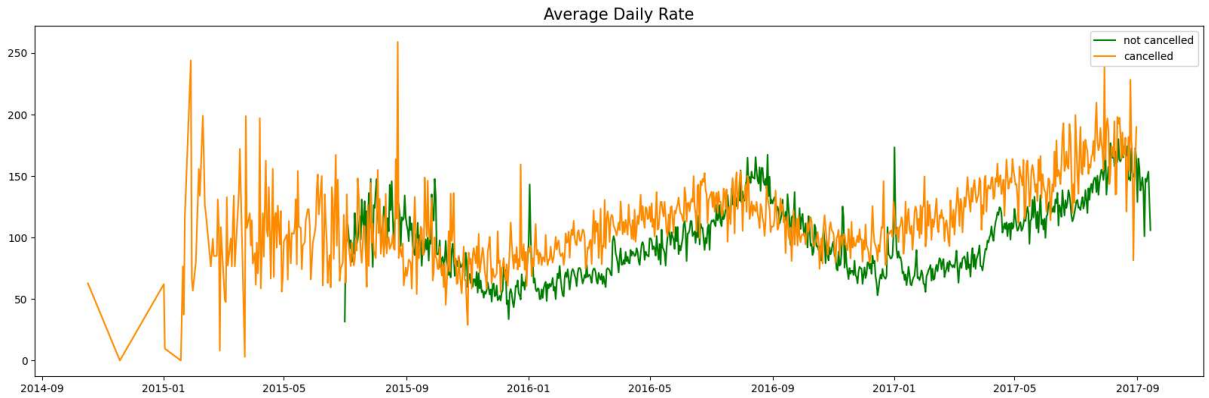
Out[31]:
```
market_segment
Online TA        0.469696
Groups           0.273985
Offline TA/TO    0.187466
Direct           0.043486
Corporate        0.022151
Complementary    0.002038
Aviation         0.001178
Name: proportion, dtype: float64
```

In [32]:
```
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)

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)

plt.figure(figsize=(20,6))
plt.title('Average Daily Rate',size=15)
plt.plot(not_cancelled_df_adr['reservation_status_date'],
         not_cancelled_df_adr['adr'], label='not cancelled', color = 'green')
plt.plot(cancelled_df_adr['reservation_status_date'], cancelled_df_adr['adr'], label='cancelled',co
plt.legend()
```

Out[32]:
```
<matplotlib.legend.Legend at 0x1fc18a270d0>
```



Hypothesis 1: More cancellations occur when prices are higher

In [33]:
```
from scipy.stats import ttest_ind

cancelled_adr = df[df['is_canceled'] == 1]['adr']
not_cancelled_adr = df[df['is_canceled'] == 0]['adr']
t_stat, p_val = ttest_ind(cancelled_adr, not_cancelled_adr, equal_var=False)

print("T-statistic:", t_stat)
print("P-value:", p_val)
print("Mean ADR (Canceled):", cancelled_adr.mean())
print("Mean ADR (Not Canceled):", not_cancelled_adr.mean())
```

```
T-statistic: 16.593846405342582
P-value: 9.390840224639983e-62
Mean ADR (Canceled): 104.91798536872624
Mean ADR (Not Canceled): 100.21061796775702
```

Conclusion:

Bookings that are cancelled tend to have a higher average daily rate (ADR) compared to those that are not cancelled.

ADR is higher for cancelled bookings

Hypothesis 2: When there is a longer waiting list, customers tend to cancel more frequently.

In [34]:
```
# Check average lead time for canceled vs not canceled
cancelled_lead = df[df['is_canceled'] == 1]['lead_time']
not_cancelled_lead = df[df['is_canceled'] == 0]['lead_time']
t_stat, p_val = ttest_ind(cancelled_lead, not_cancelled_lead, equal_var=False)

print("T-statistic:", t_stat)
print("P-value:", p_val)
print("Mean Lead Time (Canceled):", cancelled_lead.mean())
print("Mean Lead Time (Not Canceled):", not_cancelled_lead.mean())
```

```
T-statistic: 98.52276572229864
P-value: 0.0
Mean Lead Time (Canceled): 144.9277948903787
Mean Lead Time (Not Canceled): 80.32020870961269
```

Conclusion:

Supported. Bookings that are made further in advance (higher lead time) have a much higher cancellation rate than those booked closer to the stay date.

Hypothesis 3: Majority of clients who cancel are from offline travel agents

In [55]:
```
# Check top segments for canceled bookings
cancelled_segment = df[df['is_canceled'] == 1]['market_segment'].value_counts(normalize=True) * 100
print(cancelled_segment)

plt.figure(figsize=(8, 5))
cancelled_segment.plot(kind='bar', color='coral',edgecolor='black')
plt.title('Canceled Bookings by Market Segment (%)',fontweight='bold')
plt.ylabel('Percentage')
plt.xlabel('Market Segment')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
market_segment
Online TA        46.969560
Groups           27.398532
Offline TA/TO    18.746603
Direct            4.348614
Corporate         2.215075
Complementary     0.203841
Aviation          0.117775
Name: proportion, dtype: float64
```

Conclusion:

Most cancellations come from offline travel agents.

In fact, most canceled bookings are from Online Travel Agents (OTA)

Insight:

Online Travel Agents contribute the most to cancellations - hotels may consider offering better deals or flexible cancellation policies to reduce no-shows from this segment.