UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

**BSD2333 DATA WRANGLING**

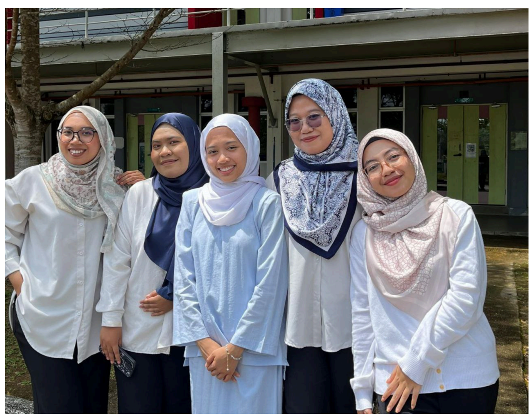
**GROUP PROJECT**

**2022/2023 SEMESTER II**

**SECTION : 01G**

**TITLE: HOTEL BOOKING ANALYSIS**

**PREPARED FOR: DR. MOHD KHAIRUL BAZLI BIN MOHD AZIZ**



**GROUP MEMBERS:**

| NAME | STUDENT ID |
|---|---|
| NUR NABILA BINTI ABD RAHMAN | SD22037 |
| SITI MAISARAH BINTI SUHARDI | SD22006 |
| NUR NABILAH BINTI SUZELAN AMIR | SD22053 |
| NOR MIMI AZURA BINTI HUZAIMI | SD22011 |
| NOR ADLIN SOFEA BINTI NOR HAIRUDIN | SD22024 |

## TABLE OF CONTENT


**1.0 Synopsis** **2**
- 1.1 Description of the assignment 2
- 1.2 Problem to be solved 3
- 1.3 Question to be answered 3
- 1.4 Objectives 4
- 1.5 Basic Description of the Data 4

**2.0 Packages Required** **8**

**3.0 Data Preparation** **10**
- 3.1 Flowchart process of Data Preparation 10
- 3.2 Data Exploration 11
- 3.3 Data Cleaning 13

**4.0 Exploratory Data Analysis** **27**
- 4.1 Bar Chart 27
- 4.2 Line Chart 28
- 4.3 Side by Side Chart 29
- 4.4 Pie Chart 30
- 4.5 Stacked Bar Chart 32

**5.0 Summary** **33**

**Appendix** **34**

**References** **34**

## 1.0 Synopsis

### 1.1 Description of the assignment

It's essential to comprehend booking trends and guest behavior in order to maximize hotel revenue and operations. This project provides actionable insights by analyzing important factors including guest loyalty, cancellations of reservations, pricing strategies, seasonal trends, and weekday versus weekend booking patterns.

First, the study will analyze the proportion of bookings made by repeated guests compared to new guests. Understanding guest loyalty is essential, and identifying strategies to increase repeat bookings can significantly benefit the hotel.

Second, this project will examine the rate of booking cancellations versus non-cancellations. By providing insights into the factors contributing to cancellations, the hotel can develop strategies to reduce them, thereby optimizing revenue and improving customer satisfaction.

Third, the analysis will focus on monthly booking trends to identify peak and off-peak seasons. This information can be used for better resource allocation and to create targeted marketing campaigns, ensuring efficient use of hotel resources throughout the year.

Fourth, the study will investigate the variation in the average daily rate (ADR) across different months. Understanding pricing strategies and their impact on occupancy and revenue is crucial for the hotel's financial performance.

Lastly, this project will compare booking patterns between weekdays and weekends. By understanding demand fluctuations, the hotel can tailor promotional offers accordingly, maximizing occupancy and revenue.

This hotel booking data analysis provides information to improve client satisfaction, make the most use of available resources, and increase income. The hotel can enhance its performance and sustain its competitive advantage by focusing on factors such as guest loyalty, cancellations, seasonal patterns, pricing, and demand changes.

### 1.2 Problem to be solved

The main challenge is to identify the factors that significantly influence hotel reservations and project strategies to optimize revenue and customer satisfaction. Additionally, we aim to address issues such as booking cancellations, guest loyalty, and pricing strategies to enhance operational efficiency and profitability.

First, the analysis will seek to respond to the improved guest loyalty by considering the difference between repeat guests and new guests. The way to solve this problem is to come up with specific marketing concepts that are aimed at enhancing the conversion rates as well as the overall guest loyalty. Second, booking cancellations pose a major task. This knowledge will help the hotel to take appropriate action such as cheaper cancellations for more bookings, good communication, and a package of other incentives for guests who confirm bookings with the aim of reducing the rates of cancellations.

Some of the problems identified on the dataset include an excess of null values and outliers, as well as the irrelevance of some columns. Before applying any situation and transformations on the given dataset the first step is to clean it appropriately this involves handling of null values, outliers, and dropping unwanted columns. To define these problems, the project's goal is to deliver solutions to the hotel on how to improve operations, increase revenues, and satisfy customers.

### 1.3 Question to be answered

1. What are the variables that affect customer behavior?
2. How can we make hotel reservations cancellation better?
3. How can all hotels be assisted in making pricing and promotional decisions?

### 1.4 Objectives

1. To analyze the factors that can affect customer behavior and booking patterns of resort and city hotels.
2. To enhance customer service to retain existing guests and encourage repeat customers..
3. To improve customer satisfaction by providing suitable pricing and promotional decisions follow market conditions and customer behavior.

### 1.5 Basic Description of the Data

| No | Attributes | Explanation | Type |
|---|---|---|---|
| 1 | hotel | One of the hotels is a resort hotel and the other is a city hotel. | Qualitative |
| 2 | canceled | Value indicating if the booking was canceled (1) or not (0). | Qualitative |
| 3 | year_arrival | Year of arrival date. | Qualitative |
| 4 | month_arrival | Month of arrival date with 12 categories: “January” to “December”. | Qualitative |
| 5 | week_arrival | Week number of the arrival date. | Qualitative |
| 6 | date_arrival | Day of the month of the arrival date. | Qualitative |

| | | | |
|---|---|---|---|
| 7 | weekend_stays | Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel. | Qualitative |
| 8 | weekdays_stays | Number of week nights (Monday to Friday) the guest stayed. | Qualitative |
| 9 | adults | Number of adults | Quantitative |
| 10 | children | Number of Children | Quantitative |
| 11 | babies | Number of Babies | Quantitative |
| 12 | meal | BB – Bed & Breakfast | Qualitative |
| 13 | country | Country of origin. | Qualitative |
| 14 | market_segment | Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators” | Qualitative |
| 15 | booking_channel | Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators” | Qualitative |
| 16 | repeat_guest | Value indicating if the booking name was from a repeated guest (1) or not (0) | Qualitative |
| 17 | prior_cancellation | Number of previous bookings that were canceled by the customer prior to the current booking | Quantitative |
| 15 | prior_noncancellation | Number of previous bookings not | Quantitative |

| | | | |
|---|---|---|---|
| | | canceled by the customer prior to the current booking | |
| 16 | reserved_room | Code of room type reserved. Code is presented instead of designation for anonymity reasons. | Qualitative |
| 17 | assigned_room | Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons | Qualitative |
| 18 | deposit_type | No Deposit – no deposit was made; Non Refund – a deposit was made in the value of the total stay cost; Refundable – a deposit was made with a value under the total cost of stay. | Qualitative |
| 19 | agent | ID of the travel agency that made the booking | Qualitative |
| 20 | company | ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons | Qualitative |
| 21 | waiting_days | Number of days the booking was in the waiting list before it was confirmed to the customer | Quantitative |

| | | | |
|---|---|---|---|
| 22 | customer_type | Group – when the booking is associated to a group; Transient – when the booking is not part of a group or contract, and is not associated to other transient booking; Transient-party – when the booking is transient, but is associated to at least other transient booking | Qualitative |
| 23 | avg_dailyrate | Average Daily Rate (Calculated by dividing the sum of all lodging transactions by the total number of staying nights) | Quantitative |
| 24 | parking_required | Number of car parking spaces required by the customer | Quantitative |
| 25 | total_request | Number of special requests made by the customer (e.g. twin bed or high floor) | Quantitative |
| 26 | reservation_status | Check-Out – customer has checked in but already departed; No-Show – customer did not check-in and did inform the hotel of the reason why | Qualitative |
| 27 | reservation_date | Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to understand when was the booking canceled or when did the customer checked-out of the hotel | Qualitative |

| | | | |
|---|---|---|---|
| 28 | name | Name of the Guest | Qualitative |
| 29 | email | Email | Qualitative |
| 30 | phoneNo | Phone number | Qualitative |

## 2.0 Packages Required

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import ipywidgets as widgets
from IPython.display import display
```

*Figure 1 Import Packages*

1. Pandas

   Pandas is used for data manipulation and it also contains analytical functions. This is as it is equipped with data structures like the Series and DataFrame which are very crucial when dealing with structured data. The code employs reading and writing data from different file formats such as CSV and Excel file data. Also, for the purposes of data cleaning and processing, for data analysis thesaurus exploration.

2. Matplotlib

   Matplotlib is a widely known and used package used for interactive plotting and visualization. The best thing about this visualization tool is that it can be highly customized and can plot the graphs, histogram, bar chart, scatter plots, etc. It is used mostly for generating 2D graf and modifying the visual of the plot including color of the plot, labels and scales.

3. Seaborn

   Matplotlib is not the only data visualization library available in Python, there exists another library called Seaborn which is actually built on top of Matplotlib. Here is a refined and comprehensive toolkit for designing great looking and enlightening statistical graphics. It seems to go hand in hand with dataframes of the Pandas library of data structures in python. It is used for creating statistical plots such as heat map and box plot for visiting the distribution and relationships of data.

4. Ipywidgets

   Ipywidgets is a library that provides interactive HTML widget. It allows for creation of interactive widgets such as sliders, buttons and dropdowns to visualize dynamically. It is to add interactive control and make the dashboard an interactive report.

5. Plotly

   Plotly is a graphing library that makes interactive, publication quality graphs online. It supports various types of plot including line chart, scatter plot and more with extensive interactive and customizability. This is to make interactive and dynamic plots using a large dataset with hover and zoom functionality.

## 3.0 Data Preparation

### 3.1 Flowchart process of Data Preparation

The first half process of this flowchart is called data preparation process which involves the process of importing data, cleaning the data where checking outliers, checking missing value and renaming data has been done. Describing data is the end of this data preparation process. Next move into the data analysis process which begins with importing necessary libraries into the python and extracting useful information to make visualization for further interpretation analysis.

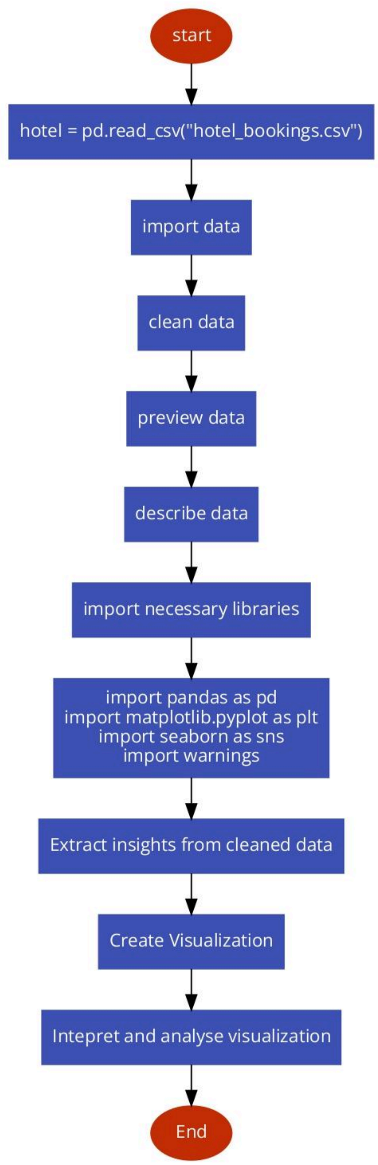



*Figure 2 Flowchart of Data preparation*

### 3.2 Data Exploration

```
hotel = pd.read_csv("hotel_bookings.csv")

<ipython-input-26-8e3d169691a7>:1: DtypeWarning: Columns (24) have mixed types. Specify dtype option on import or set low_memory=False.
  hotel = pd.read_csv("hotel_bookings.csv")
```

*Figure 3.2.1 Import Data*

Before starting the analysis, the data must be imported so that we can get access to the data.

```
hotel.head()
```

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

5 rows × 32 columns

*Figure 3.2.2 Viewing the data*

Viewing the few rows of the dataset helps in understanding the structure and content of the data.

```
hotel.shape

(58890, 32)
```

*Figure 3.2.2 Shape of the data*

The shape helps in understanding the data rows and columns.

```
hotel.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58890 entries, 0 to 58889
Data columns (total 32 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   hotel                           58890 non-null  object 
 1   is_canceled                     58890 non-null  int64  
 2   lead_time                       58890 non-null  int64  
 3   arrival_date_year               58890 non-null  int64  
 4   arrival_date_month              58890 non-null  object 
 5   arrival_date_week_number        58890 non-null  int64  
 6   arrival_date_day_of_month       58890 non-null  int64  
 7   stays_in_weekend_nights         58890 non-null  int64  
 8   stays_in_week_nights            58890 non-null  int64  
 9   adults                          58890 non-null  int64  
 10  children                        58886 non-null  float64
 11  babies                          58890 non-null  int64  
 12  meal                            58890 non-null  object 
 13  country                         58412 non-null  object 
 14  market_segment                  58890 non-null  object 
 15  distribution_channel            58890 non-null  object 
 16  is_repeated_guest               58890 non-null  int64  
 17  previous_cancellations          58890 non-null  int64  
 18  previous_bookings_not_canceled  58890 non-null  int64  
 19  reserved_room_type              58890 non-null  object 
 20  assigned_room_type              58890 non-null  object 
 21  booking_changes                 58890 non-null  int64  
 22  deposit_type                    58890 non-null  object 
 23  agent                           49758 non-null  float64
 24  company                         3479 non-null   object 
 25  days_in_waiting_list            58889 non-null  float64
 26  customer_type                   58889 non-null  object 
 27  adr                             58889 non-null  float64
 28  required_car_parking_spaces     58889 non-null  float64
 29  total_of_special_requests       58889 non-null  float64
 30  reservation_status              58889 non-null  object 
 31  reservation_status_date         58889 non-null  object 
dtypes: float64(6), int64(13), object(13)
```

*Figure 3.2.3 Data Info*

This provides information about the data types of each column.

## 3.3 Data Cleaning

<u>Rename Data</u>

```
new_column_names = {
    'is_canceled' : 'canceled',
    'lead_time' : 'total_cancel',
    'arrival_date_year' : 'year_arrival',
    'arrival_date_month' : 'month_arrival',
    'arrival_date_week_number' : 'week_arrival',
    'arrival_date_day_of_month' : 'date_arrival',
    'stays_in_weekend_nights' : 'weekend_stays',
    'stays_in_week_nights' : 'weekdays_stays',
    'distribution_channel' : 'booking_channel',
    'is_repeated_guest' : 'repeat_guest',
    'previous_cancellations' : 'prior_cancellation',
    'previous_bookings_not_canceled' : 'prior_noncancellation',
    'reserved_room_type' : 'reserved_room',
    'assigned_room_type' : 'assigned_room',
    'days_in_waiting_list' : 'waiting_days',
    'adr' : 'avg_dailyrate',
    'required_car_parking_spaces': 'parking_required',
    'total_of_special_requests': 'total_request',
    'reservation_status_date' : 'reservation_date',
    'company' : 'company',
    'agent' : 'agent',
    'phone-number' : 'phoneNo'
}


# Use the rename() method to rename columns
hotel.rename(columns=new_column_names, inplace=True)
```

```python
#to check variable name after rename
hotel.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58890 entries, 0 to 58889
Data columns (total 32 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   hotel                 58890 non-null  object 
 1   canceled              58890 non-null  int64  
 2   total_cancel          58890 non-null  int64  
 3   year_arrival          58890 non-null  int64  
 4   month_arrival         58890 non-null  object 
 5   week_arrival          58890 non-null  int64  
 6   date_arrival          58890 non-null  int64  
 7   weekend_stays         58890 non-null  int64  
 8   weekdays_stays        58890 non-null  int64  
 9   adults                58890 non-null  int64  
 10  children              58886 non-null  float64
 11  babies                58890 non-null  int64  
 12  meal                  58890 non-null  object 
 13  country               58412 non-null  object 
 14  market_segment        58890 non-null  object 
 15  booking_channel       58890 non-null  object 
 16  repeat_guest          58890 non-null  int64  
 17  prior_cancellation    58890 non-null  int64  
 18  prior_noncancellation 58890 non-null  int64  
 19  reserved_room         58890 non-null  object 
 20  assigned_room         58890 non-null  object 
 21  booking_changes       58890 non-null  int64  
 22  deposit_type          58890 non-null  object 
 23  agent                 49758 non-null  float64
 24  company               3479 non-null   object 
 25  waiting_days          58889 non-null  float64
 26  customer_type         58889 non-null  object 
 27  avg_dailyrate         58889 non-null  float64
 28  parking_required      58889 non-null  float64
 29  total_request         58889 non-null  float64
 30  reservation_status    58889 non-null  object 
 31  reservation_date      58889 non-null  object 
dtypes: float64(6), int64(13), object(13)
memory usage: 14.4+ MB
```

In this process, we rename columns to provide a better understanding of the meaning of each variable without needing to refer to documentation or additional explanation. Renaming columns also helps standardize column names across datasets while making it easier to merge or compare datasets . In this datset we rename 19 columns which are ('is_canceled' to 'canceled'), ('lead_time' to 'total_cancel') , ('arrival_date_year' to 'year_arrival'),('arrival_date_month' to 'month_arrival'), ('arrival_date_week_number' to 'week_arrival'), ('arrival_date_day_of_month' to 'date_arrival'),('stays_in_weekend_nights' to 'wekend_stays'), ('stays_in_week_nights' to 'weekdays_stays'), ('distribution_channel' to 'booking_channel'), ('is_repeated_guest' to 'repeat_guest'), (previous_bookings_not_canceled' to 'prior_noncancellation'), ('reserved_room_type' to 'reserved_room') ('assigned_room_type' to 'assigned_room'), ('days_in_waiting_list' to 'waiting_days'), ('adr' to 'avg_dailyrate'), ('required_car_parking_spaces' to 'parking_required'), ('total_of_special_requests' to 'total_request') and ('reservation_status_date' to 'reservation_date')

## Checking Missing Value

```
[ ] count_null = hotel.isnull().sum()
    missing_data_found = False

    for i, count in enumerate(count_null):
        if count > 0:
            print('Yes, have missing data.')
            missing_data_found = True
            break

    if not missing_data_found:
        print('No missing data found.')
```

```
Yes, have missing data.
```

| | Null Values |
|---|---|
| hotel | 0 |
| canceled | 0 |
| total_cancel | 0 |
| year_arrival | 0 |
| month_arrival | 0 |
| week_arrival | 0 |
| date_arrival | 0 |
| weekend_stays | 0 |
| weekdays_stays | 0 |
| adults | 0 |
| children | 4 |
| babies | 0 |
| meal | 0 |
| country | 464 |
| market_segment | 0 |
| booking_channel | 0 |
| repeat_guest | 0 |
| prior_cancellation | 0 |
| prior_noncancellation | 0 |
| reserved_room | 0 |
| assigned_room | 0 |
| booking_changes | 1 |
| deposit_type | 1 |
| agent | 8640 |
| company | 40636 |
| waiting_days | 1 |
| customer_type | 1 |
| avg_dailyrate | 1 |
| parking_required | 1 |
| total_request | 1 |
| reservation_status | 1 |
| reservation_date | 1 |

We discovered that several columns in the dataset have missing data. For example, there are four cases where the "children" column is empty, and 464 items in the "country" column are missing information. The "agent" column stands out, with 8,640 entries in which no agent information is provided. However, the largest important lack is in the "company" column, where 112,593 entries are completely missing solid information. To ensure that our analysis is thorough and correct, we must carefully address these missing variables, possibly by filling in the gaps where appropriate or altering our strategy to appropriately handle these gaps during our research.

```
[ ]  hotel=hotel.drop(['agent','company'], axis = 1)
```

Column agent and company has a high percentage of missing values and may not be very informative for predicting cancellations, so it may be better to drop it entirely.

```
[ ] #to verify the changes
    hotel.head()
```

| | hotel | canceled | total_cancel | year_arrival | month_arrival | week_arrival | date_arrival | weekend_stays | weekdays_stays | adults | ... | assigned_room | booking_changes | deposit_type | waiting_days | customer_typ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Resort Hotel | 0 | 342 | 2015 | July | 27 | 1 | 0 | 0 | 2 | ... | C | 3.0 | No Deposit | 0.0 | Transien |
| 1 | Resort Hotel | 0 | 737 | 2015 | July | 27 | 1 | 0 | 0 | 2 | ... | C | 4.0 | No Deposit | 0.0 | Transien |
| 2 | Resort Hotel | 0 | 7 | 2015 | July | 27 | 1 | 0 | 1 | 1 | ... | C | 0.0 | No Deposit | 0.0 | Transien |
| 3 | Resort Hotel | 0 | 13 | 2015 | July | 27 | 1 | 0 | 1 | 1 | ... | A | 0.0 | No Deposit | 0.0 | Transien |
| 4 | Resort Hotel | 0 | 14 | 2015 | July | 27 | 1 | 0 | 2 | 2 | ... | A | 0.0 | No Deposit | 0.0 | Transien |

5 rows × 30 columns

```
[ ] # Check for remaining missing values
    hotel=hotel.dropna(axis=0)
    hotel.isnull().sum()
```

```
hotel                    0
canceled                 0
total_cancel             0
year_arrival             0
month_arrival            0
week_arrival             0
date_arrival             0
weekend_stays            0
weekdays_stays           0
adults                   0
children                 0
babies                   0
meal                     0
country                  0
market_segment           0
booking_channel          0
repeat_guest             0
prior_cancellation       0
prior_noncancellation    0
reserved_room            0
assigned_room            0
booking_changes          0
deposit_type             0
waiting_days             0
customer_type            0
avg_dailyrate            0
parking_required         0
total_request            0
reservation_status       0
reservation_date         0
dtype: int64
```

There's no more missing value

## Checking Noisy Data

```
[ ] # Get the summary statistics for numerical variables
    hotel.describe()
```

| | canceled | total_cancel | year_arrival | week_arrival | date_arrival | weekend_stays | weekdays_stays |
|---|---|---|---|---|---|---|---|
| count | 43448.000000 | 43448.000000 | 43448.000000 | 43448.000000 | 43448.000000 | 43448.000000 | 43448.000000 |
| mean | 0.282384 | 89.403586 | 2016.024213 | 27.833663 | 15.763303 | 1.163575 | 3.044605 |
| min | 0.000000 | 0.000000 | 2015.000000 | 1.000000 | 1.000000 | 0.000000 | 0.000000 |
| 25% | 0.000000 | 11.000000 | 2015.000000 | 17.000000 | 8.000000 | 0.000000 | 1.000000 |
| 50% | 0.000000 | 55.000000 | 2016.000000 | 30.000000 | 16.000000 | 1.000000 | 3.000000 |
| 75% | 1.000000 | 146.000000 | 2017.000000 | 38.000000 | 23.000000 | 2.000000 | 5.000000 |
| max | 1.000000 | 737.000000 | 2017.000000 | 53.000000 | 31.000000 | 16.000000 | 40.000000 |
| std | 0.450164 | 94.781167 | 0.760417 | 13.541002 | 8.790938 | 1.128543 | 2.390346 |

| | adults | children | babies | repeat_guest | prior_cancellation | prior_noncancellation | bo |
|---|---|---|---|---|---|---|---|
| count | 43448.000000 | 43448.000000 | 43448.000000 | 43448.000000 | 43448.000000 | 43448.000000 | |
| mean | 1.869752 | 0.123642 | 0.013280 | 0.040830 | 0.092870 | 0.118532 | |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| 25% | 2.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| 50% | 2.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| 75% | 2.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| max | 55.000000 | 10.000000 | 2.000000 | 1.000000 | 26.000000 | 30.000000 | |
| std | 0.677763 | 0.437757 | 0.116071 | 0.197899 | 1.281947 | 0.893062 | |

The mean time from making a reservation to the actual arrival is approximately 104 days varying from 0-737 days. This points out that it is normal for guests to make their bookings in advance yet some may book at the very last minute while others may book their rooms as early as two years in advance.

On the issue of the timing of arrivals, the mean week number of arrival is equal to approximately 27. 17 with the minimum of 1 and the maximum of 53. From here, it can be deduced that there is a fairly balanced distribution of arrivals all year but slightly skewed towards the mid-year. Further, the mean day of the month of arrival is roughly 15. 80, to 31 which means that the arrivals are evenly possible throughout the month.

Looking at durations of stays, guests usually stay for an average of 0 nights. 93 for the weekends with some guests spending up to a maximum of 19 weekends at one time. It can be observed that during weekdays, the average length of stay in the hotel is about 2 nights. 5 nights, and can range from 0 to as much as 50 weeknights. This shows that most of the stays are minute, but there are instances of a long-term stay once in a while.

The average number of adults per booking is about 1.86, and bookings range from 0 to 55 adults. This wide range indicates that, while the majority of bookings are for small groups or solo passengers, there are certain cases where very big groups book together. The average number of children per booking is roughly 0.1, with a maximum of 10 children, while the average number of infants per reservation is extremely low, around 0.008, with a maximum of 10 infants. This suggests that families with children or newborns are more uncommon than adult-only bookings.

The average number of adults per booking is 1.86, with bookings ranging from 0 to 55 adults. This wide range demonstrates that, while the majority of bookings are for small groups or solo travellers, there are certain cases where large groups book together. The average number of children per booking is approximately 0.1, with a maximum of 10 children, whilst the average number of infants per reservation is extremely low, around 0.008, with a maximum of 10 babies. This shows that bookings for families with children or babies are less common than those for adults solo.

The average daily rate is about 101.83, with rates ranging from -6.38 (probably due to errors or exceptional situations) to 5400. This vast range reflects a wide range of room rates, which could reflect different room kinds and service levels. Parking needs are insignificant, with an average of 0.06 parking spots per ticket and a maximum of eight spots.

Finally, the average number of special requests made by guests is 0.57, with some guests making as many as five. This suggests that, while most guests have few specific needs, there are infrequent appointments with many requests.

```
[ ] noisy_data = {
        'avg_dailyrate':   hotel[hotel['avg_dailyrate'] < 0],
        'adults':   hotel[hotel['adults'] == 0],
        'children': hotel[hotel['children'] == 10],
        'babies':   hotel[hotel['babies'] == 10],
    }

    noisy_data_count = {key: len(value) for key, value in noisy_data.items()}
    noisy_data_count
```

```
{'avg_dailyrate': 1, 'adults': 23, 'children': 1, 'babies': 0}
```

There is one booking with a negative Average Daily Rate (ADR), which appears to be an error or a rare situation, as a negative rate does not make sense. This shows that there was an error in how the data was entered or something odd about that particular booking.

There are 403 bookings with zero adults. This could be the result of a data entering error, as it is unusual that a room would be rented without any adults. However, there may be rare occasions when just children or babies are specified in the booking.

One booking in the data included ten children, which is very high. Since there are usually not this many kids in a single booking, this could be an outlier or another data entry issue.

Another booking that appears abnormally high and may be another error or outlier is one that has ten babies on it. These rare instances highlight the need for a closer look to ensure the accuracy of the booking information.

```python
# Replace negative adr with median of adr column
hotel.loc[hotel['avg_dailyrate'] < 0, 'avg_dailyrate'] = hotel['avg_dailyrate'].median()

# Remove rows with 0 adults
hotel = hotel[hotel['adults'] != 0]

# Remove rows with 10 children or 10 babies
hotel = hotel [hotel ['children'] != 10]
hotel  = hotel [hotel ['babies'] != 10]

# Reset the index
hotel .reset_index(drop=True, inplace=True)

# Check if the noisy data has been handled
noisy_data_handled = {
    'avg_dailyrate': hotel [hotel ['avg_dailyrate'] < 0],
    'adults': hotel [hotel ['adults'] == 0],
    'children': hotel [hotel ['children'] == 10],
    'babies': hotel [hotel ['babies'] == 10],
}

noisy_data_handled_count = {key: len(value) for key, value in noisy_data_handled.items()}
noisy_data_handled_count
```

```
{'avg_dailyrate': 0, 'adults': 0, 'children': 0, 'babies': 0}
```

Since there is just one booking with a negative Average Daily Rate (ADR), the mean or median ADR should be used in its place. With this modification, the distribution as a whole won't be greatly impacted while maintaining data consistency.

It is unlikely and probably implies errors that there were 403 bookings for adults in the adults column with 0 adults. Removing these rows would be a reasonable strategy to preserve data accuracy because this is a minor percentage of the dataset.

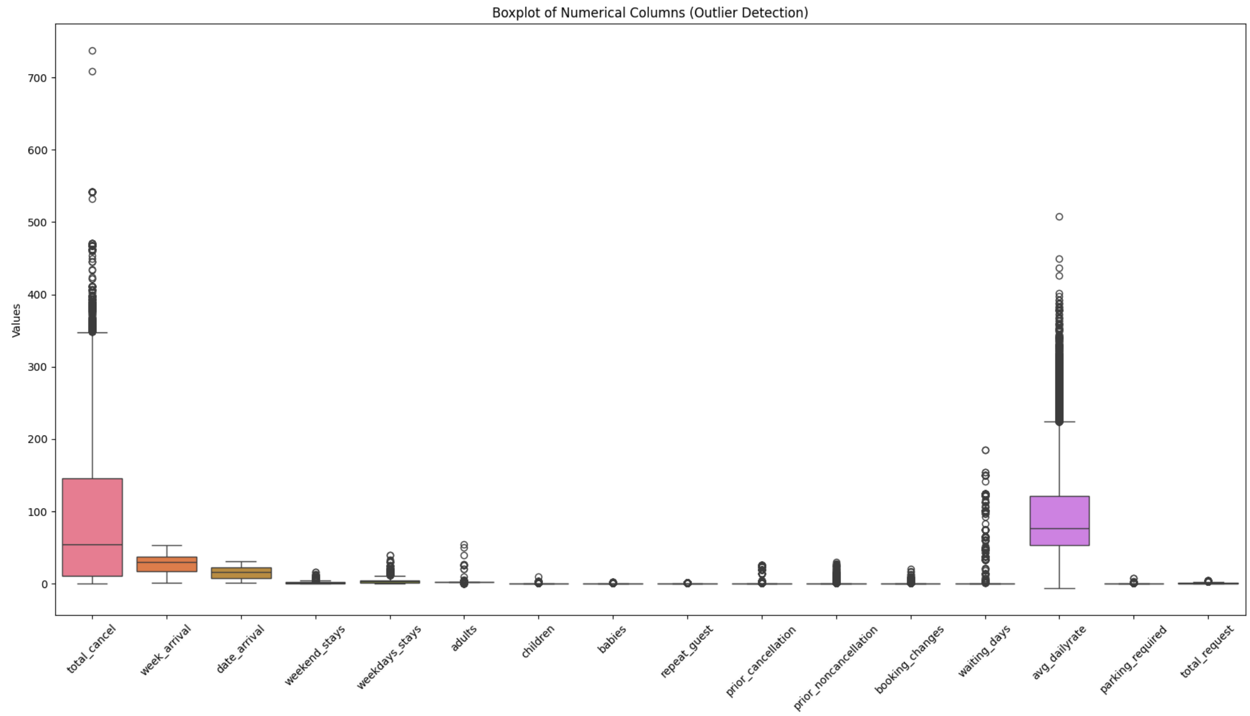
There is one reservation with ten children in the children column, which seems to be an outlier. The dataset would remain more representative and accurate if this one column were removed.

Similarly, one booking in the babies column has ten babies, which also seems like an outlier. Eliminating this column would contribute to maintaining the accuracy and dependability of the data.

## Checking Outlier

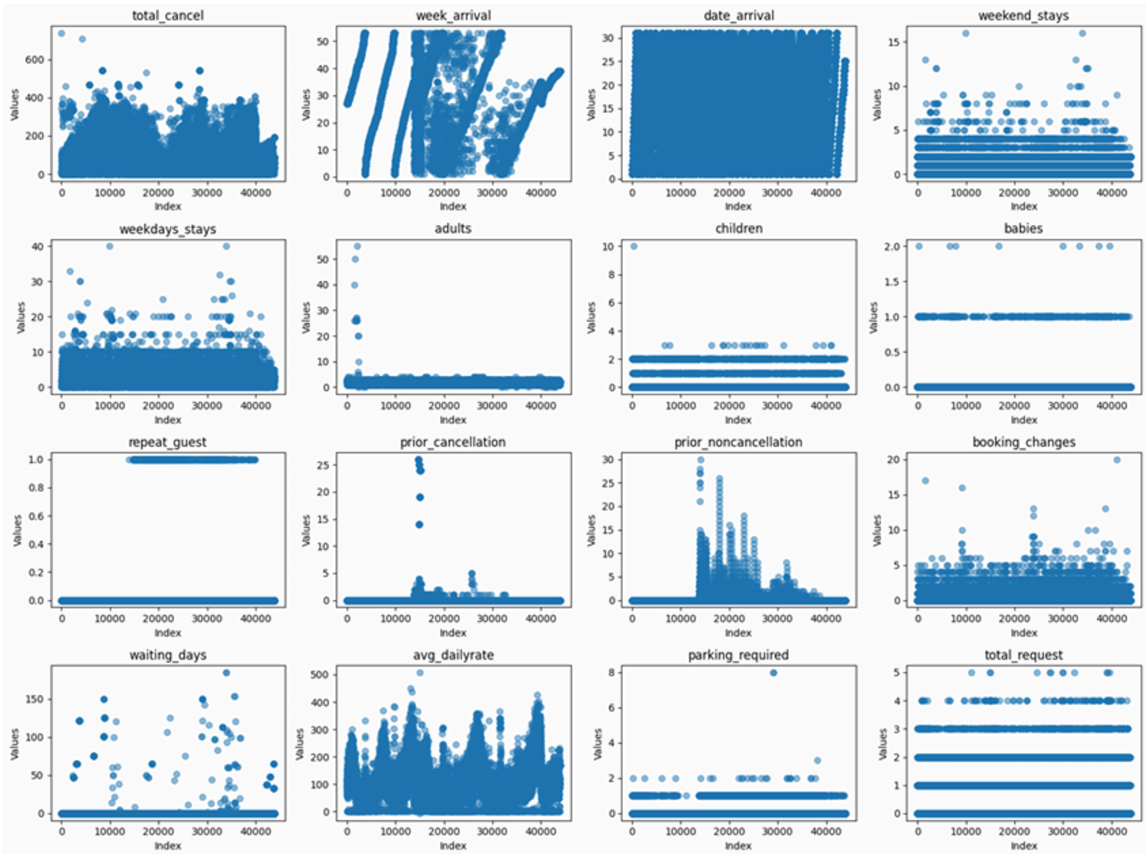
- Boxplot

```
[ ] # Selecting numerical columns
    numerical_columns = ['total_cancel', 'week_arrival',
                         'date_arrival', 'weekend_stays', 'weekdays_stays', 'adults',
                         'children', 'babies', 'repeat_guest', 'prior_cancellation',
                         'prior_noncancellation', 'booking_changes', 'waiting_days',
                         'avg_dailyrate', 'parking_required', 'total_request']

    # Boxplot
    plt.figure(figsize=(20, 10))
    sns.boxplot(data=hotel[numerical_columns], orient='v')  # Changed orient to 'v' for vertical
    plt.title('Boxplot of Numerical Columns (Outlier Detection)')
    plt.ylabel('Values')  # Changed from xlabel to ylabel since it's now vertical
    plt.xticks(rotation=45)  # Rotating x-axis labels for better readability
    plt.show()
```



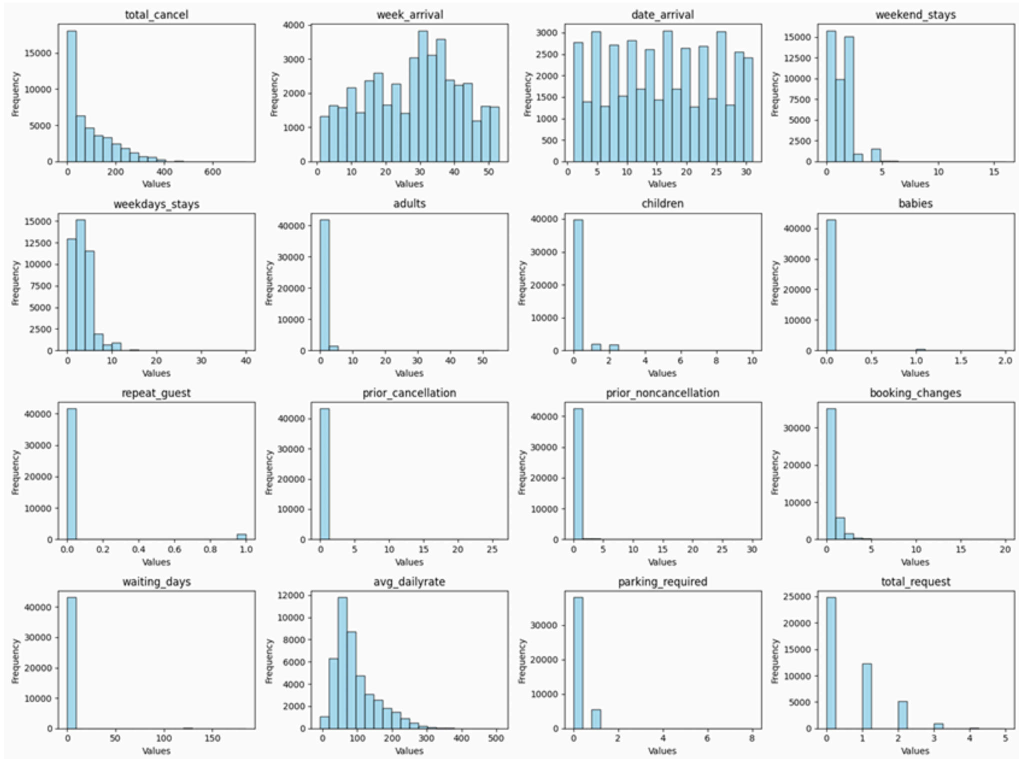

- Scatter plot

```
[ ] # Selecting numerical columns
    numerical_columns = ['total_cancel', 'week_arrival',
                         'date_arrival', 'weekend_stays', 'weekdays_stays', 'adults',
                         'children', 'babies', 'repeat_guest', 'prior_cancellation',
                         'prior_noncancellation', 'booking_changes', 'waiting_days',
                         'avg_dailyrate', 'parking_required', 'total_request']

    # Scatter plots for each numerical column
    plt.figure(figsize=(16, 12))
    for i, column in enumerate(numerical_columns, 1):
        plt.subplot(4, 4, i)
        plt.scatter(hotel.index, hotel[column], alpha=0.5)
        plt.title(column)
        plt.xlabel('Index')
        plt.ylabel('Values')
    plt.tight_layout()
    plt.show()
```




- Histogram

```
[ ]  # Selecting numerical columns
     numerical_columns = ['total_cancel', 'week_arrival',
                          'date_arrival', 'weekend_stays', 'weekdays_stays', 'adults',
                          'children', 'babies', 'repeat_guest', 'prior_cancellation',
                          'prior_noncancellation', 'booking_changes', 'waiting_days',
                          'avg_dailyrate', 'parking_required', 'total_request']

     # Histograms for each numerical column
     plt.figure(figsize=(16, 12))
     for i, column in enumerate(numerical_columns, 1):
         plt.subplot(4, 4, i)
         plt.hist(hotel[column], bins=20, color='skyblue', edgecolor='black', alpha=0.7)
         plt.title(column)
         plt.xlabel('Values')
         plt.ylabel('Frequency')
     plt.tight_layout()
     plt.show()
```




In this step, we identify outliers using box plot, scatter plot and histogram visualization. Outliers can be detected by identifying points that are away from the majority of data points. We carefully evaluate these outliers to see if they are actual data points or errors. If they are valid, they may provide fascinating insights or usual patterns in the data. However, if they contain errors or noise, we may need to remove or correct them to ensure the accuracy of our research. After the discussion with our group members, we decided not to remove the outliers data to know about the unusual patterns in datasets. This will allow us to require interesting insight about why this happened.

## Checking duplicate data

```
[ ] hotel.duplicated()
```

```
0        False
1        False
2        False
3        False
4        False
         ...
58297     True
58298     True
58299     True
58300     True
58301     True
Length: 58302, dtype: bool
```

The df. duplicated() function specifies whether or not each row in the dataset is duplicate. Each Series value corresponds to a row in the dataset, and False indicates that the row is not duplicated.

## Change Data Type

```
[ ] hotel['reservation_date'] = pd.to_datetime(hotel['reservation_date'])
    hotel
```

| | hotel | canceled | total_cancel | year_arrival | month_arrival | week_arrival | date_arrival | weekend_stays | weekdays_stays | adults | ... | assigned_room | booking_changes | deposit_type | waiting_days | customer_ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Resort Hotel | 0 | 342 | 2015 | July | 27 | 1 | 0 | 0 | 2 | ... | C | 3 | No Deposit | 0.0 | Tra |
| 1 | Resort Hotel | 0 | 737 | 2015 | July | 27 | 1 | 0 | 0 | 2 | ... | C | 4 | No Deposit | 0.0 | Tra |
| 2 | Resort Hotel | 0 | 7 | 2015 | July | 27 | 1 | 0 | 1 | 1 | ... | C | 0 | No Deposit | 0.0 | Tra |
| 3 | Resort Hotel | 0 | 13 | 2015 | July | 27 | 1 | 0 | 1 | 1 | ... | A | 0 | No Deposit | 0.0 | Tra |
| 4 | Resort Hotel | 0 | 14 | 2015 | July | 27 | 1 | 0 | 2 | 2 | ... | A | 0 | No Deposit | 0.0 | Tra |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 58297 | City Hotel | 1 | 605 | 2016 | October | 43 | 17 | 1 | 2 | 2 | ... | A | 0 | Non Refund | 0.0 | Tra |
| 58298 | City Hotel | 1 | 605 | 2016 | October | 43 | 17 | 1 | 2 | 2 | ... | A | 0 | Non Refund | 0.0 | Tra |
| 58299 | City Hotel | 1 | 605 | 2016 | October | 43 | 17 | 1 | 2 | 2 | ... | A | 0 | Non Refund | 0.0 | Tra |
| 58300 | City Hotel | 1 | 605 | 2016 | October | 43 | 17 | 1 | 2 | 2 | ... | A | 0 | Non Refund | 0.0 | Tra |
| 58301 | City Hotel | 1 | 605 | 2016 | October | 43 | 17 | 1 | 2 | 2 | ... | A | 0 | Non Refund | 0.0 | Tra |

58302 rows × 30 columns

```
[ ] hotel['children'] = hotel['children'].fillna(0)

    # Then, convert the 'children' column from float to int
    hotel['children'] = hotel['children'].astype(int)

    # Verify the change
    print(hotel['children'].dtype)
```

Changing the datatype of the 'reservation_date' column to datetime is important because it allows for accurate date calculations. Similarly, changing the 'children' column from float to int ensures data integrity by representing the number of children as whole numbers.

## 4.0 Exploratory Data Analysis

### 4.1 Bar Chart

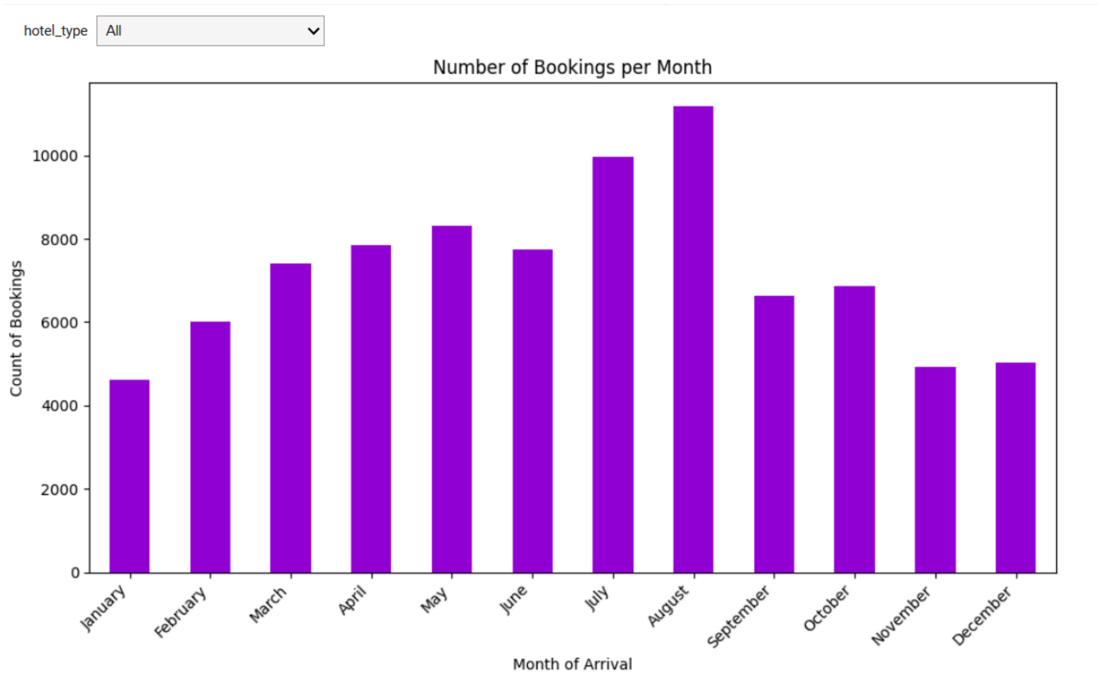



*Figure 4.1.1 Number of booking per Month*

Figure 1 shows the bar chart for the number of booking per month. This graph uses interactive visualization that will enhance user experience by allowing data filtering based on hotel type through a dropdown button. July and August have the highest number of bookings indicating that summer travel trend. This increase is likely due to parents taking their children on holiday during the school vacation period. The lowest month for booking hotels are November and December. This decrease may indicate the beginning of the school year and end of summer season. Other than that, some regions may experience cold weather that will discourage them from traveling. This pattern suggests that hotel bookings are significantly influenced by month because of the academic calendar in which families take advantage of school vacation to plan their trip.

Furthermore, the ability to analyze booking trends by month is crucial in discovering peak seasons and marketing strategy. For example, hotels can increase their workforce during the busy summer months of July and August to ensure that clients receive high-quality service despite the increase in number of guests. In contrast, during the off-peak months of November and December, hotels may consider giving special promotions or discounts to attract more guests and increase sales revenue. Thus, understanding the number of bookings helps companies to explore trends in specific months for planning marketing strategies.

## 4.2 Line Chart

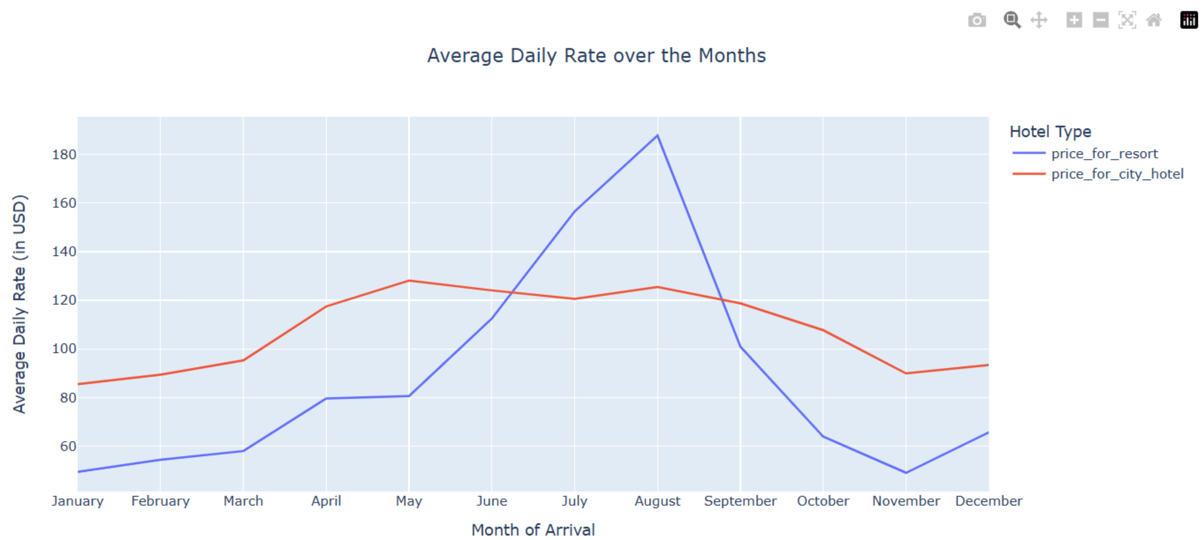



*Figure 4.2.1 Average Daily Rates by Month*

The line chart shows hotel prices across the months for the Resort Hotel and City Hotel. The highest price is in August for Resort Hotels, while City hotels' prices are consistent across the month. This indicates that the price for Resort Hotel are much higher during the summer while the prices of City Hotels vary less and have the most expensive during spring and autumn.

The increase price for Resort Hotel during the summer peak may due to the increase demand for vacation and leisure travel. However, the price decrease during the winter and autumn due to low demand for resort stay. Thus the occupancy rates is low and reduced the price. Due to business travel or events the City Hotel prices are stable throughout the year and high during spring and autumn. They have less variation but consistent demand compared to Resort Hotel. Since the continuous business travel and urban tourism likely happen during winter, the price rate are moderate.  In conclusion, understanding the average daily rate help hotel management to  manage the operations systems and improve the marketing so that the prices will increase and get the maximized profits along the year.

4.3 Side by Side Chart

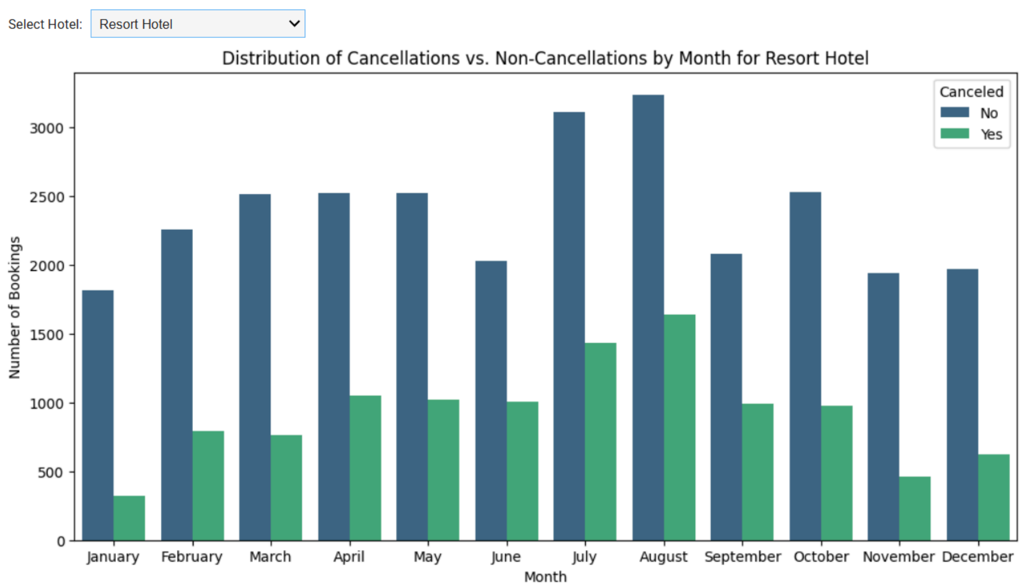



*Figure 4.3.1  Distribution of Cancellation vs Non-Cancellation by Month for Resort Hotel*

Based on Figure 3, the monthly distribution of bookings and cancellations for a Resort Hotel. Peak months like July and August show the highest number of bookings but also significant cancellations, likely due to overbooking or last-minute travel changes. Shoulder seasons, such as March to May and September to October, have steady bookings with moderate cancellations, suggesting that guests during these periods are more likely to keep their reservations. Off-peak months, including January and November, have fewer bookings and lower cancellation rates, while December sees increased bookings with notable cancellations, possibly due to changes in holiday travel plans.To reduce cancellations and improve customer satisfaction, the hotel may consider introducing methods like as stricter cancellation rules during peak months. Overall, understanding seasonal booking and cancellation patterns is critical for the Resort Hotel's operational efficiency and guest satisfaction.

## 4.4 Pie Chart

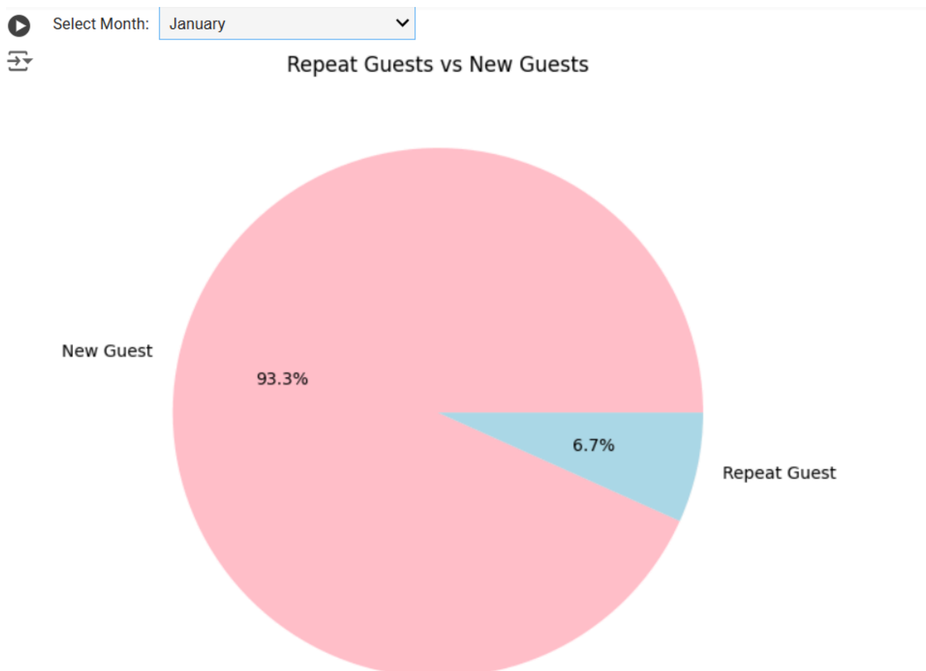



*Figure 4.4,1 Percentages of new guest and repeat guest*

By analyzing the trend of repeat guests versus new guests each month using the interactive pie chart , we can identify trends in guest loyalty. In January, 93.3% of visitors are new, compared to 6.7% who are repeated, which is that most visitors are new. There are a number of factors that may in some way be associated with this pattern. January comes immediately after the holiday period and there could be many new people who have been visiting the place for the holidays. If the accommodation is targeting tourists who would visit the hotel in winter time because of promotions or to visit other regions with unfavorable climate, the increase would be attributed to winter travels.

Furthermore, another weak point of market entries can be found in the low frequency of repeaters among guests. The following are some measures that can be put in place to increase customers. Expansion of traditions or addition of new include with loyalty programs may encourage people to visit again since they will be provided with some sort of reward or further discount on their next visit. For the problem of low repeat guests, businesses can adopt effective measures to apply good loyalty building and follow-up programs. Occasionally even identify percentage of repeat and new guest by month aids the businesses in taking the most appropriate decisions to improve the guest satisfaction and in turn mass marketing to increase the ratio of more and more repeat customers.

4.5 Stacked Bar Chart

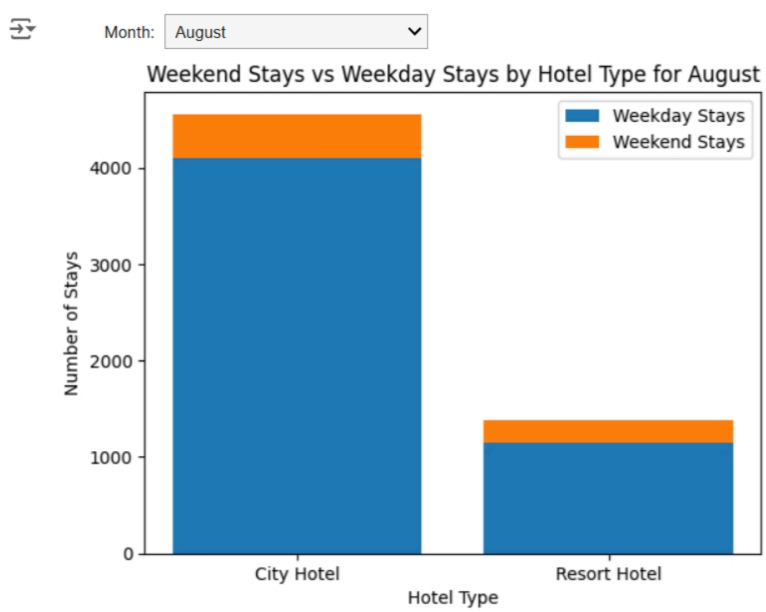



*Figure 4.5.1 Stacked Bar Chart of Weekdays vs Weekend Stays*

City hotels are likely to attract more customers compared to resort hotels because of the convenient location of the city hotel which is located near to easy access facilities like public transportation, cultural attraction and that will cater to travelers need. Tourists often choose city hotel during weekend especially during holiday season, while business traveler fill the room during weekdays for business purposes. Therefore, the number of customers during weekdays at city hotel is higher compared to resort hotel. By analyzing the number of customers for both hotel during weekend and weekday, which city hotel might have other interesting attractions or facilities that attract more customers during weekdays and weekends like special discounts or valuable packages. Therefore, Resort Hotel might have to offer good deals during the weekday or holiday season to attract more tourist and guest. By understanding the pattern of number of customers during weekdays and weekend, it allows hotel to develop specific strategies to improve the occupancy and hotel throughout the week.

## 5.0 Summary

In conclusion, this hotel booking analysis aims to improve hotel administration strategies. Analyzing hotel booking data can gain a comprehensive insight into guest behaviour toward booking patterns. In addition, this project investigates the variation in the average daily rates across different months and compares booking patterns during weekends and weekdays. This insight can help the hotel company optimize pricing strategies, marketing, and resource allocation, hence improving customer satisfaction and revenue.

This project involves cleaning the raw dataset to prepare comprehensive data. It is important to handle outliers and check for missing values for an accurate and consistent analysis. From the analysis, we can identify the factors that influence hotel bookings. All of the analysis insight helps the hotel develop strategies to improve occupancy and revenue throughout the year.

## Appendix

Project Wrangling.ipynb

## References


Hotel Engine. (2022, January 4). Hotel room pricing: Why it fluctuates + how to score a deal. Hotel Engine. Retrieved from https://www.hotelengine.com/business-travel-guide/hotel-room-pricing

Mostipak, J. (2019). *Hotel booking demand* [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand/data