The DataViz Challenge - Transforming EDA Projects to Dashboards- Airbnb Listings Data

Name :- Dharmesh gadhiya

Email :- dharmeshgadhiya161@gmail.com

Github :-

Google Colab Notes :- **https://colab.research.google.com/drive/1L7EelYzVUbGtMkU-gAcL7KzKFOs83L-h#scrollTo=mo7KIbe3xkWp**

Create Stunning PowerBI Dashboard from the same dataset on which EDA was performed for the capstone project.

What is Airbnb?

Airbnb is an American company that operates an online marketplace.that allows people to rent out their properties to travelers. It's short for "Air Bed and Breakfast".it works to property owners, or "hosts", can list their properties on Airbnb.Travelers can search for listings based on location, price, and travel times.

Dataset Selection:

For this EDA project, we have chosen the "Airbnb Listings Data" dataset from 2 major cities: **Chicago and New Orleans.**

Dataset Details:

- **Dataset Name: Airbnb Listings Data**
- **Source: Link to dataset**
- **Cities: Chicago & New Orleans**
- Description: The Airbnb Listings Data contains information about different properties available for rent on Airbnb in a specific city. Each record represents a unique listing and includes attributes such as ~~property type~~, neighbourhood, ~~number of bedrooms~~, pricing, availability, host information, and more.

Key Attributes:-

1. id: Unique identifier for each listing.

2. name: The title or name of the listing.

3. host_id: Unique identifier for the host of the property.

4. host_name: Name of the host.

5. neighbourhood_group: The broader area or group that the neighbourhood belongs to.

6. neighbourhood: Specific neighbourhood where the property is located.

7. latitude: Latitude coordinate of the property.

8. longitude: Longitude coordinate of the property.

9. room_type: Type of room (e.g., Private room, Entire home/apt, Shared room).

10. price: Price of the listing per night.

11. minimum_nights: Minimum number of nights required for booking.

12. number_of_reviews: Total number of reviews received for the listing.

13. last_review: Date of the last review.(Date/month/year)

14. reviews_per_month: Average number of reviews per month.

15. calculated_host_listings_count: Total number of listings managed by the same host.

16. availability_365: Number of days the listing is available for booking in a year.

17. number_of_reviews_ltm :- The total number of reviews a listing received in the last 12 months (LTM - Last Twelve Months).

18. license: Legal license information for the listing (if required by local regulations).

How to proceed with the dashboard:

1.Data Cleaning

Begin by addressing the disorder and inconsistency within the dataset. Utilise Google colab Notebook and PowerBI Prep to systematically cleanse the data, rectifying discrepancies, eliminating duplicates, and standardising formats.

2. Data Transformation

Generate supplementary columns by utilising pre-existing categorical data. These columns will be derived from extensive descriptive text, which, in its original form, proved arduous to comprehend and unsuitable for visualisation purposes. The extra columns that we created gave a much clear sense of how to approach and make an effective visualisation.

3. PowerBI

Employ PowerBI Prep to leverage its distinctive "Group and Replace" feature. Under the column denoted as Neighbourhood there are instances where identical entities are variably represented due to disparities in letter casing, spelling variations, or phonetic similarity. The "Group and Replace" algorithm inherent to PowerBI Prep proved instrumental in mitigating this issue.

This dataset has around Chicago 8269 and New Orleans 7118 observations in it with 18 columns and it is a mix between categorical and numeric values.

```python
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```python
# Load Dataset
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```python
# Load Dataset
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Module: Data Visualization Tools/Capstone Project/Chicago_Dataset.csv')
```

```python
df_new_orleans = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Module: Data Visualization Tools/Capstone Project/NewOrleans_Dataset.csv')
```

```python
## Dataset First Look
df.head()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	l
0	2384	Hyde Park - Walk to The University of Chicago	2613	Rebecca	NaN	Hyde Park	41.78790	-87.58780	Private room	125.0	3	246	
1	7126	Tiny Studio Apartment 94 Walk Score	17928	Sarah	NaN	West Town	41.90166	-87.68021	Entire home/apt	94.0	2	544	
2	10945	The Biddle House (#1)	33004	At Home Inn	NaN	Lincoln Park	41.91196	-87.63981	Entire home/apt	174.0	4	104	
3	12140	Lincoln Park Guest House	46734	Shay And Robert	NaN	Lincoln Park	41.92357	-87.64947	Private room	339.0	2	16	
		Quirky Bucktown											

```
df_new_orleans.head(5)
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	nu
0	970324090541593163	Comfort in Bywater	56212513	Amanda	NaN	St. Claude	29.966211	-90.041057	Entire home/apt	83.0	30	
1	768293169006662066	Cozy and Comfortable	136665450	Joshua	NaN	St. Claude	29.972380	-90.042320	Private room	180.0	30	
2	1069600842896440330	Tons of Space,10 Beds,3 1/2 Bath	556686789	Joe	NaN	Algiers Point	29.953003	-90.050669	Entire home/apt	638.0	1	
3	23588926	Superdome Lux Home	55472287	Kamiah	NaN	Central City	29.942904	-90.090524	Entire home/apt	350.0	3	
4	12111787	Historic Courtyard Cabana New Orleans	54478905	Kevin	NaN	Algiers Point	29.950940	-90.049440	Entire home/apt	90.0	30	

```
# Dataset Rows & Columns count
df.shape
```

```
(8269, 18)
```

```
[8] df_new_orleans.shape
```

```
(7118, 18)
```

```python
# Dataset Info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8269 entries, 0 to 8268
Data columns (total 18 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              8269 non-null   int64  
 1   name                            8269 non-null   object 
 2   host_id                         8269 non-null   int64  
 3   host_name                       8269 non-null   object 
 4   neighbourhood_group             0 non-null      float64
 5   neighbourhood                   8269 non-null   object 
 6   latitude                        8269 non-null   float64
 7   longitude                       8269 non-null   float64
 8   room_type                       8269 non-null   object 
 9   price                           7844 non-null   float64
 10  minimum_nights                  8269 non-null   int64  
 11  number_of_reviews               8269 non-null   int64  
 12  last_review                     6679 non-null   object 
 13  reviews_per_month               6679 non-null   float64
 14  calculated_host_listings_count  8269 non-null   int64  
 15  availability_365                8269 non-null   int64  
 16  number_of_reviews_ltm           8269 non-null   int64  
 17  license                         6577 non-null   object 
dtypes: float64(5), int64(7), object(6)
memory usage: 1.1+ MB
```

```python
df_new_orleans.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7118 entries, 0 to 7117
Data columns (total 18 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              7118 non-null   int64  
 1   name                            7118 non-null   object 
 2   host_id                         7118 non-null   int64  
 3   host_name                       7118 non-null   object 
 4   neighbourhood_group             0 non-null      float64
 5   neighbourhood                   7118 non-null   object 
 6   latitude                        7118 non-null   float64
 7   longitude                       7118 non-null   float64
 8   room_type                       7118 non-null   object 
 9   price                           5969 non-null   float64
 10  minimum_nights                  7118 non-null   int64  
 11  number_of_reviews               7118 non-null   int64  
 12  last_review                     5977 non-null   object 
 13  reviews_per_month               5977 non-null   float64
 14  calculated_host_listings_count  7118 non-null   int64  
 15  availability_365                7118 non-null   int64  
 16  number_of_reviews_ltm           7118 non-null   int64  
 17  license                         5822 non-null   object 
dtypes: float64(5), int64(7), object(6)
memory usage: 1001.1+ KB
```

```
# Missing Values/Null Values Count
df.isnull().sum().loc[lambda x:x > 0].sort_values(ascending=False)# lambda using check less than 0 vaules
```

	0
neighbourhood_group	8269
license	1692
last_review	1590
reviews_per_month	1590
price	425

dtype: int64

- The columns neighbourhood_group have not a data so columns remove
- The columns last_review and reviews_per_month have 1590 null values each.

```
[14] df_new_orleans.isnull().sum().loc[lambda x:x > 0].sort_values(ascending=False)# lambda using check less than 0 vaules
```

	0
neighbourhood_group	7118
license	1296
price	1149
last_review	1141
reviews_per_month	1141

dtype: int64

- The columns neighbourhood_group have not a data so columns remove
- The columns last_review and reviews_per_month have 1141 null values each.

```
# main data set in copy data Farme
ch_df =df.copy()
```

```
[18] new_orleans_df=df_new_orleans.copy()
```

```
[19] ch_df.head()
```

```
[ ] new_orleans_df.head()
```

```
[20] # neighbourhood_group columns delete
ch_df.drop('neighbourhood_group',axis=1,inplace=True)
```

```
[21] new_orleans_df.drop('neighbourhood_group',axis=1,inplace=True)
```

```
[22] ch_df.fillna({'license':0,'reviews_per_month': 0}, inplace=True)
```

```
[23] new_orleans_df.fillna({'license':0,'reviews_per_month': 0}, inplace=True)
```

- The neighbourhood_group column drop, because there is not a data found.
- so there is no null value now in **'license', 'reviews_per_month'** column because we replaced null value by 0 value. this will make sense because there is no any such data to find those null value

```
# converting to datetime
[24] ch_df['last_review'] = pd.to_datetime(ch_df['last_review'])
```

```
[25] # converting to datetime
     new_orleans_df['last_review'] = pd.to_datetime(new_orleans_df['last_review'])
```

```
[26] # price in missing vaules in fillna using mean() vaules set.
     ch_df['price'].fillna(ch_df['price'].mean(),inplace=True)
```

```
[27] new_orleans_df['price'].fillna(new_orleans_df['price'].mean(),inplace=True)
```

```
[28] #new column add
     ch_df['city']= 'Chicago'
```

```
new_orleans_df['city']= 'New Orleans'
```

Data set in one new column 'city' add.

```
[78] ch_df.describe().T
```

	count	mean	min	25%	50%	75%	max	
id	8269.0	570572079287361216.0	2384.0	43514955.0	710820308716380928.0	1008815233647371008.0	1247067926762205696.0	48
host_id	8269.0	213834830.639618	2153.0	43215726.0	142223344.0	395806758.0	651286463.0	
latitude	8269.0	41.892207	41.646767	41.863499	41.895919	41.93558	42.0222	
longitude	8269.0	-87.662237	-87.846898	-87.685882	-87.657947	-87.63095	-87.52842	
price	8269.0	215.663437	18.0	101.0	160.0	230.0	10000.0	
minimum_nights	8269.0	14.266175	1.0	2.0	2.0	32.0	365.0	
number_of_reviews	8269.0	52.701778	0.0	1.0	17.0	67.0	1142.0	
last_review	6679	2024-05-21 05:42:35.412486912	2015-07-17 00:00:00	2024-07-21 00:00:00	2024-08-28 00:00:00	2024-09-07 00:00:00	2024-09-16 00:00:00	
reviews_per_month	8269.0	1.58658	0.0	0.13	1.11	2.56	58.33	
calculated_host_listings_count	8269.0	43.268956	1.0	1.0	4.0	17.0	516.0	
availability_365	8269.0	220.850647	0.0	130.0	244.0	334.0	365.0	
number_of_reviews_ltm	8269.0	14.322893	0.0	0.0	6.0	23.0	387.0	

Chicago city average price **215.66 , minimum_nights 14.27 and availability_365 220.85**

```
new_orleans_df.describe().T
```

	count	mean	min	25%	50%	75%	max	
id	7118.0	445200076537636736.0	19091.0	29337379.0	54037240.0	913306446418334208.0	1238854101672315392.0	47555499483674
host_id	7118.0	212753125.377213	971.0	42536338.0	135042143.5	396738040.0	596955775.0	193131641.
latitude	7118.0	29.958115	29.89768	29.942531	29.955467	29.969475	30.16104	0.
longitude	7118.0	-90.073857	-90.13646	-90.08933	-90.07349	-90.064159	-89.73709	0.
price	7118.0	199.731781	10.0	96.0	156.0	199.731781	28001.0	417.
minimum_nights	7118.0	13.282242	1.0	1.0	2.0	30.0	1125.0	24.
number_of_reviews	7118.0	65.660298	0.0	3.0	26.0	89.0	1935.0	100.
last_review	5977	2023-11-19 04:09:35.807260928	2014-04-08 00:00:00	2024-03-02 00:00:00	2024-07-08 00:00:00	2024-08-18 00:00:00	2024-09-05 00:00:00	
reviews_per_month	7118.0	1.436967	0.0	0.18	1.0	2.2	44.8	1.
calculated_host_listings_count	7118.0	13.299803	1.0	1.0	3.0	14.0	85.0	20.
availability_365	7118.0	189.340545	0.0	84.0	186.0	308.0	365.0	123.
number_of_reviews_ltm	7118.0	13.138803	0.0	0.0	5.0	21.0	395.0	19

New orleans city average price **199.73 , minimum_nights 13.28 and availability_365 189.34**

```
[34] # Concatenate along rows two dataset
     chicago_newOrleans_df = pd.concat([ch_df, new_orleans_df],sort=False)
```

```
[35] chicago_newOrleans_df.head(5)
```

```
chicago_newOrleans_df.city.value_counts()
```

	count
city	
Chicago	8269
New Orleans	7118

dtype: int64

```
[ ] chicago_newOrleans_df.duplicated().sum()
```

```
0
```

```
[37] # save to use to powerbi data
     chicago_newOrleans_df.to_csv("Chicago_NewOrleansDataset.csv",index=False)
```

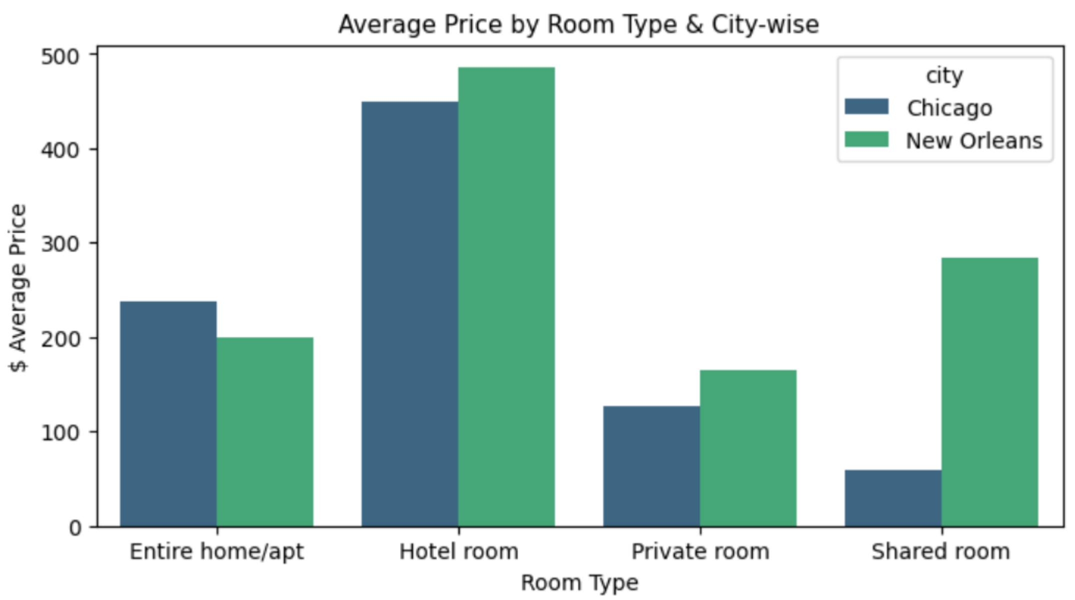
```
chicago_newOrleans_df.columns
```

```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood', 'latitude',
       'longitude', 'room_type', 'price', 'minimum_nights',
       'number_of_reviews', 'last_review', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365',
       'number_of_reviews_ltm', 'license', 'city'],
      dtype='object')
```

```
[39] city_chicago= chicago_newOrleans_df[chicago_newOrleans_df['city']=='Chicago']
     city_newOrleans= chicago_newOrleans_df[chicago_newOrleans_df['city']=='New Orleans']
```

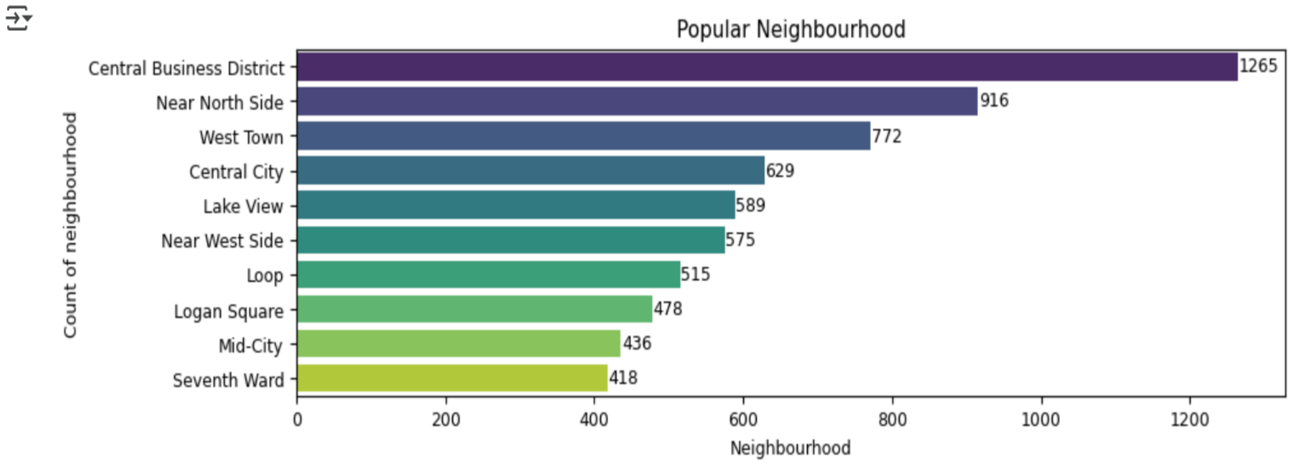
Visual Chart

1) Average price by room type city wise.



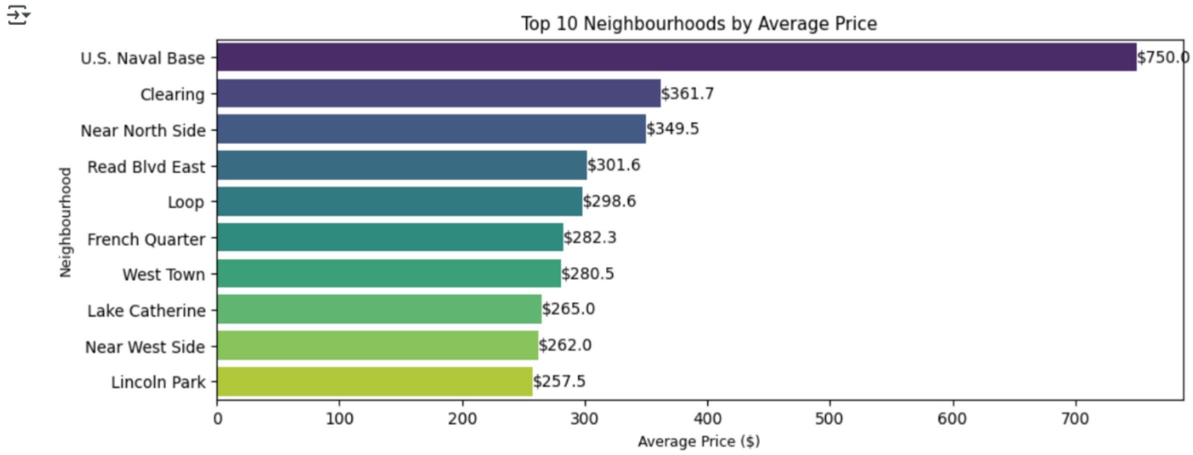
- The chart provides insights into city-wise average room prices for different room types.
- Hotel room typically has the highest average price in all cities. This makes sense since guests get the whole property instead of just a room.
- Entire home/apt and Private rooms are cheaper than entire homes but more expensive than shared rooms.
- Shared rooms have the lowest prices by chicago city, making them ideal for budget travelers.

2) Popular neighbourhood count of listing.



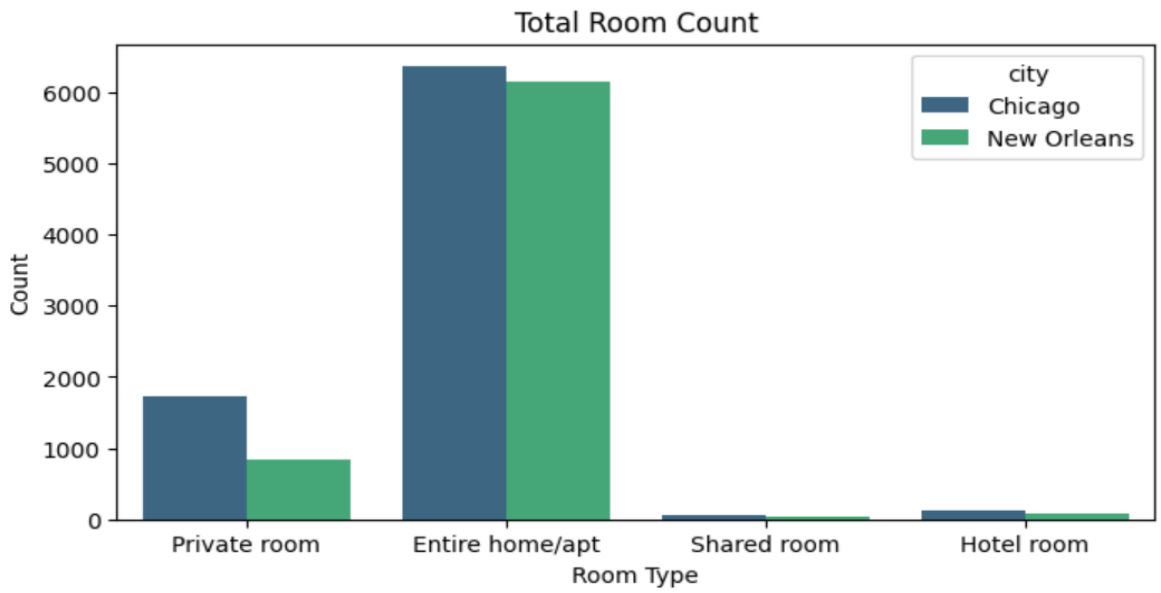
- The top neighborhood has the highest number of listings, indicating it is a prime Airbnb location with strong demand.

3) Top Neighbourhoods by average price



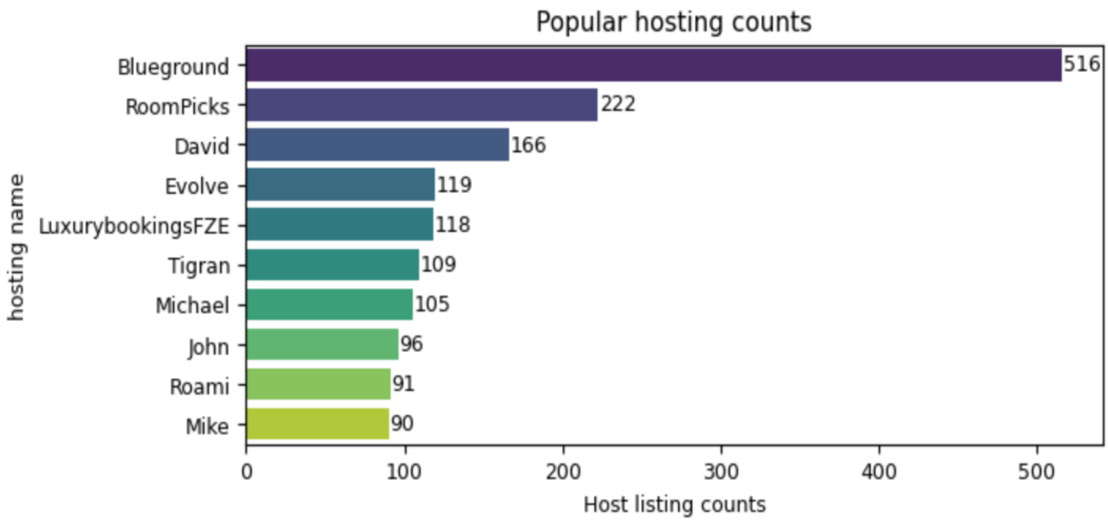
- The top-ranked neighborhoods have significantly higher average prices compared to others.

4) Total Room count.



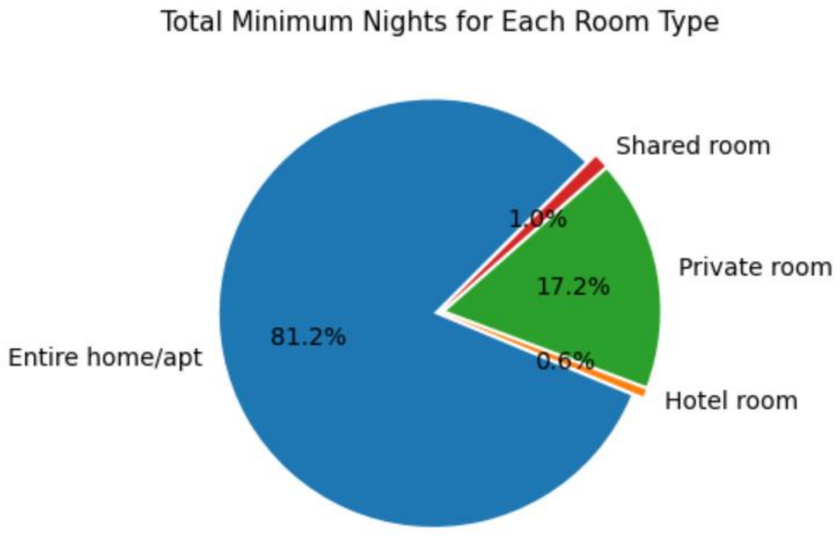
- Entire home/apt typically has the highest room type in all cities. suggesting a preference for full-space rentals over shared spaces.

5) Popular hosting listing count.



- The top hosting has the highest number of listings, indicating it is a prime Airbnb location with strong demand.

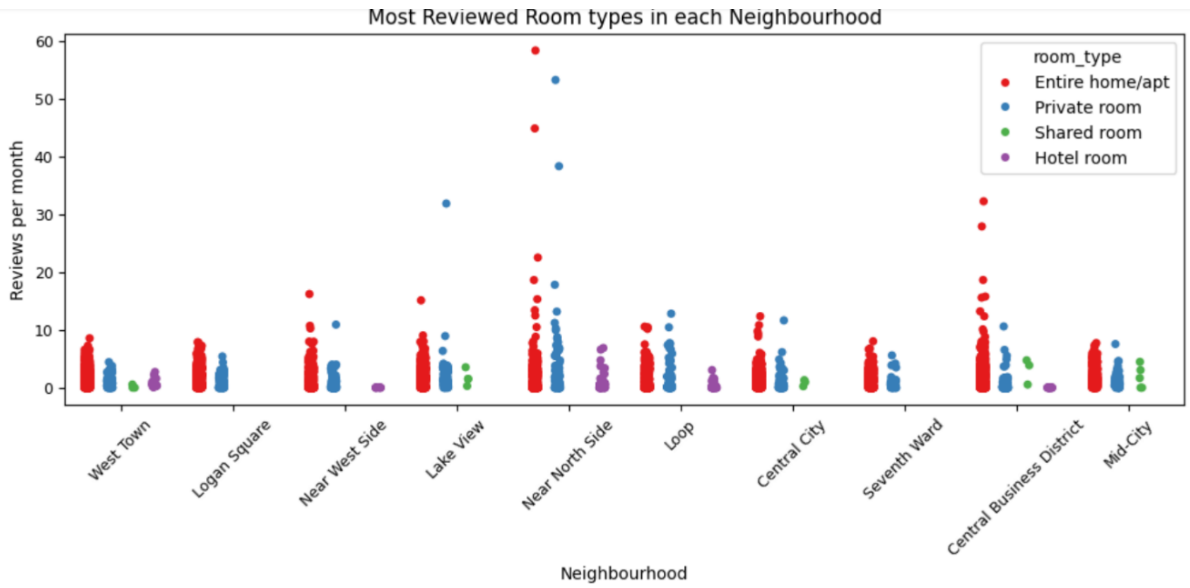
6) Total Minimum Nights for each room type saty.




```
room_type
Entire home/apt    172583
Hotel room           1179
Private room        36632
Shared room          2116
Name: minimum_nights, dtype: int64
```

- The majority of total minimum nights are from Entire home/apt 81.2% , meaning guests tend to book entire properties more frequently and for longer stays.

7) Most Reviewed room types in each neighbourhood.

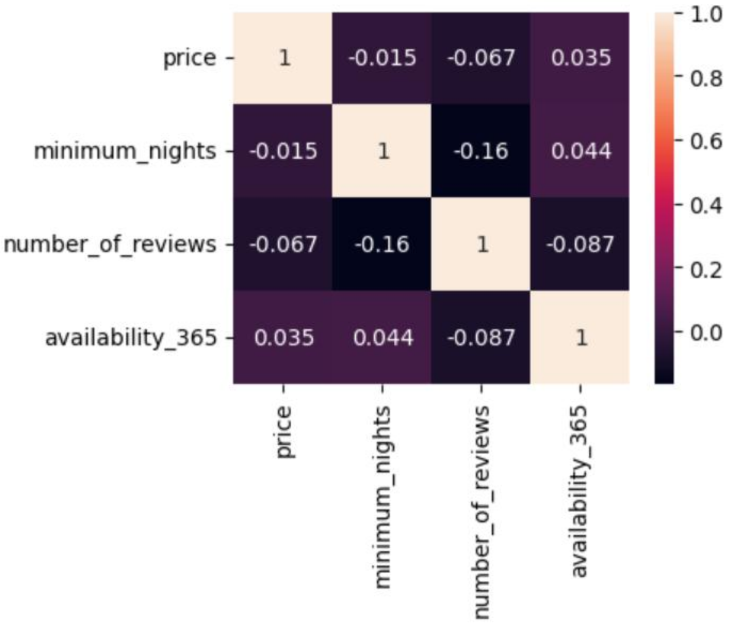


- Entire home/apt listings tend to have the highest reviews per month in most neighborhoods.
- Guests likely prefer the entire homes/apartments are a popular choice among travelers.

8) Relationship between price and number of reviews.

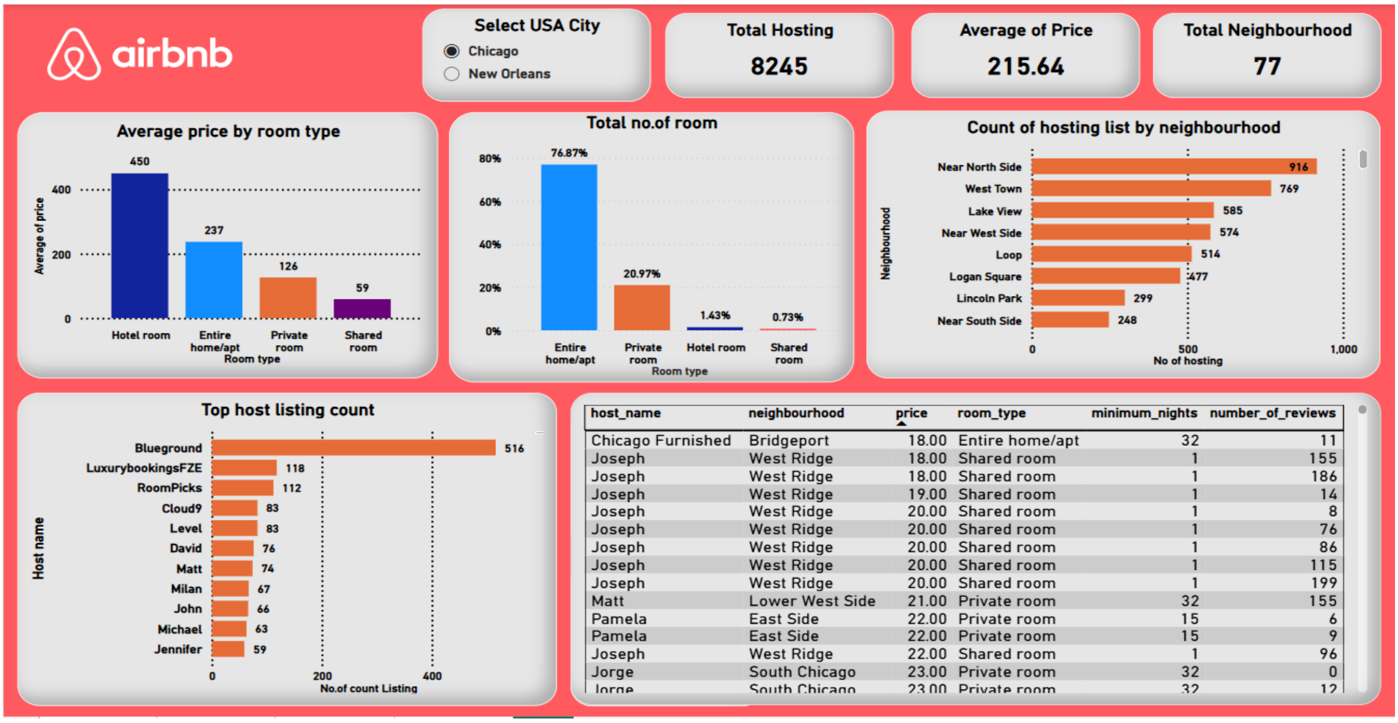
- Correlation Between Price and Reviews indicating that price and number of reviews do not have a strong linear relationship.
- Higher-priced listings don't necessarily receive more reviews.
- Lower-priced listings are not guaranteed to have fewer reviews.
- Engage with past guests to encourage more reviews, as they help build credibility.
- Hosts should focus on competitive pricing to attract more bookings and reviews.

9**) Relationship price between review other attributes.**



Power-BI Dashboard

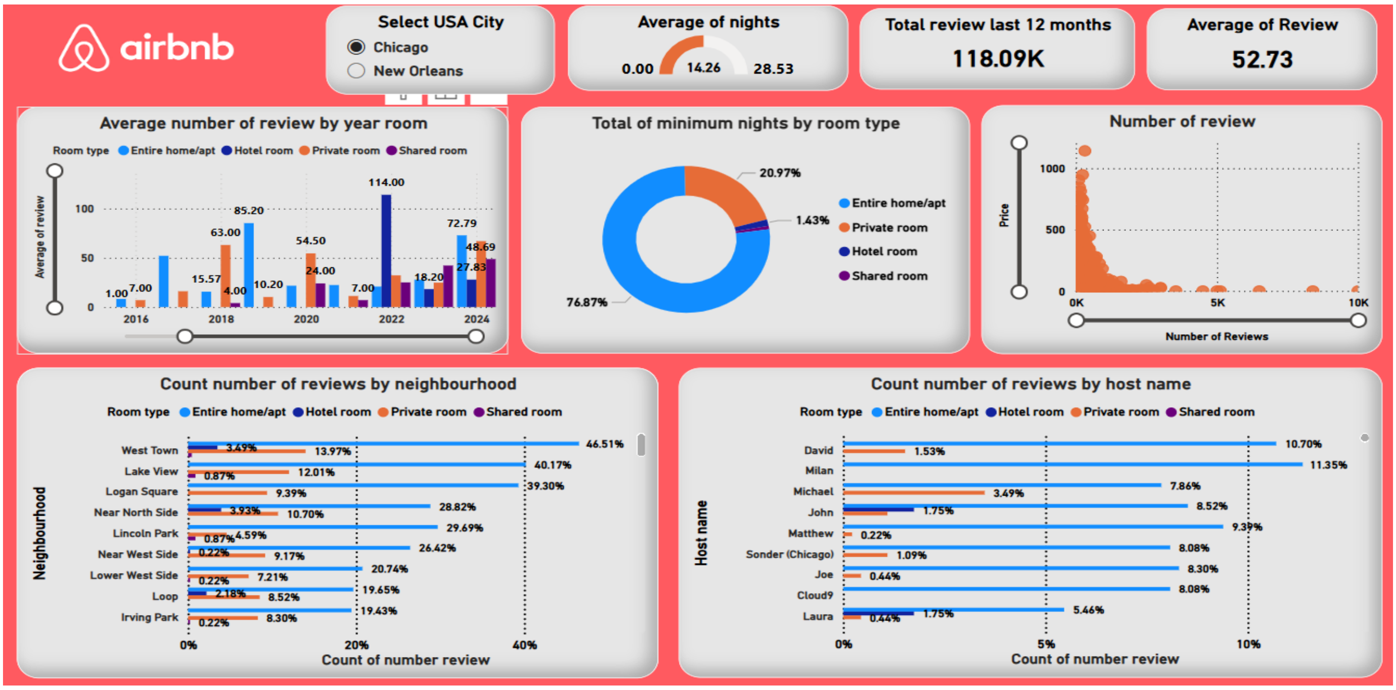
Dashboard- Room



Key Insights & Recommendations :-

- Airbnb listing dataset in city **Chicago/New Orleans** Hotel rooms have the highest average price.
- That Hotel rooms target luxury travelers, while shared/private rooms are budget-friendly options.
- Optimize pricing strategies should be used to adjust rates based on seasonality, demand, and competitor pricing.
- Borth city Entire home/apartments dominate the market (76.87%), indicating high demand for full-property rentals.
- Hotel rooms and shared rooms are rare, confirming that Airbnb in these cities primarily serves vacationers over budget travelers.
- The **Chicago** city top neighborhoods for hosting listing Near North Side , West Town , and Lake View and **New Orleans** city Central Business district, Central city, Mid city , Seventh ward are the most popular areas .
- The Neighborhood trends a making them prime locations for investment, which have high guest demand likely due to high tourist activity, nightlife, and accessibility.
- The Chicago city top hosting blueground listings, with significantly more listings than competitors. like Other major hosts LuxurybookingsFZE , RoomPicks , and Cloud9 hosting listings.
- Expand listings in high-demand

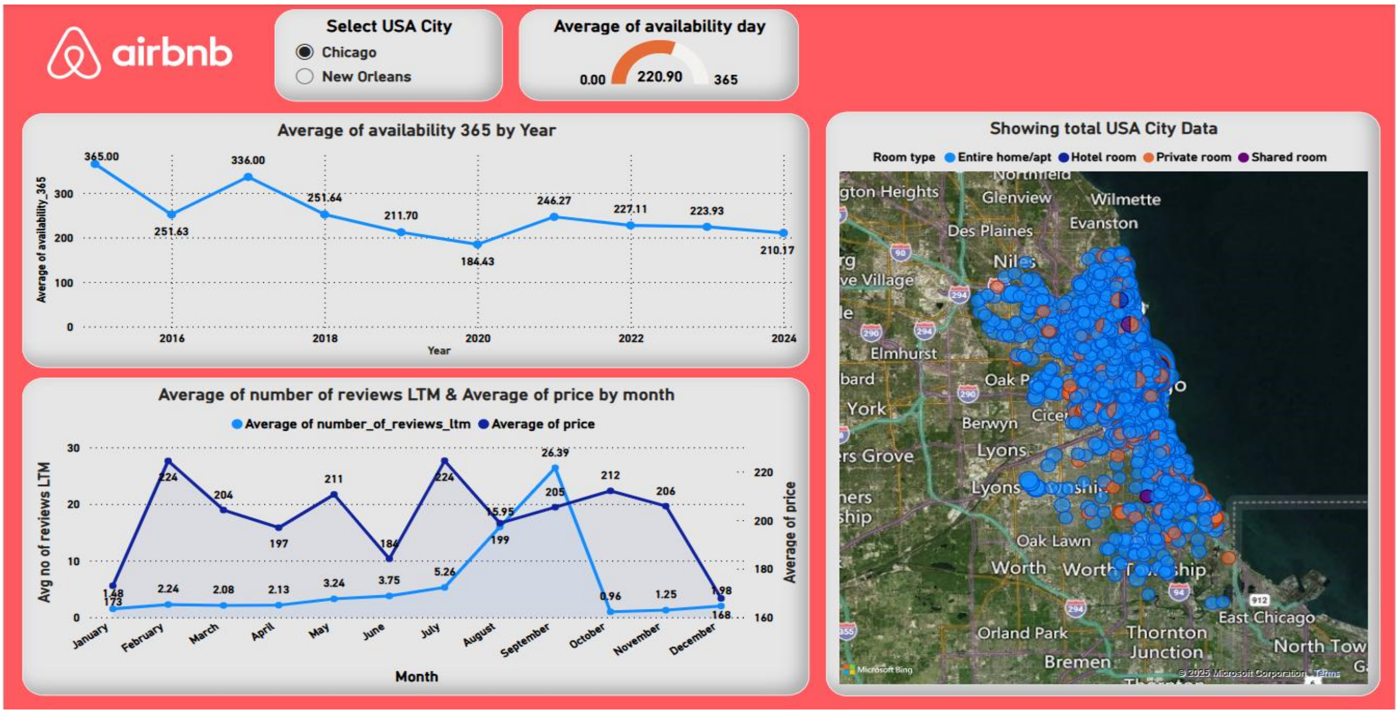
Dashboard- Average review/nights Room type



Key Insights & Recommendations:

- The insights a **Chicago** city data visual.
- Average reviews per listing This indicates moderate engagement, suggesting room for improvement in guest feedback collection.
- Compete against dominant hosts by offering unique guest experiences & improved facility.
- Chicago city 76.87% of listings have a minimum stay of 14+ nights, showing strong long-term rental demand.
- Chicago city neighborhood Review Distribution West Town (46.51%) and Lake View (40.17%) lead in review counts, showing high guest activity.
- Building relationships with local businesses can enhance guest satisfaction and attract more reviews.
- Differentiate with unique facility, flexible policies, and personalized guest experiences.

Dashboard- Average last 12 month review & Total USA City data by MAP.



Key Insights & Recommendations:

- The Chicago City availability trends over time average availability days per year is declining from 365 (2015) to 210.17 (2024).
- This last 12-month data review in compere review count peaks in February (224) and July (224), indicating high booking activity and Pricing peaks in September ($205) and October ($212), suggesting high-demand periods.
- Use dynamic pricing tools to adjust rates based on seasonality, demand spikes, and competitor pricing.
- Offer discounts for longer stays to improve occupancy rates.

Conclusion

This Airbnb booking analysis provides valuable insights into the City Airbnb market. Identify the most popular neighborhoods and room types. This information can be used to focus marketing efforts and allocate inventory accordingly. Analyze price trends by neighborhood, room type, and time of year. This information can be used to set competitive prices and maximize revenue. If investing in new properties, target high-demand areas (like Downtown, North Side, Logan Square). Offer discounts in low-demand months to maintain occupancy.