In [ ]:

In [ ]:

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# import pycountry as pc
import matplotlib.ticker as mtick
```

In [ ]:
```
data = pd.read_csv('data.csv')

data.head()
```

Out[ ]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	a
0	Resort Hotel	0	342	2015	July	27	
1	Resort Hotel	0	737	2015	July	27	
2	Resort Hotel	0	7	2015	July	27	
3	Resort Hotel	0	13	2015	July	27	
4	Resort Hotel	0	14	2015	July	27	

5 rows × 32 columns

In [ ]:
```
df = data.copy()
```

Dealing with missing values

In [ ]:
```
df.isnull().sum().sort_values(ascending=False)[:10]
```

Out[ ]:
```
company                 112593
agent                    16340
country                    488
children                     4
reserved_room_type           0
assigned_room_type           0
booking_changes              0
deposit_type                 0
hotel                        0
previous_cancellations       0
dtype: int64
```

In [ ]:
```
df[['agent','company']] = df[['agent','company']].fillna(0.0)
```

In [ ]:
```
df['country'].fillna(data.country.mode().to_string(), inplace=True)


 # for missing children value, replace it with rounded mean value
df['children'].fillna(round(data.children.mean()), inplace=True)
```

In [ ]:
```
df = df.drop(df[(df.adults+df.babies+df.children)==0].index)
```

converting data type

In [ ]:
```
df[['children', 'company', 'agent']] = df[['children', 'company', 'agent']].astype('
```

Creating get_count() function

As we will be using the value_counts function a lot and the values are very big, so we are creating a get_count gunction to fetch the percentage values instead of count values

Creating a plot function

Same as get_count() we need to plot the same graph using the same code a lot of times, so instead, am creating a function for it

In [ ]:
```
def plot(x, y, x_label=None,y_label=None, title=None, figsize=(7,5), type='bar'):
    sns.set_style('darkgrid')

    fig, ax = plt.subplots(figsize=figsize)

    ax.yaxis.set_major_formatter(mtick.PercentFormatter())

    if x_label != None:
        ax.set_xlabel(x_label)

    if y_label != None:
        ax.set_ylabel(y_label)

    if title != None:
        ax.set_title(title)

    if type == 'bar':
        sns.barplot(x,y, ax = ax)
    elif type == 'line':
        sns.lineplot(x,y, ax = ax, sort=False)


    plt.show()
```

1. x: Array containing values for x-axis
2. y: Array containing values for y-axis
3. x_label: String value for x-axis label
4. y_label: String value for y-axis label
5. title: String value for plot title
6. figsize: tuple value, for figure size
7. type: type of plot (default is bar plot)

OUTPUT: the plot

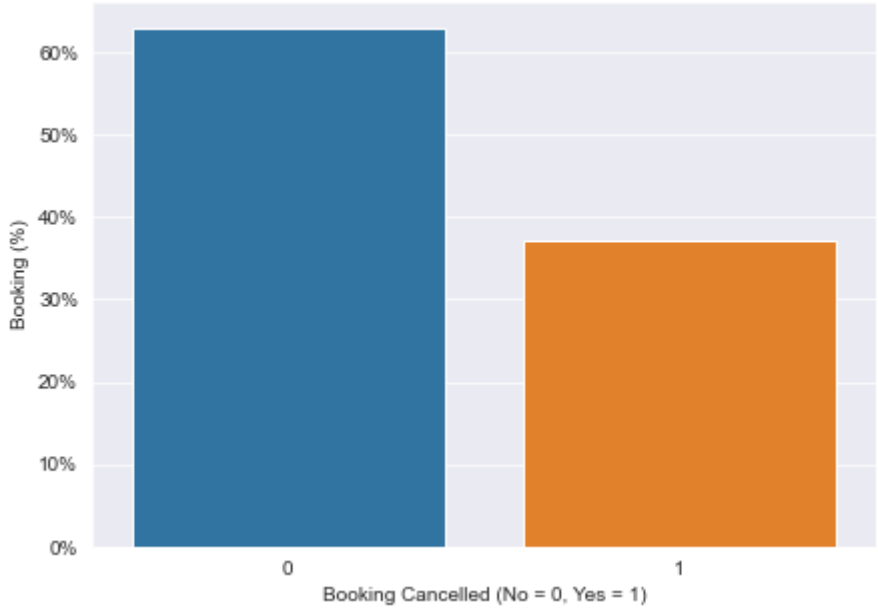
In [ ]:
```
def get_count(series, limit=None):
    if limit != None:
        series = series.value_counts()[:limit]
    else:
        series = series.value_counts()

    x = series.index
    y = series/series.sum()*100

    return x.values,y.values
```

1. How Many Booking Were Cancelled?

In [ ]:
```
x,y = get_count(df['is_canceled'])
```

In [ ]:
```
plot(x,y, x_label='Booking Cancelled (No = 0, Yes = 1)', y_label='Booking (%)')
```

```
C:\Users\rachi\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [ ]:
```
# Seperating the database for not canceled bookings
df_not_canceled = df[df['is_canceled'] == 0]
```

2. What is the booking ratio between Resort Hotel and City Hotel?

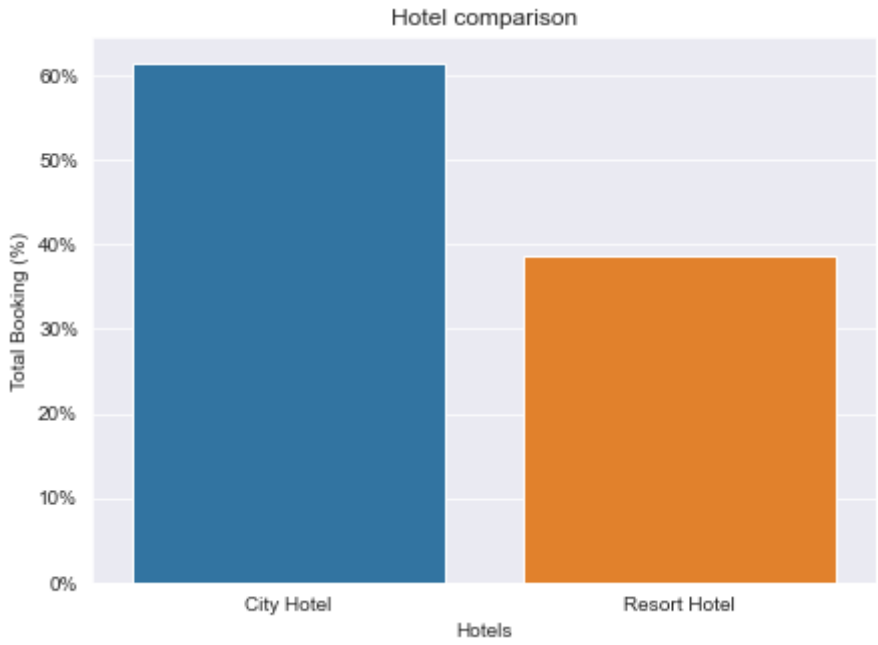
In [ ]:
```
x,y = get_count(df_not_canceled['hotel'])
```

In [ ]:
```
for i in range(len(x)):
    print(f"percent bookings in {x[i]} : {y[i]} ")
```

```
percent bookings in City Hotel : 61.43632267267467
percent bookings in Resort Hotel : 38.56367732732532
```

In [ ]:
```
plot(x,y, x_label='Hotels', y_label='Total Booking (%)', title='Hotel comparison')
```

```
C:\Users\rachi\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



61 % of the people booked in City Hotel

3. What is the percentage of booking for each year?

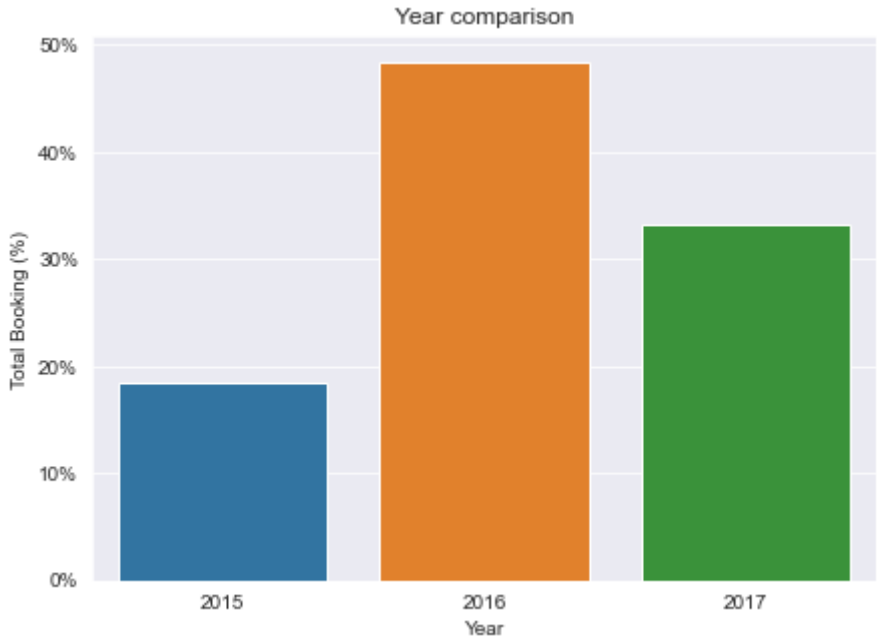
In [ ]:
```
x,y = get_count(df_not_canceled['arrival_date_year'])
```

In [ ]:
```
for i in range(len(x)):
    print(f"percent bookings in {x[i]} : {y[i]} ")
```

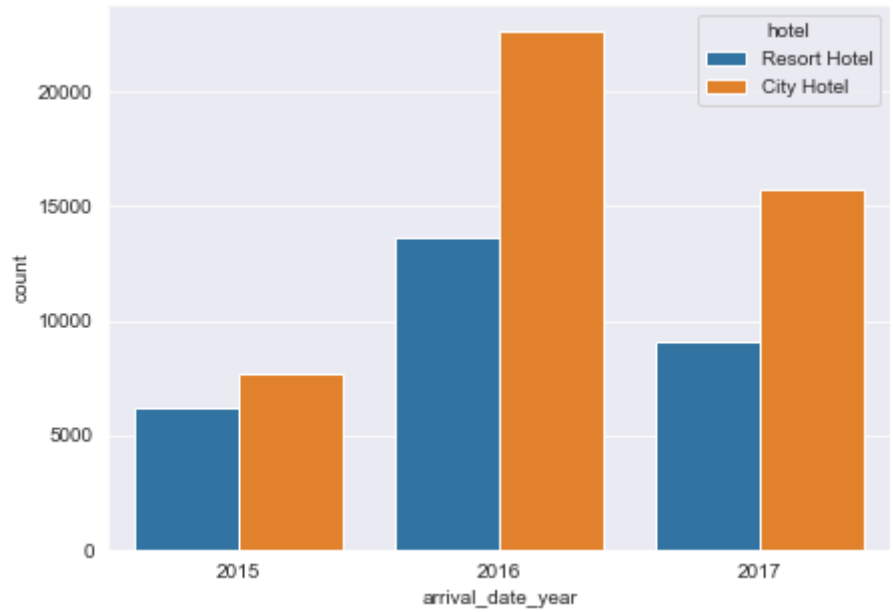
```
percent bookings in 2016 : 48.39156923651198
percent bookings in 2017 : 33.17646745144046
percent bookings in 2015 : 18.431963312047568
```

In [ ]:
```
plot(x,y, x_label='Year', y_label='Total Booking (%)', title='Year comparison')
```

```
C:\Users\rachi\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [ ]:
```
plt.subplots(figsize=(7,5))
sns.countplot(x='arrival_date_year', hue='hotel',  data=df_not_canceled);
```

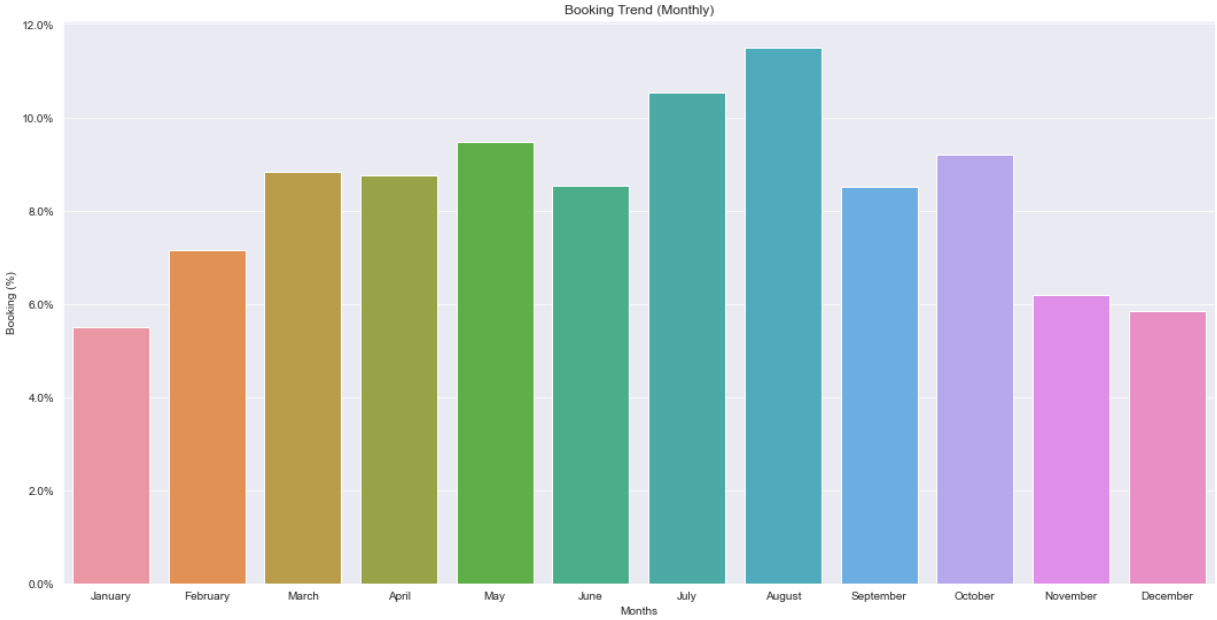


Most of the bookings were done in the year of 2016 which is almost double than what was in 2015,but the booking were reduced by appx. 15% in 2017

4. Which is the busiest month for hotels?

In [ ]:
```
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August
             'October', 'November', 'December']

sorted_months = df_not_canceled['arrival_date_month'].value_counts().reindex(new_ord

x = sorted_months.index
y = sorted_months/sorted_months.sum()*100


#sns.lineplot(x, y.values)
plot(x, y.values, x_label='Months', y_label='Booking (%)', title='Booking Trend (Mon
```

```
C:\Users\rachi\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\_dec
orators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. Fr
om version 0.12, the only valid positional argument will be `data`, and passing othe
r arguments without an explicit keyword will result in an error or misinterpretatio
n.
  warnings.warn(
```



August was the busiest month for the hotels in these 3 years

5. From which country most guests come?

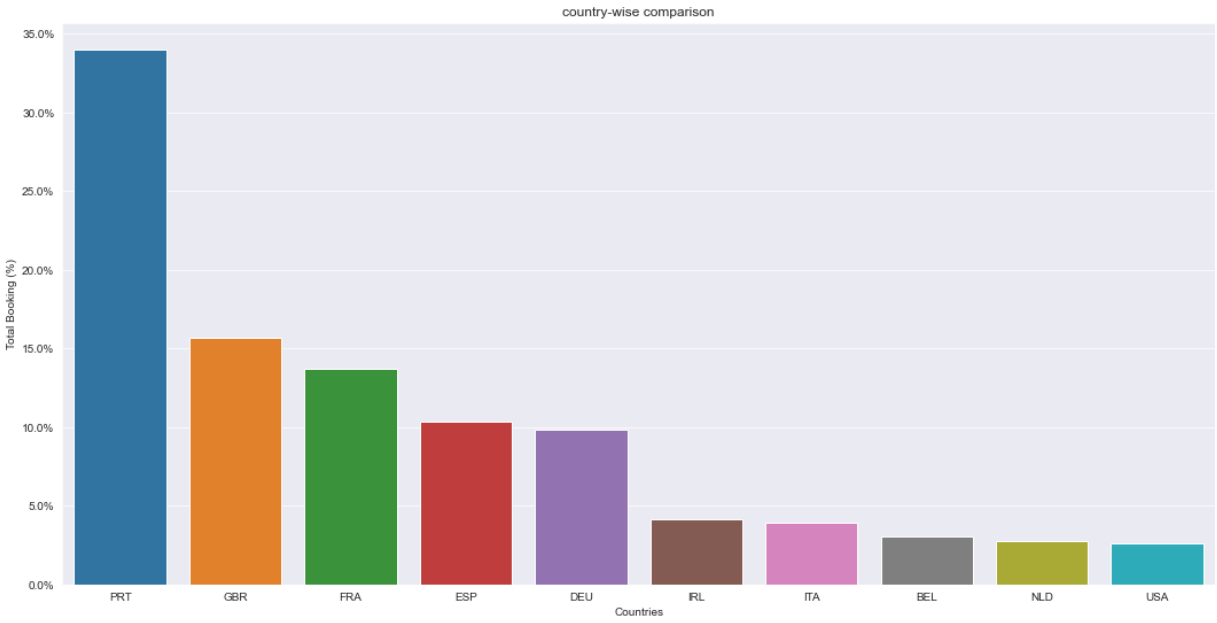
In [ ]:
```
x,y = get_count(df_not_canceled['country'], limit=10)
```

In [ ]:
```
for i in range(len(x)):
    print(f"percent bookings from {x[i]} : {y[i]} ")
```

```
percent bookings from PRT : 33.99342073279424
percent bookings from GBR : 15.667082597352088
percent bookings from FRA : 13.722471600576903
percent bookings from ESP : 10.343709993680013
percent bookings from DEU : 9.83162909786255
percent bookings from IRL : 4.119334294835437
percent bookings from ITA : 3.934596250141795
percent bookings from BEL : 3.027111118313374
percent bookings from NLD : 2.7807937253885173
percent bookings from USA : 2.579850589055081
```

In [ ]:
```
plot(x,y, x_label='Countries', y_label='Total Booking (%)', title='country-wise comp
```

```
C:\Users\rachi\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\_dec
orators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. Fr
om version 0.12, the only valid positional argument will be `data`, and passing othe
r arguments without an explicit keyword will result in an error or misinterpretatio
n.
  warnings.warn(
```



The visualizations suggests that most of the guests are from PRT i.e Portugal

6. How Long People Stay in the hotel?

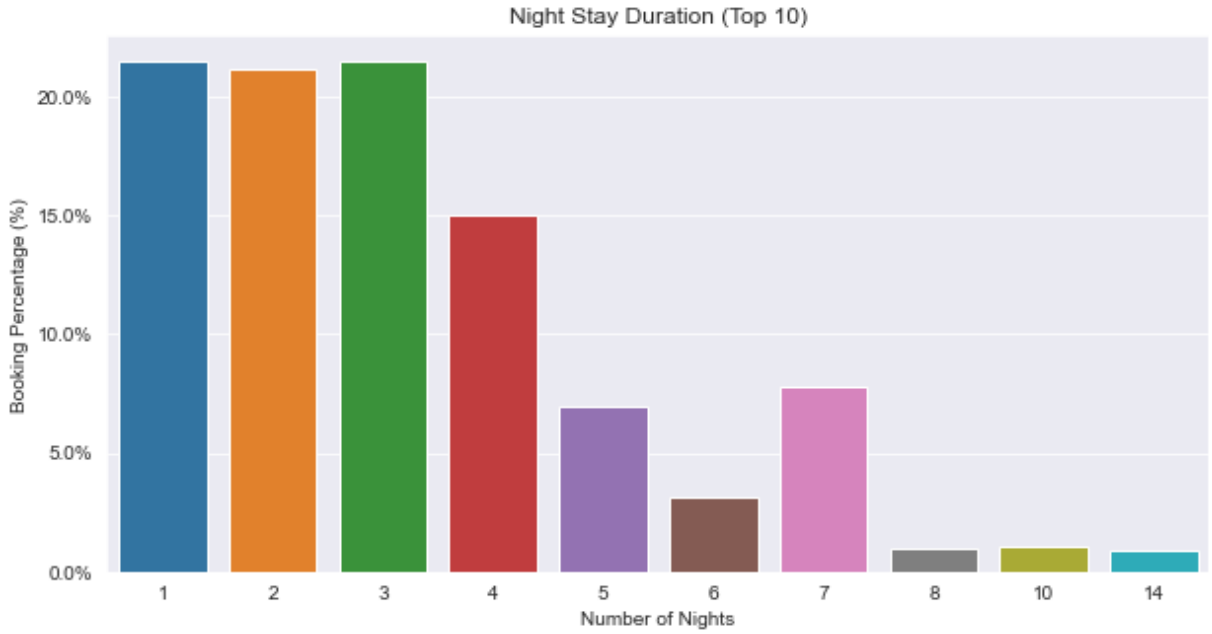
In [ ]:
```
total_nights = df_not_canceled['stays_in_weekend_nights']+ df_not_canceled['stays_in
x,y = get_count(total_nights, limit=10)
```

In [ ]:
```
for i in range(len(x)):
    print(f"count for people who stayed for {x[i]} night(s) : {y[i]} ")
```

```
count for people who stayed for 1 night(s) : 21.49747233228583
count for people who stayed for 3 night(s) : 21.47424511545293
count for people who stayed for 2 night(s) : 21.138133624812134
count for people who stayed for 4 night(s) : 15.04987020084711
count for people who stayed for 7 night(s) : 7.759256729061346
count for people who stayed for 5 night(s) : 6.980461811722913
count for people who stayed for 6 night(s) : 3.1643667167645857
count for people who stayed for 10 night(s) : 1.06435305369586
count for people who stayed for 8 night(s) : 0.9755431069818281
count for people who stayed for 14 night(s) : 0.8962973083754612
```

In [ ]:
```
plot(x,y, x_label='Number of Nights', y_label='Booking Percentage (%)', title='Night
```

```
C:\Users\rachi\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\_dec
orators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. Fr
om version 0.12, the only valid positional argument will be `data`, and passing othe
r arguments without an explicit keyword will result in an error or misinterpretatio
n.
  warnings.warn(
```



Most of the people stayed in for 1,2 or 3 nights

7. Which was the most booked accommodation type (Single, Couple, Family)?

In [ ]:
```
single   = df_not_canceled[(df_not_canceled.adults==1) & (df_not_canceled.children==
couple   = df_not_canceled[(df_not_canceled.adults==2) & (df_not_canceled.children==
family   = df_not_canceled[df_not_canceled.adults + df_not_canceled.children + df_no
```

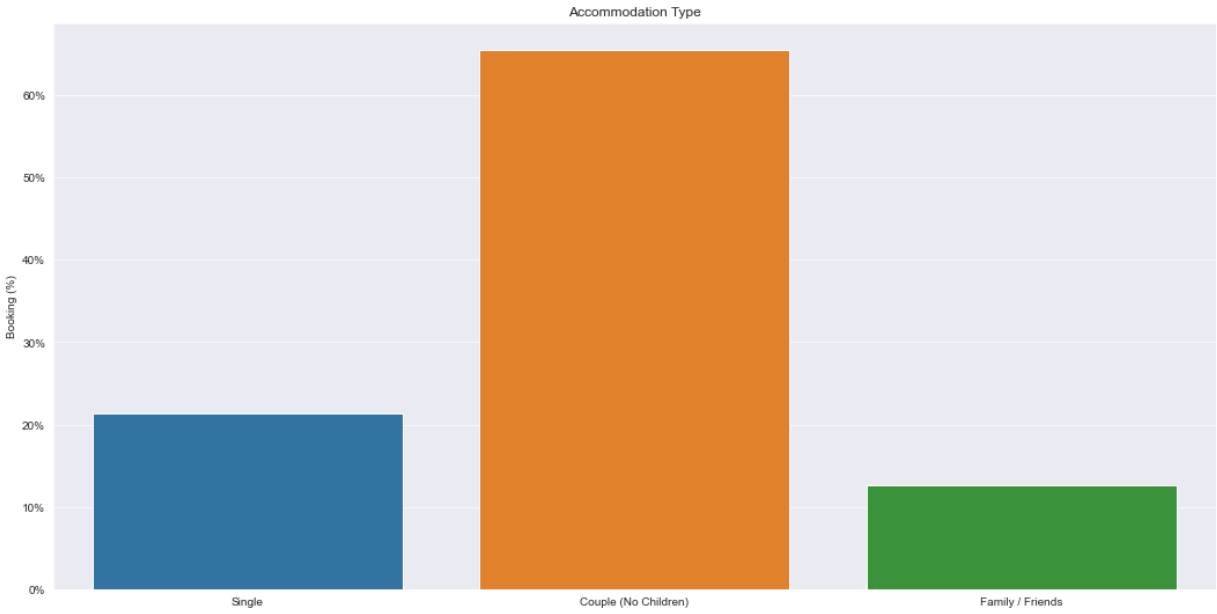
In [ ]:
```
names = ['Single', 'Couple (No Children)', 'Family / Friends']
count = [single.shape[0],couple.shape[0], family.shape[0]]
count_percent = [x/df_not_canceled.shape[0]*100 for x in count]
```

In [ ]:
```
for i in range(len(count)):
    print(f" {names[i]} : count:{count[i]} , percentage: {count_percent[i]} ")
```

```
Single : count:16022 , percentage: 21.359533935022863
Couple (No Children) : count:49136 , percentage: 65.50505925797549
Family / Friends : count:9506 , percentage: 12.672807988161736
```

In [ ]:
```
plot(names,count_percent,  y_label='Booking (%)', title='Accommodation Type', figsiz
```

```
C:\Users\rachi\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\_dec
orators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. Fr
om version 0.12, the only valid positional argument will be `data`, and passing othe
r arguments without an explicit keyword will result in an error or misinterpretatio
n.
  warnings.warn(
```



---

8. Create a suitable model for prediction

In [ ]:
```
df_subset = df.copy()
```

In [ ]:
```
## Making a new column which contains 1 if guest received the same room which was re
df_subset['Room'] = 0
df_subset.loc[ df_subset['reserved_room_type'] == df_subset['assigned_room_type'] ,


## Make the new column which contain 1 if the guest has cancelled more booking in th

df_subset['net_cancelled'] = 0
df_subset.loc[ df_subset['previous_cancellations'] > df_subset['previous_bookings_no
```

Removing unecessary features

In [ ]:
```
df_subset = df_subset.drop(['arrival_date_year','arrival_date_week_number','arrival_
                            'arrival_date_month','assigned_room_type','reserved_room
                            'previous_cancellations','previous_bookings_not_canceled
```

In [ ]:
```
df_subset = df_subset.drop(['reservation_status'], axis=1)
```

In [ ]:
```
df_subset = df_subset.drop(['hotel'], axis=1)
```

In [ ]:
```
df_subset = df_subset.drop(['meal'], axis=1)
```

In [ ]:
```
df_subset.head()
```

Out[ ]:

	is_canceled	lead_time	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	c
0	0	342	0	0	2	0	0	
1	0	737	0	0	2	0	0	

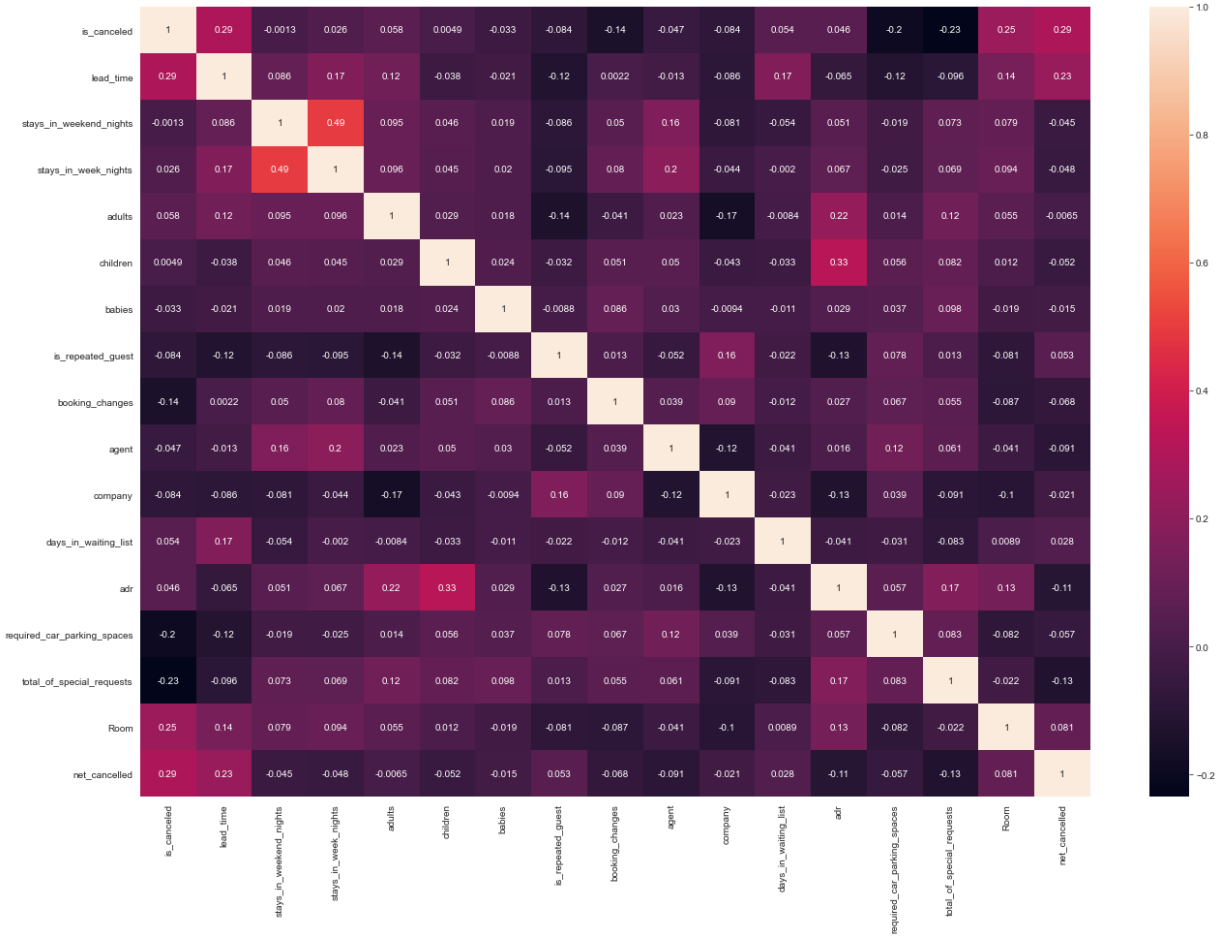
	is_canceled	lead_time	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	c
2	0	7	0	1	1	0	0	
3	0	13	0	1	1	0	0	
4	0	14	0	2	2	0	0	

5 rows × 22 columns

In [ ]:
```
df_subset = df_subset.drop(['country','customer_type','market_segment','distribution
```

Plotting Correlation

In [ ]:
```
fig, ax = plt.subplots(figsize=(22,15))
sns.heatmap(df_subset.corr(), annot=True, ax=ax);
```



Converting Categories to numbers

In [ ]:
```
def transform(dataframe):
    from sklearn.preprocessing import LabelEncoder
    le = LabelEncoder()
    categorical_features = list(dataframe.columns[dataframe.dtypes == object])


    return dataframe[categorical_features].apply(lambda x: le.fit_transform(x))

df = transform(df)
```

Train/Test Split

In [ ]:
```
def data_split(df, label):

    from sklearn.model_selection import train_test_split

    X = df.drop(label, axis=1)
    Y = df[label]

    x_train, x_test, y_train, y_test = train_test_split(X,Y,random_state=0)

    return x_train, x_test, y_train, y_test



x_train, x_test, y_train, y_test = data_split(df_subset, 'is_canceled')
```

Model Creation

Here we are going to use Decision Tree Classifier model

In [ ]:
```
def train(x_train, y_train):
    from sklearn.tree import DecisionTreeClassifier

    clf = DecisionTreeClassifier(random_state=0)
    clf.fit(x_train,y_train)

    return clf


clf = train(x_train, y_train)
```

Model Evaluation

In [ ]:
```
def Score(clf,x_train,y_train,x_test,y_test):
    train_score = clf.score(x_train,y_train)
    test_score = clf.score(x_test,y_test)


    print(f'Training Accuracy: {train_score * 100} %')
    print(f'Test Accuracy : {test_score * 100} %')



Score(clf,x_train,y_train,x_train,y_train)
```

```
Training Accuracy: 98.99448589036652 %
Test Accuracy : 98.99448589036652 %
```