Capstone Project

EDA Hotel Booking Analysis

Name- Pranav Singhal

Batch-AlmaBetter Pro

Github- https://github.com/pranav168/Hotel-Booking-Analysis

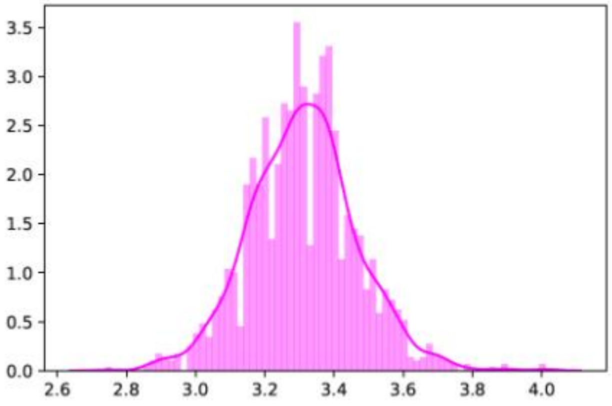
Why Hotel Booking Analysis?

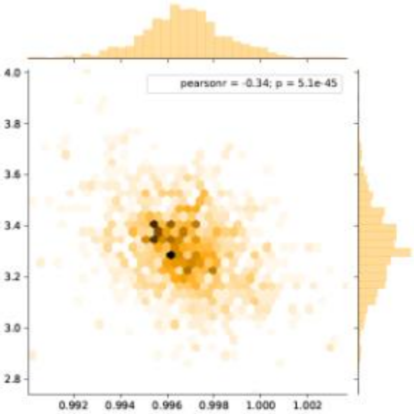
Hotel Booking Analysis on intern analysis of any business is really very important for making the base of our business strong it throw light on the working or the firm, therefore give a chance for introspection thus a chance to improve, also it provides with various insights, which could be used to uplift:

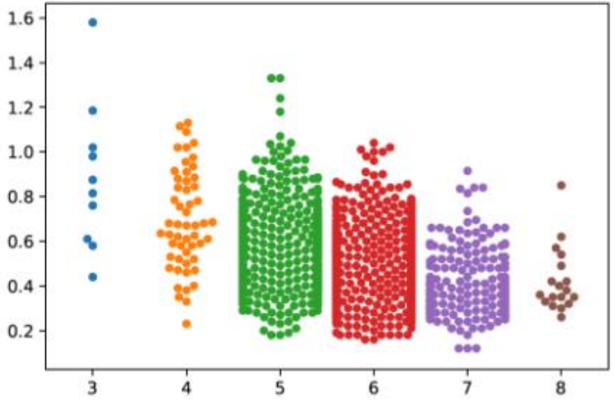
- Marketing efficiency
- Determine new market opportunities
- Better brand strategy
- Improve distribution strategies
- Customer retention

What is Exploratory Data Analysis?

- Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.







The success of a Exploratory Data Analysis

The success of a **Exploratory Data Analysis**, however, does not depend solely on the selection on the type of plots or visualizations. Key factors contributing to the success of the EDA include:

- **Data**

Data is the very prerequisite for any EDA, you cannot get a reliable insights from the analysis without a sufficient amount of rich data.

- **Feature Engineering**

It includes **data cleaning, creating new features, and feature selection**. Feature engineering usually is the most time-consuming process.

- **Perfect Plots**

Choosing the perfect Plots and Features to depict your point.

Problem Description

- Have you ever wondered when the best time of year to book a hotel room is?
- Or the optimal length of stay in order to get the best daily rate?
- What if you wanted to predict whether or not a hotel was likely to receive a disproportionately high number of special requests?

This hotel booking dataset can help you explore those questions!

Objective

Explore and analyze the data to discover important factors that govern the bookings.

Data Description

This 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. All personally identifying information has been removed from the data.

```
df.head()
```

	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
0	Resort Hotel	0	342	2015	July	27	1	0	0	2	0.0	0	BB	PRT	Direct
1	Resort Hotel	0	737	2015	July	27	1	0	0	2	0.0	0	BB	PRT	Direct
2	Resort Hotel	0	7	2015	July	27	1	0	1	1	0.0	0	BB	GBR	Direct
3	Resort Hotel	0	13	2015	July	27	1	0	1	1	0.0	0	BB	GBR	Corporate
4	Resort Hotel	0	14	2015	July	27	1	0	2	2	0.0	0	BB	GBR	Online TA

Exploration, namely: head, tail, summary, data dictionary

```
df.tail()
```

	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
119385	City Hotel	0	23	2017	August	35	30	2	5	2	0.0	0	BB	BEL	Offline TA/TO
119386	City Hotel	0	102	2017	August	35	31	2	5	3	0.0	0	BB	FRA	Online TA
119387	City Hotel	0	34	2017	August	35	31	2	5	2	0.0	0	BB	DEU	Online TA
119388	City Hotel	0	109	2017	August	35	31	2	5	2	0.0	0	BB	GBR	Online TA
119389	City Hotel	0	205	2017	August	35	29	2	7	2	0.0	0	HB	DEU	Online TA

```
df.describe()
#minimum adr is -ve which is not possible lets look into it
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	is_repeated_guest	previous_cancella
count	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119386.000000	119390.000000	119390.000000	119390.0
mean	0.370416	104.011416	2016.156554	27.165173	15.798241	0.927599	2.500302	1.856403	0.103890	0.007949	0.031912	0.0
std	0.482918	106.863097	0.707476	13.605138	8.780829	0.998613	1.908286	0.579261	0.398561	0.097436	0.175767	0.8
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	0.000000	1.000000	2.000000	0.000000	0.000000	0.000000	0.0
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	1.000000	2.000000	2.000000	0.000000	0.000000	0.000000	0.0
75%	1.000000	160.000000	2017.000000	38.000000	23.000000	2.000000	3.000000	2.000000	0.000000	0.000000	0.000000	0.0
max	1.000000	737.000000	2017.000000	53.000000	31.000000	19.000000	50.000000	55.000000	10.000000	10.000000	1.000000	26.0

	country	country_count	hotel	hotel_count	is_canceled	is_canceled_count	lead_time	lead_time_count	arrival_date_year	arrival_date_year_count	arrival_date_month	arrival_date_month_count	arrival_date_week_number	arrival_date_week_number_count	arrival_date_
0	PRT	47040	City Hotel	78122.000000	0.000000	73419.000000	0	5769	2016.000000	55789.000000	August	13711	33	3544	17
1	GBR	12055	Resort Hotel	39308.000000	1.000000	44011.000000	1	3285	2017.000000	40231.000000	July	12491	30	3030	5
2	FRA	10360					2	1991	2015.000000	21410.000000	May	11611	34	3015	15
3	ESP	8489					3	1741			April	10953	32	2997	25
4	DEU	7246					4	1646			October	10929	18	2895	26
5	ITA	3749					5	1503			June	10819	28	2814	9
6	IRL	3369					6	1374			September	10351	21	2791	16
7	BEL	2334					7	1299			March	9641	17	2778	12
8	BRA	2211					8	1116			February	7921	20	2745	19
9	NLD	2100					12	1054			November	6641	29	2736	2

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   arrival_date_year               119390 non-null  int64
 4   arrival_date_month              119390 non-null  object
 5   arrival_date_week_number        119390 non-null  int64
 6   arrival_date_day_of_month       119390 non-null  int64
 7   stays_in_weekend_nights         119390 non-null  int64
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         118902 non-null  object
 14  market_segment                  119390 non-null  object
 15  distribution_channel            119390 non-null  object
 16  is_repeated_guest               119390 non-null  int64
 17  previous_cancellations          119390 non-null  int64
 18  previous_bookings_not_canceled  119390 non-null  int64
 19  reserved_room_type              119390 non-null  object
 20  assigned_room_type              119390 non-null  object
 21  booking_changes                 119390 non-null  int64
 22  deposit_type                    119390 non-null  object
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64
 26  customer_type                   119390 non-null  object
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  object
```

Looking for and handling NaN/ Missing Values

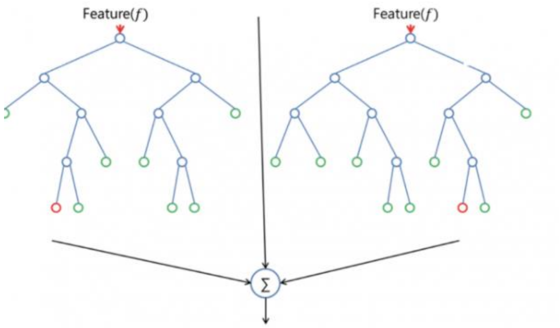
- Only 4 Nan Values in children which I will replace by Zero as it will have no effect on our Data
- There are 488 Nan Values in Country which I will predict using ML Model called Random Forest
- Same Goes For agents, I will use Random Forest to predict these Values.
- Out of 119390 total Values 112593 Values are missing for Company thus I will have to drop this column as reproductivity of this column could lead to wrong interpretations.

```
df.isna().sum()

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
```

Predicting Values of Nan using Random Forest

- Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.
- One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems.



- Predicting Countries:

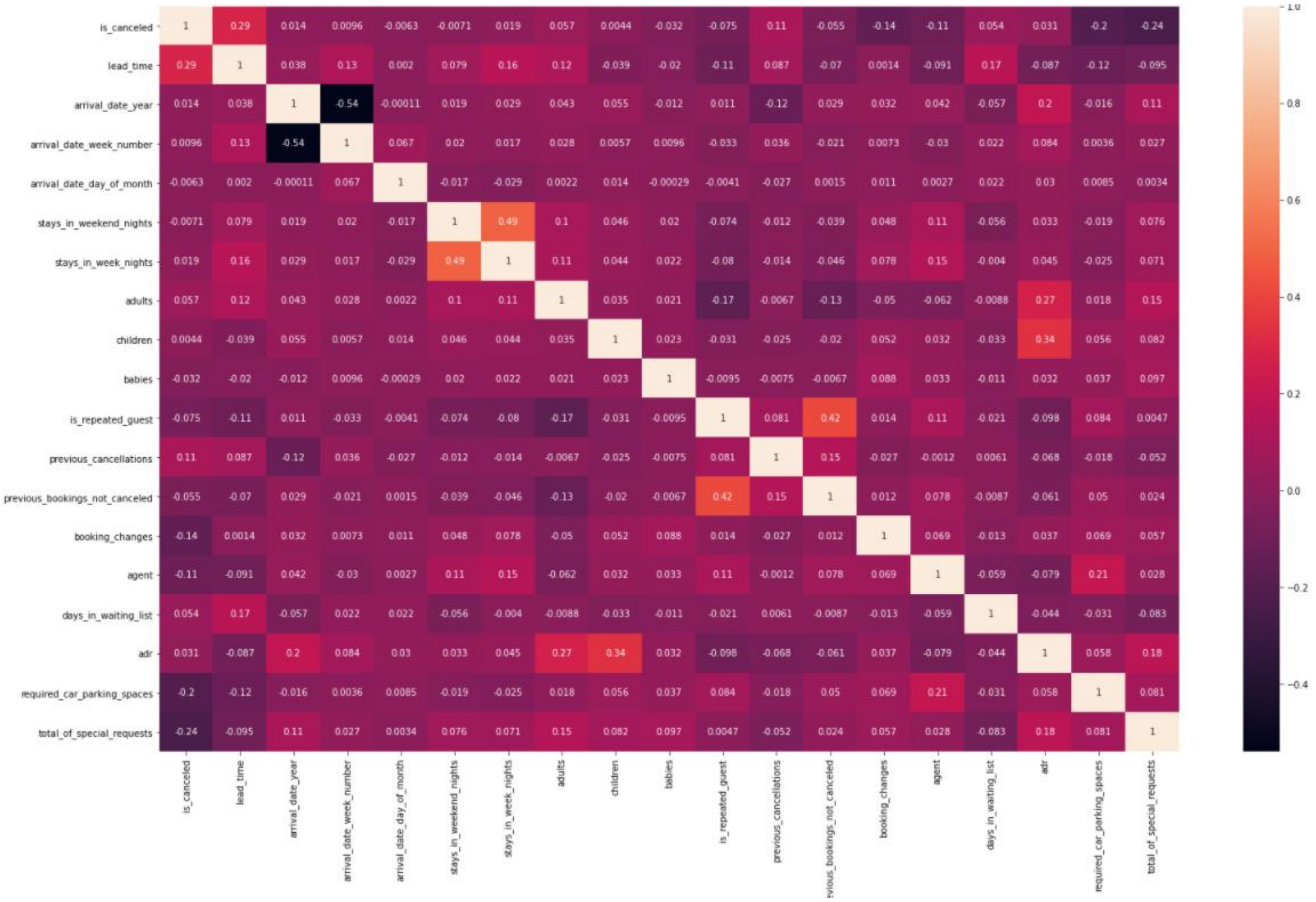
```
df_top5_country=df[(df['country']=='GBR') | (df['country']=='PRT')| (df['country']=='FRA')| (df['country']=='ESP')| (df['country']=='DEU')]
nan_country=df[df['country'].isna()]
'''-----------------------------------------------------------------------------------------------------------predicting countries-
def country2int(x):
  country_list=['PRT','GBR','FRA','ESP','DEU']
  count=0
  for i in country_list:
    count+=1
    if x==i:
      return count

df_top5_country['country']=df_top5_country['country'].apply(lambda x :country2int(x))
X_train=df_top5_country[df_top5_country.describe().columns].drop(['country','agent'],axis=1)
y_train=df_top5_country['country']
X_test= nan_country[nan_country.describe().columns].drop(['agent'],axis=1)
def RandomForest_model(X_train,y_train,X_test):
  model=RandomForestClassifier(n_estimators=100,max_depth=20,random_state=0)
  model.fit(X_train,y_train)
  y_pred=model.predict(X_test)
  return y_pred
y_pred=RandomForest_model(X_train,y_train,X_test)
nan_country['country']=y_pred
def int2country(x):
  country_list=['PRT','GBR','FRA','ESP','DEU']
  for i in range(1,6):
    if str(x)==str(i):
      return country_list[i-1]
nan_country['country']=nan_country['country'].apply(lambda x : int2country(x))
nan_country.country.value_counts()
for i in nan_country.index:
    df.loc[i]=nan_country.loc[i]
```

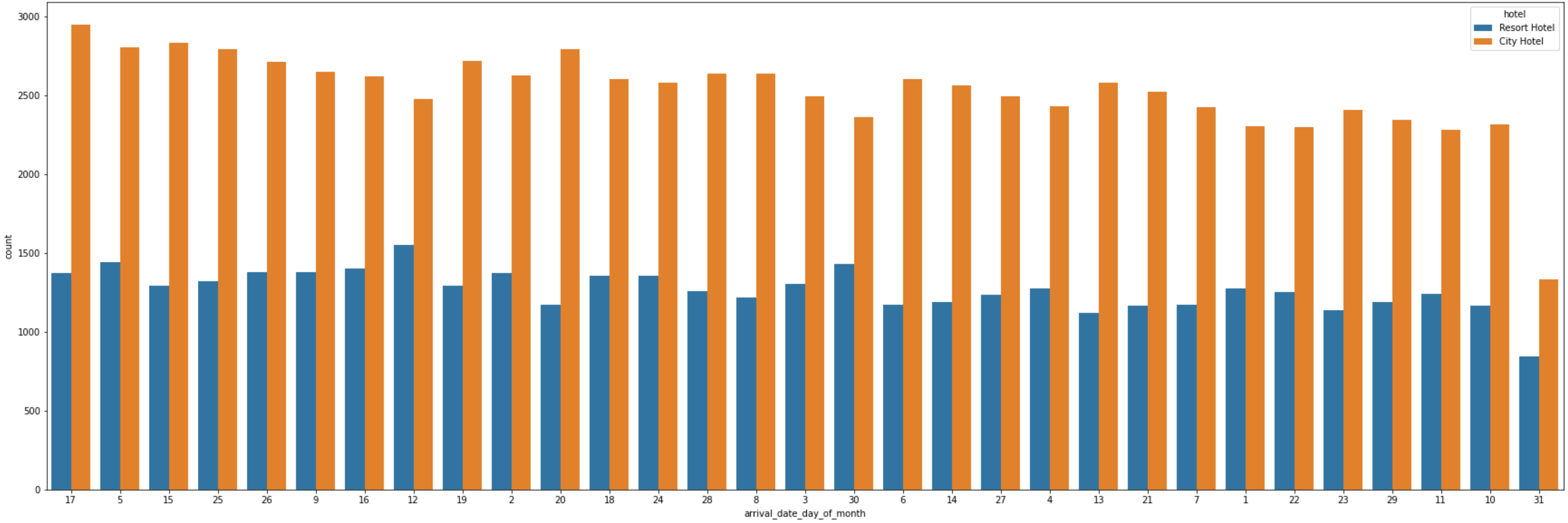
- Predicting agents:

```
df_top5_agent=df[(df['agent']==9) | (df['agent']==240)| (df['agent']==1)| (df['agent']==14)| (df['agent']==7)]
nan_agent=df[df['agent'].isna()]
'''--------------------------------------------------------------------------------------------------------predicting agents-
X_train=df_top5_agent[df_top5_agent.describe().columns].drop(['agent'],axis=1)
y_train=df_top5_agent['agent']
X_test= nan_agent[nan_agent.describe().columns].drop(['agent'],axis=1)
y_pred=RandomForest_model(X_train,y_train,X_test)
nan_agent['agent']=y_pred
print(nan_agent.agent.value_counts())
index_list=[]
for i in nan_agent.index:
  index_list.append(i)
df.loc[index_list]=nan_agent.loc[index_list]
```

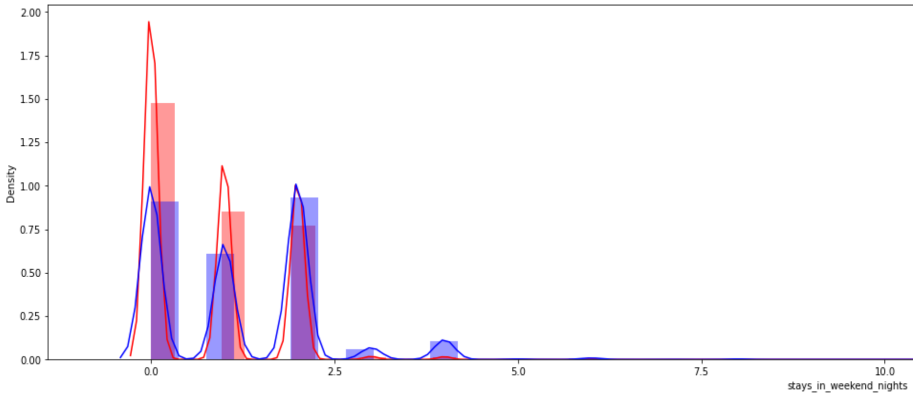
Correlation Chart

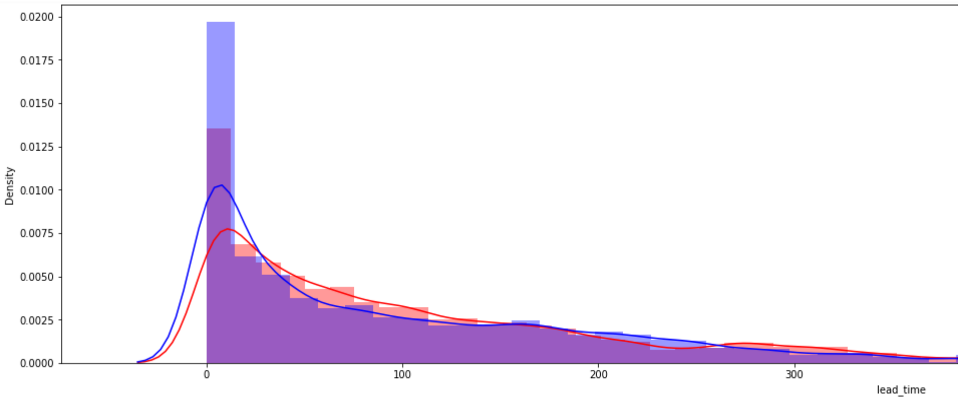


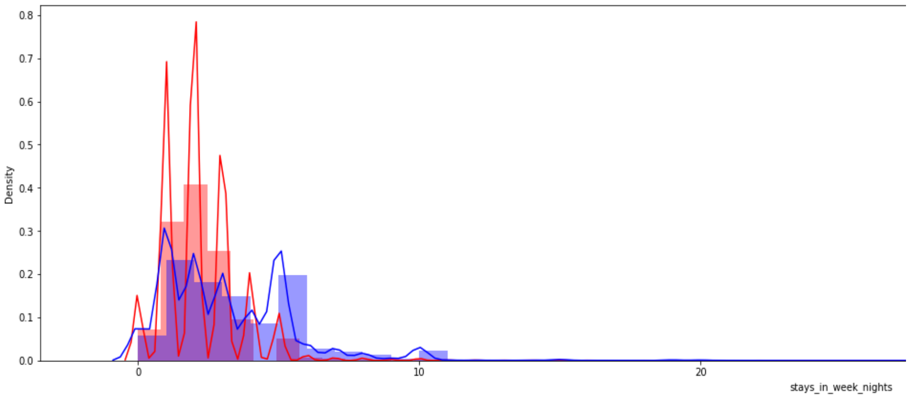
Count Plots for every feature Keeping Hotel as Hue

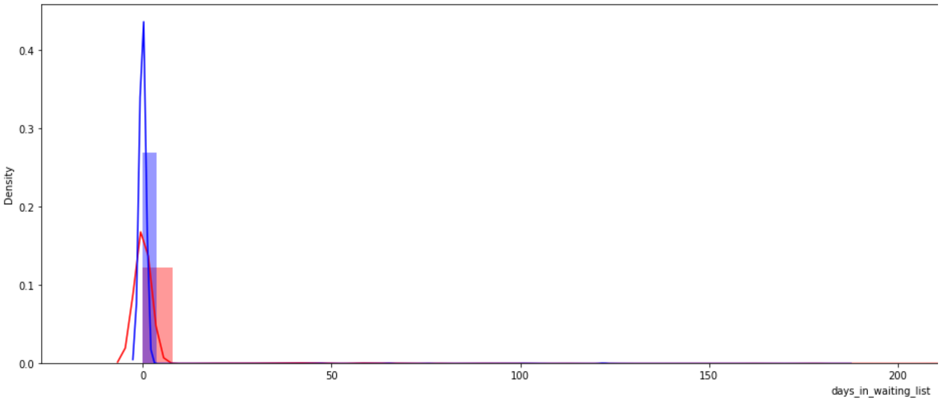


Distribution Plots

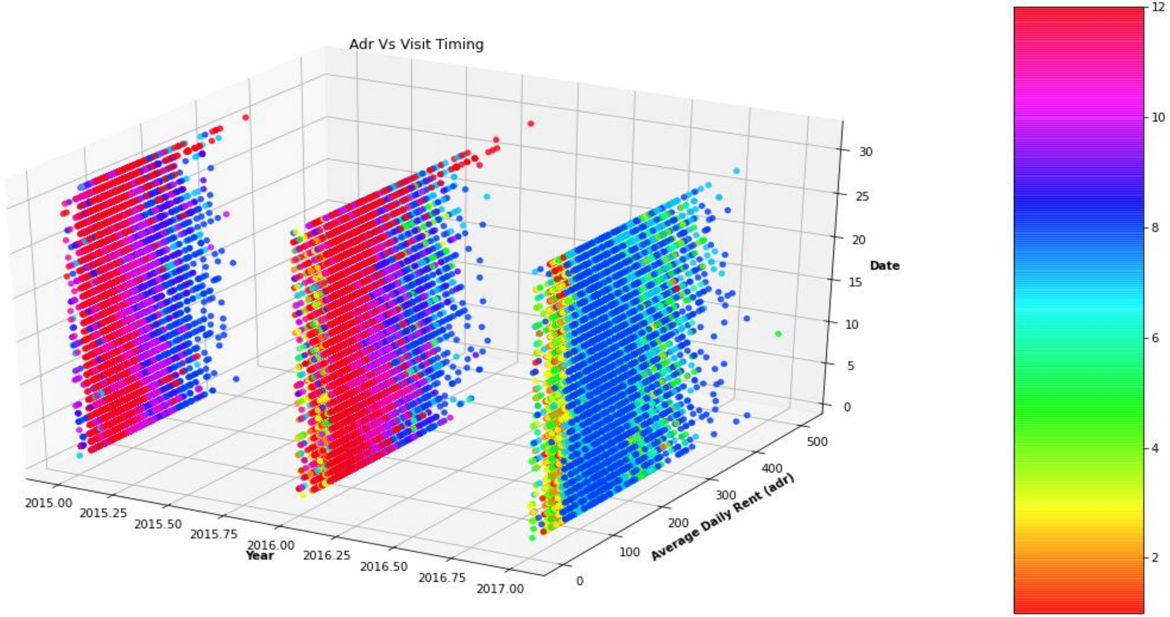




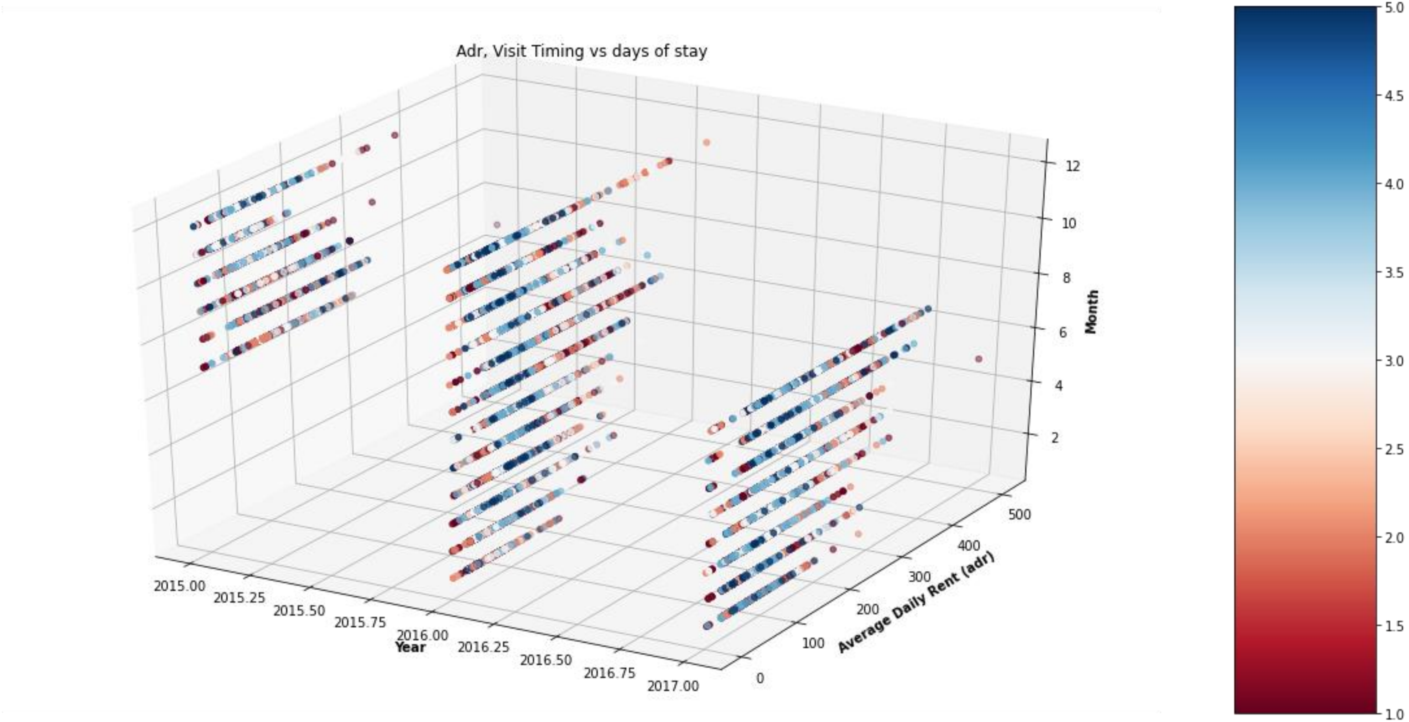




3D Scatter Plot

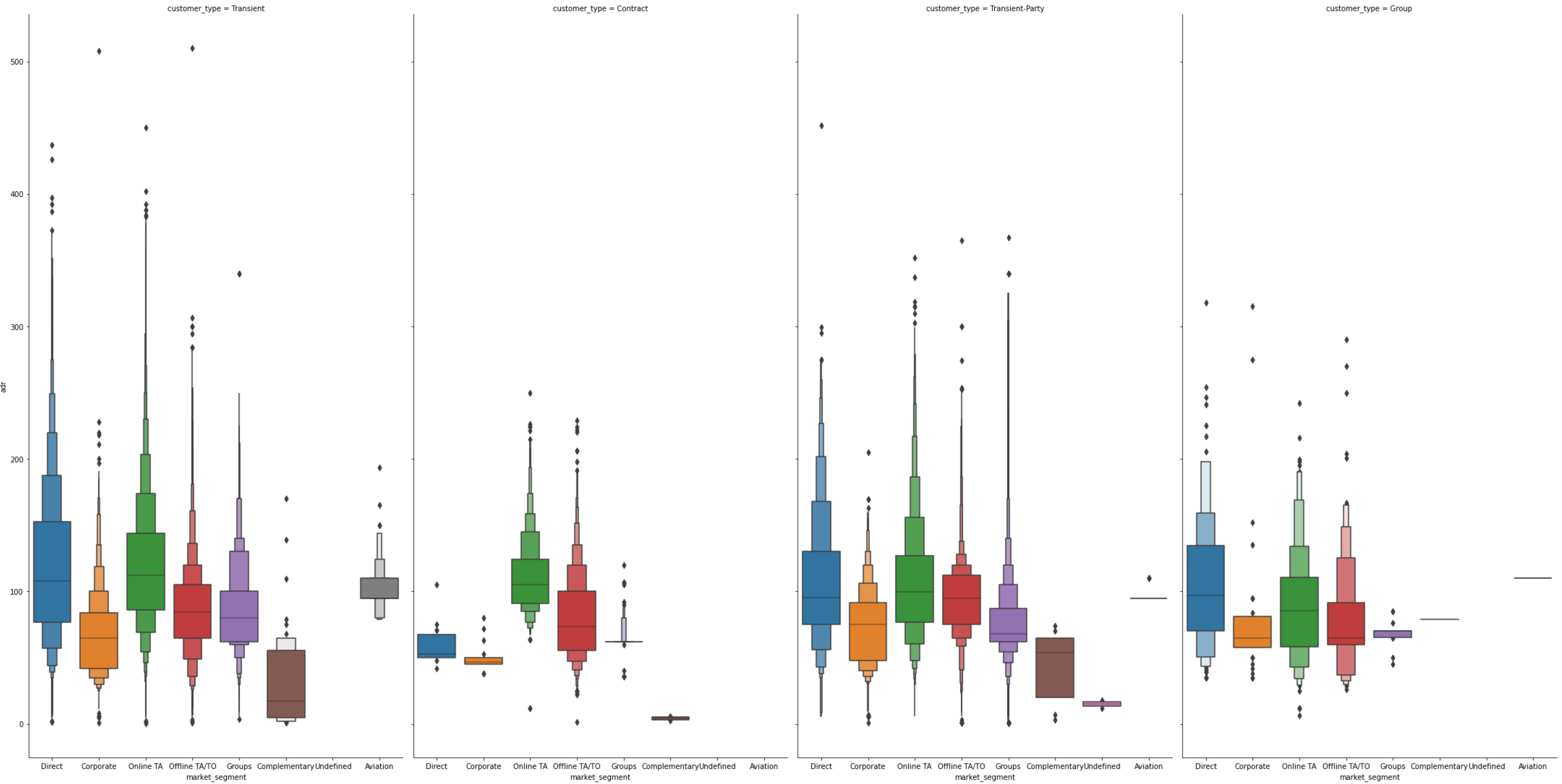


Average Daily rent does not depend on Date, but certainly Months plays a Major role as we can see that rent is less for the month of Dec and Jan Possibly due to off season, while july and august have the highest adr, also rent of all the hotels are increasing rapidly this can be concluded by seeing the width of the adr plane



- Mid priced hotel have longer staying customers, also people stay longer in the months of July and August, so to get the best daily rate one should book the hotel for at least 5 days.

Boxen Plot



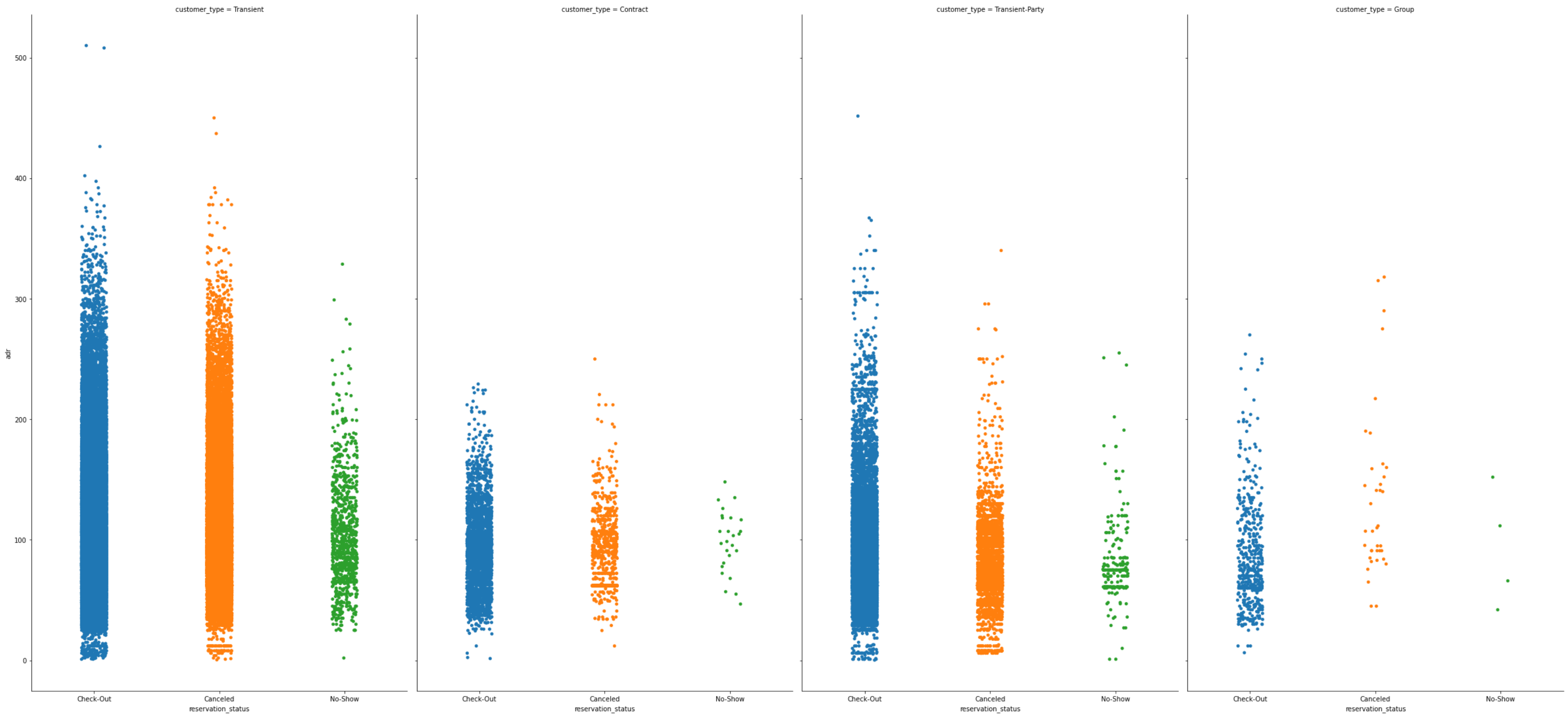
customer_type = Transient
customer_type = Contract
customer_type = Transient-Party
customer_type = Group
adr
500
400
300
200
100
0
Direct
Corporate
Online TA
Offline TA/TO
Groups
Complementary
Undefined
Aviation
market_segment

Analyzing the plot we can say that :

- For Transient, Transient-Party and Group customers Online TA/To and direct market Segments Stay at hotels with high Adr followed by Aviation, offine TA/To and then Corporate

- where else in contract type Online and offline Ta Dominated

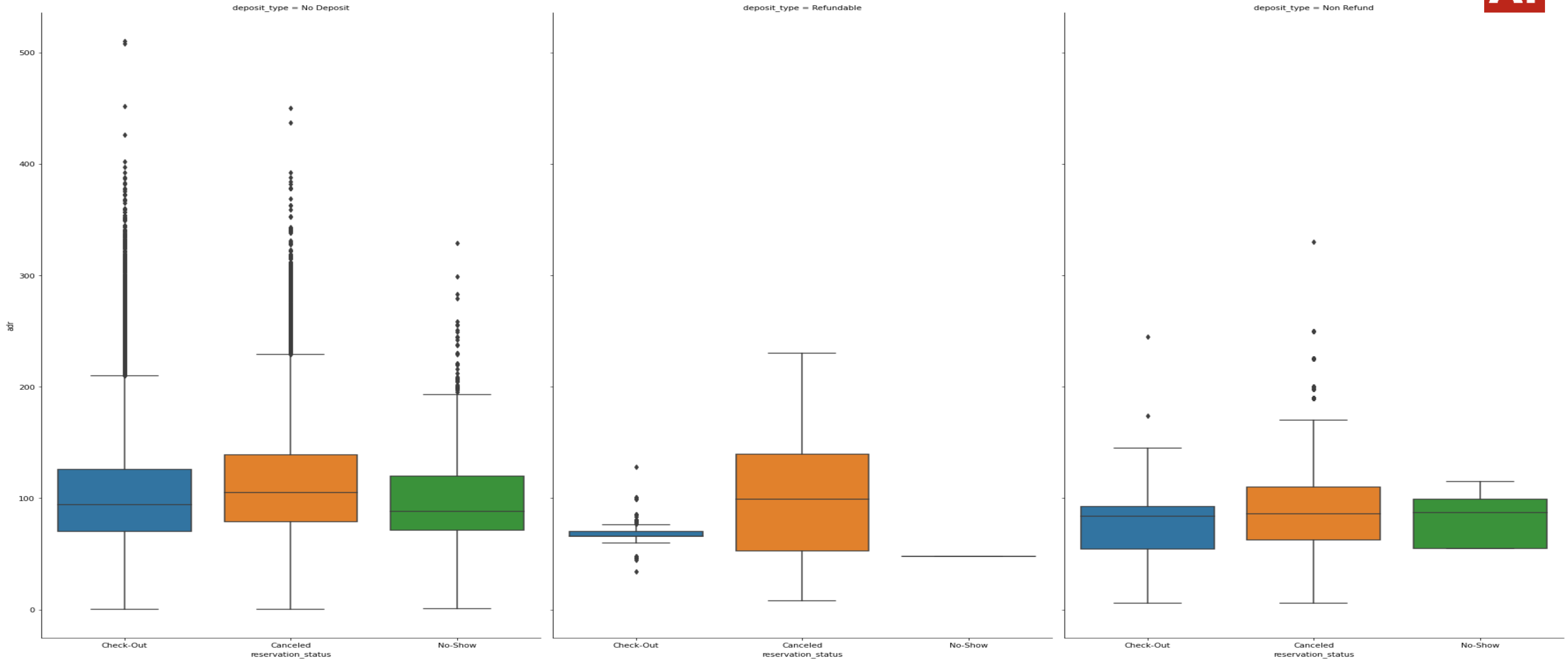
Scatter Category Plot



Analyzing the plot we can say that :

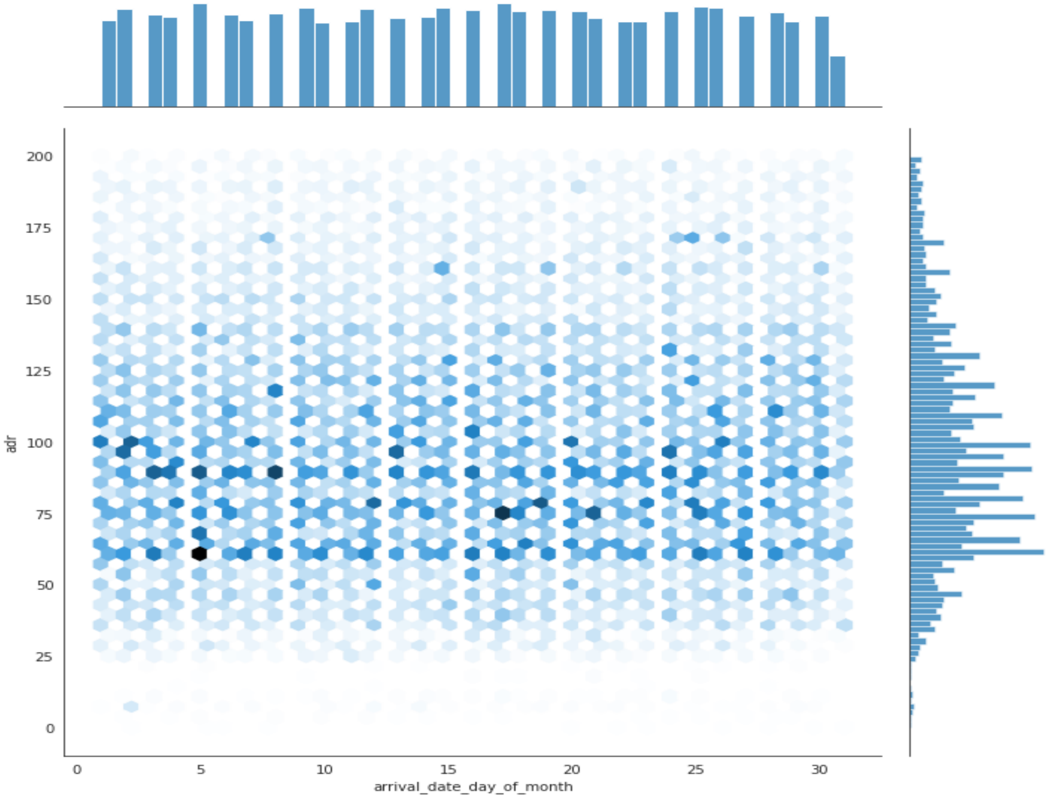
- Transient Customers cancels their booking most of the times, also they are the ones who does not show even after making bookings, where else groups are the most consistent in regards to their plans, also Transient customers books the most expensive rooms out of all segments. While the groups and contract segment pay the least Average Daily rate which actually make sense.

Box Plot



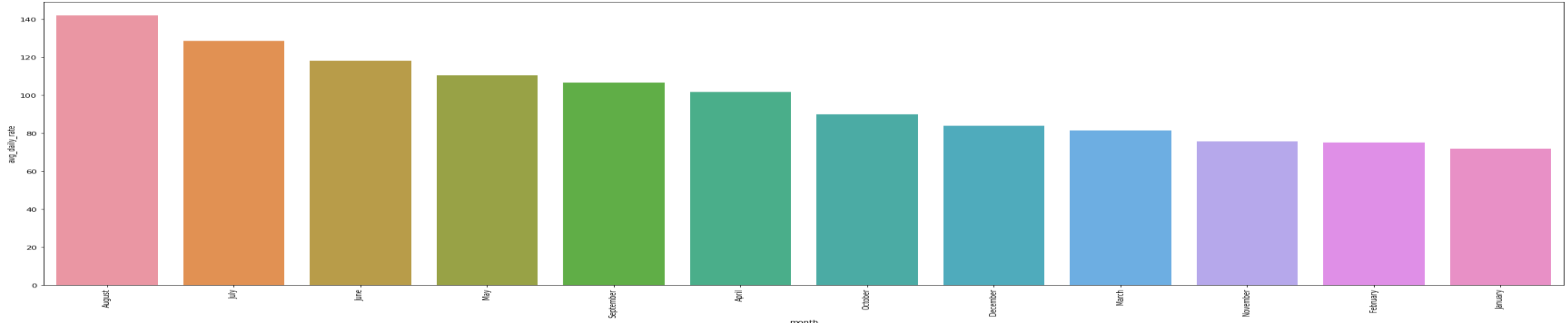
This plot actually tells a lot about customer behaviour, the customers who make non refundable booking are the least,so is their no show ratio, wherelse most people make no deposits while bookings and thus are most inconsistent as well

Hex Type joint plot

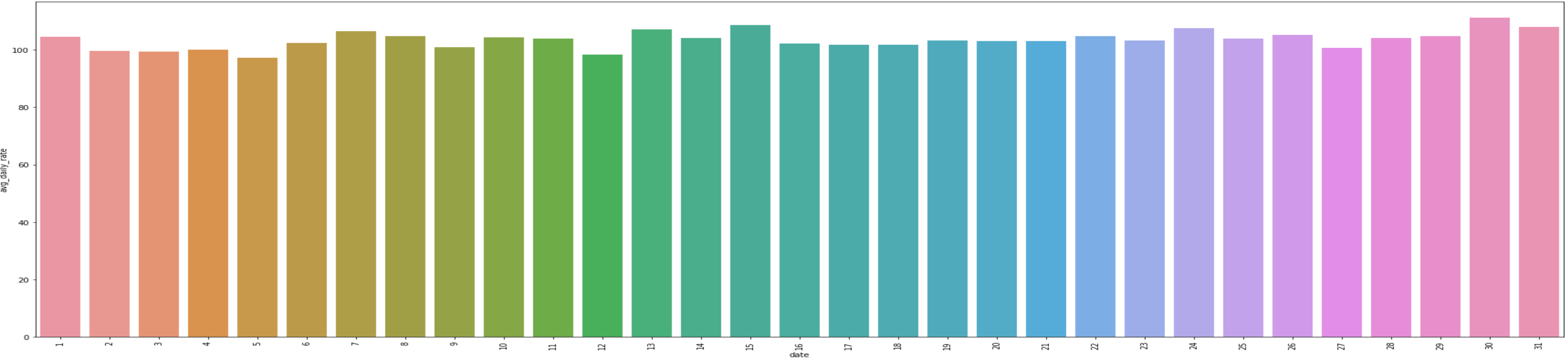


Foot Fall for customers is almost the same for all the 30 days of the month, also the maximum bookings are made with average Daily rate of 50-100 Pounds.

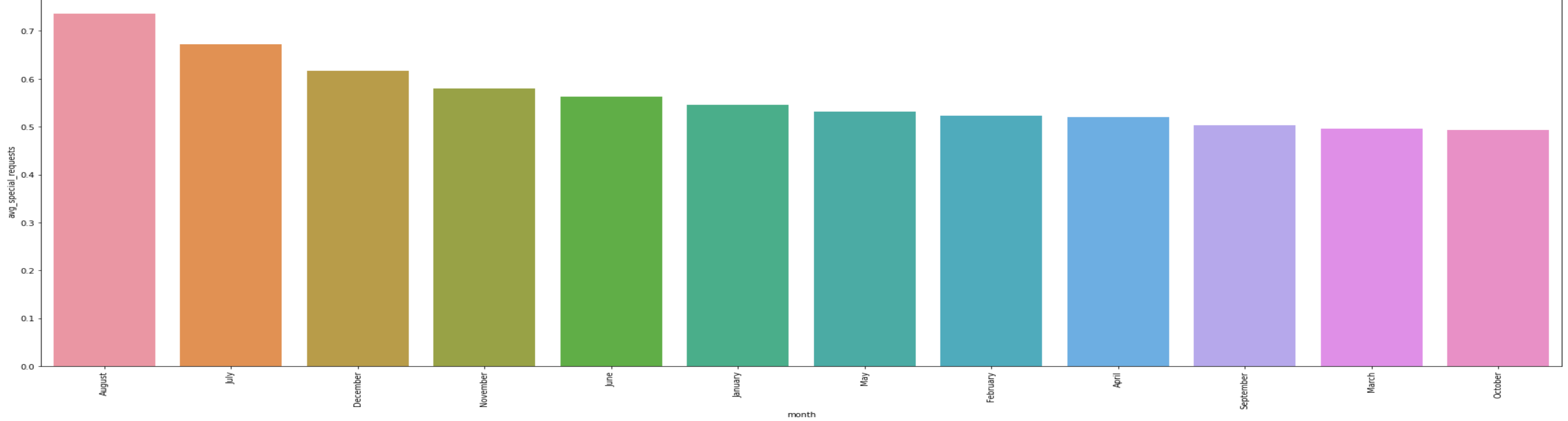
Bar Charts Considering the Avg Values



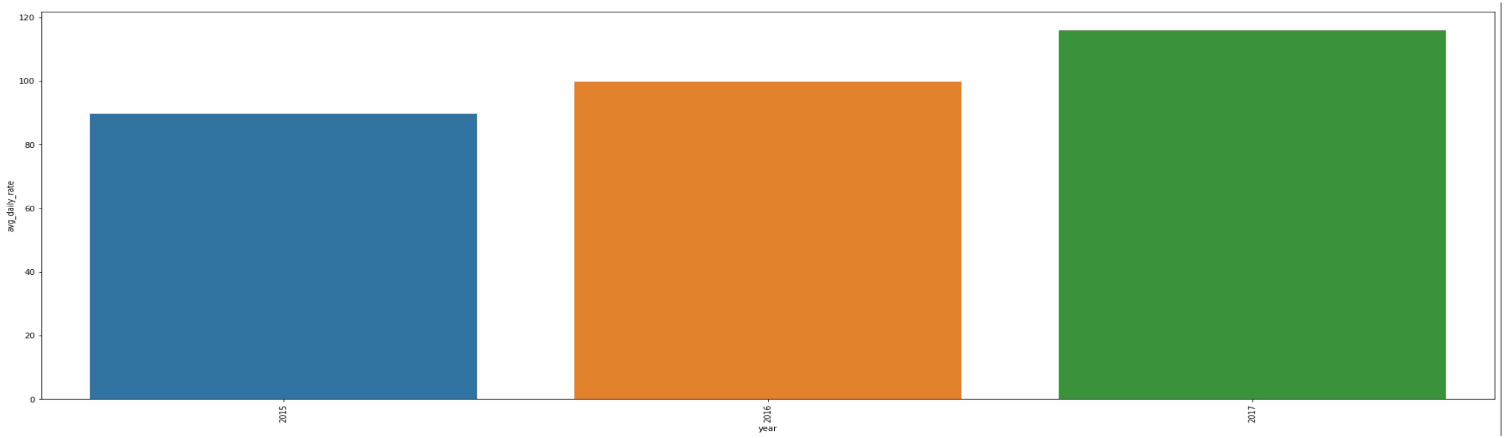
Rooms are most Expensive in the Month of July and August and cheapest in Feb and Jan



Again Price is almost the same for all Days in a Month



Most Request Come at the Peak time August followed by July



Average Daily Price are Increasing Rapidly Each Year and have hiked almost 30% from 2015-2017

Let's customize the Data for our Indian Public who want to get hotel room booked in this area, and tell them which agent they should contact.

- Our priority is to get hotel which assign the room which we have reserved while booking
- Hotel should have hosted some Indian guest previously as well
- Avg_daily_price should be less than 100 Pound
- No intial Deposit is to be made

	arrival_date_month	arrival_date_month_count	agent	agent_count	hotel	hotel_count	country	country_count	is_canceled	is_canceled_count	lead_time	lead_time_count	arrival_date_year	arrival_date_year_count	arrival_date_week_number	arrival_date_week_num
0	November	17	9.000000	18	City Hotel	44.000000	IND	59.000000	0.000000	52.000000	98	7	2016.000000	32.000000	48	6
1	April	12	7.000000	9	Resort Hotel	15.000000			1.000000	7.000000	1	3	2017.000000	16.000000	45	6
2	March	6	240.000000	7							76	3	2015.000000	11.000000	15	5
3	February	6	37.000000	6							73	3			46	4
4	January	5	241.000000	5							31	2			40	4
5	December	3	14.000000	3							34	2			11	4
6	October	3	83.000000	3							25	2			18	3
7	September	3	115.000000	1							56	2			3	3
8	May	2	250.000000	1							17	2			7	3
9	June	1	242.000000	1							47	2			9	2

From the Data it can be said that agent 9 should be the go to choice for Indian with perfect time to visit being either November as the prices are low, Requests are highly entertained and you can see some native people around you as well.

Calculating Footfall for Sep month using basic math's:

```
df_sept=df[df['arrival_date_month']=='September']
```

```
foot_fall_df=pd.DataFrame(df[df['arrival_date_year']!= 2015].groupby('arrival_date_month')['arrival_date_year'].value_counts())
```

```
months=['January','February','March','April','May','June','July','August']
growth_rate_list=[]
for i in months:
  val=(foot_fall_df.loc[i].loc[2017]-foot_fall_df.loc[i].loc[2016])/100
  growth_rate_list.append(val)
avg_growth_rate=np.sum(growth_rate_list)/len(months)
print(f'{avg_growth_rate}%')
month=['September','October','November','December']
for i in month:
  foot_fall_sept=(foot_fall_df.loc[i].loc[2016])*(100+avg_growth_rate)/100
  print(Fore.RED+f'Expected FootFall for {i} Month = {int(foot_fall_sept)}')
```

- Avg Change in Crowd percentage compared to previous year =5.02%
- Expected Footfall for September Month = 5610
- Expected Footfall for October Month = 6412
- Expected Footfall for November Month = 4586
- Expected Footfall for December Month = 3955

THANK YOU