# Hotel_Reservation_Analysis

May 4, 2024

[12]:
```
import pandas as pd
```

[13]:
```
bookings = pd.read_csv('bookings.csv', encoding = 'windows-1251', sep=';')
```

[15]:
```
# We are looking at the first 7 entries
bookings_head = bookings.head(7)
bookings_head
```

[15]:
```
           Hotel  Is Canceled  Lead Time arrival full date  Arrival Date Year  \
0  Resort Hotel            0        342       2015-07-01               2015
1  Resort Hotel            0        737       2015-07-01               2015
2  Resort Hotel            0          7       2015-07-01               2015
3  Resort Hotel            0         13       2015-07-01               2015
4  Resort Hotel            0         14       2015-07-01               2015
5  Resort Hotel            0         14       2015-07-01               2015
6  Resort Hotel            0          0       2015-07-01               2015

  Arrival Date Month  Arrival Date Week Number  Arrival Date Day of Month  \
0               July                        27                          1
1               July                        27                          1
2               July                        27                          1
3               July                        27                          1
4               July                        27                          1
5               July                        27                          1
6               July                        27                          1

   Stays in Weekend nights  Stays in week nights  …  Adults  Children  \
0                        0                     0  …       2       0.0
1                        0                     0  …       2       0.0
2                        0                     1  …       1       0.0
3                        0                     1  …       1       0.0
4                        0                     2  …       2       0.0
5                        0                     2  …       2       0.0
6                        0                     2  …       2       0.0

   Babies  Meal Country Reserved Room Type Assigned room type customer type  \
0       0    BB     PRT                  C                  C     Transient
1       0    BB     PRT                  C                  C     Transient
```

```
2        0     BB     GBR                      A                    C     Transient
3        0     BB     GBR                      A                    A     Transient
4        0     BB     GBR                      A                    A     Transient
5        0     BB     GBR                      A                    A     Transient
6        0     BB     PRT                      C                    C     Transient

  Reservation Status Reservation status_date
0          Check-Out              2015-07-01
1          Check-Out              2015-07-01
2          Check-Out              2015-07-02
3          Check-Out              2015-07-02
4          Check-Out              2015-07-03
5          Check-Out              2015-07-03
6          Check-Out              2015-07-03

[7 rows x 21 columns]
```

[16]:
```
# Replace the spaces with underscores, and put them in lowercase
for column in bookings.columns:
    column_rename = column.replace(' ', '_').lower()
    bookings = bookings.rename(columns={f'{column}':f'{column_rename}'})
```

[20]:
```
bookings.columns
```

[20]:
```
Index(['hotel', 'is_canceled', 'lead_time', 'arrival_full_date',
       'arrival_date_year', 'arrival_date_month', 'arrival_date_week_number',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'stays_total_nights', 'adults', 'children',
       'babies', 'meal', 'country', 'reserved_room_type', 'assigned_room_type',
       'customer_type', 'reservation_status', 'reservation_status_date'],
      dtype='object')
```

[47]:
```
# Users of the countries have made the largest number of successful bookings in␣
 ↪the top 5
bookings.query('is_canceled == 0') \
.groupby('country') \
.agg({'is_canceled':'count'}) \
.sort_values('is_canceled', ascending=False).head()
```

[47]:
```
         is_canceled
country
PRT            21071
GBR             9676
FRA             8481
ESP             6391
DEU             6069
```

[32]:
```
# How many nights do City Hotel type hotels book on average
print(round(bookings.query('hotel == "City Hotel"') \
.agg({'stays_total_nights':'mean'}),2))

# How many nights do Resort Hotel type hotels book on average
print(round(bookings.query('hotel == "Resort Hotel"') \
.agg({'stays_total_nights':'mean'}),2))
```

```
stays_total_nights    2.98
dtype: float64
stays_total_nights    4.32
dtype: float64
```

[33]:
```
# the type of room assigned to the client differs from the one originally␣
 ↪booked due to overbooking, how many such observations?
bookings.query('assigned_room_type != reserved_room_type').
 ↪agg({'reserved_room_type':'count'})
```

[33]:
```
reserved_room_type    14917
dtype: int64
```

[34]:
```
# Which month was the most frequently booked in 2016?
print(bookings.query("arrival_date_year == 2016").groupby('arrival_date_month').
 ↪agg({'arrival_date_month':'count'}).idxmax())

# Which month was the most frequently booked in 2017?
print(bookings.query("arrival_date_year == 2017").groupby('arrival_date_month').
 ↪agg({'arrival_date_month':'count'}).idxmax())
```

```
arrival_date_month    October
dtype: object
arrival_date_month    May
dtype: object
```

[49]:
```
# for which month were City Hotel bookings cancelled most often in 2015? 2016?␣
 ↪2017?
bookings.query('hotel == "City Hotel" and is_canceled == 1').
 ↪groupby('arrival_date_year')['arrival_date_month'].value_counts()
```

[49]:
```
arrival_date_year  arrival_date_month
2015               September             1543
                   October               1321
                   August                1232
                   July                   939
                   December               668
                   November               301
2016               October               1947
                   June                  1720
                   September          1567
                   April              1539
                   May                1436
                   November           1360
                   August             1247
                   March              1108
                   December           1072
                   July               1043
                   February            930
                   January             438
2017               May                2217
                   April              1926
                   June               1808
                   July               1324
                   March              1278
                   August             1123
                   January            1044
                   February            971
Name: count, dtype: int64
```

[36]:
```
# # Will look at the numerical characteristics of three columns: adults,␣
  ↪children and babies. Which one has the highest average value?
bookings.agg({'adults':'mean', 'children':'mean','babies':'mean'}).idxmax()
```

[36]:
```
'adults'
```

[37]:
```
# Create total_kids by combining the children and babies columns.
bookings['total_kids'] = bookings.children + bookings.babies
```

[38]:
```
#                                           ?
round(bookings.groupby('hotel').agg({'total_kids': 'mean'}),2)
```

[38]:
```
              total_kids
hotel
City Hotel          0.10
Resort Hotel        0.14
```

[41]:
```
# Not all bookings were completed successfully, how many customers were lost in␣
  ↪the process?
bookings['has_kids'] = bookings.total_kids > 0
bookings['has_kids'].value_counts()
```

[41]:
```
has_kids
False    110058
True       9332
Name: count, dtype: int64
```

[50]:
```
# let's check which user group has a higher churn rate with or without children
print(round(bookings.query('has_kids == True and is_canceled == 1').shape[0] /␣
 ↪bookings.query('has_kids == True').shape[0] * 100,2))
print(round(bookings.query('has_kids == False and is_canceled == 1').shape[0] /␣
 ↪bookings.query('has_kids == False').shape[0] * 100,2))
```

```
34.92
37.22
```

[ ]:
```

```