

# EDA

July 30, 2021

```
[1]: pip install folium
```

```
Collecting folium
  Using cached folium-0.12.1-py2.py3-none-any.whl (94 kB)
Collecting branca>=0.3.0
  Using cached branca-0.4.2-py3-none-any.whl (24 kB)
Requirement already satisfied: requests in /opt/conda/lib/python3.7/site-packages (from folium) (2.23.0)
Requirement already satisfied: Jinja2>=2.9 in /opt/conda/lib/python3.7/site-packages (from folium) (2.11.2)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages (from folium) (1.18.4)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.7/site-packages (from requests->folium) (2020.4.5.2)
Requirement already satisfied: chardet<4,>=3.0.2 in /opt/conda/lib/python3.7/site-packages (from requests->folium) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-packages (from requests->folium) (2.9)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /opt/conda/lib/python3.7/site-packages (from requests->folium) (1.25.9)
Requirement already satisfied: MarkupSafe>=0.23 in /opt/conda/lib/python3.7/site-packages (from Jinja2>=2.9->folium) (1.1.1)
Installing collected packages: branca, folium
Successfully installed branca-0.4.2 folium-0.12.1
Note: you may need to restart the kernel to use updated packages.
```

```
[2]: pip install missingno
```

```
Collecting missingno
  Using cached missingno-0.5.0-py3-none-any.whl (8.8 kB)
Requirement already satisfied: seaborn in /opt/conda/lib/python3.7/site-packages (from missingno) (0.10.1)
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.7/site-packages (from missingno) (3.2.1)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages (from missingno) (1.18.4)
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages (from missingno) (1.4.1)
```

```
Requirement already satisfied: pandas>=0.22.0 in /opt/conda/lib/python3.7/site-  
packages (from seaborn->missingno) (1.0.3)  
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-  
packages (from matplotlib->missingno) (0.10.0)  
Requirement already satisfied: python-dateutil>=2.1 in  
/opt/conda/lib/python3.7/site-packages (from matplotlib->missingno) (2.8.1)  
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in  
/opt/conda/lib/python3.7/site-packages (from matplotlib->missingno) (2.4.7)  
Requirement already satisfied: kiwisolver>=1.0.1 in  
/opt/conda/lib/python3.7/site-packages (from matplotlib->missingno) (1.2.0)  
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-  
packages (from pandas>=0.22.0->seaborn->missingno) (2020.1)  
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages  
(from cycler>=0.10->matplotlib->missingno) (1.14.0)  
Installing collected packages: missingno  
Successfully installed missingno-0.5.0  
Note: you may need to restart the kernel to use updated packages.
```

```
[3]: pip install plotly
```

```
Collecting plotly  
  Using cached plotly-5.1.0-py2.py3-none-any.whl (20.6 MB)  
Collecting tenacity>=6.2.0  
  Using cached tenacity-8.0.1-py3-none-any.whl (24 kB)  
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages  
(from plotly) (1.14.0)  
Installing collected packages: tenacity, plotly  
Successfully installed plotly-5.1.0 tenacity-8.0.1  
Note: you may need to restart the kernel to use updated packages.
```

```
[4]: pip install xgboost
```

```
Collecting xgboost  
  Using cached xgboost-1.4.2-py3-none-manylinux2010_x86_64.whl (166.7 MB)  
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages  
(from xgboost) (1.18.4)  
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages  
(from xgboost) (1.4.1)  
Installing collected packages: xgboost  
Successfully installed xgboost-1.4.2  
Note: you may need to restart the kernel to use updated packages.
```

```
[5]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
import missingno as msno  
import xgboost as xgb
```



0	July	27
1	July	27
2	July	27
3	July	27
4	July	27
...	...	...
119385	August	35
119386	August	35
119387	August	35
119388	August	35
119389	August	35

	arrival_date_day_of_month	stays_in_weekend_nights	\
0	1	0	
1	1	0	
2	1	0	
3	1	0	
4	1	0	
...	...	...	
119385	30	2	
119386	31	2	
119387	31	2	
119388	31	2	
119389	29	2	

	stays_in_week_nights	adults	children	babies	meal	country	\
0	0	2	0.0	0	BB	PRT	
1	0	2	0.0	0	BB	PRT	
2	1	1	0.0	0	BB	GBR	
3	1	1	0.0	0	BB	GBR	
4	2	2	0.0	0	BB	GBR	
...	...	...	...	...	...	...	
119385	5	2	0.0	0	BB	BEL	
119386	5	3	0.0	0	BB	FRA	
119387	5	2	0.0	0	BB	DEU	
119388	5	2	0.0	0	BB	GBR	
119389	7	2	0.0	0	HB	DEU	

	market_segment	distribution_channel	is_repeated_guest	\
0	Direct	Direct	0	
1	Direct	Direct	0	
2	Direct	Direct	0	
3	Corporate	Corporate	0	
4	Online TA	TA/TO	0	
...	...	...	...	
119385	Offline TA/TO	TA/TO	0	
119386	Online TA	TA/TO	0	
119387	Online TA	TA/TO	0	

119388	Online TA	TA/TO	0
119389	Online TA	TA/TO	0

	previous_cancellations	previous_bookings_not_canceled	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
...	...	...	
119385	0	0	
119386	0	0	
119387	0	0	
119388	0	0	
119389	0	0	

	reserved_room_type	assigned_room_type	booking_changes	deposit_type	\
0	C	C	3	No Deposit	
1	C	C	4	No Deposit	
2	A	C	0	No Deposit	
3	A	A	0	No Deposit	
4	A	A	0	No Deposit	
...	...	...	...	...	
119385	A	A	0	No Deposit	
119386	E	E	0	No Deposit	
119387	D	D	0	No Deposit	
119388	A	A	0	No Deposit	
119389	A	A	0	No Deposit	

	agent	company	days_in_waiting_list	customer_type	adr	\
0	NaN	NaN	0	Transient	0.00	
1	NaN	NaN	0	Transient	0.00	
2	NaN	NaN	0	Transient	75.00	
3	304.0	NaN	0	Transient	75.00	
4	240.0	NaN	0	Transient	98.00	
...	...	...	...	...	...	
119385	394.0	NaN	0	Transient	96.14	
119386	9.0	NaN	0	Transient	225.43	
119387	9.0	NaN	0	Transient	157.71	
119388	89.0	NaN	0	Transient	104.40	
119389	9.0	NaN	0	Transient	151.20	

	required_car_parking_spaces	total_of_special_requests	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	1	

```

...
119385      0      0
119386      0      2
119387      0      4
119388      0      0
119389      0      2

```

```

      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
...
119385      Check-Out      2017-09-06
119386      Check-Out      2017-09-07
119387      Check-Out      2017-09-07
119388      Check-Out      2017-09-07
119389      Check-Out      2017-09-07

```

[119390 rows x 32 columns]

## 1 New Section

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

```

[7]:      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  children  babies \
0                0                0        2        0.0        0
1                0                0        2        0.0        0
2                0                1        1        0.0        0
3                0                1        1        0.0        0
4                0                2        2        0.0        0

```

	meal	country	market_segment	distribution_channel	is_repeated_guest	\
0	BB	PRT	Direct	Direct	0	
1	BB	PRT	Direct	Direct	0	
2	BB	GBR	Direct	Direct	0	
3	BB	GBR	Corporate	Corporate	0	
4	BB	GBR	Online TA	TA/TO	0	

	previous_cancellations	previous_bookings_not_canceled	reserved_room_type	\
0	0	0	C	
1	0	0	C	
2	0	0	A	
3	0	0	A	
4	0	0	A	

	assigned_room_type	booking_changes	deposit_type	agent	company	\
0	C	3	No Deposit	NaN	NaN	
1	C	4	No Deposit	NaN	NaN	
2	C	0	No Deposit	NaN	NaN	
3	A	0	No Deposit	304.0	NaN	
4	A	0	No Deposit	240.0	NaN	

	days_in_waiting_list	customer_type	adr	required_car_parking_spaces	\
0	0	Transient	0.0	0	
1	0	Transient	0.0	0	
2	0	Transient	75.0	0	
3	0	Transient	75.0	0	
4	0	Transient	98.0	0	

	total_of_special_requests	reservation_status	reservation_status_date
0	0	Check-Out	2015-07-01
1	0	Check-Out	2015-07-01
2	0	Check-Out	2015-07-02
3	0	Check-Out	2015-07-02
4	1	Check-Out	2015-07-03

```
[8]: df.shape
```

```
[8]: (119390, 32)
```

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

```
[9]:
```

	is_canceled	lead_time	arrival_date_year	\
count	119390.000000	119390.000000	119390.000000	
mean	0.370416	104.011416	2016.156554	
std	0.482918	106.863097	0.707476	
min	0.000000	0.000000	2015.000000	

25%	0.000000	18.000000	2016.000000
50%	0.000000	69.000000	2016.000000
75%	1.000000	160.000000	2017.000000
max	1.000000	737.000000	2017.000000

	arrival_date_week_number	arrival_date_day_of_month	\
count	119390.000000	119390.000000	
mean	27.165173	15.798241	
std	13.605138	8.780829	
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	

	stays_in_weekend_nights	stays_in_week_nights	adults	\
count	119390.000000	119390.000000	119390.000000	
mean	0.927599	2.500302	1.856403	
std	0.998613	1.908286	0.579261	
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	19.000000	50.000000	55.000000	

	children	babies	is_repeated_guest	\
count	119386.000000	119390.000000	119390.000000	
mean	0.103890	0.007949	0.031912	
std	0.398561	0.097436	0.175767	
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	

	previous_cancellations	previous_bookings_not_canceled	\
count	119390.000000	119390.000000	
mean	0.087118	0.137097	
std	0.844336	1.497437	
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	

	booking_changes	agent	company	days_in_waiting_list	\
count	119390.000000	103050.000000	6797.000000	119390.000000	



mean	0.221124	86.693382	189.266735	2.321149
std	0.652306	110.774548	131.655015	17.594721
min	0.000000	1.000000	6.000000	0.000000
25%	0.000000	9.000000	62.000000	0.000000
50%	0.000000	14.000000	179.000000	0.000000
75%	0.000000	229.000000	270.000000	0.000000
max	21.000000	535.000000	543.000000	391.000000

	adr	required_car_parking_spaces	total_of_special_requests
count	119390.000000	119390.000000	119390.000000
mean	101.831122	0.062518	0.571363
std	50.535790	0.245291	0.792798
min	-6.380000	0.000000	0.000000
25%	69.290000	0.000000	0.000000
50%	94.575000	0.000000	0.000000
75%	126.000000	0.000000	1.000000
max	5400.000000	8.000000	5.000000

37 % of the people have cancelled their booking as per the dataset. Avg. lead time is 104 days, that is almost 3.5 months. Each booking has on an average 1.8 adults and 0.1 children. Only 3% of the guests are repeated. Median lead time is 69 days.

## MAJOR OBSERVATIONS:

- 1.Number of bookings made were highest in the month of July and August and lowest in January.
- 2.Bookings were more for the City hotel than the Resort hotel. 3.41.7% of the total bookings were cancelled for City hotel and 21.7% for the Resort hotel.
- 4.Number of days that elapsed between the entering date of the booking and the arrival date is less for the people who cancelled.
- 5.As the hotels are in Portugal Europe, the bookings are mostly with European countries, Highest is Portugal with 48.59k bookings.
- 6.77% of the bookings are made with bed and breakfast.
- 7.Only 3% are repeated guests.

## EXPLORATORY DATA ANALYSIS

```
[10]: # dealing with null values
null = pd.DataFrame({'Count of Missing values' : df.isna().sum(), 'Percentage_
↳of missing values' : (df.isna().sum()) / (df.shape[0]) * (100)})
null
```

```
[10]: Count of Missing values \
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

	Percentage of missing values
hotel	0.000000
is_canceled	0.000000
lead_time	0.000000
arrival_date_year	0.000000
arrival_date_month	0.000000
arrival_date_week_number	0.000000
arrival_date_day_of_month	0.000000
stays_in_weekend_nights	0.000000
stays_in_week_nights	0.000000
adults	0.000000
children	0.003350
babies	0.000000
meal	0.000000
country	0.408744
market_segment	0.000000
distribution_channel	0.000000
is_repeated_guest	0.000000
previous_cancellations	0.000000
previous_bookings_not_canceled	0.000000
reserved_room_type	0.000000
assigned_room_type	0.000000
booking_changes	0.000000

deposit_type	0.000000
agent	13.686238
company	94.306893
days_in_waiting_list	0.000000
customer_type	0.000000
adr	0.000000
required_car_parking_spaces	0.000000
total_of_special_requests	0.000000
reservation_status	0.000000
reservation_status_date	0.000000

There are 32 columns, 12 were Categorical and 20 Numerical There are 4 columns with the missing values namely- country, agent, company, children 'company' column has maximum null values which is 94

```
[11]: hotel = df.drop(columns=['company'])
      hotel
```

```
[11]:
```

	hotel	is_canceled	lead_time	arrival_date_year	\
0	Resort Hotel	0	342	2015	
1	Resort Hotel	0	737	2015	
2	Resort Hotel	0	7	2015	
3	Resort Hotel	0	13	2015	
4	Resort Hotel	0	14	2015	
...	...	...	...	...	
119385	City Hotel	0	23	2017	
119386	City Hotel	0	102	2017	
119387	City Hotel	0	34	2017	
119388	City Hotel	0	109	2017	
119389	City Hotel	0	205	2017	

	arrival_date_month	arrival_date_week_number	\
0	July	27	
1	July	27	
2	July	27	
3	July	27	
4	July	27	
...	...	...	
119385	August	35	
119386	August	35	
119387	August	35	
119388	August	35	
119389	August	35	

	arrival_date_day_of_month	stays_in_weekend_nights	\
0	1	0	
1	1	0	

2	1	0
3	1	0
4	1	0
...	...	...
119385	30	2
119386	31	2
119387	31	2
119388	31	2
119389	29	2

	stays_in_week_nights	adults	children	babies	meal	country	\
0	0	2	0.0	0	BB	PRT	
1	0	2	0.0	0	BB	PRT	
2	1	1	0.0	0	BB	GBR	
3	1	1	0.0	0	BB	GBR	
4	2	2	0.0	0	BB	GBR	
...	...	...	...	...	...	...	
119385	5	2	0.0	0	BB	BEL	
119386	5	3	0.0	0	BB	FRA	
119387	5	2	0.0	0	BB	DEU	
119388	5	2	0.0	0	BB	GBR	
119389	7	2	0.0	0	HB	DEU	

	market_segment	distribution_channel	is_repeated_guest	\
0	Direct	Direct	0	
1	Direct	Direct	0	
2	Direct	Direct	0	
3	Corporate	Corporate	0	
4	Online TA	TA/TO	0	
...	...	...	...	
119385	Offline TA/TO	TA/TO	0	
119386	Online TA	TA/TO	0	
119387	Online TA	TA/TO	0	
119388	Online TA	TA/TO	0	
119389	Online TA	TA/TO	0	

	previous_cancellations	previous_bookings_not_canceled	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
...	...	...	
119385	0	0	
119386	0	0	
119387	0	0	
119388	0	0	

119389

0

0

	reserved_room_type	assigned_room_type	booking_changes	deposit_type	\
0	C	C	3	No Deposit	
1	C	C	4	No Deposit	
2	A	C	0	No Deposit	
3	A	A	0	No Deposit	
4	A	A	0	No Deposit	
...	...	...	...	...	
119385	A	A	0	No Deposit	
119386	E	E	0	No Deposit	
119387	D	D	0	No Deposit	
119388	A	A	0	No Deposit	
119389	A	A	0	No Deposit	

	agent	days_in_waiting_list	customer_type	adr	\
0	NaN	0	Transient	0.00	
1	NaN	0	Transient	0.00	
2	NaN	0	Transient	75.00	
3	304.0	0	Transient	75.00	
4	240.0	0	Transient	98.00	
...	...	...	...	...	
119385	394.0	0	Transient	96.14	
119386	9.0	0	Transient	225.43	
119387	9.0	0	Transient	157.71	
119388	89.0	0	Transient	104.40	
119389	9.0	0	Transient	151.20	

	required_car_parking_spaces	total_of_special_requests	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	1	
...	...	...	
119385	0	0	
119386	0	2	
119387	0	4	
119388	0	0	
119389	0	2	

	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

```

...
119385      Check-Out      2017-09-06
119386      Check-Out      2017-09-07
119387      Check-Out      2017-09-07
119388      Check-Out      2017-09-07
119389      Check-Out      2017-09-07

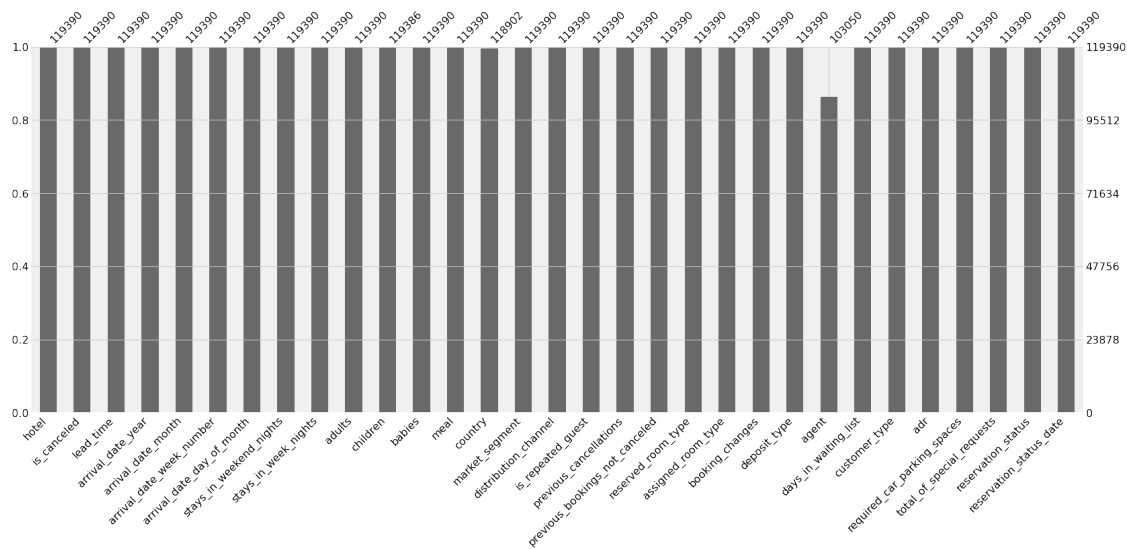
```

```
[119390 rows x 31 columns]
```

```

[12]: #Lets use Missingno library which offers a fair visualization of the
      ↪distribution of NaN values.
      msno.bar(hotel)
      plt.show()

```



We have almost 120,000 observations, its kind of difficult to make any observation regarding the columns containing NaN values. So, we shall check the distribution of these columns individually.

```
[13]: hotel['children'].value_counts()
```

```

[13]: 0.0      110796
      1.0       4861
      2.0       3652
      3.0         76
      10.0         1
      Name: children, dtype: int64

```

```

[14]: hotel['children'].fillna(0,inplace=True) #In order to deal with the missing
      ↪information in children's column, we fill it with 0 as we see maximum
      ↪travellers had 0 children

```

```
[15]: hotel['country'].value_counts()
```

```
[15]: PRT      48590
      GBR      12129
      FRA      10415
      ESP       8568
      DEU       7287
      ...
      PYF         1
      UMI         1
      MMR         1
      HND         1
      VGB         1
      Name: country, Length: 177, dtype: int64
```

```
[16]: hotel['country'].fillna(hotel['country'].mode()[0], inplace=True) # Since, only
      ↳ 0.4% rows are missing from 'country' column we shall replace it using its
      ↳ mode value
```

```
[17]: hotel['agent'].value_counts()
```

```
[17]: 9.0         31961
      240.0       13922
      1.0         7191
      14.0        3640
      7.0         3539
      ...
      213.0         1
      433.0         1
      197.0         1
      367.0         1
      337.0         1
      Name: agent, Length: 333, dtype: int64
```

```
[18]: hotel['agent'].fillna(0,inplace=True) # For the sake of simplicity, we shall
      ↳ replace the 13% Nan values in column agent with '0'
```

```
[19]: #Rechecking if the null values are handled properly
      missing = pd.DataFrame({'Count of Missing values' : hotel.isna().sum()})
      missing
```

```
[19]:
```

	Count of Missing values
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
agent                         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

```

```

[20]: # There are a few rows where number of adults is zero, Hence, trying to remove
      ↪ such rows
filter = (hotel.children == 0) & (hotel.adults == 0) & (hotel.babies == 0)
hotel[filter]

```

```

[20]:
      hotel  is_canceled  lead_time  arrival_date_year  \
2224  Resort Hotel      0         1         2015
2409  Resort Hotel      0         0         2015
3181  Resort Hotel      0        36         2015
3684  Resort Hotel      0       165         2015
3708  Resort Hotel      0       165         2015
...      ...      ...      ...      ...
115029  City Hotel      0       107         2017
115091  City Hotel      0         1         2017
116251  City Hotel      0        44         2017
116534  City Hotel      0         2         2017
117087  City Hotel      0       170         2017

      arrival_date_month  arrival_date_week_number  \
2224      October      41

```



2409	October	42
3181	November	47
3684	December	53
3708	December	53
...	...	...
115029	June	26
115091	June	26
116251	July	28
116534	July	28
117087	July	30

	arrival_date_day_of_month	stays_in_weekend_nights	\
2224	6	0	
2409	12	0	
3181	20	1	
3684	30	1	
3708	30	2	
...	...	...	
115029	27	0	
115091	30	0	
116251	15	1	
116534	15	2	
117087	27	0	

	stays_in_week_nights	adults	children	babies	meal	country	\
2224	3	0	0.0	0	SC	PRT	
2409	0	0	0.0	0	SC	PRT	
3181	2	0	0.0	0	SC	ESP	
3684	4	0	0.0	0	SC	PRT	
3708	4	0	0.0	0	SC	PRT	
...	...	...	...	...	...	...	
115029	3	0	0.0	0	BB	CHE	
115091	1	0	0.0	0	SC	PRT	
116251	1	0	0.0	0	SC	SWE	
116534	5	0	0.0	0	SC	RUS	
117087	2	0	0.0	0	BB	BRA	

	market_segment	distribution_channel	is_repeated_guest	\
2224	Corporate	Corporate	0	
2409	Corporate	Corporate	0	
3181	Groups	TA/TO	0	
3684	Groups	TA/TO	0	
3708	Groups	TA/TO	0	
...	...	...	...	
115029	Online TA	TA/TO	0	
115091	Complementary	Direct	0	
116251	Online TA	TA/TO	0	

116534	Online TA	TA/TO	0
117087	Offline TA/TO	TA/TO	0

	previous_cancellations	previous_bookings_not_canceled	\
2224	0	0	
2409	0	0	
3181	0	0	
3684	0	0	
3708	0	0	
...	...	...	
115029	0	0	
115091	0	0	
116251	0	0	
116534	0	0	
117087	0	0	

	reserved_room_type	assigned_room_type	booking_changes	deposit_type	\
2224	A	I	1	No Deposit	
2409	A	I	0	No Deposit	
3181	A	C	0	No Deposit	
3684	A	A	1	No Deposit	
3708	A	C	1	No Deposit	
...	...	...	...	...	
115029	A	A	1	No Deposit	
115091	E	K	0	No Deposit	
116251	A	K	2	No Deposit	
116534	A	K	1	No Deposit	
117087	A	A	0	No Deposit	

	agent	days_in_waiting_list	customer_type	adr	\
2224	0.0	0	Transient-Party	0.00	
2409	0.0	0	Transient	0.00	
3181	38.0	0	Transient-Party	0.00	
3684	308.0	122	Transient-Party	0.00	
3708	308.0	122	Transient-Party	0.00	
...	...	...	...	...	
115029	7.0	0	Transient	100.80	
115091	0.0	0	Transient	0.00	
116251	425.0	0	Transient	73.80	
116534	9.0	0	Transient-Party	22.86	
117087	52.0	0	Transient	0.00	

	required_car_parking_spaces	total_of_special_requests	\
2224	0	0	
2409	0	0	
3181	0	0	
3684	0	0	

3708	0	0
...	...	...
115029	0	0
115091	1	1
116251	0	0
116534	0	1
117087	0	0

	reservation_status	reservation_status_date
2224	Check-Out	2015-10-06
2409	Check-Out	2015-10-12
3181	Check-Out	2015-11-23
3684	Check-Out	2016-01-04
3708	Check-Out	2016-01-05
...	...	...
115029	Check-Out	2017-06-30
115091	Check-Out	2017-07-01
116251	Check-Out	2017-07-17
116534	Check-Out	2017-07-22
117087	Check-Out	2017-07-29

[180 rows x 31 columns]

```
[21]: #Removing these rows with 0 adults, 0 children and babies
hotel = hotel[~filter]
hotel
```

```
[21]:
```

	hotel	is_canceled	lead_time	arrival_date_year	\
0	Resort Hotel	0	342	2015	
1	Resort Hotel	0	737	2015	
2	Resort Hotel	0	7	2015	
3	Resort Hotel	0	13	2015	
4	Resort Hotel	0	14	2015	
...	...	...	...	...	
119385	City Hotel	0	23	2017	
119386	City Hotel	0	102	2017	
119387	City Hotel	0	34	2017	
119388	City Hotel	0	109	2017	
119389	City Hotel	0	205	2017	

	arrival_date_month	arrival_date_week_number	\
0	July	27	
1	July	27	
2	July	27	
3	July	27	
4	July	27	
...	...	...	

119385	August	35
119386	August	35
119387	August	35
119388	August	35
119389	August	35

	arrival_date_day_of_month	stays_in_weekend_nights	\
0	1	0	
1	1	0	
2	1	0	
3	1	0	
4	1	0	
...	...	...	
119385	30	2	
119386	31	2	
119387	31	2	
119388	31	2	
119389	29	2	

	stays_in_week_nights	adults	children	babies	meal	country	\
0	0	2	0.0	0	BB	PRT	
1	0	2	0.0	0	BB	PRT	
2	1	1	0.0	0	BB	GBR	
3	1	1	0.0	0	BB	GBR	
4	2	2	0.0	0	BB	GBR	
...	...	...	...	...	...	...	
119385	5	2	0.0	0	BB	BEL	
119386	5	3	0.0	0	BB	FRA	
119387	5	2	0.0	0	BB	DEU	
119388	5	2	0.0	0	BB	GBR	
119389	7	2	0.0	0	HB	DEU	

	market_segment	distribution_channel	is_repeated_guest	\
0	Direct	Direct	0	
1	Direct	Direct	0	
2	Direct	Direct	0	
3	Corporate	Corporate	0	
4	Online TA	TA/TO	0	
...	...	...	...	
119385	Offline TA/TO	TA/TO	0	
119386	Online TA	TA/TO	0	
119387	Online TA	TA/TO	0	
119388	Online TA	TA/TO	0	
119389	Online TA	TA/TO	0	

	previous_cancellations	previous_bookings_not_canceled	\
0	0	0	

1	0	0
2	0	0
3	0	0
4	0	0
...	...	...
119385	0	0
119386	0	0
119387	0	0
119388	0	0
119389	0	0

	reserved_room_type	assigned_room_type	booking_changes	deposit_type	\
0	C	C	3	No Deposit	
1	C	C	4	No Deposit	
2	A	C	0	No Deposit	
3	A	A	0	No Deposit	
4	A	A	0	No Deposit	
...	...	...	...	...	
119385	A	A	0	No Deposit	
119386	E	E	0	No Deposit	
119387	D	D	0	No Deposit	
119388	A	A	0	No Deposit	
119389	A	A	0	No Deposit	

	agent	days_in_waiting_list	customer_type	adr	\
0	0.0	0	Transient	0.00	
1	0.0	0	Transient	0.00	
2	0.0	0	Transient	75.00	
3	304.0	0	Transient	75.00	
4	240.0	0	Transient	98.00	
...	...	...	...	...	
119385	394.0	0	Transient	96.14	
119386	9.0	0	Transient	225.43	
119387	9.0	0	Transient	157.71	
119388	89.0	0	Transient	104.40	
119389	9.0	0	Transient	151.20	

	required_car_parking_spaces	total_of_special_requests	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	1	
...	...	...	
119385	0	0	
119386	0	2	
119387	0	4	

119388	0	0
119389	0	2

	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
...	...	...
119385	Check-Out	2017-09-06
119386	Check-Out	2017-09-07
119387	Check-Out	2017-09-07
119388	Check-Out	2017-09-07
119389	Check-Out	2017-09-07

[119210 rows x 31 columns]

After dealing with the null values and dropping few unwanted rows the new shape of our dataset is (119210,31)

```
[22]: ## Converting Datatype: Children are listed as float datatype but in reality
      ↪ its interger, so needs to be changed

hotel['children'] = hotel['children'].astype('int64')
hotel['agent'] = hotel['agent'].astype('int64')
hotel['country'] = hotel['country'].astype('str')
hotel['reservation_status_date'] = hotel['reservation_status_date'].
      ↪astype('datetime64')
#looking at the reservation_status_date we can see it doesnt have correct
      ↪Dtype, hence we need to change it to datetime 64
hotel.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 119210 entries, 0 to 119389
Data columns (total 31 columns):
```

#	Column	Non-Null Count	Dtype
0	hotel	119210 non-null	object
1	is_canceled	119210 non-null	int64
2	lead_time	119210 non-null	int64
3	arrival_date_year	119210 non-null	int64
4	arrival_date_month	119210 non-null	object
5	arrival_date_week_number	119210 non-null	int64
6	arrival_date_day_of_month	119210 non-null	int64
7	stays_in_weekend_nights	119210 non-null	int64
8	stays_in_week_nights	119210 non-null	int64

```

9  adults          119210 non-null int64
10 children        119210 non-null int64
11 babies          119210 non-null int64
12 meal            119210 non-null object
13 country          119210 non-null object
14 market_segment  119210 non-null object
15 distribution_channel 119210 non-null object
16 is_repeated_guest 119210 non-null int64
17 previous_cancellations 119210 non-null int64
18 previous_bookings_not_canceled 119210 non-null int64
19 reserved_room_type 119210 non-null object
20 assigned_room_type 119210 non-null object
21 booking_changes  119210 non-null int64
22 deposit_type      119210 non-null object
23 agent             119210 non-null int64
24 days_in_waiting_list 119210 non-null int64
25 customer_type      119210 non-null object
26 adr               119210 non-null float64
27 required_car_parking_spaces 119210 non-null int64
28 total_of_special_requests 119210 non-null int64
29 reservation_status 119210 non-null object
30 reservation_status_date 119210 non-null datetime64[ns]
dtypes: datetime64[ns](1), float64(1), int64(18), object(11)
memory usage: 29.1+ MB

```

## Data Visualization

```

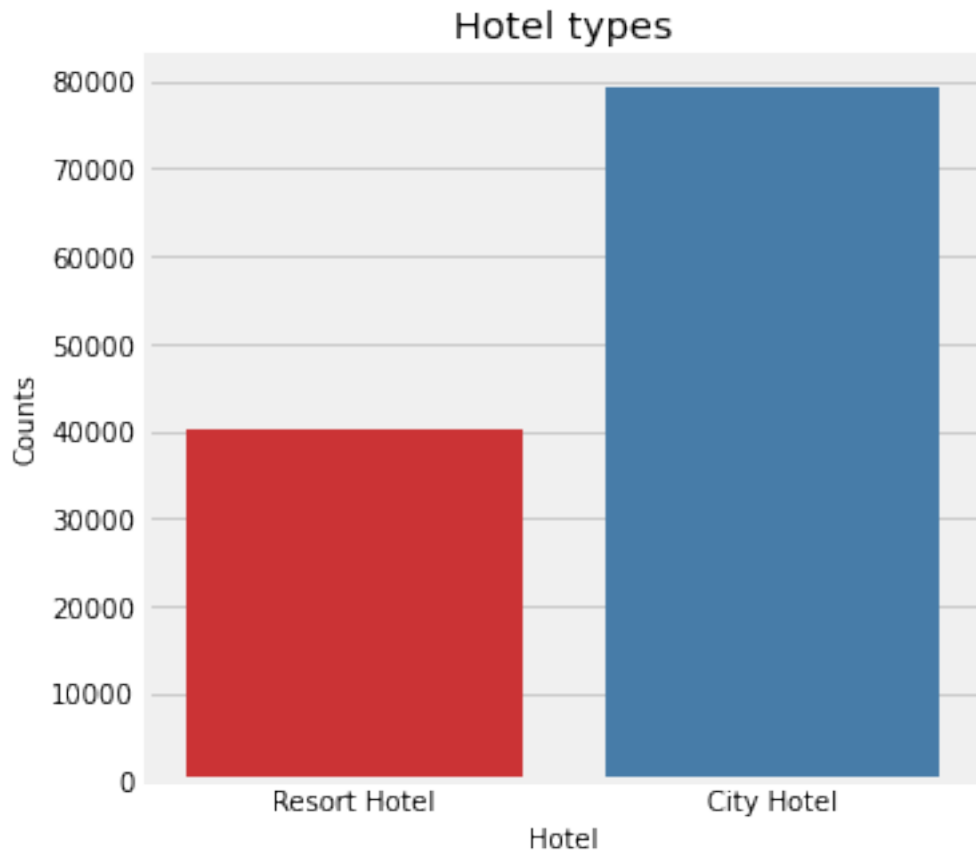
[23]: plt.figure(figsize=(5,5))
      sns.countplot(x = 'hotel', data = hotel, palette = 'Set1')
      plt.title('Hotel types')
      plt.xlabel('Hotel', fontsize = 10)
      plt.ylabel('Counts', fontsize = 10)

```

```

[23]: Text(0, 0.5, 'Counts')

```



```
[24]: hotel['hotel'].value_counts()/hotel.shape[0]*100
```

```
[24]: City Hotel      66.406342  
      Resort Hotel   33.593658  
      Name: hotel, dtype: float64
```

66 % reservations were made for city hotel and the remaining 34% for the Resort Hotel, which means higher number of reservations were made for the City Hotel

```
[25]: hotel['is_canceled'].value_counts()/hotel.shape[0]*100
```

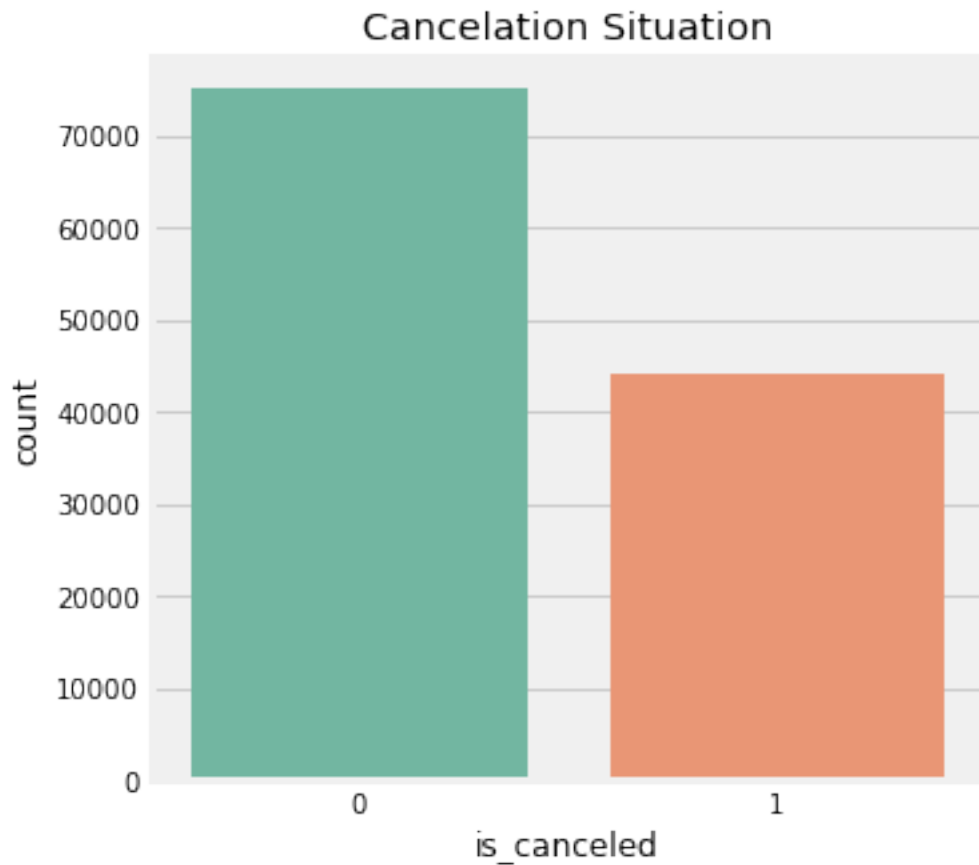
```
[25]: 0    62.923412  
      1    37.076588  
      Name: is_canceled, dtype: float64
```

63% of the total reservations were not canceled and 37% were canceled combined from both the hotels

```
[26]: #Checking the cancelation status  
      plt.figure(figsize=(5,5))
```



```
sns.countplot(x='is_canceled' , data = hotel, palette = 'Set2')
plt.title('Cancellation Situation')
plt.show()
```



Higher number of “cancellations” and “not cancellations” were made for the City Hotel

```
[27]: #calculation of ratio of uncanceled and canceled bookings at City and Resort,
      ↪Hotels
a = hotel [hotel['is_canceled']==0].groupby('hotel').is_canceled.count()
b = hotel [hotel['is_canceled']==1].groupby('hotel').is_canceled.count()

data = pd.DataFrame({'hotel':a.index,
                     '0':a.values,
                     '1':b.values
                     })
data["Ratio of uncanceled bookings"] = data['0']/( data['0']+ data['1'])
data["Ratio of canceled bookings"] = data['1']/( data['0']+ data['1'])
data
```

```
[27]:
```

	hotel	0	1	Ratio of uncanceled bookings \
0	City Hotel	46084	33079	0.582141
1	Resort Hotel	28927	11120	0.722326

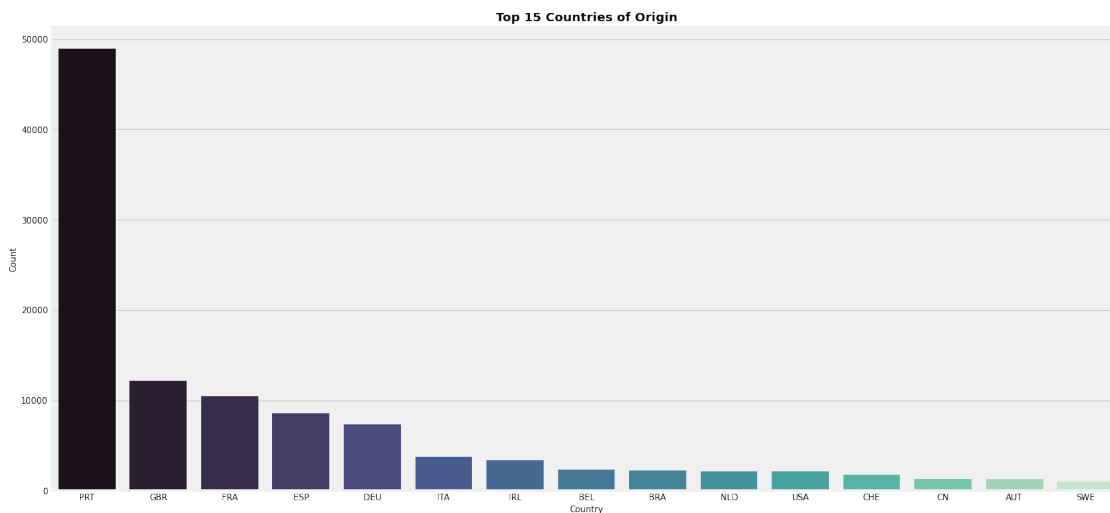
  

	Ratio of canceled bookings
0	0.417859
1	0.277674

Looking at the ratio of cancellations it can be noted that higher cancellations were observed in City Hotel as compared to Resort Hotel.

```
[28]: # Plotting these countries on a graph
plt.figure(figsize=(20,10))
sns.countplot(x='country', data=hotel,
              order=pd.value_counts(hotel['country']).iloc[:15].
               ↪index,palette="mako")
plt.title('Top 15 Countries of Origin', weight='bold')
plt.xlabel('Country', fontsize=10)
plt.ylabel('Count', fontsize=10)
```

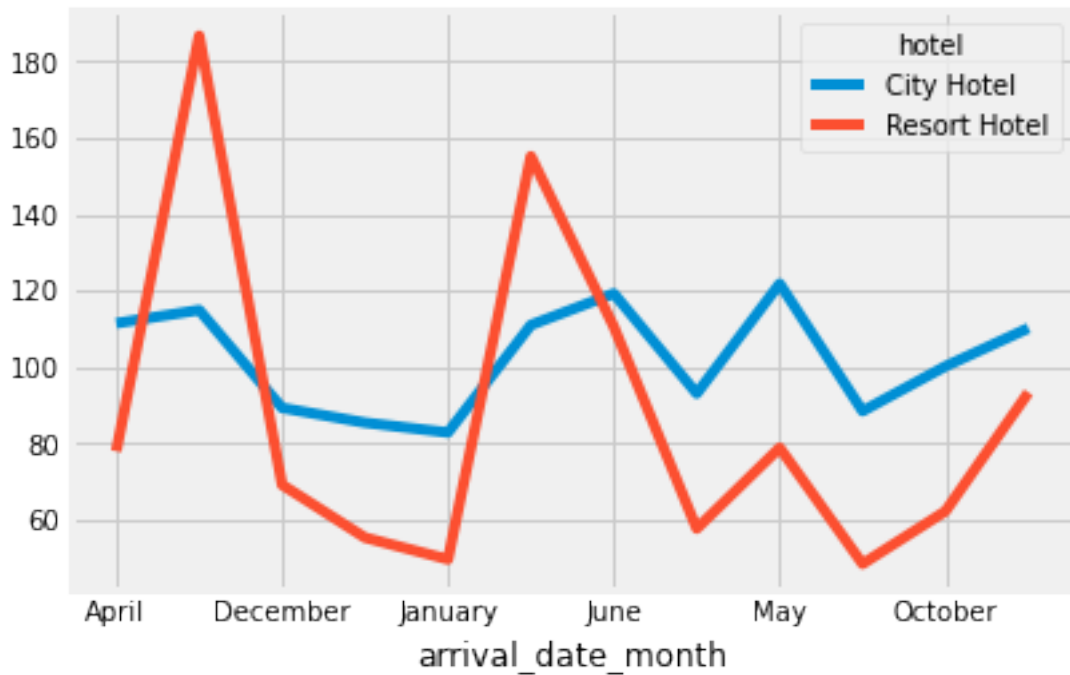
```
[28]: Text(0, 0.5, 'Count')
```



Tourists are traveling from across the globe to stay at these hotels. Home country for most of the guests is Portugal along with other countries in Europe.

```
[29]: #Comparison of average daily charges of two hotels by month
hotel.
      ↪pivot_table(values='adr',index='arrival_date_month',columns='hotel',aggfunc='mean').
      ↪plot()
```

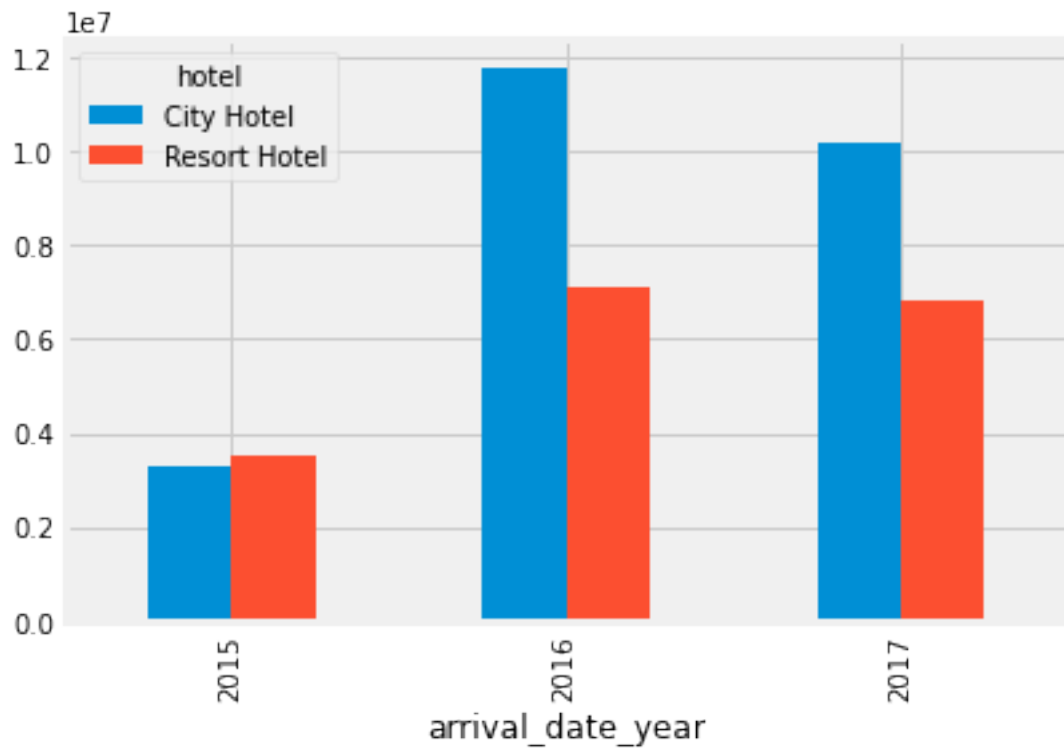
[29]: <matplotlib.axes.\_subplots.AxesSubplot at 0x7fdada30b090>



Prices at both the hotels are quite variable. It can be noted that ADR is higher in July and August months since 2015.

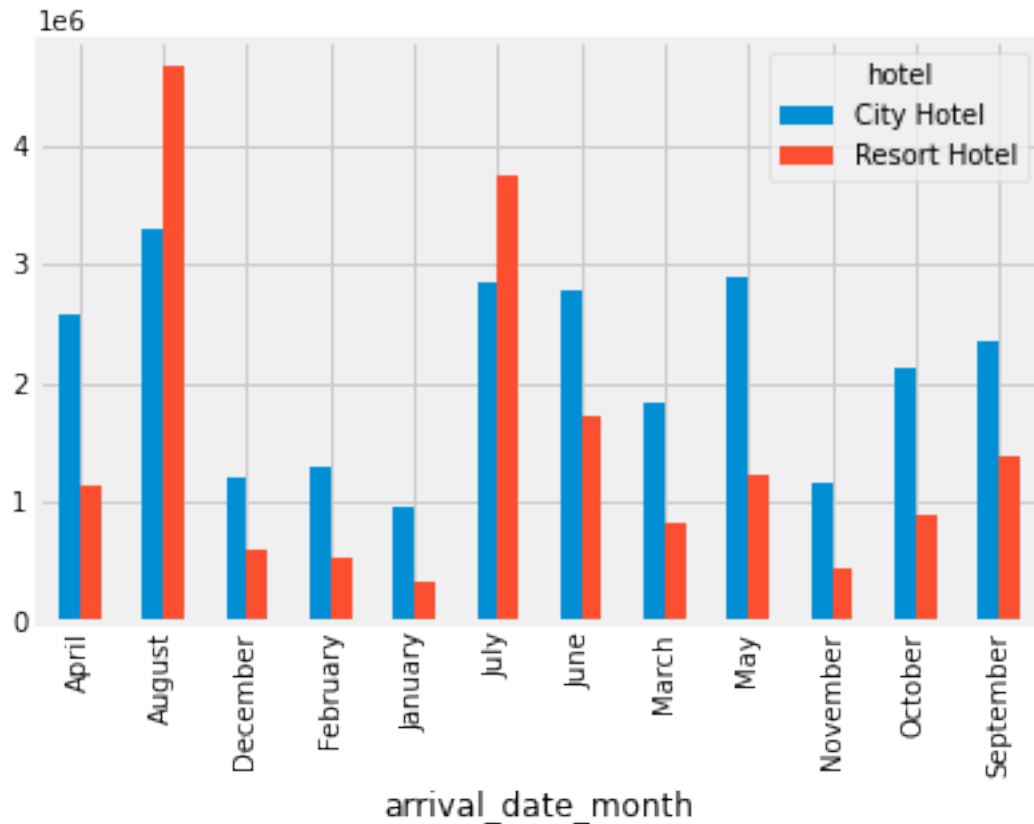
### Comparing the turnover of two hotels from 2015-2017

```
[30]: hotel['total_adr']=(hotel['stays_in_weekend_nights']+hotel['stays_in_week_nights'])*hotel['adr']
      hotel.
      ↪pivot_table(values='total_adr',index='arrival_date_year',columns='hotel',aggfunc='sum').
      ↪plot.bar()
      plt.show()
```



The turnover in the 2016 and 2017 was higher for City hotel, but lower in 2015 as compared Resort Hotel

```
[31]: hotel.  
      ↪pivot_table(values='total_adr',index='arrival_date_month',columns='hotel',aggfunc='sum').  
      ↪plot.bar()  
      plt.show()  
      plt.figure(figsize=(10,10))
```



[31]: <Figure size 720x720 with 0 Axes>

<Figure size 720x720 with 0 Axes>

Resort Hotel has higher turnover than City Hotel in the months July and August and in the rest of the months City Hotel makes higher revenues.

[ ]:

**Rearranging the data by 'Month'**

[32]: `pip install sort-dataframeby-monthorweek`

```
Processing ./cache/pip/wheels/de/e1/ad/5fe265a9780676079c4b8caaaffaa8d5c4ab2f37
cf823e8aa8/sort_dataframeby_monthorweek-0.4-py3-none-any.whl
Installing collected packages: sort-dataframeby-monthorweek
Successfully installed sort-dataframeby-monthorweek-0.4
Note: you may need to restart the kernel to use updated packages.
```

[33]: `pip install sorted-months-weekdays`

Processing ./cache/pip/wheels/4f/4f/78/3f1b8fc72651f7c766a6f73d667fccb12a8aabe240b38df7a4/sorted\_months\_weekdays-0.2-py3-none-any.whl  
Installing collected packages: sorted-months-weekdays  
Successfully installed sorted-months-weekdays-0.2  
Note: you may need to restart the kernel to use updated packages.

```
[34]: from sorted_months_weekdays import *

from sort_dataframeby_monthorweek import *

final = Sort_Dataframeby_Month(df=df, monthcolumnname='arrival_date_month')
final.head()
```

```
[34]:      hotel  is_canceled  lead_time  arrival_date_year  arrival_date_month \
0  Resort Hotel          0        109             2016           January
1  Resort Hotel          0        109             2016           January
2  Resort Hotel          1         2             2016           January
3  Resort Hotel          0         88             2016           January
4  Resort Hotel          1         20             2016           January

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

      stays_in_weekend_nights  stays_in_week_nights  adults  children  babies \
0                             0                      1       2         0.0      0
1                             0                      1       2         2.0      0
2                             0                      1       2         0.0      0
3                             0                      2       2         0.0      0
4                             0                      2       2         2.0      0

      meal  country  market_segment  distribution_channel  is_repeated_guest \
0  BB      RUS      Online TA      TA/TO                      0
1  BB      RUS      Online TA      TA/TO                      0
2  BB      PRT      Online TA      TA/TO                      0
3  HB      ARG      Online TA      TA/TO                      0
4  BB      PRT      Online TA      TA/TO                      0

      previous_cancellations  previous_bookings_not_canceled  reserved_room_type \
0                             0                              0                  A
1                             0                              0                  H
2                             0                              0                  D
3                             0                              0                  A
4                             0                              0                  G
```

	assigned_room_type	booking_changes	deposit_type	agent	company	\
0	D	0	No Deposit	240.0	NaN	
1	H	0	No Deposit	240.0	NaN	
2	D	0	No Deposit	240.0	NaN	
3	D	0	No Deposit	241.0	NaN	
4	G	0	No Deposit	240.0	NaN	

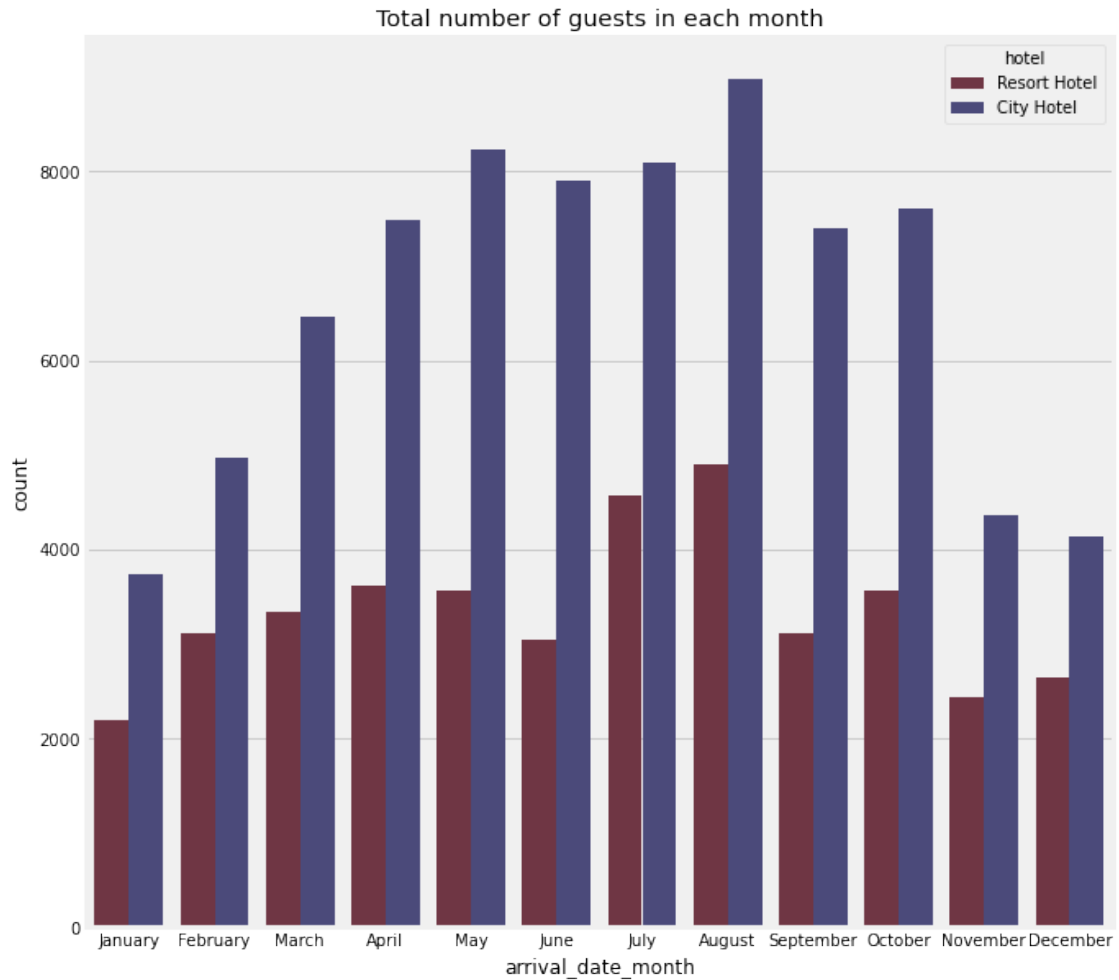
  

	days_in_waiting_list	customer_type	adr	required_car_parking_spaces	\
0	0	Transient-Party	59.94		0
1	0	Transient-Party	116.10		1
2	0	Transient	89.00		0
3	0	Transient	73.46		0
4	0	Transient	119.00		0

	total_of_special_requests	reservation_status	reservation_status_date
0	1	Check-Out	2016-01-02
1	1	Check-Out	2016-01-02
2	1	No-Show	2016-01-01
3	2	Check-Out	2016-01-03
4	0	Canceled	2015-12-22

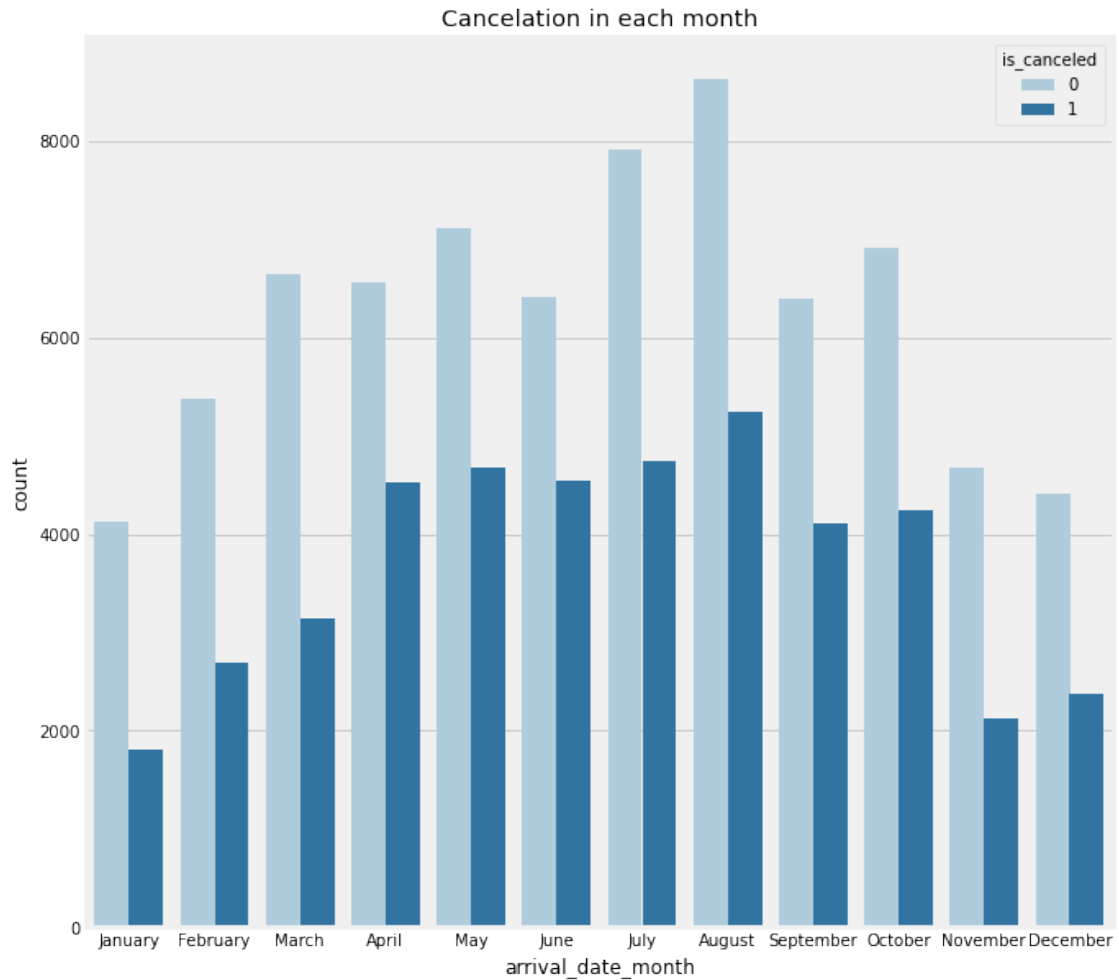
```
[35]: plt.figure(figsize=(10,10))
sns.countplot(x='arrival_date_month', hue= 'hotel', data= final, palette = "icefire_r")
plt.title('Total number of guests in each month')
plt.show()
```



Observing the bar chart we can see that Resort Hotel gets busy in July and September, whereas demand in the city hotel stays from May to October.

```
[36]: plt.figure(figsize=(10,10))
sns.countplot(x='arrival_date_month', hue= 'is_canceled', data= final, palette=
↪ "Paired")
plt.title('Cancellation in each month')
plt.show()
```





```
[37]: country_wise_guests = hotel[hotel['is_canceled'] == 0]['country'].
      ↪value_counts().reset_index()
      country_wise_guests.columns = ['Country', 'Total no of guests']
      country_wise_guests
```

```
[37]:
```

	Country	Total no of guests
0	PRT	21398
1	GBR	9668
2	FRA	8468
3	ESP	6383
4	DEU	6067
..	...	...
160	MRT	1
161	KIR	1
162	DMA	1
163	BHS	1

[165 rows x 2 columns]

```
[38]: country_wise_guests = df[df['is_canceled'] == 0]['country'].value_counts().
      ↪reset_index()
country_wise_guests.columns = ['country', 'No of guests']
country_wise_guests
basemap = folium.Map()
guests_map = px.choropleth(country_wise_guests, locations =_
      ↪country_wise_guests['country'],
                        color = country_wise_guests['No of guests'],_
      ↪hover_name = country_wise_guests['country'])
guests_map.show()
```

### Price variation per night at both the hotels

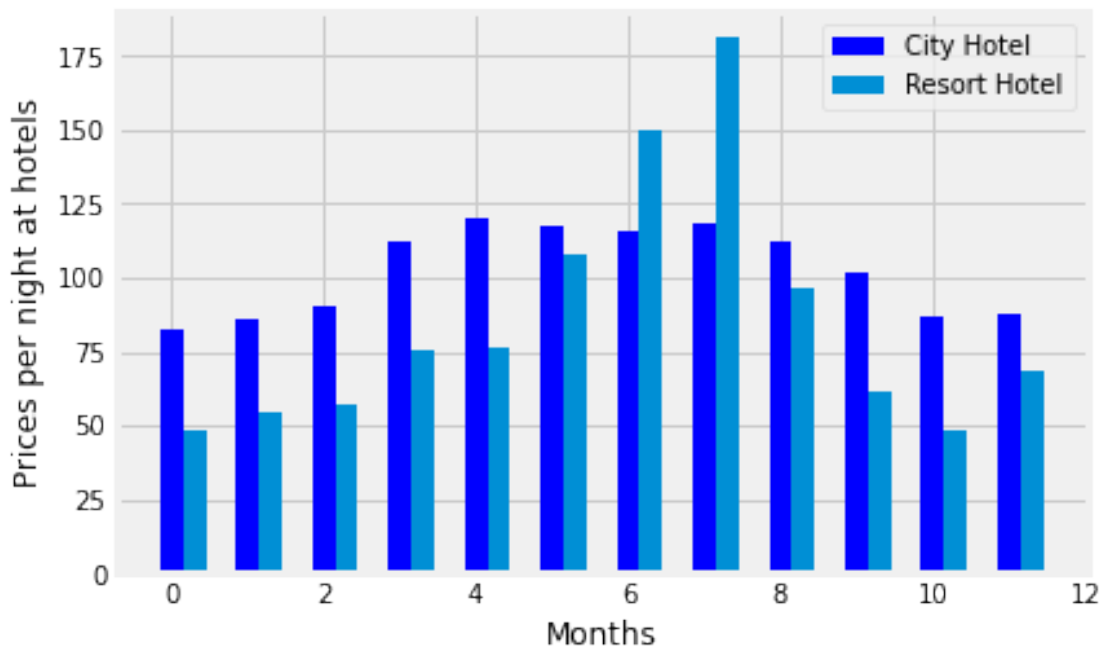
```
[39]: data_city =final[(final['hotel'] == 'City Hotel') & (final['is_canceled'] == 0)]
data_resort =final[(final['hotel'] == 'Resort Hotel') & (final['is_canceled']_
      ↪== 0)]

city_hotel = data_city.groupby(['arrival_date_month'])['adr'].mean().
      ↪reset_index()
resort_hotel = data_resort.groupby(['arrival_date_month'])['adr'].mean().
      ↪reset_index()

final_hotel = city_hotel.merge(resort_hotel, on = 'arrival_date_month')
final_hotel.columns = ['arrival_month', 'price_for_city_hotel',_
      ↪'price_for_resort_hotel']
price_per_night =_
      ↪Sort_Dataframeby_Month(df=final_hotel,monthcolumnname='arrival_month')
price_per_night
```

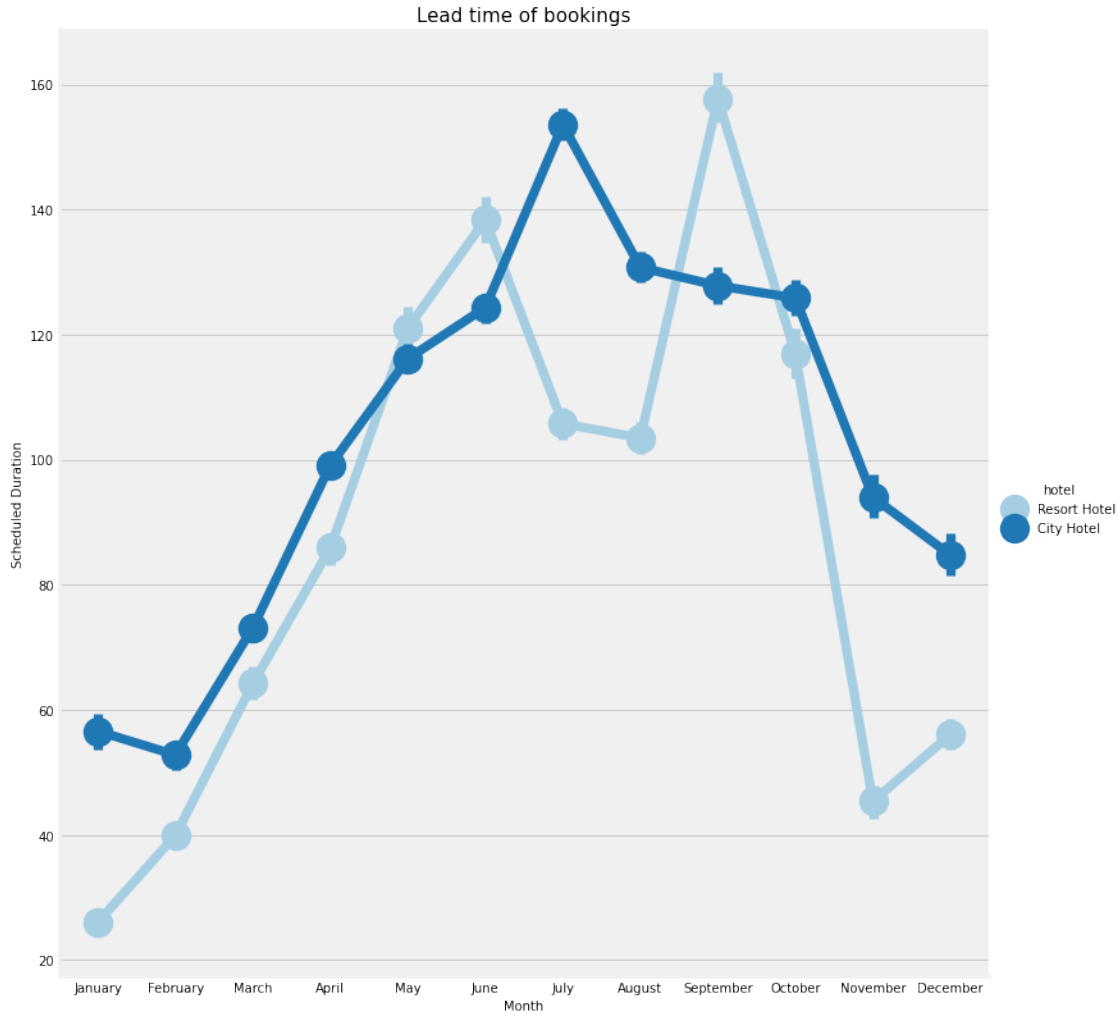
```
[39]:   arrival_month  price_for_city_hotel  price_for_resort_hotel
0      January          82.160634          48.708919
1    February          86.183025          54.147478
2      March          90.170722          57.012487
3      April         111.856824          75.867816
4        May         120.445842          76.657558
5        June         117.702075         107.921869
6        July         115.563810         150.122528
7      August         118.412083         181.205892
8    September         112.598452          96.416860
9      October         101.745956          61.727505
10    November          86.500456          48.681640
11    December          87.856764          68.322236
```

```
[40]: w = 0.3
x = price_per_night.arrival_month
bar1 = np.arange(len(x))
bar2 = [i+w for i in bar1]
plt.bar(bar1,price_per_night.price_for_city_hotel,w,color="blue",label="City
Hotel")
plt.bar(bar2,price_per_night.price_for_resort_hotel,w,label="Resort Hotel")
plt.xlabel("Months")
plt.ylabel("Prices per night at hotels")
plt.legend()
plt.show()
```



Throughout the year price per night was higher at City Hotel in comparison to Resort Hotel, except for the months 'July and August.

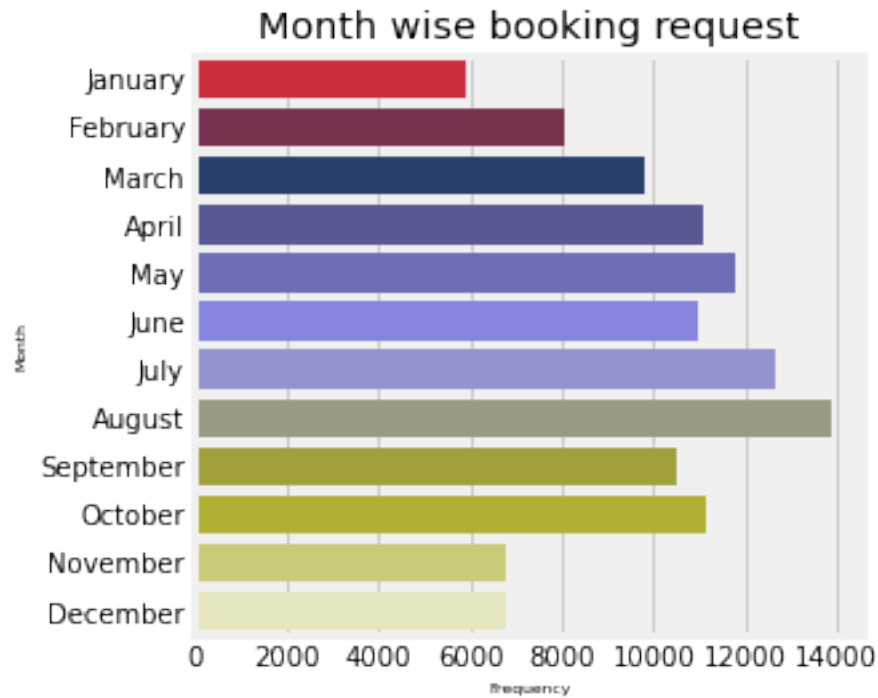
```
[41]: sns.factorplot(x='arrival_date_month',y='lead_time', hue='hotel', palette =
'Paired', data =final,size=11)
plt.title("Lead time of bookings", fontsize=15)
plt.xlabel("Month", fontsize=10)
plt.ylabel("Scheduled Duration", fontsize=10)
plt.show()
```



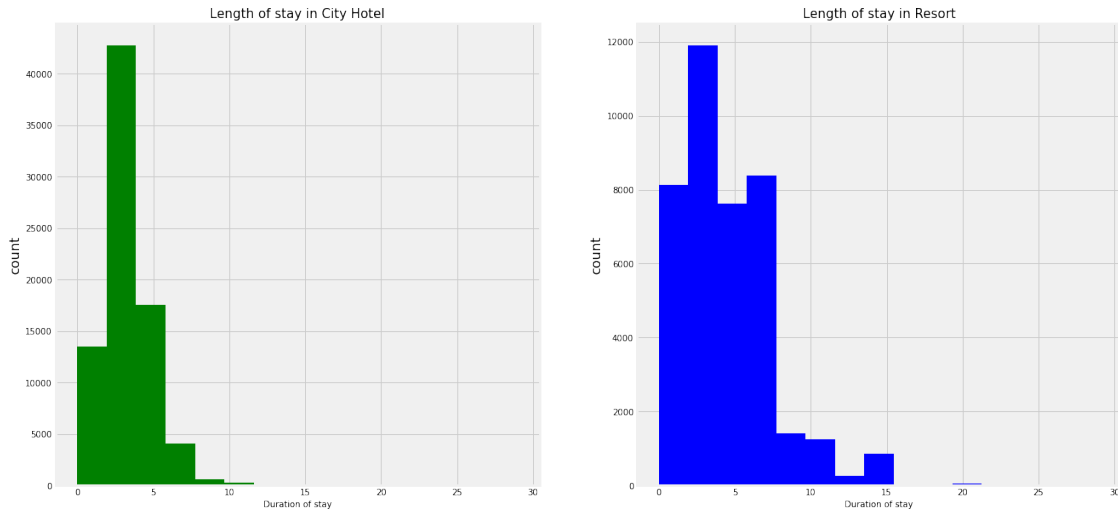
Guests tend to make reservations in advance for June to October in City and Hotel and June and September for Resort Hotel

```
[42]: #Cumulative Monthwise bookings for 3 years
plt.figure(figsize=(4,4))
sns.countplot(y='arrival_date_month', data= final, palette='gist_stern', orient='v')
plt.title('Month wise booking request')
plt.xlabel('Frequency', fontsize=6)
plt.ylabel('Month', fontsize=6)
```

```
[42]: Text(0, 0.5, 'Month')
```



```
[43]: hotel['total_stay']=hotel['stays_in_weekend_nights']+hotel['stays_in_week_nights']
plt.figure(figsize=(20,10))
plt.subplot(1,2,1)
hotel.query("total_stay<30&hotel=='City Hotel'").total_stay.plot.
    ↳hist(bins=15,color='g')
plt.title("Length of stay in City Hotel", fontsize=15)
plt.xlabel("Duration of stay", fontsize=10)
plt.ylabel("count", fontsize=16)
plt.subplot(1,2,2)
hotel.query("total_stay<30&hotel=='Resort Hotel'").total_stay.plot.
    ↳hist(bins=15,color='b')
plt.title("Length of stay in Resort", fontsize=15)
plt.xlabel("Duration of stay", fontsize=10)
plt.ylabel("count", fontsize=16)
plt.show()
```



Guests prefer to stay longer in Resort Hotel as compared to City Hotel. On an average, guests stay at City Hotel for 2.92 nights and 4.14 nights at Resort Hotel. For resort hotel, often 1-4 nights are booked for both City and Resort Hotels.

```
[44]: #Reserved Room Type vs Assigned Room Type
df2 = pd.crosstab(index = hotel['reserved_room_type'], columns = hotel['assigned_room_type'], normalize='index').round(2)*100
df2
```

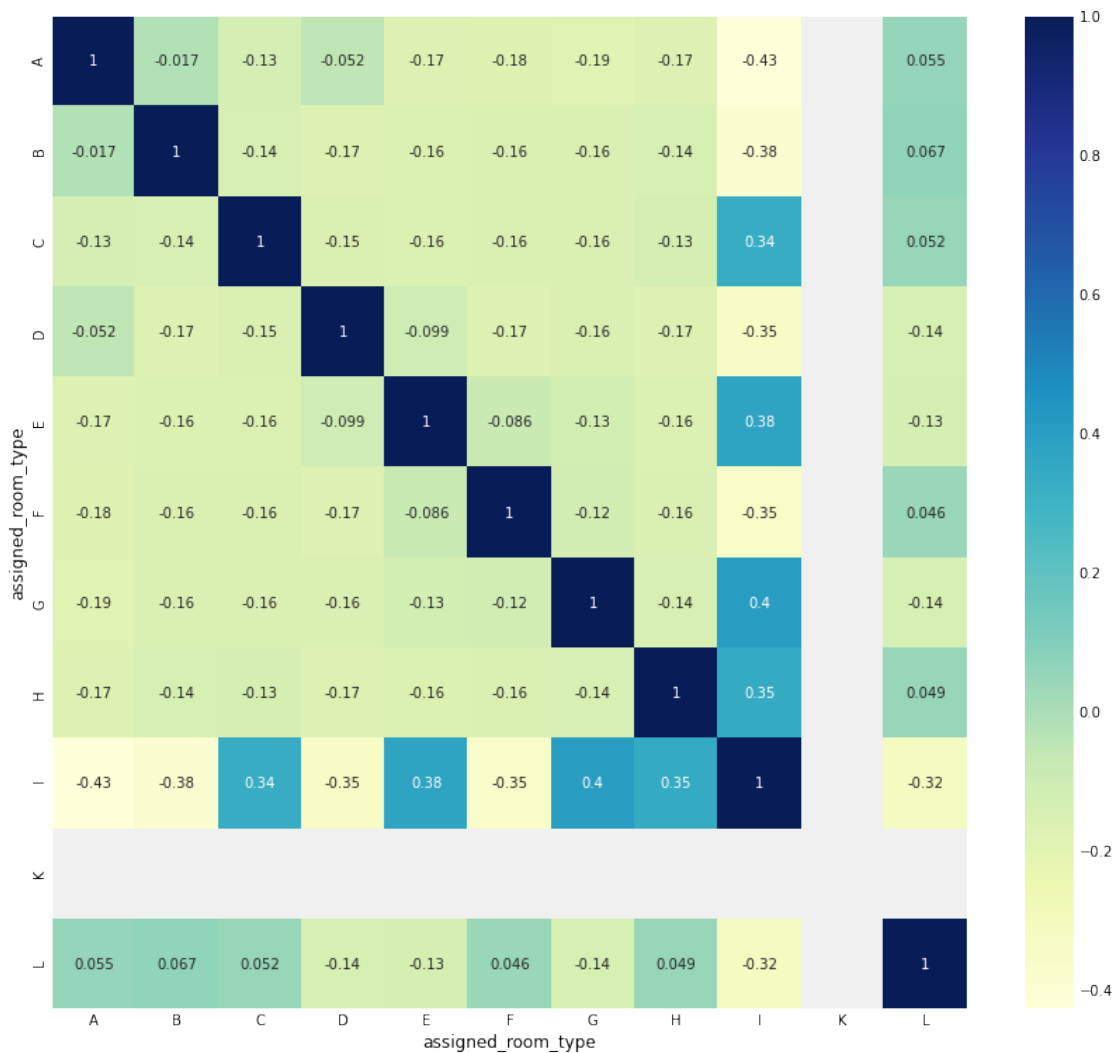
```
[44]: assigned_room_type    A      B      C      D      E      F      G      H      I      K  \
reserved_room_type
A      86.0    1.0    2.0    9.0    1.0    0.0    0.0    0.0    0.0    0.0
B      10.0   88.0    0.0    0.0    0.0    0.0    1.0    0.0    0.0    0.0
C       1.0    0.0   95.0    1.0    0.0    0.0    1.0    1.0    1.0    0.0
D       2.0    0.0    0.0   92.0    4.0    1.0    0.0    0.0    0.0    0.0
E       0.0    0.0    0.0    0.0   91.0    6.0    2.0    0.0    1.0    0.0
F       0.0    0.0    0.0    0.0    1.0   94.0    4.0    0.0    0.0    0.0
G       0.0    0.0    0.0    0.0    0.0    1.0   98.0    0.0    1.0    0.0
H       0.0    0.0    0.0    0.0    0.0    0.0    2.0   97.0    1.0    0.0
L      17.0   17.0   17.0    0.0    0.0   17.0    0.0   17.0    0.0    0.0
```

```
assigned_room_type    L
reserved_room_type
A      0.0
B      0.0
C      0.0
D      0.0
E      0.0
F      0.0
G      0.0
```

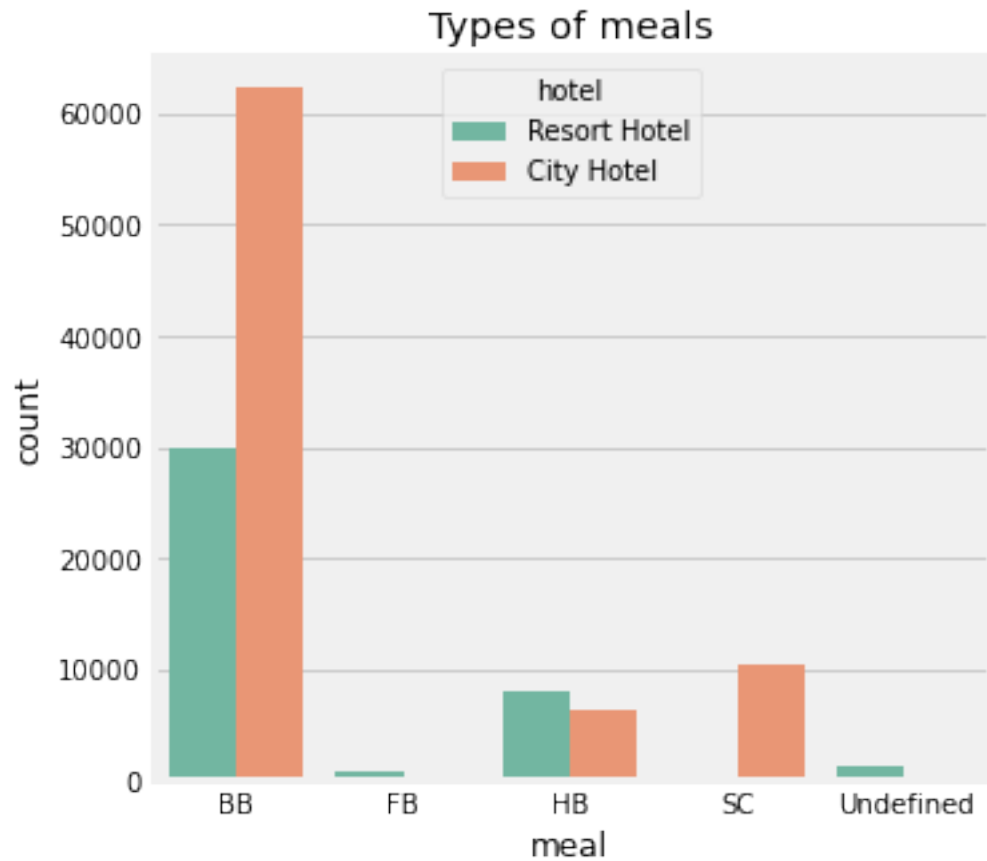
```
H          0.0
L          17.0
```

The above cross table shows the reserved type of room distribution over assigned room type. Almost at all the occasions guests received the same room type as they booked.

```
[45]: #Heatmap for the dataset of Reserved Room Type vs Assigned Room Type
plt.figure(figsize=(13,13))
dataplot = sns.heatmap(df2.corr(), cmap="YlGnBu", annot=True)
```



```
[46]: plt.figure(figsize=(5,5))
sns.countplot(x='meal', hue= 'hotel', data= hotel, palette = "Set2")
plt.title('Types of meals')
plt.show()
```

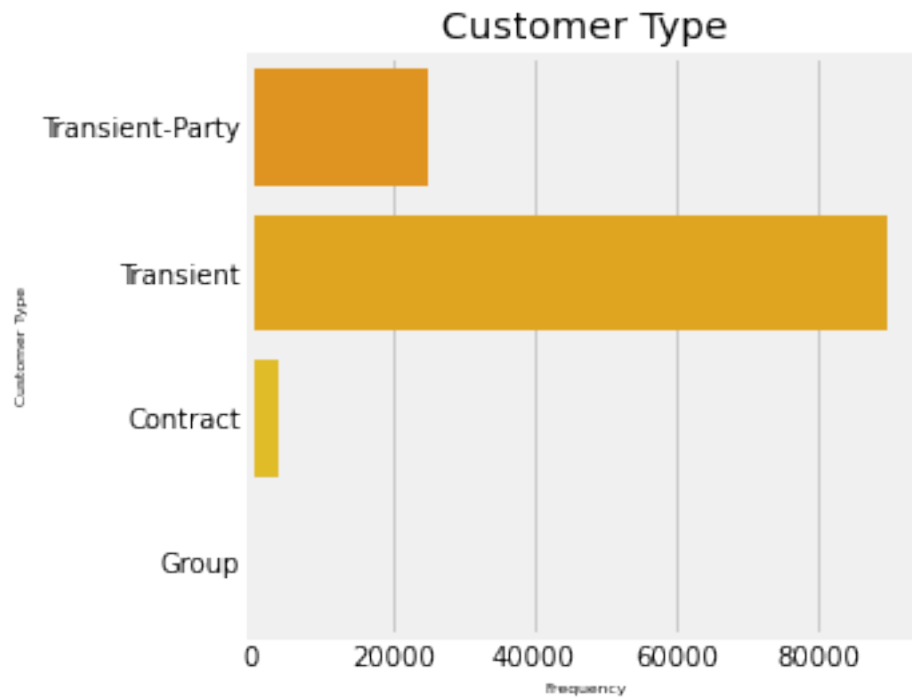


Maximum tourists requested for 'Bed and Breakfast' at both the hotels

```
[47]: #Customer type
plt.figure(figsize=(4,4))
sns.countplot(y='customer_type', data= final, palette='Wistia_r', orient = 's')
plt.title('Customer Type')
plt.xlabel('Frequency', fontsize=6)
plt.ylabel('Customer Type', fontsize=6)
```

```
[47]: Text(0, 0.5, 'Customer Type')
```

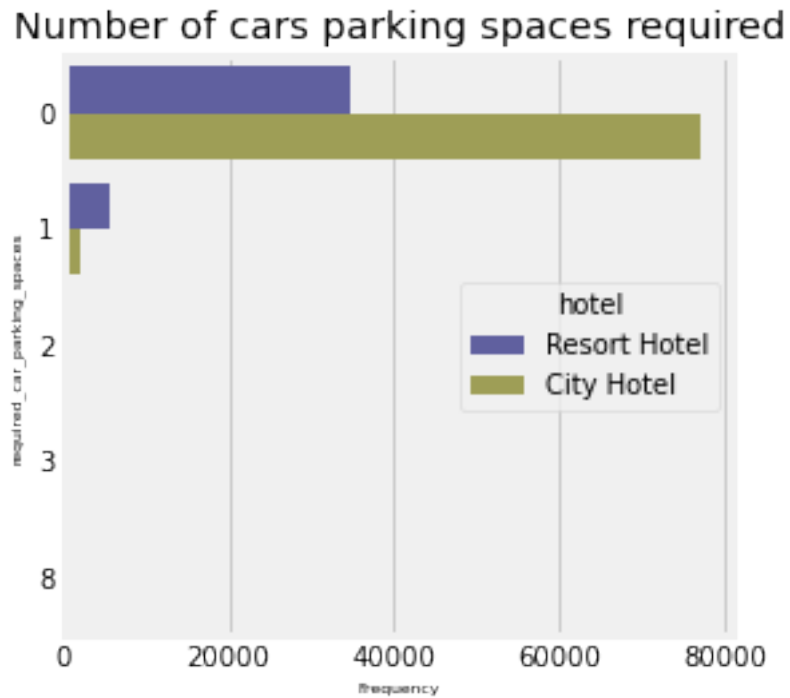




Most of the guests are transient, which indicates that they are walk-in guests or they booked last minute.

```
[48]: #Required parking spaces
plt.figure(figsize=(4,4))
sns.countplot(y='required_car_parking_spaces', hue = 'hotel', data= hotel,
             palette='gist_stern', orient = 'v')
plt.title('Number of cars parking spaces required')
plt.xlabel('Frequency', fontsize=6)
plt.ylabel('required_car_parking_spaces', fontsize=6)
```

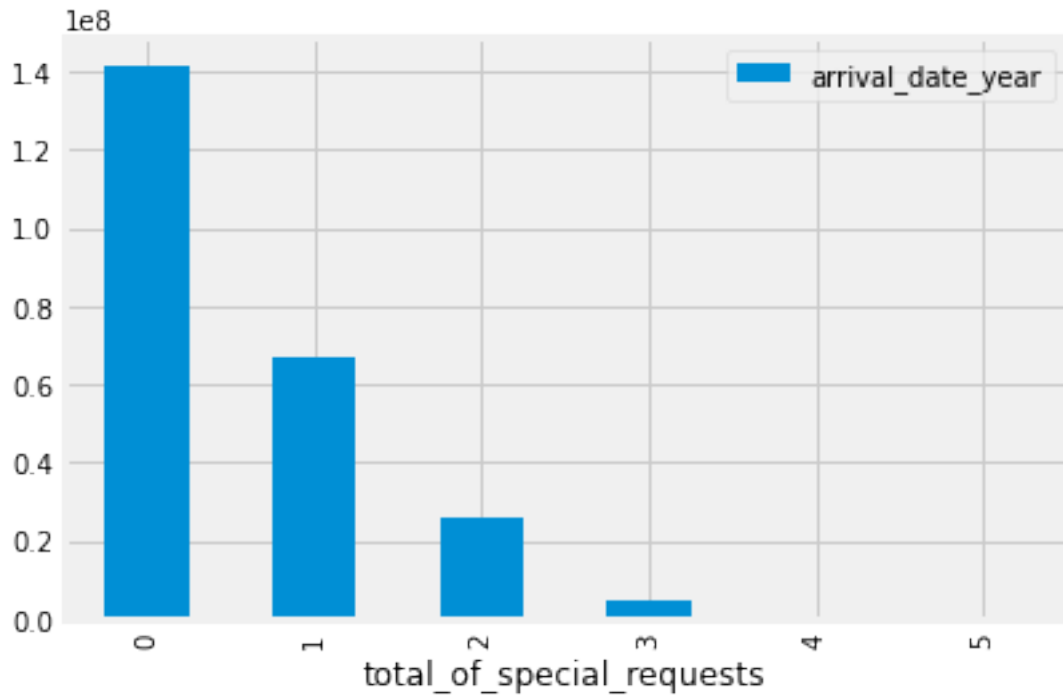
```
[48]: Text(0, 0.5, 'required_car_parking_spaces')
```



Majority of the guests travelling to these hotels donot require car parking spaces. Few guests need 1 car parking at Resort Hotel as per the graph.

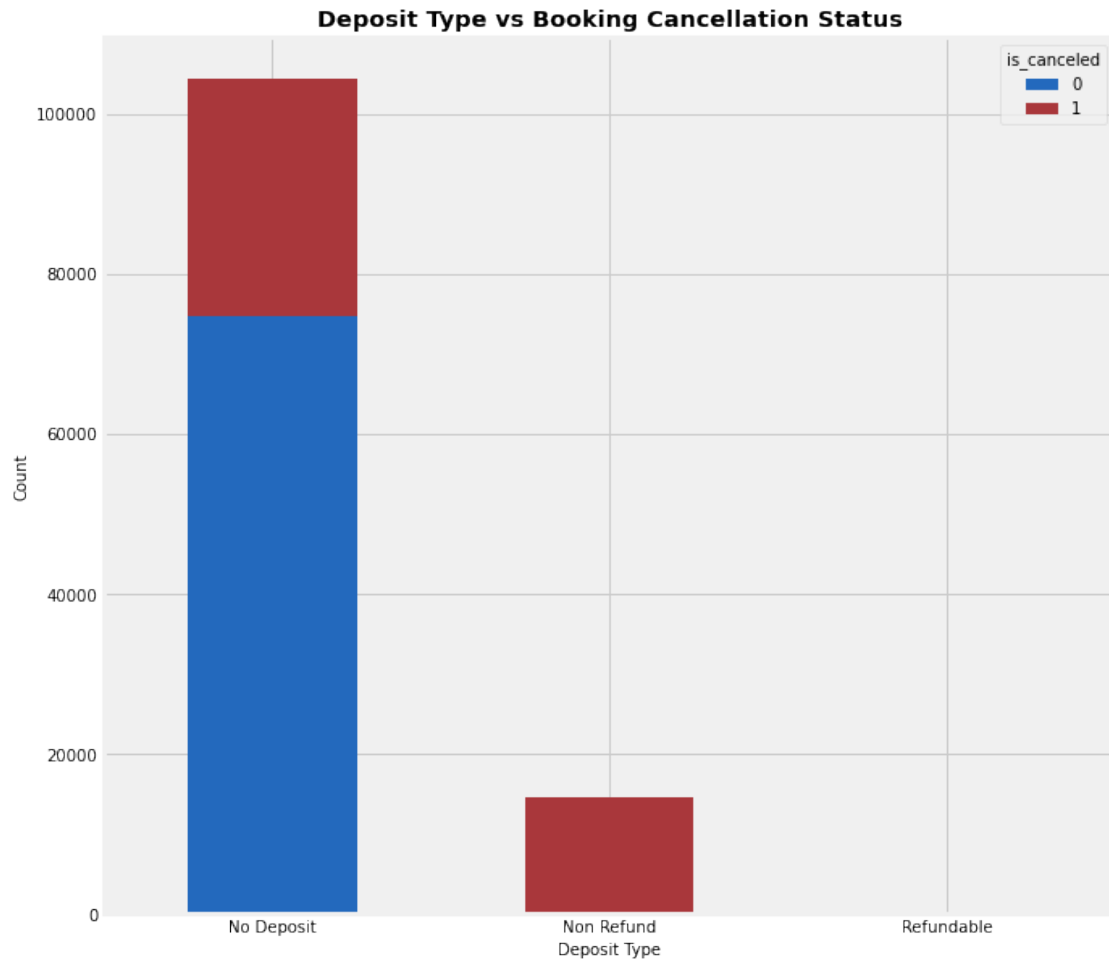
```
[49]: #Number of special requests :
hotel.
      ↪pivot_table(values='arrival_date_year',index='total_of_special_requests',aggfunc='sum').
      ↪plot.bar()
```

```
[49]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdb1dd18450>
```



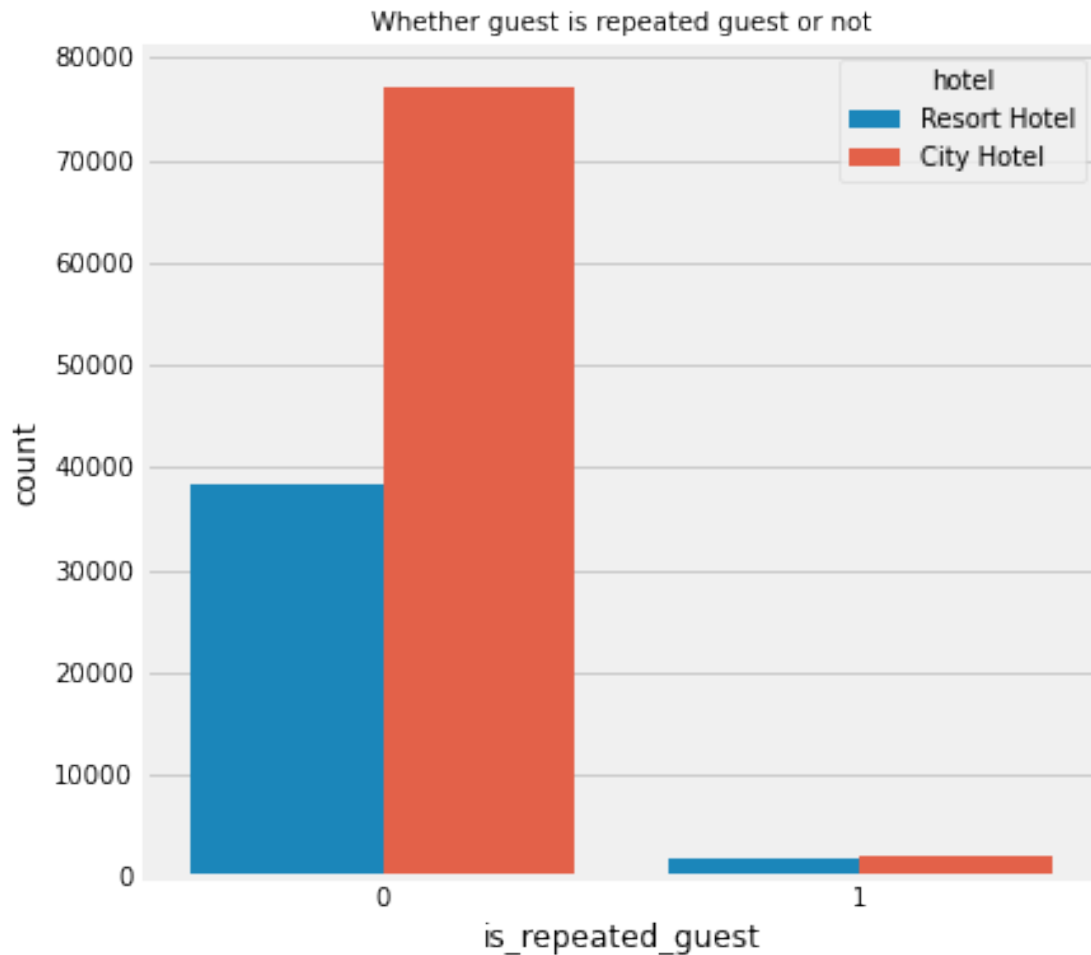
```
[50]: group_deposit = hotel.groupby(['deposit_type', 'is_canceled']).size().
      ↪unstack(fill_value=0)
group_deposit.plot(kind='bar', stacked=True, cmap='vlag', figsize=(10,10))
plt.title('Deposit Type vs Booking Cancellation Status', weight='bold')
plt.xlabel('Deposit Type', fontsize=10)
plt.xticks(rotation=360)
plt.ylabel('Count', fontsize=10)
```

```
[50]: Text(0, 0.5, 'Count')
```



For the variable 'is\_canceled' 1(red) color stands for booking was canceled, we observe that lower bookings were canceled even when No Deposit was made for the booking

```
[51]: plt.figure(figsize=(6,6))
sns.countplot(data = hotel, x = 'is_repeated_guest', hue = 'hotel').
    ↳set_title('Whether guest is repeated guest or not', fontsize = 10)
plt.show()
```



There weren't many repeated guests at both the hotels.

### Correlation Matrix

Next, categorical variables shall be converted to numerical form in order to utilize such variables in machine readable form. For this purpose, Label Encoding method will be implemented, it is an important pre-processing step for the structured dataset in supervised machine learning algorithms.

```
[52]: from sklearn import preprocessing

label_encoder = preprocessing.LabelEncoder()

# Encode labels in all the categorical columns
hotel['hotel'] = label_encoder.fit_transform(hotel['hotel'])
hotel['arrival_date_month'] = label_encoder.
    ↳fit_transform(hotel['arrival_date_month'])
hotel['meal'] = label_encoder.fit_transform(hotel['meal'])
hotel['country'] = label_encoder.fit_transform(hotel['country'])
```

```

hotel['market_segment']= label_encoder.fit_transform(hotel['market_segment'])
hotel['distribution_channel']= label_encoder.
    ↳fit_transform(hotel['distribution_channel'])
hotel['is_repeated_guest']= label_encoder.
    ↳fit_transform(hotel['is_repeated_guest'])
hotel['reserved_room_type']= label_encoder.
    ↳fit_transform(hotel['reserved_room_type'])
hotel['assigned_room_type']= label_encoder.fit_transform(hotel['deposit_type'])
hotel['deposit_type']= label_encoder.fit_transform(hotel['is_repeated_guest'])
hotel['agent']= label_encoder.fit_transform(hotel['agent'])
hotel['customer_type']= label_encoder.fit_transform(hotel['customer_type'])
hotel['reservation_status']= label_encoder.
    ↳fit_transform(hotel['reservation_status'])

```

```

[53]: hotel = hotel.
    ↳drop(['stays_in_week_nights', 'stays_in_weekend_nights', 'reservation_status_date', 'adr'], axis=1)
    ↳= 1)
hotel.info()

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 119210 entries, 0 to 119389
Data columns (total 29 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   hotel                                119210 non-null  int64
1   is_canceled                          119210 non-null  int64
2   lead_time                            119210 non-null  int64
3   arrival_date_year                    119210 non-null  int64
4   arrival_date_month                  119210 non-null  int64
5   arrival_date_week_number             119210 non-null  int64
6   arrival_date_day_of_month            119210 non-null  int64
7   adults                               119210 non-null  int64
8   children                             119210 non-null  int64
9   babies                              119210 non-null  int64
10  meal                                 119210 non-null  int64
11  country                             119210 non-null  int64
12  market_segment                       119210 non-null  int64
13  distribution_channel                  119210 non-null  int64
14  is_repeated_guest                     119210 non-null  int64
15  previous_cancellations                119210 non-null  int64
16  previous_bookings_not_canceled        119210 non-null  int64
17  reserved_room_type                    119210 non-null  int64
18  assigned_room_type                    119210 non-null  int64
19  booking_changes                       119210 non-null  int64
20  deposit_type                          119210 non-null  int64
21  agent                                119210 non-null  int64
22  days_in_waiting_list                  119210 non-null  int64

```

```

23 customer_type                119210 non-null  int64
24 required_car_parking_spaces  119210 non-null  int64
25 total_of_special_requests    119210 non-null  int64
26 reservation_status          119210 non-null  int64
27 total_adr                    119210 non-null  float64
28 total_stay                   119210 non-null  int64
dtypes: float64(1), int64(28)
memory usage: 32.3 MB

```

```

[54]: #Creating new dataframe for categorical data
hotel_categorical_data =
    hotel[['hotel', 'is_canceled', 'arrival_date_month', 'meal',
           'country', 'market_segment', 'distribution_channel',
           'is_repeated_guest', 'reserved_room_type',
           'assigned_room_type', 'deposit_type', 'agent',
           'customer_type', 'reservation_status']]
hotel_categorical_data.info()

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 119210 entries, 0 to 119389
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   hotel                  119210 non-null  int64
1   is_canceled            119210 non-null  int64
2   arrival_date_month    119210 non-null  int64
3   meal                   119210 non-null  int64
4   country                 119210 non-null  int64
5   market_segment         119210 non-null  int64
6   distribution_channel   119210 non-null  int64
7   is_repeated_guest      119210 non-null  int64
8   reserved_room_type     119210 non-null  int64
9   assigned_room_type     119210 non-null  int64
10  deposit_type           119210 non-null  int64
11  agent                  119210 non-null  int64
12  customer_type           119210 non-null  int64
13  reservation_status     119210 non-null  int64
dtypes: int64(14)
memory usage: 18.6 MB

```

```

[55]: #Creating new dataframe for numerical data
hotel_numerical_data= hotel.drop(['hotel', 'is_canceled',
    'arrival_date_month', 'meal',
    'country', 'market_segment', 'distribution_channel',

```

```

                                'is_repeated_guest'],
    ↪ 'reserved_room_type',
                                ↵
    ↪ 'assigned_room_type', 'deposit_type', 'agent',
                                'customer_type', 'reservation_status'],
    ↪ axis = 1)
hotel_numerical_data.info()

```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 119210 entries, 0 to 119389
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   lead_time                            119210 non-null  int64
1   arrival_date_year                    119210 non-null  int64
2   arrival_date_week_number             119210 non-null  int64
3   arrival_date_day_of_month            119210 non-null  int64
4   adults                               119210 non-null  int64
5   children                             119210 non-null  int64
6   babies                              119210 non-null  int64
7   previous_cancellations               119210 non-null  int64
8   previous_bookings_not_canceled       119210 non-null  int64
9   booking_changes                     119210 non-null  int64
10  days_in_waiting_list                 119210 non-null  int64
11  required_car_parking_spaces          119210 non-null  int64
12  total_of_special_requests            119210 non-null  int64
13  total_adr                           119210 non-null  float64
14  total_stay                          119210 non-null  int64
dtypes: float64(1), int64(14)
memory usage: 19.6 MB

```

```

[56]: # Correlation Matrix of Categorical Data with Spearman method
plt.figure(figsize=(13,13))
corr_categorical=hotel_categorical_data.corr(method='spearman')
mask_categorical = np.triu(np.ones_like(corr_categorical, dtype=np.bool))
sns.heatmap(corr_categorical, annot=True, fmt=".2f", cmap='RdBu', vmin=-1,
    ↪ vmax=1, center= 0,
            square=True, linewidths=2, cbar_kws={"shrink": .5}).set(ylim=(15,
    ↪ 0))
plt.title("Correlation Matrix of Categorical Data using Spearman
    ↪ Method",size=14)

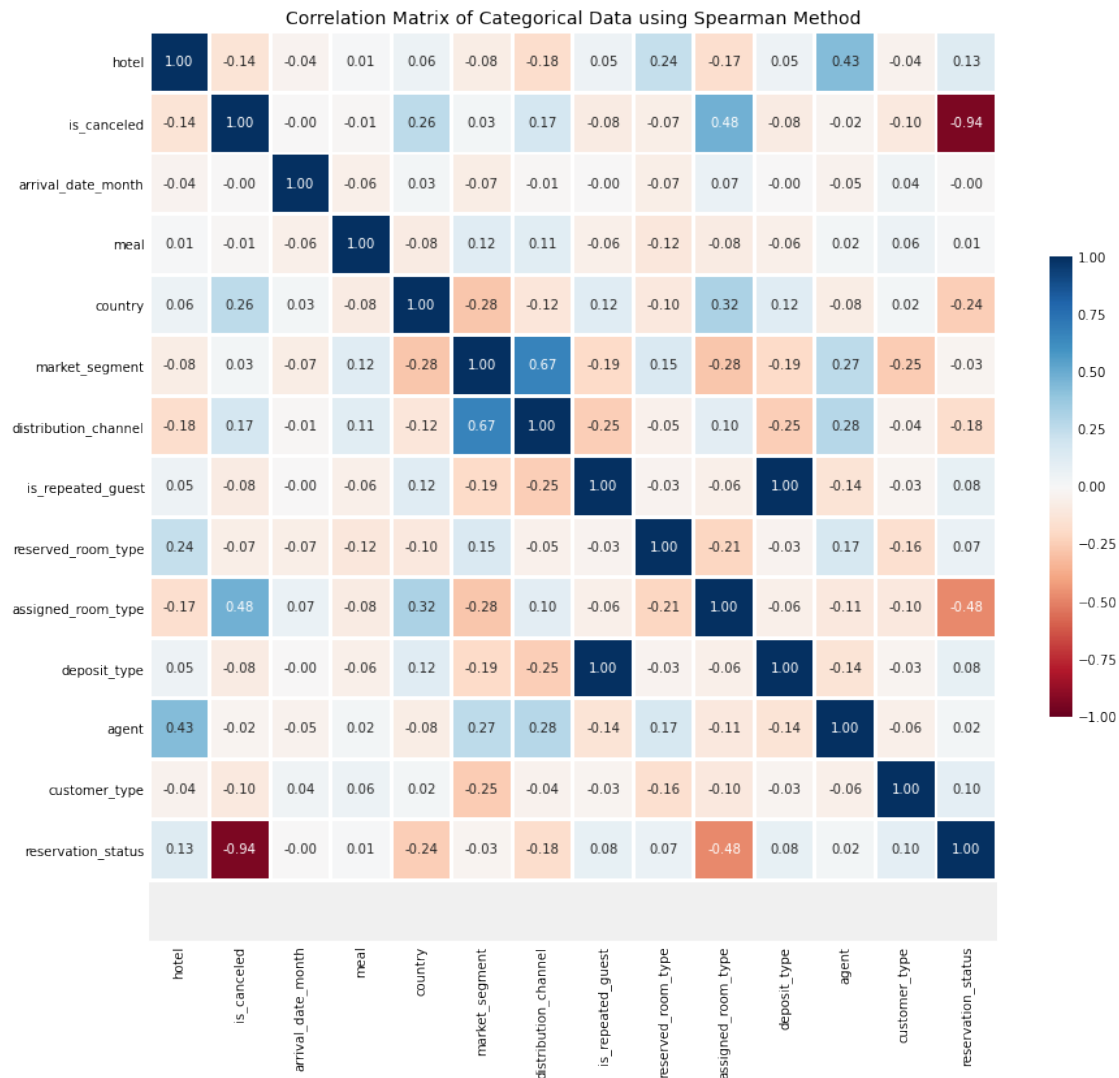
```

```

[56]: Text(0.5, 1.0, 'Correlation Matrix of Categorical Data using Spearman Method')

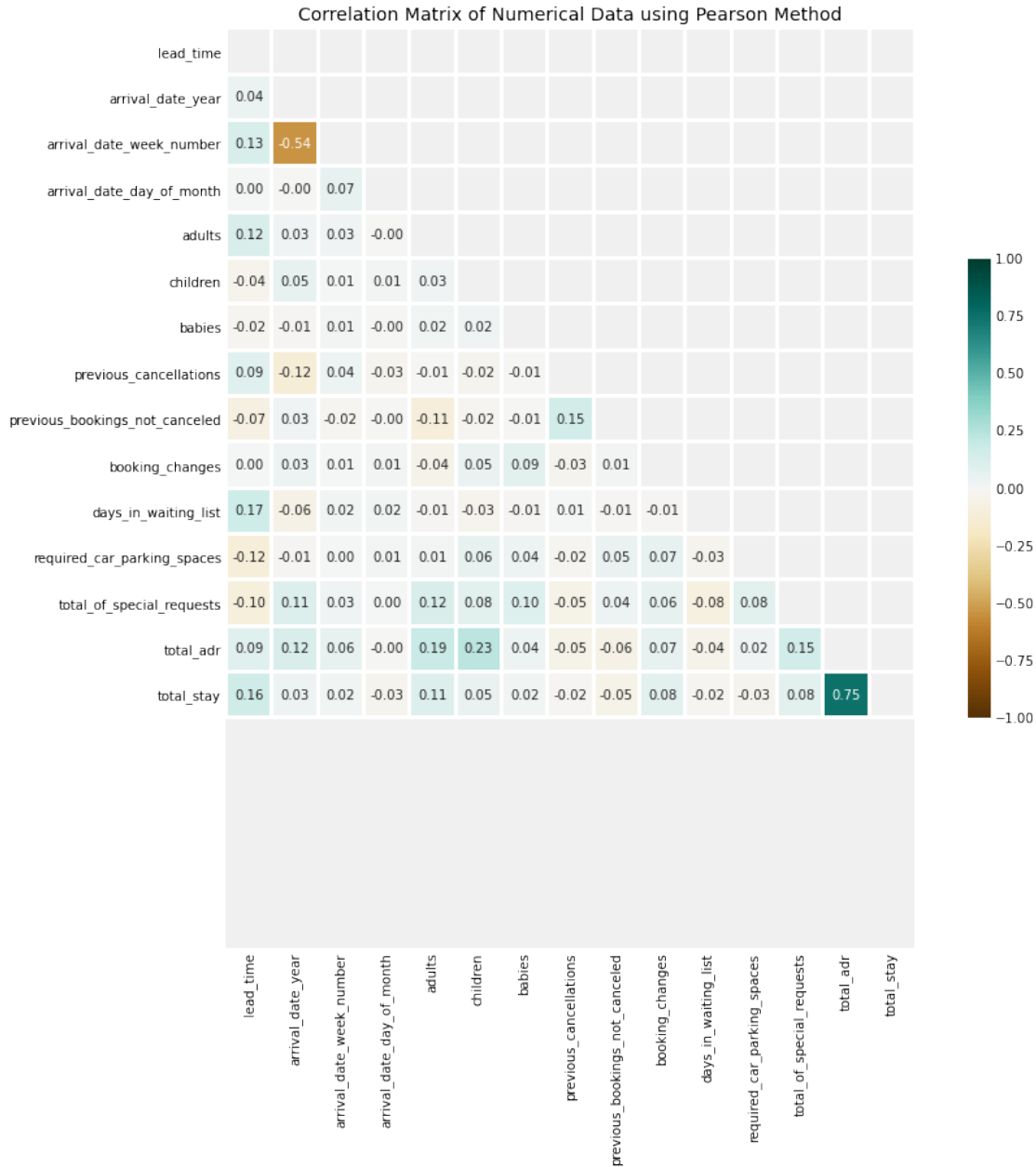
```





```
[57]: # Correlation Matrix of Numerical Data with Spearman method
plt.figure(figsize=(13,13))
corr_numerical=hotel_numerical_data.corr(method='pearson')
mask_numerical = np.triu(np.ones_like(corr_numerical, dtype=np.bool))
sns.heatmap(corr_numerical, annot=True, fmt=".2f", cmap='BrBG', mask=
    ↪mask_numerical, vmin=-1, vmax=1, center= 0,
            square=True, linewidths=2, cbar_kws={"shrink": .5}).set(ylim=(20,
    ↪0))
plt.title("Correlation Matrix of Numerical Data using Pearson Method",size=14)
```

```
[57]: Text(0.5, 1.0, 'Correlation Matrix of Numerical Data using Pearson Method')
```



```
[58]: corr_mask_categorical = corr_categorical.mask(mask_categorical)
corr_values_categorical = [c for c in corr_mask_categorical.columns if any_
    ↳(corr_mask_categorical[c] > 0.90)]
corr_mask_numerical = corr_numerical.mask(mask_numerical)
corr_values_numerical = [c for c in corr_mask_numerical.columns if any_
    ↳(corr_mask_numerical[c] > 0.90)]
print(corr_values_categorical, corr_values_numerical)
```

```
['is_repeated_guest'] []
```

Looking at the first heatmap for categorical variables 'reservation\_status' feature has very high negative correlation with 'is\_canceled', so in order avoid over fitting 'reservation\_status' feature shall be dropped.

As per the results from the correlation matrix, we shall drop 'is\_repeated\_guest' and 'arrival\_date\_week\_number'

```
[59]: frames = [hotel_numerical_data, hotel_categorical_data]
```

```
[60]: Hotel = pd.concat(frames, axis = 1)
```

```
[61]: Hotel_data = hotel.drop(['reservation_status', 'arrival_date_week_number'], axis=1)
```

```
[62]: Hotel_data.shape
```

```
[62]: (119210, 27)
```

```
[63]: Hotel_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 119210 entries, 0 to 119389
Data columns (total 27 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   hotel                                119210 non-null  int64
1   is_canceled                          119210 non-null  int64
2   lead_time                            119210 non-null  int64
3   arrival_date_year                    119210 non-null  int64
4   arrival_date_month                  119210 non-null  int64
5   arrival_date_day_of_month            119210 non-null  int64
6   adults                               119210 non-null  int64
7   children                             119210 non-null  int64
8   babies                              119210 non-null  int64
9   meal                                 119210 non-null  int64
10  country                              119210 non-null  int64
11  market_segment                      119210 non-null  int64
12  distribution_channel                 119210 non-null  int64
13  is_repeated_guest                    119210 non-null  int64
14  previous_cancellations                119210 non-null  int64
15  previous_bookings_not_canceled        119210 non-null  int64
16  reserved_room_type                   119210 non-null  int64
17  assigned_room_type                    119210 non-null  int64
18  booking_changes                       119210 non-null  int64
19  deposit_type                         119210 non-null  int64
20  agent                                119210 non-null  int64
21  days_in_waiting_list                  119210 non-null  int64
22  customer_type                         119210 non-null  int64
```

```

23 required_car_parking_spaces      119210 non-null  int64
24 total_of_special_requests        119210 non-null  int64
25 total_adr                        119210 non-null  float64
26 total_stay                      119210 non-null  int64
dtypes: float64(1), int64(26)
memory usage: 30.5 MB

```

## Hyperparameter Tuning and Feature Importance

```

[64]: Hotel_data_tunning = Hotel_data
      y = Hotel_data_tunning.iloc[:,1]
      X = pd.concat([Hotel_data_tunning.iloc[:,0],Hotel_data_tunning.iloc[:,2:26]],
                    ↪axis=1)

```

```

[65]: from sklearn.inspection import permutation_importance

```

```

[66]: # Permutation Importance graph with XGB Classifier algorithm.
      params = {
          'criterion': 'gini',
          'learning_rate': 0.01,
          'max_depth': 5,
          'n_estimators': 100,
          'objective': 'binary:logistic',
      }
      model = XGBClassifier(parameters=params)
      # fit the model
      model.fit(X, y)
      # perform permutation importance
      result = permutation_importance(model, X, y, scoring='accuracy', n_repeats = 5,
                                     ↪n_jobs=-1)
      sorted_idx = result.importances_mean.argsort()

```

```

[20:02:48] WARNING: ../src/learner.cc:573:
Parameters: { "parameters" } might not be used.

```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

```

[20:02:48] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the
default evaluation metric used with the objective 'binary:logistic' was changed
from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore
the old behavior.

```

```
[67]: # Permutation Importance graph with XGB Classifier algorithm.

params = {
    'criterion': 'gini',
    'learning_rate': 0.01,
    'max_depth': 5,
    'n_estimators': 100,
    'objective': 'binary:logistic',
}
model = XGBClassifier(parameters=params)
# fit the model
model.fit(X, y)
# perform permutation importance
result = permutation_importance(model, X, y, scoring='accuracy', n_repeats = 5,
    ↪n_jobs=-1)
sorted_idx = result.importances_mean.argsort()
```

[20:03:25] WARNING: ../src/learner.cc:573:  
Parameters: { "parameters" } might not be used.

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

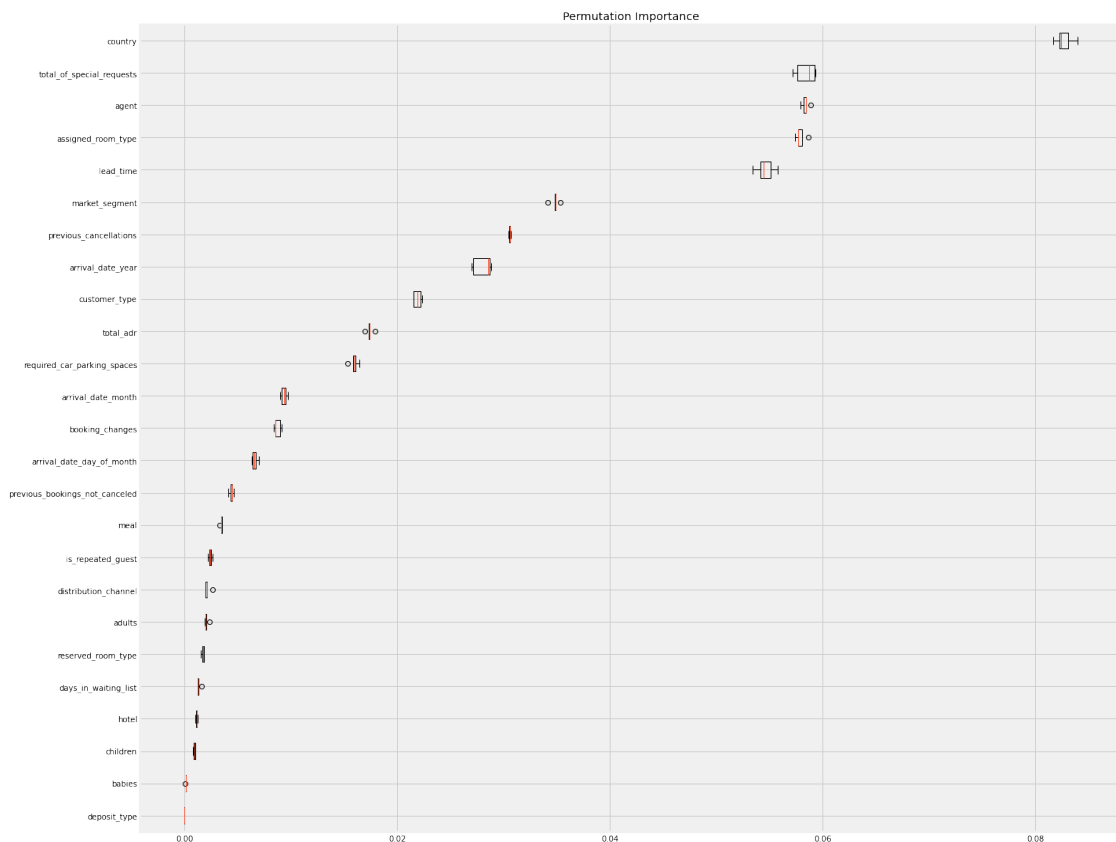
[20:03:26] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval\_metric if you'd like to restore the old behavior.

```
[68]: # Feature scores table
for i,v in enumerate(sorted_idx):
    print('Feature: %0d, Score: %.5f' % (i,v))
```

```
Feature: 0, Score: 18.00000
Feature: 1, Score: 7.00000
Feature: 2, Score: 6.00000
Feature: 3, Score: 0.00000
Feature: 4, Score: 20.00000
Feature: 5, Score: 15.00000
Feature: 6, Score: 5.00000
Feature: 7, Score: 11.00000
Feature: 8, Score: 12.00000
Feature: 9, Score: 8.00000
Feature: 10, Score: 14.00000
```

Feature: 11, Score: 4.00000  
 Feature: 12, Score: 17.00000  
 Feature: 13, Score: 3.00000  
 Feature: 14, Score: 22.00000  
 Feature: 15, Score: 24.00000  
 Feature: 16, Score: 21.00000  
 Feature: 17, Score: 2.00000  
 Feature: 18, Score: 13.00000  
 Feature: 19, Score: 10.00000  
 Feature: 20, Score: 1.00000  
 Feature: 21, Score: 16.00000  
 Feature: 22, Score: 19.00000  
 Feature: 23, Score: 23.00000  
 Feature: 24, Score: 9.00000

```
[69]: #Permutation Importance graph
fig, ax = plt.subplots(figsize=(20,15))
ax.boxplot(result.importances[sorted_idx].T,
           vert=False, labels=X.columns[sorted_idx])
ax.set_title("Permutation Importance")
fig.tight_layout()
plt.show()
```



```
[70]: hotel_model = Hotel_data_tunning.drop(['babies', 'deposit_type'], axis = 1)
```

```
[71]: hotel_model.head()
```

```
[71]:   hotel  is_canceled  lead_time  arrival_date_year  arrival_date_month \
0      1           0      342          2015           5
1      1           0      737          2015           5
2      1           0       7          2015           5
3      1           0      13          2015           5
4      1           0      14          2015           5

   arrival_date_day_of_month  adults  children  meal  country  market_segment \
0                        1        2          0    0      135           3
1                        1        2          0    0      135           3
2                        1        1          0    0       59           3
3                        1        1          0    0       59           2
4                        1        2          0    0       59           6

   distribution_channel  is_repeated_guest  previous_cancellations \
0                      1                  0                      0
1                      1                  0                      0
2                      1                  0                      0
3                      0                  0                      0
4                      3                  0                      0

   previous_bookings_not_canceled  reserved_room_type  assigned_room_type \
0                                0                    2                    0
1                                0                    2                    0
2                                0                    0                    0
3                                0                    0                    0
4                                0                    0                    0

   booking_changes  agent  days_in_waiting_list  customer_type \
0                 3     0                    0                2
1                 4     0                    0                2
2                 0     0                    0                2
3                 0   221                    0                2
4                 0   174                    0                2

   required_car_parking_spaces  total_of_special_requests  total_adr \
0                             0                          0        0.0
1                             0                          0        0.0
2                             0                          0       75.0
3                             0                          0       75.0
4                             0                          1      196.0
```

	total_stay
0	0
1	0
2	1
3	1
4	2

```
[72]: hotel_model.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 119210 entries, 0 to 119389
Data columns (total 25 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   hotel                                119210 non-null  int64
1   is_canceled                          119210 non-null  int64
2   lead_time                            119210 non-null  int64
3   arrival_date_year                    119210 non-null  int64
4   arrival_date_month                  119210 non-null  int64
5   arrival_date_day_of_month            119210 non-null  int64
6   adults                               119210 non-null  int64
7   children                             119210 non-null  int64
8   meal                                 119210 non-null  int64
9   country                              119210 non-null  int64
10  market_segment                       119210 non-null  int64
11  distribution_channel                  119210 non-null  int64
12  is_repeated_guest                     119210 non-null  int64
13  previous_cancellations                 119210 non-null  int64
14  previous_bookings_not_canceled         119210 non-null  int64
15  reserved_room_type                    119210 non-null  int64
16  assigned_room_type                    119210 non-null  int64
17  booking_changes                       119210 non-null  int64
18  agent                                 119210 non-null  int64
19  days_in_waiting_list                  119210 non-null  int64
20  customer_type                         119210 non-null  int64
21  required_car_parking_spaces            119210 non-null  int64
22  total_of_special_requests              119210 non-null  int64
23  total_adr                             119210 non-null  float64
24  total_stay                            119210 non-null  int64
dtypes: float64(1), int64(24)
memory usage: 28.6 MB
```