Hotel Booking Demand

Data set contains booking information for a city hotel and a resort hotel, and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things.

you have to perform exploratory data analysis and try to solve this question

1. How Many Booking Were Cancelled?
2. What is the booking ratio between Resort Hotel and City Hotel?
3. What is the percentage of booking for each year?
4. Which is the most busy month for hotel?
5. From which country most guest come?
6. How Long People Stay in the hotel?
7. Which was the most booked accommodation type (Single, Couple, Family)?

Details of Dataset

1. hotel :(H1 = Resort Hotel or H2 = City Hotel).
2. is_canceled Value: showing if the booking had been cancelled (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: The months in which guests are coming.
6. arrival_date_week_number: Week number of year for arrival date.
7. arrival_date_day_of_month: Which day of the months guest is arriving.
8. stays_in_weekend_nights: Number of weekend stay at night (Saturday or Sunday) the guest stayed or booked to stay at the hotel.
9. stays_in_week_nights: Number of weekdays stay at night (Monday to Friday) in the hotel.

10. adults: Number of adults.
11. children: Number of children.
12. babies: Number of babies.
13. meal: Type of meal booked.
14. country: Country of origin.
15. market_segment: Through which channel hotels were booked.
16. distribution_channel: Booking distribution channel.
17. is_repeated_guest: The values indicating if the booking name was from a repeated guest (1) or not (0).
18. previous_cancellations: Show if the repeated guest has cancelled the booking before.
19. previous_bookings_not_canceled: Show if the repeated guest has not cancelled the booking before.
20. reserved_room_type: Code of room type reserved. Code is presented instead of designation for anonymity reasons.
21. assigned_room_type: Code for the type of room assigned to the booking. Code is presented instead of designation for anonymity reasons.
22. booking_changes: How many times did booking changes happen.
23. deposit_type: Indication on if the customer deposited something to confirm the booking.
24. agent: If the booking happens through agents or not.
25. company: If the booking happens through companies, the company ID that made the booking or responsible for paying the booking.
26. days_in_waiting_list: Number of days the booking was on the waiting list before the confirmation to the customer.
27. customer_type: Booking type like Transient – Transient-Party – Contract – Group.

28. adr: Average Daily Rates that described via way of means of dividing the sum of all accommodations transactions using entire numbers of staying nights.
29. required_car_parking_spaces: How many parking areas are necessary for the customers.
30. total_of_special_requests: Total unique requests from consumers.
31. reservation_status: The last status of reservation, assuming one of three categories: Canceled – booking was cancelled by the customer; Check-Out
32. reservation_status_date: The last status date.

Exploratory Data Analysis

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

```
data = pd.read_csv('hotel_bookings.csv')
```

```
df = data.copy()
```

Find the missing value, show the total null values for each column and sort it in descending order

```
df.isnull().sum().sort_values(ascending=False)[:10]
```

```
## Drop Rows where there is no adult, baby and child
df = df.drop(df[(df.adults+df.babies+df.children)==0].index)
```

```
## If no id of agent or company is null, just replace it with 0
df['company']=df['company'].fillna(0)
df['agent'] = df['agent'].fillna(0)
```

```
#For the missing values in the country column, replace it with mode
df['country'].fillna(data.country.mode().to_string(), inplace=True)
```

```
## for missing children value, replace it with rounded mean value
df['children'].fillna(round(data.children.mean()), inplace=True)
```

```
df.info()
```

```
## convert datatype of these columns from float to integer
df['children'] = df['children'].astype('int64')
df['company'] = df['company'].astype('int64')
df['agent'] = df['agent'].astype('int64')
```

```
df.info()
```

```
# 1. How Many Booking Were Cancelled
df['is_canceled'].value_counts()
```

```
df['is_canceled'].value_counts().plot(kind='bar')
```

```
# 2. What is the booking ratio between Resort Hotel and City Hotel?
```

```
df_not_canceled = df[df['is_canceled'] == 0]
```

```
df_not_canceled['hotel'].value_counts()
```

```
series=df_not_canceled['hotel'].value_counts()
ratio = round(series/series.sum()*100)
```

```
sns.barplot(x=ratio.index,y=ratio.values,data=ratio)
```

```
# 3. What is the percentage of booking for each year?
```

```
df_not_canceled['arrival_date_year'].value_counts()
```

```
yr_data=df_not_canceled['arrival_date_year'].value_counts()
```

```
yr=round(yr_data/yr_data.sum()*100)
yr
```

```
sns.barplot(x=yr.index,y=yr.values,data=yr)
```

```
sns.countplot(x='arrival_date_year', hue='hotel',  data=df_not_canceled);
```

```
# 4. Which is the most busy month for hotel?
```

```
df_not_canceled['arrival_date_month'].value_counts()
```

```
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July',
'August', 'September', 'October', 'November', 'December']

sorted_months =
df_not_canceled['arrival_date_month'].value_counts().reindex(new_order)
sorted_months
```

```
plt.figure(figsize=(18, 6))
sorted_data = sorted_months/sorted_months.sum()*100

sns.lineplot(x=sorted_data.index,y=sorted_data.values,data=sorted_data)
```

```
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July',
'August', 'September', 'October', 'November', 'December']

## Select only City Hotel
sorted_months = df_not_canceled.loc[df.hotel=='City Hotel'
,'arrival_date_month'].value_counts().reindex(new_order)

city_data=sorted_months/sorted_months.sum()*100


## Select only Resort Hotel
sorted_months = df_not_canceled.loc[df.hotel=='Resort Hotel'
,'arrival_date_month'].value_counts().reindex(new_order)


resort_data = sorted_months/sorted_months.sum()*100
```

```
plt.subplots(figsize=(18,6))
sns.lineplot(x=city_data.index,y=city_data.values,data=city_data,label='city
Hotel')
sns.lineplot(x=resort_data.index,y=resort_data.values,data=resort_data,label='res
ort Hotel')
plt.xlabel('Months')
plt.ylabel('Booking (%)')
```

```
### `5. From which country most guest come? `
```

```
# `**pycountry** is very useful python package.`
# `We will use this package to get country names from country codes`

# - `https://github.com/flyingcircusio/pycountry`
# - `https://pypi.org/project/pycountry/`
```

```
pip install pycountry
```

```
import pycountry as pc
```

```
country_data=df_not_canceled['country'].value_counts()[:10]
```

```
country_name = [pc.countries.get(alpha_3=name).name for name in
country_data.index]
country_name
```

```
# sns.barplot(country_name,country_data.values)
sns.barplot(x=country_name,y=country_data.values,data=country_data)
plt.xticks(rotation = 90)
plt.show()
```

```
### `6. How Long People Stay in the hotel?`
```

```
total_nights = df_not_canceled['stays_in_weekend_nights']+
df_not_canceled['stays_in_week_nights']
total_data=total_nights.value_counts()[0:10]
```

```
total_night_data=total_data/total_data.sum()*100
```

```
#
sns.barplot(x=total_night_data.index,y=total_night_data.values,data=total_night_d
ata)
plt.bar(total_night_data.index,total_night_data.values)
```

```
df_not_canceled.loc[:,'total_nights'] =
df_not_canceled['stays_in_weekend_nights']+
df_not_canceled['stays_in_week_nights']

fig, ax = plt.subplots(figsize=(12,6))
ax.set_xlabel('No of Nights')
ax.set_ylabel('No of Nights')
ax.set_title('Hotel wise night stay duration (Top 10)')
sns.countplot(x='total_nights', hue='hotel', data=df_not_canceled,
              order =
df_not_canceled.total_nights.value_counts().iloc[:10].index, ax=ax)
```

```
### `7.Which was the most booked accommodation type (Single, Couple, Family)?`
```

```
## Select single, couple, multiple adults and family
single   = df_not_canceled[(df_not_canceled.adults==1) &
(df_not_canceled.children==0) & (df_not_canceled.babies==0)]
couple   = df_not_canceled[(df_not_canceled.adults==2) &
(df_not_canceled.children==0) & (df_not_canceled.babies==0)]
#n_adults = df_not_canceled[(df_not_canceled.adults>2) &
(df_not_canceled.children==0) & (df_not_canceled.babies==0)]
family   = df_not_canceled[df_not_canceled.adults + df_not_canceled.children +
df_not_canceled.babies > 2]


## Make the list of Category names, and their total percentage
names = ['Single', 'Couple (No Children)', 'Family / Friends']
count = [single.shape[0],couple.shape[0], family.shape[0]]
count_percent = [x/df_not_canceled.shape[0]*100 for x in count]
```

```
sns.barplot(x=names,y=count_percent)
```