EDA PROJECT (INT-353)

Submitted by – ANSHUL SONI

Section – K20RU

Roll No. – RK20RUA19

Registration No. – 12016175

TOPIC –

Chicago Airbnb Open Dataset

GitHub Link-

https://github.com/Sonianshul2011/chicago_airbnb_data_analysis.git

Dataset link-

https://www.kaggle.com/datasets/jinbonnie/chicago-airbnb-open-data

Context

People love traveling(not during this pandemic time thought, please stay safe), and Airbnb can always offer different travel experiences for the travellers. I personally went to Chicago last year and used Airbnb found an amazing apartment to stay. So here are some dataset from Airbnb open data about the reservation in Chicago from 2017-2019.

Content

This dataset including the information about the hosts, the information about the position of the Airbnb(neighbourhood, latitude, longitude) which can use for map plot, describe of the room, and price, etc. Which will be a good dataset for data visualization and prediction.

Acknowledgements

This public dataset is part of Airbnb, and the original source can be found from Airbnb Open Data.

Our Objectives

- How the neighbourhood can influence the price of Airbnb?
- What can we learn from predictions? (ex: locations, prices, reviews, etc)
- If the host will influence the popularity of the Airbnb?

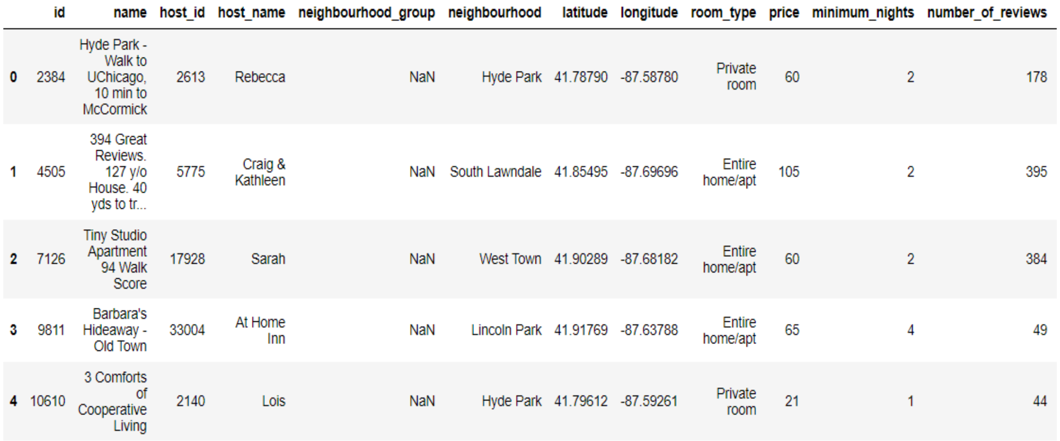
Approach

- Import all the required libraries.
- Import the dataset.
- See the shape of the dataset.
- Check out the datatypes of each columns.
- Check the null values & treat it with the required actions necessary.
- Draw the statistical insights as required.

- Finding the correlations between the attributes.
- Doing univariate, Bivariate and multivariate analysis on our dataset.

GLIMPSE OF THE DATASET

1.Dataset preview-



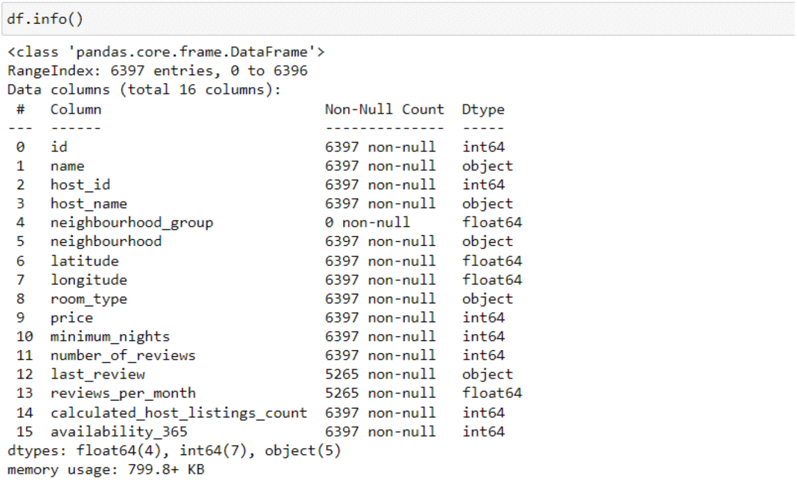
	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews
0	2384	Hyde Park - Walk to UChicago, 10 min to McCormick	2613	Rebecca	NaN	Hyde Park	41.78790	-87.58780	Private room	60	2	178
1	4505	394 Great Reviews. 127 y/o House. 40 yds to tr...	5775	Craig & Kathleen	NaN	South Lawndale	41.85495	-87.69696	Entire home/apt	105	2	395
2	7126	Tiny Studio Apartment 94 Walk Score	17928	Sarah	NaN	West Town	41.90289	-87.68182	Entire home/apt	60	2	384
3	9811	Barbara's Hideaway - Old Town	33004	At Home Inn	NaN	Lincoln Park	41.91769	-87.63788	Entire home/apt	65	4	49
4	10610	3 Comforts of Cooperative Living	2140	Lois	NaN	Hyde Park	41.79612	-87.59261	Private room	21	1	44

We can see that there are different types of fields, containing both numerical and categorical data.

Column Information-

- host_id - The id of the host
- host_name - The name of the host
- neighborhood - Which area is the airbnb belongs to
- latitude - The latitude of the position
- longitude - The longitude of the position
- room_type - Entire home/apt, Private room or Other
- price - The price of the apartment(per day)
- minimum_nights - The least nights you need to book
- number*of*reviews - The total number of the reviews on this dataset
- last_review - The last review time
- reviews*per*month - How many reviews the airbnb can receive per month
- calculated*host*listings_count - The total listing number of the host
- availability_365 - The available days

2.Data Types-



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6397 entries, 0 to 6396
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              6397 non-null   int64  
 1   name                            6397 non-null   object 
 2   host_id                         6397 non-null   int64  
 3   host_name                       6397 non-null   object 
 4   neighbourhood_group             0 non-null      float64
 5   neighbourhood                   6397 non-null   object 
 6   latitude                        6397 non-null   float64
 7   longitude                       6397 non-null   float64
 8   room_type                       6397 non-null   object 
 9   price                           6397 non-null   int64  
 10  minimum_nights                  6397 non-null   int64  
 11  number_of_reviews               6397 non-null   int64  
 12  last_review                     5265 non-null   object 
 13  reviews_per_month               5265 non-null   float64
 14  calculated_host_listings_count  6397 non-null   int64  
 15  availability_365                6397 non-null   int64  
dtypes: float64(4), int64(7), object(5)
memory usage: 799.8+ KB
```

We can observe that there are three types of data types present in the dataset, integer, float, and object.

Business Issues in hospitality Industry

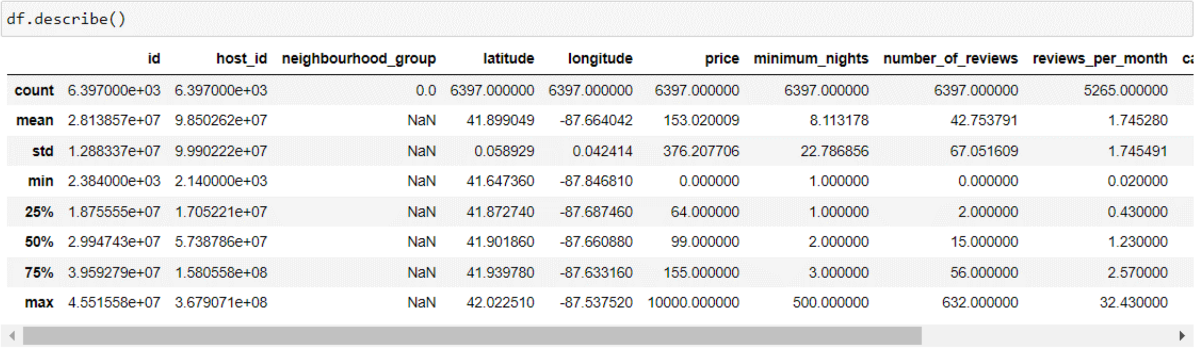
Online platforms are becoming increasingly popular every year, with customers giving companies online reviews, through comments, ratings and photos. The hospitality industry has been battling to establish strong relationships with their consumers to increase their reputation.

Reviews and comments can destroy or glamorise companies, thus the industry needs to utilise certain platforms to their advantage and manage their reputation, a challenge that organisations will face in 2021.

Customers today have grown to expect to be recognised and treated as individuals, rather than a steam-lined operations system. While consumers expect a greater level of personalization, businesses still struggle to translate data and insights into actions.

This information provides companies with customers past buying habits and their interests, enabling the hospitality industry to tailor their offers and promotions to specific customers. The industry needs to continually find new and unique ways to personalise a customer’s experience to keep a competitive edge.

DESCRIBING THE DATASET

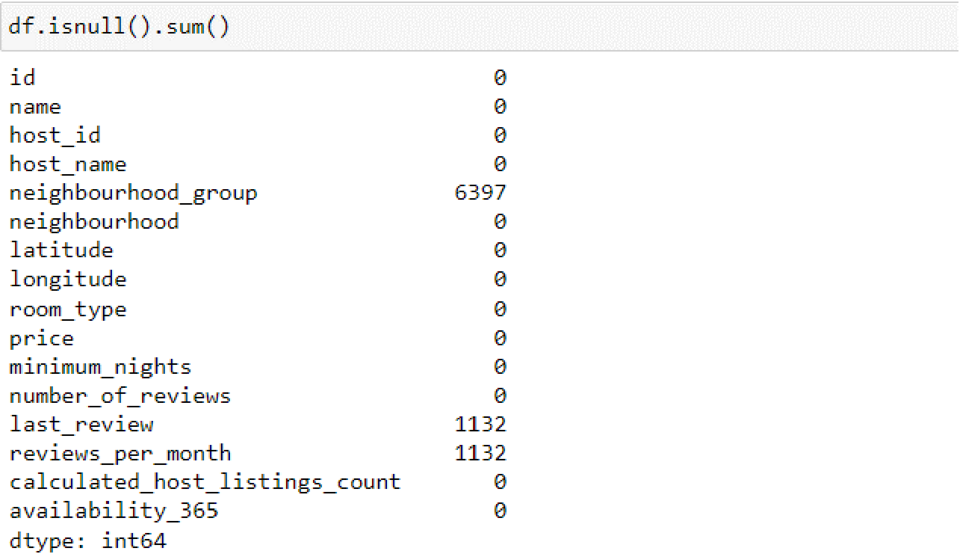


```
df.describe()
```

	id	host_id	neighbourhood_group	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	ca
count	6.397000e+03	6.397000e+03	0.0	6397.000000	6397.000000	6397.000000	6397.000000	6397.000000	5265.000000	
mean	2.813857e+07	9.850262e+07	NaN	41.899049	-87.664042	153.020009	8.113178	42.753791	1.745280	
std	1.288337e+07	9.990222e+07	NaN	0.058929	0.042414	376.207706	22.786856	67.051609	1.745491	
min	2.384000e+03	2.140000e+03	NaN	41.647360	-87.846810	0.000000	1.000000	0.000000	0.020000	
25%	1.875555e+07	1.705221e+07	NaN	41.872740	-87.687460	64.000000	1.000000	2.000000	0.430000	
50%	2.994743e+07	5.738786e+07	NaN	41.901860	-87.660880	99.000000	2.000000	15.000000	1.230000	
75%	3.959279e+07	1.580558e+08	NaN	41.939780	-87.633160	155.000000	3.000000	56.000000	2.570000	
max	4.551558e+07	3.679071e+08	NaN	42.022510	-87.537520	10000.000000	500.000000	632.000000	32.430000	

The dataset contains different types of data and at the first sight we can observe different types of statistical data like mean, median and mode. We can also observe the quantile values for 25%, 50% and 75% for all the different columns.

CHECKING FOR NULL VALUES

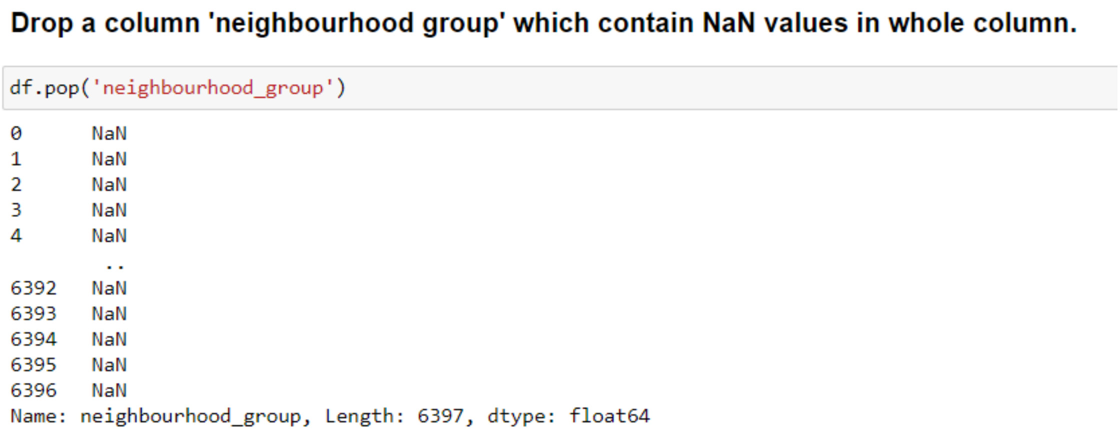


```
df.isnull().sum()
```

```
id                                   0
name                                 0
host_id                              0
host_name                            0
neighbourhood_group               6397
neighbourhood                        0
latitude                             0
longitude                            0
room_type                            0
price                                0
minimum_nights                       0
number_of_reviews                    0
last_review                       1132
reviews_per_month                 1132
calculated_host_listings_count       0
availability_365                     0
dtype: int64
```

We can observe that only “neighbourhood_group” ,“last_review” and“reviews_per_month” column have null values and all other columns does not have any null values.

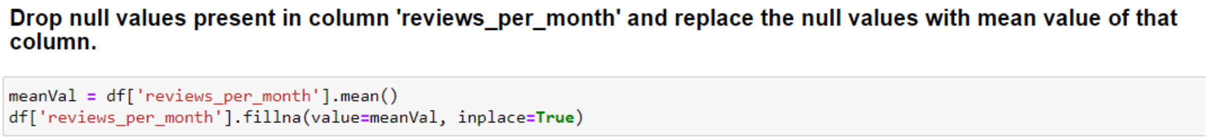
TREATING NULL VALUES



Drop a column 'neighbourhood group' which contain NaN values in whole column.

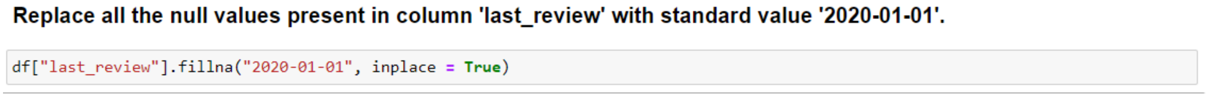
```
df.pop('neighbourhood_group')
```

```
0       NaN
1       NaN
2       NaN
3       NaN
4       NaN
         ..
6392    NaN
6393    NaN
6394    NaN
6395    NaN
6396    NaN
Name: neighbourhood_group, Length: 6397, dtype: float64
```



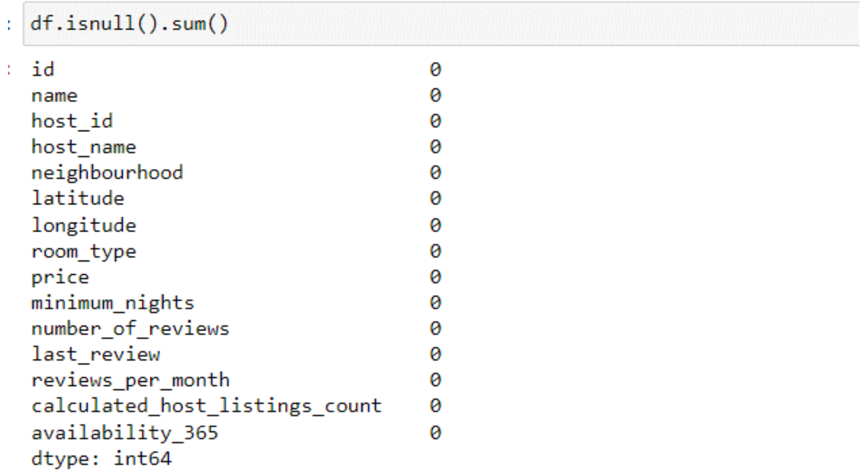
Drop null values present in column 'reviews_per_month' and replace the null values with mean value of that column.

```
meanVal = df['reviews_per_month'].mean()
df['reviews_per_month'].fillna(value=meanVal, inplace=True)
```



Replace all the null values present in column 'last_review' with standard value '2020-01-01'.

```
df["last_review"].fillna("2020-01-01", inplace = True)
```



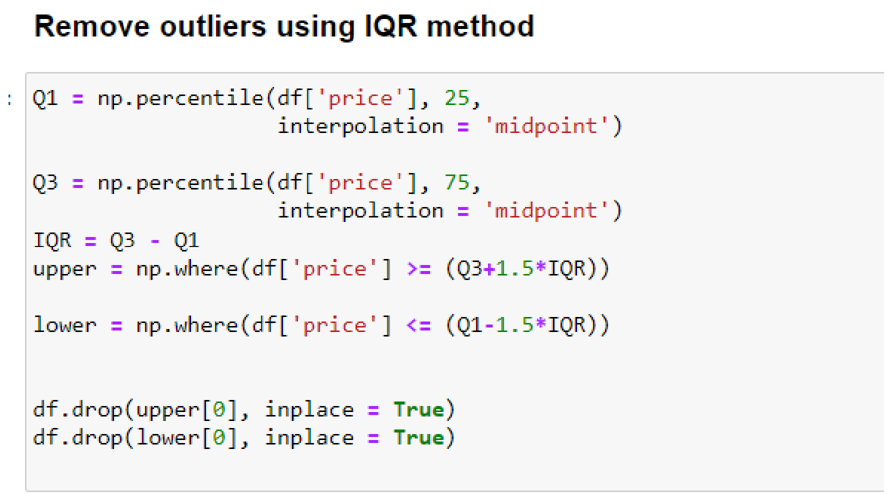
```
df.isnull().sum()
```

```
id                                0
name                              0
host_id                           0
host_name                         0
neighbourhood                     0
latitude                          0
longitude                         0
room_type                         0
price                             0
minimum_nights                    0
number_of_reviews                 0
last_review                       0
reviews_per_month                 0
calculated_host_listings_count    0
availability_365                  0
dtype: int64
```

Now all the null values are removed.

CLEANING THE DATASET AND REMOVING OUTLIERS

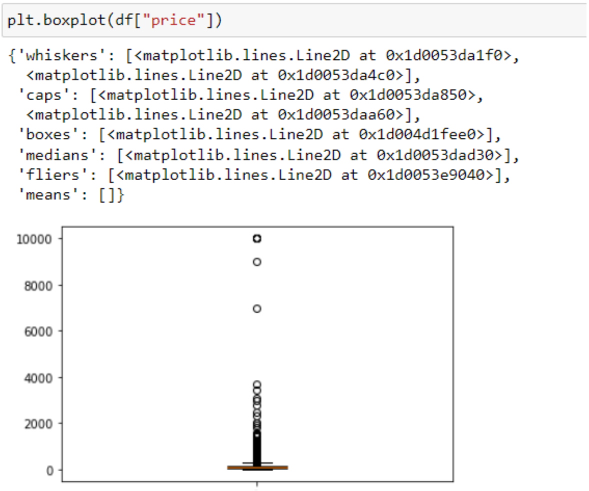
There are some outliers in “price” column.



Remove outliers using IQR method

```
Q1 = np.percentile(df['price'], 25,
                   interpolation = 'midpoint')

Q3 = np.percentile(df['price'], 75,
                   interpolation = 'midpoint')
IQR = Q3 - Q1
upper = np.where(df['price'] >= (Q3+1.5*IQR))

lower = np.where(df['price'] <= (Q1-1.5*IQR))


df.drop(upper[0], inplace = True)
df.drop(lower[0], inplace = True)
```

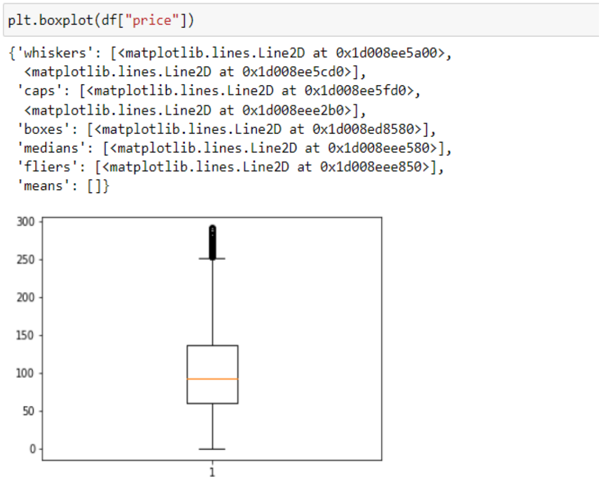
Before



```
plt.boxplot(df["price"])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1d0053da1f0>,
  <matplotlib.lines.Line2D at 0x1d0053da4c0>],
 'caps': [<matplotlib.lines.Line2D at 0x1d0053da850>,
  <matplotlib.lines.Line2D at 0x1d0053daa60>],
 'boxes': [<matplotlib.lines.Line2D at 0x1d004d1fee0>],
 'medians': [<matplotlib.lines.Line2D at 0x1d0053dad30>],
 'fliers': [<matplotlib.lines.Line2D at 0x1d0053e9040>],
 'means': []}
```

After



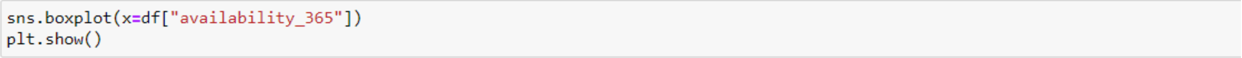
```
plt.boxplot(df["price"])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1d008ee5a00>,
  <matplotlib.lines.Line2D at 0x1d008ee5cd0>],
 'caps': [<matplotlib.lines.Line2D at 0x1d008ee5fd0>,
  <matplotlib.lines.Line2D at 0x1d008eee2b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1d008ed8580>],
 'medians': [<matplotlib.lines.Line2D at 0x1d008eee580>],
 'fliers': [<matplotlib.lines.Line2D at 0x1d008eee850>],
 'means': []}
```

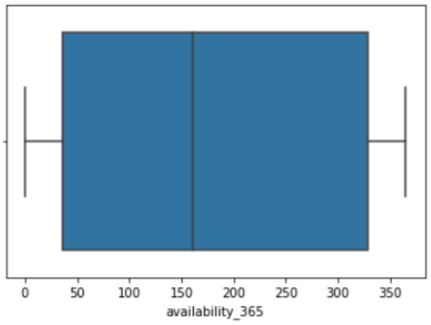
We can observe that all the outliers are now removed from the dataset, all the statistical values are now different and more accurate than previous ones so we can begin with our analysis.

UNIVARIATE ANALYSIS

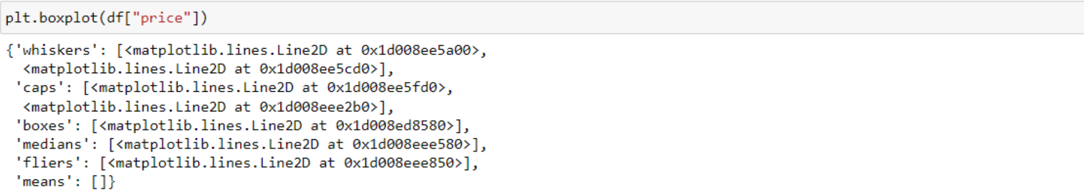
Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so in other words your data has only one variable. It doesn’t deal with causes or relationships (unlike regression) and it’s major purpose is to describe; It takes data, summarizes that data and finds patterns in the data.



```
sns.boxplot(x=df["availability_365"])
plt.show()
```

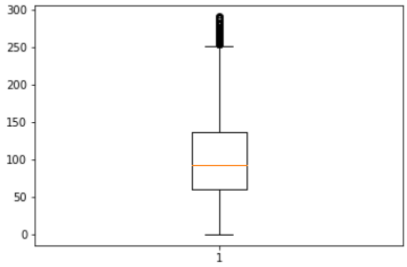


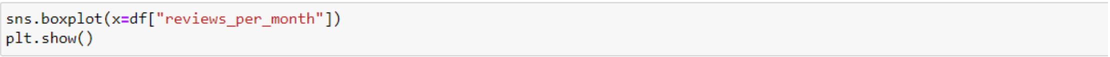
In this boxplot we can see that, numbers of availability of rooms is lie in range between 40 to 330 means rooms are available in that area.



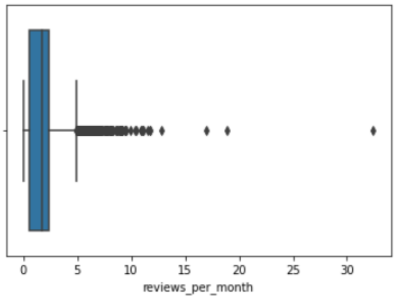
```
plt.boxplot(df["price"])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1d008ee5a00>,
  <matplotlib.lines.Line2D at 0x1d008ee5cd0>],
 'caps': [<matplotlib.lines.Line2D at 0x1d008ee5fd0>,
  <matplotlib.lines.Line2D at 0x1d008eee2b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1d008ed8580>],
 'medians': [<matplotlib.lines.Line2D at 0x1d008eee580>],
 'fliers': [<matplotlib.lines.Line2D at 0x1d008eee850>],
 'means': []}
```





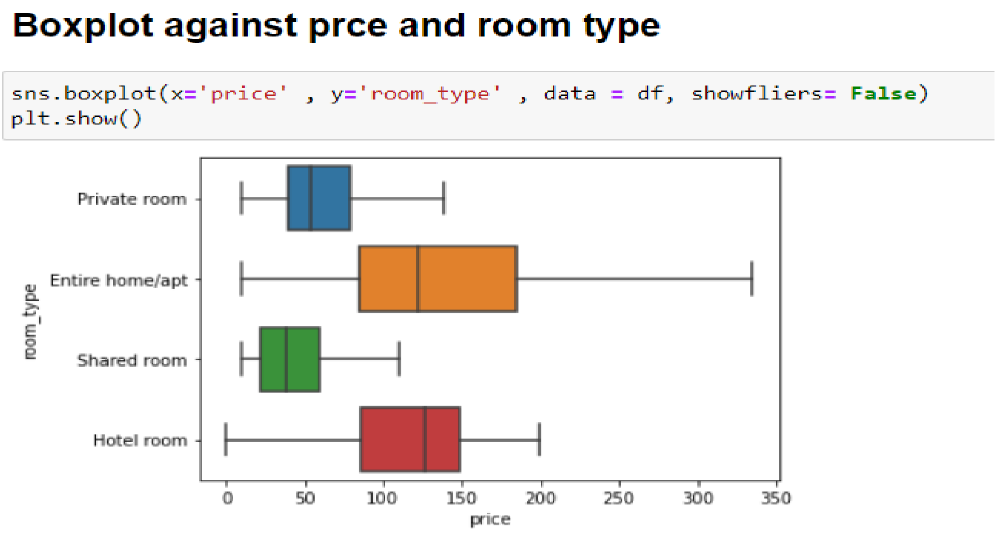
```
sns.boxplot(x=df["reviews_per_month"])
plt.show()
```



In this box plot,we can see that on an average reviews per month is lie in range between 1-5

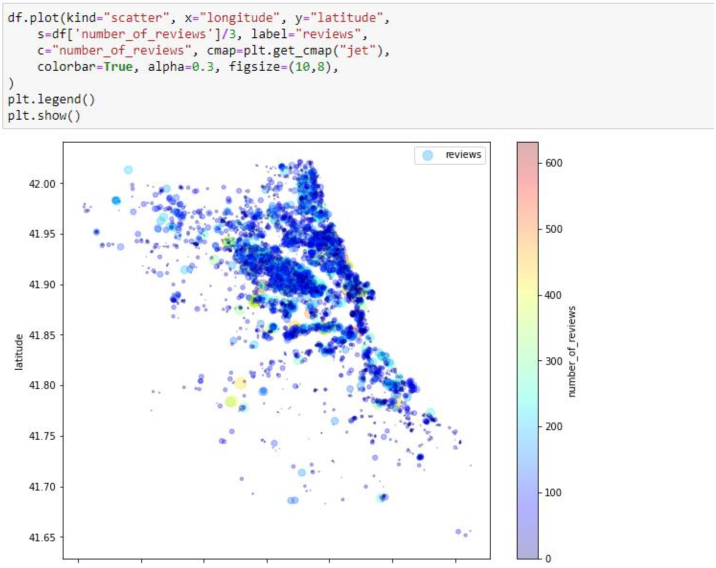
BIVARIATE ANALYSIS

Bivariate analysis refers to the analysis of two variables to determine relationships between them. Bivariate analyses are often reported in quality of life research.

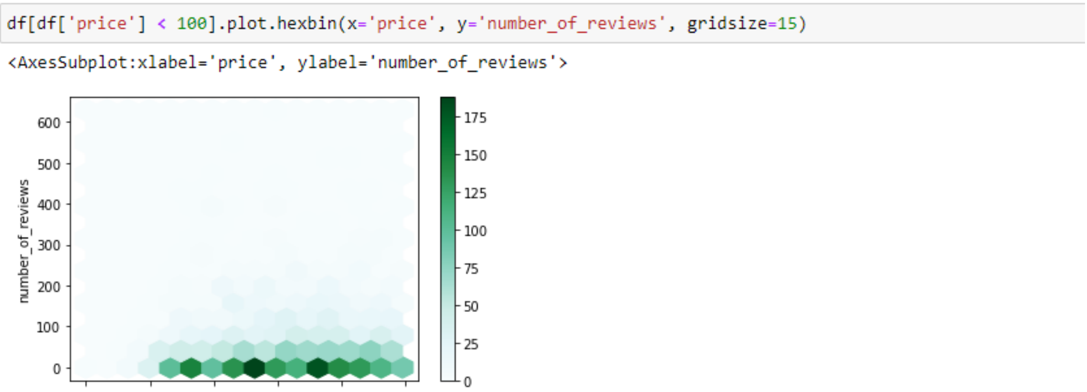


Boxplot against prce and room type

```
sns.boxplot(x='price' , y='room_type' , data = df, showfliers= False)
plt.show()
```



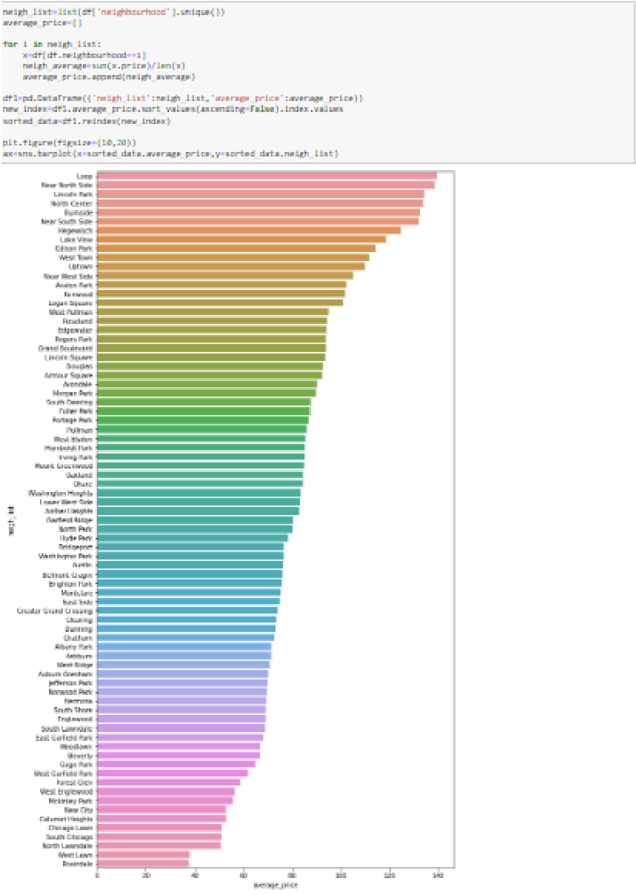
```
df.plot(kind="scatter", x="longitude", y="latitude",
    s=df['number_of_reviews']/3, label="reviews",
    c="number_of_reviews", cmap=plt.get_cmap("jet"),
    colorbar=True, alpha=0.3, figsize=(10,8),
)
plt.legend()
plt.show()
```



```
df[df['price'] < 100].plot.hexbin(x='price', y='number_of_reviews', gridsize=15)
```

```
<AxesSubplot:xlabel='price', ylabel='number_of_reviews'>
```

In this hex plot, we can see that no. of reviews is lies between 0 to 100 every every price point.

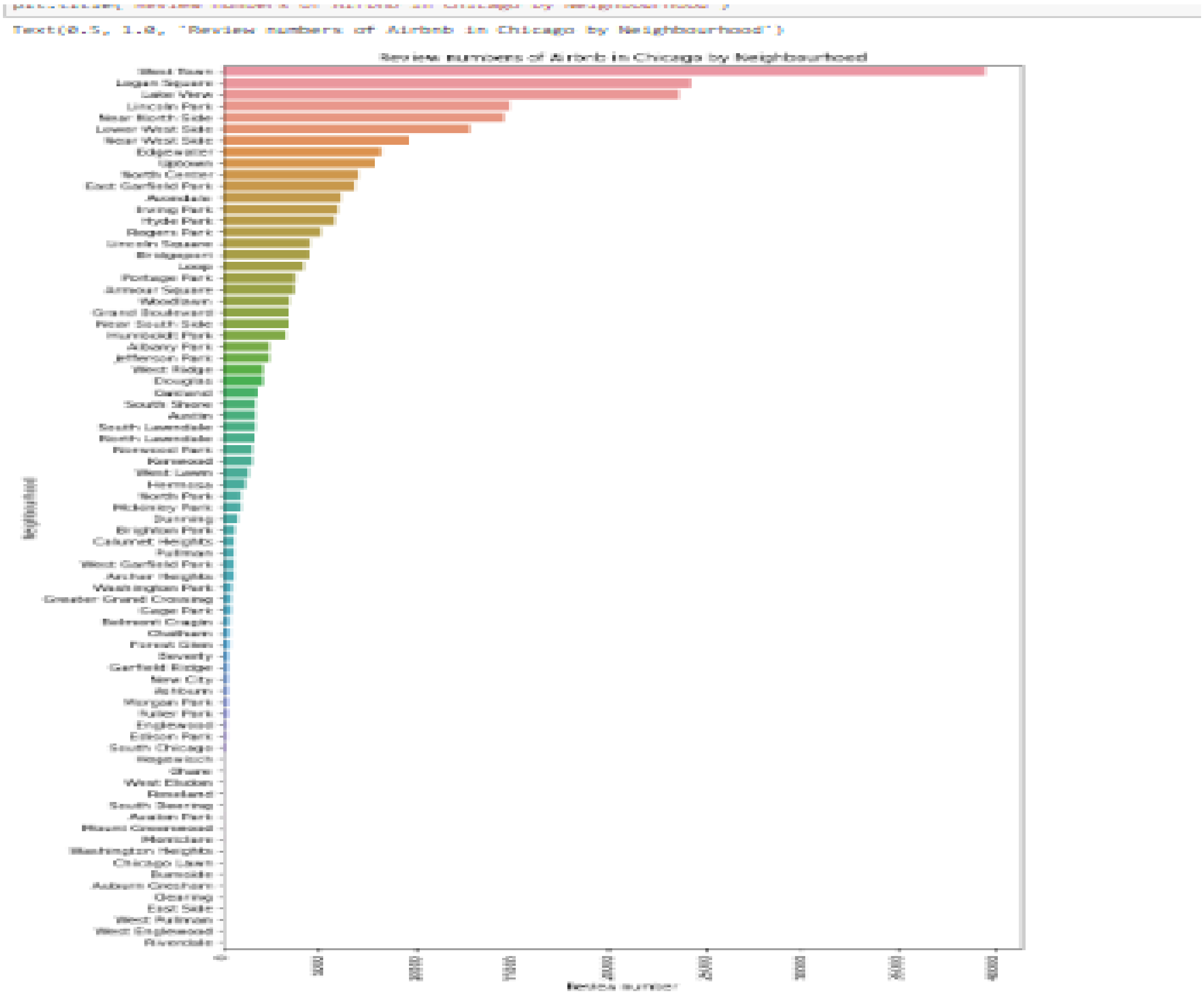


```
neigh_list=list(df['neighbourhood'].unique())
average_price=[]

for i in neigh_list:
    x=df[df.neighbourhood==i]
    neigh_average=sum(x.price)/len(x)
    average_price.append(neigh_average)

df1=pd.DataFrame({'neigh_list':neigh_list,'average_price':average_price})
new_index=df1.average_price.sort_values(ascending=False).index.values
sorted_data=df1.reindex(new_index)

plt.figure(figsize=(10,20))
ax=sns.barplot(x=sorted_data.average_price,y=sorted_data.neigh_list)
```

North and side Loop are one of the best area you can visit in Chicago, of the downtown community areas, the Near North Side has the second-largest total area after the Near West Side, the second highest number of skyscrapers (after the Loop) and the largest population.

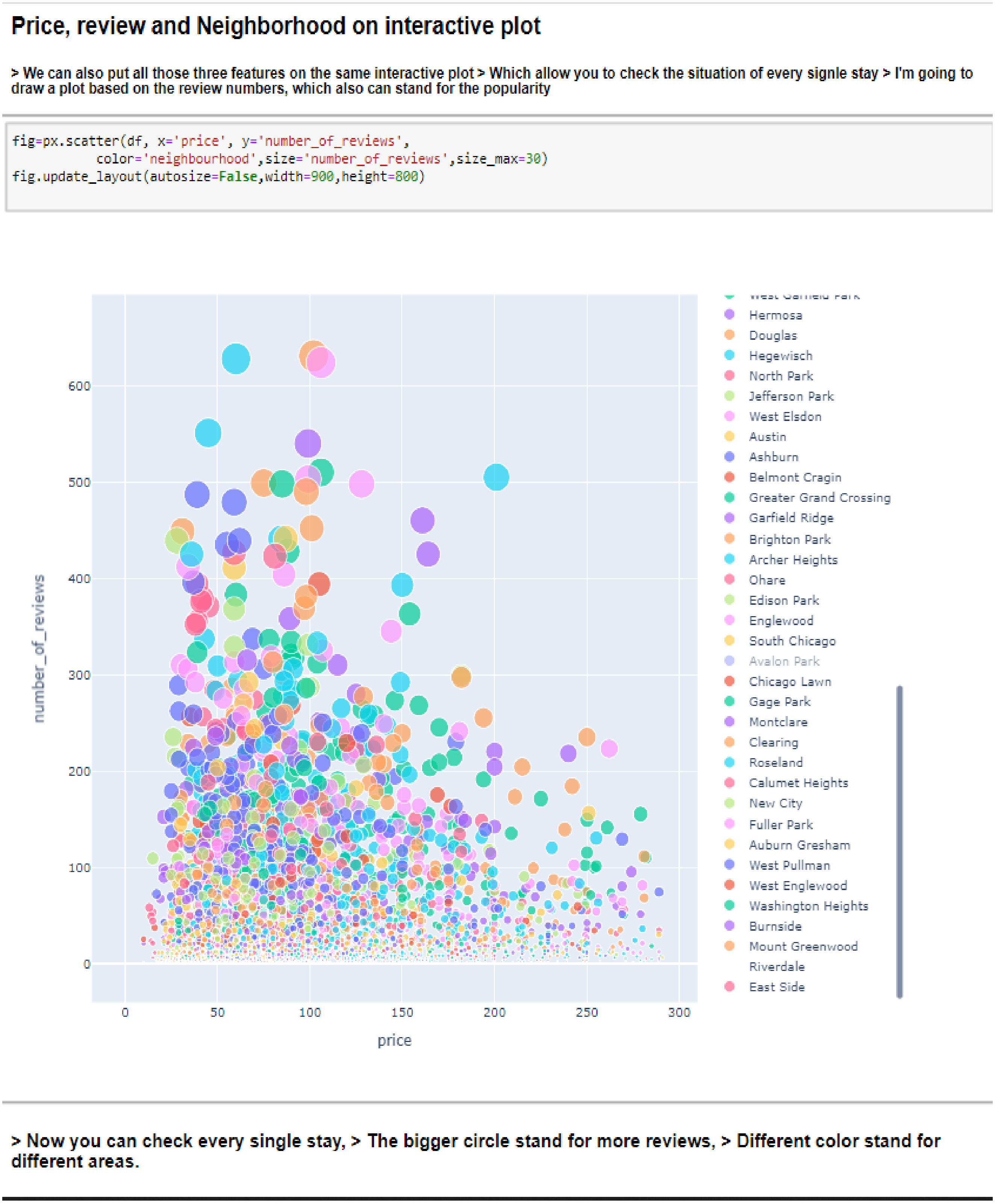
Loop is the central business district of the city and is the main section of Downtown Chicago. Home to Chicago's commercial core, it is the second largest commercial business district in North America and contains the headquarters and regional offices of several global and national businesses, retail establishments, restaurants, hotels, and theaters, as well as many of Chicago's most famous attractions.

It's actually make sense that the most expensive place has the most review, > but the places also has fair location with fair price are more popular, > especially there will be more hotels in downtown area, people probably will not choose Airbnb > So choose wisely next time you go there.



```
plt.title('Review numbers of Airbnb in Chicago by Neighbourhood')
```

```
Text(0.5, 1.0, 'Review numbers of Airbnb in Chicago by Neighbourhood')
```



Price, review and Neighborhood on interactive plot

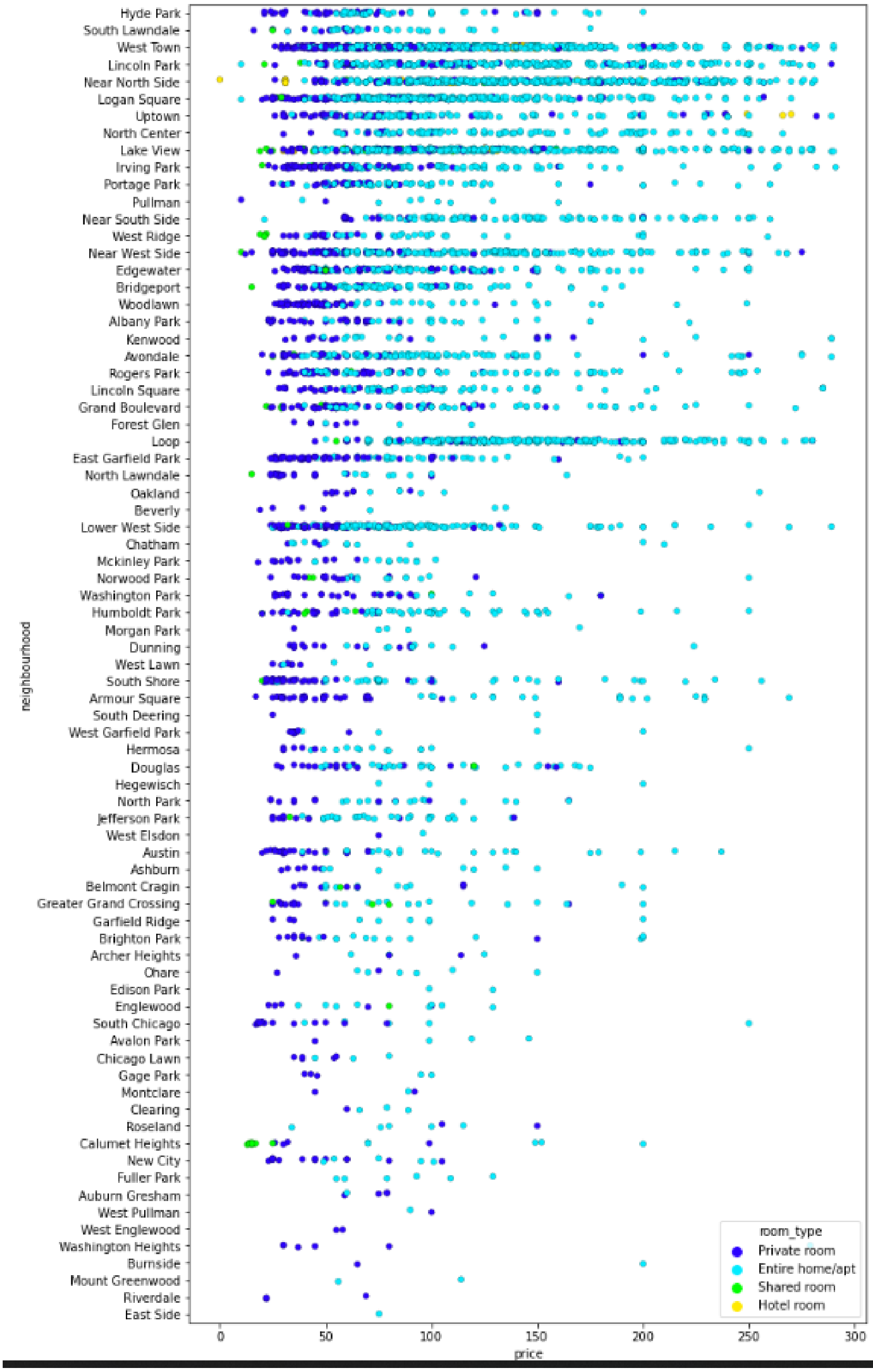
> We can also put all those three features on the same interactive plot > Which allow you to check the situation of every signle stay > I'm going to draw a plot based on the review numbers, which also can stand for the popularity

```
fig=px.scatter(df, x='price', y='number_of_reviews',
           color='neighbourhood',size='number_of_reviews',size_max=30)
fig.update_layout(autosize=False,width=900,height=800)
```

> Now you can check every single stay, > The bigger circle stand for more reviews, > Different color stand for different areas.

Scatter plot

```
<AxesSubplot:xlabel='price', ylabel='neighbourhood'>
```

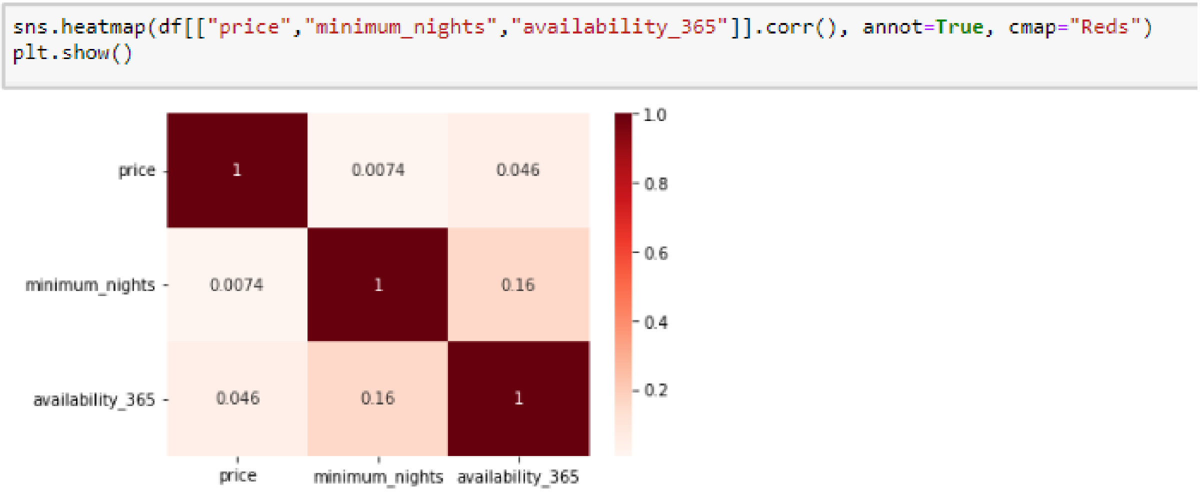


The room type

Airbnb has different room types, you can have an entire place, > but you can also choose to stay with other people, > By knowing hat kind of room type is the most popular room type, > you probably also can become a great Airbnb host.

MULTIVARIATE ANALYSIS

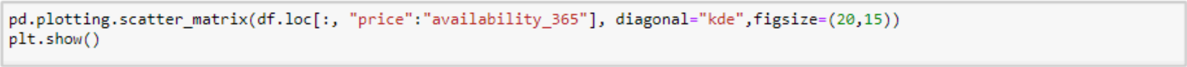
The statistical study of data where multiple measurements are made on each experimental unit and where the relationships among multivariate measurements and their structure are important.



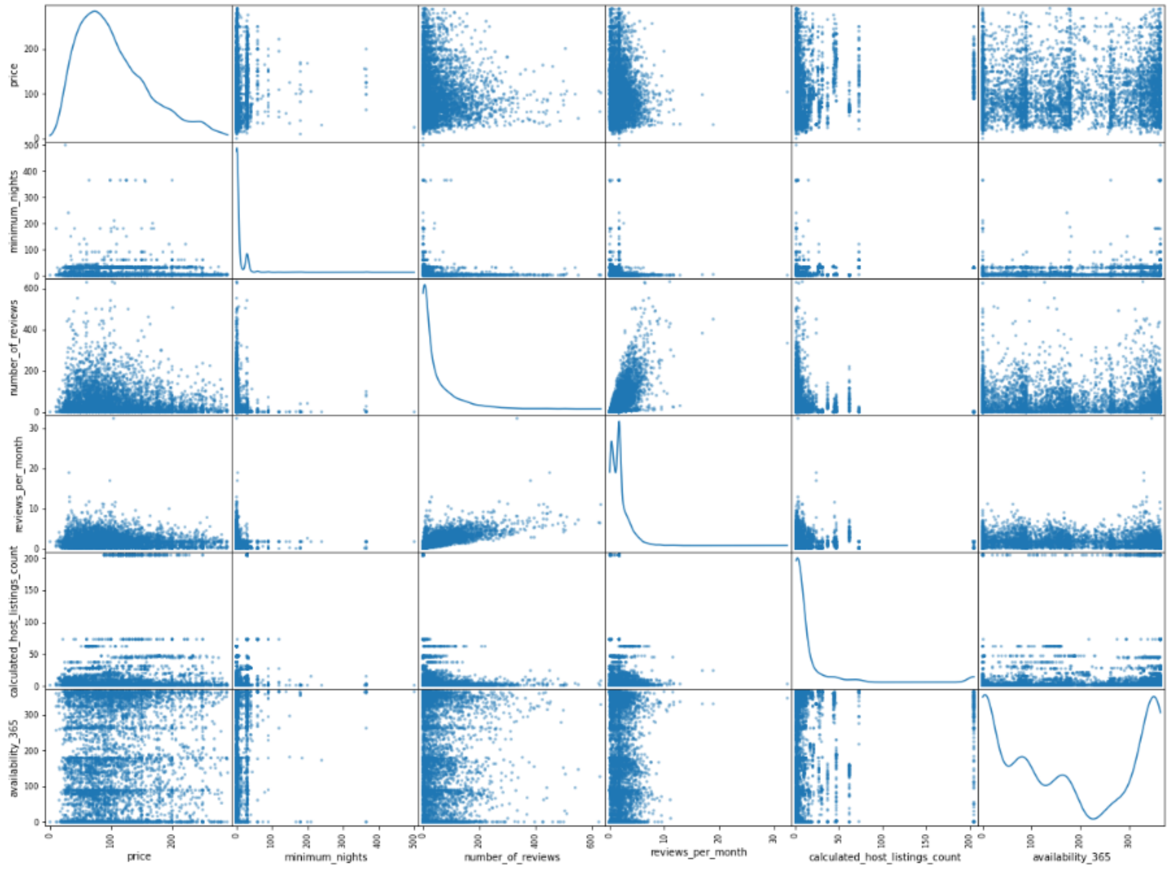
```
sns.heatmap(df[["price","minimum_nights","availability_365"]].corr(), annot=True, cmap="Reds")
plt.show()
```

	price	minimum_nights	availability_365
price	1	0.0074	0.046
minimum_nights	0.0074	1	0.16
availability_365	0.046	0.16	1

Observing the heatmap, there seems to be an interesting relation between price and availability, as they are highly co-reletated with each other. However, there seems to be low corelation with minimum nights and availability and moderate corelation between price and minimum nights. It shows that price highly affects the availability of the house, and minimum nights does not decides the availability as much.



```
pd.plotting.scatter_matrix(df.loc[:, "price":"availability_365"], diagonal="kde",figsize=(20,15))
plt.show()
```



In that scatter matrix graph, we can see the relation between the column of our dataset. Plot graph between two columns of the dataset and saw how they affect each other.

CONCLUSION

1. Seems the price is really concetrate on a specific range, about 100-250 dollars the average price will be 141 dol lars per night, which is seems a fair price for a big city.

2. North and side Loop are one of the best area you can visit in Chicago, of the downtown community areas, the Near North Side has the second-largest total area after the Near West Side, the second highest number of skyscrapers (after the Loop) and the largest population.

3.It's actually make sense that the most expensive place has the most review, but the places also has fair location with fair price are more popular, especially there will be more hotels in downtown area, people probably will not choose airbnb So choose wisely next time you go there.

4.When we talk about availibilty of room type, it seems than Entire apartment / Home is available and then private room is available and price are also fair of that room type.

5. Burnside has the wide range of price distribution by neighbourhood.

6. Rooms are available in all price range.

7. Airbnb has different room types, you can have an entire place, but you can also choose to stay with other people, By knowing hat kind of room type is the most popluar room type, you probably also can become a great airbnb host.

8. Entire home/ apartment > private room > shared room > hotel. these is the sequence of room type available in different neighbourhood cities.

9. West town has the highest number of reviews followed by logan square and lake view.

10. Loop and Near North Side has highest average price of the room and West Lawn and Riverdale has the lowest average price of the rooms.