```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
```

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files hotel_bookings.csv
```

- **hotel_bookings.csv**(text/csv) - 17009960 bytes, last modified: 6/3/2022 - 100% done

```
Saving hotel_bookings.csv to hotel_bookings.csv
```

```
df = pd.read_csv("hotel_bookings.csv", encoding = "unicode_escape")
```

Data Exploration and Cleaning

```
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  object 
 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(15), object(13)
memory usage: 29.1+ MB
```

```
# changing data type of reservation_status_date to datetime
df["reservation_status_date"] = df["reservation_status_date"].astype(np.datetime64)


#finding unique values of all columns whose datatype is object
for columns in df.describe(include = "object"):
  print(columns)
  print(df[columns].unique())
  print("------------------")


#checking null values
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
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
```

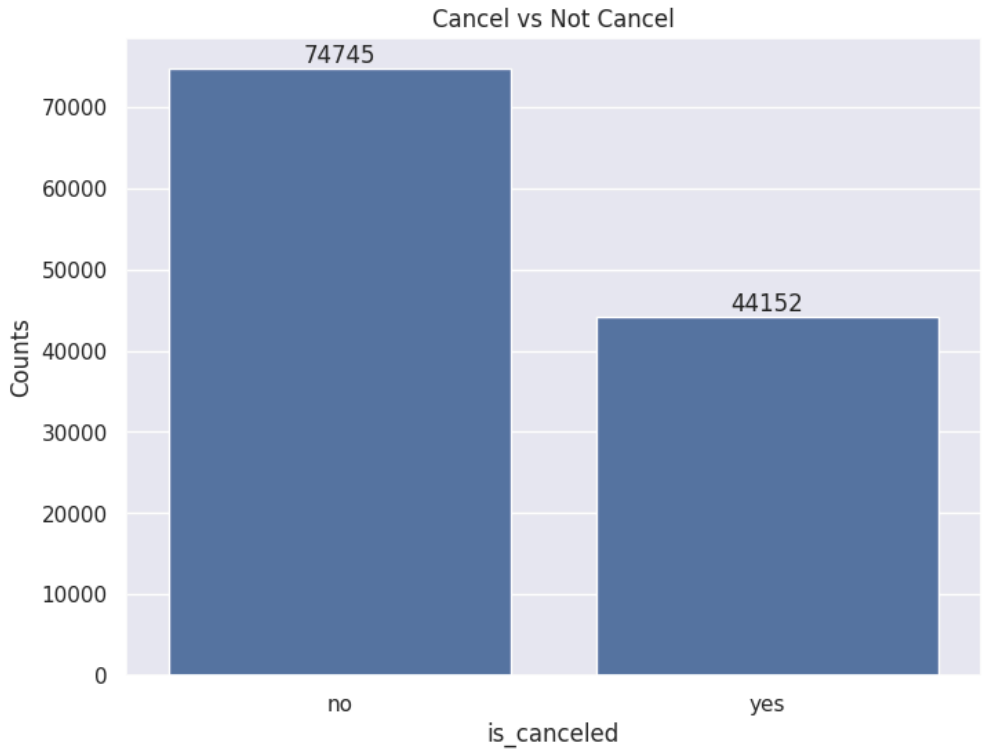
```
#dropping column with very high null values and deleting rows with null values which are not much high number
del[[df["agent"],df["company"]]]



df.dropna(inplace = True)


df.describe()


#as you can see max value in adr is 5400 which is outliner so you want to remove it
df = df[df["adr"] < 5400]
```

Data Analysis and Visualizations

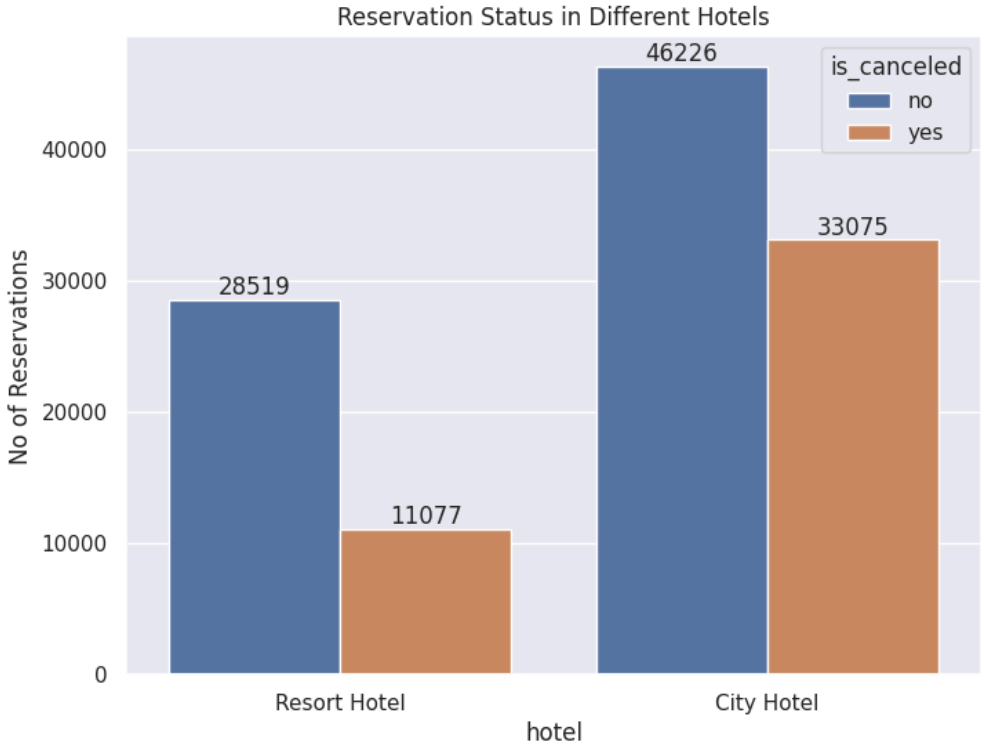
```
data1 = df["is_canceled"].value_counts(normalize = True)


data1
```

```
no     0.628653
yes    0.371347
Name: is_canceled, dtype: float64
```

```
#barplot showing canceled reservations and not-canceled reservations
ax = sb.countplot(x = "is_canceled", data = df)
plt.title("Cancel vs Not Cancel")
plt.ylabel("Counts")
for bars in ax.containers:
  ax.bar_label(bars)
```

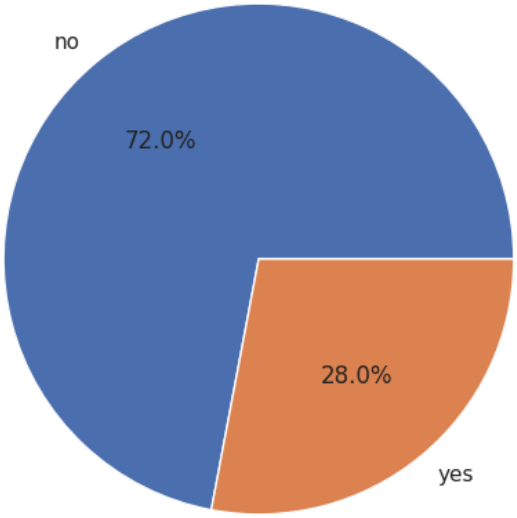



```
#clustered column chart showing resevation status in different hotels
qw = sb.countplot(x = "hotel", hue = "is_canceled", data = df)
sb.set(rc = {"figure.figsize" : (7,6)})
plt.ylabel("No of Reservations")
plt.title("Reservation Status in Different Hotels")
for bars in qw.containers:
  qw.bar_label(bars)
```



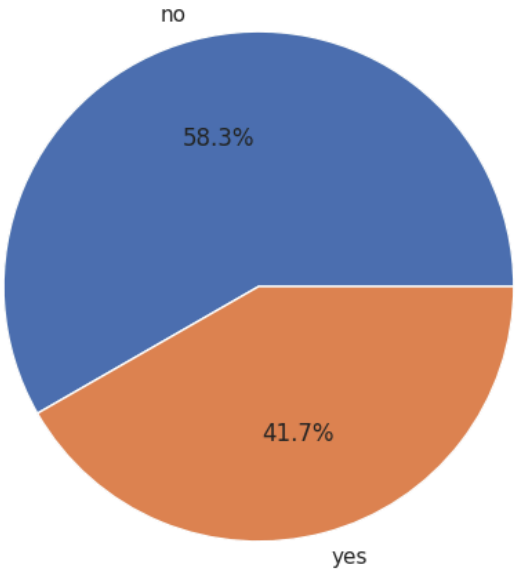

```
#distribution of cancelled and non canceled reservations in both type of hotels
resort_hotel = df[df["hotel"] == "Resort Hotel"]
city_hotel = df[df["hotel"] == "City Hotel"]


resort = resort_hotel["is_canceled"].value_counts()


plt.pie(resort,labels = resort.index, autopct = "%1.1f%%")
```

```
([<matplotlib.patches.Wedge at 0x795bb48611b0>,
  <matplotlib.patches.Wedge at 0x795bb4861090>],
 [Text(-0.7018306071158671, 0.8470146391387657, 'no'),
  Text(0.7018306071158673, -0.8470146391387655, 'yes')],
 [Text(-0.3828166947904729, 0.46200798498478124, '72.0%'),
  Text(0.38281669479047303, -0.46200798498478113, '28.0%')])
```



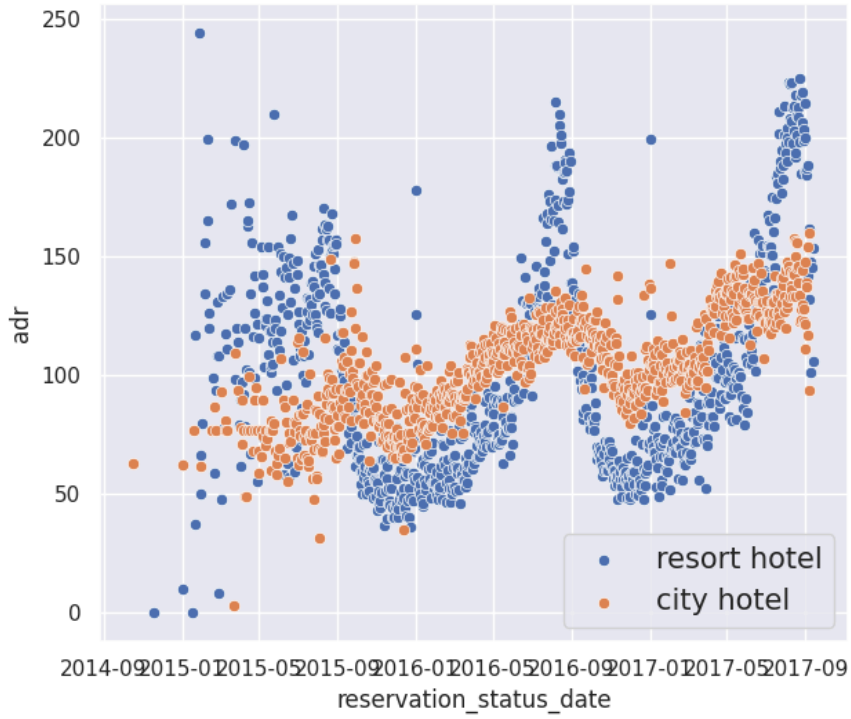

```
city = city_hotel['is_canceled'].value_counts()


plt.pie(city, labels = city.index, autopct = "%1.1f%%")
```

```
([<matplotlib.patches.Wedge at 0x795bb48c81c0>,
  <matplotlib.patches.Wedge at 0x795bb48c80a0>],
 [Text(-0.2833150820244359, 1.0628887826567215, 'no'),
  Text(0.28331498250960474, -1.0628888091825892, 'yes')],
 [Text(-0.15453549928605592, 0.5797575178127571, '58.3%'),
  Text(0.15453544500523891, -0.5797575322814122, '41.7%')])
```



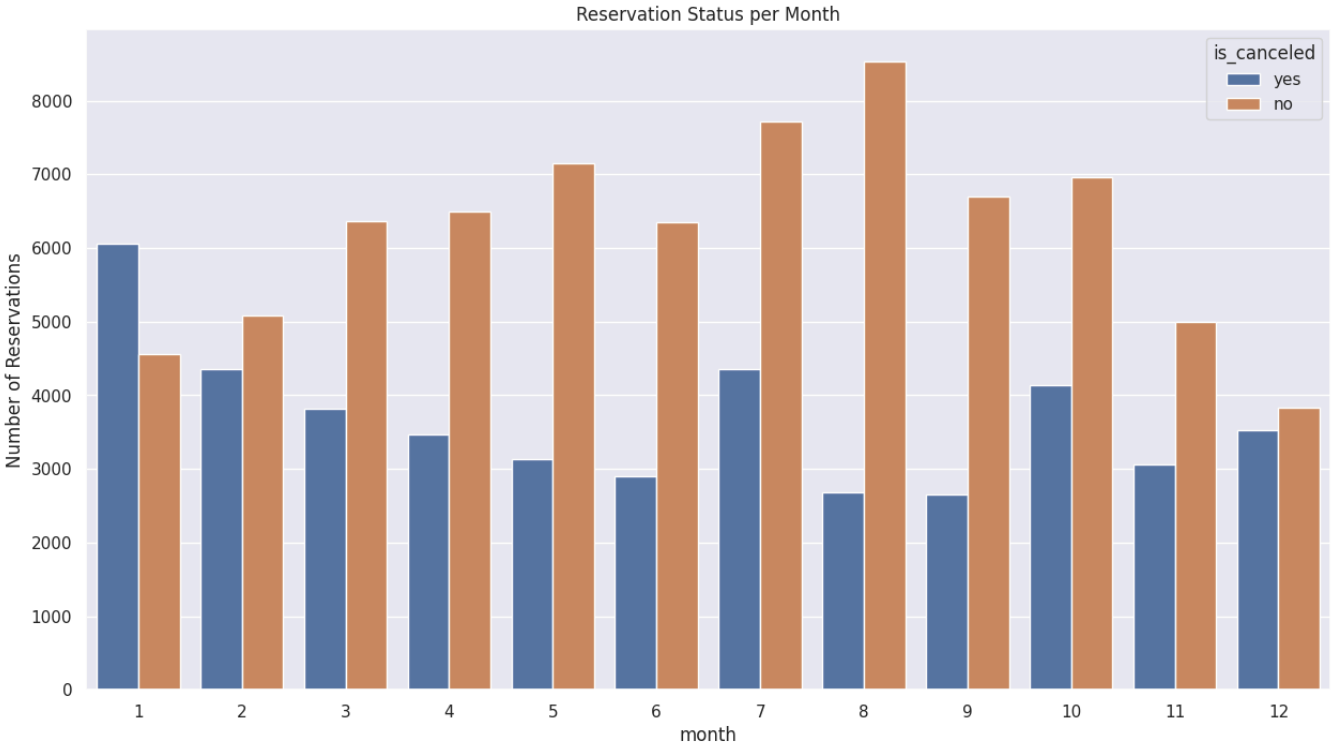

```
#from above pie charts we conclude the cancelation rate is much higher in city hotels compared to resprt hotels.


#showing average adr for both types of hotels vs date using scatter plot
rmean = resort_hotel.groupby('reservation_status_date').agg({"adr":"mean"})
cmean = city_hotel.groupby("reservation_status_date").agg({"adr":"mean"})
```

```
sb.scatterplot(x = rmean.index, y = rmean["adr"], label = "resort hotel")
sb.scatterplot(x = cmean.index, y = cmean["adr"], label = "city hotel")
plt.legend(fontsize = 15)
sb.set(rc = {"figure.figsize": (15,8)})
```



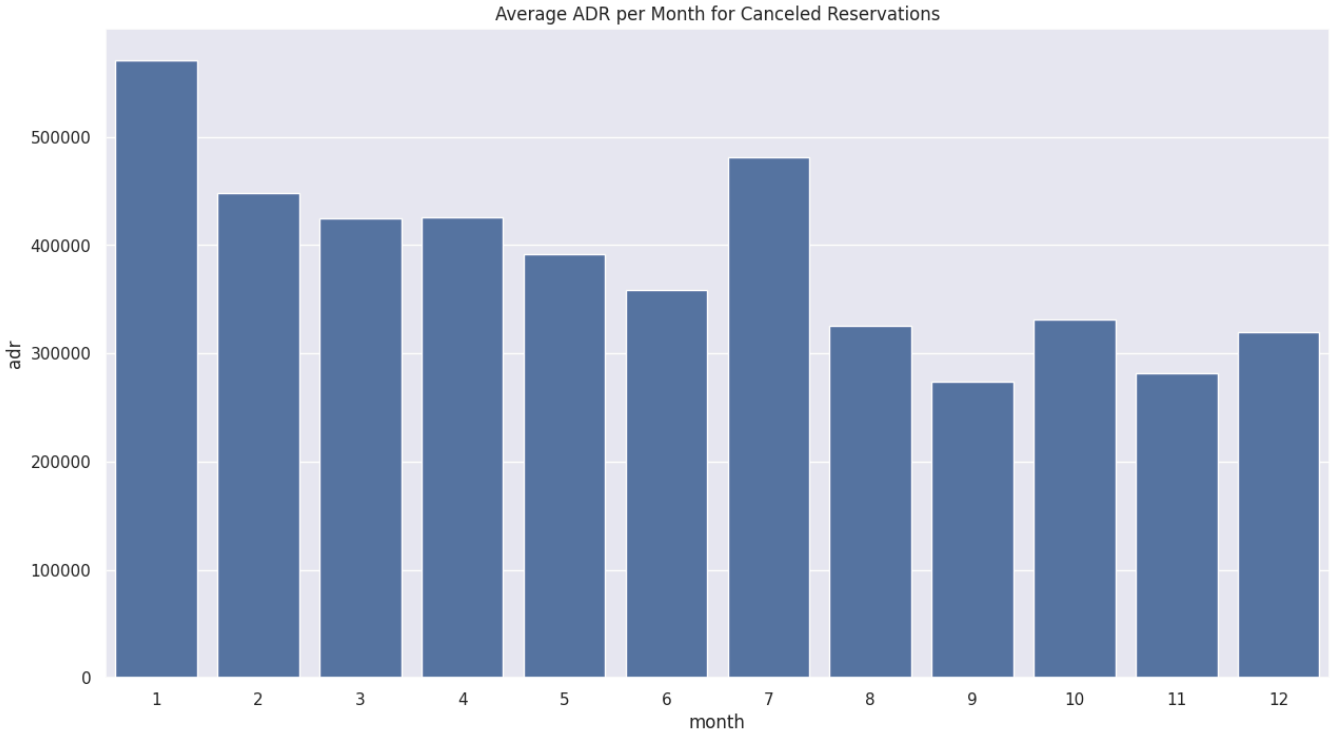

```
#adding month column from reservation_start_date column
df["month"] = df["reservation_status_date"].dt.month


#per month count of canceled and non-canceled reservations
sb.countplot(x = df["month"],hue = "is_canceled", data = df)
plt.title("Reservation Status per Month")
plt.ylabel("Number of Reservations")
```

```
Text(0, 0.5, 'Number of Reservations')
```

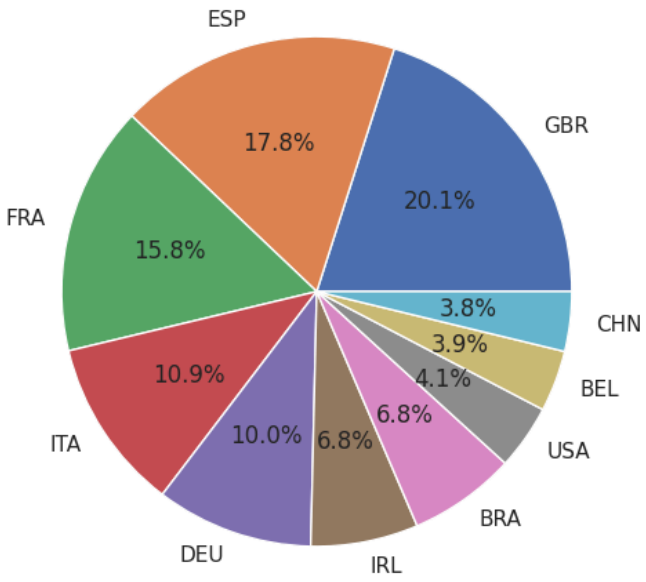


```
#average adr per month for cancelled reservations
data20 = df[df["is_canceled"] == "yes"]
data40 = data20.groupby("month").agg({"adr" : "sum"})
sb.barplot(x = data40.index, y = "adr", data = data40)
plt.title("Average ADR per Month for Canceled Reservations")
```

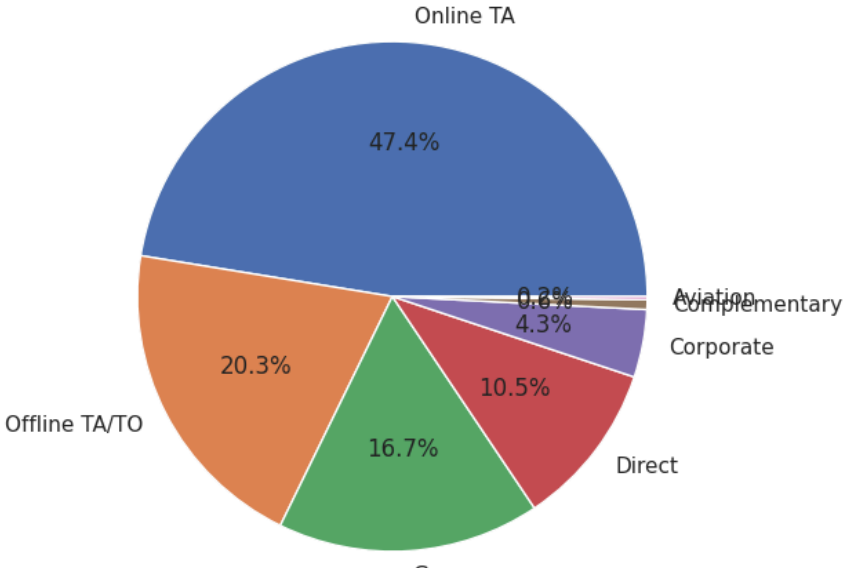
```
Text(0.5, 1.0, 'Average ADR per Month for Canceled Reservations')
```




```
#top 10 countries having highest number of canceled reservations
top_10 = data20["country"].value_counts()
top10 = top_10[1:11]
plt.pie(top10, labels = top10.index, autopct = "%1.1f%%")
sb.set(rc = {"figure.figsize" : (8,6)})
```




```
#number of reservations market segment wise
data100 =  df["market_segment"].value_counts()
plt.pie(data100, labels = data100.index, autopct = "%1.1f%%" )
sb.set(rc = {"figure.figsize": (8,6)})
```



Start coding or generate with AI.

```
#number of canceled reservations market segment wise
data200 = data20["market_segment"].value_counts()
plt.pie(data200, labels = data200.index, autopct = "%1.1f%%")
sb.set(rc = {"figure.figsize": (8,6)})
```