Hotel Booking Analysis

In this Project I am going to analyze a dataset which contains the demand of hotel bookings. I got this dataset from kaggle

The 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.

Table of Contents

Import Libraries, Functions and Data

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
plt.style.use('seaborn')
import warnings
warnings.filterwarnings('ignore')
import plotly.express as px
sns.set_style("darkgrid")
sns.set_context("poster")
plt.rcParams['font.size'] = 20
plt.rcParams['figure.figsize'] = 20,12
```

In [2]:
```
def missing_percentage(data_frame):
    """
    This function return take data frame and return data frame
    with columns name, the number of missing values and the percentage
    of them in each column.
    """
    total_missing = data_frame.isnull().sum().sort_values(ascending=False)
    total_missing = total_missing[total_missing.values !=0]
    per = np.round(total_missing.values*100.00/len(data_frame), 2)
    df = pd.DataFrame(total_missing, columns=['Number Of Missing Values'])
    return(df.assign(Percentage = per))
```

In [3]:
```
# Loading data
data = pd.read_csv('hotel_bookings.csv')
data.head(1)
```

Out[3]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_nu
0	Resort Hotel	0	342	2015	July	

1 rows × 32 columns

Exploring Data

I will check the things below to Understand the Dataset:

- **Number of samples and columns in each dataset**
- **Head of the dataset**
- **Duplicate rows in each dataset**
- **Datatypes of columns**
- **Features with missing values**
- **Number of non-null unique values for features in each dataset**
- **What those unique values are and counts for each**

In [4]:
```
data.shape
```

Out[4]: (119390, 32)

Based on the cell above

The data set being has `119390 records/cells` ***and*** `32 columns/variables` ***in our data frame.***

In [5]:
```
pd.set_option('display.max_columns', None)
# taking a look at the data
data.head()
```

Out[5]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_nu
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	

In [6]:
```
data.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
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB
```

In [7]:
```
#Looking at the number of null values for each column
#data.isnull().sum().sort_values(ascending = False)
#OR
#missing = data.isna().sum().sort_values(ascending=False) / len(data)
#missing
```

In [8]:
```
#Show Missing Values
missing_percentage(data)
```

Out[8]:

	Number Of Missing Values	Percentage
company	112593	94.31
agent	16340	13.69
country	488	0.41
children	4	0.00

In [9]:
```
# Check if the data contain duplicated row or not
data.duplicated().sum()
```

Out[9]: 31994

Exploring Summary

I will be making a few changes to clean the data and adding a few columns that might be useful for analysis, Fixing any missing or incorrect values.

- ***Droping*** `Company` ***column as it has sizable missing values , while*** `country`, `children`, ***and*** `agent` ***have missing entries that can be replaced using the mode and median.***
- ***Droping unncessary columns such*** `Arrival Date Week Number`
- ***Droping duplicated values***
- ***Droping rows with 0 guests***
- ***Droping rows with 0 or less than 0 adr (average daily rate)***
- ***Making a coulmn for arrival_date***
- ***Making a seperate data with non-cancelled reservations***
- ***Making a coulmn for total guests by adding the number of adults, children and babies***
- ***Making a coulmn to find price per guest by dividing adr with total guests***

Cleaning

- ***I'd perform cleaning operations ,which would help me to reach a more accurate result in creating meaningful and informative visualizations.***

In [10]:
```
#Drop Duplicated Values
```

```
data.drop_duplicates(inplace=True)
# confirming changes
data.duplicated().sum()
```

Out[10]: 0

Dropping `Company` for being mostly null values and `Arrival Date Week Number` for not being necessary.

In [11]:
```
#data.drop('company',axis=1,inplace=True)
#data.drop('arrival_date_week_number',axis=1,inplace=True)
data=data.drop(columns=['company','arrival_date_week_number'])
```

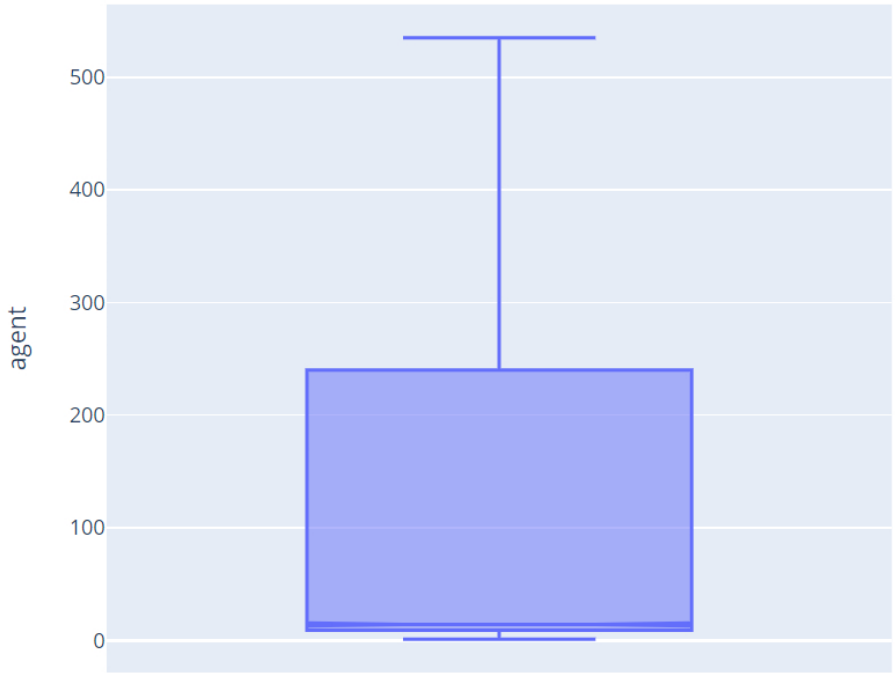
In [12]:
```
data['arrival_date'] = data['arrival_date_month'] + " " + data['arrival_date
data['arrival_date'] = pd.to_datetime(data['arrival_date'])
data.head(1)
```

Out[12]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_day_of_m
0	Resort Hotel	0	342	2015	July	

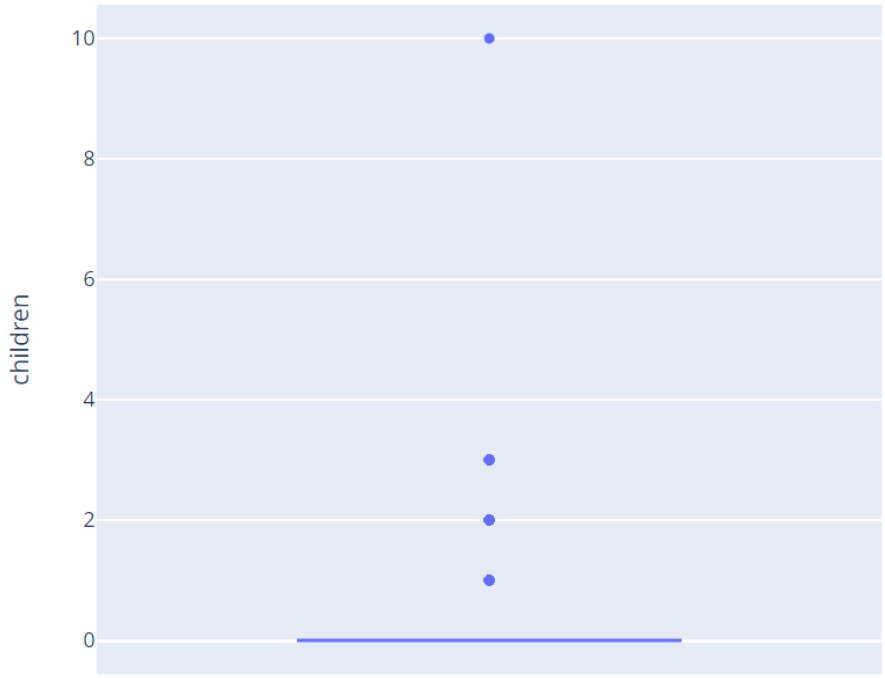
In [13]:
```
data['reservation_status_date'] = pd.to_datetime(data['reservation_status_da
```

In [14]:
```
fig = px.box(data, y='agent',notched=True)
fig.show()
```



It's skwed positive and there is no outliers, so we can replace the missing values with median.

In [15]:
```
fig = px.box(data, y='children')
fig.show()
```



We can replace the missing values with median because of the outliers.

Filling missing values for `country` ***using mode and filling the missing values of*** `children` ***and*** `agent` ***with the median because of the outliers.***

In [16]:
```
#Fill missing values
data.fillna({
    'country' :data['country'].mode()[0],
    'children' : data['children'].median(),
    'agent': data['agent'].median()
}, inplace=True)

#Check
data.isnull().sum().sort_values(ascending=False)[:4]
```

Out[16]:
```
hotel                       0
previous_cancellations      0
reservation_status_date     0
reservation_status          0
dtype: int64
```

In [17]:
```
data = data[data['adults'] != 0]
data = data[data['adr'] > 0.00]
# confirming changes
data.shape
```

Out[17]:
```
(85373, 31)
```

Adding a `total guests` ***column that includes*** `adults` ***and*** `children` ***and*** `babies`***.***

In [18]:
```
data['total_guests'] = data['adults'] + data['children']+ data['babies']
```

Steps to make analysis easier

In [19]:
```
cols = ['is_canceled', 'is_repeated_guest']
for i in cols:
    data[i].replace(1, 'Yes', inplace=True)
    data[i].replace(0, 'No', inplace=True)
data[cols].head()
```

Out[19]:

	is_canceled	is_repeated_guest
2	No	No
3	No	No
4	No	No
6	No	No
7	No	No

In [20]:
```
data.duplicated().sum()
```

Out[20]: 26

In [21]:
```
data.drop_duplicates(inplace = True)
# confirming changes
data.duplicated().sum()
```

Out[21]: 0

In [22]:
```
non_cancelled_reservations_df = data[data['is_canceled'] == 'No']
non_cancelled_reservations_df.shape
```

Out[22]: (61587, 32)

In [23]:
```
non_cancelled_reservations_df['price_per_guest'] = non_cancelled_reservation
```

In [24]:
```
data.to_csv('Clean Hotel Booking', index=False)
```

Visualization

In [25]:
```
#Displaying some statistical summaries of the numerical columns:
data.describe()
```

Out[25]:

	lead_time	arrival_date_year	arrival_date_day_of_month	stays_in_weekend_nights
count	85347.000000	85347.000000	85347.000000	85347.000000
mean	80.626290	2016.215590	15.821552	1.017282
std	85.884725	0.684698	8.836634	1.026558
min	0.000000	2015.000000	1.000000	0.000000
25%	12.000000	2016.000000	8.000000	0.000000
50%	50.000000	2016.000000	16.000000	1.000000

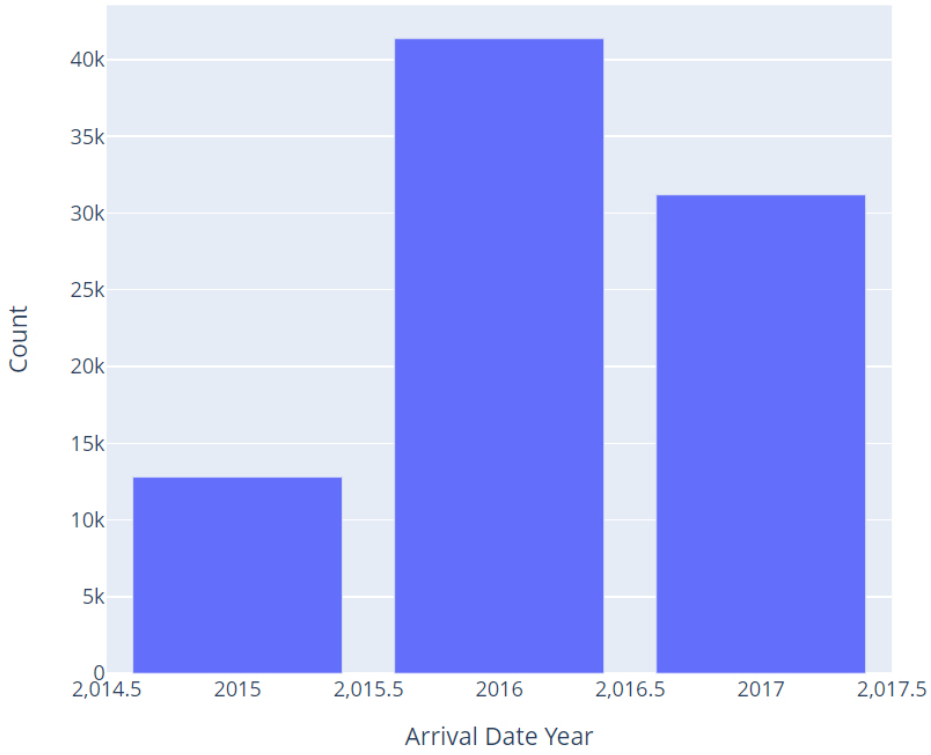
75%	**126.000000**	**2017.000000**	**23.000000**	**2.000000**
max	**709.000000**	**2017.000000**	**31.000000**	**19.000000**

In [26]:
```
non_cancelled_reservations_df['arrival_date_year'].unique()
```

Out[26]:
```
array([2015, 2016, 2017], dtype=int64)
```

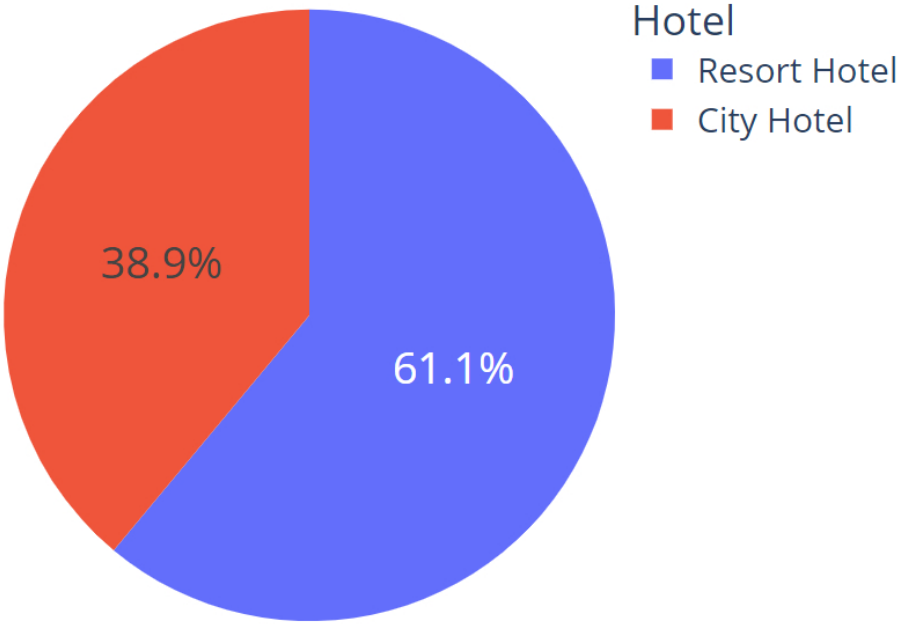
The dataset has data from the year 2015 to 2017.

In [27]:
```
fig = px.bar(data, x = data.arrival_date_year.value_counts().index,
             y = data['arrival_date_year'].value_counts().to_list(),
             labels = {'y' : 'Count',
                       'x' : 'Arrival Date Year'}
            )
fig.show()
```



Where Are More Cancelation?

In [28]:
```
fig =px.pie(data,
       values = data['hotel'].value_counts(),
       names = data['hotel'].unique(),
       )
fig.update_layout(
    legend_title="Hotel",
    font=dict(
        size=20
    )
)
fig.layout.template = 'plotly'
fig.update_traces(textposition='inside',textfont_size=25)#, textinfo='percen
fig.show()
```

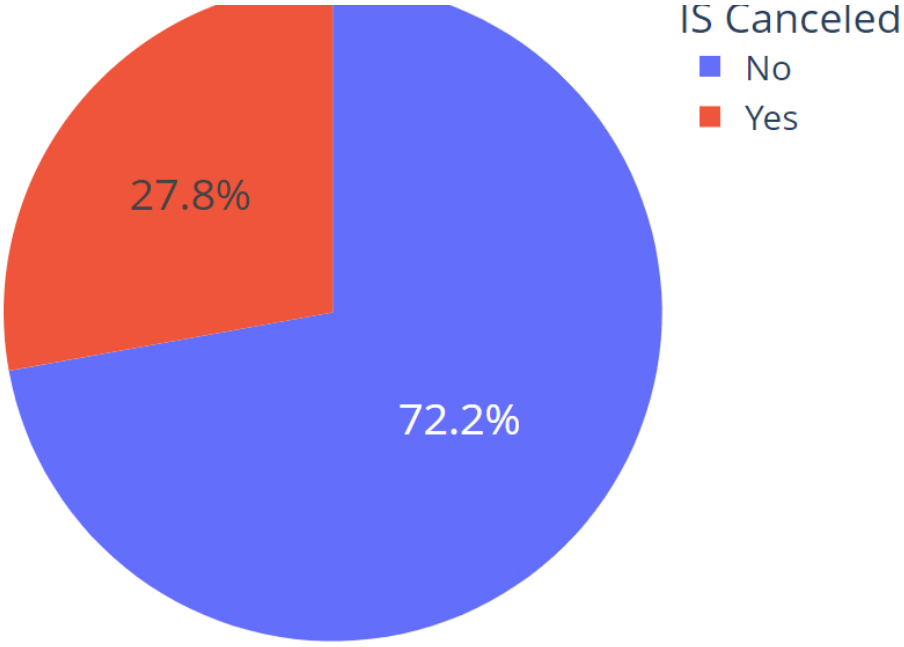


It appears that City Hotel has more reservations than Resort Hotel .

Is Canceled Column

In [29]:
```
#fig = px.bar(data, x = data.is_canceled.value_counts().index,
              #y = data['is_canceled'].value_counts().to_list(),
              #labels = {'y' : 'Count',
                         #'x' : 'IS Canceled'}
             #)
#fig.show()
```

In [30]:
```
fig =px.pie(data,
       values = data['is_canceled'].value_counts(),
       names = data['is_canceled'].unique(),
       )
fig.update_layout(
    title = 'Is The Reservation Cancelled ?',
    legend_title="IS Canceled",
    font=dict(
        size=20
    )
)
fig.layout.template = 'plotly'
fig.update_traces(textposition='inside',textfont_size=25)#, textinfo='percen
fig.show()
```



It seems like close to 28% of the reservations were canceled.

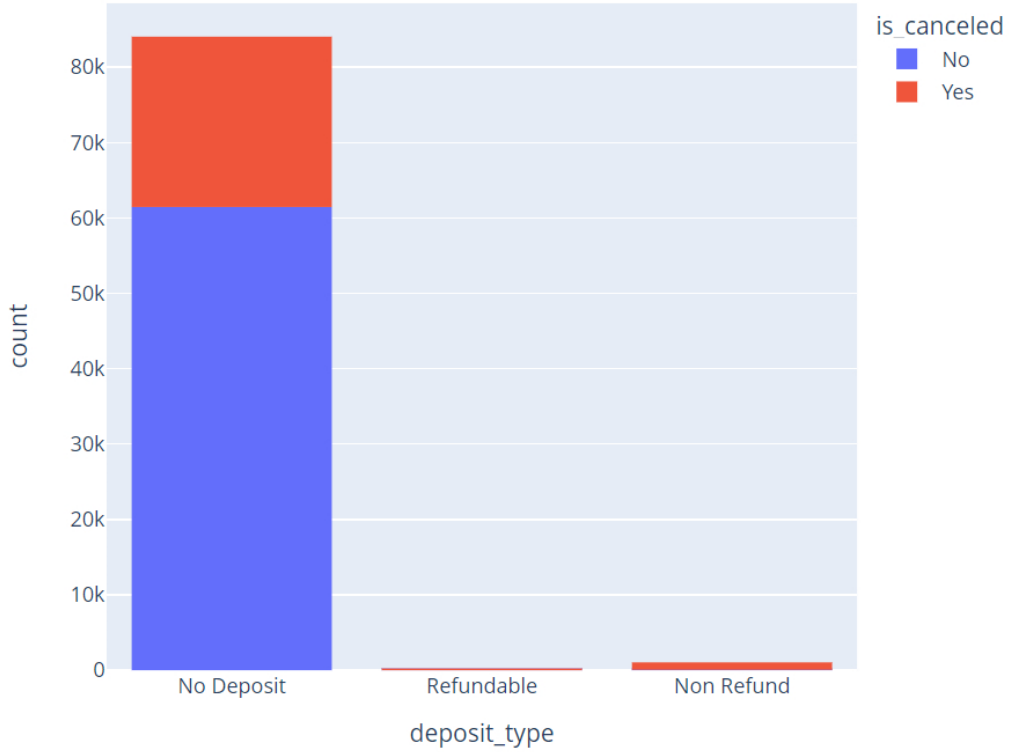
IS Canceled VS. Hotel

In [31]:
```
1 px.histogram (data, x = "hotel", color = 'is_canceled')
```

The higher the percentage of reservations is , the higher the percentage of cancellation .

IS Canceled VS. Deposit Type

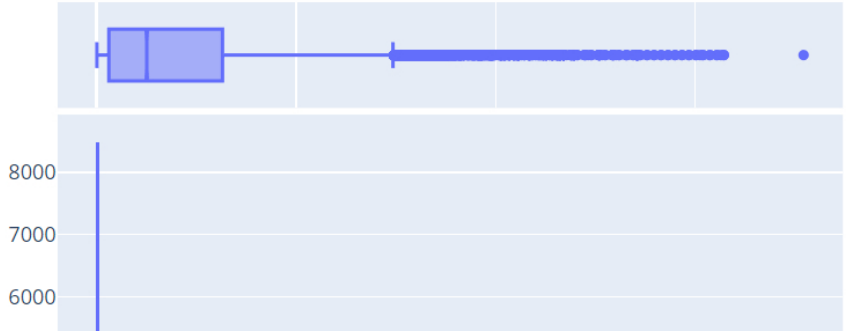
In [32]:
```
1 px.histogram(data,x='deposit_type', color='is_canceled')
```

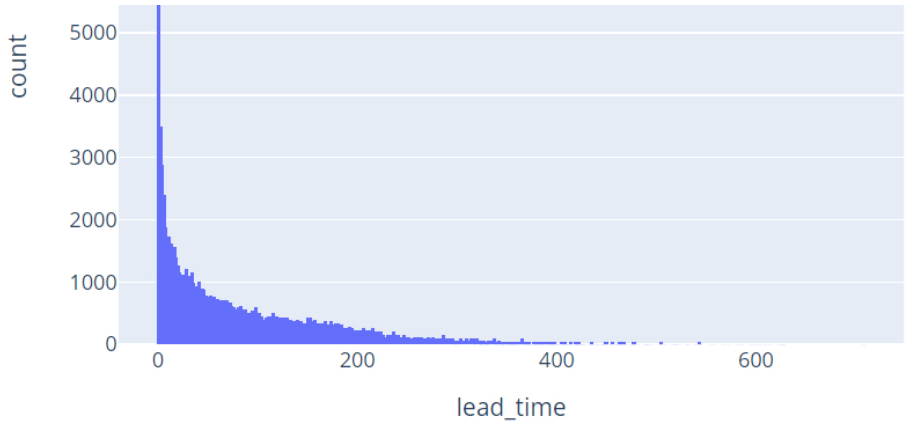


Most of the reservation do not require an initial deposit. It is surprising to see that all non-refundable reservations were canceled. It also looks like the hotel does not charge an initial deposit fee while the customers make their reservations.

Lead Time Column

In [33]:
```
1 px.histogram (data, x = "lead_time", nbins = 500, marginal = "box")
```

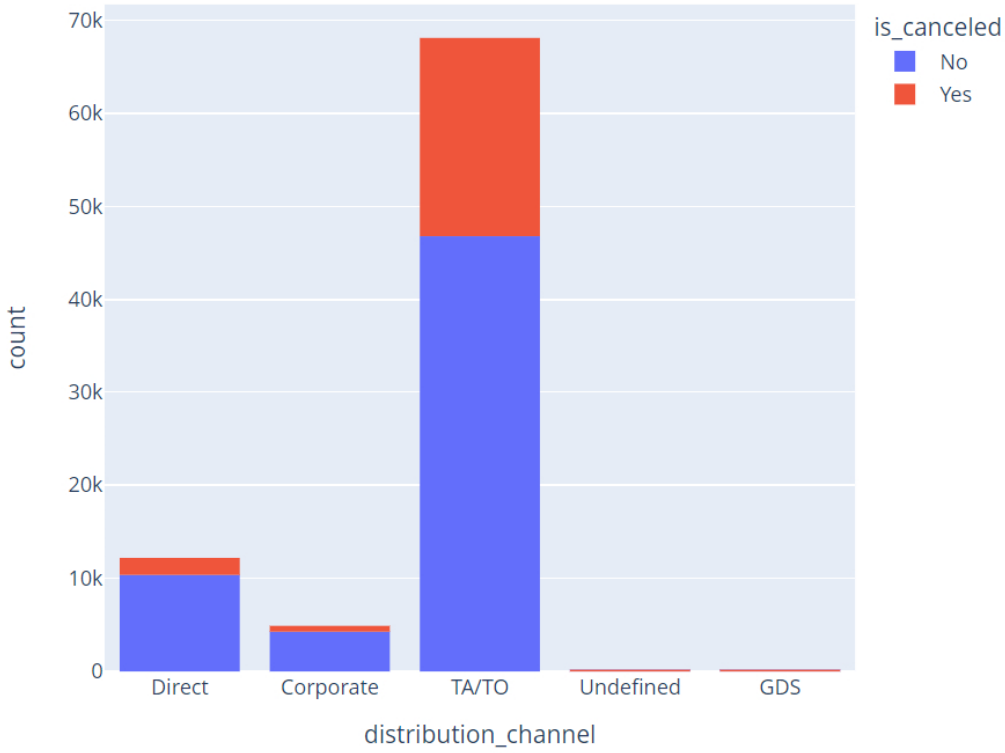




Most of the reservations are made in the system 0 to 1 day before the guests arrive in the hotel.

Distribution Channel Column

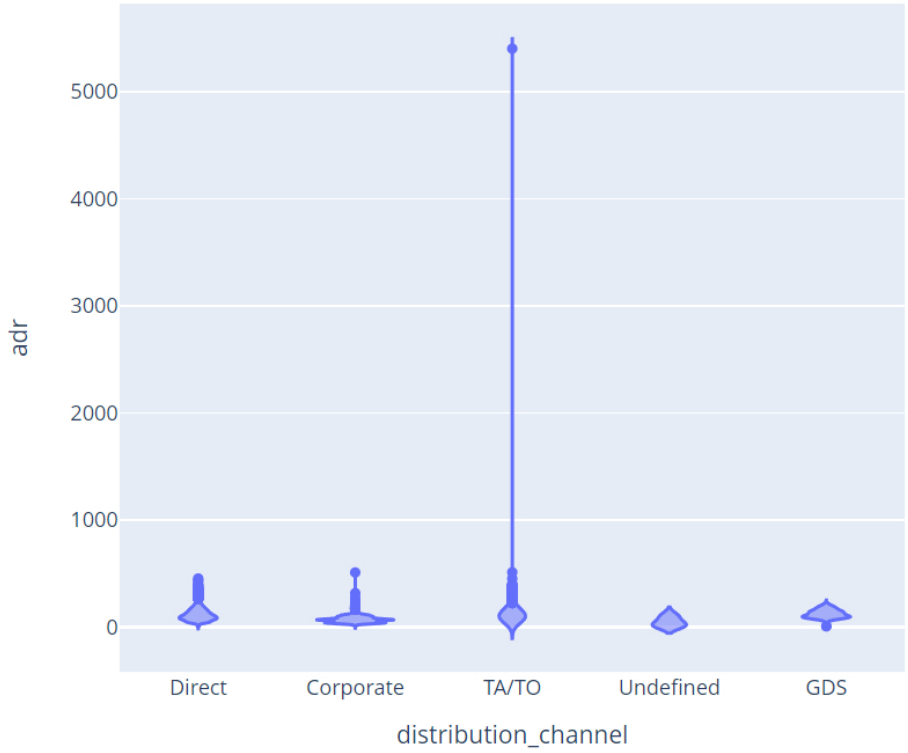
In [34]:
```
1 px.histogram(data, x = "distribution_channel", color = 'is_canceled')
```



Most of the bookings come from Travel Agents or Tour Operators. It will be intresting to see how much the customers coming from these channels pay daily.

ADR VS. Distribution Channel

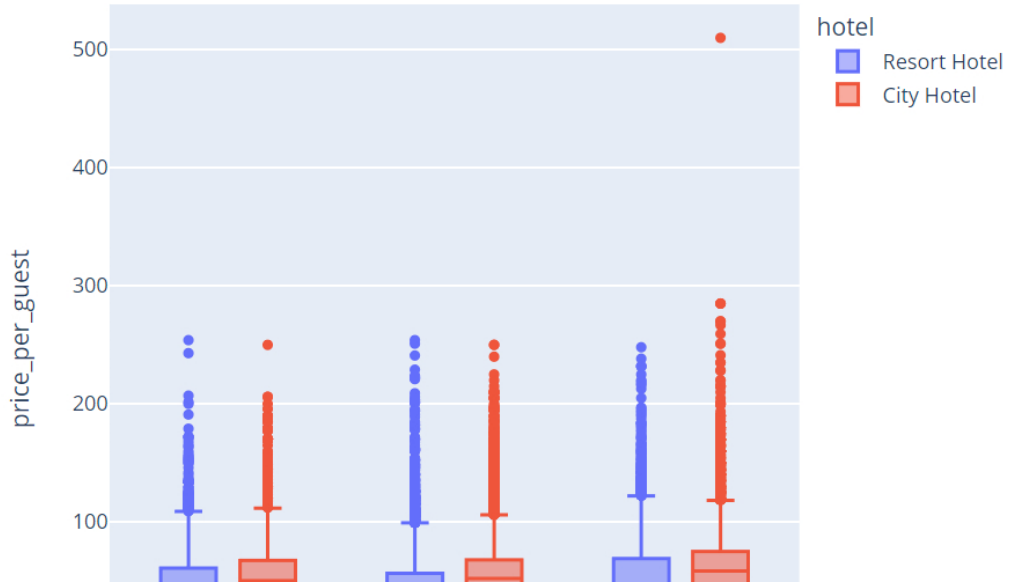
In [35]:
```
1 px.violin(data, x = "distribution_channel", y = "adr")
```



Customers who make Direct booking tend to pay a higher ADR compared to the customers from other channel. The reservation with ADR of 5400 for TA/TO clearly looks like an outlier.

Price Per Guest VS. Hotel Type

In [36]:
```
1 px.box(non_cancelled_reservations_df, x = "arrival_date_year", y = "price_pe
```



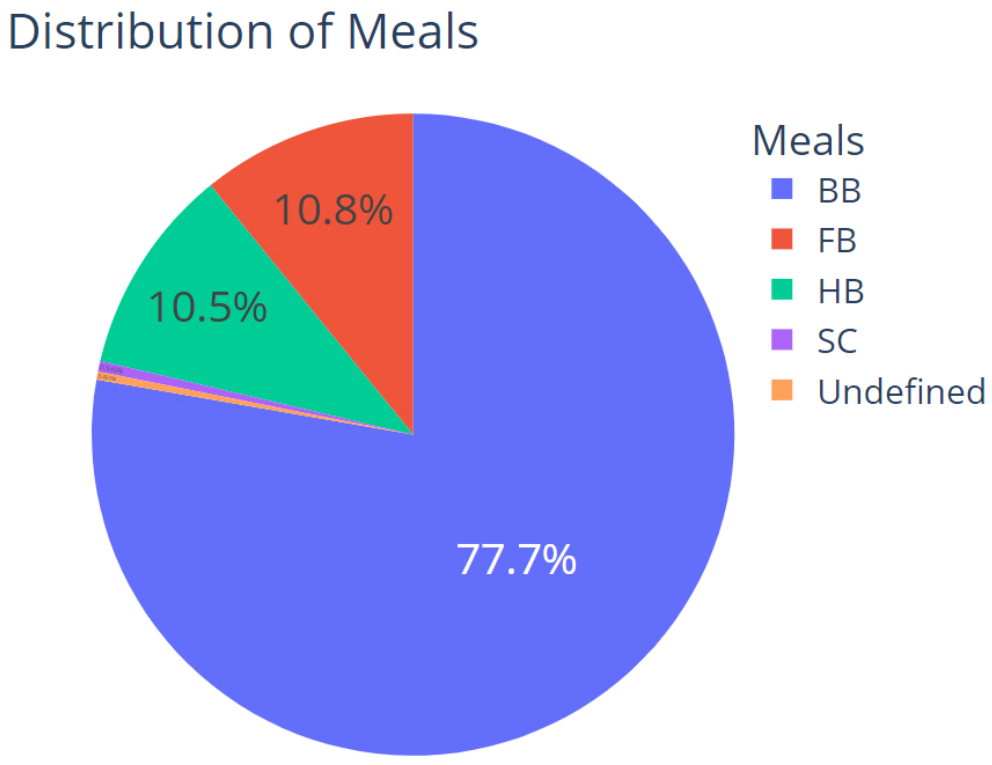
`City Hotel` ***is costlier than*** `Resort Hotel` ***and it looks like the*** `price per guest` ***of City Hotel is increasing over the years.***

Meal Column

In [37]:
```
#fig = px.bar(data, x = data.meal.value_counts().index,
            # y = data['meal'].value_counts().to_list(),
            # labels = {'y' : 'Count',
                        #'x' : 'Meals'})

#fig.show()
```

In [38]:
```
fig =px.pie(data,
       values = data['meal'].value_counts(),
       names = data['meal'].unique(),
       )
fig.update_layout(
    title = 'Distribution of Meals',
    legend_title="Meals",
    font=dict(
        size=20
    )
)
fig.layout.template = 'plotly'
fig.update_traces(textposition='inside',textfont_size=25)#, textinfo='percen
fig.show()
```



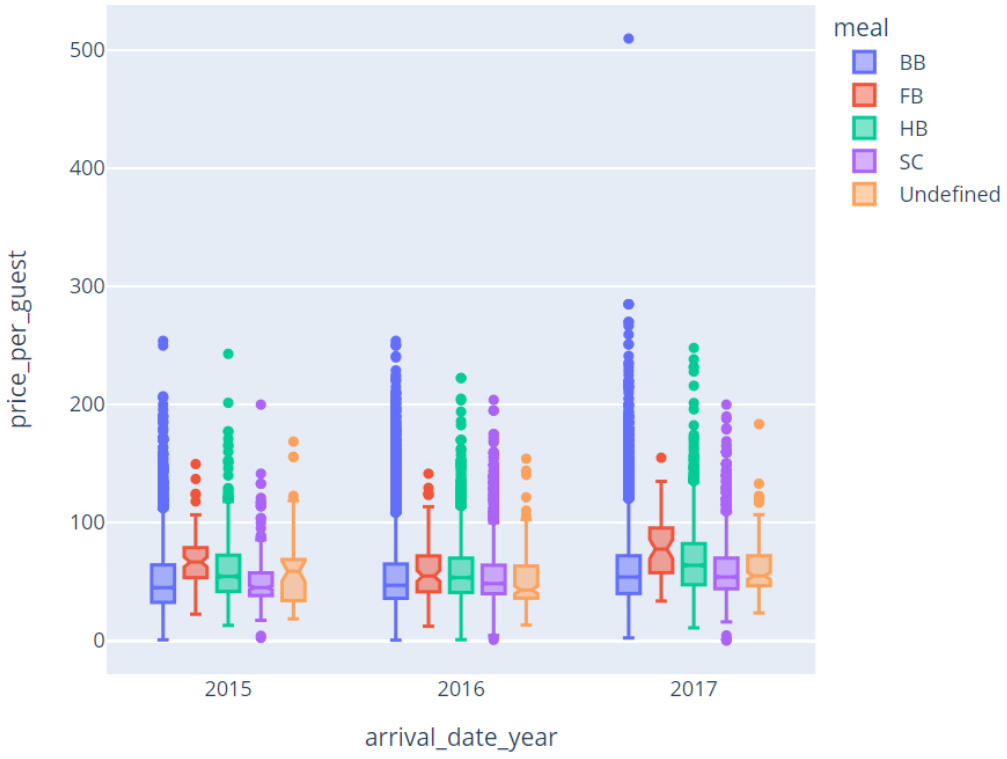
The categories in the meal column mean:

- ***BB: Bed & Breakfast***
- ***HB:: Half board (breakfast and one other meal – usually dinner)***
- ***FB: Full board (breakfast, lunch and dinner)***
- ***Undefined/SC: no meal package***

Bed & Breakfast is the most common option among the customers.

Meal VS. Price Per Guest

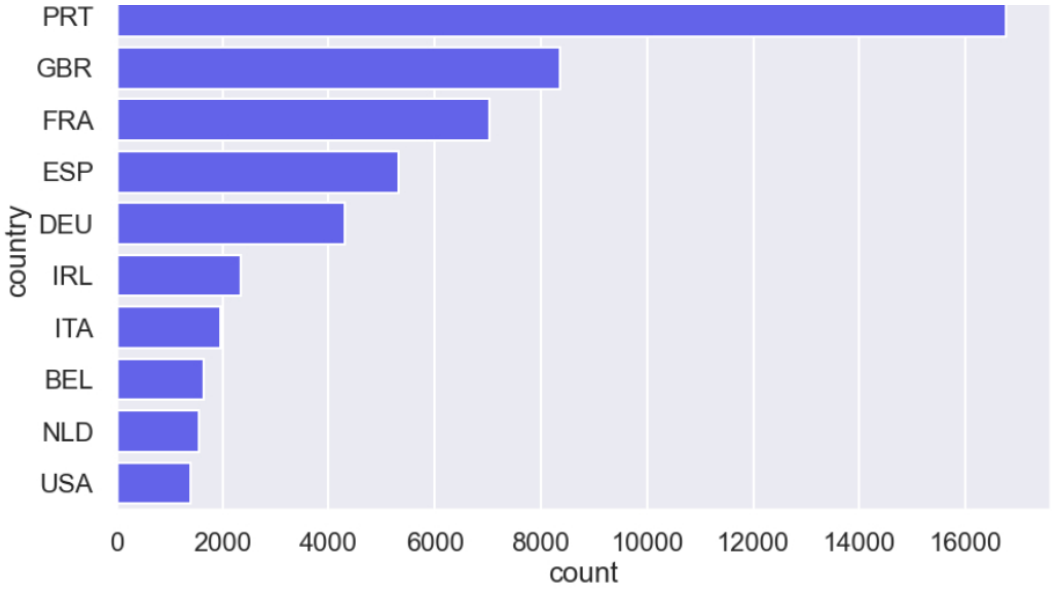
In [39]:
```
px.box(non_cancelled_reservations_df, x = "arrival_date_year", y = "price_pe
```



Where do the guests come from?

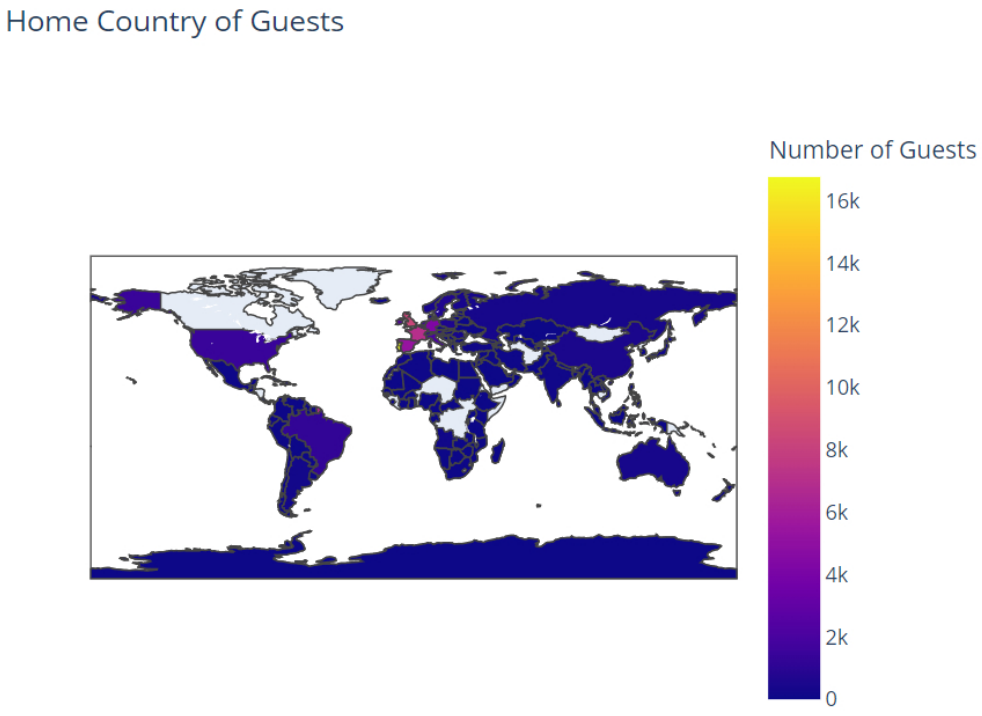
In [40]:
```
country_counts = non_cancelled_reservations_df['country'].value_counts()
country_counts_df = pd.DataFrame({'Country' : country_counts.index,
                                  'Count' : country_counts.values})
```

In [41]:
```
plt.figure(figsize=(14,8))
sns.countplot(data=non_cancelled_reservations_df, y='country', order=country
plt.title('Country', fontsize=30);
```



In [42]:

```
px.choropleth(
    country_counts_df,
    locations = 'Country',
    color = 'Count',
    labels={'Count':'Number of Guests'},
    title = 'Home Country of Guests'
    )
```



The guests come from all over the world but we can see that the majority of the guests come from Portugal and other parts of Europe.

How long do the guests stay?

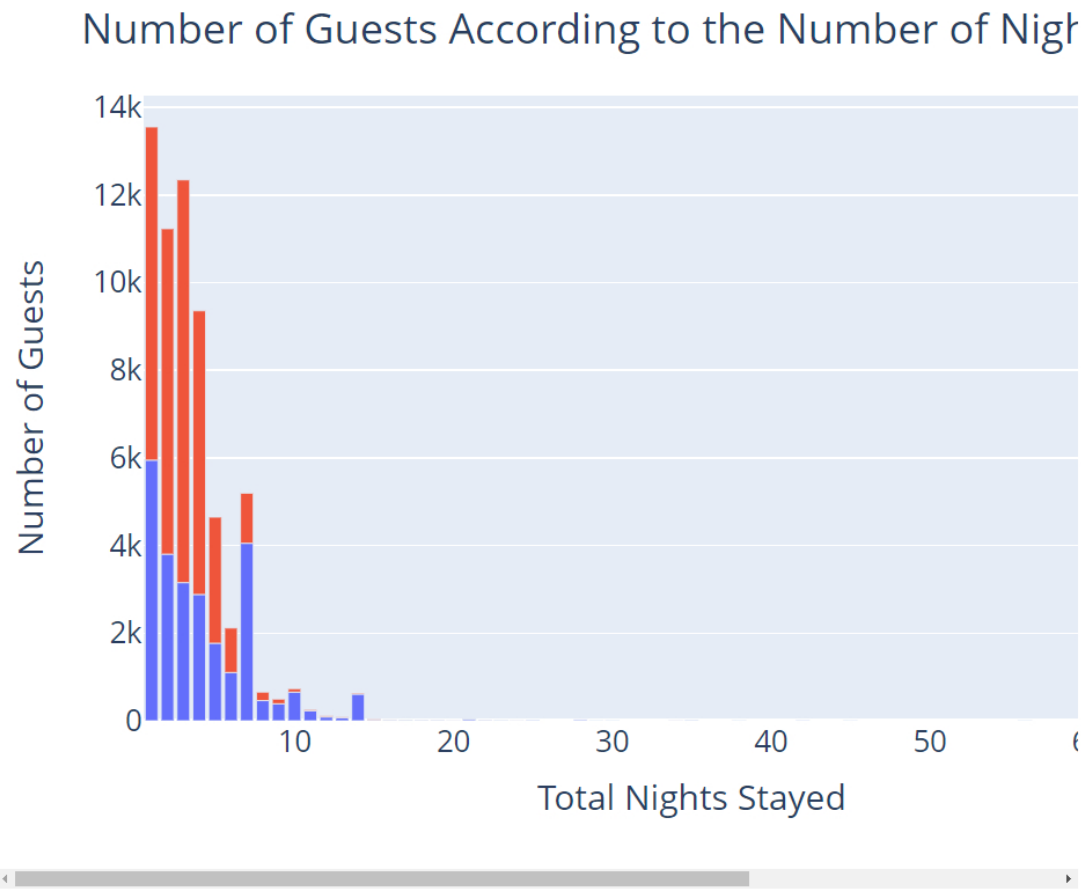
In [43]:
```
non_cancelled_reservations_df['total_nights_stays'] = non_cancelled_reservat
```

In [44]:
```
resort_hotel_df = non_cancelled_reservations_df[non_cancelled_reservations_d
resort_hotel_stays_count = resort_hotel_df['total_nights_stays'].value_count
resort_hotel_stays_df = pd.DataFrame( {'hotel': 'Resort Hotel','total_nights
```

In [45]:
```
city_hotel_df = non_cancelled_reservations_df[non_cancelled_reservations_df[
city_hotel_stays_count = city_hotel_df['total_nights_stays'].value_counts()
city_hotel_stays_df = pd.DataFrame( {'hotel': 'City Hotel','total_nights_sta
```

In [46]:
```
stays_df = resort_hotel_stays_df.append(city_hotel_stays_df)
```

In [47]:
```
fig = px.bar(stays_df, x='total_nights_stays',y='count', color ='hotel',widt
fig.update_layout(
    title='Number of Guests According to the Number of Nights Booked',
    xaxis_title="Total Nights Stayed",
    yaxis_title="Number of Guests",
    font=dict(
        size=17
    )
)
fig.layout.template = 'plotly'
fig.show()
```



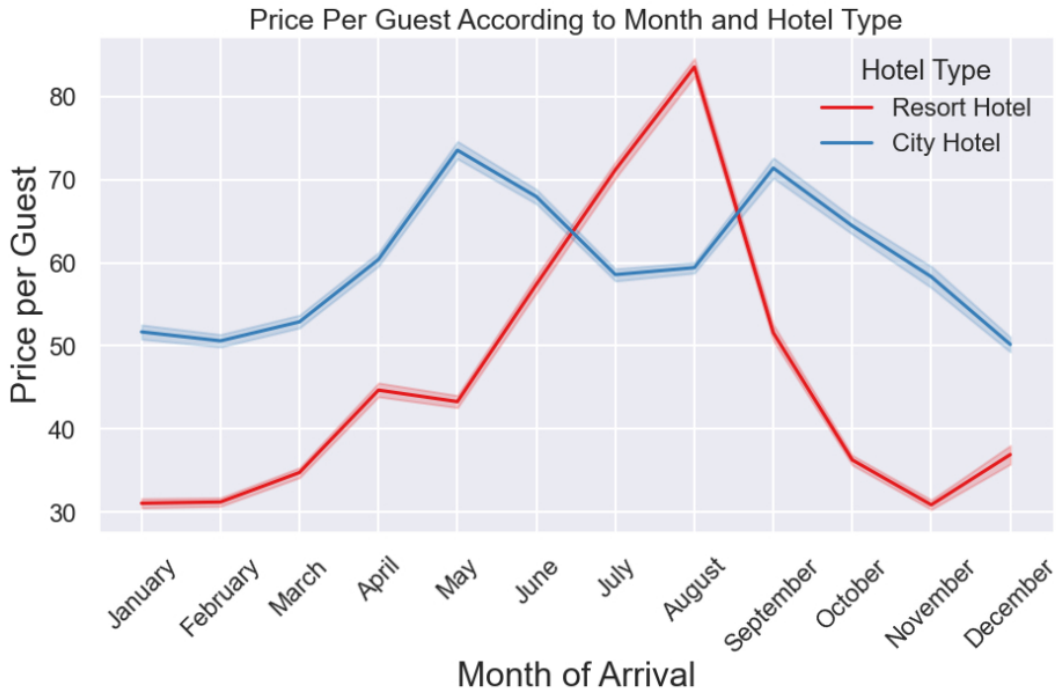
On an average most of the guests stay for around 1 to 4 nights.

What is the cost of stay for a guest for one night

In [48]:
```
non_cancelled_reservations_df['arrival_date_month'] = pd.Categorical(non_car
                                                                     categor
                                                                     ordered
```

In [49]:
```
sns.set_palette("Set1")
```

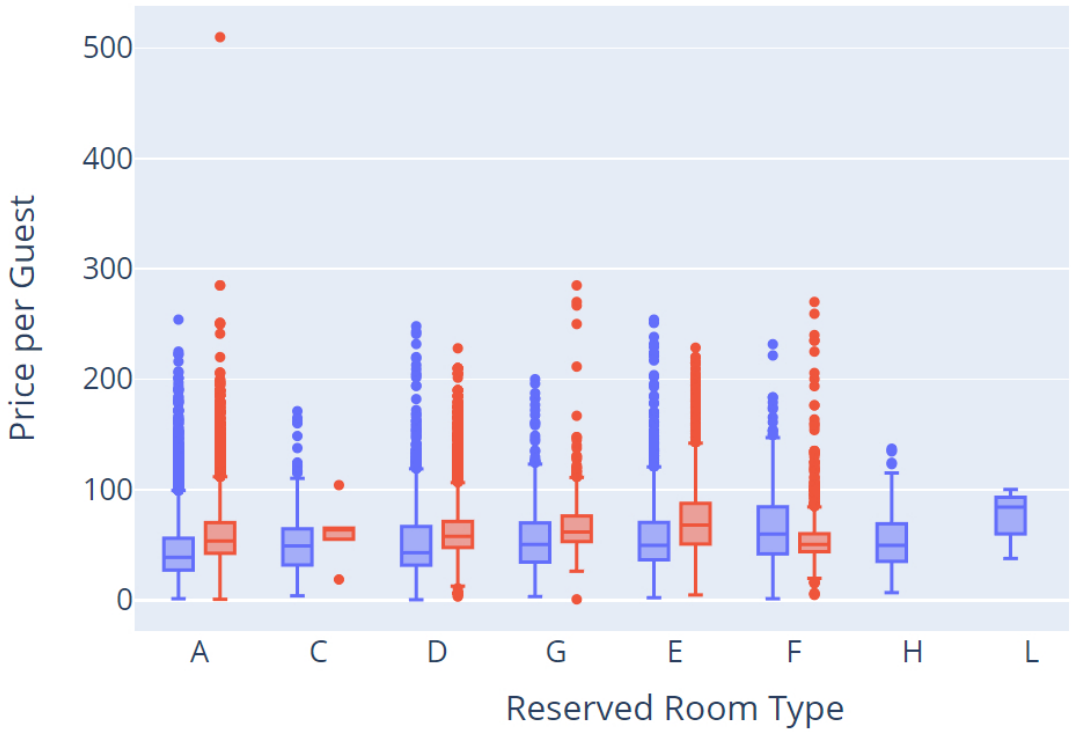
In [50]:
```
plt.figure(figsize = (15,8))
fig = sns.lineplot(x = 'arrival_date_month',
                   y = 'price_per_guest',
                   hue = 'hotel',
                   data = non_cancelled_reservations_df,
                  )
plt.xlabel('Month of Arrival', fontsize = 30)
plt.xticks(rotation=45)
plt.ylabel('Price per Guest', fontsize = 30)
plt.legend(title = 'Hotel Type')
plt.title('Price Per Guest According to Month and Hotel Type', fontsize = 25
```



The Resort Hotel is most expensive for the Summer months, on the other hand the City Hotel is pricey in Spring and Autumn months.

In [51]:
```
fig = px.box(non_cancelled_reservations_df,
             x = 'reserved_room_type',
             y = 'price_per_guest',
             color = 'hotel',width=870, height=500)
fig.update_layout(
    title='Price Distribution According to Room Type',
    xaxis_title="Reserved Room Type",
    yaxis_title="Price per Guest",
    legend_title="Hotel Type",
    font=dict(
        size=17
    )
)
fig.layout.template = 'plotly'
fig.show()
```

Price Distribution According to Room Type



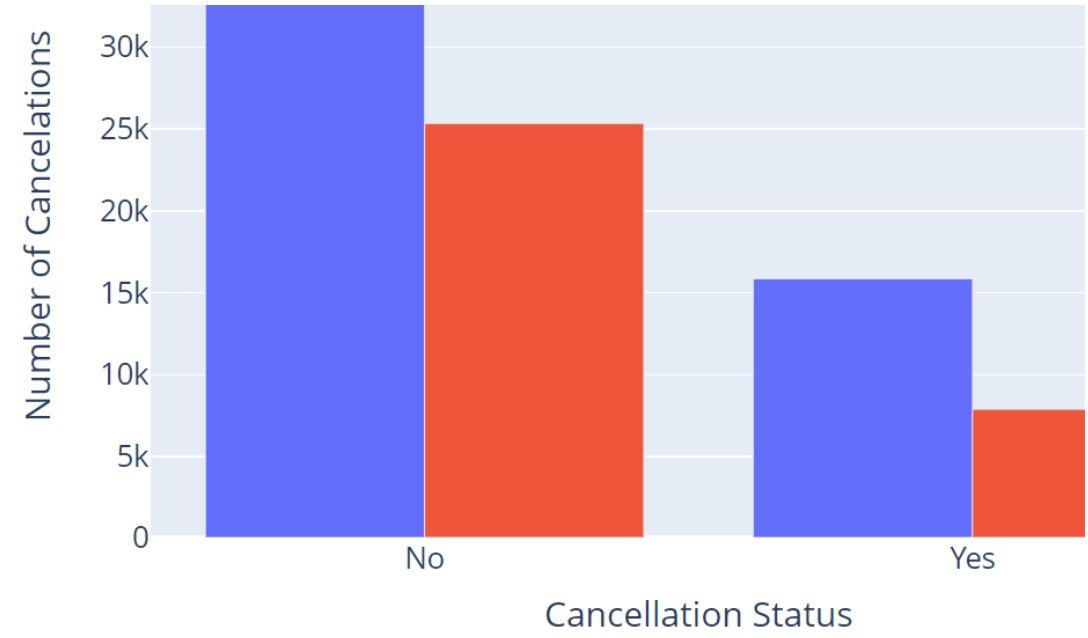
It is clear that the price per guest greatly depends on the reserved room type. The price per room type is also more expensive for City Hotel than Resort Hotel.

How many bookings were cancelled?

In [52]:
```
city_hotel_cancellation = data[data['hotel'] == 'City Hotel']
city_hotel_cancellation_count = city_hotel_cancellation['is_canceled'].value
city_hotel_cancellation_df = pd.DataFrame( {'hotel': 'City Hotel','is_cancel

resort_hotel_cancellation = data[data['hotel'] == 'Resort Hotel']
resort_hotel_cancellation_count = resort_hotel_cancellation['is_canceled'].v
resort_hotel_cancellation_df = pd.DataFrame( {'hotel': 'Resort Hotel','is_ca
cancellation_df = city_hotel_cancellation_df.append(resort_hotel_cancellatic
```

In [53]:
```
fig = px.bar(cancellation_df, x='is_canceled',y='count', color ='hotel',barm
fig.update_layout(
    title='Number of Cancelations According to Hotel Type',
    xaxis_title="Cancellation Status",
    yaxis_title="Number of Cancelations",
    legend_title="Hotel Type",
    font=dict(
        size=17
    )
)
fig.layout.template = 'plotly'
fig.show()
```

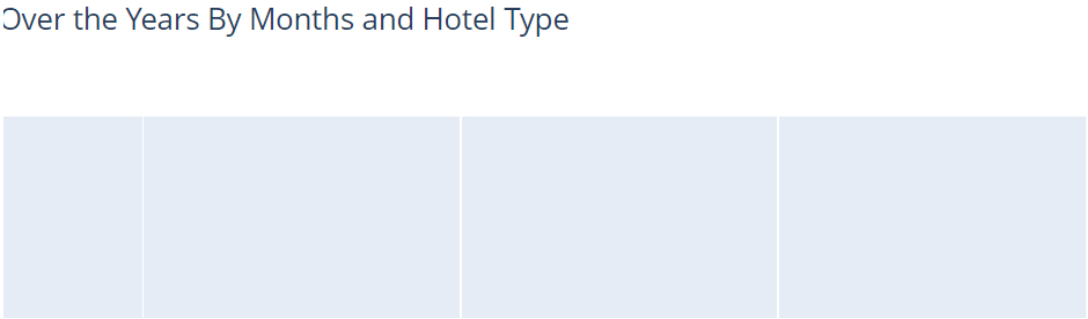


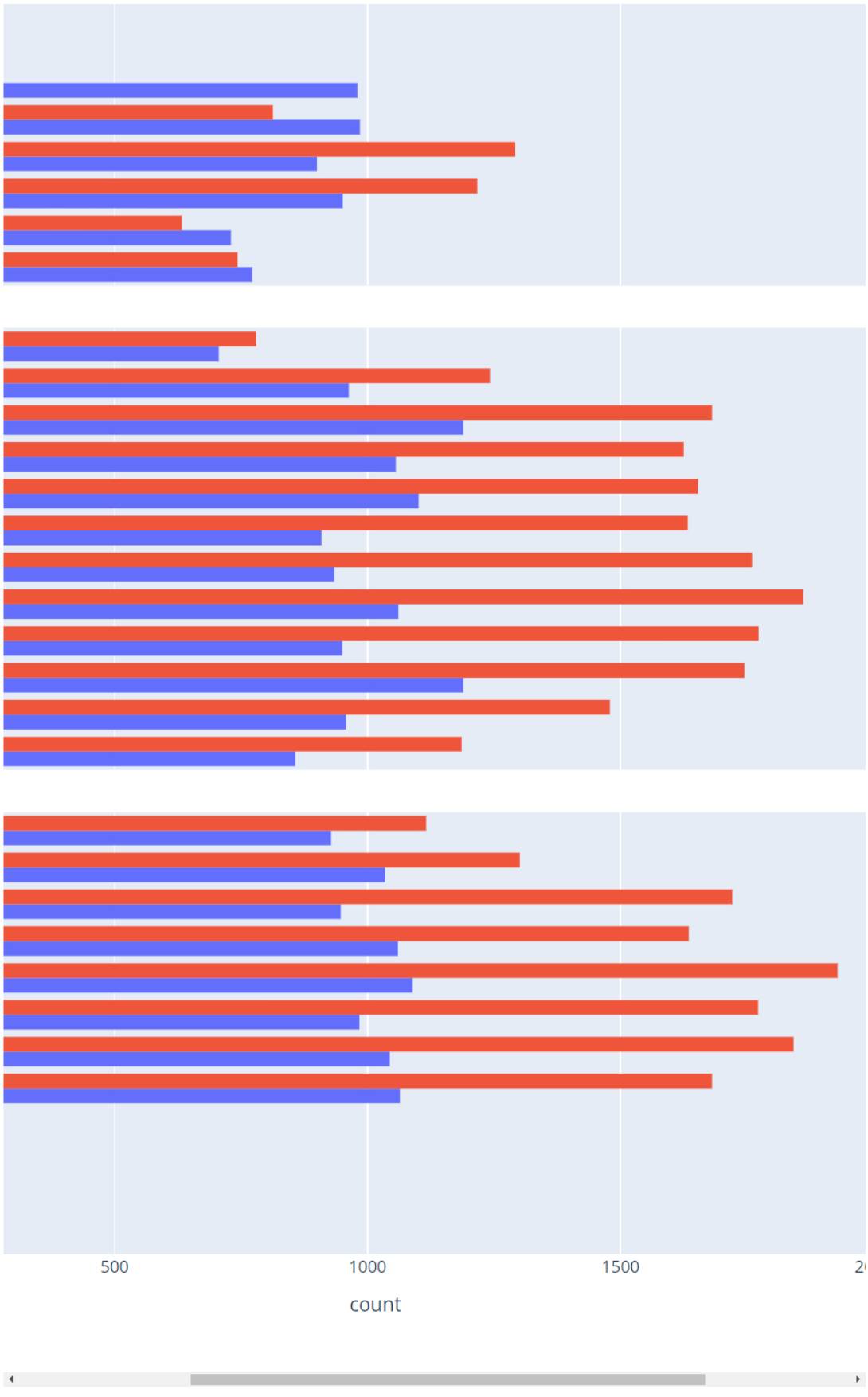
`City Hotel` ***has more number of cancellations than*** `Resort Hotel` ***and this is making sense because it has a greater number of bookings.***

What is the busiest month?

In [54]:
```
non_cancelled_reservations_df['arrival_date_month'] = pd.Categorical(non_car
                                                                     categor
                                                                     ordered
```

In [55]:
```
fig = px.histogram(non_cancelled_reservations_df,
             y = 'arrival_date_month',
             color = 'hotel',
             facet_row = 'arrival_date_year',
             category_orders = {'arrival_date_month' : ['January', 'February
             barmode = 'group',
             height = 1200, width = 1000,
             labels = {
                 'arrival_date_month' : 'Month of Arrival',
                 'count' : 'Number of Guests',
                 'arrival_date_year' : 'Year of Arrival',
                 'hotel' : 'Hotel Type'
             },
             title = "Number of Guests Over the Years By Months and Hotel Ty
                  )
fig.show()
```



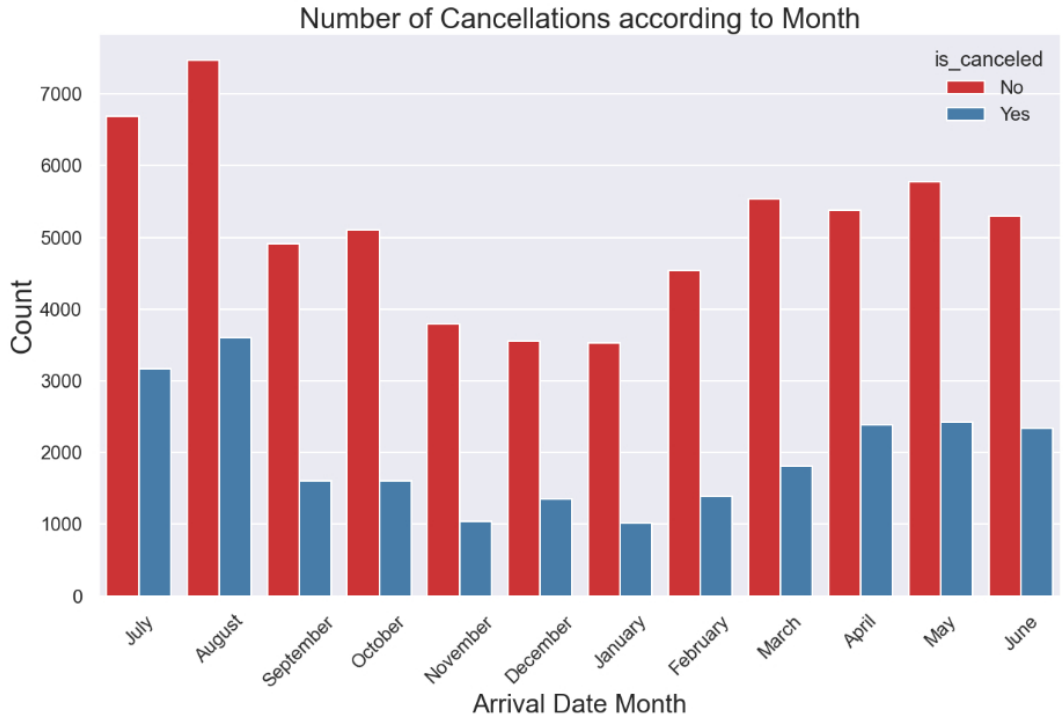


The dataset has data from the year 2015 to 2017 with months of data missing for year 2015 and 2017. It seems like August has the highest number of bookings followed by July with second highest bookings if we do not take an account of the missing data. Overall, I feel May should have the most number of booking for the three years.

It also clear that the number of bookings for City Hotel is more than the number of

booking for Resort Hotel.

In [65]:
```
sns.countplot(data = data,
              x = 'arrival_date_month',
              hue = 'is_canceled'
              )
plt.xlabel('Arrival Date Month', fontsize = 30)
plt.xticks(rotation=45)
plt.ylabel('Count', fontsize = 33)
plt.title('Number of Cancellations according to Month', fontsize = 33);
```



August and July have the most cancellations, but we need to keep in mind that those are the only two months which have data for all the three years. I believe, cancellations for the month of April, May, and June would be the most if data for all three years are compared.

Which room type is most popular?

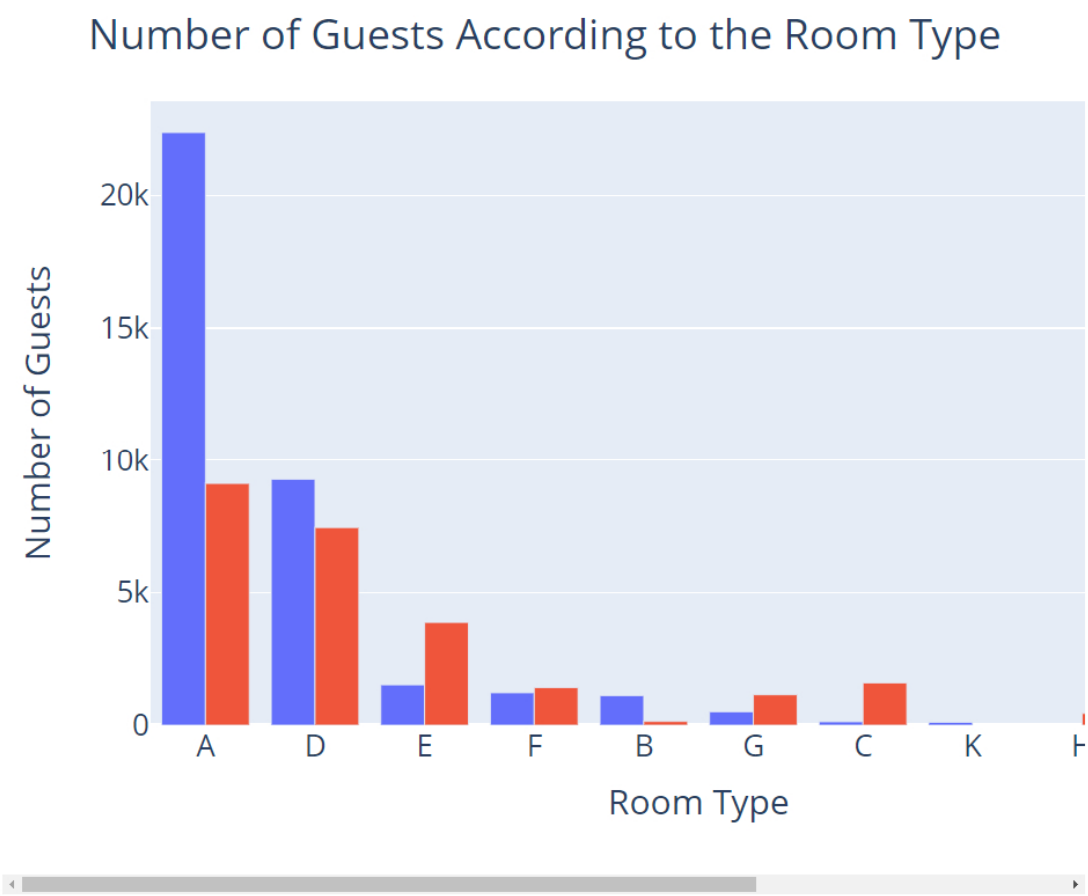
In [57]:
```
city_hotel_room_type = city_hotel_df['assigned_room_type'].value_counts()
city_hotel_room_type_df = pd.DataFrame({ 'hotel' : 'City Hotel', 'room_type'

resort_hotel_room_type = resort_hotel_df['assigned_room_type'].value_counts(
resort_hotel_room_type_df = pd.DataFrame({ 'hotel' : 'Resort Hotel', 'room_t
```

In [58]:
```
room_type_df = city_hotel_room_type_df.append(resort_hotel_room_type_df)
```

In [59]:
```
fig = px.bar(room_type_df, x='room_type',y='count', color ='hotel',barmode="
fig.update_layout(
    title='Number of Guests According to the Room Type',
    xaxis_title="Room Type",
    yaxis_title="Number of Guests",
    legend_title="Hotel Type",
    font=dict(
        size=17
    )
)
fig.layout.template = 'plotly'
```

```
fig.show()
```

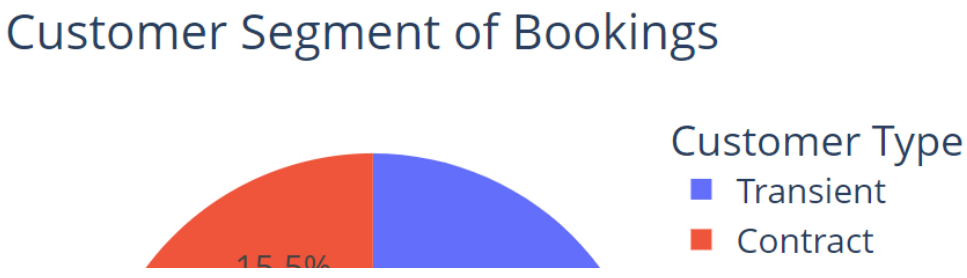


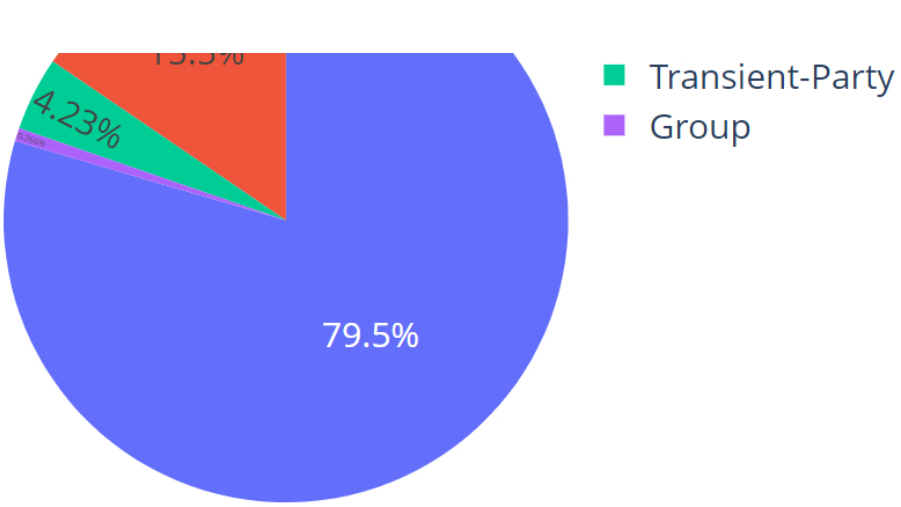
Room Type A for City Hotel seems to be the most popular choice of guests. This can be due to the fact that City Hotel type A has the lowest price per guest.

What is the customer type of the bookings?

In [60]:

```
fig =px.pie(non_cancelled_reservations_df,
       values = non_cancelled_reservations_df['customer_type'].value_counts(
       names = non_cancelled_reservations_df['customer_type'].unique(),
       )
fig.update_layout(
    title = 'Customer Segment of Bookings',
    legend_title="Customer Type",
    font=dict(
        size=20
    )
)
fig.layout.template = 'plotly'
fig.update_traces(textposition='inside',textfont_size=20)#, textinfo='percen
fig.show()
```





Conclusion

- ***The dataset has hotel reservations for two types of hotel City Hotel and Resort Hotel. Although the dataset has reservation details from the year 2015 to 22017 there is missing data from January to June for the year 2015 and from September to December for the year 2017.Since the reservations for months July and August were only present in all the years from 2015-2017 it makes sense why we had the most number of bookings for just those months.***
- ***Over the years the number of reservations for City Hotel is more than the number of reservations for Resort Hotel for each month. The same goes for the number of cancellations. Also, The guests seem to stay for a longer duration in City Hotel than Resort Hotel.***
- ***The overall cancellation rate is around 28% . I believe requiring an initial deposit while booking the reservation might help lower this.***
- ***The season of the year plays a high factor in determining the price per guest for the two hotel types. Resort Hotel is more expensive for the Summer months, on the other hand the City Hotel is pricey in Spring and Autumn months.***
- ***Room Type A is most reserved type of room. I believe this is due to the fact that it the cheapest of all the rooms for City Hotels.***

```
In [ ]: 1
```