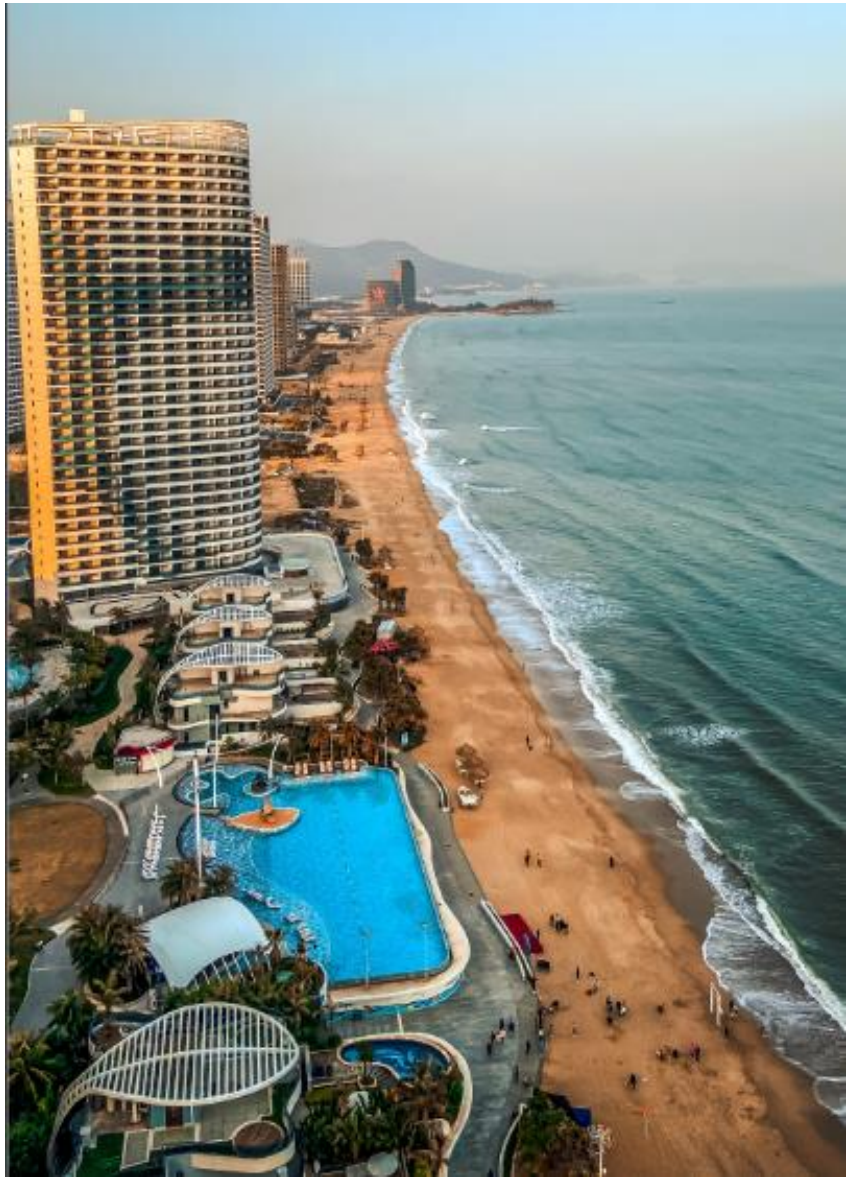
Hotel Bookings

SQL And NoSQL

Assignment 3

Biraj Parikh



Content:

- This data set contains booking information for a city hotel and a resort hotel and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things.

- Source: https://www.kaggle.com/jessemostipak/hotel-booking-demand

Objectives:

This dataset can help answer following questions

- Have you ever wondered when the best time of year to book a hotel room is?
- What is the optimal length of stay in order to get the best daily rate?
- What if you wanted to predict whether or not a guest would actually come or not ? (Data Modeling)

Importing csv data

Using command line interface to import the data:

mongoimport -d project -c hotel_booking --type csv --drop --file "D:\My stuff\IU Courses\SEM 2\SQL AND NOSQL\datasets\Hotel\hotel_bookings.csv" –headerline

The above command line will create database '**project'** and collection '**hotel_booking'**

```
> db.hotel_booking.count()
119390
```

Checking the distribution of the hotels

- The dataset contains data from two different hotels.

```
> db.getCollection("hotel_booking").aggregate(
...    [
...      { $group: {_id: { hotel: "$hotel" }, count: { $sum : 1 } } },
...      { $sort: { count: -1 }}
...    ], { allowDiskUse: true}
... )
{ "_id" : { "hotel" : "City Hotel" }, "count" : 79330 }
{ "_id" : { "hotel" : "Resort Hotel" }, "count" : 40060 }
```

➢ Most guest prefer to book City hotels compared to Resort hotels.

How much do guests pay for a room per night?

```
> var map = function() { emit(this.hotel, this.adr); };
> var reduce = function(key, value) { return Array.avg(value); };
>
> db.hotel_booking.mapReduce(map, reduce, {out: "avg_pay_per_night"});
{
        "result" : "avg_pay_per_night",
        "timeMillis" : 3182,
        "counts" : {
                "input" : 119390,
                "emit" : 119390,
                "reduce" : 1195,
                "output" : 2
        },
        "ok" : 1
}
> db.avg_pay_per_night.find()
{ "_id" : "City Hotel", "value" : 132.21252327431242 }
{ "_id" : "Resort Hotel", "value" : 140.41135653127458 }
```

Resort hotel: 140.41 € per night.
City hotel: 132.22 € per night.
Thus, guest pay a bit extra for Resort hotel per night compared to City hotel.

How does the price per night vary over the year?

```
> var map = function() { emit(this.arrival_date_year, this.adr); };
> var reduce = function(key, value) { return Array.avg(value); };
>
> db.hotel_booking.mapReduce(map, reduce, {out: "avg_pay_per_night_year"});
{
        "result" : "avg_pay_per_night_year",
        "timeMillis" : 3497,
        "counts" : {
                "input" : 119390,
                "emit" : 119390,
                "reduce" : 1559,
                "output" : 3
        },
        "ok" : 1
}
> db.avg_pay_per_night_year.find()
{ "_id" : 2015, "value" : 67.27789995441458 }
{ "_id" : 2016, "value" : 79.17638707554346 }
{ "_id" : 2017, "value" : 84.16596865098506 }
```

- Average pay per night increases every year regardless of type of hotel. That make sense.

Which are the busiest month?

```
> db.hotel_booking.mapReduce(
...     function() {
...        emit(this.arrival_date_month,1); },
...     function(key, values) {return Array.sum(values)}, {
...        out: "busy_month"
...     }
... )
{
        "result" : "busy_month",
        "timeMillis" : 1311,
        "counts" : {
                "input" : 119390,
                "emit" : 119390,
                "reduce" : 2281,
                "output" : 12
        },
        "ok" : 1
}
>
> db.busy_month.aggregate([
...    { $sort: { 'value': -1 }}
... ])
{ "_id" : "August", "value" : 13877 }
{ "_id" : "July", "value" : 12661 }
{ "_id" : "May", "value" : 11791 }
{ "_id" : "October", "value" : 11160 }
{ "_id" : "April", "value" : 11089 }
{ "_id" : "June", "value" : 10939 }
{ "_id" : "September", "value" : 10508 }
{ "_id" : "March", "value" : 9794 }
{ "_id" : "February", "value" : 8068 }
{ "_id" : "November", "value" : 6794 }
{ "_id" : "December", "value" : 6780 }
{ "_id" : "January", "value" : 5929 }
```

As expected the busiest month for all the years in the given data are over the summer i.e. from May to August.

Let’s find the average price per night for all the years over all the month.

- The prices are much higher in the month from May to September i.e. during the summer.
- Thus, the best time to book the hotel room is during the initial month from January to March.
- Spring Break is the best option to hangout with the family with cheap hotel rates.

```
> db.hotel_booking.mapReduce(
...     function() { emit(this.arrival_date_month,this.adr); },
...     function(key, values) {return Array.avg(values)}, {
...         out: "avg_pay_per_night_month"
...     }
... )
{
        "result" : "avg_pay_per_night_month",
        "timeMillis" : 1445,
        "counts" : {
                "input" : 119390,
                "emit" : 119390,
                "reduce" : 2281,
                "output" : 12
        },
        "ok" : 1
}
>
> db.avg_pay_per_night_month.aggregate([
...    { $sort: { 'value': 1 }}
... ])
{ "_id" : "January", "value" : 58.192173974415105 }
{ "_id" : "February", "value" : 59.26588176741394 }
{ "_id" : "November", "value" : 60.96443325209749 }
{ "_id" : "March", "value" : 61.882790001558284 }
{ "_id" : "October", "value" : 68.98726964976154 }
{ "_id" : "April", "value" : 74.6154310158737 }
{ "_id" : "December", "value" : 76.51374233072289 }
{ "_id" : "May", "value" : 80.53101487025869 }
{ "_id" : "September", "value" : 87.14163645545162 }
{ "_id" : "June", "value" : 92.09134138459719 }
{ "_id" : "August", "value" : 107.68692445244817 }
{ "_id" : "July", "value" : 110.28032644717455 }
```

How many bookings were canceled?

```
> db.hotel_booking.aggregate(
... [
...   { $group: {_id: { hotel: "$hotel" }, booking_canceled: { $sum : "$is_canceled" }} },
...   { $project:{_id: 0, hotel: "$_id.hotel", booking_canceled: 1}},
...   { $sort: { count: -1 }}
... ], { allowDiskUse: true}
... ).pretty()
{ "booking_canceled" : 33102, "hotel" : "City Hotel" }
{ "booking_canceled" : 11122, "hotel" : "Resort Hotel" }
```

- Total booking cancelled : 44224 (37%)
- Resort hotel booking cancelled : 11122 (28%)
- City hotel booking cancelled : 33102 (42%)

Which month have the highest number of cancelations?

```
> db.hotel_booking.aggregate(
... [
...   { $group: {_id: { Month: "$arrival_date_month" }, booking_canceled: { $sum : "$is_canceled" }} },
...   { $project:{_id: 0, booking_canceled: 1, Month: "$_id.Month"}},
...   { $sort: { booking_canceled: -1 }}
... ], { allowDiskUse: true}
... ).pretty()
{ "booking_canceled" : 5239, "Month" : "August" }
{ "booking_canceled" : 4742, "Month" : "July" }
{ "booking_canceled" : 4677, "Month" : "May" }
{ "booking_canceled" : 4535, "Month" : "June" }
{ "booking_canceled" : 4524, "Month" : "April" }
{ "booking_canceled" : 4246, "Month" : "October" }
{ "booking_canceled" : 4116, "Month" : "September" }
{ "booking_canceled" : 3149, "Month" : "March" }
{ "booking_canceled" : 2696, "Month" : "February" }
{ "booking_canceled" : 2371, "Month" : "December" }
{ "booking_canceled" : 2122, "Month" : "November" }
{ "booking_canceled" : 1807, "Month" : "January" }
```

As before, since most number of hotel bookings are done from May to August, thus we can expect a lot of booking cancelations during that time.

How long do people stay at the hotels?

```
> db.hotel_booking.mapReduce(
...     function() {
...        emit(this.stays_in_week_nights,1); },
...     function(key, values) {return Array.sum(values)}, {
...        out: "week_night"
...     }
... )
{
        "result" : "week_night",
        "timeMillis" : 1296,
        "counts" : {
                "input" : 119390,
                "emit" : 119390,
                "reduce" : 6160,
                "output" : 35
        },
        "ok" : 1
}
>
> db.week_night.aggregate([
...    { $sort: { 'value': -1 }},
...    { $limit: 10}
... ])
{ "_id" : 2, "value" : 33684 }
{ "_id" : 1, "value" : 30310 }
{ "_id" : 3, "value" : 22258 }
{ "_id" : 5, "value" : 11077 }
{ "_id" : 4, "value" : 9563 }
{ "_id" : 0, "value" : 7645 }
{ "_id" : 6, "value" : 1499 }
{ "_id" : 10, "value" : 1036 }
{ "_id" : 7, "value" : 1029 }
{ "_id" : 8, "value" : 656 }
```

- Most number of hotel bookings are targeted towards 1-4 days. Thus, optimal length of stay would be 3-4 days during the month from January to March to get the best rate.

How many guest repeat their bookings?

```
> db.hotel_booking.aggregate(
... [
...   { $group : { _id: {hotel : "$hotel"}, repeated_guest : { $sum : "$is_repeated_guest"}} },
...   { $project : { _id : 0, hotel : "$_id.hotel", repeated_guest : 1}},
...   { $sort: { 'repeated_guest' : -1}}
... ],{ allowDiskUse: true}
... )
{ "repeated_guest" : 2032, "hotel" : "City Hotel" }
{ "repeated_guest" : 1778, "hotel" : "Resort Hotel" }
>
```

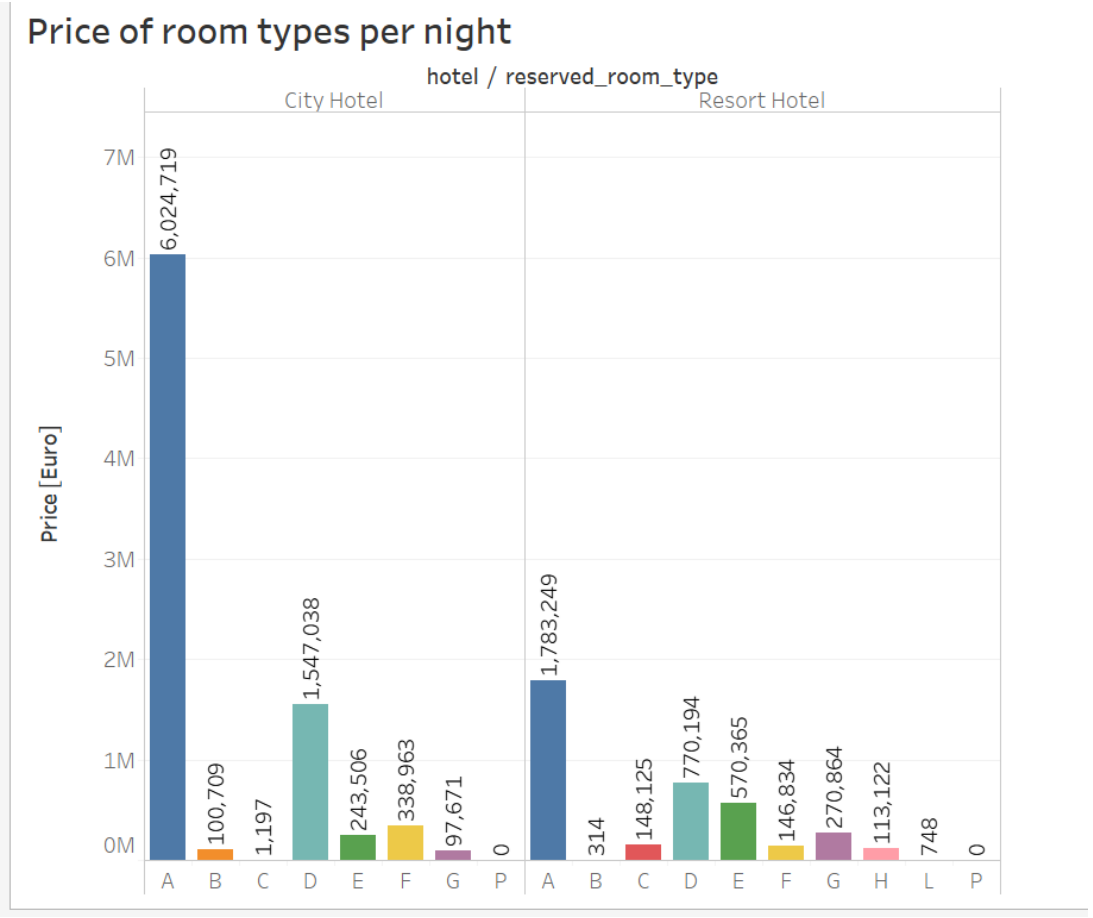
- Resort Hotel repeated guest : 1778 (1.4%)
- City Hotel repeated guest : 2032 (1.7%)
- Total repeated guest : 3810 (3.1%)
- Comparatively, the guest prefer to book again in the city hotels than resort hotels.

What type of rooms guest prefer?

```
> db.hotel_booking.aggregate(
... [
...   { $group : { _id: {Room_type : "$reserved_room_type"}, count : { $sum : 1} } },
...   { $sort: { 'count' : -1}}
... ],{ allowDiskUse: true}
... )
{ "_id" : { "Room_type" : "A" }, "count" : 85994 }
{ "_id" : { "Room_type" : "D" }, "count" : 19201 }
{ "_id" : { "Room_type" : "E" }, "count" : 6535 }
{ "_id" : { "Room_type" : "F" }, "count" : 2897 }
{ "_id" : { "Room_type" : "G" }, "count" : 2094 }
{ "_id" : { "Room_type" : "B" }, "count" : 1118 }
{ "_id" : { "Room_type" : "C" }, "count" : 932 }
{ "_id" : { "Room_type" : "H" }, "count" : 601 }
{ "_id" : { "Room_type" : "P" }, "count" : 12 }
{ "_id" : { "Room_type" : "L" }, "count" : 6 }
>
```

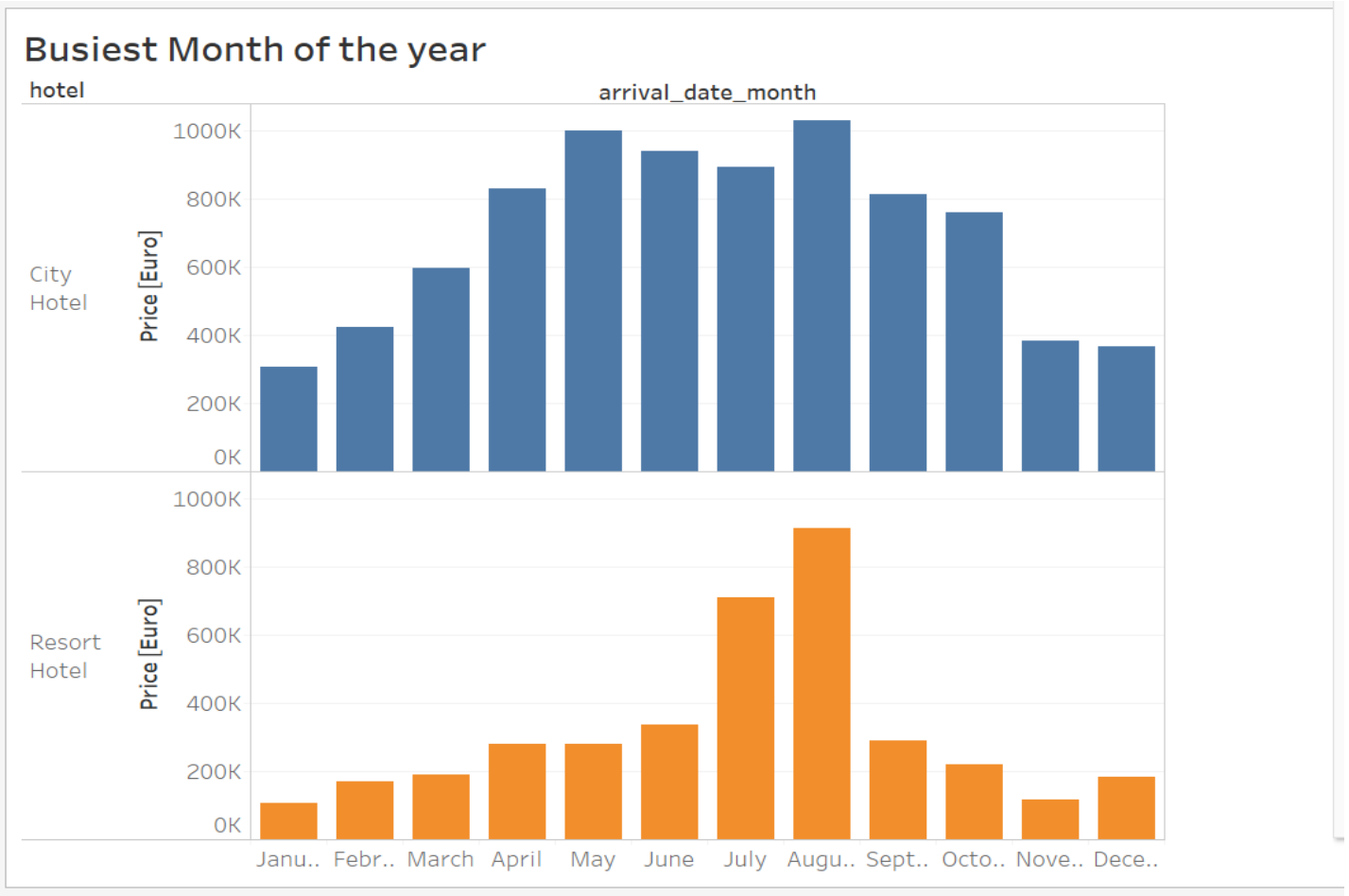
As obvious, majority of the guest prefer luxurious rooms to stay while booking.

Exploratory Data Analysis in Tableau



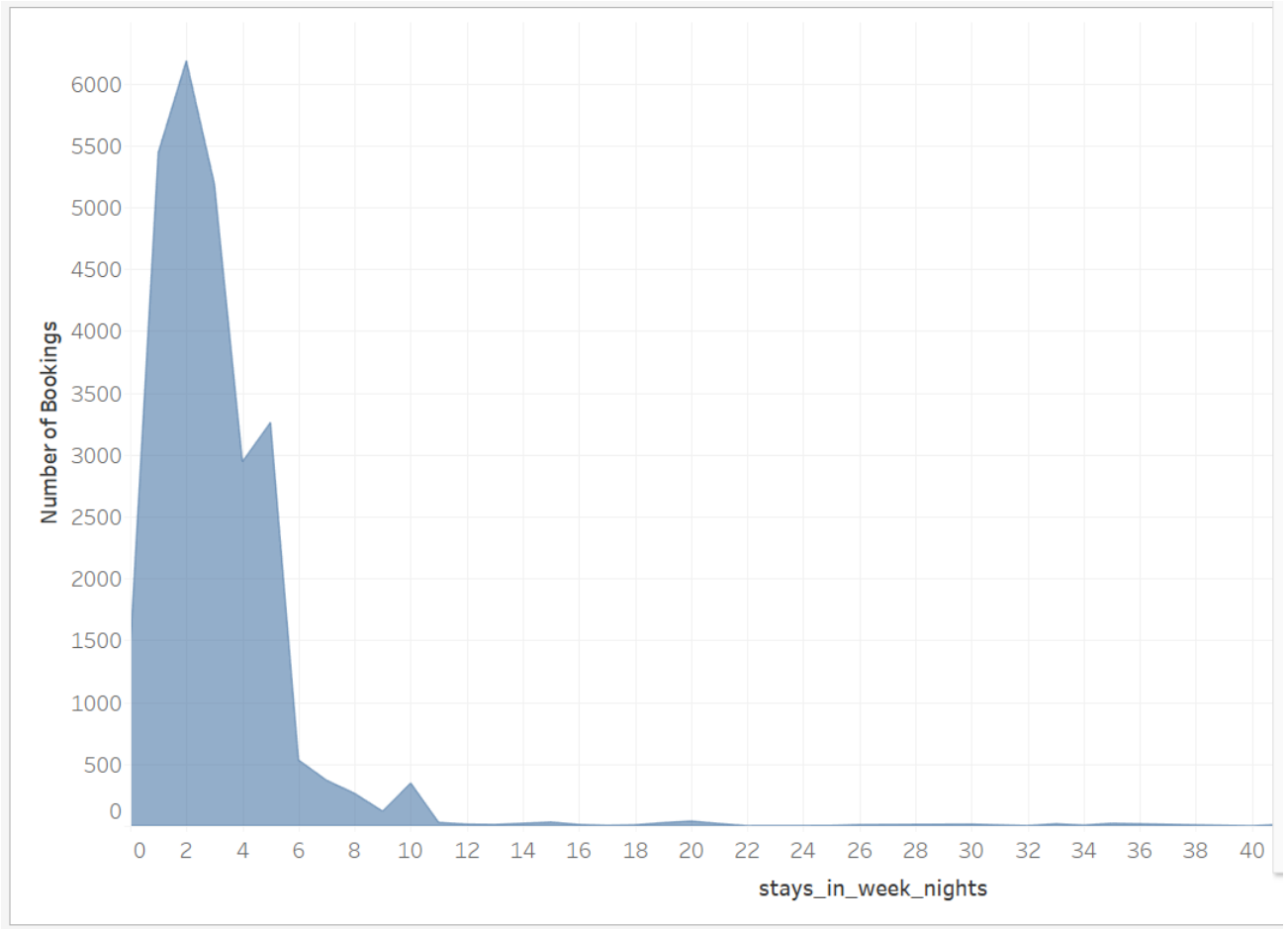
The prices of City hotels are much higher than Resort hotels for room type A.

Busiest Month of the year

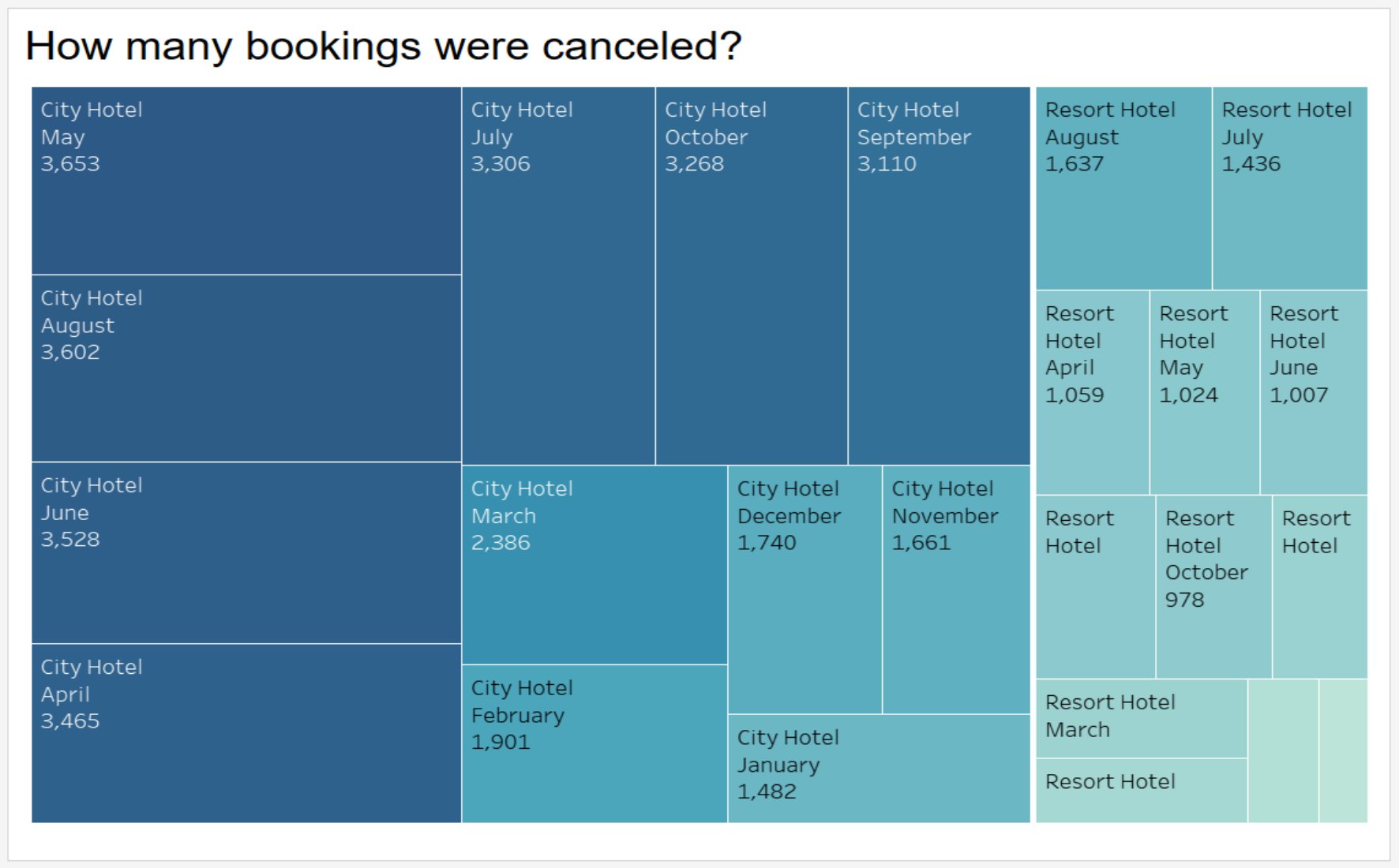


Busiest Month of the year are from May to August i.e. during the summer break.

How long do people stay at the hotels?



- Thus, the graph is inline with the map reduce results i.e. most hotel bookings are often for 2-4 days.



- Most number of bookings are canceled during the month May to September.
- Also, City hotel bookings are canceled more often than Resort hotel bookings.

Data Modeling:

Hotel Bookings

In [1]:
```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

In [2]:
```
#Import the pymongo package

import pymongo
from pymongo import MongoClient
import pprint #for pretty printing
import re
```

In [3]:
```
#Use MongoClient to connect to the already running mongod server

client = MongoClient('localhost', 27017)

#Database 'project' and dataset 'hotel_booking' being selected
db = client['project']
collection = db['hotel_booking']

doc_iterator = collection.find()
```

In [4]:
```
# Counting the number of records
doc_iterator.count()
```

```
D:\Programs\Anaconda\lib\site-packages\ipykernel_launcher.py:2: DeprecationWarning: count is deprecated. U
se Collection.count_documents instead.
```

Out[4]: 119390

Used PyMongo to establish the Client connection.

Converting to pandas DataFrame:

In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# set some display options:
sns.set(style="whitegrid")
pd.set_option("display.max_columns", 36)
from pandas.io.json import json_normalize
```

In [7]:
```
# Converting into pandas dataframe
df = json_normalize(list(collection.find()))
df.head(5)
df.shape
```

Out[7]:

	_id	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_nu
0	5e519cd3056ec54ade7eda15	Resort Hotel	0	342	2015	July	27
1	5e519cd3056ec54ade7eda16	Resort Hotel	0	7	2015	July	27
2	5e519cd3056ec54ade7eda17	Resort Hotel	0	737	2015	July	27
3	5e519cd3056ec54ade7eda18	Resort Hotel	0	14	2015	July	27
4	5e519cd3056ec54ade7eda19	Resort Hotel	0	13	2015	July	27

Out[7]: (119390, 33)

Data Preprocessing:

In [19]:
```
# manually choose columns to include
# some columns make no sense to include (arrival_date_year, assigned_room_type,
# booking_changes, reservation_status, reservation_status)

num_features = ["lead_time","arrival_date_week_number","arrival_date_day_of_month",
                "stays_in_weekend_nights","stays_in_week_nights","adults",
                "babies","is_repeated_guest", "previous_cancellations",
                "previous_bookings_not_canceled","agent","company","days_in_waiting_list",
                "required_car_parking_spaces", "total_of_special_requests",
               "adr"]

cat_features = ["hotel","arrival_date_month","meal","market_segment",
                "distribution_channel","reserved_room_type","deposit_type","customer_type"]
```

In [20]:
```
X_train = X_train_df[num_features + cat_features].copy()
X_test = X_test_df[num_features + cat_features].copy()

# preprocess numerical feats:
num_transformer = SimpleImputer(strategy="constant")

# Preprocessing for categorical features:
cat_transformer = Pipeline(steps=[
    ("imputer", SimpleImputer(strategy="constant", fill_value="Unknown")),
    ("onehot", OneHotEncoder(handle_unknown='ignore'))])

# Bundle preprocessing for numerical and categorical features:
preprocessor = ColumnTransformer(transformers=[('num', num_transformer, num_features),('cat', cat_transformer, cat_features)])
```

Applying the Machine learning Model:

Predict whether the guest would cancel the hotel bookings or not.

In [21]:
```
# Define model:
model = RandomForestClassifier(random_state=0)
my_pipeline = Pipeline(steps=[('preprocessor', preprocessor),('model', model)])

# # Preprocessing of training data, fit model:
my_pipeline.fit(X_train, y_train)

# Preprocessing of validation data, get predictions:
preds = my_pipeline.predict(X_test)

# # Evaluate the model:
score = accuracy_score(y_test, preds)
print("\n")
print("accuracy_score:{} % ".format(round(score,2)))
```

```
accuracy_score: 86 %
```

- Thus, the hotel owners can make a guess whether the guest would cancel the hotel bookings or not using the above prediction.
- Complete Data Analysis: https://github.com/birajparikh16/Exploratory-Data-Analysis-Hotel-Bookings

Thank you.