Hotel Booking Project

TRỊNH TRÀ VINH

Let's Analyse Dataset!

Data Exploration

- Observe the Data
- Find Missing

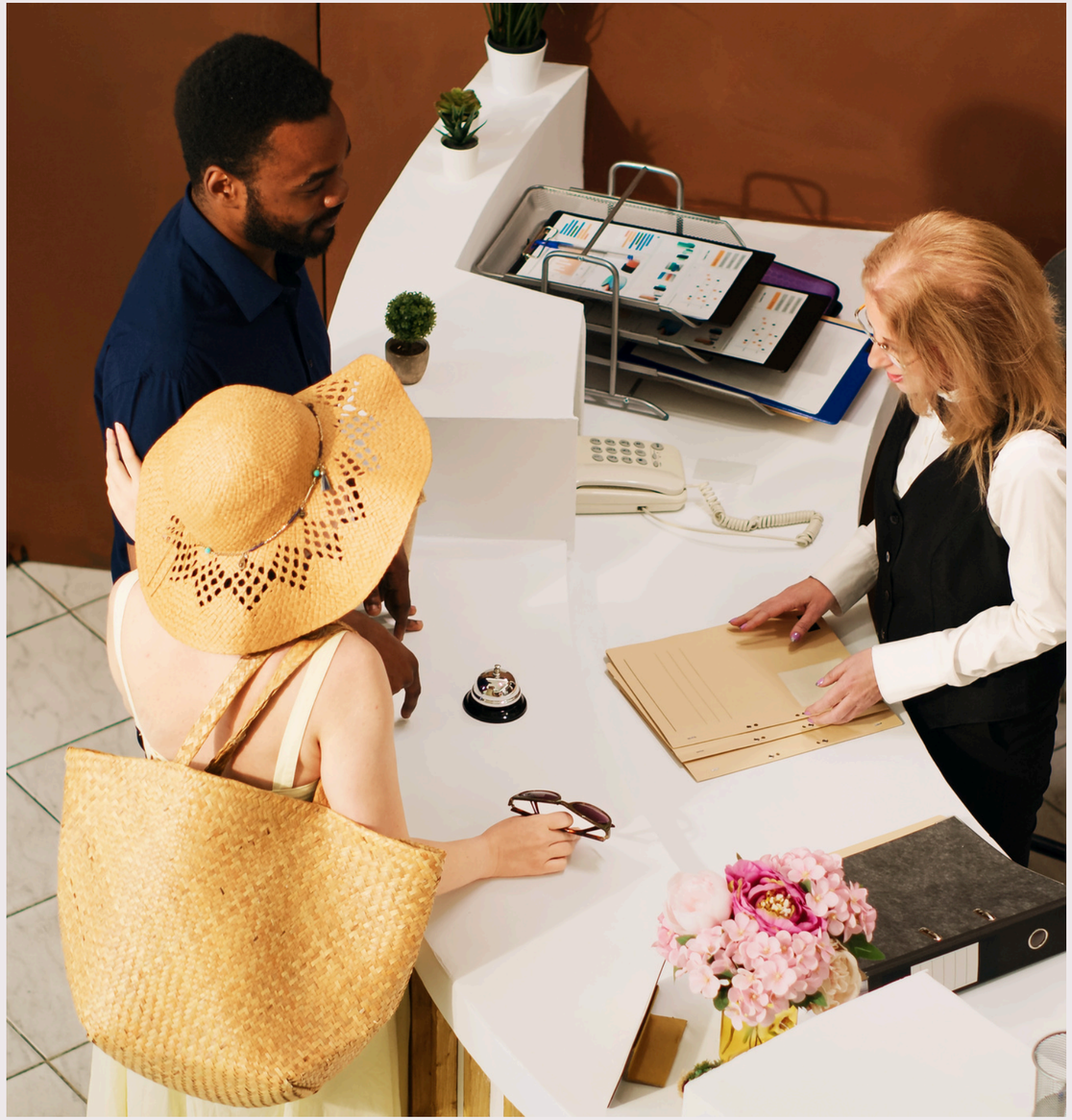
Data Cleaning

- Replace the Null Values
- Drop un-necessary columns

Analyse the data

Visualise the data

Conclusion



Observe the Data

```
[ ] # Read data from file csv
    hotel_bookings = pd.read_csv("/content/drive/My Drive/Data Analyst/hotel_bookings.csv")
    # Print data
    hotel_bookings.head(10)
```

Hiện kết quả đã ẩn

```
[ ] #Find shape of data file
    hotel_bookings.shape
```

```
(119390, 32)
```

```
# info of each column in data
hotel_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   arrival_date_year               119390 non-null  int64
 4   arrival_date_month              119390 non-null  object
 5   arrival_date_week_number        119390 non-null  int64
 6   arrival_date_day_of_month       119390 non-null  int64
 7   stays_in_weekend_nights         119390 non-null  int64
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
```

Drop Duplicate Row

FINDING DUPLICATE ROW

```
[ ]  print(hotel_bookings.duplicated().sum())

     31994
```

DROP DUPLICATE DATA

```
[ ]  # Drop duplicate data
     hotel_bookings.drop_duplicates(inplace=True)
```

RESULT

```
[ ]  print(hotel_bookings.duplicated().sum())

     0
```

Finding Missing

```
[11] # Count data is null in each column in data
     # Only show column have missing values != 0
     null_counts = hotel_bookings.isnull().sum().reset_index()
     null_counts.columns = ['Column', 'MissingValues']
     have_null_value_columns = null_counts[null_counts['MissingValues'] != 0]
     have_null_value_columns
```

	Column	MissingValues
10	children	4
13	country	488
23	agent	16340
24	company	112593

CHECK FOR COMPANY

As we can see, Company column has too many null value

```
hotel_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87396 entries, 0 to 119389
Data columns (total 29 columns):
```

21	booking_changes	87396 non-null	int64
22	deposit_type	87396 non-null	object
23	agent	75203 non-null	float64
24	company	5259 non-null	float64
25	days_in_waiting_list	87396 non-null	int64
26	customer_type	87396 non-null	object
27	adr	87396 non-null	float64

CHECK FOR COMPANY

It appears that the Company column is **not particularly significant** and could be removed from the dataset.

```
hotel_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87396 entries, 0 to 119389
Data columns (total 29 columns):
```

```
hotel_bookings.drop('company',inplace=True, axis=1)
hotel_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 87396 entries, 0 to 119389
Data columns (total 28 columns):
```

RESOLVE OTHERS

For the children and agent features we will use the median value to replace them

```
[94] hotel_bookings['children'] = hotel_bookings['children'].fillna(hotel_bookings['children'].median()
     hotel_bookings['agent'] = hotel_bookings['agent'].fillna(hotel_bookings['agent'].median())
```

For the country feature we are going to use the mode method, which will replace the missing values with the most frequent one.

```
# find which country is more frequent in our data
print(hotel_bookings['country'].mode())
```

```
0    PRT
Name: country, dtype: object
```

```
[96] hotel_bookings.fillna({'country': hotel_bookings['country'].mode()[0]}, inplace=True)
```

CHECK RESULT OF CLEAN MISSING VALUE

The dataset is now complete and contains no missing values.

```
[108] null_counts = hotel_bookings.isnull().sum().reset_index()
      null_counts.columns = ['Column', 'MissingValues']
      print(null_counts[null_counts['MissingValues'] != 0].size)
```

```
0
```

CHECK FOR MISWRITINGS

As we can see, the dataset has no miswritings

```
] df = pd.DataFrame(hotel_bookings, columns=["hotel", "arrival_date_month","arrival_date_year", "country", "market_segment", "distribution_channel", "reserved_roo
  for col in df:
     print(col, df[col].unique())
```

```
hotel ['Resort Hotel' 'City Hotel']
arrival_date_month [nan]
arrival_date_year [nan]
country ['PRT' 'GBR' 'USA' 'ESP' 'IRL' 'FRA' 'ROU' 'NOR' 'OMN' 'ARG' 'POL' 'DEU'
 'BEL' 'CHE' 'CN' 'GRC' 'ITA' 'NLD' 'DNK' 'RUS' 'SWE' 'AUS' 'EST' 'CZE'
 'BRA' 'FIN' 'MOZ' 'BWA' 'LUX' 'SVN' 'ALB' 'IND' 'CHN' 'MEX' 'MAR' 'UKR'
 'SMR' 'LVA' 'PRI' 'SRB' 'CHL' 'AUT' 'BLR' 'LTU' 'TUR' 'ZAF' 'AGO' 'ISR'
 'CYM' 'ZMB' 'CPV' 'ZWE' 'DZA' 'KOR' 'CRI' 'HUN' 'ARE' 'TUN' 'JAM' 'HRV'
 'HKG' 'IRN' 'GEO' 'AND' 'GIB' 'URY' 'JEY' 'CAF' 'CYP' 'COL' 'GGY' 'KWT'
 'NGA' 'MDV' 'VEN' 'SVK' 'FJI' 'KAZ' 'PAK' 'IDN' 'LBN' 'PHL' 'SEN' 'SYC'
 'AZE' 'BHR' 'NZL' 'THA' 'DOM' 'MKD' 'MYS' 'ARM' 'JPN' 'LKA' 'CUB' 'CMR'
 'BIH' 'MUS' 'COM' 'SUR' 'UGA' 'BGR' 'CIV' 'JOR' 'SYR' 'SGP' 'BDI' 'SAU'
 'VNM' 'PLW' 'QAT' 'EGY' 'PER' 'MLT' 'MWI' 'ECU' 'MDG' 'ISL' 'UZB' 'NPL'
 'BHS' 'MAC' 'TGO' 'TWN' 'DJI' 'STP' 'KNA' 'ETH' 'IRQ' 'HND' 'RWA' 'KHM'
 'MCO' 'BGD' 'IMN' 'TJK' 'NIC' 'BEN' 'VGB' 'TZA' 'GAB' 'GHA' 'TMP' 'GLP'
 'KEN' 'LIE' 'GNB' 'MNE' 'UMI' 'MYT' 'FRO' 'MMR' 'PAN' 'BFA' 'LBY' 'MLI'
 'NAM' 'BOL' 'PRY' 'BRB' 'ABW' 'AIA' 'SLV' 'DMA' 'PYF' 'GUY' 'LCA' 'ATA'
 'GTM' 'ASM' 'MRT' 'NCL' 'KIR' 'SDN' 'ATF' 'SLE' 'LAO']
market_segment ['Direct' 'Corporate' 'Online TA' 'Offline TA/TO' 'Complementary' 'Groups'
 'Undefined' 'Aviation']
distribution_channel ['Direct' 'Corporate' 'TA/TO' 'Undefined' 'GDS']
reserved_room_type ['C' 'A' 'D' 'E' 'G' 'F' 'H' 'L' 'P' 'B']
assigned_room_type ['C' 'A' 'D' 'E' 'G' 'F' 'I' 'B' 'H' 'P' 'L' 'K']
deposit_type ['No Deposit' 'Refundable' 'Non Refund']
customer_type ['Transient' 'Contract' 'Transient-Party' 'Group']
reservation_status ['Check-Out' 'Canceled' 'No-Show']
```

CHECK FOR LOGICAL ERRORS

We have three columns (children, adults, babies) that can't be zero all the them at the same row.

```
[99] count_0 = hotel_bookings[(hotel_bookings.children == 0) & (hotel_bookings.adults == 0) & (hotel_bookings.babies == 0)].shape[0]
     print(count_0)
```

```
166
```

Drop Rows with logical errors

```
[100] logic_error = hotel_bookings[(hotel_bookings['children'] == 0) & (hotel_bookings['adults'] == 0) & (hotel_bookings['babies'] == 0)].index
      hotel_bookings.drop(logic_error , inplace=True)

      hotel_bookings.reset_index(drop=True, inplace=True)
      hotel_bookings.head()
```

Check again if there's still a logical error

```
[101] count_0 = hotel_bookings[(hotel_bookings.children == 0) & (hotel_bookings.adults == 0) & (hotel_bookings.babies == 0)].shape[0]
      print(count_0)
```

```
0
```

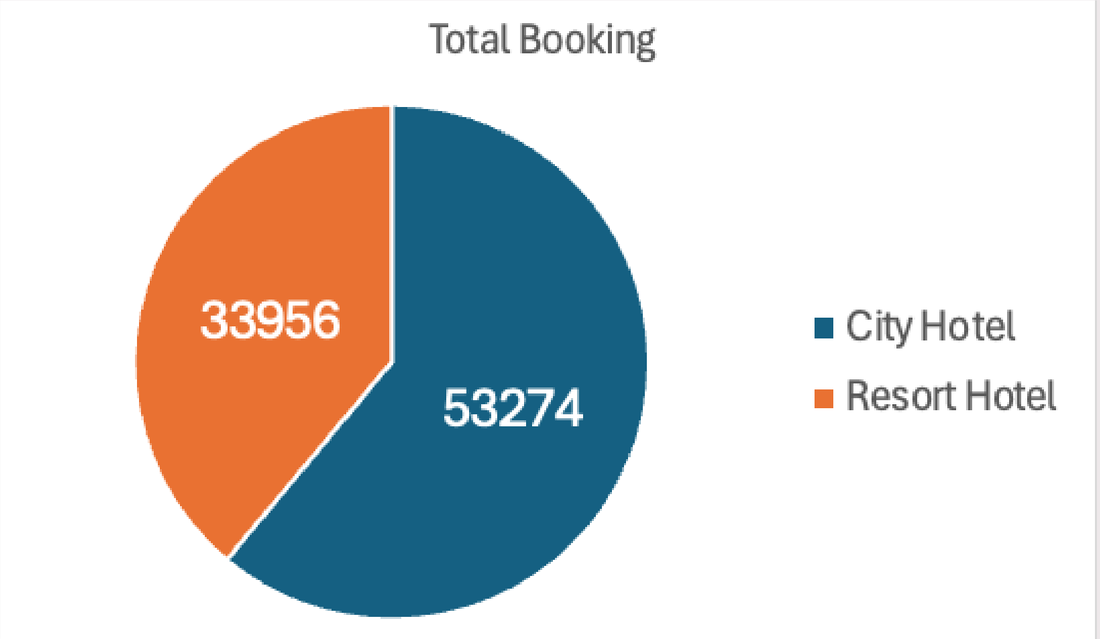
ANALYSE THE DATA

📊 **Insight:**

- City Hotel has 53,274 bookings (~61% of total).
- Resort Hotel has 33,956 bookings (~39% of total).
- City Hotels seem to be the preferred choice for travelers, possibly due to convenience and business-related travel.

📌 **Actionable Insight:**

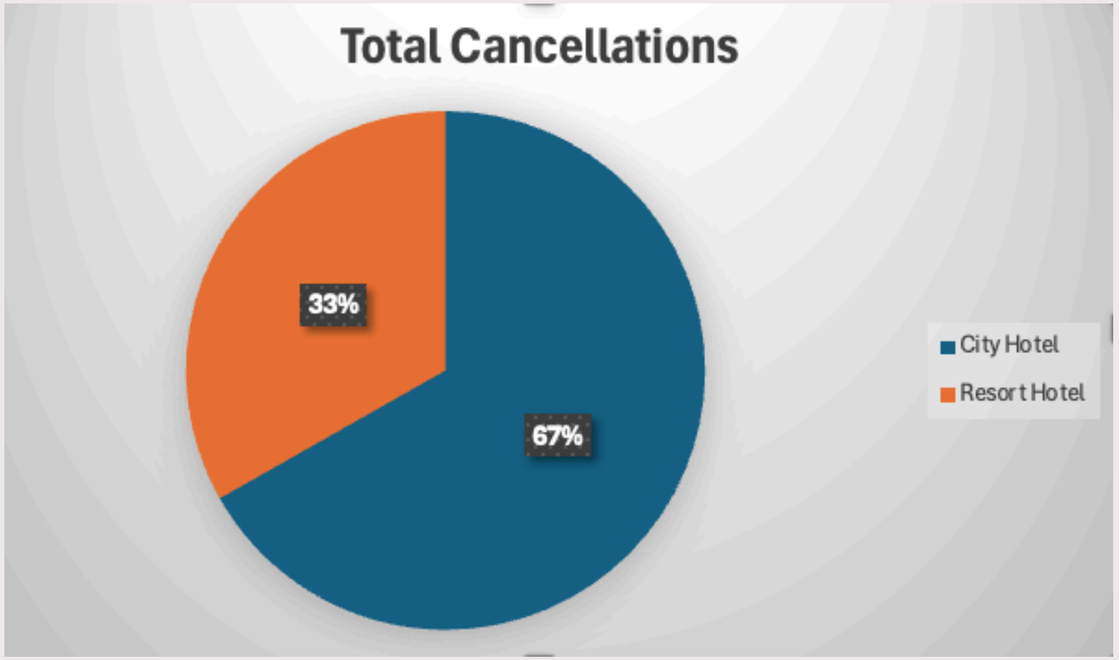
- If targeting growth, Resort Hotels may need additional marketing strategies or promotions to attract more bookings.



ANALYSE THE DATA

📊 **Insight:**

- City Hotel has 16,035 cancellations (~66.8% of total cancellations).
- Resort Hotel has 7,974 cancellations (~33.2% of total cancellations).
- The higher cancellation rate in City Hotels might be due to business travelers making last-minute changes.



📌 **Actionable Insight:**

- Hotels can introduce stricter cancellation policies or offer flexible date modifications to retain bookings

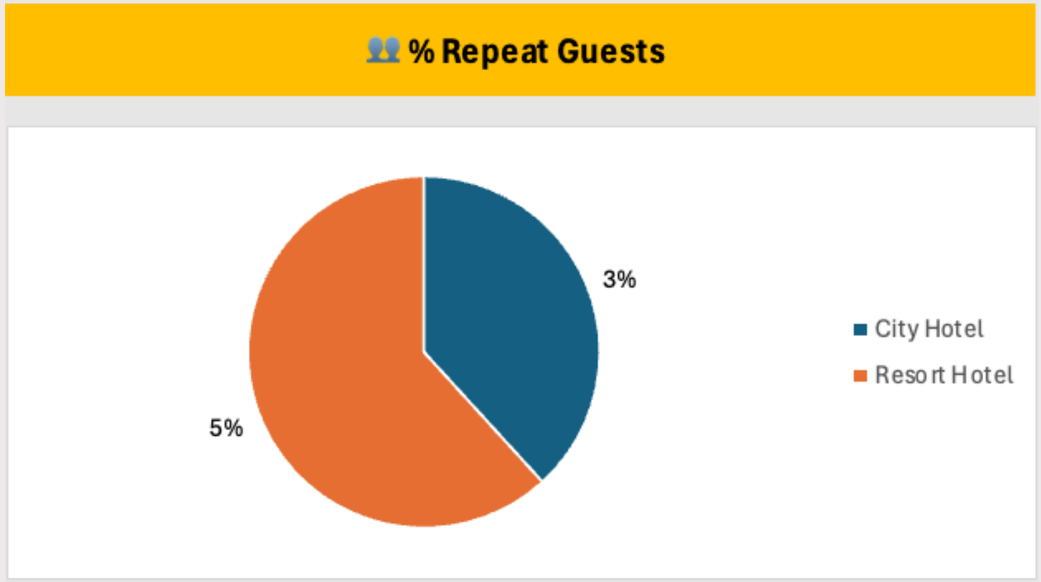
ANALYSE THE DATA

📊 **Insight:**

- Only 3% of City Hotel guests are repeat guests, while 5% of Resort Hotel guests are repeat customers.
- Overall, only 4% of guests return, which is relatively low.

📌 **Actionable Insight:**

- Implement loyalty programs or exclusive discounts for returning guests to increase repeat visits.



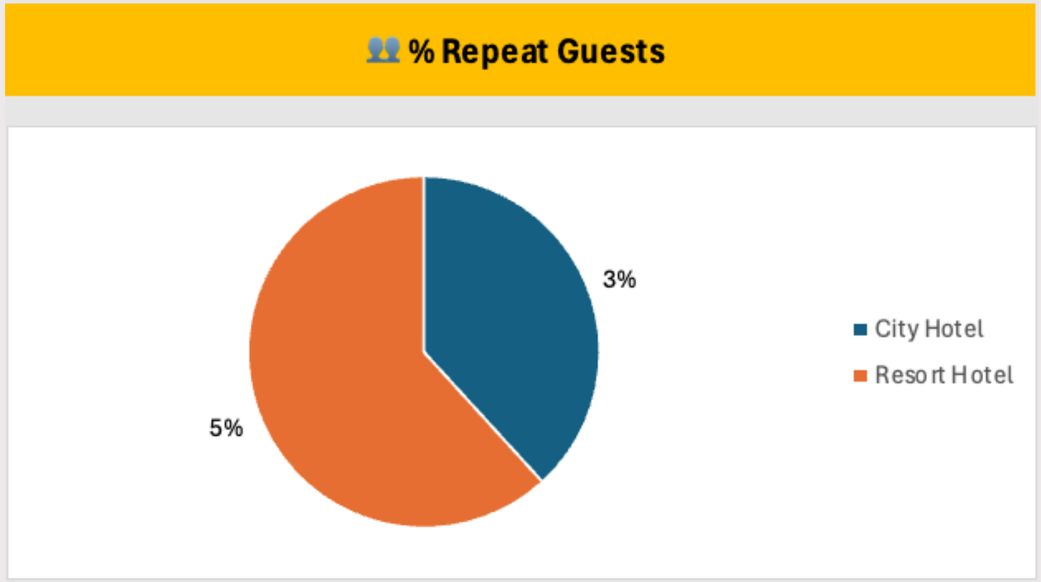
ANALYSE THE DATA

📊 **Insight:**

- Only 3% of City Hotel guests are repeat guests, while 5% of Resort Hotel guests are repeat customers.
- Overall, only 4% of guests return, which is relatively low.

📌 **Actionable Insight:**

- Implement loyalty programs or exclusive discounts for returning guests to increase repeat visits.



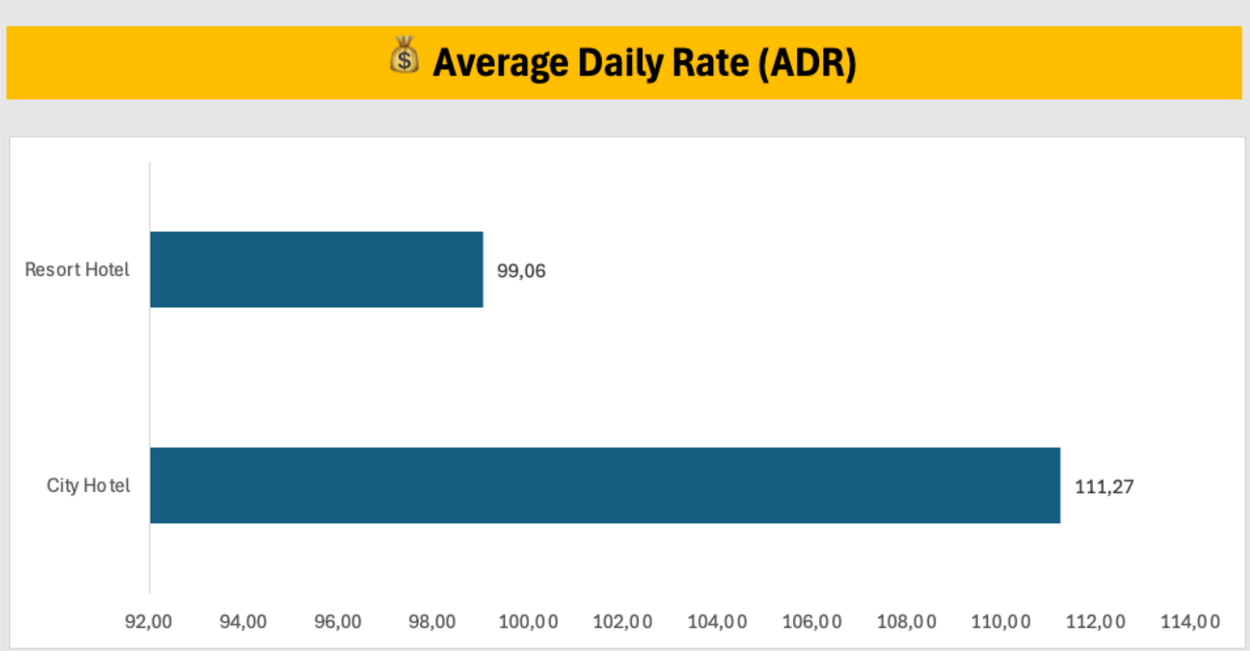
ANALYSE THE DATA

📊 **Insight:**

- City Hotels have a higher ADR ($111.27) compared to Resort Hotels ($99.06).
- Overall ADR: $106.52
- This suggests that City Hotels might be pricing higher due to demand from business travelers.

📌 **Actionable Insight:**

- Resort Hotels might explore dynamic pricing to increase ADR, especially during peak travel seasons.



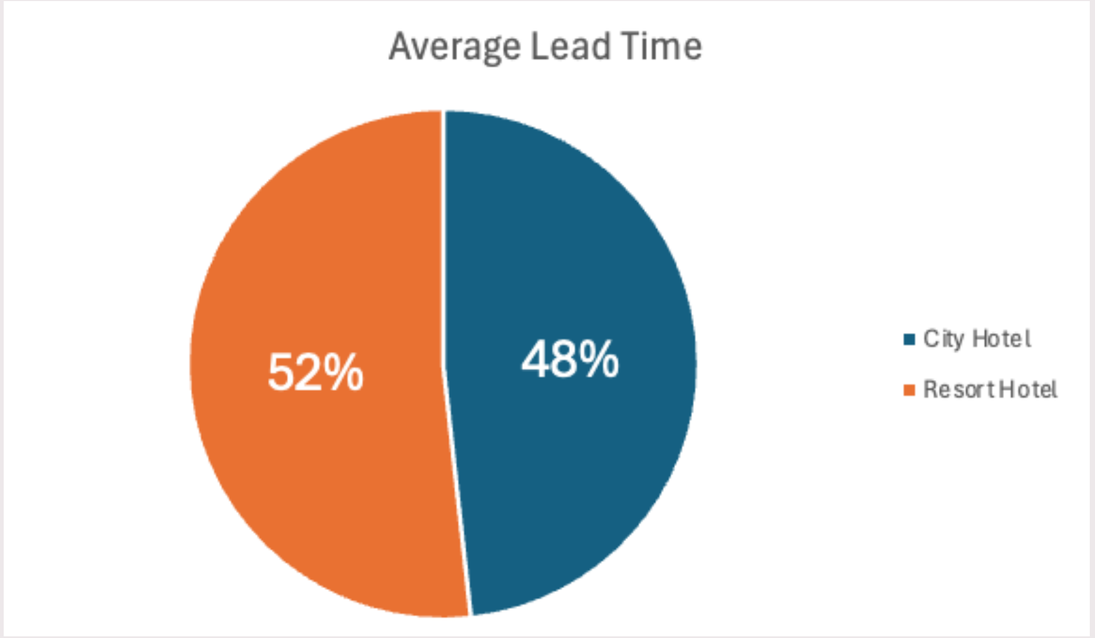
ANALYSE THE DATA

📊 **Insight:**

- City Hotels: 77.8 days
- Resort Hotels: 83.4 days
- Overall lead time: 80 days
- Resort Hotels have a slightly longer booking lead time, possibly due to planned vacations compared to last-minute city hotel stays.

📌 **Actionable Insight:**

- Consider early booking discounts for City Hotels to encourage **advanced reservations**.



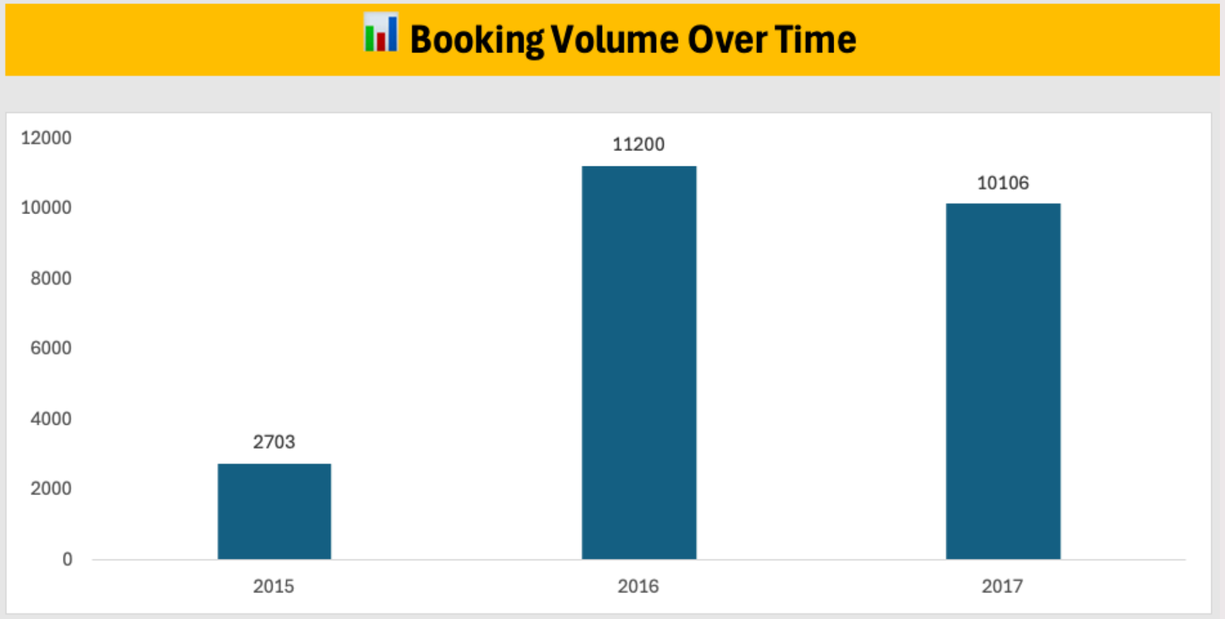
ANALYSE THE DATA

📊 **Insight:**

- The number of cancellations increased significantly from 2015 (2,703) ➜ 2016 (11,200) ➜ 2017 (10,106).
- The highest volume of cancellations occurred in 2016, followed by a slight drop in 2017.

📌 **Actionable Insight:**

- Investigate what happened in 2016 (economic downturn, policy change, marketing campaigns?).
- Introduce prepaid non-refundable booking options to reduce cancellations.



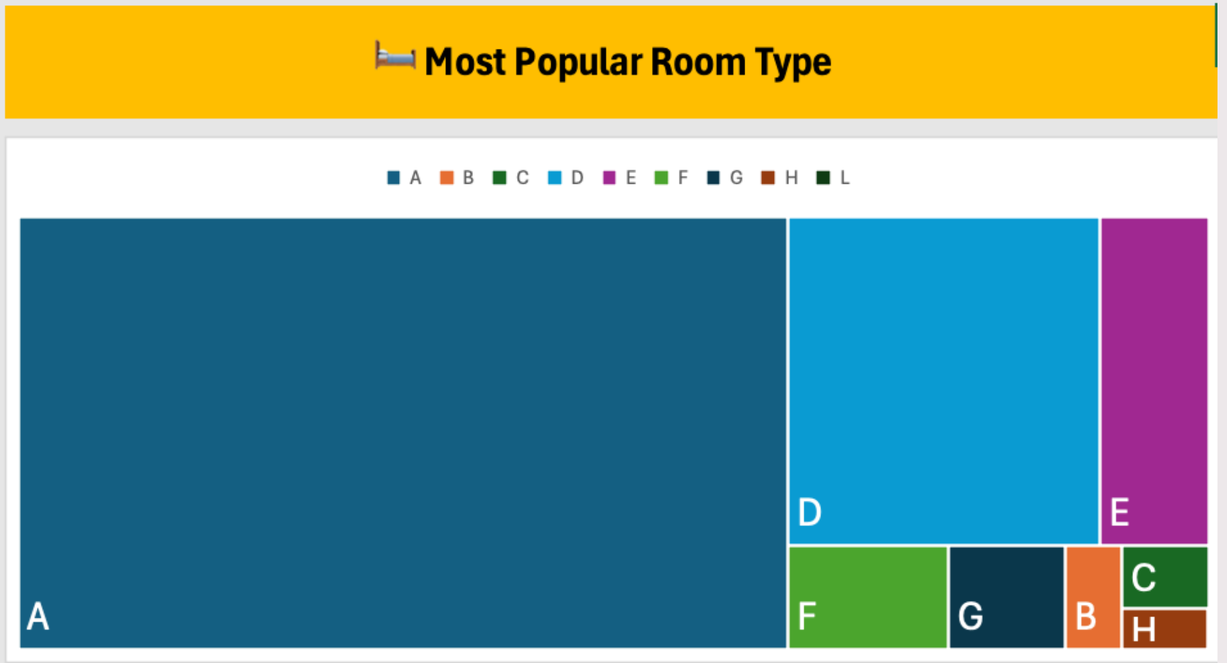
ANALYSE THE DATA

📊 **Insight:**

- Room Type A is overwhelmingly the most booked (56,436 times).
- Other room types have much lower booking counts.
- Room Type D (17,376 bookings) is the second most popular.

📌 **Actionable Insight:**

- Hotels should analyze why Room Type A is in high demand (pricing, availability, amenities).
- Consider bundling other room types with added perks to increase their demand.



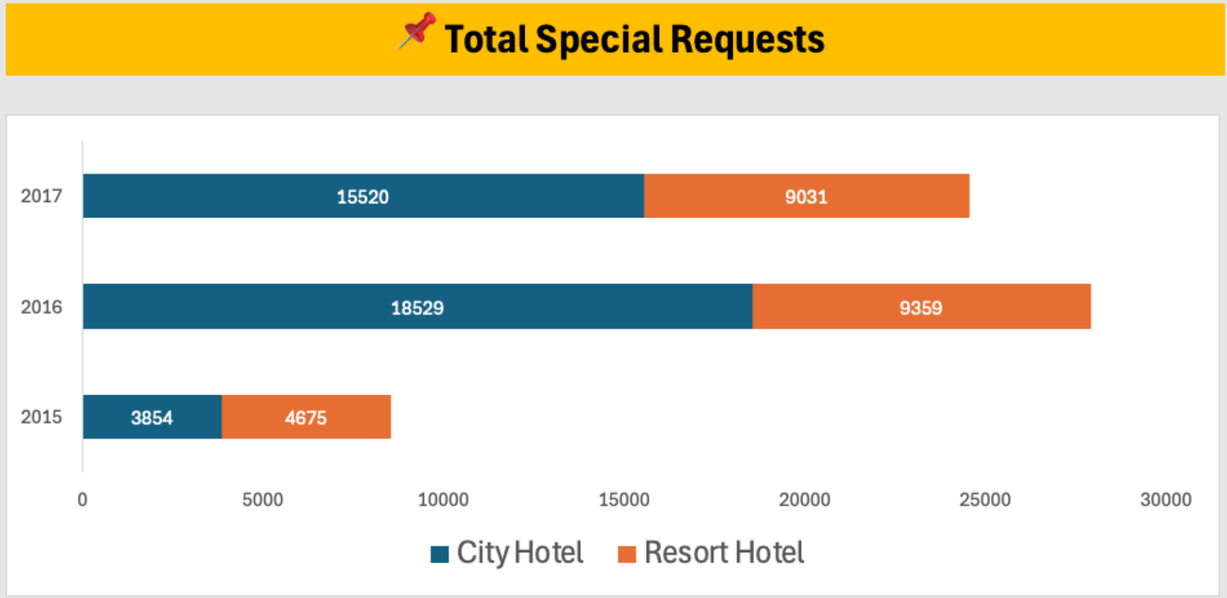
ANALYSE THE DATA

📊 **Insight:**

- Special requests peaked in 2016 (27,888 requests) but declined in 2017 (24,551 requests).
- City Hotels receive more special requests than Resort Hotels.

📌 **Actionable Insight:**

- Consider automating or improving request handling to enhance guest satisfaction.



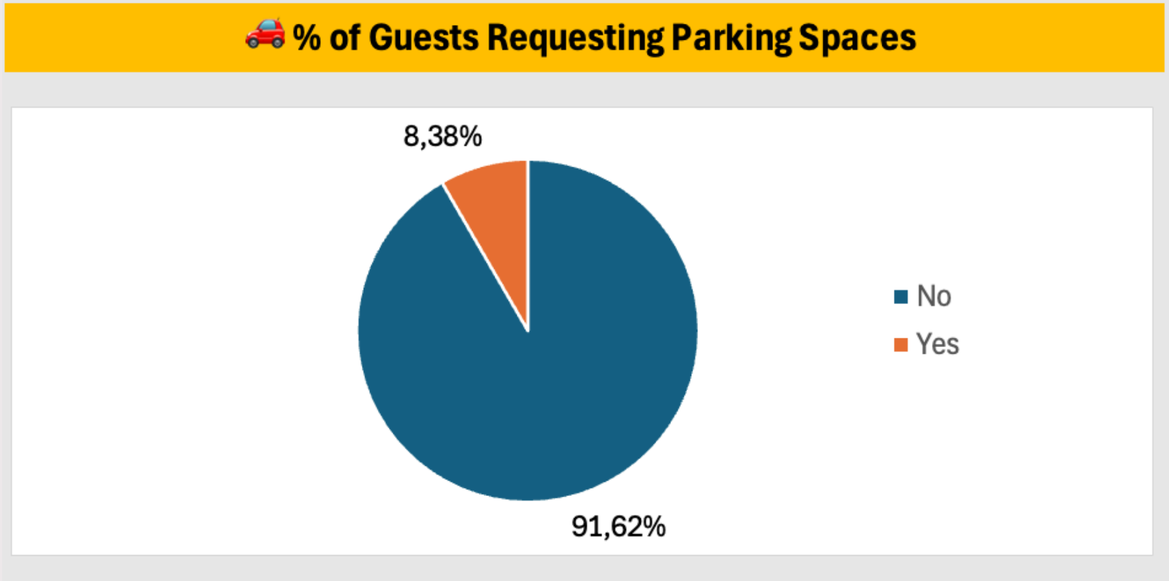
ANALYSE THE DATA

📊 **Insight:**

- 91.62% of guests do NOT request parking.
- Only 8.38% of guests request parking.

📌 **Actionable Insight:**

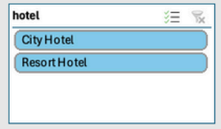
- Hotels may optimize parking space allocation or offer parking discounts to increase utilization.

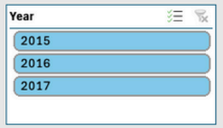


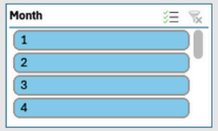
VISUALIZE THE DATA

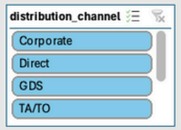
This interactive Hotel Booking Dashboard was created using Excel, leveraging Pivot Tables, Charts, and Slicers to provide a clear and insightful view of booking trends, cancellations, guest behavior, and revenue performance.

SUMMARY OVERVIEW



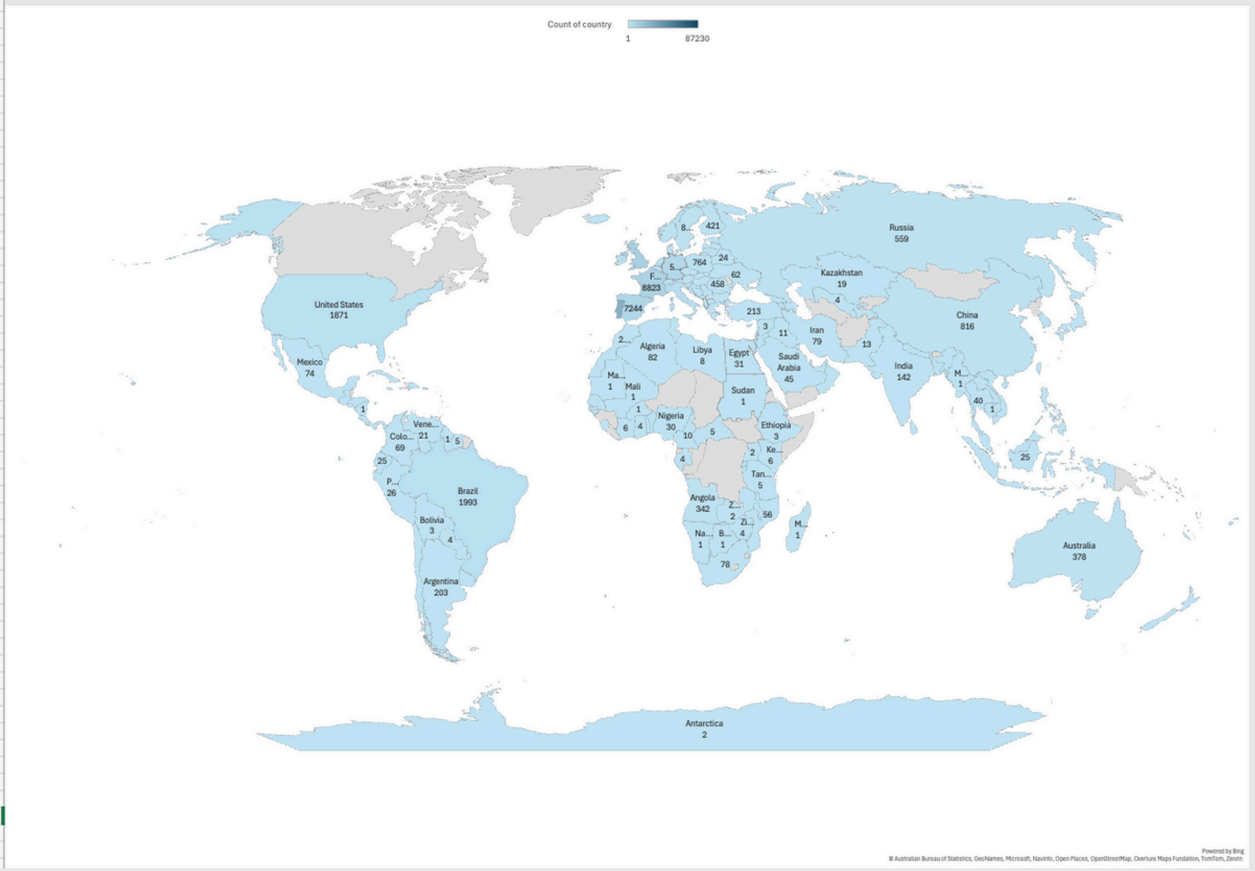




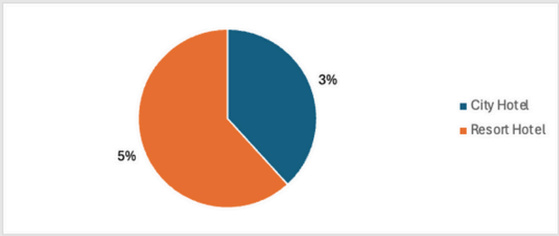


Total Bookings	Total Cancellations	Average Lead Time	Total Revenue
87230	24009	79,97	0

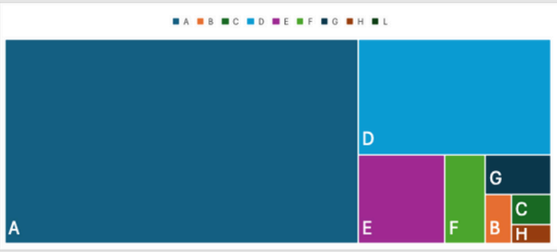
Top Guest Nationalities



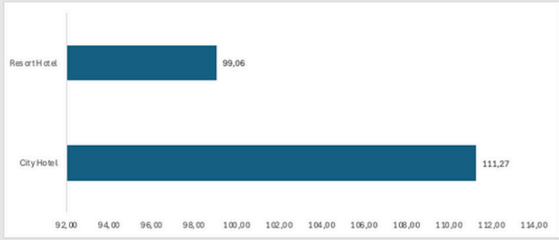
% Repeat Guests



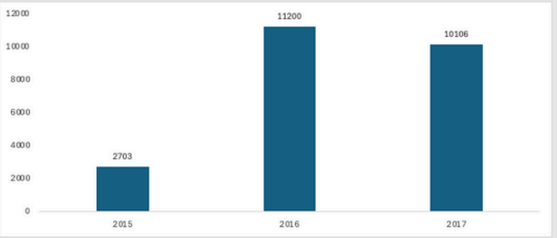
Most Popular Room Type



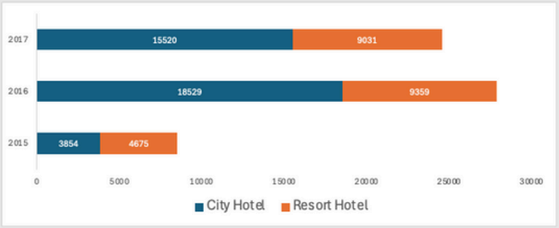
Average Daily Rate (ADR)



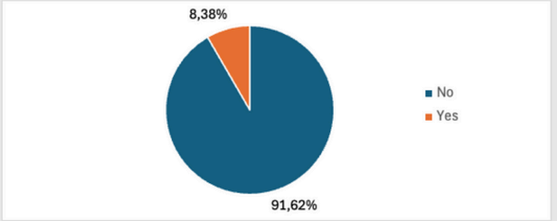
Booking Volume Over Time



Total Special Requests

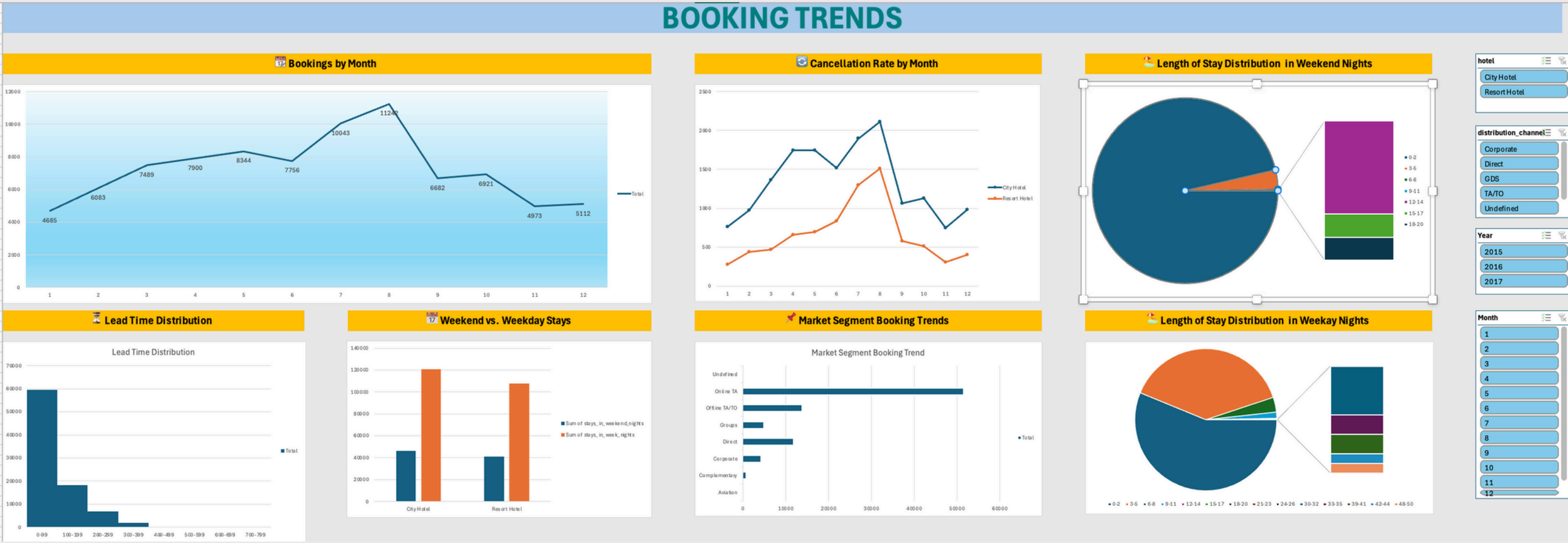


% of Guests Requesting Parking Spaces



VISUALIZE THE DATA

This Booking Trends Dashboard analyzes hotel reservations over time, revealing customer behavior, peak seasons, and booking preferences. Built in Excel, it visualizes key trends to support data-driven decisions for hotel management.



Contact Me

I am a final-year student with experience in data analysis, dashboard visualization, and e-commerce insights, using tools like Python, SQL, Excel, and Figma. I am seeking an opportunity in a Data Analyst or Business Analyst role to further develop my skills and contribute to business growth.

0339417393

Linkedin Profile

vinhtrinhtra123@gmail.com

Hoang Mai, Ha Noi