

In [1]:

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

In [2]:

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

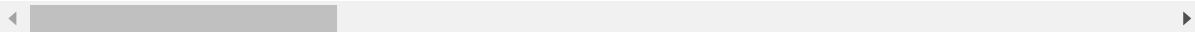
In [3]:

```
data = pd.read_csv(r'D:\Career\Udemy\DA 2\Hotel Booking\hotel_bookings.csv')
data.head()
```

Out[3]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_numb
0	Resort Hotel	0	342	2015	July	
1	Resort Hotel	0	737	2015	July	
2	Resort Hotel	0	7	2015	July	
3	Resort Hotel	0	13	2015	July	
4	Resort Hotel	0	14	2015	July	

5 rows × 32 columns



In [4]:

```
data.fillna(0,inplace = True)
```

In [5]:

```
data['meal'].value_counts()
```

Out[5]:

```
BB      92310
HB      14463
SC      10650
Undefined  1169
FB        798
Name: meal, dtype: int64
```

In [6]:

```
data['children'].unique()
```

Out[6]:

```
array([ 0.,  1.,  2., 10.,  3.])
```

In [7]:

```
data['adults'].unique()
```

Out[7]:

```
array([ 2,  1,  3,  4, 40, 26, 50, 27, 55,  0, 20,  6,  5, 10],
      dtype=int64)
```

In [8]:

```
data['babies'].unique()
```

Out[8]:

```
array([ 0,  1,  2, 10,  9], dtype=int64)
```

In [9]:

```
filter = (data['children'] == 0) & (data['adults'] == 0) & (data['babies'] == 0)
data = data[~filter]
```

Where do the guests come from?

In [10]:

```
resort = data[(data['hotel'] == 'Resort Hotel') & (data['is_canceled'] == 0)]
city = data[(data['hotel'] == 'City Hotel') & (data['is_canceled'] == 0)]
```

In [11]:

```
resort.head()
```

Out[11]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_num
0	Resort Hotel	0	342	2015	July	
1	Resort Hotel	0	737	2015	July	
2	Resort Hotel	0	7	2015	July	
3	Resort Hotel	0	13	2015	July	
4	Resort Hotel	0	14	2015	July	

5 rows × 32 columns

In [12]:

```
city.head()
```

Out[12]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_nu
40060	City Hotel	0	6	2015	July	
40066	City Hotel	0	3	2015	July	
40070	City Hotel	0	43	2015	July	
40071	City Hotel	0	43	2015	July	
40072	City Hotel	0	43	2015	July	

5 rows × 32 columns

In [13]:

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

Out[13]:

```
City Hotel      79163
Resort Hotel    40047
Name: hotel, dtype: int64
```

Spatial Analysis

In [14]:

```
import plotly.graph_objs as go
from plotly.offline import iplot
import plotly.express as px
```

In [15]:

```
resort['country'].value_counts()
```

Out[15]:

```
PRT    10184
GBR     5922
ESP     3105
IRL     1734
FRA     1399
```

...

```
ZMB         1
BIH         1
CYM         1
MKD         1
BHS         1
```

Name: country, Length: 119, dtype: int64

In [16]:

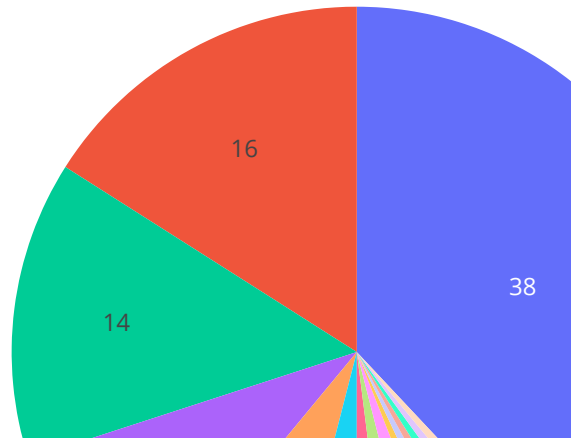
```
labels = resort['country'][0:100].value_counts().index
values = resort['country'][0:100].value_counts()
```

In [17]:

```
trace1 = go.Pie(labels = labels, values = values, hoverinfo = 'label+percent', textinfo = 'label+percent')
```

In [18]:

```
iplob([trace1])  
plt.figure(figsize = (15,15))
```



Out[18]:

&lt;Figure size 1080x1080 with 0 Axes&gt;

&lt;Figure size 1080x1080 with 0 Axes&gt;

In [19]:

```
country_wise_data = data[data['is_canceled'] == 0]['country'].value_counts().reset_index()  
country_wise_data.columns = ['Country', 'No of Guests']  
country_wise_data.head()
```

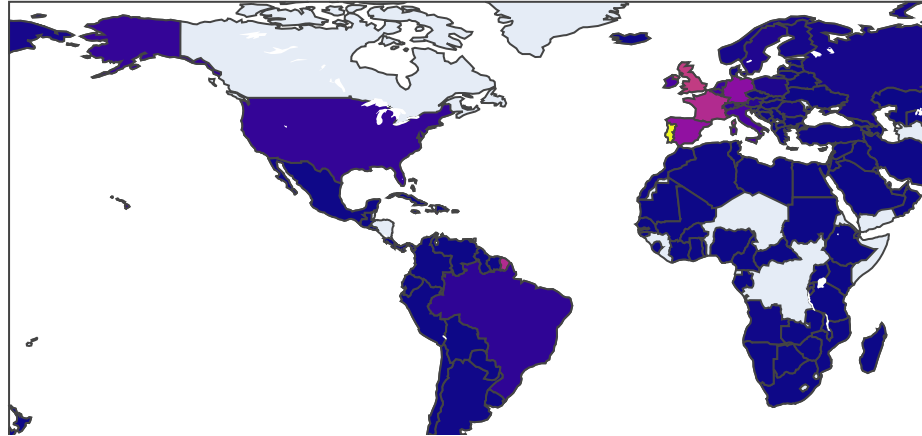
Out[19]:

	Country	No of Guests
0	PRT	20977
1	GBR	9668
2	FRA	8468
3	ESP	6383
4	DEU	6067

In [20]:

```
px.choropleth(country_wise_data,  
               locations = country_wise_data['Country'],  
               color = country_wise_data['No of Guests'],  
               hover_name = country_wise_data['Country'],  
               title = 'Home Country of Guests'  
            )
```

## Home Country of Guests



How much do guests pay for a room per night?

In [21]:

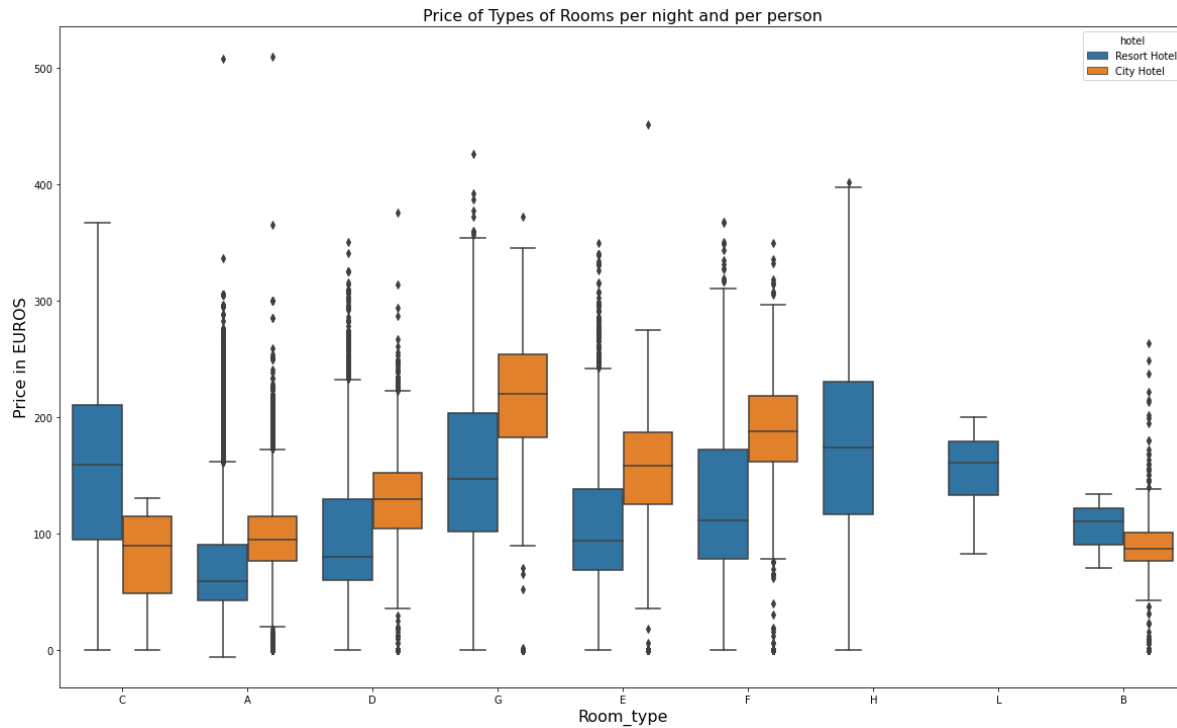
```
data2 = data[data['is_canceled'] == 0]
```

In [22]:

```
plt.figure(figsize = (20,12))
sns.boxplot(x = 'reserved_room_type', y = 'adr', data = data2, hue = 'hotel')
plt.title('Price of Types of Rooms per night and per person', fontsize = 16)
plt.xlabel('Room_type',fontsize = 16)
plt.ylabel('Price in EUROS',fontsize = 16)
```

Out[22]:

Text(0, 0.5, 'Price in EUROS')



Price per night vary over the year

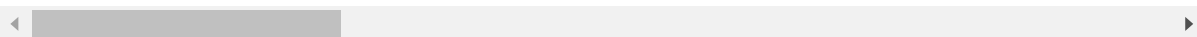
In [23]:

```
resort.head(2)
```

Out[23]:

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

2 rows × 32 columns



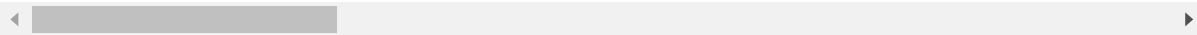
In [24]:

```
city.head(2)
```

Out[24]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_nu
40060	City Hotel	0	6	2015	July	
40066	City Hotel	0	3	2015	July	

2 rows × 32 columns



In [25]:

```
data_resort = resort[resort['is_canceled'] == 0]  
data_city = city[city['is_canceled'] == 0]
```

In [26]:

```
resort_hotel = data_resort.groupby('arrival_date_month')['adr'].mean().reset_index()  
city_hotel = data_city.groupby('arrival_date_month')['adr'].mean().reset_index()
```



In [27]:

```
final = resort_hotel.merge(city_hotel, on = 'arrival_date_month')
final.columns = ['Month', 'Price_per_resort', 'Price_per_city_hotel']
final.head()
```

Out[27]:

	Month	Price_per_resort	Price_per_city_hotel
0	April	75.867816	111.962267
1	August	181.205892	118.674598
2	December	68.410104	88.401855
3	February	54.147478	86.520062
4	January	48.761125	82.330983

In [28]:

```
import sort_dataframeby_monthorweek as sd
```

In [29]:

```
final2 = sd.Sort_Dataframeby_Month(final, 'Month')
final2.head()
```

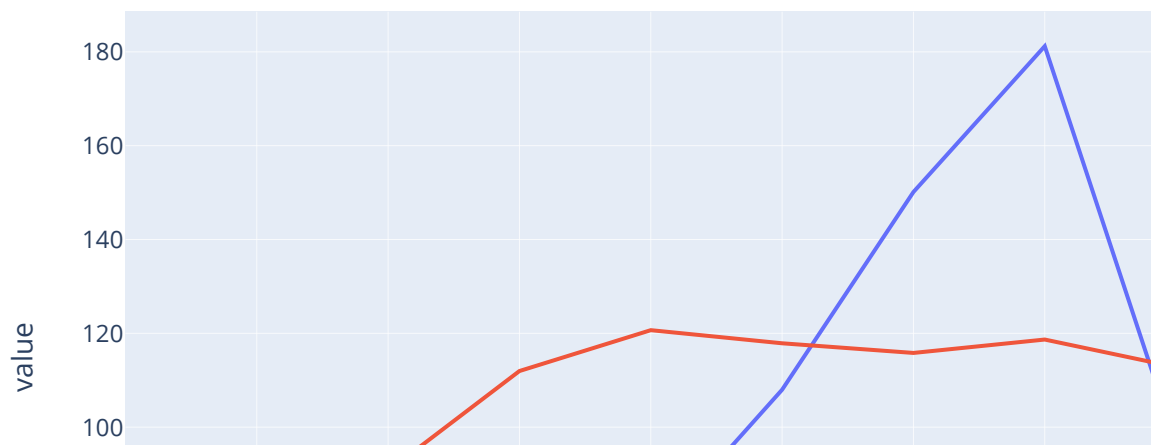
Out[29]:

	Month	Price_per_resort	Price_per_city_hotel
0	January	48.761125	82.330983
1	February	54.147478	86.520062
2	March	57.056838	90.658533
3	April	75.867816	111.962267
4	May	76.657558	120.669827

In [30]:

```
px.line(final2, x = 'Month', y = ['Price_per_resort', 'Price_per_city_hotel'], title = 'Room
```

Room price per night over the years



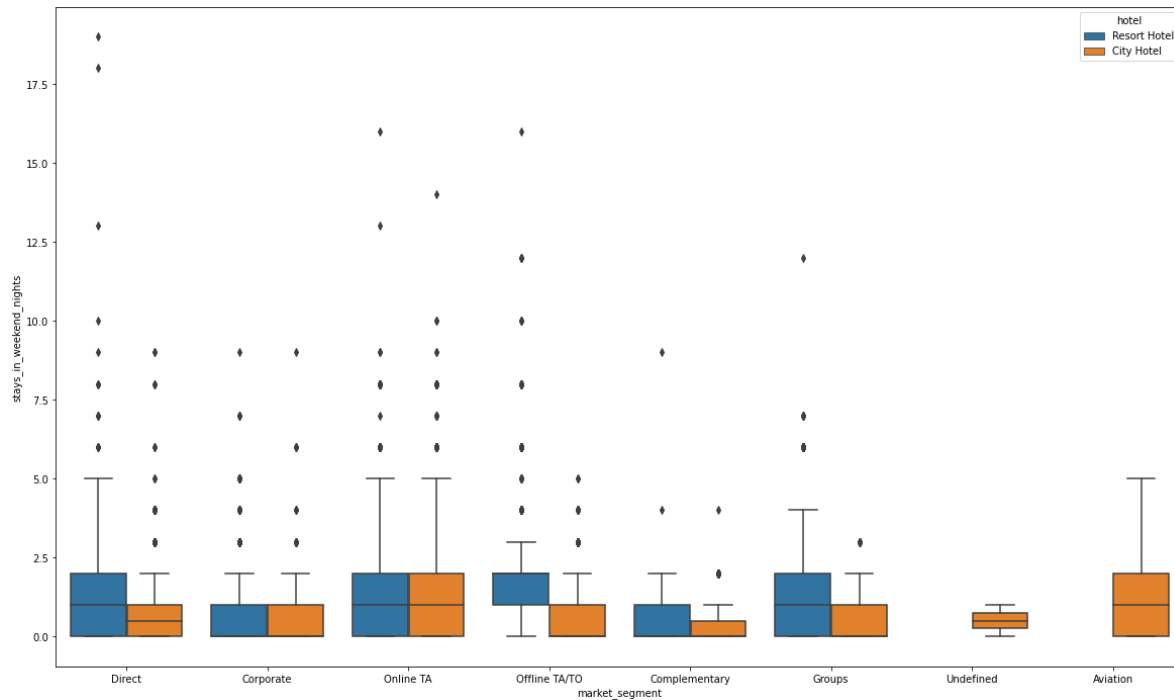
Distribution of nights spent at hotels by market segment and hotel type

In [31]:

```
plt.figure(figsize = (20,12))
sns.boxplot(x = 'market_segment', y = 'stays_in_weekend_nights', data = data, hue = 'hotel')
```

Out[31]:

```
<AxesSubplot:xlabel='market_segment', ylabel='stays_in_weekend_nights'>
```



## Preference of Guests

In [32]:

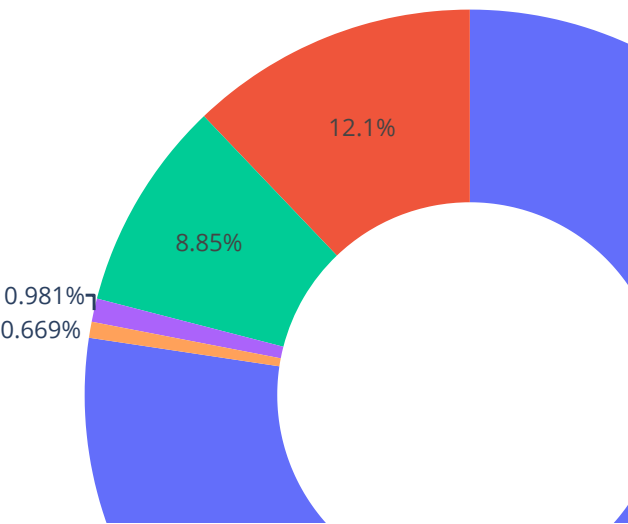
```
data['meal'].value_counts()
```

Out[32]:

```
BB          92236
HB          14458
SC          10549
Undefined    1169
FB           798
Name: meal, dtype: int64
```

In [33]:

```
px.pie(data, values = data['meal'].value_counts(), names = data['meal'].value_counts().inde
```



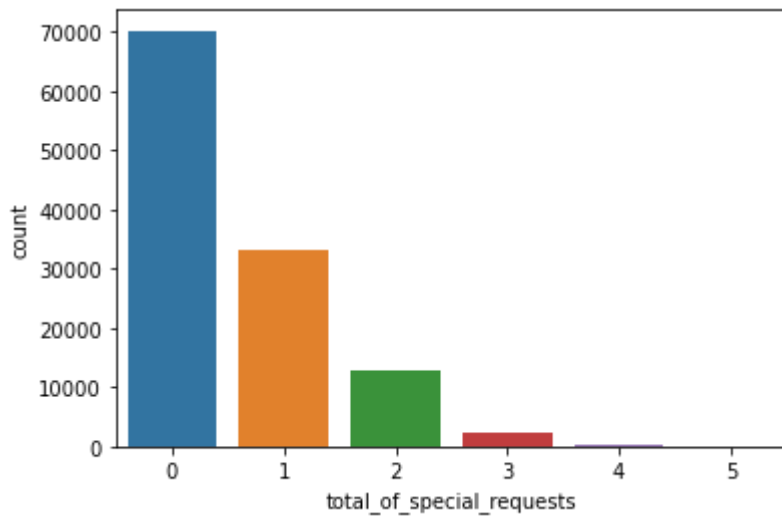
Special Request by Customers

In [34]:

```
sns.countplot(data['total_of_special_requests'])
```

Out[34]:

<AxesSubplot:xlabel='total\_of\_special\_requests', ylabel='count'>



Pivot table of relationship between Special requets and Cancellation of bookings

In [35]:

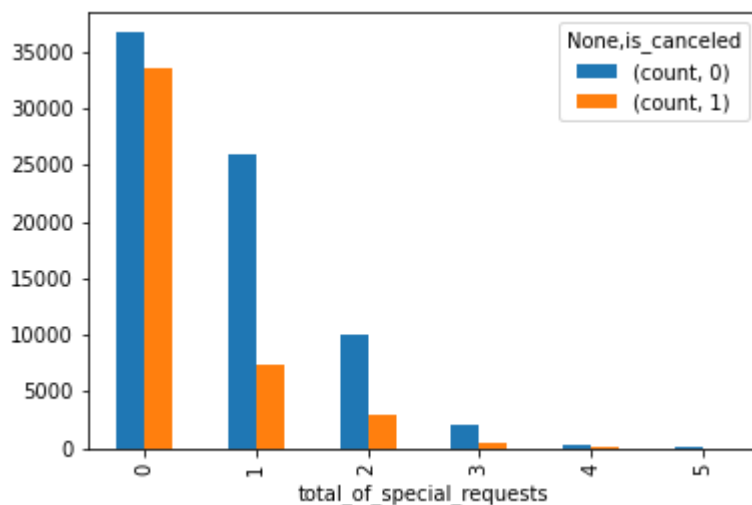
```
pivot = data.groupby(['total_of_special_requests', 'is_canceled']).agg({'total_of_special_re
```

In [36]:

```
pivot.plot(kind = 'bar')
```

Out[36]:

<AxesSubplot:xlabel='total\_of\_special\_requests'>



Most busy Month

In [37]:

```
rush_resort = data_resort['arrival_date_month'].value_counts().reset_index()
rush_resort.columns = ['Month', 'No of Guests']
rush_resort.head()
```

Out[37]:

	Month	No of Guests
0	August	3257
1	July	3137
2	October	2575
3	March	2571
4	April	2550

In [38]:

```
rush_city = data_city['arrival_date_month'].value_counts().reset_index()
rush_city.columns = ['Month', 'No of Guests']
rush_city.head()
```

Out[38]:

	Month	No of Guests
0	August	5367
1	July	4770
2	May	4568
3	June	4358
4	October	4326

In [39]:

```
final_rush = rush_resort.merge(rush_city, on = 'Month')
final_rush.columns = ['Month', 'No of Guests in Resort Hotel', 'No of Guests in City Hotel']
```

In [40]:

```
finale = sd.Sort_Dataframeby_Month(final_rush, 'Month')  
finale.head()
```

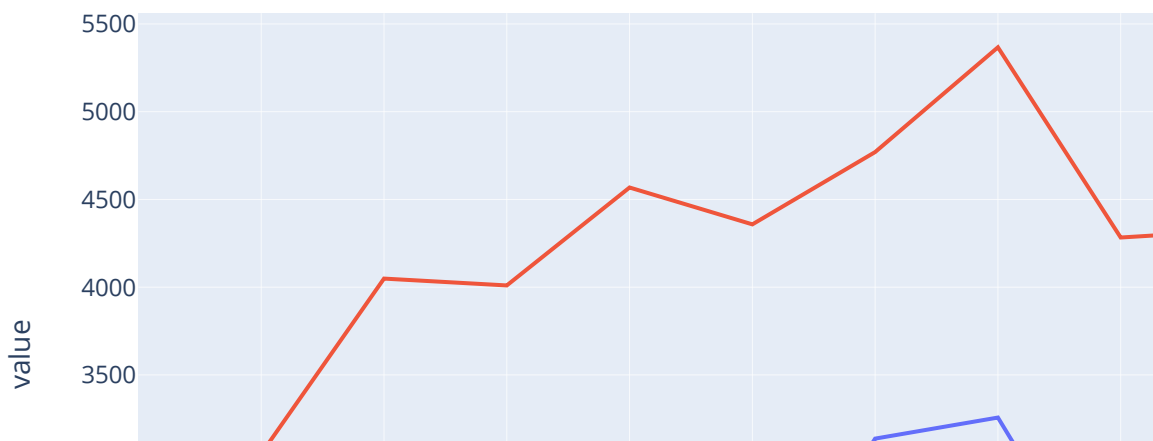
Out[40]:

	Month	No of Guests in Resort Hotel	No of Guests in City Hotel
0	January	1866	2249
1	February	2308	3051
2	March	2571	4049
3	April	2550	4010
4	May	2535	4568

In [41]:

```
px.line(finale, x = 'Month', y = ['No of Guests in Resort Hotel', 'No of Guests in City Hote
```

### Most busy Month



How long do people prefer to stay in the hotels

In [42]:

```
filter = data['is_canceled'] == 0  
clean_data = data[filter]
```

In [43]:

```
clean_data['total_nights'] = clean_data['stays_in_weekend_nights'] + clean_data['stays_in_w
```

In [44]:

```
stay = clean_data.groupby(['total_nights', 'hotel']).agg('count').reset_index()
stay = stay.iloc[:,0:3]
stay.head()
```

Out[44]:

	total_nights	hotel	is_canceled
0	0	City Hotel	251
1	0	Resort Hotel	371
2	1	City Hotel	9155
3	1	Resort Hotel	6579
4	2	City Hotel	10983

In [45]:

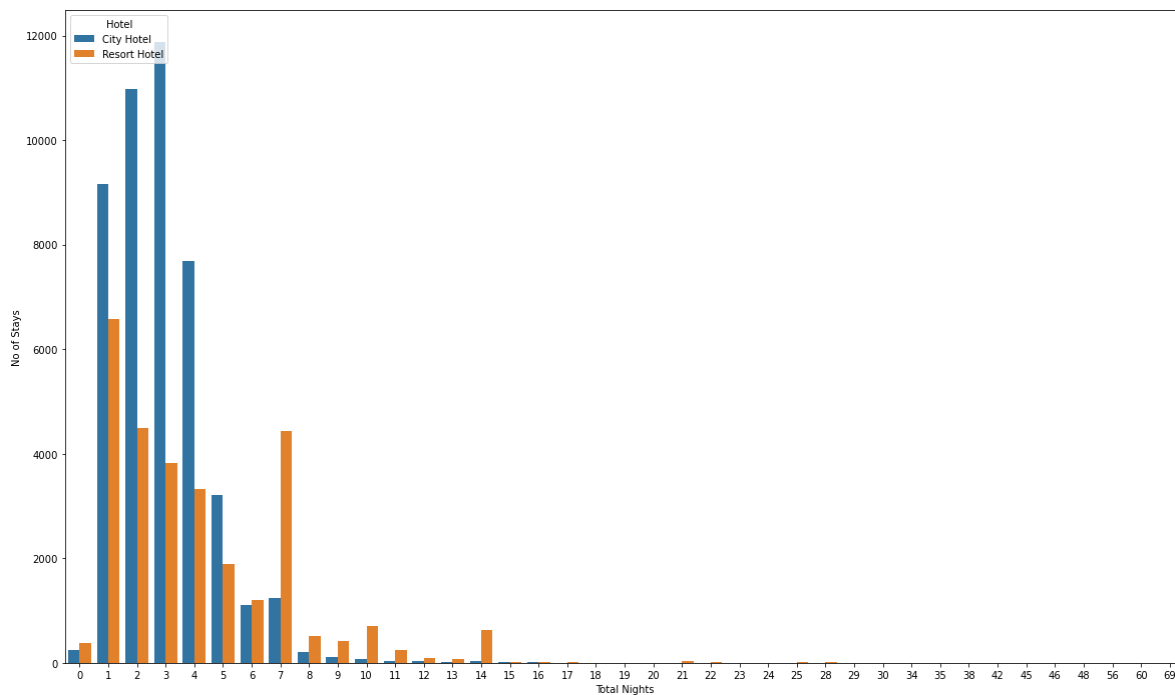
```
stay.columns = ['Total Nights', 'Hotel', 'No of Stays']
```

In [46]:

```
plt.figure(figsize = (20,12))
sns.barplot(x = 'Total Nights', y = 'No of Stays', hue = 'Hotel', hue_order = ['City Hotel'
```

Out[46]:

&lt;AxesSubplot:xlabel='Total Nights', ylabel='No of Stays'&gt;



Bookings by Market Segment



In [47]:

```
clean_data['market_segment'].value_counts()
```

Out[47]:

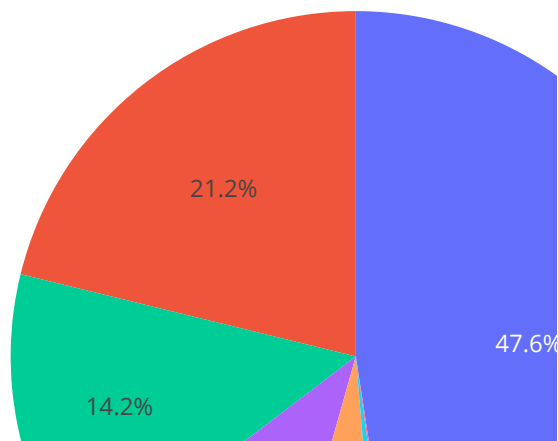
Online TA	35673
Offline TA/TO	15880
Direct	10648
Groups	7697
Corporate	4291
Complementary	639
Aviation	183

Name: market\_segment, dtype: int64

In [48]:

```
clean_data['market_segment'].value_counts() , names = clean_data['market_segment'].value_counts().index, t
```

### Bookings per Market Segment



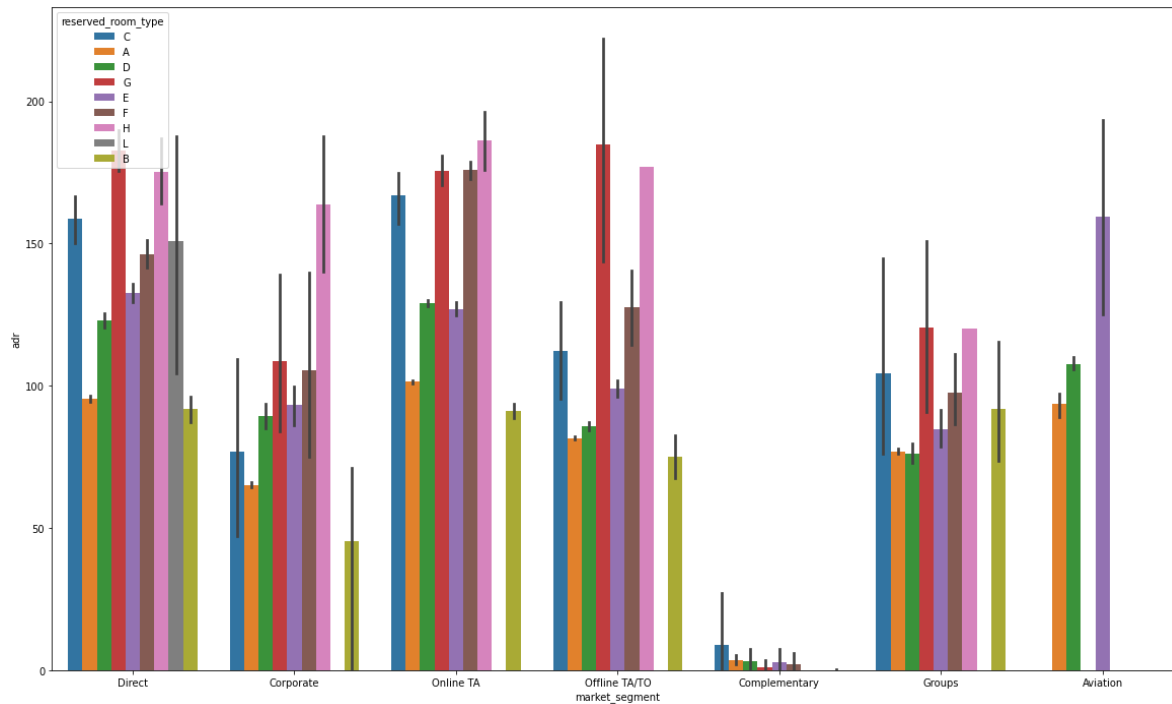
price per night and person based on booking and room

In [49]:

```
plt.figure(figsize = (20,12))  
sns.barplot(x = 'market_segment', y = 'adr', hue = 'reserved_room_type', data = clean_data)
```

Out[49]:

<AxesSubplot:xlabel='market\_segment', ylabel='adr'>



No of bookings cancelled

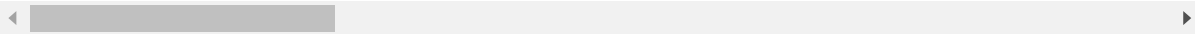
In [50]:

```
cancel = data[data['is_canceled'] == 1]
cancel.head()
```

Out[50]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_num
8	Resort Hotel	1	85	2015	July	
9	Resort Hotel	1	75	2015	July	
10	Resort Hotel	1	23	2015	July	
27	Resort Hotel	1	60	2015	July	
32	Resort Hotel	1	96	2015	July	

5 rows × 32 columns



In [51]:

```
len(cancel[cancel['hotel'] == 'Resort Hotel'])
```

Out[51]:

11120

In [52]:

```
len(cancel[cancel['hotel'] == 'City Hotel'])
```

Out[52]:

33079

In [53]:

```
px.pie(values = [11120,33079], names = ['Rh_cancel', 'Ch_cancel'])
```

