

initial analysis and modeling

November 10, 2021

0.1 Importing libraries

```
[2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn.linear_model import LinearRegression
import warnings
warnings.filterwarnings('ignore')
```

```
[1]: #pip install -U seaborn
```

Collecting seaborn

Using cached seaborn-0.11.2-py3-none-any.whl (292 kB)
Requirement already satisfied, skipping upgrade: pandas>=0.23 in /opt/conda/lib/python3.7/site-packages (from seaborn) (1.0.3)
Requirement already satisfied, skipping upgrade: matplotlib>=2.2 in /opt/conda/lib/python3.7/site-packages (from seaborn) (3.2.1)
Requirement already satisfied, skipping upgrade: scipy>=1.0 in /opt/conda/lib/python3.7/site-packages (from seaborn) (1.4.1)
Requirement already satisfied, skipping upgrade: numpy>=1.15 in /opt/conda/lib/python3.7/site-packages (from seaborn) (1.18.4)
Requirement already satisfied, skipping upgrade: pytz>=2017.2 in /opt/conda/lib/python3.7/site-packages (from pandas>=0.23->seaborn) (2020.1)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.6.1 in /opt/conda/lib/python3.7/site-packages (from pandas>=0.23->seaborn) (2.8.1)
Requirement already satisfied, skipping upgrade: cycler>=0.10 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied, skipping upgrade: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn) (2.4.7)
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn) (1.2.0)
Requirement already satisfied, skipping upgrade: six>=1.5 in /opt/conda/lib/python3.7/site-packages (from python-dateutil>=2.6.1->pandas>=0.23->seaborn) (1.14.0)
Installing collected packages: seaborn
Attempting uninstall: seaborn

```

Found existing installation: seaborn 0.10.1
Uninstalling seaborn-0.10.1:
  Successfully uninstalled seaborn-0.10.1
Successfully installed seaborn-0.11.2
Note: you may need to restart the kernel to use updated packages.

```

```
[2]: #from IPython.core.interactiveshell import InteractiveShell
      #InteractiveShell.ast_node_interactivity="all"
```

```
[3]: df =pd.read_csv("hotaldataClean1.csv")
      df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 87230 entries, 0 to 87229
Data columns (total 31 columns):
 #   Column                                  Non-Null Count  Dtype
---  -
 0   IsCanceled                             87230 non-null  int64
 1   LeadTime                               87230 non-null  int64
 2   ArrivalDateYear                        87230 non-null  int64
 3   ArrivalDateMonth                       87230 non-null  object
 4   ArrivalDateWeekNumber                  87230 non-null  int64
 5   ArrivalDateDayOfMonth                  87230 non-null  int64
 6   StaysInWeekendNights                   87230 non-null  int64
 7   StaysInWeekNights                      87230 non-null  int64
 8   Adults                                 87230 non-null  int64
 9   Children                               87230 non-null  float64
10   Babies                                 87230 non-null  int64
11   Meal                                   87230 non-null  object
12   Country                                87230 non-null  object
13   MarketSegment                          87230 non-null  object
14   DistributionChannel                    87230 non-null  object
15   IsRepeatedGuest                        87230 non-null  int64
16   PreviousCancellations                   87230 non-null  int64
17   PreviousBookingsNotCanceled             87230 non-null  int64
18   ReservedRoomType                       87230 non-null  object
19   AssignedRoomType                       87230 non-null  object
20   BookingChanges                          87230 non-null  int64
21   DepositType                            87230 non-null  object
22   Agent                                  87230 non-null  float64
23   DaysInWaitingList                      87230 non-null  int64
24   CustomerType                           87230 non-null  object
25   ADR                                    87230 non-null  float64
26   RequiredCarParkingSpaces                87230 non-null  int64
27   TotalOfSpecialRequests                  87230 non-null  int64
28   ReservationStatus                       87230 non-null  object
29   ReservationStatusDate                   87230 non-null  object
30   Hotal                                  87230 non-null  object

```

```
dtypes: float64(3), int64(16), object(12)
memory usage: 20.6+ MB
```

0.2 Modifying to relevant attribute types in the dataframe

```
[3]: df['ReservationStatusDate'] = pd.to_datetime(df['ReservationStatusDate'])
df["IsCanceled"] = df["IsCanceled"].astype("category")
df["ArrivalDateYear"] = df["ArrivalDateYear"].astype("category")
df["ArrivalDateMonth"] = df["ArrivalDateMonth"].astype("category")
df["Meal"] = df["Meal"].astype("category")
df["Country"] = df["Country"].astype("category")
df["MarketSegment"] = df["MarketSegment"].astype("category")
df["DistributionChannel"] = df["DistributionChannel"].astype("category")
df["IsRepeatedGuest"] = df["IsRepeatedGuest"].astype("category")
df["ReservedRoomType"] = df["ReservedRoomType"].astype("category")
df["AssignedRoomType"] = df["AssignedRoomType"].astype("category")
df["DepositType"] = df["DepositType"].astype("category")
df["Agent"] = df["Agent"].astype("category")
df["CustomerType"] = df["CustomerType"].astype("category")
df["ReservationStatus"] = df["ReservationStatus"].astype("category")
df["Hotal"] = df["Hotal"].astype("category")
```

1 Displaying Dataframe Structure

```
[4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 87230 entries, 0 to 87229
Data columns (total 31 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   IsCanceled                          87230 non-null  category
 1   LeadTime                           87230 non-null  int64
 2   ArrivalDateYear                    87230 non-null  category
 3   ArrivalDateMonth                   87230 non-null  category
 4   ArrivalDateWeekNumber              87230 non-null  int64
 5   ArrivalDateDayOfMonth              87230 non-null  int64
 6   StaysInWeekendNights               87230 non-null  int64
 7   StaysInWeekNights                 87230 non-null  int64
 8   Adults                             87230 non-null  int64
 9   Children                           87230 non-null  float64
10   Babies                             87230 non-null  int64
11   Meal                               87230 non-null  category
12   Country                            87230 non-null  category
13   MarketSegment                      87230 non-null  category
```

```

14 DistributionChannel      87230 non-null category
15 IsRepeatedGuest          87230 non-null category
16 PreviousCancellations    87230 non-null int64
17 PreviousBookingsNotCanceled 87230 non-null int64
18 ReservedRoomType         87230 non-null category
19 AssignedRoomType         87230 non-null category
20 BookingChanges           87230 non-null int64
21 DepositType              87230 non-null category
22 Agent                    87230 non-null category
23 DaysInWaitingList        87230 non-null int64
24 CustomerType             87230 non-null category
25 ADR                      87230 non-null float64
26 RequiredCarParkingSpaces 87230 non-null int64
27 TotalOfSpecialRequests   87230 non-null int64
28 ReservationStatus        87230 non-null category
29 ReservationStatusDate     87230 non-null datetime64[ns]
30 Hotal                    87230 non-null category
dtypes: category(15), datetime64[ns](1), float64(2), int64(13)
memory usage: 12.1 MB

```

2 describing the stats for numerical attributes

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

```

[5]:
count      LeadTime  ArrivalDateWeekNumber  ArrivalDateDayOfMonth  \
count      87230.000000      87230.000000      87230.000000
mean        79.971019        26.835091        15.815832
std         86.058683        13.669216         8.835545
min           0.000000         1.000000         1.000000
25%         11.000000        16.000000         8.000000
50%         49.000000        27.000000        16.000000
75%        125.000000        37.000000        23.000000
max        737.000000        53.000000        31.000000

count      StaysInWeekendNights  StaysInWeekNights  Adults  Children  \
count      87230.000000      87230.000000  87230.000000  87230.000000
mean         1.004609         2.623925         1.879365         0.138897
std         1.027408         2.039830         0.621724         0.456265
min           0.000000         0.000000         0.000000         0.000000
25%           0.000000         1.000000         2.000000         0.000000
50%           1.000000         2.000000         2.000000         0.000000
75%           2.000000         4.000000         2.000000         0.000000
max          19.000000        50.000000        55.000000        10.000000

count      Babies  PreviousCancellations  PreviousBookingsNotCanceled  \
count      87230.000000      87230.000000      87230.000000

```

| | | | |
|------|-----------|-----------|-----------|
| mean | 0.010845 | 0.030402 | 0.184054 |
| std | 0.113704 | 0.369344 | 1.733033 |
| 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 | 26.000000 | 72.000000 |

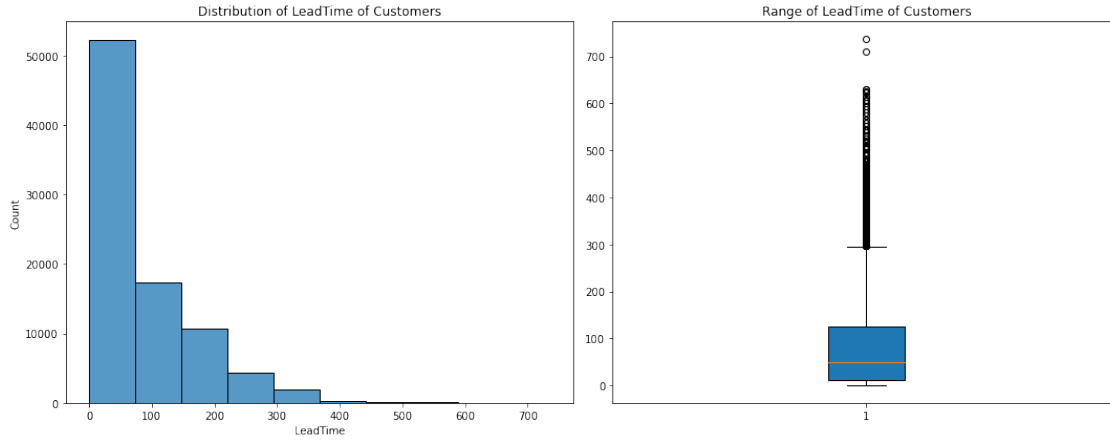
| | BookingChanges | DaysInWaitingList | ADR \ |
|-------|----------------|-------------------|--------------|
| count | 87230.000000 | 87230.000000 | 87230.000000 |
| mean | 0.268497 | 0.746291 | 106.518031 |
| std | 0.710633 | 10.001001 | 54.891227 |
| min | 0.000000 | 0.000000 | -6.380000 |
| 25% | 0.000000 | 0.000000 | 72.250000 |
| 50% | 0.000000 | 0.000000 | 98.200000 |
| 75% | 0.000000 | 0.000000 | 134.100000 |
| max | 18.000000 | 391.000000 | 5400.000000 |

| | RequiredCarParkingSpaces | TotalOfSpecialRequests |
|-------|--------------------------|------------------------|
| count | 87230.000000 | 87230.000000 |
| mean | 0.084306 | 0.698934 |
| std | 0.281659 | 0.832051 |
| min | 0.000000 | 0.000000 |
| 25% | 0.000000 | 0.000000 |
| 50% | 0.000000 | 0.000000 |
| 75% | 0.000000 | 1.000000 |
| max | 8.000000 | 5.000000 |

```
[7]: df.shape
```

```
[7]: (87230, 31)
```

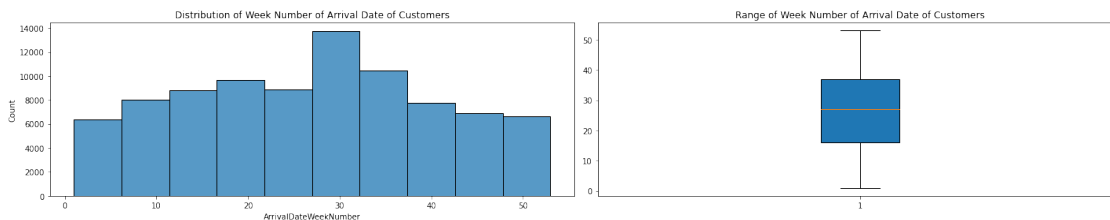
```
[6]: fig, axes = plt.subplots(1,2,figsize=(15,6))
sb.histplot(df['LeadTime'],bins=10,ax=axes[0])
plt.boxplot(df['LeadTime'],patch_artist = True)
axes[0].set_title('Distribution of LeadTime of Customers')
axes[1].set_title('Range of LeadTime of Customers')
plt.tight_layout()
plt.show()
```



```
[7]: fig, axes = plt.subplots(1,2, figsize=(20,4))

sb.histplot(df['ArrivalDateWeekNumber'],bins=10,ax=axes[0])
plt.boxplot(df['ArrivalDateWeekNumber'],patch_artist = True)

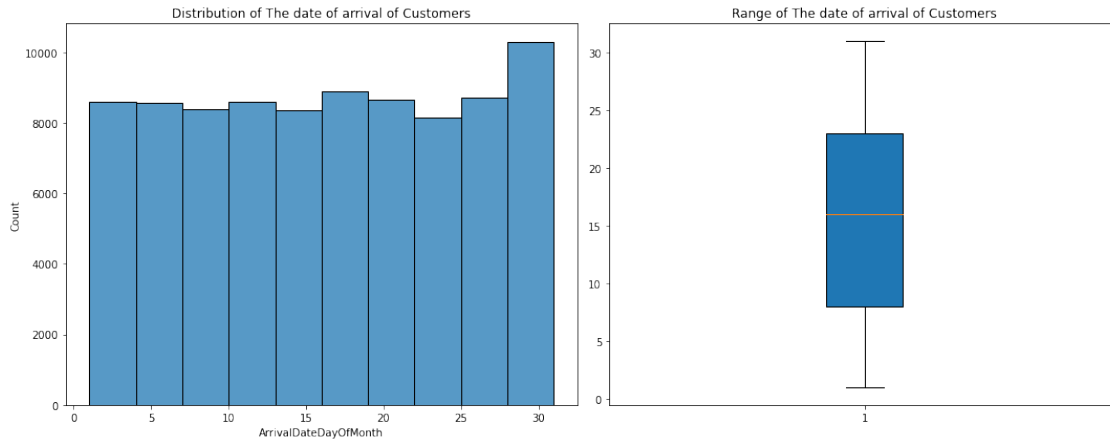
axes[0].set_title('Distribution of Week Number of Arrival Date of Customers')
axes[1].set_title('Range of Week Number of Arrival Date of Customers')
plt.tight_layout()
#plt.savefig("hist of ArrivalDateWeekNumber.png" )
plt.show()
```



```
[8]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['ArrivalDateDayOfMonth'],bins=10,ax=axes[0])
plt.boxplot(df['ArrivalDateDayOfMonth'],patch_artist = True)

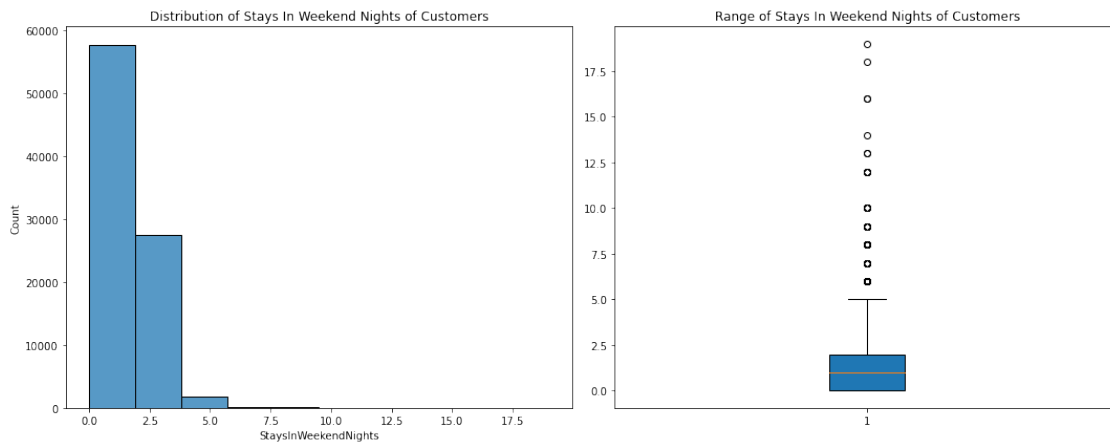
axes[0].set_title('Distribution of The date of arrival of Customers')
axes[1].set_title('Range of The date of arrival of Customers')
plt.tight_layout()
#plt.savefig("hist of ArrivalDateDayOfMonth.png" )
plt.show()
```



```
[9]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['StaysInWeekendNights'],bins=10,ax=axes[0])
plt.boxplot(df['StaysInWeekendNights'],patch_artist=True)

axes[0].set_title('Distribution of Stays In Weekend Nights of Customers')
axes[1].set_title('Range of Stays In Weekend Nights of Customers')
plt.tight_layout()
#plt.savefig("hist of StaysInWeekendNights.png" )
plt.show()
```



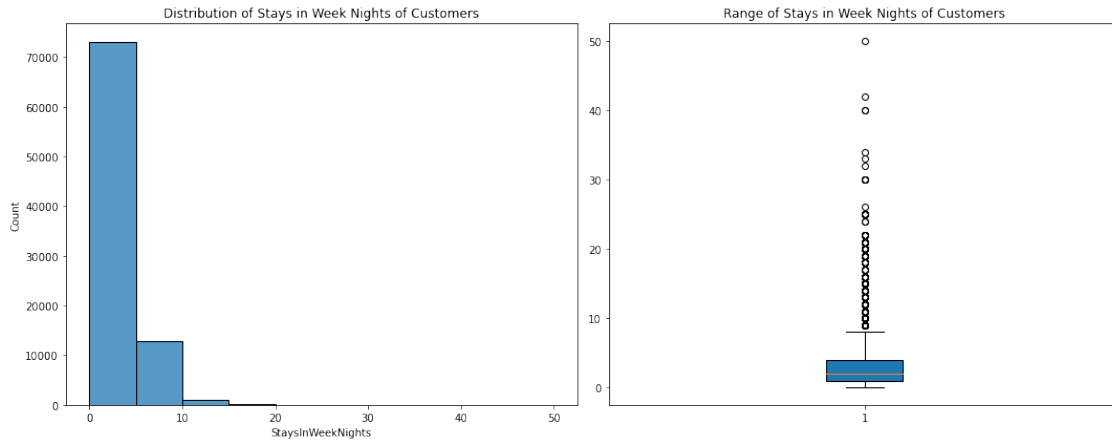
```
[10]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['StaysInWeekNights'],bins= 10,ax=axes[0])
plt.boxplot(df['StaysInWeekNights'],patch_artist=True)
axes[0].set_title('Distribution of Stays in Week Nights of Customers')
```

```

axes[1].set_title('Range of Stays in Week Nights of Customers')
plt.tight_layout()
#plt.savefig("hist of StaysInWeekNights.png" )
plt.show()

```



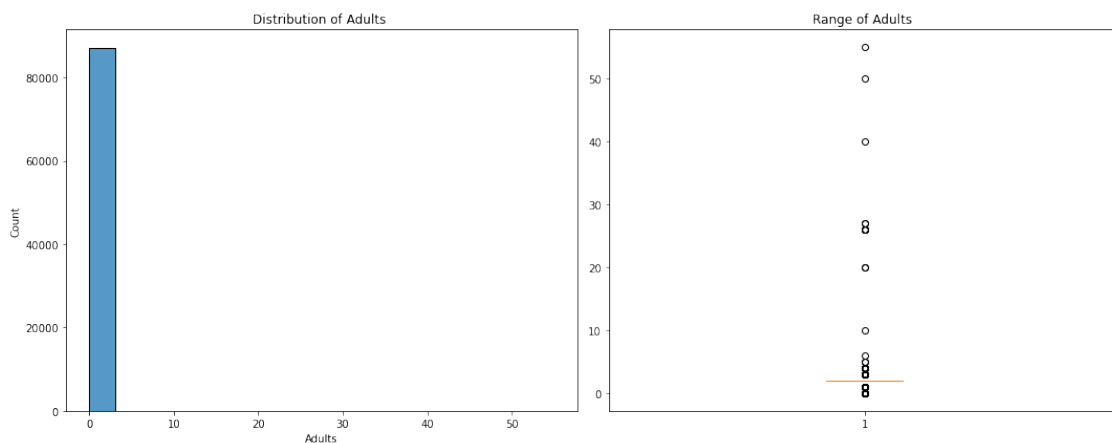
```

[11]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['Adults'],ax=axes[0])

plt.boxplot(df['Adults'],patch_artist=True)
axes[0].set_title('Distribution of Adults ')
axes[1].set_title('Range of Adults ')
plt.tight_layout()
#plt.savefig("hist of Adults.png" )
plt.show()

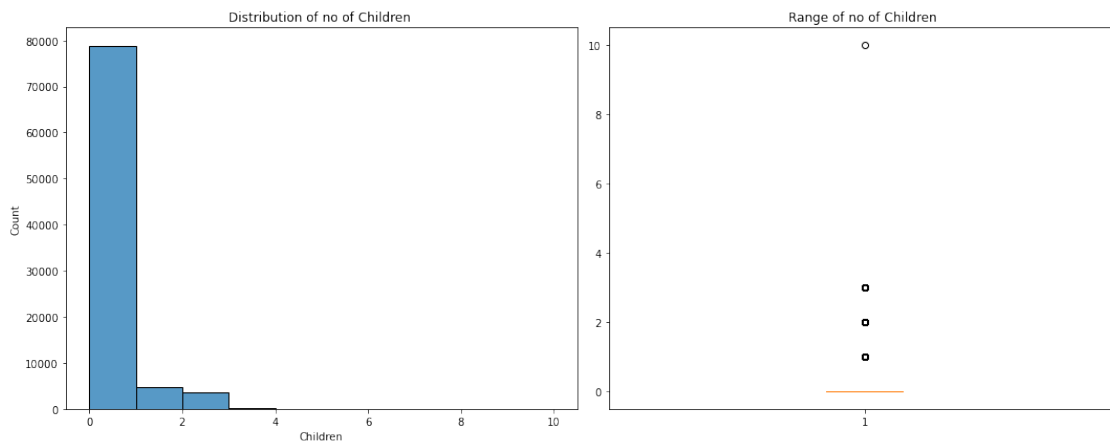
```




```
[12]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['Children'],bins= 10,ax=axes[0])

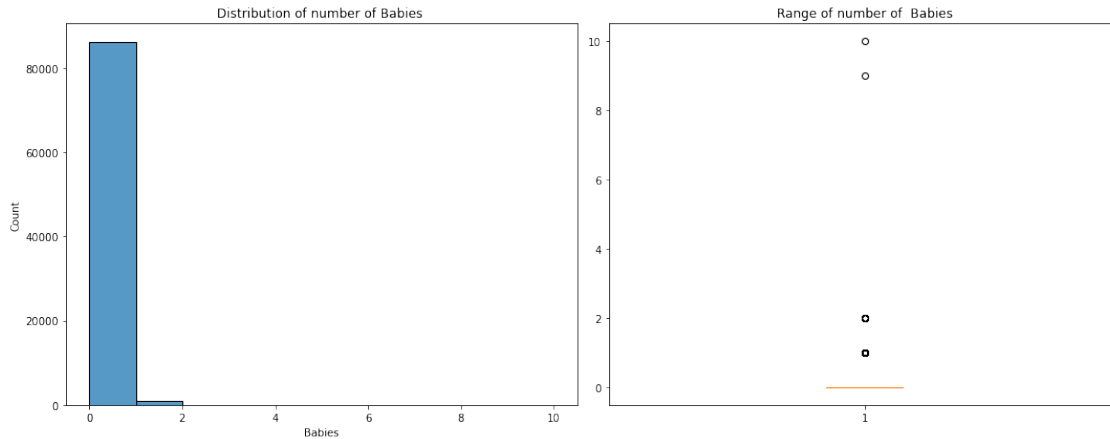
plt.boxplot(df['Children'],patch_artist=True)
axes[0].set_title('Distribution of no of Children ')
axes[1].set_title('Range of no of Children ')
plt.tight_layout()
#plt.savefig("hist of Children.png" )
plt.show()
```



```
[13]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['Babies'],bins= 10,ax=axes[0])

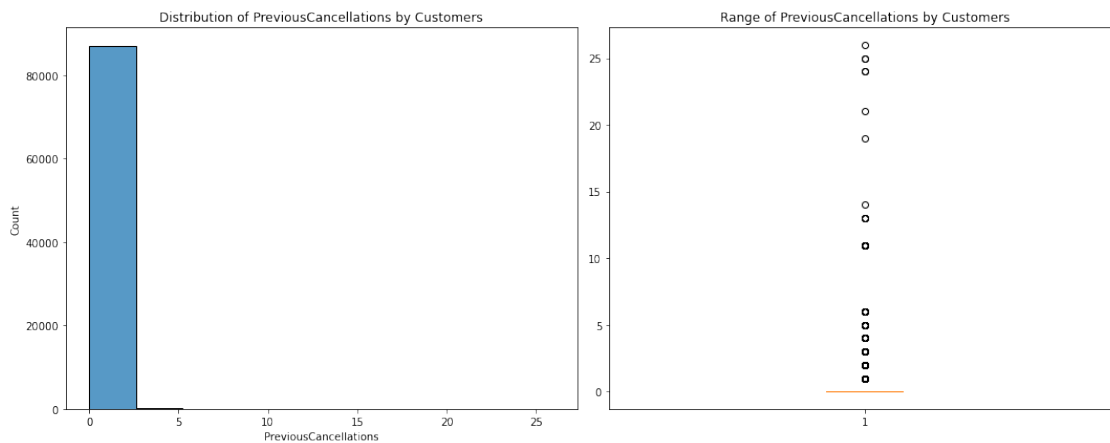
plt.boxplot(df['Babies'],patch_artist=True)## Whiskers in the Boxplot shows
↳ that there are outliers
axes[0].set_title('Distribution of number of Babies ')
axes[1].set_title('Range of number of Babies')
plt.tight_layout()
#plt.savefig("hist of Babies.png" )
plt.show()
```



```
[14]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['PreviousCancellations'],bins= 10,ax=axes[0])

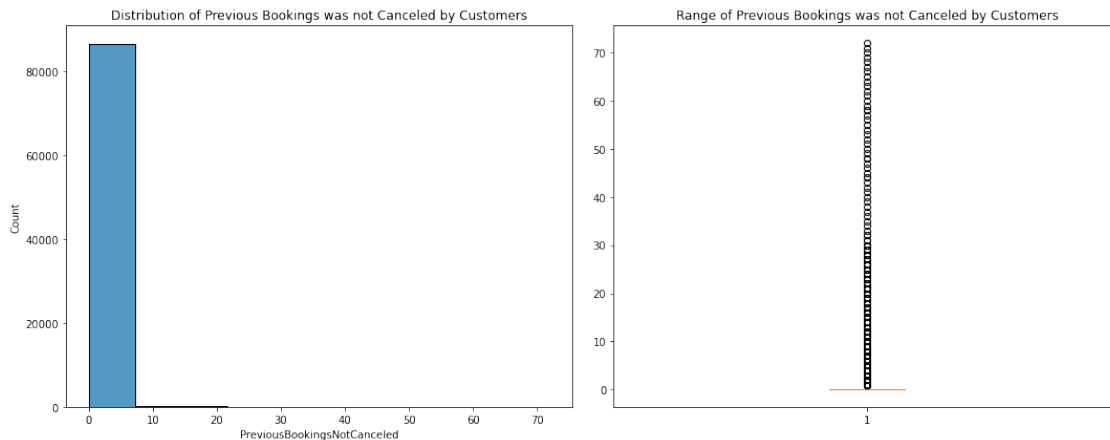
plt.boxplot(df['PreviousCancellations'],patch_artist=True)
axes[0].set_title('Distribution of PreviousCancellations by Customers')
axes[1].set_title('Range of PreviousCancellations by Customers')
plt.tight_layout()
#plt.savefig("hist of PreviousCancellations.png" )
plt.show()
```



```
[15]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['PreviousBookingsNotCanceled'],bins= 10,ax=axes[0])
```

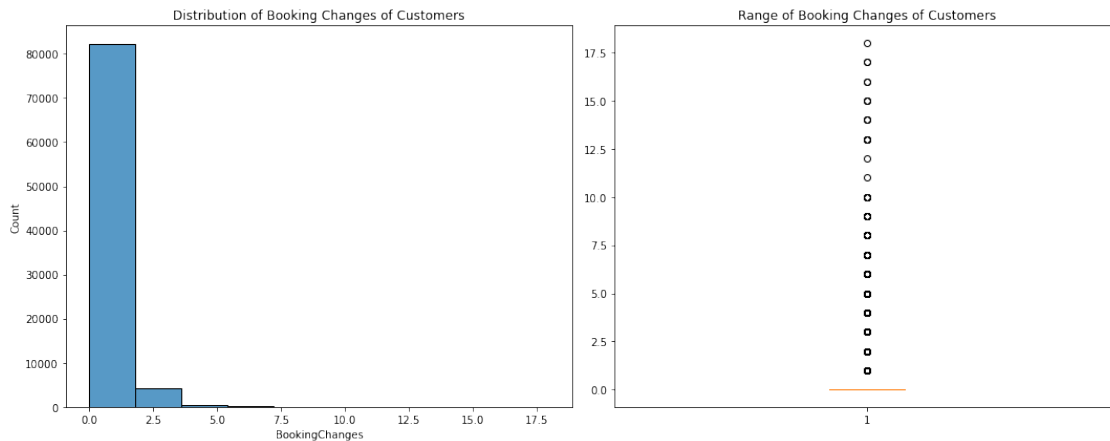
```
plt.boxplot(df['PreviousBookingsNotCanceled'],patch_artist=True)## Whiskers in
↳the Boxplot shows that there are outliers
axes[0].set_title('Distribution of Previous Bookings was not Canceled by
↳Customers')
axes[1].set_title('Range of Previous Bookings was not Canceled by Customers')
plt.tight_layout()
#plt.savefig("hist of PreviousBookingsNotCanceled.png" )
plt.show()
```



```
[16]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['BookingChanges'],bins= 10,ax=axes[0])

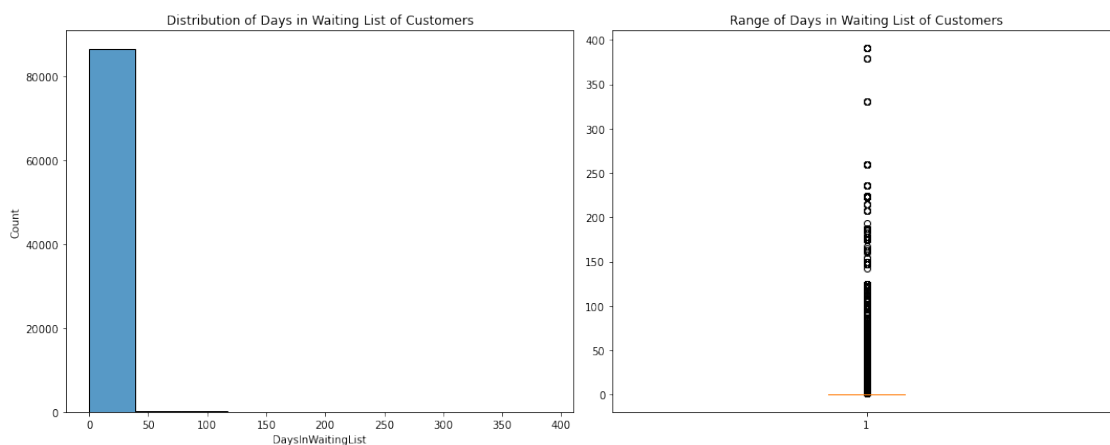
plt.boxplot(df['BookingChanges'],patch_artist=True)## Whiskers in the Boxplot
↳shows that there are outliers
axes[0].set_title('Distribution of Booking Changes of Customers')
axes[1].set_title('Range of Booking Changes of Customers')
plt.tight_layout()
#plt.savefig("hist of BookingChanges.png" )
plt.show()
```



```
[17]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['DaysInWaitingList'],bins=10,ax=axes[0])

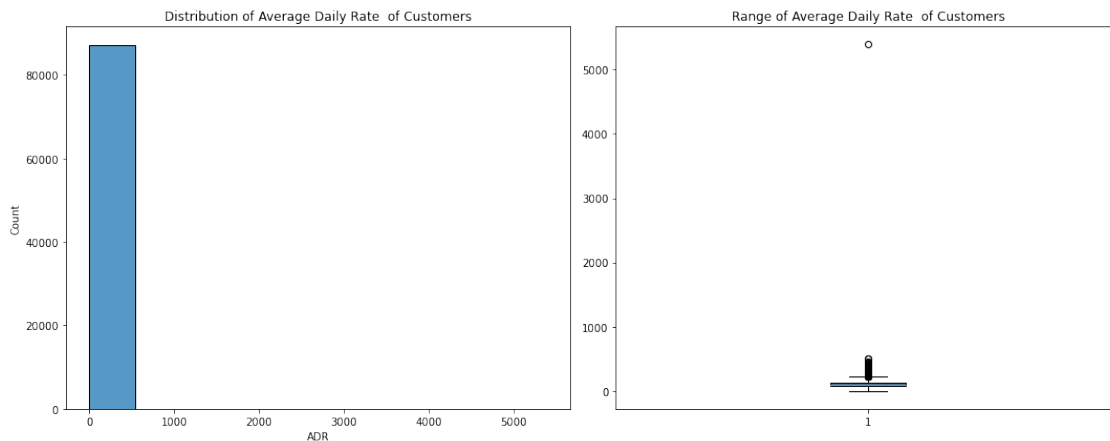
plt.boxplot(df['DaysInWaitingList'],patch_artist=True)## Whiskers in the
↳Boxplot shows that there are outliers
axes[0].set_title('Distribution of Days in Waiting List of Customers')
axes[1].set_title('Range of Days in Waiting List of Customers')
plt.tight_layout()
#plt.savefig("hist of DaysInWaitingList.png" )
plt.show()
```



```
[18]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['ADR'],bins=10,ax=axes[0])
```

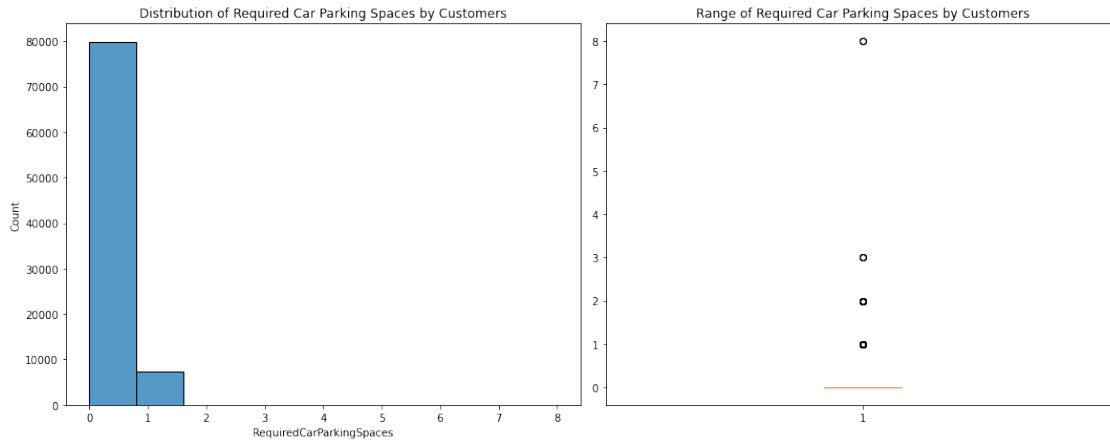
```
plt.boxplot(df['ADR'],patch_artist=True)
axes[0].set_title('Distribution of Average Daily Rate of Customers')
axes[1].set_title('Range of Average Daily Rate of Customers')
plt.tight_layout()
#plt.savefig("hist of ADR.png" )
plt.show()
```



```
[19]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['RequiredCarParkingSpaces'],bins= 10, ax=axes[0])

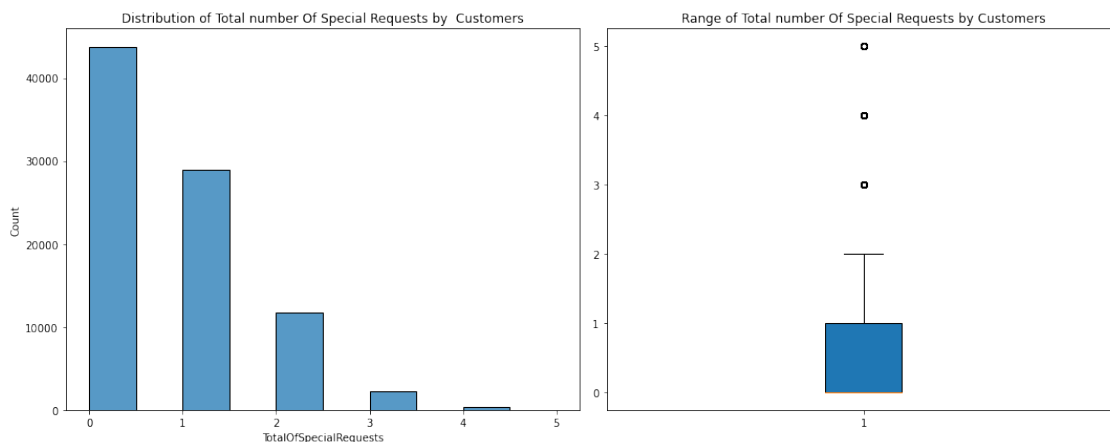
plt.boxplot(df['RequiredCarParkingSpaces'],patch_artist=True)## Whiskers in the
↳Boxplot shows that there are outliers
axes[0].set_title('Distribution of Required Car Parking Spaces by Customers')
axes[1].set_title('Range of Required Car Parking Spaces by Customers')
plt.tight_layout()
#plt.savefig("hist of RequiredCarParkingSpaces.png" )
plt.show()
```



```
[20]: fig, axes = plt.subplots(1,2, figsize=(15,6))

sb.histplot(df['TotalOfSpecialRequests'],bins= 10,ax=axes[0])

plt.boxplot(df['TotalOfSpecialRequests'],patch_artist=True)## Whiskers in the
↳Boxplot shows that there are outliers
axes[0].set_title('Distribution of Total number Of Special Requests by 
↳Customers')
axes[1].set_title('Range of Total number Of Special Requests by Customers')
plt.tight_layout()
#plt.savefig("hist of TotalOfSpecialRequests.png" )
plt.show()
```



```
[21]: numeric = []
category = []
```

```

for col in df:
    if pd.api.types.is_numeric_dtype(df[col]):
        numeric.append(col)
    else:
        category.append(col)
print("category :",category)

```

```

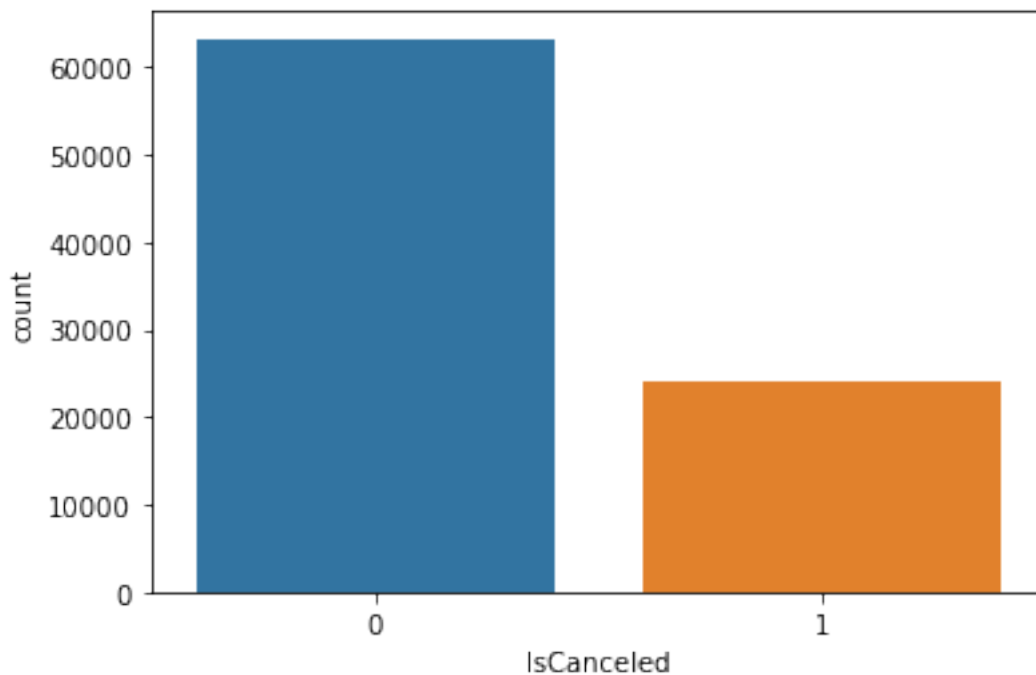
category : ['IsCanceled', 'ArrivalDateYear', 'ArrivalDateMonth', 'Meal',
'Country', 'MarketSegment', 'DistributionChannel', 'IsRepeatedGuest',
'ReservedRoomType', 'AssignedRoomType', 'DepositType', 'Agent', 'CustomerType',
'ReservationStatus', 'ReservationStatusDate', 'Hotal']

```

```

[23]: chart = sb.countplot(df["IsCanceled"]) # frequency distribution
chart.set_xticklabels(chart.get_xticklabels())
#plt.savefig("hist of IsCanceled.png" )
plt.show()

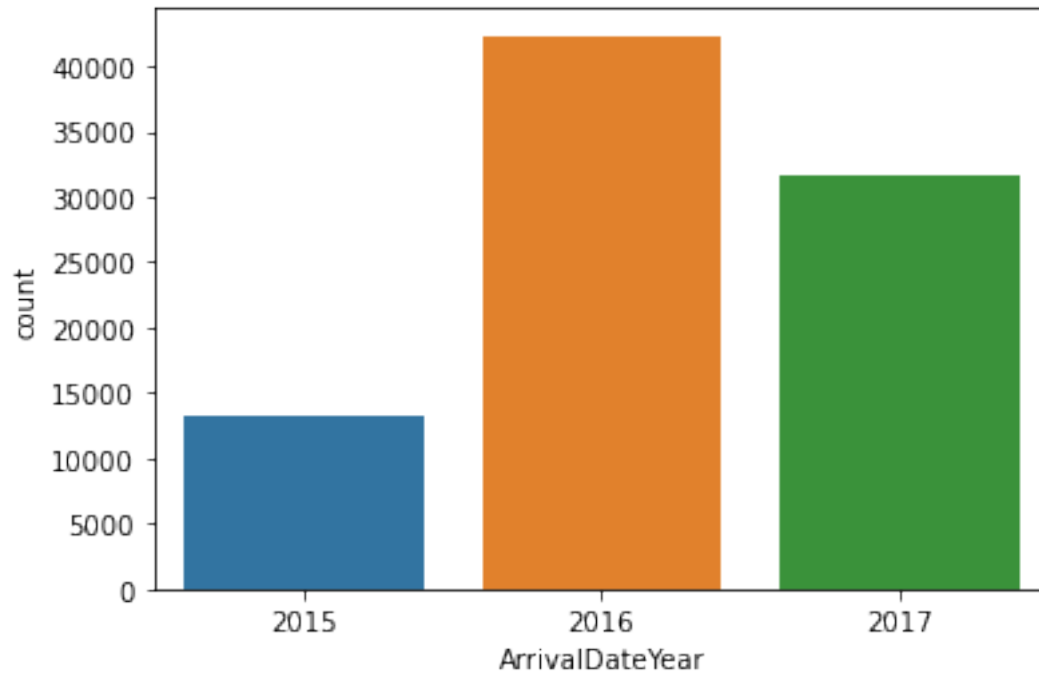
```



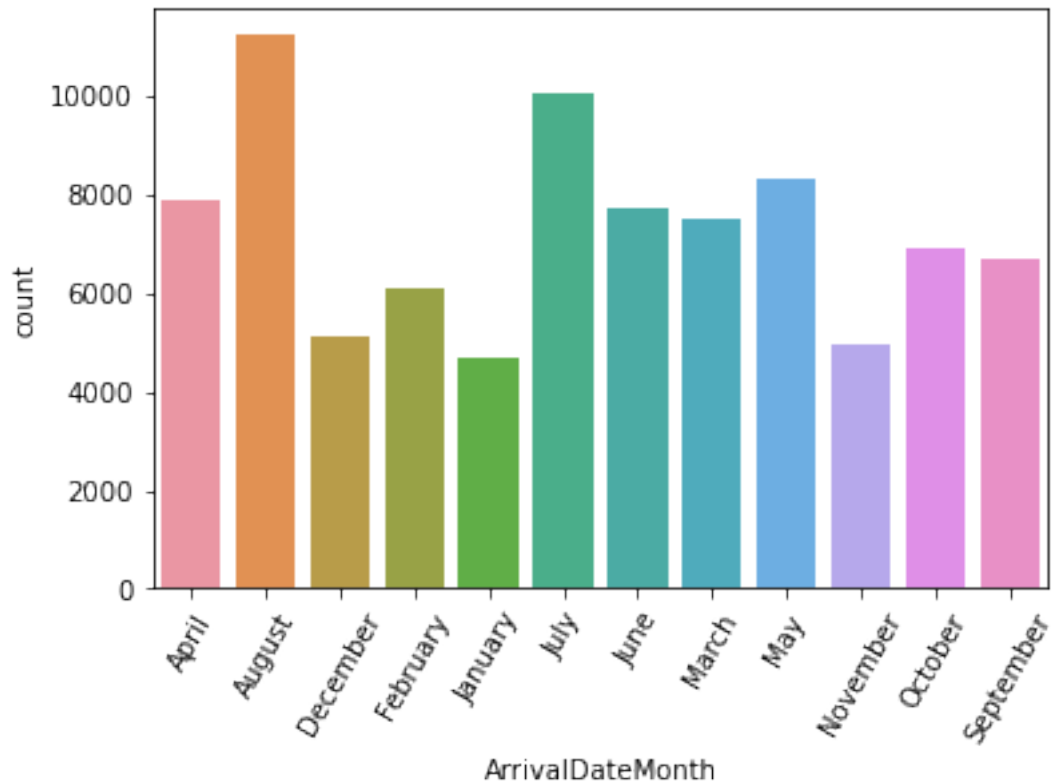
```

[24]: chart = sb.countplot(df["ArrivalDateYear"]) # frequency distribution
chart.set_xticklabels(chart.get_xticklabels())
#plt.savefig("hist of ArrivalDateYear.png" )
plt.show()

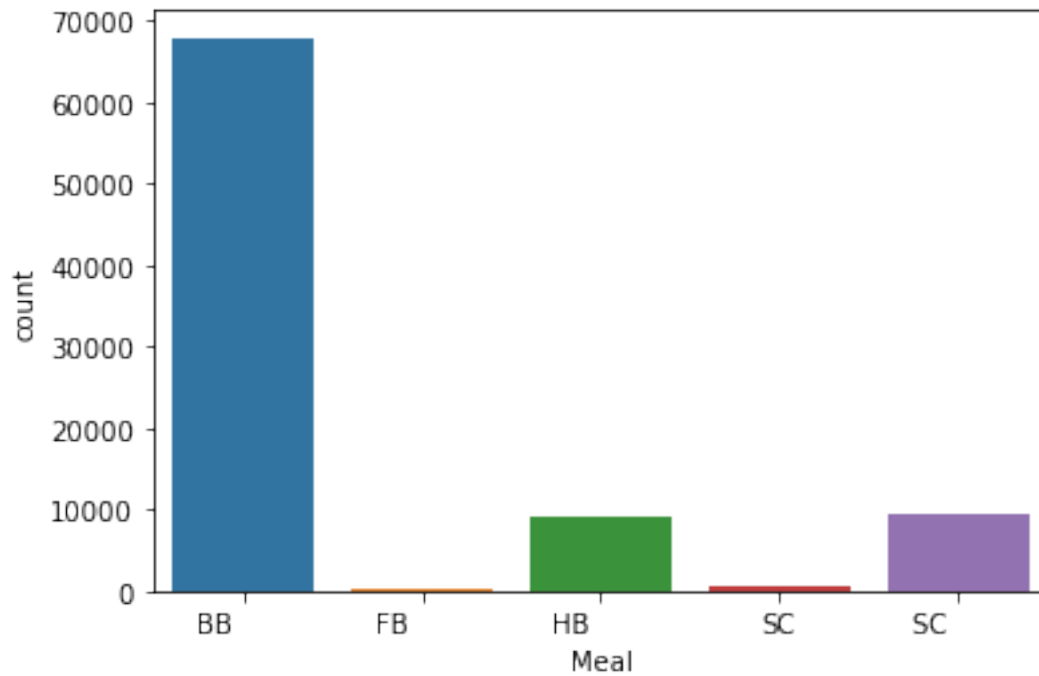
```



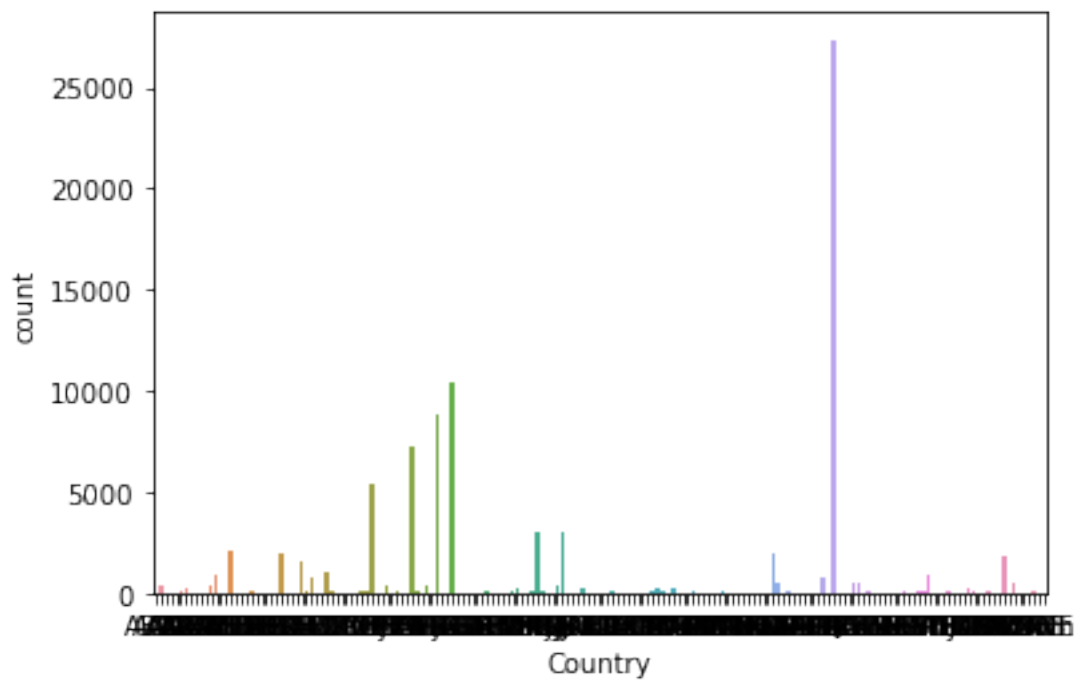
```
[25]: chart = sb.countplot(df["ArrivalDateMonth"]) # frequency distribution
chart.set_xticklabels(chart.get_xticklabels(),rotation=60)
#plt.savefig("A.png",bbox_inches='tight' )
plt.show()
```

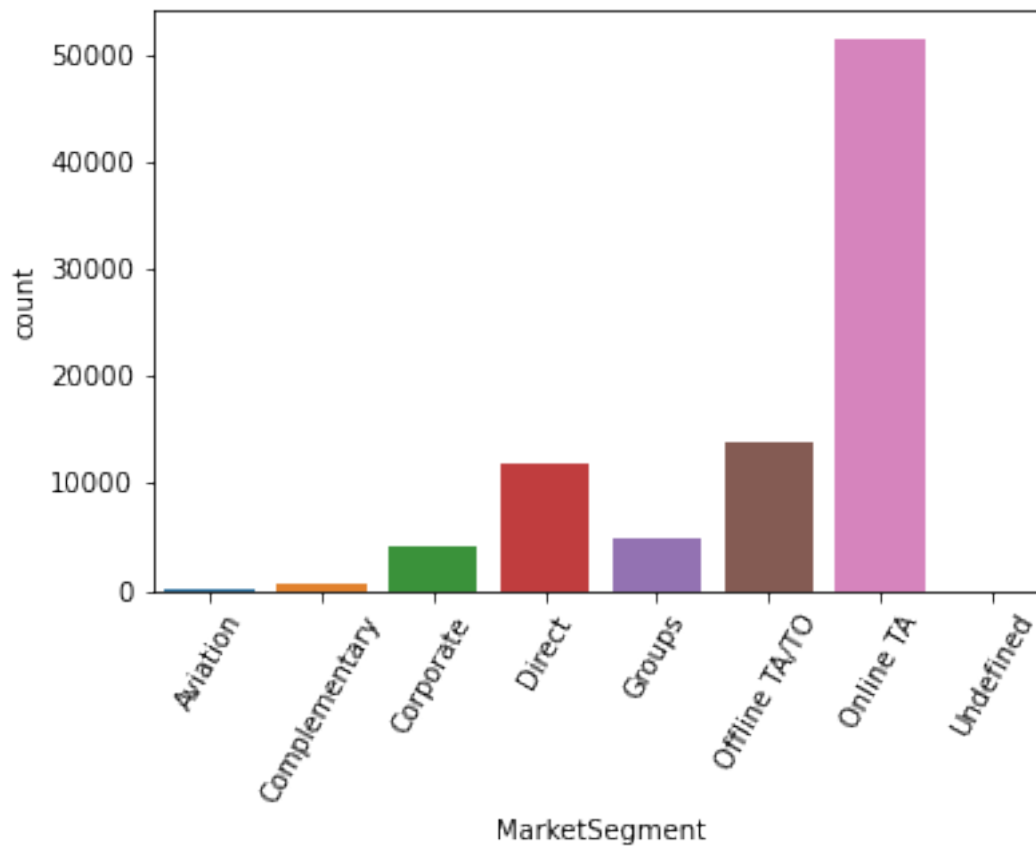
```
[26]: chart = sb.countplot(df["Meal"]) # frequency distribution
      chart.set_xticklabels(chart.get_xticklabels())
      #plt.savefig("hist of Meal.png" )
      plt.show()
```



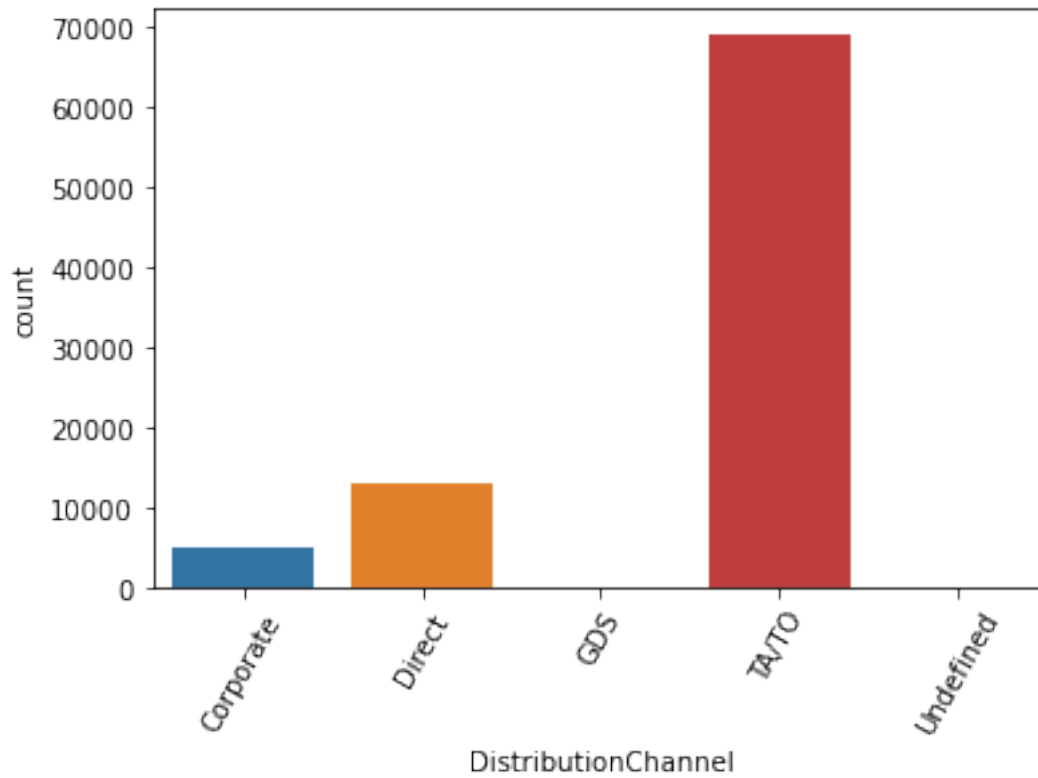
```
[27]: chart = sb.countplot(df["Country"]) # frequency distribution
      chart.set_xticklabels(chart.get_xticklabels())
      #plt.savefig("hist of Country.png" )
      plt.show()
```



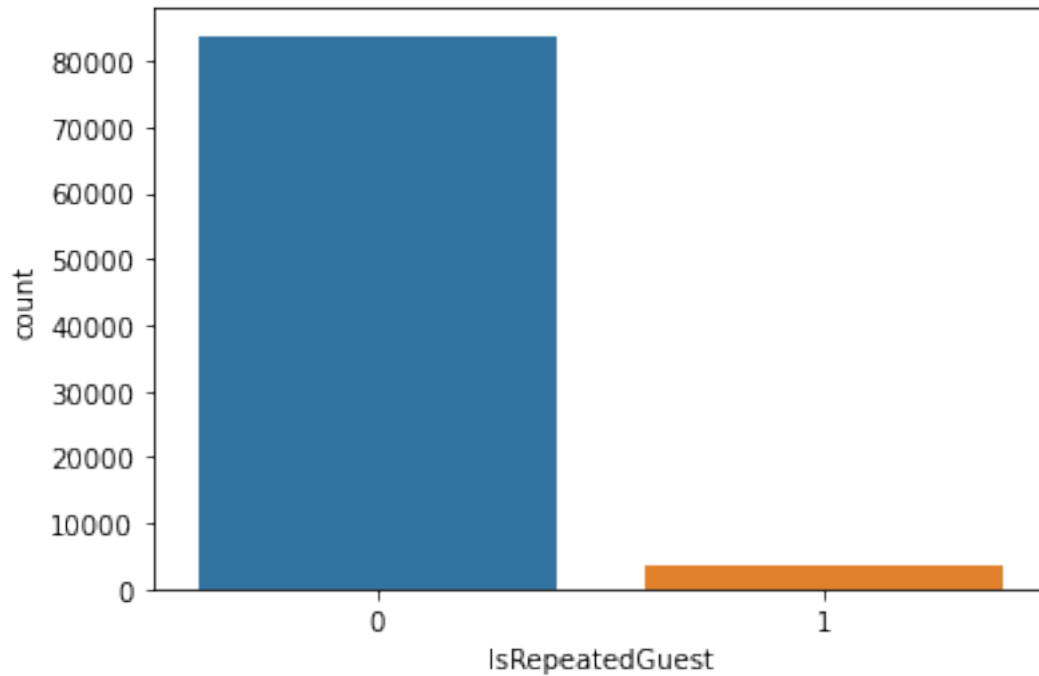
```
[299]: chart = sb.countplot(df["MarketSegment"]) # frequency distribution
chart.set_xticklabels(chart.get_xticklabels(),rotation=60)
#plt.savefig("hist of MarketSegment.png",bbox_inches='tight' )
plt.show()
```



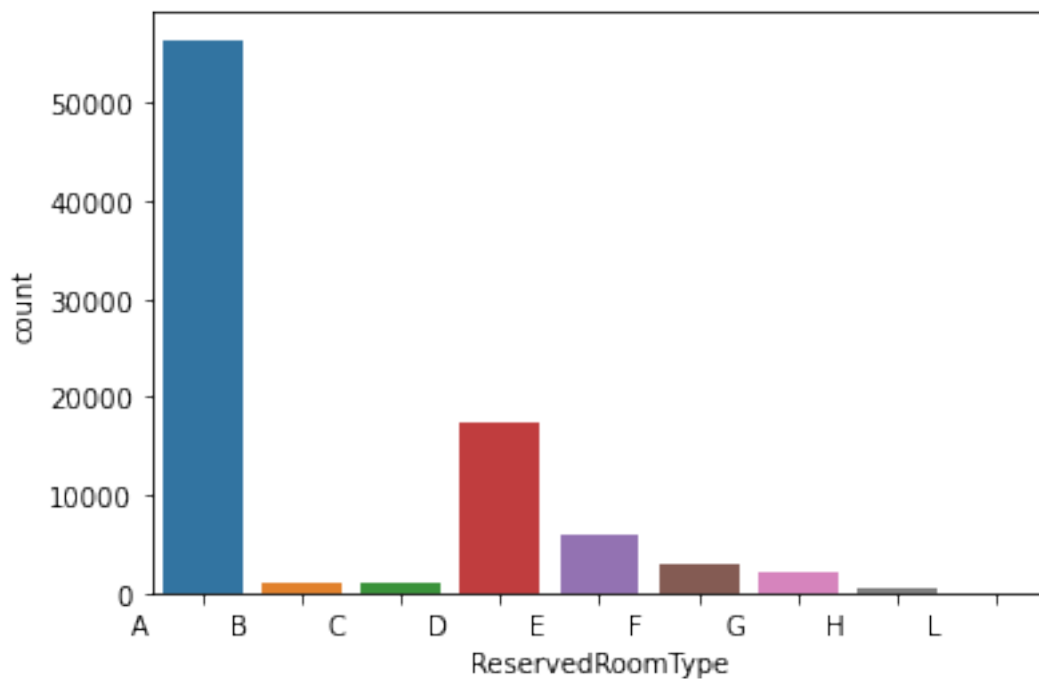
```
[28]: chart = sb.countplot(df["DistributionChannel"]) # frequency distribution
chart.set_xticklabels(chart.get_xticklabels(),rotation=60)
plt.savefig("hist of DistributionChannel.png",bbox_inches='tight' )
plt.show()
```



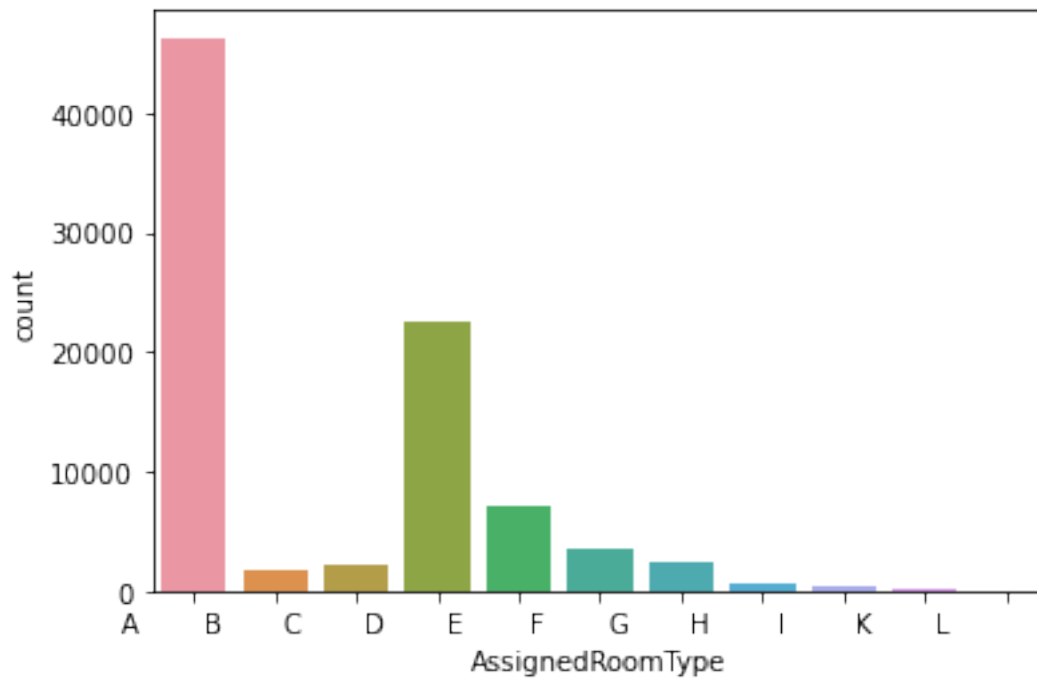
```
[29]: chart = sb.countplot(df["IsRepeatedGuest"]) # frequency distribution
chart.set_xticklabels(chart.get_xticklabels())
#plt.savefig("hist of IsRepeatedGuest.png" )
plt.show()
```



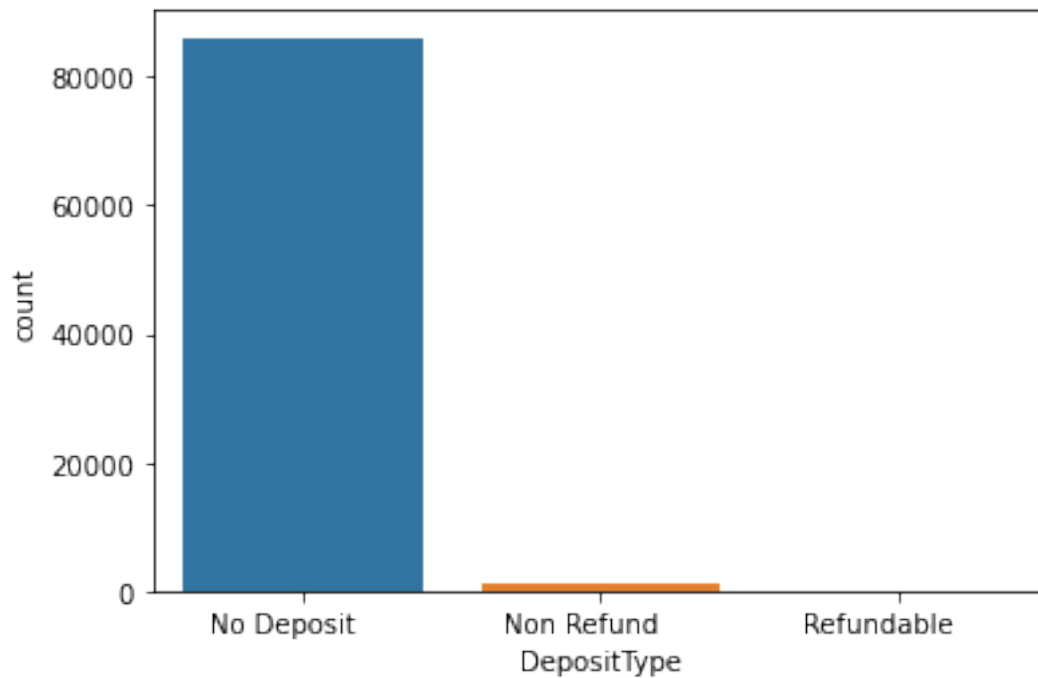
```
[30]: chart = sb.countplot(df["ReservedRoomType"]) # frequency distribution
      chart.set_xticklabels(chart.get_xticklabels())
      #plt.savefig("hist of ReservedRoomType.png" )
      plt.show()
```



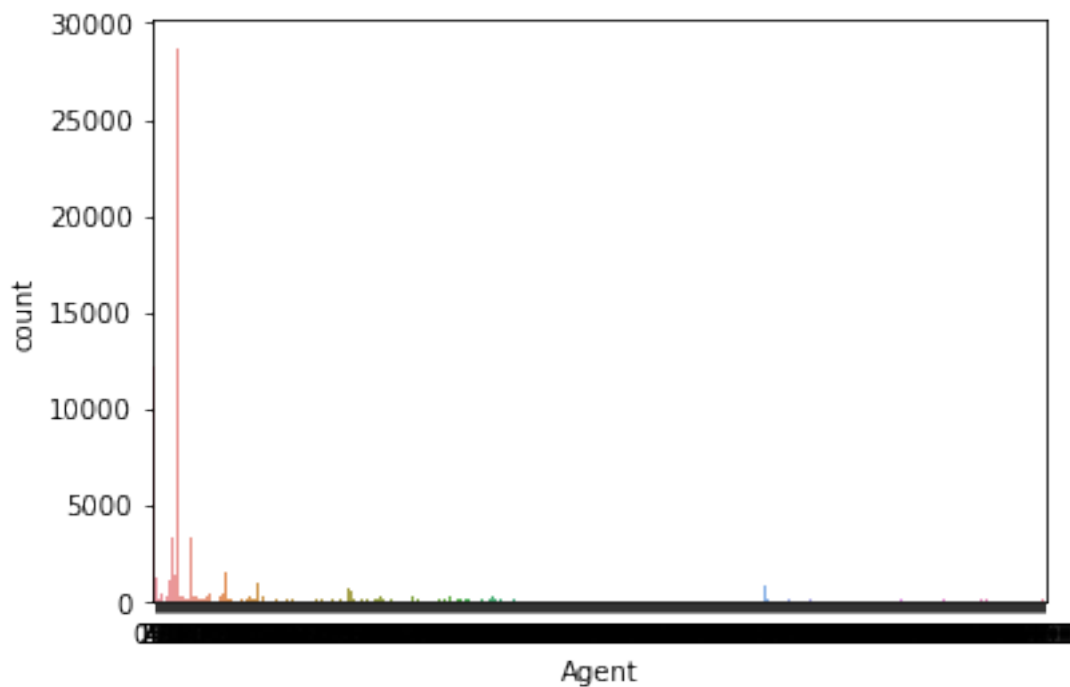
```
[31]: chart = sb.countplot(df["AssignedRoomType"]) # frequency distribution
chart.set_xticklabels(chart.get_xticklabels())
#plt.savefig("hist of AssignedRoomType.png" )
plt.show()
```



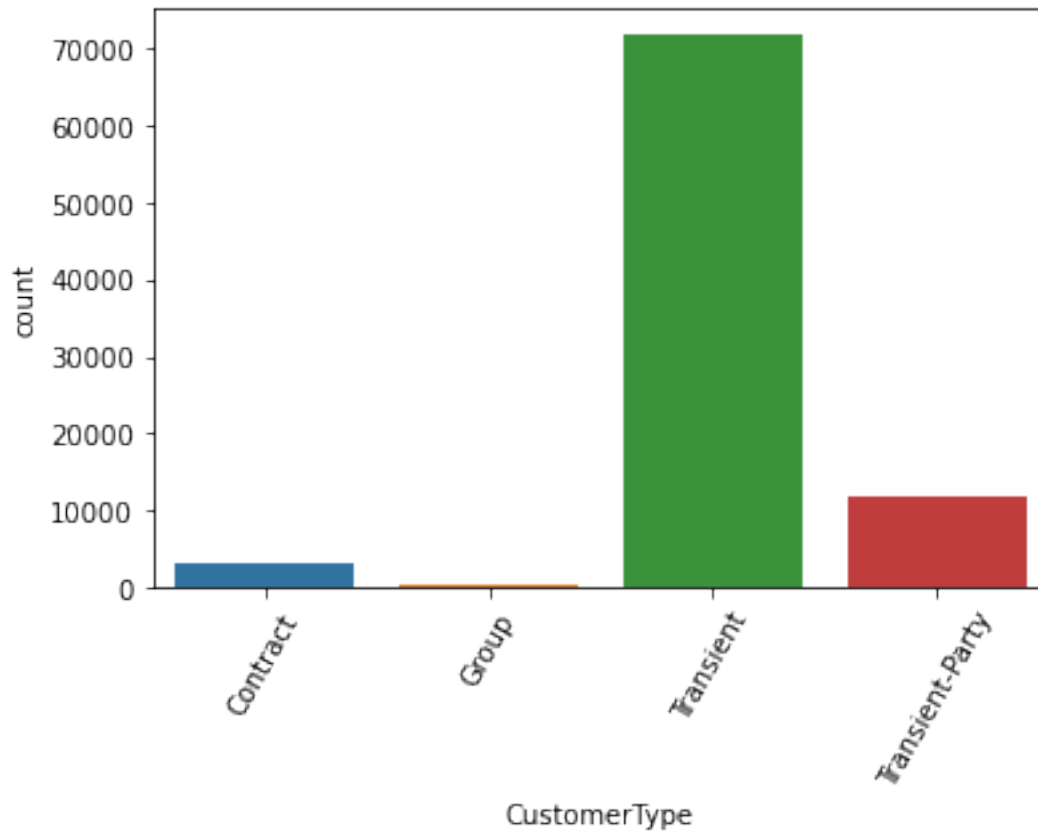
```
[32]: chart = sb.countplot(df["DepositType"]) # frequency distribution
chart.set_xticklabels(chart.get_xticklabels())
#plt.savefig("hist of DepositType.png" )
plt.show()
```



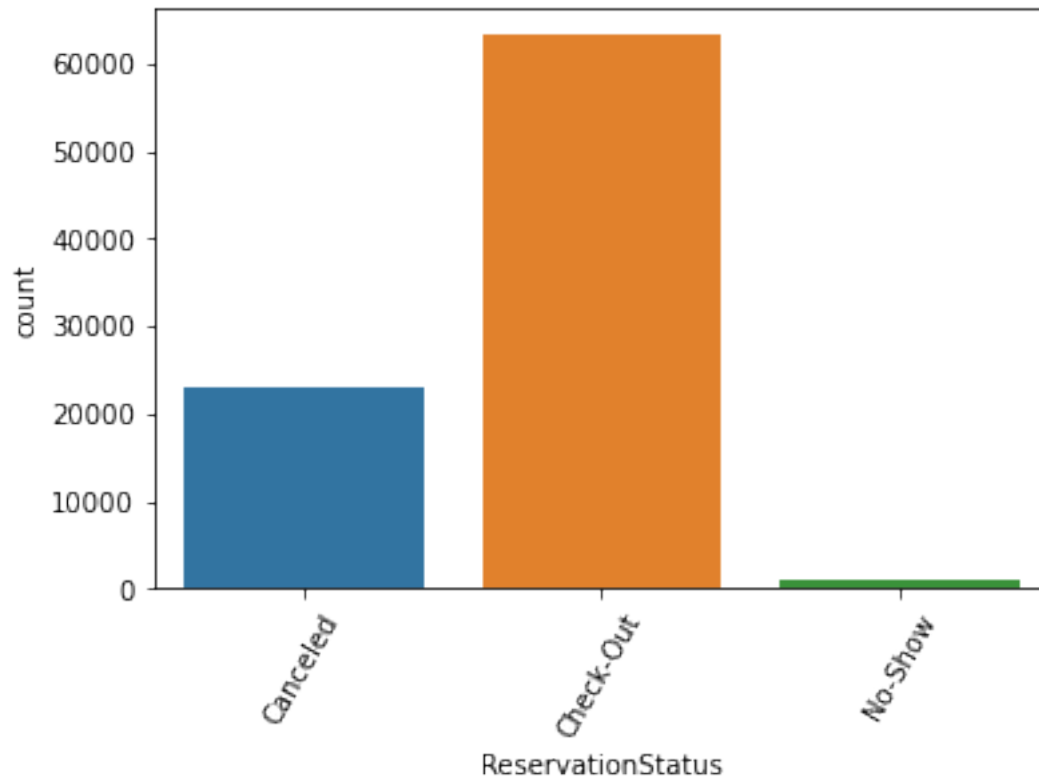
```
[33]: chart = sb.countplot(df["Agent"]) # frequency distribution
      chart.set_xticklabels(chart.get_xticklabels())
      #plt.savefig("hist of Agent.png" )
      plt.show()
```



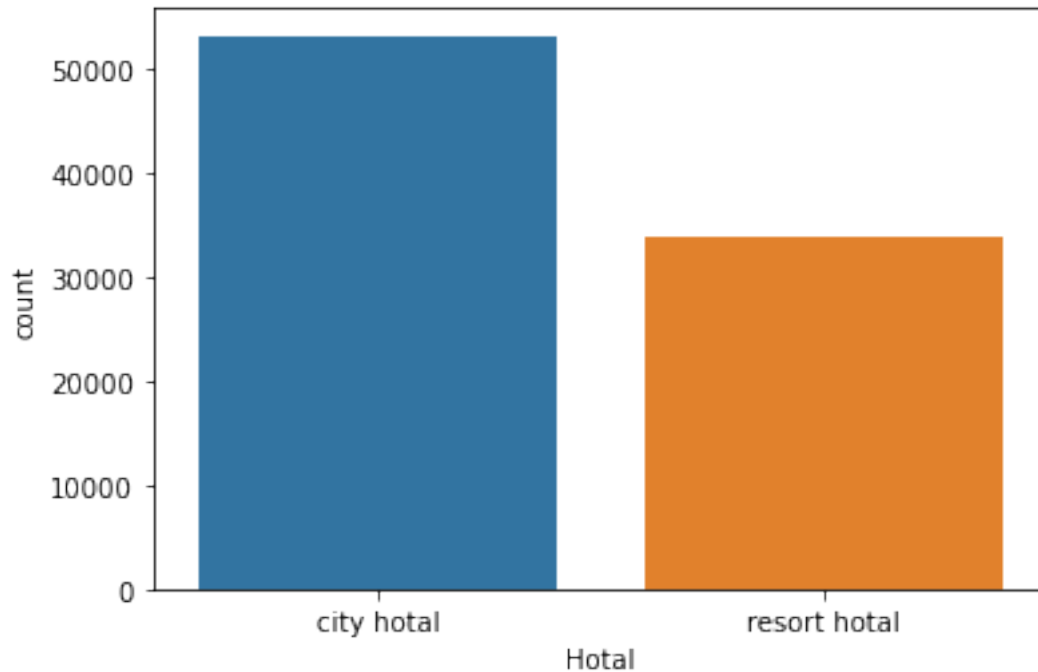
```
[34]: chart = sb.countplot(df["CustomerType"]) # frequency distribution
chart.set_xticklabels(chart.get_xticklabels(),rotation=60)
#plt.savefig("hist of CustomerType.png",bbox_inches='tight' )
plt.show()
```



```
[35]: chart = sb.countplot(df["ReservationStatus"]) # frequency distribution
chart.set_xticklabels(chart.get_xticklabels(),rotation=60)
#plt.savefig("hist of ReservationStatus.png",bbox_inches='tight' )
plt.show()
```

```
[36]: chart = sb.countplot(df["Hotal"]) # frequency distribution
      chart.set_xticklabels(chart.get_xticklabels())
      #plt.savefig("hist of Hotal.png" )
      plt.show()
```



3 correlation

```
[109]: corr = df.corr().round(2)
corr
#corr.to_csv("corr2.csv", index=True)
```

```
[109]:
```

| | LeadTime | ArrivalDateWeekNumber \ |
|-----------------------------|----------|-------------------------|
| LeadTime | 1.00 | 0.10 |
| ArrivalDateWeekNumber | 0.10 | 1.00 |
| ArrivalDateDayOfMonth | 0.01 | 0.09 |
| StaysInWeekendNights | 0.24 | 0.03 |
| StaysInWeekNights | 0.31 | 0.03 |
| Adults | 0.14 | 0.03 |
| Children | 0.03 | 0.01 |
| Babies | -0.00 | 0.01 |
| PreviousCancellations | 0.01 | 0.01 |
| PreviousBookingsNotCanceled | -0.08 | -0.02 |
| BookingChanges | 0.08 | 0.01 |
| DaysInWaitingList | 0.13 | 0.01 |
| ADR | 0.02 | 0.10 |
| RequiredCarParkingSpaces | -0.09 | 0.01 |
| TotalOfSpecialRequests | 0.03 | 0.05 |

| | ArrivalDateDayOfMonth | StaysInWeekendNights \ |
|-----------------------------|-----------------------|------------------------|
| LeadTime | 0.01 | 0.24 |
| ArrivalDateWeekNumber | 0.09 | 0.03 |
| ArrivalDateDayOfMonth | 1.00 | -0.02 |
| StaysInWeekendNights | -0.02 | 1.00 |
| StaysInWeekNights | -0.03 | 0.55 |
| Adults | -0.00 | 0.09 |
| Children | 0.02 | 0.03 |
| Babies | -0.00 | 0.01 |
| PreviousCancellations | -0.01 | -0.02 |
| PreviousBookingsNotCanceled | 0.00 | -0.06 |
| BookingChanges | 0.01 | 0.03 |
| DaysInWaitingList | 0.01 | -0.03 |
| ADR | 0.02 | 0.04 |
| RequiredCarParkingSpaces | 0.01 | -0.04 |
| TotalOfSpecialRequests | -0.00 | 0.03 |

| | StaysInWeekNights | Adults | Children | Babies \ |
|-----------------------------|-------------------|--------|----------|----------|
| LeadTime | 0.31 | 0.14 | 0.03 | -0.00 |
| ArrivalDateWeekNumber | 0.03 | 0.03 | 0.01 | 0.01 |
| ArrivalDateDayOfMonth | -0.03 | -0.00 | 0.02 | -0.00 |
| StaysInWeekendNights | 0.55 | 0.09 | 0.03 | 0.01 |
| StaysInWeekNights | 1.00 | 0.10 | 0.03 | 0.02 |
| Adults | 0.10 | 1.00 | 0.02 | 0.02 |
| Children | 0.03 | 0.02 | 1.00 | 0.02 |
| Babies | 0.02 | 0.02 | 0.02 | 1.00 |
| PreviousCancellations | -0.02 | -0.04 | -0.02 | -0.01 |
| PreviousBookingsNotCanceled | -0.06 | -0.12 | -0.03 | -0.01 |
| BookingChanges | 0.07 | -0.04 | 0.03 | 0.08 |
| DaysInWaitingList | 0.00 | -0.01 | -0.02 | -0.01 |
| ADR | 0.06 | 0.24 | 0.33 | 0.02 |
| RequiredCarParkingSpaces | -0.04 | 0.01 | 0.04 | 0.03 |
| TotalOfSpecialRequests | 0.04 | 0.11 | 0.04 | 0.09 |

| | PreviousCancellations \ |
|-----------------------------|-------------------------|
| LeadTime | 0.01 |
| ArrivalDateWeekNumber | 0.01 |
| ArrivalDateDayOfMonth | -0.01 |
| StaysInWeekendNights | -0.02 |
| StaysInWeekNights | -0.02 |
| Adults | -0.04 |
| Children | -0.02 |
| Babies | -0.01 |
| PreviousCancellations | 1.00 |
| PreviousBookingsNotCanceled | 0.39 |
| BookingChanges | -0.01 |
| DaysInWaitingList | 0.00 |

| | |
|--------------------------|-------|
| ADR | -0.05 |
| RequiredCarParkingSpaces | -0.00 |
| TotalOfSpecialRequests | 0.00 |

| | | | |
|-----------------------------|-----------------------------|----------------|---|
| | PreviousBookingsNotCanceled | BookingChanges | \ |
| LeadTime | -0.08 | 0.08 | |
| ArrivalDateWeekNumber | -0.02 | 0.01 | |
| ArrivalDateDayOfMonth | 0.00 | 0.01 | |
| StaysInWeekendNights | -0.06 | 0.03 | |
| StaysInWeekNights | -0.06 | 0.07 | |
| Adults | -0.12 | -0.04 | |
| Children | -0.03 | 0.03 | |
| Babies | -0.01 | 0.08 | |
| PreviousCancellations | 0.39 | -0.01 | |
| PreviousBookingsNotCanceled | 1.00 | 0.01 | |
| BookingChanges | 0.01 | 1.00 | |
| DaysInWaitingList | -0.01 | 0.02 | |
| ADR | -0.09 | 0.01 | |
| RequiredCarParkingSpaces | 0.04 | 0.05 | |
| TotalOfSpecialRequests | 0.03 | 0.02 | |

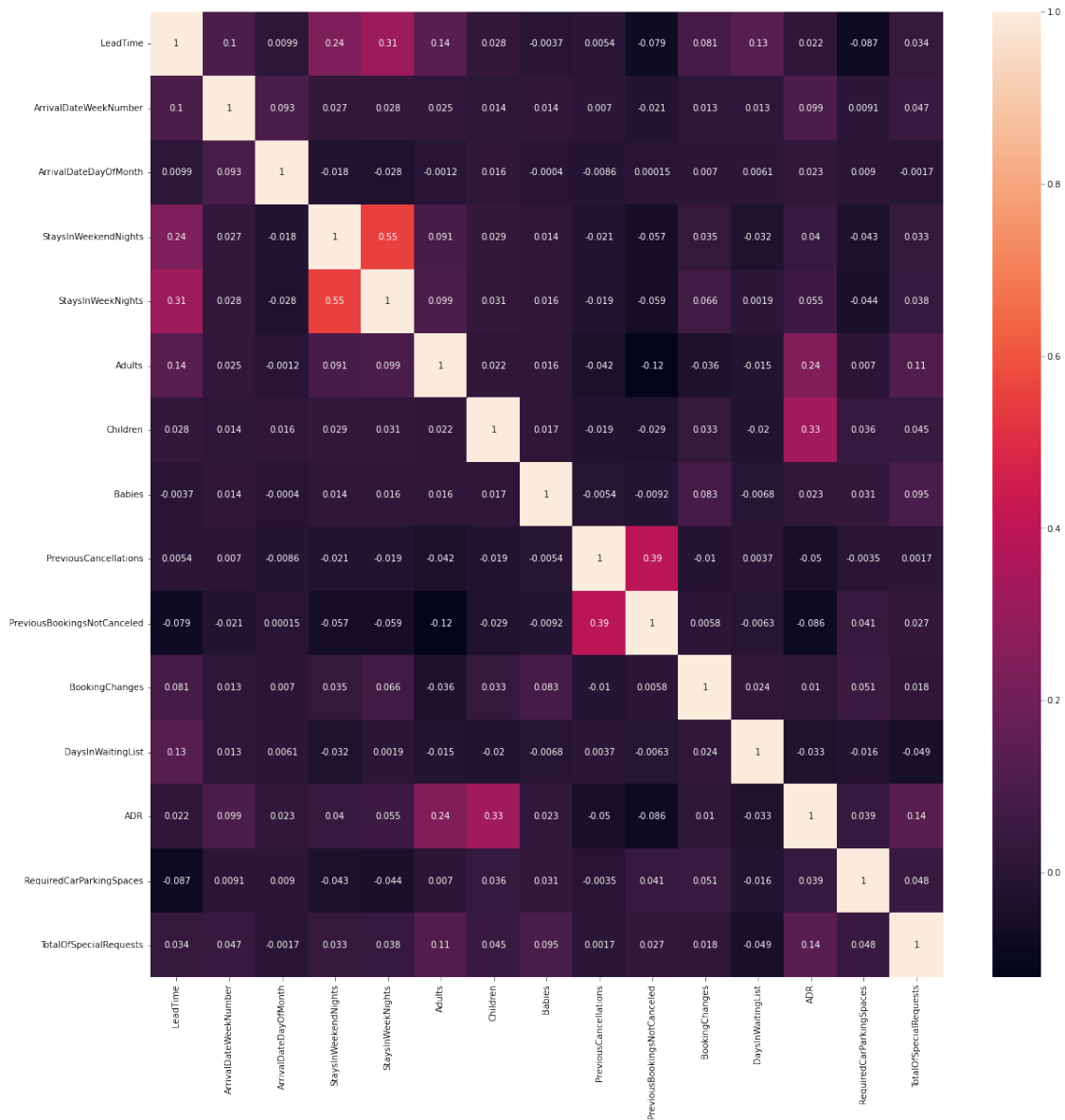
| | | | |
|-----------------------------|-------------------|-------|---|
| | DaysInWaitingList | ADR | \ |
| LeadTime | 0.13 | 0.02 | |
| ArrivalDateWeekNumber | 0.01 | 0.10 | |
| ArrivalDateDayOfMonth | 0.01 | 0.02 | |
| StaysInWeekendNights | -0.03 | 0.04 | |
| StaysInWeekNights | 0.00 | 0.06 | |
| Adults | -0.01 | 0.24 | |
| Children | -0.02 | 0.33 | |
| Babies | -0.01 | 0.02 | |
| PreviousCancellations | 0.00 | -0.05 | |
| PreviousBookingsNotCanceled | -0.01 | -0.09 | |
| BookingChanges | 0.02 | 0.01 | |
| DaysInWaitingList | 1.00 | -0.03 | |
| ADR | -0.03 | 1.00 | |
| RequiredCarParkingSpaces | -0.02 | 0.04 | |
| TotalOfSpecialRequests | -0.05 | 0.14 | |

| | | |
|-----------------------|--------------------------|------------------------|
| | RequiredCarParkingSpaces | TotalOfSpecialRequests |
| LeadTime | -0.09 | 0.03 |
| ArrivalDateWeekNumber | 0.01 | 0.05 |
| ArrivalDateDayOfMonth | 0.01 | -0.00 |
| StaysInWeekendNights | -0.04 | 0.03 |
| StaysInWeekNights | -0.04 | 0.04 |
| Adults | 0.01 | 0.11 |
| Children | 0.04 | 0.04 |
| Babies | 0.03 | 0.09 |

| | | |
|-----------------------------|-------|-------|
| PreviousCancellations | -0.00 | 0.00 |
| PreviousBookingsNotCanceled | 0.04 | 0.03 |
| BookingChanges | 0.05 | 0.02 |
| DaysInWaitingList | -0.02 | -0.05 |
| ADR | 0.04 | 0.14 |
| RequiredCarParkingSpaces | 1.00 | 0.05 |
| TotalOfSpecialRequests | 0.05 | 1.00 |

4 Correlation Heatmap

```
[39]: plt.figure(figsize=(20,20))
      sb.heatmap(df.corr(), annot=True)
      #plt.savefig("corr.png",bbox_inches='tight' )
      plt.show()
```



```
[41]: dfX = df[df.columns[~df.columns.isin(category)]]
dfX
```

```
[41]:
```

| | LeadTime | ArrivalDateWeekNumber | ArrivalDateDayOfMonth | \ |
|-------|----------|-----------------------|-----------------------|---|
| 0 | 342 | 27 | 1 | |
| 1 | 737 | 27 | 1 | |
| 2 | 7 | 27 | 1 | |
| 3 | 13 | 27 | 1 | |
| 4 | 14 | 27 | 1 | |
| ... | ... | ... | ... | |
| 87225 | 23 | 35 | 30 | |

| | | | |
|-------|-----|----|----|
| 87226 | 102 | 35 | 31 |
| 87227 | 34 | 35 | 31 |
| 87228 | 109 | 35 | 31 |
| 87229 | 205 | 35 | 29 |

| | StaysInWeekendNights | StaysInWeekNights | 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 | |
| ... | ... | ... | ... | ... | | |
| 87225 | 2 | 5 | 2 | 0.0 | 0 | |
| 87226 | 2 | 5 | 3 | 0.0 | 0 | |
| 87227 | 2 | 5 | 2 | 0.0 | 0 | |
| 87228 | 2 | 5 | 2 | 0.0 | 0 | |
| 87229 | 2 | 7 | 2 | 0.0 | 0 | |

| | PreviousCancellations | PreviousBookingsNotCanceled | BookingChanges | \ |
|-------|-----------------------|-----------------------------|----------------|---|
| 0 | 0 | | 3 | |
| 1 | 0 | | 4 | |
| 2 | 0 | | 0 | |
| 3 | 0 | | 0 | |
| 4 | 0 | | 0 | |
| ... | ... | ... | ... | |
| 87225 | 0 | | 0 | 0 |
| 87226 | 0 | | 0 | 0 |
| 87227 | 0 | | 0 | 0 |
| 87228 | 0 | | 0 | 0 |
| 87229 | 0 | | 0 | 0 |

| | DaysInWaitingList | ADR | RequiredCarParkingSpaces | \ |
|-------|-------------------|--------|--------------------------|---|
| 0 | 0 | 0.00 | 0 | |
| 1 | 0 | 0.00 | 0 | |
| 2 | 0 | 75.00 | 0 | |
| 3 | 0 | 75.00 | 0 | |
| 4 | 0 | 98.00 | 0 | |
| ... | ... | ... | ... | |
| 87225 | 0 | 96.14 | 0 | |
| 87226 | 0 | 225.43 | 0 | |
| 87227 | 0 | 157.71 | 0 | |
| 87228 | 0 | 104.40 | 0 | |
| 87229 | 0 | 151.20 | 0 | |

| | TotalOfSpecialRequests |
|---|------------------------|
| 0 | 0 |
| 1 | 0 |

```

2          0
3          0
4          1
...
87225      0
87226      2
87227      4
87228      0
87229      2

```

```
[87230 rows x 15 columns]
```

5 Normalize the data set using Min-Max Scaling:

```
[42]: from sklearn import preprocessing
```

```
[43]: scaler = preprocessing.MinMaxScaler()
```

```
[44]: names = dfX.columns
```

```
[45]: d = scaler.fit_transform(dfX)
```

```
[46]: scaled_df = pd.DataFrame(d, columns=names)
```

```
[110]: scaled_df.head().T
```

```
[110]:
```

| | 0 | 1 | 2 | 3 | 4 |
|-----------------------------|----------|----------|----------|----------|----------|
| LeadTime | 0.464043 | 1.000000 | 0.009498 | 0.017639 | 0.018996 |
| ArrivalDateWeekNumber | 0.500000 | 0.500000 | 0.500000 | 0.500000 | 0.500000 |
| ArrivalDateDayOfMonth | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| StaysInWeekendNights | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| StaysInWeekNights | 0.000000 | 0.000000 | 0.020000 | 0.020000 | 0.040000 |
| Adults | 0.036364 | 0.036364 | 0.018182 | 0.018182 | 0.036364 |
| Children | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| Babies | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| PreviousCancellations | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| PreviousBookingsNotCanceled | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| BookingChanges | 0.166667 | 0.222222 | 0.000000 | 0.000000 | 0.000000 |
| DaysInWaitingList | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| ADR | 0.001180 | 0.001180 | 0.015053 | 0.015053 | 0.019307 |
| RequiredCarParkingSpaces | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| TotalOfSpecialRequests | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.200000 |

6 converting response variable to int so that KNN will treat this attribute as numeric attribute

```
[ ]: dfY = df["IsCanceled"]  
dfY = dfY.astype(int)
```

7 Divide the dataset to training and test sets.

```
[49]: from sklearn.model_selection import train_test_split
```

```
[50]: X_train, X_test, y_train, y_test = train_test_split(scaled_df, dfY, test_size=0.  
    ↳ 2, random_state=40)
```

8 Modeling with KNN algorithm

```
[ ]: # The K-NN algorithm can be used for both classification and regression,  
    # but it is more commonly employed for classification.  
    # The K-Nearest Neighbour algorithm is based on the Supervised Learning technique  
    # it is one of the simplest Machine Learning algorithms.  
    # The K-NN method thinks that the new case/data and existing cases are  
    ↳ comparable,  
    # and it places the new case in the category that is closest to the existing  
    ↳ categories.
```

```
[51]: from sklearn.neighbors import KNeighborsClassifier
```

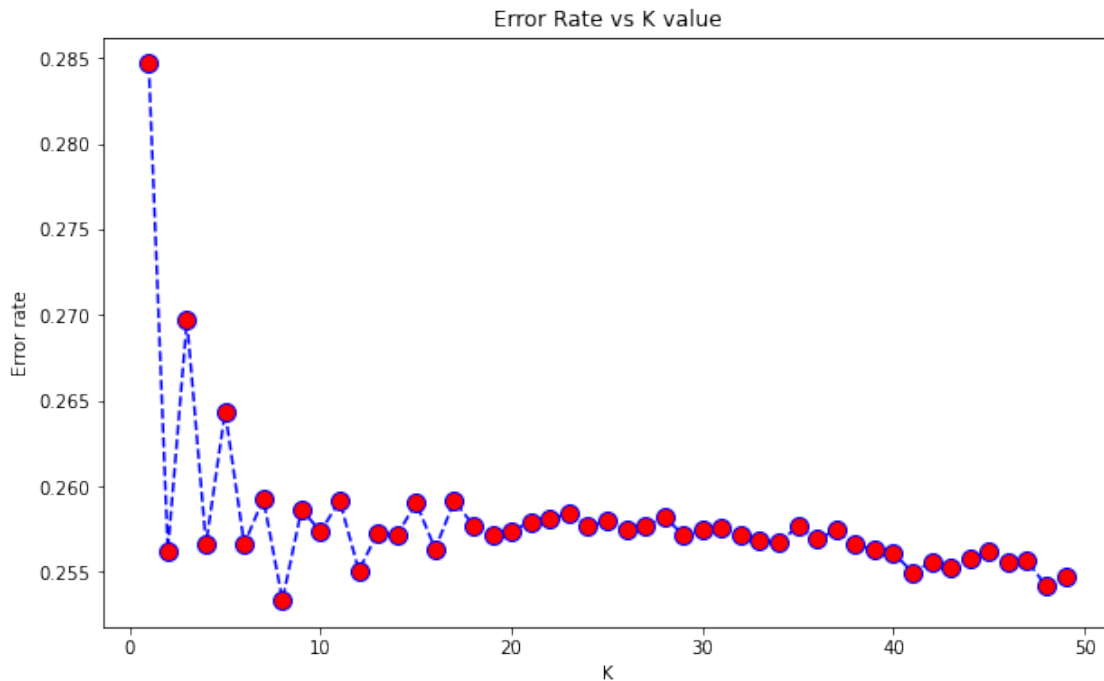
8.0.1 Choosing k-value

```
[52]: error_rate=[]  
for i in range(1,50): # sqrt of rows is 47.3  
    knn=KNeighborsClassifier(n_neighbors=i)  
    knn.fit(X_train,y_train)  
    pred=knn.predict(X_test)  
    error_rate.append(np.mean(pred!=y_test))
```

```
[77]: # error rate = (1 - (correct predictions / total predictions)) * 100
```

```
[78]: plt.figure(figsize=(10,6))  
plt.  
    ↳ plot(range(1,50),error_rate,color='blue',linestyle='dashed',marker='o',markerfacecolor='red'  
plt.title('Error Rate vs K value')
```

```
plt.xlabel('K')
plt.ylabel('Error rate')
## Display the visualization of the Confusion Matrix.
plt.savefig("Error Rate corresponding knn.png",bbox_inches='tight' )
plt.show()
```



8.0.2 checking with k=8

```
[54]: knn=KNeighborsClassifier(n_neighbors=8)
      knn.fit(X_train,y_train)
      pred_k=knn.predict(X_test)
      pred_k
```

```
[54]: array([0, 0, 0, ..., 0, 0, 0])
```

```
[66]: from sklearn.metrics import
      ↪ classification_report, confusion_matrix, accuracy_score
      cf_matrix = confusion_matrix(y_test, pred_k)
      print(cf_matrix)
      print(classification_report(y_test, pred_k))
      print(accuracy_score(y_test, pred_k)*100)
```

```
[[11741  872]
 [ 3548 1285]]
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.77 | 0.93 | 0.84 | 12613 |
| 1 | 0.60 | 0.27 | 0.37 | 4833 |
| accuracy | | | 0.75 | 17446 |
| macro avg | 0.68 | 0.60 | 0.60 | 17446 |
| weighted avg | 0.72 | 0.75 | 0.71 | 17446 |

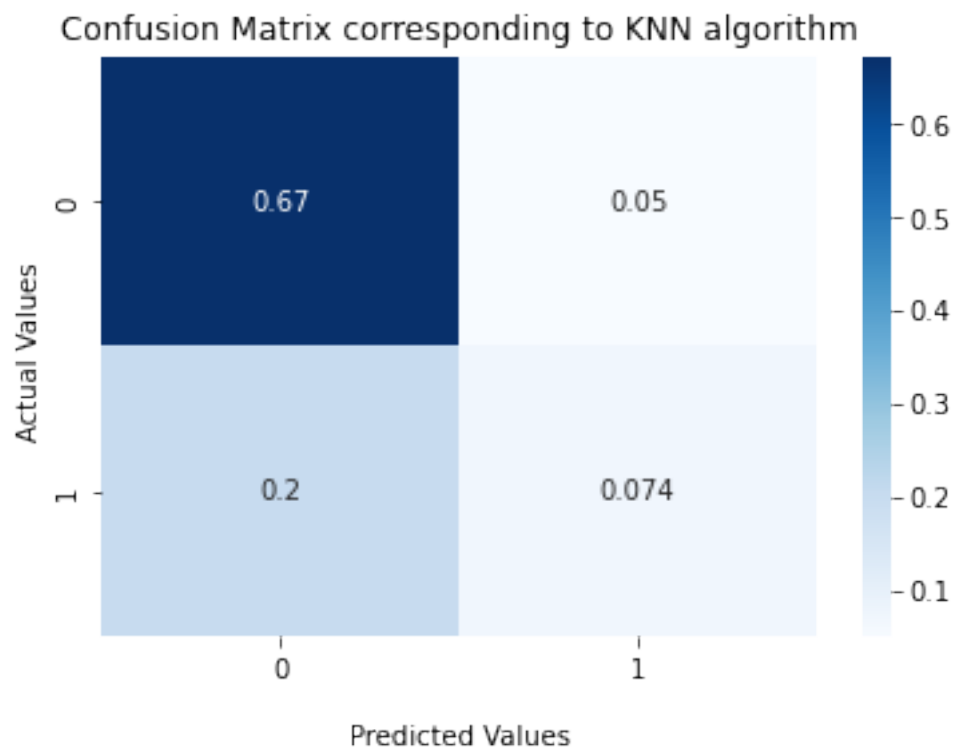
74.66467958271237

```
[99]: ax = sb.heatmap(cf_matrix/np.sum(cf_matrix), annot=True, cmap='Blues')

ax.set_title('Confusion Matrix corresponding to KNN algorithm');
ax.set_xlabel('\nPredicted Values')
ax.set_ylabel('Actual Values ');

## Ticket labels - List must be in alphabetical order
ax.xaxis.set_ticklabels(['0', '1'])
ax.yaxis.set_ticklabels(['0', '1'])

## Display the visualization of the Confusion Matrix.
# plt.savefig("cf_matrix1.png",bbox_inches='tight' )
plt.show()
```

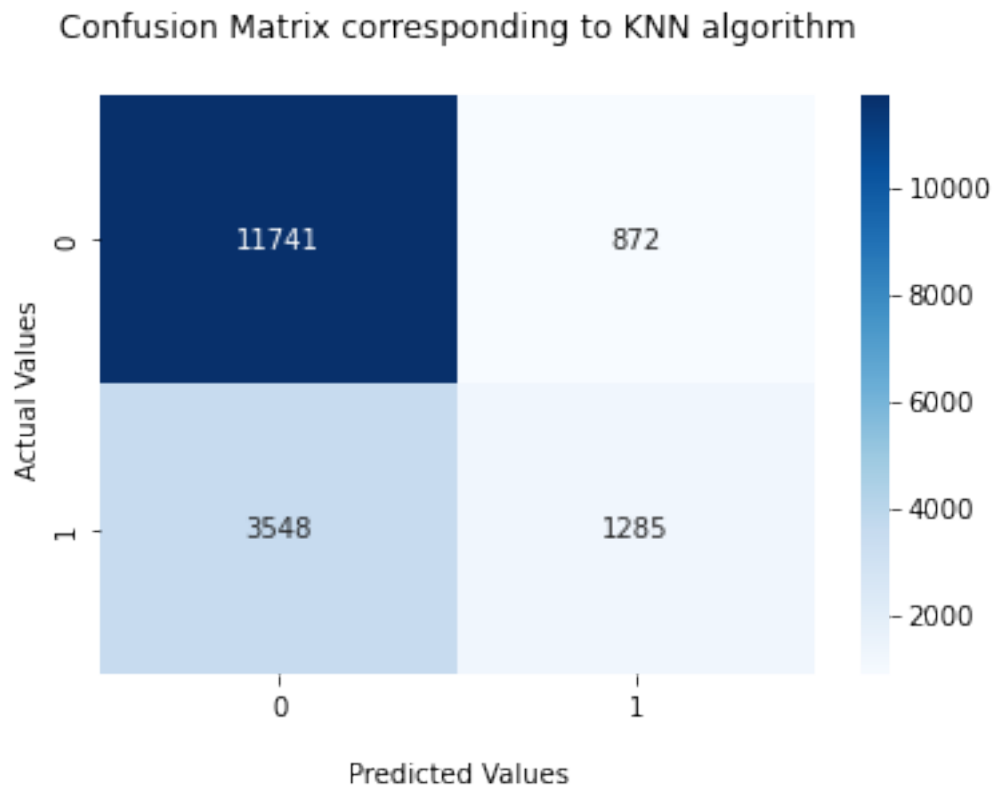


```
[97]: ax = sb.heatmap(cf_matrix, annot=True, fmt="d", cmap='Blues')

ax.set_title(' Confusion Matrix corresponding to KNN algorithm \n');
ax.set_xlabel('\nPredicted Values')
ax.set_ylabel('Actual Values ');

## Ticket labels - List must be in alphabetical order
ax.xaxis.set_ticklabels(['0','1'])
ax.yaxis.set_ticklabels(['0','1'])

## Display the visualization of the Confusion Matrix.
plt.savefig("cf_matrix_KNN.png",bbox_inches='tight' )
plt.show()
```

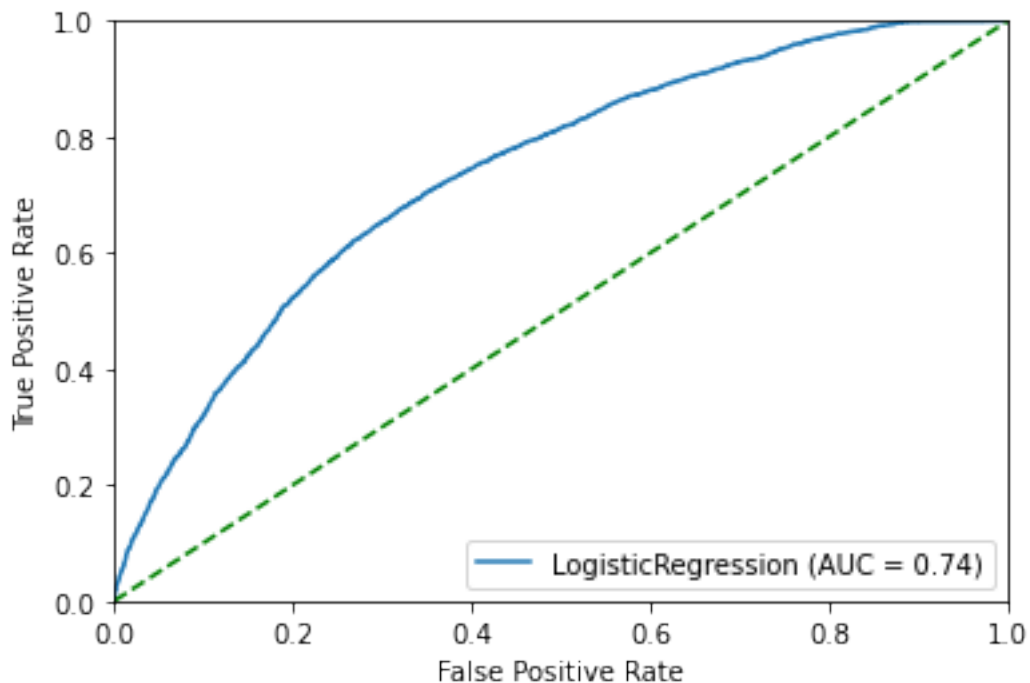


8.0.3 Logistic Regression

```
[83]: # import sklearn.cross_validation from train_test_split
```

```
[87]: from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(X_train,y_train)
metrics.plot_roc_curve(model, X_test, y_test)
plt.plot([0, 1], [0, 1], 'g--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.0])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve\n\n')
#plt.savefig("ROC Curve.png",bbox_inches='tight' )
plt.show()
```

ROC Curve



```
[89]: predict_logistic=model.predict(X_test)
```

```
[94]: from sklearn import metrics
from sklearn.metrics import
    confusion_matrix,classification_report,accuracy_score
cf_matrix_logistic = confusion_matrix(y_test,predict_logistic)
print(confusion_matrix(y_test,predict_logistic))
classification_report_logistic = classification_report(y_test,predict_logistic)
```

```
print(classification_report_logistic)
print(accuracy_score(y_test,predict_logistic)*100)
```

```
[[12185   428]
 [ 4124   709]]

              precision    recall  f1-score   support

     0       0.75         0.97         0.84        12613
     1       0.62         0.15         0.24         4833

 accuracy                   0.74        17446
 macro avg              0.69         0.56         0.54        17446
 weighted avg           0.71         0.74         0.67        17446
```

73.90805915396079

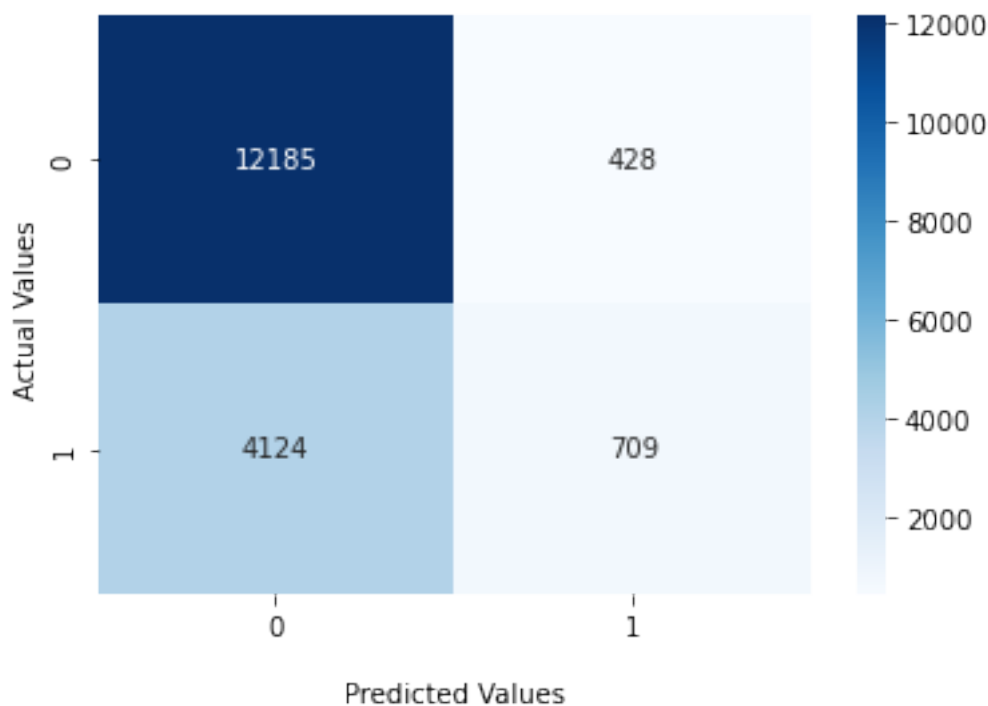
```
[100]: ax = sb.heatmap(cf_matrix_logistic, annot=True, fmt="d", cmap='Blues')

ax.set_title('Confusion Matrix corresponding to Logistic regression model \n');
ax.set_xlabel('\nPredicted Values')
ax.set_ylabel('Actual Values ');

## Ticket labels - List must be in alphabetical order
ax.xaxis.set_ticklabels(['0','1'])
ax.yaxis.set_ticklabels(['0','1'])

## Display the visualization of the Confusion Matrix.
# plt.savefig("Confusion Matrix corresponding to Logistic regression model.
→png",bbox_inches='tight' )
plt.show()
```

Confusion Matrix corresponding to Logistic regression model



```
[101]: from sklearn.ensemble import RandomForestClassifier
```

```
[102]: model=RandomForestClassifier(n_estimators=20)
```

```
[103]: model.fit(X_train,y_train)
```

```
[103]: RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                             criterion='gini', max_depth=None, max_features='auto',
                             max_leaf_nodes=None, max_samples=None,
                             min_impurity_decrease=0.0, min_impurity_split=None,
                             min_samples_leaf=1, min_samples_split=2,
                             min_weight_fraction_leaf=0.0, n_estimators=20,
                             n_jobs=None, oob_score=False, random_state=None,
                             verbose=0, warm_start=False)
```

9 predicted value

```
[105]: model_predicted=model.predict(X_test)
model_predicted
```

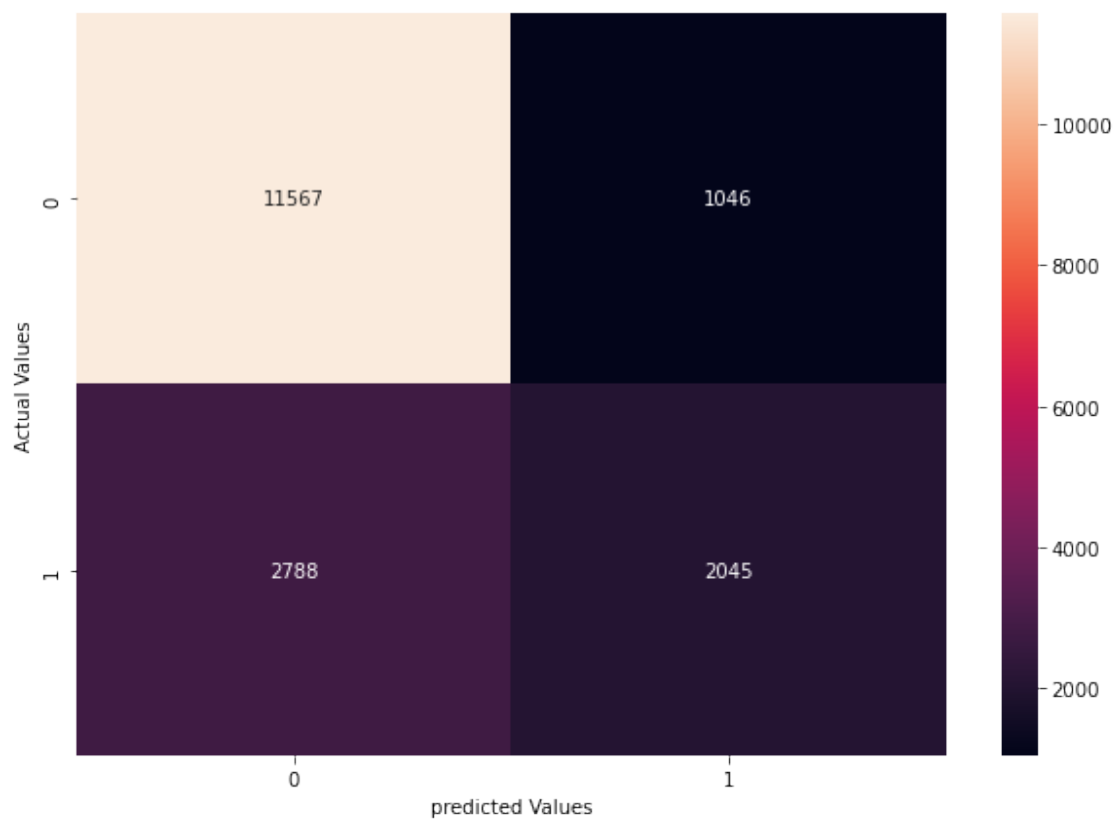
```
[105]: array([0, 0, 0, ..., 0, 0, 1])
```

```
[106]: from sklearn.metrics import confusion_matrix
matrix=confusion_matrix(y_test,model_predicted)
matrix
```

```
[106]: array([[11567, 1046],
       [ 2788, 2045]])
```

```
[108]: plt.figure(figsize=(10,7))
sb.heatmap(matrix,annot=True,fmt="d")
plt.xlabel("predicted Values")
plt.ylabel("Actual Values")
```

```
[108]: Text(69.0, 0.5, 'Actual Values')
```




```
[339]: #import pandas as pd
feature_imp = pd.Series(model.feature_importances_, index=dfX.columns).
        ↪sort_values(ascending=False)
feature_imp
```

```
[339]: LeadTime          0.236495
      ADR              0.211342
      ArrivalDateDayOfMonth 0.133497
      ArrivalDateWeekNumber 0.133080
      StaysInWeekNights    0.067852
      TotalOfSpecialRequests 0.047616
      StaysInWeekendNights  0.038512
      RequiredCarParkingSpaces 0.031503
      BookingChanges        0.024730
      PreviousCancellations  0.023641
      Adults               0.022558
      Children             0.014759
      PreviousBookingsNotCanceled 0.008484
      DaysInWaitingList     0.003910
      Babies               0.002022
      dtype: float64
```

```
[340]: feature_imp.sort_values(ascending=False)
```

```
[340]: LeadTime          0.236495
      ADR              0.211342
      ArrivalDateDayOfMonth 0.133497
      ArrivalDateWeekNumber 0.133080
      StaysInWeekNights    0.067852
      TotalOfSpecialRequests 0.047616
      StaysInWeekendNights  0.038512
      RequiredCarParkingSpaces 0.031503
      BookingChanges        0.024730
      PreviousCancellations  0.023641
      Adults               0.022558
      Children             0.014759
      PreviousBookingsNotCanceled 0.008484
      DaysInWaitingList     0.003910
      Babies               0.002022
      dtype: float64
```