Author: Mr. Jay Jitesh Chavan
Data Analyst

Hotel Booking Analysis

1 Importing libraries

[8]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings ('ignore')
```

2 Loading the Dataset

[9]:
```
df = pd.read_csv(r'C:\Users\delll\OneDrive\Desktop\hotel_booking.csv')
```

3 Exploratory Data Analysis [EDA] & Data Cleaning

#View the dataset

[10]:
```
df.head()
```

[10]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	\
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	

	arrival_date_week_number	arrival_date_day_of_month	\
0	27	1	
1	27	1	
2	27	1	
3	27	1	
4	27	1	

	stays_in_weekend_nights	stays_in_week_nights	adults	…	deposit_type	\
0	0	0	2	…	No Deposit	
1	0	0	2	…	No Deposit	
2	0	1	1	…	No Deposit	

```
3                       0                   1       1  …    No Deposit
4                       0                   2       2  …    No Deposit

   agent company days_in_waiting_list customer_type   adr  \
0    NaN     NaN                    0     Transient   0.0
1    NaN     NaN                    0     Transient   0.0
2    NaN     NaN                    0     Transient  75.0
3  304.0     NaN                    0     Transient  75.0
4  240.0     NaN                    0     Transient  98.0

   required_car_parking_spaces  total_of_special_requests  reservation_status  \
0                            0                          0           Check-Out
1                            0                          0           Check-Out
2                            0                          0           Check-Out
3                            0                          0           Check-Out
4                            0                          1           Check-Out

  reservation_status_date
0                7/1/2015
1                7/1/2015
2                7/2/2015
3                7/2/2015
4                7/3/2015

[5 rows x 32 columns]
```

#Display the total number of columns and rows

[11]:
```
df.shape
```

[11]:
```
(119390, 32)
```

#Display the data types of each column

[12]:
```
df.dtypes
```

[12]:
```
hotel                           object
is_canceled                      int64
lead_time                        int64
arrival_date_year                int64
arrival_date_month              object
arrival_date_week_number         int64
arrival_date_day_of_month        int64
stays_in_weekend_nights          int64
stays_in_week_nights             int64
adults                           int64
children                       float64
babies                           int64
```

```
meal                                object
country                             object
market_segment                      object
distribution_channel                object
is_repeated_guest                    int64
previous_cancellations               int64
previous_bookings_not_canceled       int64
reserved_room_type                  object
assigned_room_type                  object
booking_changes                      int64
deposit_type                        object
agent                              float64
company                            float64
days_in_waiting_list                 int64
customer_type                       object
adr                                float64
required_car_parking_spaces          int64
total_of_special_requests            int64
reservation_status                  object
reservation_status_date             object
dtype: object
```

[13]:
```
# Convert the 'reservation_status_date' column to datetime format
df['reservation_status_date'] = pd.to_datetime(df['reservation_status_date'])
```

[14]:
```
df.dtypes
```

[14]:
```
hotel                               object
is_canceled                          int64
lead_time                            int64
arrival_date_year                    int64
arrival_date_month                  object
arrival_date_week_number             int64
arrival_date_day_of_month            int64
stays_in_weekend_nights              int64
stays_in_week_nights                 int64
adults                               int64
children                           float64
babies                               int64
meal                                object
country                             object
market_segment                      object
distribution_channel                object
is_repeated_guest                    int64
previous_cancellations               int64
previous_bookings_not_canceled       int64
reserved_room_type                  object
```

```
assigned_room_type                      object
booking_changes                          int64
deposit_type                            object
agent                                  float64
company                                float64
days_in_waiting_list                     int64
customer_type                           object
adr                                    float64
required_car_parking_spaces              int64
total_of_special_requests                int64
reservation_status                      object
reservation_status_date          datetime64[ns]
dtype: object
```

#View the number of values in object (categorical) columns

[15]:
```
df.describe(include= 'object')
```

[15]:
```
              hotel arrival_date_month    meal country market_segment  \
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

       distribution_channel reserved_room_type assigned_room_type  \
count                119390             119390              119390
unique                    5                 10                  12
top                   TA/TO                  A                   A
freq                  97870              85994               74053

       deposit_type customer_type reservation_status
count        119390        119390             119390
unique            3             4                  3
top      No Deposit     Transient          Check-Out
freq         104641         89613              75166
```

#Viewing all data in the categorical column

[16]:
```
# Loop through each column in the DataFrame that contains object (string) data␣
 ↪types
for col in df.describe(include='object').columns:
    print(col)  # Print the name of the column
    print(df[col].unique())  # Print the unique values present in that column
    print('-' * 50)  # Print a separator line for better readability
```

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

#Checking for Missing Values

[17]:
```
df.isnull().sum()
```

[17]:
```
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
```

[18]:
```
# Dropping the columns 'company' and 'agent' from the DataFrame df.
# The 'axis=1' parameter specifies that columns (not rows) are being dropped.
# 'inplace=True' means the changes will be applied directly to df, modifying it␣
 ↪in place.
df.drop(['company', 'agent'], axis=1, inplace=True)
```

[19]:
```
df.dropna(inplace = True)
```

[20]:
```
df.isnull().sum()
```

[20]:
```
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
```

#Viewing Summary Statistics of Numerical Columns

[21]:
```
df.describe()
```

[21]:
```
          is_canceled      lead_time  arrival_date_year  \
count  118898.000000  118898.000000      118898.000000
mean        0.371352     104.311435        2016.157656
min         0.000000       0.000000        2015.000000
25%         0.000000      18.000000        2016.000000
50%         0.000000      69.000000        2016.000000
75%         1.000000     161.000000        2017.000000
max         1.000000     737.000000        2017.000000
std         0.483168     106.903309           0.707459

       arrival_date_week_number  arrival_date_day_of_month  \
```

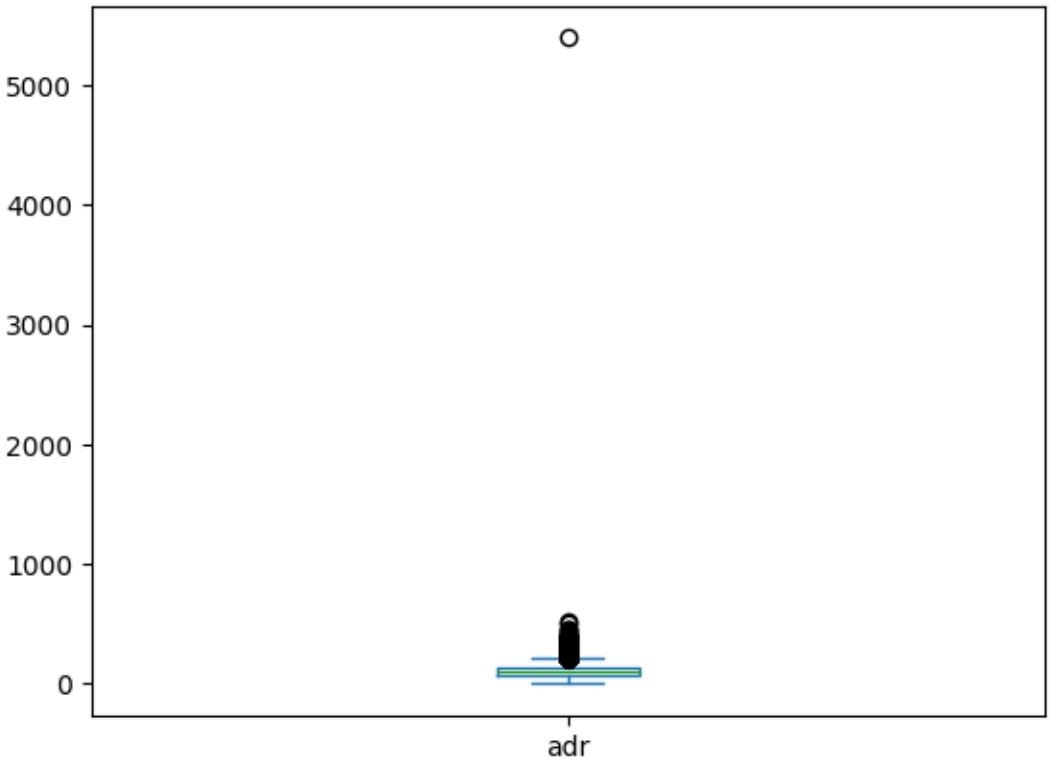
```
count             118898.000000              118898.000000
mean                  27.166555                  15.800880
min                    1.000000                   1.000000
25%                   16.000000                   8.000000
50%                   28.000000                  16.000000
75%                   38.000000                  23.000000
max                   53.000000                  31.000000
std                   13.589971                   8.780324

       stays_in_weekend_nights  stays_in_week_nights         adults  \
count            118898.000000         118898.000000  118898.000000
mean                  0.928897              2.502145       1.858391
min                   0.000000              0.000000       0.000000
25%                   0.000000              1.000000       2.000000
50%                   1.000000              2.000000       2.000000
75%                   2.000000              3.000000       2.000000
max                  16.000000             41.000000      55.000000
std                   0.996216              1.900168       0.578576

            children         babies  is_repeated_guest  \
count  118898.000000  118898.000000      118898.000000
mean        0.104207       0.007948           0.032011
min         0.000000       0.000000           0.000000
25%         0.000000       0.000000           0.000000
50%         0.000000       0.000000           0.000000
75%         0.000000       0.000000           0.000000
max        10.000000      10.000000           1.000000
std         0.399172       0.097380           0.176029

       previous_cancellations  previous_bookings_not_canceled  \
count           118898.000000                   118898.000000
mean                 0.087142                        0.131634
min                  0.000000                        0.000000
25%                  0.000000                        0.000000
50%                  0.000000                        0.000000
75%                  0.000000                        0.000000
max                 26.000000                       72.000000
std                  0.845869                        1.484672

       booking_changes  days_in_waiting_list            adr  \
count    118898.000000         118898.000000  118898.000000
mean          0.221181              2.330754     102.003243
min           0.000000              0.000000      -6.380000
25%           0.000000              0.000000      70.000000
50%           0.000000              0.000000      95.000000
75%           0.000000              0.000000     126.000000
max          21.000000            391.000000    5400.000000
```

```
std           0.652785             17.630452      50.485862

       required_car_parking_spaces  total_of_special_requests  \
count                 118898.000000              118898.000000
mean                       0.061885                   0.571683
min                        0.000000                   0.000000
25%                        0.000000                   0.000000
50%                        0.000000                   0.000000
75%                        0.000000                   1.000000
max                        8.000000                   5.000000
std                        0.244172                   0.792678

            reservation_status_date
count                        118898
mean   2016-07-30 07:37:53.336809984
min              2014-10-17 00:00:00
25%              2016-02-02 00:00:00
50%              2016-08-08 00:00:00
75%              2017-02-09 00:00:00
max              2017-09-14 00:00:00
std                              NaN
```

#Checking for Outliers Using a Box Plot

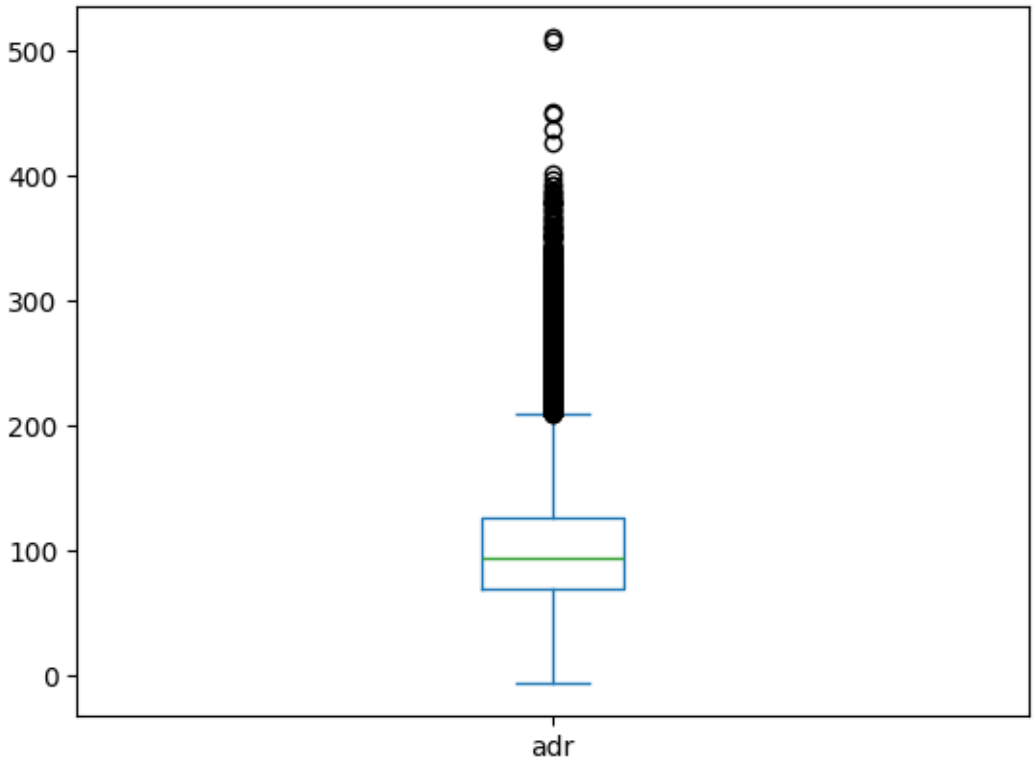
[22]:
```
# Plot a box plot for the 'adr' column to visualize its distribution.
df['adr'].plot(kind='box')
```

[22]:
```
<Axes: >
```



[23]:
```
#Filter rows where 'adr' is less than 5000
df=df[df['adr']<5000]

# Plot the boxplot for the 'adr' column
df['adr'].plot(kind='box')
```

[23]: <Axes: >



4 Data Analysis and Visualizations

#Note: “0” = Booking Not Cancelled & “1” = Booking Cancelled

#Checking Percentage of Reservation Status

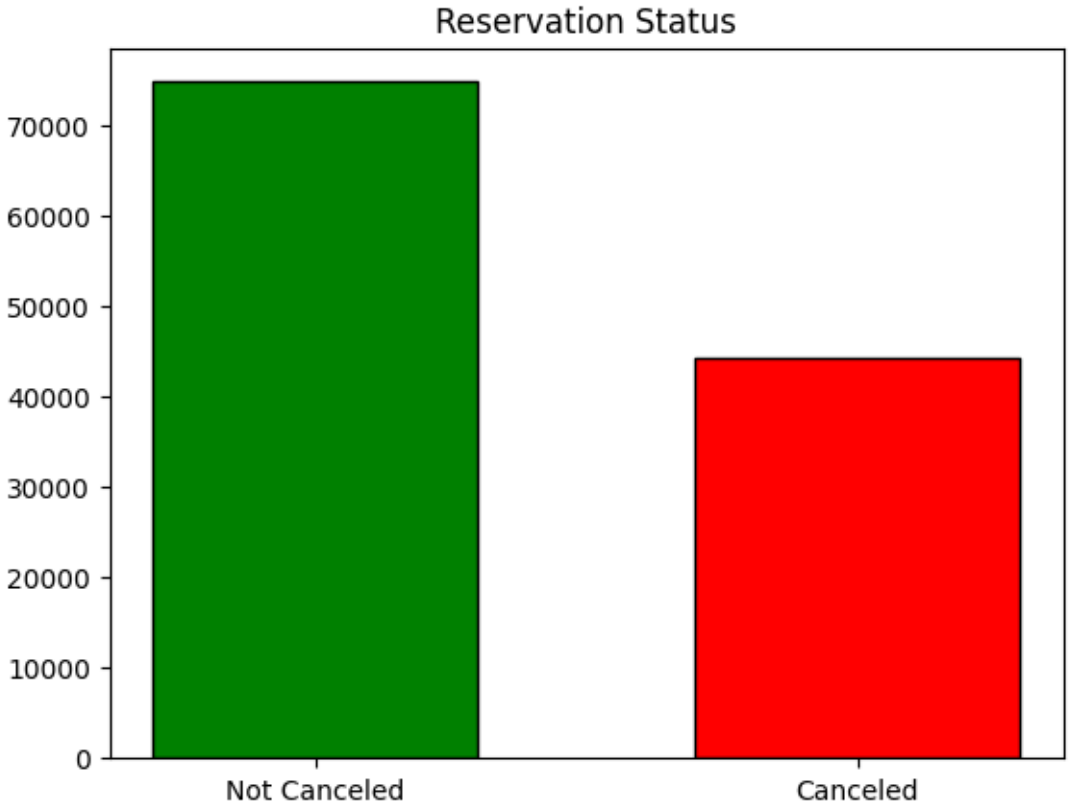
[24]:
```
# Calculate the percentage of cancellation occurrences
cancelled_per = df['is_canceled'].value_counts(normalize=True)

# Print the result
print(cancelled_per)
```

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

#Plotting the Bar Graph of Reservation Status

[25]:
```
# Set the title for the plot
plt.title('Reservation Status')

# Define colors for each category
```

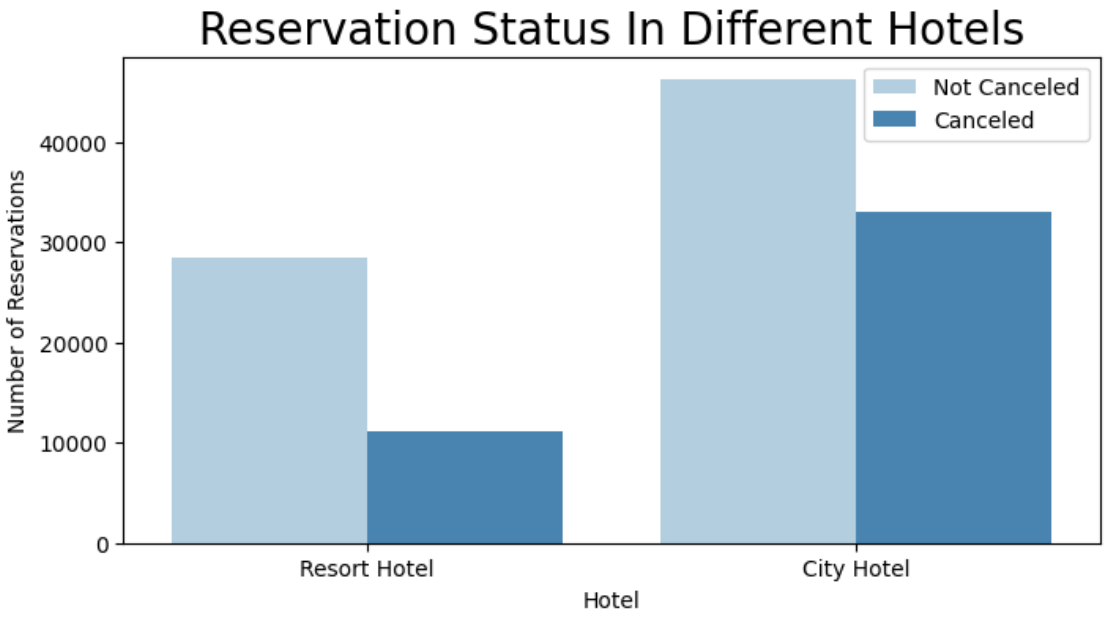
```
colors = ['green', 'red']  # 'Not Canceled' will be green, 'Canceled' will be␣
 ↪red

# Create a bar plot with specified colors
plt.bar(['Not Canceled', 'Canceled'], df['is_canceled'].value_counts(),␣
 ↪edgecolor='k', width=0.6, color=colors)

# Show the plot
plt.show()
```



Checking the cancellation and non-cancellation ratio based on hotels.

[26]:
```
# Set the figure size
plt.figure(figsize=(8, 4))

# Create a count plot using Seaborn
ax1 = sns.countplot(x='hotel', hue='is_canceled', data=df, palette='Blues')

# Adjust the legend position
legend_labels, _ = ax1.get_legend_handles_labels()
ax1.legend(bbox_to_anchor=(1, 1), labels=['Not Canceled', 'Canceled'])
```

```
# Set plot title and labels
plt.title('Reservation Status In Different Hotels', size=20)
plt.xlabel('Hotel')
plt.ylabel('Number of Reservations')

# Show the plot
plt.show()
```



#Checking Booking Cancellation for Resort Hotels

[27]:
```
# Filter the DataFrame to include only rows where the hotel is 'Resort Hotel'
resort_hotel = df[df['hotel'] == 'Resort Hotel']

# Calculate the percentage of cancellations for Resort Hotel
cancellation_rates = resort_hotel['is_canceled'].value_counts(normalize=True)

# Print the cancellation rates
print(cancellation_rates)
```

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

#Checking Booking Cancellation for City Hotels

[28]:
```
# Filter the DataFrame to include only rows where the hotel is 'City Hotel'
city_hotel = df[df['hotel'] == 'City Hotel']

# Calculate the percentage of cancellations for City Hotel
cancellation_rates_city_hotel = city_hotel['is_canceled'].
 ↪value_counts(normalize=True)

# Print the cancellation rates
print(cancellation_rates_city_hotel)
```

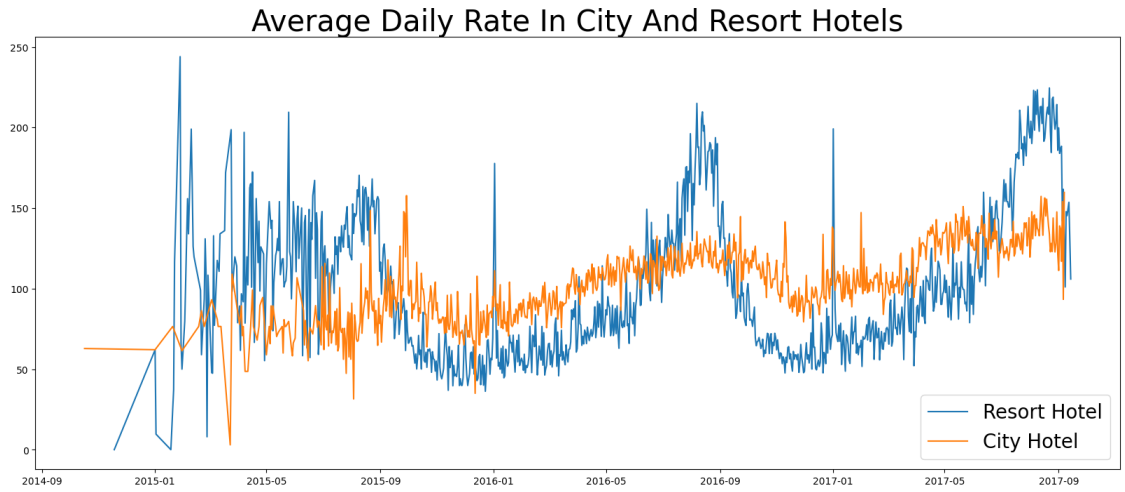
```
is_canceled
0    0.582918
1    0.417082
Name: proportion, dtype: float64
```

#Analyzing the Effect of Price on Resort Hotel and City Hotel Cancellations

[29]:
```
# Group the resort_hotel DataFrame by 'reservation_status_date' and calculate␣
 ↪the mean 'adr' (average daily rate) for each date
resort_hotel = resort_hotel.groupby('reservation_status_date')[['adr']].mean()

# Group the city_hotel DataFrame by 'reservation_status_date' and calculate the␣
 ↪mean 'adr' (average daily rate) for each date
city_hotel = city_hotel.groupby('reservation_status_date')[['adr']].mean()
```

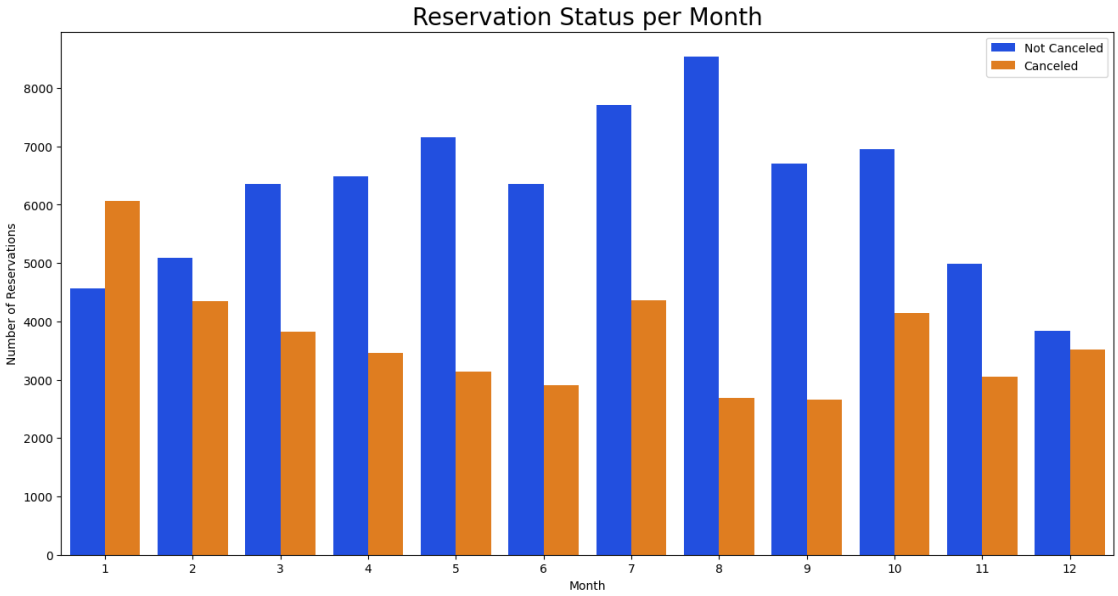
#Generating a line plot to compare the average daily rates of resort and city hotels.

[30]:
```
# Set the size of the figure
plt.figure(figsize=(20, 8))

# Set the title of the plot
plt.title('Average Daily Rate In City And Resort Hotels', size=30)

# Plot the 'adr' values over time for Resort Hotel
plt.plot(resort_hotel.index, resort_hotel['adr'], label='Resort Hotel')

# Plot the 'adr' values over time for City Hotel
plt.plot(city_hotel.index, city_hotel['adr'], label='City Hotel')

# Add a legend to distinguish between City and Resort Hotels
plt.legend(fontsize=20)

# Show the plot
plt.show()
```



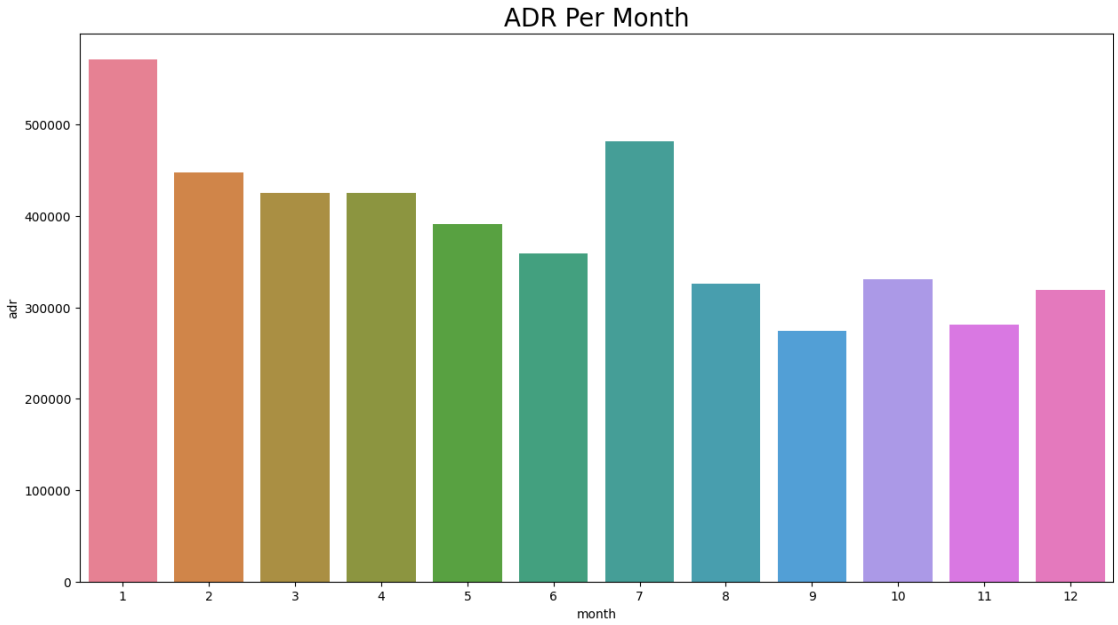
#Visualizing the number of reservations per month

[31]:
```
# Extract the month from reservation_status_date
df['month'] = df['reservation_status_date'].dt.month

# Set figure size
plt.figure(figsize=(16,8))

# Create a countplot
ax1 = sns.countplot(x='month', hue='is_canceled', data=df, palette='bright')

# Customize legend to appear outside the plot
legend_labels, _ = ax1.get_legend_handles_labels()
ax1.legend(bbox_to_anchor=(1, 1))

# Add title and axis labels
plt.title('Reservation Status per Month', size=20)
plt.xlabel('Month')
plt.ylabel('Number of Reservations')

# Set legend labels
plt.legend(['Not Canceled', 'Canceled'], loc='upper right')

# Show plot
plt.show()
```



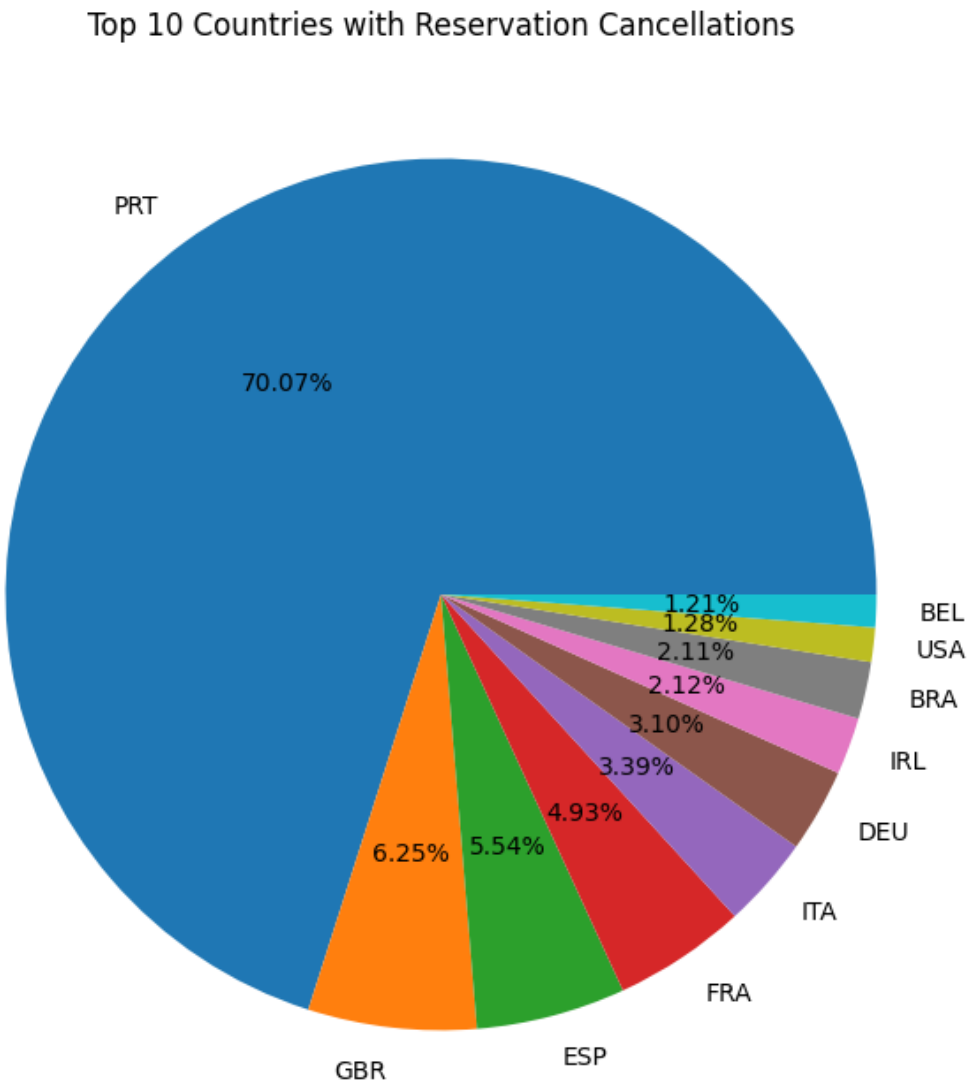
#Total ADR for each month

[32]:
```
# Set figure size
plt.figure(figsize=(15,8))

# Set title
plt.title('ADR Per Month', fontsize=20)

# Define a custom list of colors
num_months = len(df['month'].unique())
colors = sns.color_palette("husl", n_colors=num_months)  # Adjust colors based␣
 ↪on the number of months

# Plot average ADR for each month for canceled reservations with color palette
sns.barplot(x='month', y='adr', data=df[df['is_canceled'] == 1].
 ↪groupby('month')[['adr']].sum().reset_index(), palette=colors)

# Show the plot
plt.show()
```



#Top 10 countries with the highest cancellation rates

[33]:
```
# Filter the data for canceled reservations
cancelled_data = df[df['is_canceled'] == 1]

# Get the top 10 countries with the highest cancellation rates
top_10_countries = cancelled_data['country'].value_counts()[:10]

# Set up the figure and title
plt.figure(figsize=(8, 8))
plt.title("Top 10 Countries with Reservation Cancellations")

# Create a pie chart with percentages and country labels
plt.pie(top_10_countries, autopct='%.2f%%', labels=top_10_countries.index)

# Show the plot
plt.show()
```



#Count the occurrences of each unique value in the "market_segment" column

[34]:
```
df["market_segment"].value_counts()
```

[34]:
```
market_segment
Online TA        56402
Offline TA/TO    24159
Groups           19806
Direct           12448
Corporate         5111
Complementary      734
Aviation           237
```

```
Name: count, dtype: int64
```

#Proportion of each unique value in the “market_segment” column instead of the absolute counts. (Results In the Form of %)

[35]:
```
df["market_segment"].value_counts(normalize = True)
```

[35]:
```
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
```

#Calculates the relative frequencies of each unique value in the “market_segment” column from the cancelled_data.

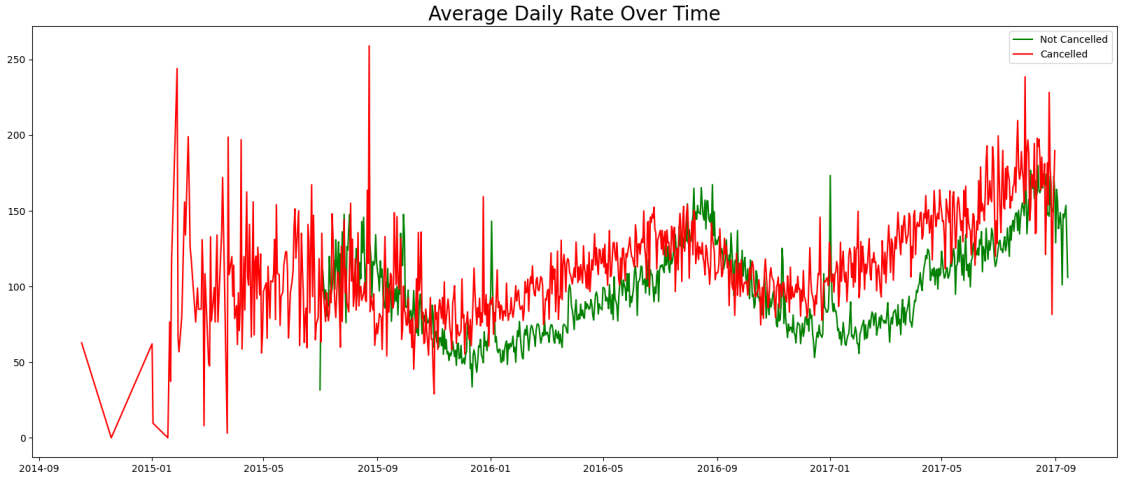
[36]:
```
cancelled_data["market_segment"].value_counts(normalize = True)
```

[36]:
```
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
```

#Average Daily Rate (ADR) for Cancelled and Non-Cancelled Reservations Over Time

[44]:
```
# Calculate the average daily rate (ADR) for cancelled reservations
cancelled_df_adr = cancelled_data.groupby('reservation_status_date')[['adr']].
 ↪mean()

# Reset the index to make 'reservation_status_date' a column
cancelled_df_adr.reset_index(inplace=True)

# Sort the values by 'reservation_status_date'
cancelled_df_adr.sort_values('reservation_status_date', inplace=True)

# Create a DataFrame for non-cancelled reservations
not_cancelled_df = df[df['is_canceled'] == 0]

# Calculate the average daily rate (ADR) for non-cancelled reservations
```

```
not_cancelled_df_adr = not_cancelled_df.
 ↪groupby('reservation_status_date')[['adr']].mean()

# Reset the index to make 'reservation_status_date' a column
not_cancelled_df_adr.reset_index(inplace=True)

# Sort the values by 'reservation_status_date'
not_cancelled_df_adr.sort_values('reservation_status_date', inplace=True)

# Plot the ADR for both cancelled and non-cancelled reservations
plt.figure(figsize=(20, 8))
plt.title('Average Daily Rate Over Time', fontsize=20)

# Plot the ADR for non-cancelled reservations
plt.plot(not_cancelled_df_adr['reservation_status_date'],␣
 ↪not_cancelled_df_adr['adr'], label='Not Cancelled', color='green')

# Plot the ADR for cancelled reservations
plt.plot(cancelled_df_adr['reservation_status_date'], cancelled_df_adr['adr'],␣
 ↪label='Cancelled', color='red')

# Add a legend
plt.legend()

# Display the plot
plt.show()
```



#Filtering ADR Data for Cancelled and Not Cancelled Reservations from 2016 to September 2017

[45]:
```
# Filter cancelled ADR data based on the reservation status date
cancelled_df_adr =␣
 ↪cancelled_df_adr[(cancelled_df_adr['reservation_status_date'] >= '2016') &
                                    ␣
 ↪(cancelled_df_adr['reservation_status_date'] <= '2017-09')]

# Filter not cancelled ADR data based on the reservation status date
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')]
```

#ADR Over Time for Cancelled and Non-Cancelled Reservations,.

[46]:
```
# Plot the ADR for both cancelled and non-cancelled reservations
plt.figure(figsize=(20, 8))
plt.title('Average Daily Rate Over Time', fontsize=20)

# Plot the ADR for non-cancelled reservations
plt.plot(not_cancelled_df_adr['reservation_status_date'],␣
 ↪not_cancelled_df_adr['adr'], label='Not Cancelled', color='green')

# Plot the ADR for cancelled reservations
plt.plot(cancelled_df_adr['reservation_status_date'], cancelled_df_adr['adr'],␣
 ↪label='Cancelled', color='red')

# Add a legend
plt.legend()

# Display the plot
plt.show()
```