

```
In [1]: import pandas as pd  
        import numpy as np  
        import matplotlib.pyplot as plt  
        import seaborn as sns
```

```
In [2]: df = pd.read_csv(r'C:\Users\naila.iram\Downloads\Hotel_Room_Booking_Anlaysis\hotel_b
```

In [3]: df.head(5)

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_ni
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	

5 rows × 32 columns

In [4]: df.dtypes

```
Out[4]: 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
dtype: object
```

In [5]: df.shape

Out[5]: (119390, 32)

In [6]: df.columns

```
Out[6]: Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',
       'arrival_date_month', 'arrival_date_week_number',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'adults', 'children', 'babies', 'meal',
       'country', 'market_segment', 'distribution_channel',
       'is_repeated_guest', 'previous_cancellations',
       'previous_bookings_not_canceled', 'reserved_room_type',
       'assigned_room_type', 'booking_changes', 'deposit_type', 'agent',
       'company', 'days_in_waiting_list', 'customer_type', 'adr',
       'required_car_parking_spaces', 'total_of_special_requests',
       'reservation_status', 'reservation_status_date'],
      dtype='object')
```

In [9]: df.duplicated().sum()

Out[9]: 31994

In [7]: `filter1=(df['adults']==0) & (df['babies']==0) & (df['children']==0)`In [8]: `df[filter1]`

Out[8]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_weekday
40984	City Hotel	0	1	2015	August	
41048	City Hotel	0	104	2015	August	
41446	City Hotel	0	3	2015	August	
41952	City Hotel	0	15	2015	August	
45158	City Hotel	1	48	2015	October	
...	...	...	...	...	...	...
117204	City Hotel	0	296	2017	July	
117274	City Hotel	0	276	2017	July	
117303	City Hotel	0	291	2017	July	
117453	City Hotel	0	159	2017	July	
118200	City Hotel	0	10	2017	August	

220 rows × 32 columns

In [10]: `df[~filter1].shape`

Out[10]: (119170, 32)

In [11]: `df2=df[~filter1]`In [12]: `df2.duplicated().sum()`

Out[12]: 31990

In [14]: `data=df2.drop_duplicates()`

```
In [15]: data.shape
```

```
Out[15]: (87180, 32)
```

```
In [16]: data.isna().sum()
```

```
Out[16]: 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                        452
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                           12176
company                          81923
days_in_waiting_list             0
customer_type                   0
adr                             0
required_car_parking_spaces      0
total_of_special_requests        0
reservation_status                0
reservation_status_date          0
dtype: int64
```

```
In [17]: ### Where do the guest come from?
```

```
In [19]: not_cancelled=data[data['is_canceled']==0]
```

```
In [22]: country_wise_data=not_cancelled['country'].value_counts().reset_index()
```

```
In [24]: country_wise_data.columns = ['Country','No of Guests']
```

```
In [25]: country_wise_data
```

Out[25]:

	Country	No of Guests
0	PRT	17632
1	GBR	8436
2	FRA	7084
3	ESP	5380
4	DEU	4324
...	...	...
160	ZMB	1
161	SYC	1
162	MDG	1
163	SMR	1
164	FRO	1

165 rows × 2 columns

In [26]:

```
!pip install chart-studio
!pip install plotly
```

Requirement already satisfied: chart-studio in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (1.1.0)  
Requirement already satisfied: plotly in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (from chart-studio) (5.9.0)  
Requirement already satisfied: requests in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (from chart-studio) (2.31.0)  
Requirement already satisfied: retrying>=1.3.3 in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (from chart-studio) (1.4.2)  
Requirement already satisfied: six in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (from chart-studio) (1.16.0)  
Requirement already satisfied: tenacity>=6.2.0 in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (from plotly->chart-studio) (8.2.2)  
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (from requests->chart-studio) (2.0.4)  
Requirement already satisfied: idna<4,>=2.5 in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (from requests->chart-studio) (3.4)  
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (from requests->chart-studio) (2.0.7)  
Requirement already satisfied: certifi>=2017.4.17 in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (from requests->chart-studio) (2024.2.2)  
Requirement already satisfied: plotly in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (5.9.0)  
Requirement already satisfied: tenacity>=6.2.0 in c:\users\naila.iram\appdata\local\anaconda3\lib\site-packages (from plotly) (8.2.2)

In [27]:

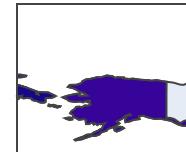
```
import chart_studio.plotly as py
import plotly.graph_objs as go
import plotly.express as px
```

```
from plotly.offline import download_plotlyjs , init_notebook_mode ,plot ,iplot  
init_notebook_mode(connected=True)
```

```
In [28]: map_guests = px.choropleth(data_frame=country_wise_data,  
locations=country_wise_data['Country'],  
color=country_wise_data['No of Guests'],  
hover_name=country_wise_data['Country'],  
title="Home Country of Guests")
```

```
In [29]: map_guests.show()
```

## Home Country of Guests



```
In [30]: ### Analysing D/F between assigned and reserved room types.
```

```
In [31]: data.columns
```

```
Out[31]: Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',
       'arrival_date_month', 'arrival_date_week_number',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'adults', 'children', 'babies', 'meal',
       'country', 'market_segment', 'distribution_channel',
       'is_repeated_guest', 'previous_cancellations',
       'previous_bookings_not_canceled', 'reserved_room_type',
       'assigned_room_type', 'booking_changes', 'deposit_type', 'agent',
       'company', 'days_in_waiting_list', 'customer_type', 'adr',
       'required_car_parking_spaces', 'total_of_special_requests',
       'reservation_status', 'reservation_status_date'],
      dtype='object')
```

```
In [35]: pivot = pd.crosstab(index=data['reserved_room_type'], columns=data['assigned_room_ty
```

```
In [36]: pivot
```

assigned_room_type	A	B	C	D	E	F	G	H	I	K	L	P
reserved_room_type												
<b>A</b>	81.0	2.0	2.0	11.0	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
<b>B</b>	10.0	88.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
<b>C</b>	1.0	0.0	95.0	1.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0
<b>D</b>	2.0	0.0	0.0	92.0	4.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
<b>E</b>	0.0	0.0	0.0	0.0	90.0	6.0	2.0	0.0	1.0	0.0	0.0	0.0
<b>F</b>	0.0	1.0	0.0	0.0	1.0	93.0	4.0	0.0	0.0	0.0	0.0	0.0
<b>G</b>	0.0	0.0	0.0	0.0	0.0	1.0	97.0	0.0	1.0	0.0	0.0	0.0
<b>H</b>	0.0	0.0	0.0	0.0	0.0	0.0	2.0	97.0	1.0	0.0	0.0	0.0
<b>L</b>	17.0	17.0	17.0	0.0	0.0	17.0	0.0	17.0	0.0	0.0	17.0	0.0
<b>P</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
<b>All</b>	53.0	2.0	2.0	26.0	8.0	4.0	3.0	1.0	0.0	0.0	0.0	0.0

```
In [37]: ### Which market segment has highest booking ?
data['market_segment'].value_counts()
```

```
Out[37]: market_segment
Online TA      51437
Offline TA/T0  13886
Direct         11776
Groups          4942
Corporate       4212
Complementary   698
Aviation        227
Undefined       2
Name: count, dtype: int64
```

```
In [38]: fig = px.pie(data,
                     values=data['market_segment'].value_counts().values,
                     names = data['market_segment'].value_counts().index )
```

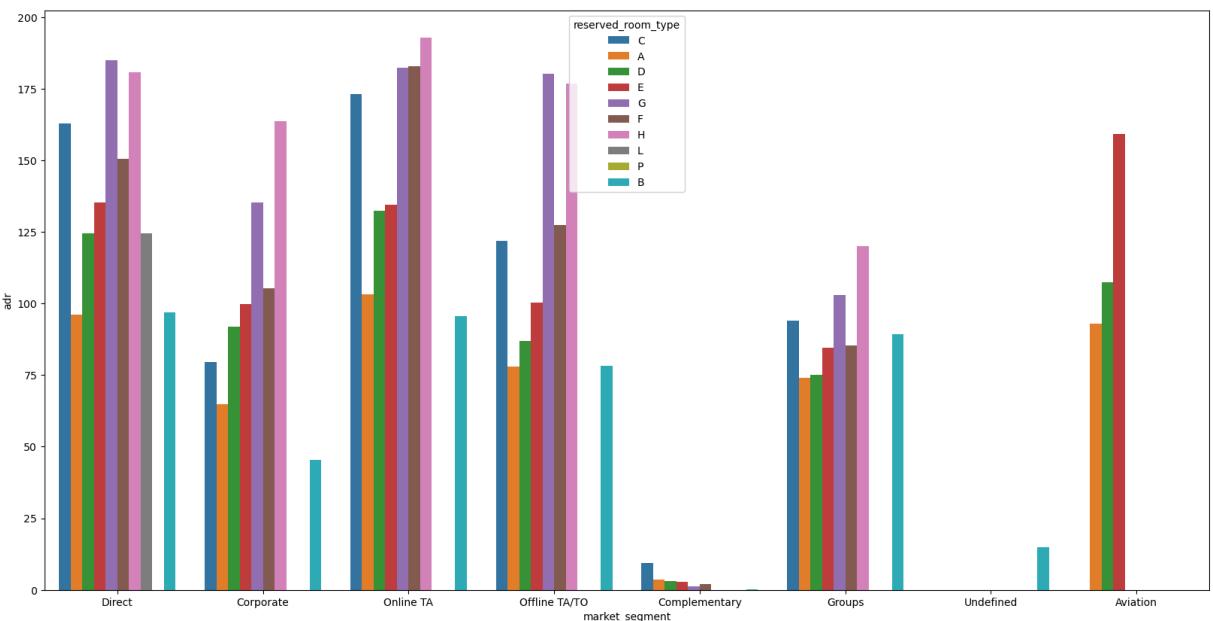
```
In [39]: fig.show()
```

```
In [40]: ### Analysis of Avg price per night (ADR) of various room-types for all the market
plt.figure(figsize=(20,10))
sns.barplot(x="market_segment",y= "adr" , hue= "reserved_room_type",data=data , ci=
```

C:\Users\naila.iram\AppData\Local\Temp\ipykernel\_25752\2254527706.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

```
Out[40]: <Axes: xlabel='market_segment', ylabel='adr'>
```



In [41]: *### Under Pattern in Guest Arrival*

In [44]: 

```
import warnings
from warnings import filterwarnings
filterwarnings('ignore')
```

In [42]: `data['arrival_date_month'].unique()`

Out[42]: `array(['July', 'August', 'September', 'October', 'November', 'December',
 'January', 'February', 'March', 'April', 'May', 'June'],
 dtype=object)`

In [43]: `dict_month={'July':7, 'August':8, 'September':9, 'October':10, 'November':11, 'Dece
 'January':1, 'February':2, 'March':3, 'April':4, 'May':5, 'June':6}`

In [45]: `data['arrival_date_month_index'] = data['arrival_date_month'].map(dict_month)`

In [46]: `data.columns`

Out[46]: `Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',
 'arrival_date_month', 'arrival_date_week_number',
 'arrival_date_day_of_month', 'stays_in_weekend_nights',
 'stays_in_week_nights', 'adults', 'children', 'babies', 'meal',
 'country', 'market_segment', 'distribution_channel',
 'is_repeated_guest', 'previous_cancellations',
 'previous_bookings_not_canceled', 'reserved_room_type',
 'assigned_room_type', 'booking_changes', 'deposit_type', 'agent',
 'company', 'days_in_waiting_list', 'customer_type', 'adr',
 'required_car_parking_spaces', 'total_of_special_requests',
 'reservation_status', 'reservation_status_date',
 'arrival_date_month_index'],
 dtype='object')`

In [48]: `data[['arrival_date_year', 'arrival_date_month_index', 'arrival_date_day_of_month']]`

Out[48]:

	arrival_date_year	arrival_date_month_index	arrival_date_day_of_month
0	2015	7	1
1	2015	7	1
2	2015	7	1
3	2015	7	1
4	2015	7	1
...	...	...	...
119385	2017	8	30
119386	2017	8	31
119387	2017	8	31
119388	2017	8	31
119389	2017	8	29

87180 rows × 3 columns

In [52]: `data['arrival_date']=data['arrival_date_year'].astype(str) + '-' + data['arrival_da`In [53]: `data.head(2)`

Out[53]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date屿
0	Resort Hotel	0	342	2015	July		
1	Resort Hotel	0	737	2015	July		

2 rows × 34 columns

In [54]: `data['Total_guests'] = data['adults'] + data['babies'] + data['children']`In [55]: `data.head(2)`

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_ni
0	Resort Hotel	0	342	2015	July	
1	Resort Hotel	0	737	2015	July	

2 rows × 35 columns

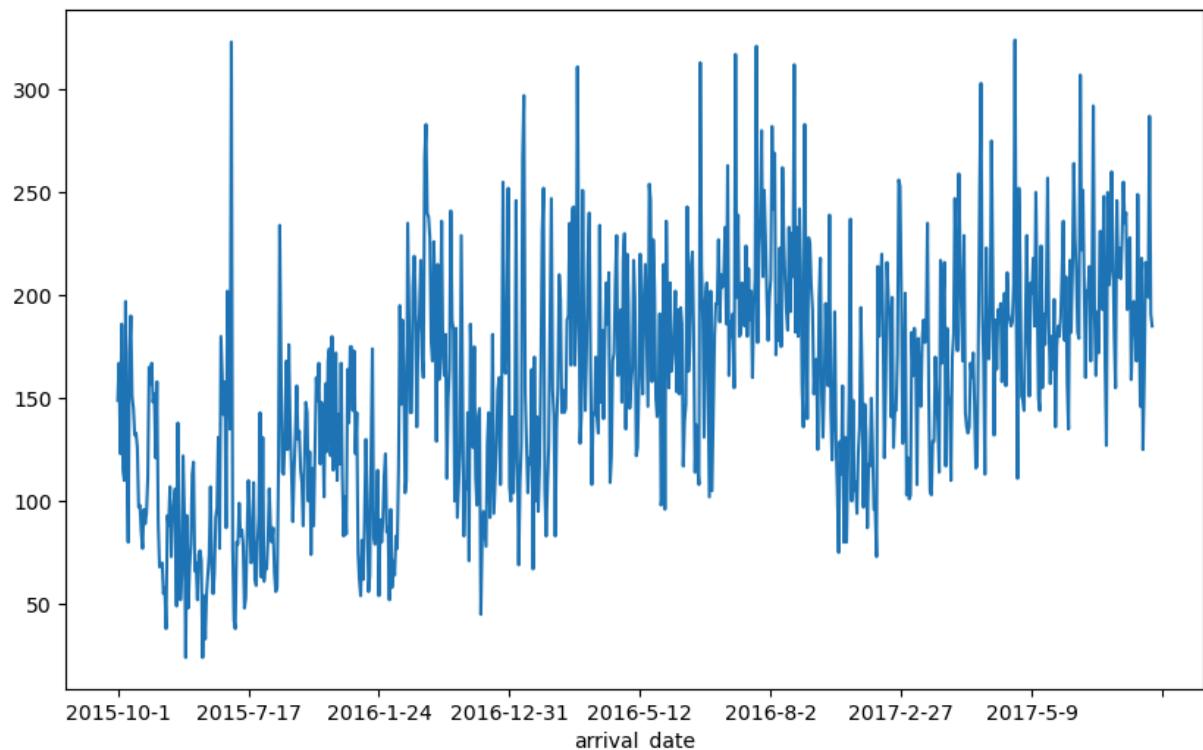


```
In [58]: data_not_cancelled = data[data['is_canceled']==0]
```

```
In [60]: guest_arrical_series = data_not_cancelled.groupby(data['arrival_date'])['Total_gues
```

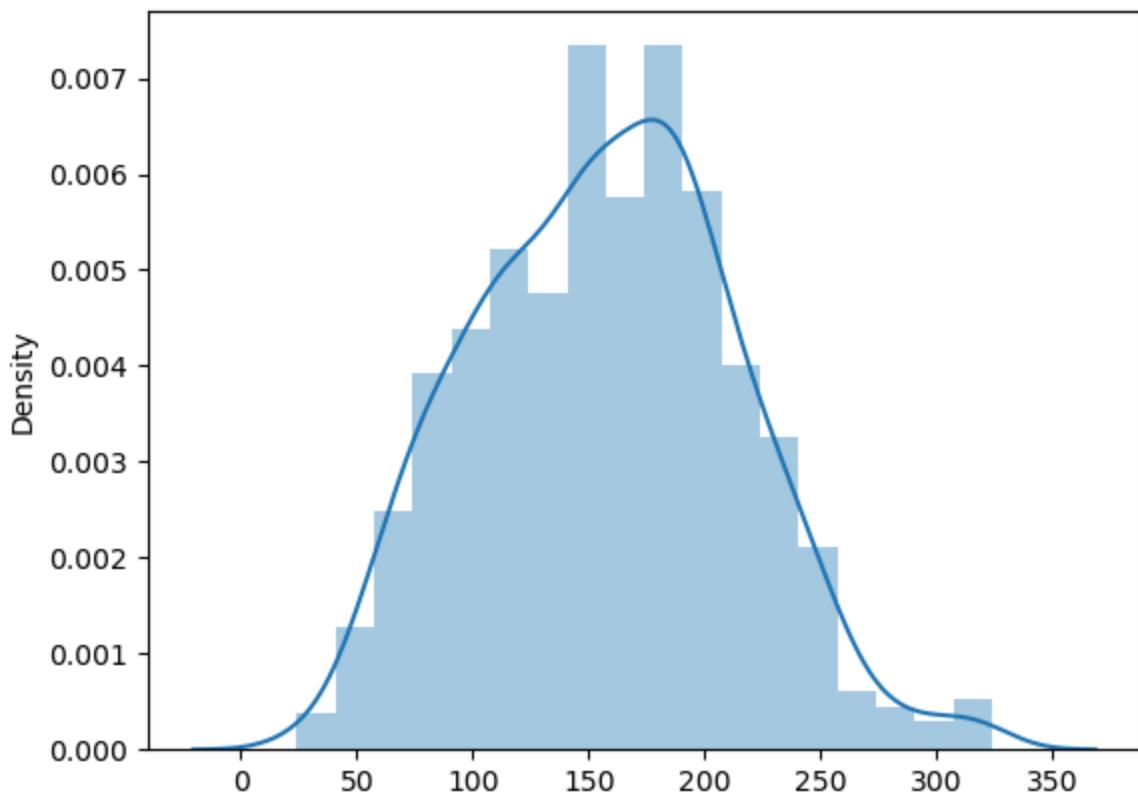
```
In [63]: guest_arrical_series.plot(figsize=(10,6))
```

```
Out[63]: <Axes: xlabel='arrival_date'>
```



```
In [64]: sns.distplot(guest_arrical_series.values)
```

```
Out[64]: <Axes: ylabel='Density'>
```



In [ ]: