

# Hotel Booking Demand

By: Jessica

Last updated : 29 January 2022

# Data preview

The hotel booking demand data is consists of 32 columns, and 119,390 rows, here are the description of each column:

#	Column name	Description
1	Hotel	Hotel (H1 = Resort Hotel or H2 = City Hotel)
2	is_canceled	Value indicating if the booking was canceled (1) or not (0)
3	lead_time	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
4	arrival_date_year	Year of arrival date
5	arrival_date_month	Month of arrival date
6	arrival_date_week_number	Week number of year for arrival date
7	arrival_date_day_of_month	Day of arrival date
8	stays_in_weekend_nights	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel

#	Column name	Description
9	stays_in_week_nights	Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
10	adults	Number of adults
11	children	Number of children
12	babies	Number of babies
13	meal	Type of meal booked. Categories are presented in standard hospitality meal packages: Undefined/SC – no meal package; BB – Bed & Breakfast; HB – Half board (breakfast and one other meal – usually dinner); FB – Full board (breakfast, lunch and dinner)
14	market_segment	Market segment designation. In categories, the term "TA" means "Travel Agents" and "TO" means "Tour Operators"
15	distribution_channel	Booking distribution channel. The term "TA" means "Travel Agents" and "TO" means "Tour Operators"
16	is_repeated_guest	Value indicating if the booking name was from a repeated guest (1) or not (0)

# Data preview

The hotel booking demand data is consists of 32 columns, and 119,390 rows, here are the description of each column:

#	Column name	Description
17	previous_cancellations	Number of previous bookings that were cancelled by the customer prior to the current booking
18	previous_bookings_not_canceled	Number of previous bookings not cancelled by the customer prior to the current booking
19	reserved_room_type	Code of room type reserved. Code is presented instead of designation for anonymity reasons
20	assigned_room_type	Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons.
21	booking_changes	Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation
22	deposit_type	Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories: No Deposit – no deposit was made; Non Refund – a deposit was made in the value of the total stay cost; Refundable – a deposit was made with a value under the total cost of stay.

#	Column name	Description
23	days_in_waiting_list	Number of days the booking was in the waiting list before it was confirmed to the customer
24	customer_type	Type of booking, assuming one of four categories: Contract - when the booking has an allotment or other type of contract associated to it; Group – when the booking is associated to a group; Transient – when the booking is not part of a group or contract, and is not associated to other transient booking; Transient-party – when the booking is transient, but is associated to at least other transient booking
25	adr	Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights
26	required_car_parking_spaces	Number of car parking spaces required by the customer
27	total_of_special_requests	Number of special requests made by the customer (e.g. twin bed or high floor)
28	reservation_status	Reservation last status, assuming one of three categories: Canceled – booking was canceled by the customer; Check-Out – customer has checked in but already departed; No-Show – customer did not check-in and did inform the hotel of the reason why
29	reservation_status_date	Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to understand when was the booking canceled or when did the customer checked-out of the hotel

This data is retrieved from :<https://www.kaggle.com/jessemostipak/hotel-booking-demand>


# Data preview

The hotel booking demand data is consists of 32 columns, and 119,390 rows, here are the description of each column:

#	Column name	Description
30	agent	ID of the travel agency that made the booking
31	company	ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons
32	Country	Country of origin. Categories are represented in the ISO 3155-3:2013 format


# Treating Missing Values

```
hotel_null = hotel.isna().sum()  
print(hotel_null)
```



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

```
hotel = hotel.dropna()  
print(hotel.isna().sum())
```



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
agent	0
company	0

## Treating Duplicated Values

```
print(hotel.duplicated().sum())  
hotel=hotel.drop_duplicates()  
print(hotel.shape)
```

28  
(189, 32)

# of rows changes  
in every process

119,390



```
hotel = hotel.dropna()
```

217



```
hotel=hotel.drop_duplicates()
```

189

# Data type

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

children data types can not be an float since it is the number of children, it has to be an integer.

```
hotel['children']=hotel['children'].astype('int64')
```

The agent and company column represent the ID of the agent and company so it can not be in float but in string (object)

```
hotel['agent']=hotel['agent'].astype('object')  
hotel['company']=hotel['company'].astype('object')
```

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	int64
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	object
company	object
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



# Date

Since the month is in string format, we need to convert them into datetime format so then we can combine it with the month and day.

```
month_df = hotel['arrival_date_month']  
hotel['month'] = pd.to_datetime(month_df, format="%B").dt.month
```

Combine them into one column named arrival\_date.

```
dict_date= {'day': hotel['arrival_date_day_of_month'],  
            'month': hotel['month'],  
            'year': hotel['arrival_date_year']}  
  
hotel['arrival_date']=pd.to_datetime(dict_date)
```

Drop the combined columns

```
hotel = hotel.drop(columns=['arrival_date_year', 'arrival_date_day_of_month', 'arrival_date_month', 'month'])  
print(hotel.columns)
```

arrival_date_year	int64
arrival_date_month	object
arrival_date_week_number	int64
arrival_date_day_of_month	int64



arrival_date	datetime64[ns]
--------------	----------------

## Question to answer

What is the busiest week of the hotel and how does it affect the hotel services?

# What is the busiest week based on arrival date?

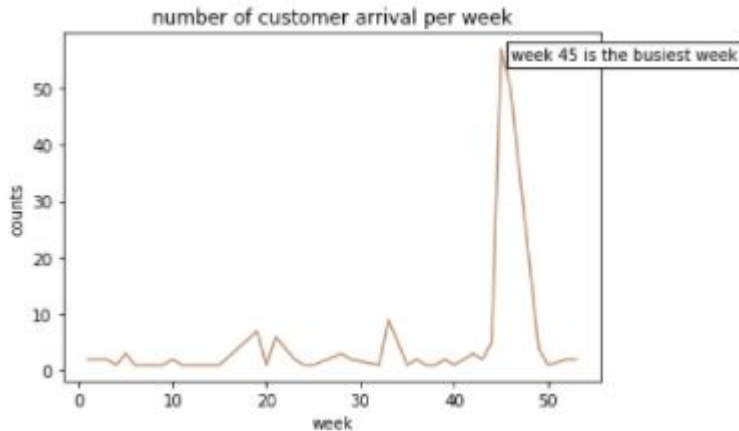
filter the data that we want to use to plot

```
num_res = hotel['arrival_date_week_number'].value_counts().rename_axis('week').reset_index(name='counts')
print(num_res.head())
```

	week	counts
0	45	57
1	46	50
2	33	9
3	19	7
4	21	6

create the visualization.

```
sns.lineplot(x=week_res['week'],y=week_res['counts'],color='#8F9270')
plt.title('number of customer arrival per week')
t= ("week 45 is "
    "the busiest week")
plt.text(46,55,t,fontsize=10,ha='left',wrap=True,bbox=dict(boxstyle="square,pad=0.3",fc="white"))
```



# What happen during the busiest week ?

**42% of the room assigned on the 46th week are different than the room that are reserved**

since the hotel has been busy on the 45th week, there might be a chance that the number of room available are not sustaining the 46th week reservation.

```
hotel.loc[hotel['reserved_room_type']==hotel['assigned_room_type'],'is_aligned'] = True
hotel.loc[hotel['reserved_room_type']!=hotel['assigned_room_type'],'is_aligned'] = False
hotel.loc[hotel['is_aligned']==False,'arrival_date_week_number'].value_counts()
```

46	21	46	0.411765
33	4	33	0.078431
44	4	44	0.078431
45	4	45	0.078431
21	3	21	0.058824
3	2	3	0.039216
5	2	5	0.039216
42	2	42	0.039216
49	2	49	0.039216
52	2	52	0.039216
1	1	1	0.019608
15	1	15	0.019608
19	1	19	0.019608
28	1	28	0.019608
39	1	39	0.019608

Name: arrival\_date\_week\_number, dtype: int64    Name: arrival\_date\_week\_number, dtype: float64

\*total arrival on the 46th week are 50, 21 of them are assigned to the different room ,  $21/50 \times 100 = 0,42$

to know which week have the most number of cancelation

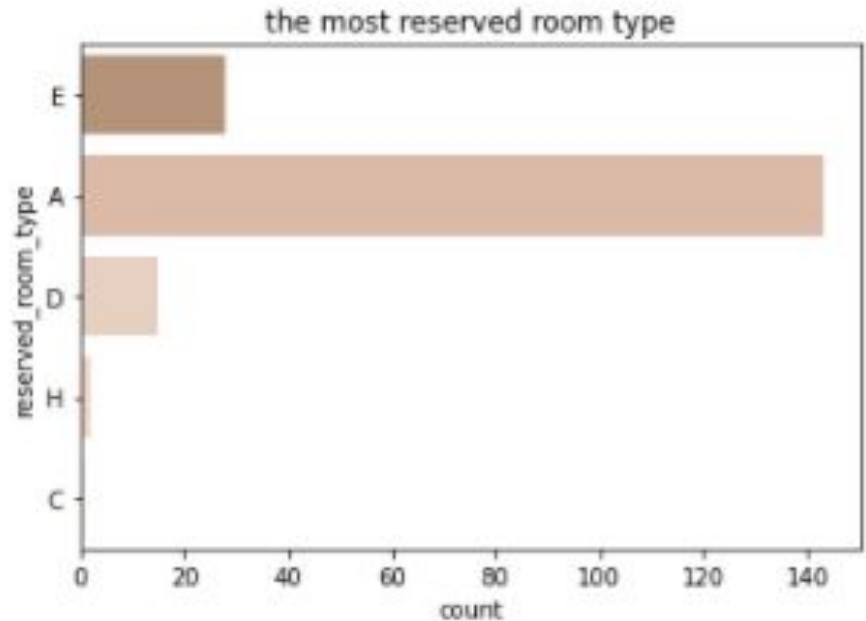
```
canceled = hotel.loc[hotel['is_canceled']== 1, 'arrival_date_week_number'].value_counts().rename_axis('week').reset_index(name='n')  
print(canceled)
```

	week	num_canceled
0	46	10
1	33	1
2	3	1
3	20	1
4	25	1
5	45	1
6	15	1

**Week 46th has the most number of cancelation.**

What is the most reserved room type?

```
pal= {'E': '#BF9270', 'A': '#E3B7A0', 'D': '#EDCDBB', 'H': '#FDCEB9', 'C': '#FFEDDB'}  
sns.countplot(y=hotel['reserved_room_type'], palette=pal)  
plt.title('the most reserved room type')  
plt.show()
```



## What kind of room type reserved for the 46th week ?

to select with condition where week is week 46, and what type of room has been reserved

```
hotel.loc[hotel['arrival_date_week_number']== 46,'reserved_room_type'].value_counts()
```

```
A    41  
E     9  
Name: reserved_room_type, dtype: int64
```

to select with condition where week is week 46, and what type of room has been assigned

```
hotel.loc[hotel['arrival_date_week_number']== 46,'assigned_room_type'].value_counts()
```

```
A    21  
E    11  
C     8  
D     7  
F     2  
B     1  
Name: assigned_room_type, dtype: int64
```