**AIRBNB BOOKING ANALYSIS**

**RAHUL SINGH WALDIA**

**Data science trainees,**

**AlmaBetter, Bangalore**

# Data Exploration on Airbnb

## **Exploratory Data Analysis on Airbnb Data in Python**



# Introduction

Hi! My name is Rahul Singh Waldia, I am an intern who is interested in the learning data science. This article is a write-up of my personal data science project while I am learning data science using python programming language for a few weeks.

Nowadays, we live in an era where data are produced and circulated in an enormous amount. Those data can be collected and allow us to infer meaningful results and make well-informed decisions. However, as the number of data increases, we need to visualize the data to help us in conducting data analysis. By using visualization tools, we able to deliver a message to our audience and inform them about our findings.

The purpose of this article is to explore a publicly open dataset from a technology company, map the result clearly through visualization tools, and give new insight to the public and other relevant parties. To make the topics on each article more focused, this article will be divided into a series of several articles.

For the first article, we will explore and visualize the dataset from Airbnb in using basic exploratory data analysis techniques. We will be finding out the distribution of every Airbnb listing based on their location, including their price range, room type, listing name, and other related factors.

**What is Exploratory Data Analysis?**

I’m referring to Terence S, a data scientist, on his [explanation of exploratory data analysis.](https://towardsdatascience.com/an-extensive-guide-to-exploratory-data-analysis-ddd99a03199e) To simplify, exploratory data analysis(EDA), also known as Data Exploration is a step in the Data Analysis process, where several techniques are used to better understand the dataset being used.

Some of the techniques are:

* Extracting important variables and leaving behind useless variables
* Identifying outliers, missing values, and human error
* Understanding the data, maximizing our insight on a dataset and minimizing potential error that may occur later in the process

By conducting EDA, we can turn an almost useable or unusable dataset into a useable dataset.

Main components of Exploratory Data Analysis:

1. Acquire and loading data
2. Cleaning dataset
3. Exploring and Visualizing Data

**Why Airbnb?**

Since 2008, guests and hosts have used Airbnb to expand on traveling possibilities and present a more unique, personalized way of experiencing the world. Today, Airbnb became one of a kind service that is used and recognized by the whole world. Data analysis on millions of listings provided through Airbnb is a crucial factor for the company. These millions of listings generate a lot of data — data that can be analyzed.

# 1. Acquire and loading data

For this project, we are using jupyter notebook IDE with a python programming language to write our script. IDE or Integrated Development Environment is a software application used for software development.

To get the data, we are using Airbnb data that publicly shared on the internet under the Creative Commons License. Before we are able to load the data into our IDE, first we need to import various external libraries/modules that needed for visualization and analysis.

**a. Load python libraries**

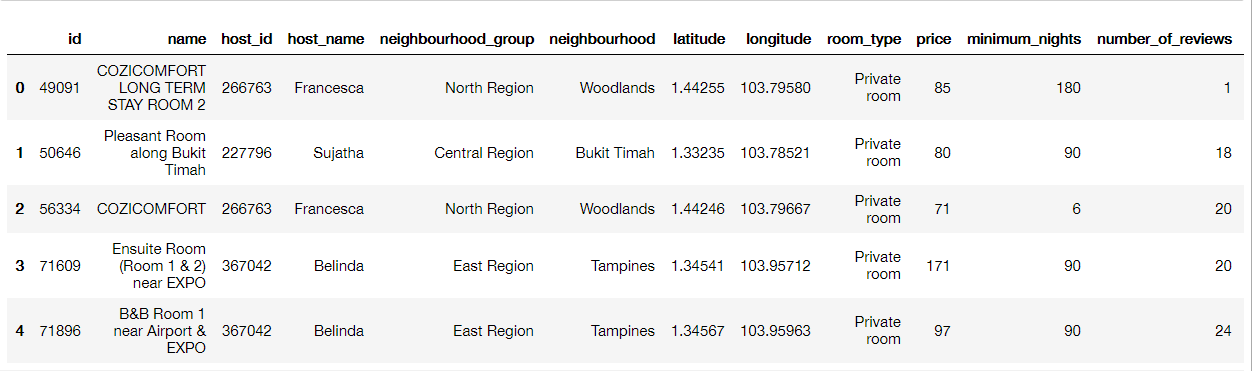
* **Pandas** and **Numpy** library used for data analysis
* **Matplotlib** and **Seaborn** library used for data visualization

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import matplotlib.image as mpimg  
%matplotlib inline  
import seaborn as sns

**b. Load dataset**

To load the dataset, we use pandas library and function to read the CSV file of data = pd.read\_csv(r'/content/sample\_data/Airbnb.csv')

**airbnb.head()**



**c. Understanding data**

After we load the dataset, we need to understand the dataset by using various techniques. First, we need to look for information on how big is our dataset. By using **shape** attributes, we get to know our data size from a number of rows which consist of listing index, and the number of columns with the content of every features related to the index.

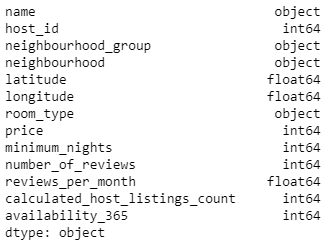
airbnb.shape

https://miro.medium.com/max/166/1*Bwxi8McMC_HpXBEVKDpOAw.png

(number of rows, number of columns)

Then we check all the data type of every column if it already matches our requirement. For instance, we need a numerical data type (integer and float) on the longitude and latitude, for listing names we need to make sure the data is using string/object data type.

airbnb.dtypes



Data Type Information

We found out that our dataset has **7395 listings**. The features include listing name, host id, location information, location coordinate, room type, the price per night, and so on.

Next, we look up all the unique values of the ‘neighbourhood\_group’ that is consists of a list of all the Singapore region

airbnb['neighbourhood\_group'].unique()

Region Area Map

From the list above, we see that Singapore has 5 region area.

The region area is divided further by the Urban Redevelopment Authority (URA) into 55 areas called planning areas for urban planning purposes. We will use the ‘neighbourhood’ columns to look at which planning area that has the Airbnb listing.

airbnb['neighbourhood'].unique()

neighbourhood

Planning Area Map

Now, we know that 43 planning areas have the Airbnb listing.

We also look up the ‘room\_type’ columns for each room type of the listing

airbnb['room\_type'].unique()

https://miro.medium.com/max/1152/1*1gdcj5ZK6nIaps_33V_Dtw.png



room type

From the list above, we see that Airbnb have 4 room type. Based on the information on the Airbnb website, the definition of each room type are:

* **Private room**

Guests have exclusive access to the bedroom/sleeping area of the listing. Other parts area such as the living room, kitchen, and bathroom are likely open either to the host even to other guests.

* **Entire home/apt**

Guests have the whole place for themselves. It usually includes a bedroom, bathroom, and kitchen.

* **Shared Room**

Guest sleep in a bedroom or a common area that could be shared with others.

* **Hotel Room**

A typical hotel room with its facilities. Since 2018, Airbnb allows some boutique hotels and high rated independent hotel to list their rooms on their site.

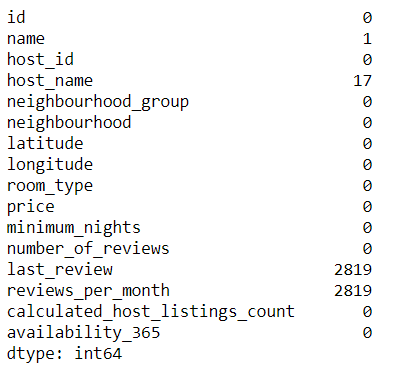
# 2. Cleaning dataset

The next step is cleaning up the data, oftentimes the data we load have various faults, such as typo, missing value, incomplete data, etc. By doing cleaning up, the data quality will have better quality to be used for further analysis.

**a. Checking column with missing values**

Let’s check first if there are any missing values within our dataset

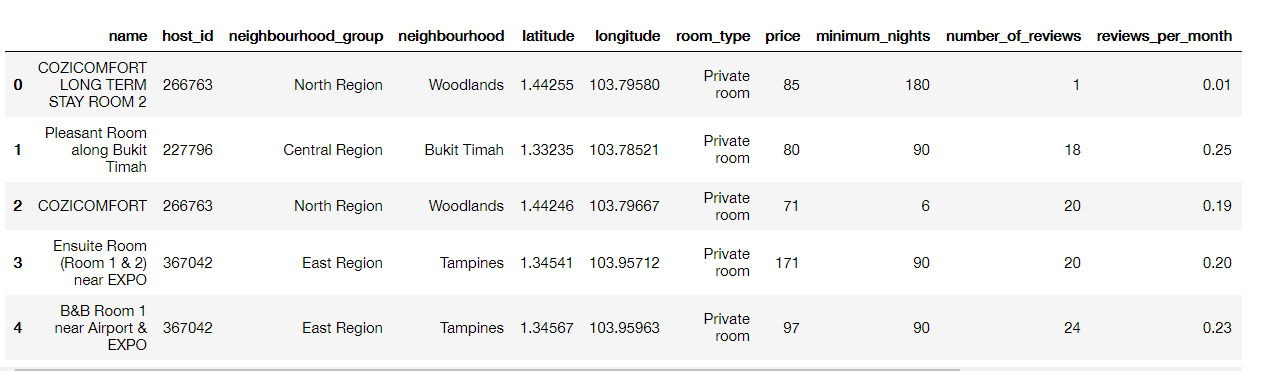
**airbnb.isnull().sum()**



**b. Removing redundant variables**

In our case, the missing values that are observed do not need too much treatment. Looking into our dataset, we can state columns ‘ name’ and ‘host\_name’, ‘last\_review’ are irrelevant and unethical for further data exploration analysis. Therefore, we can get rid of those columns.

airbnb.drop(['id','host\_name','last\_review'],axis=1,inplace=True)  
**airbnb.head()**



**c. Replacing all the missing values**

Next, we need to replace all the missing values in the ‘review\_per\_month’ column with 0 (zero) to make sure the missing values do not interfere with our analysis

airbnb['reviews\_per\_month'].fillna(0,inplace=True)

# 3. Exploring and visualizing data

After we clean up the data, the next step is exploring the data by visualizing and analyzing the values of the features, explaining the process and the results.

For our case, we will look up a various listing category consisting of each biggest value, visualize the listing distribution using a map, create a room type proportion for each area, looking for selling value from their listing name, and finding the average price of the most popular listing.

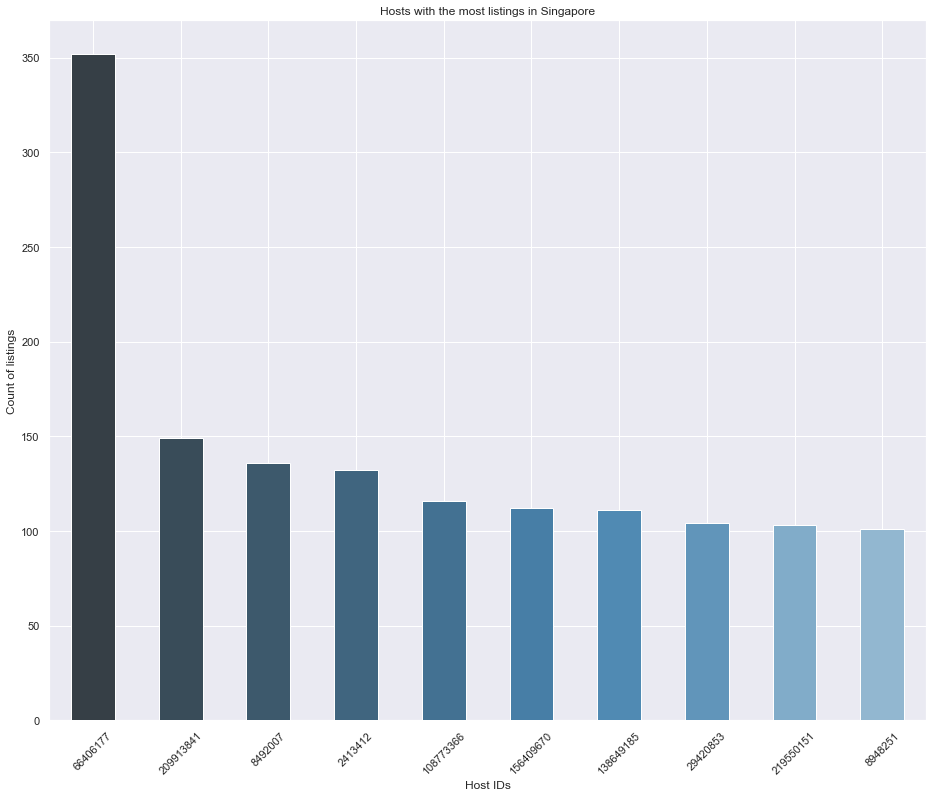
**a. Top listing counts**

First, we skip the first column of ‘name’ and begin from the ‘host\_id’ column. Then we slice the top 10 hosts in terms of listing count

top\_host\_id = airbnb['host\_id'].value\_counts().head(10)

Next, we set the figure size and setting it up for data visualizations plot using a bar chart

sns.set(rc={'figure.figsize':(10,8)})viz\_bar = top\_host\_id.plot(kind='bar')  
viz\_bar.set\_title('Hosts with the most listings in Singapore')  
viz\_bar.set\_xlabel('Host IDs')  
viz\_bar.set\_ylabel('Count of listings')  
viz\_bar.set\_xticklabels(viz\_bar.get\_xticklabels(), rotation=45)



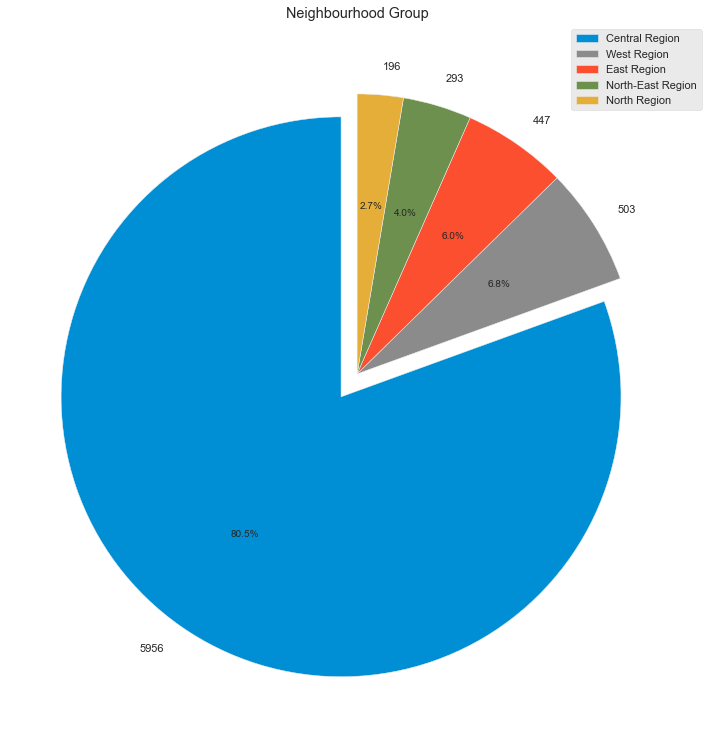
From the chart above, we can see the total of **top 10 hosts is almost 20%( 1416 listings) of the whole dataset (7395 listings)**. Even one of the hosts has more than 350 listings!

**b. Top Region Area**

Next, we visualize the proportion of the listing count on each region area using the ‘neighbourhood\_group’ columns

labels = airbnb.neighbourhood\_group.value\_counts().index

colors = ['#008fd5','#fc4f30','#e5ae38','#6d904f','#8b8b8b']  
explode = (0.1,0,0,0,0)shape = airbnb.neighbourhood\_group.value\_counts().valuesplt.figure(figsize=(12,12))  
plt.pie(shape, explode = explode, labels=shape, colors= colors, autopct = '%1.1f%%', startangle=90)  
plt.legend(labels)  
plt.title('Neighbourhood Group')  
**plt.show()**

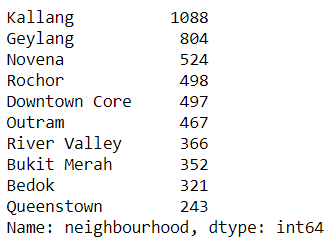


From the chart above, we see the **Central Region has the most listings** with almost 6000 listings number, covering more than 80% of the total listings.

**c. Top Planning Areas**

Next, we look up the top 10 planning areas that have the highest number of listings

airbnb.neighbourhood.value\_counts().head(10)

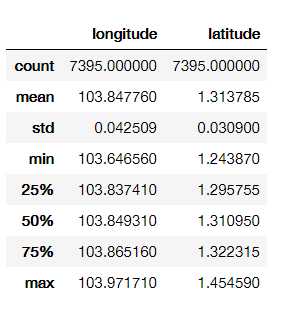


As we can see, **Kallang has the highest number of listings.**We also found that**9 out of the top 10 planning areas are located in the Central Region**, with Bedok located in East Region as an exception.

**d. Listing Map**

To create a map of the listing location, we will use the ‘longitude’ and ‘latitude’ column. But first, we need to check the values within the column

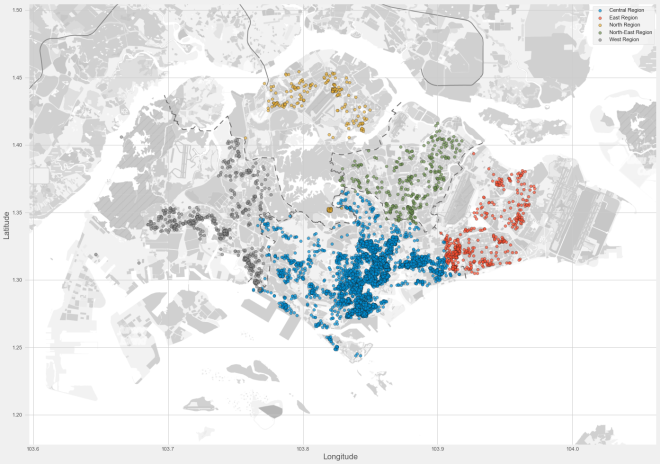
coord = airbnb.loc[:,['longitude','latitude']]  
**coord.describe()**



From the data above, we can see the outer values of **longitude**and **latitude** from the **min** and **max**index.

Next, we visualize the scatter plot map of every listing and group it by color on each different region

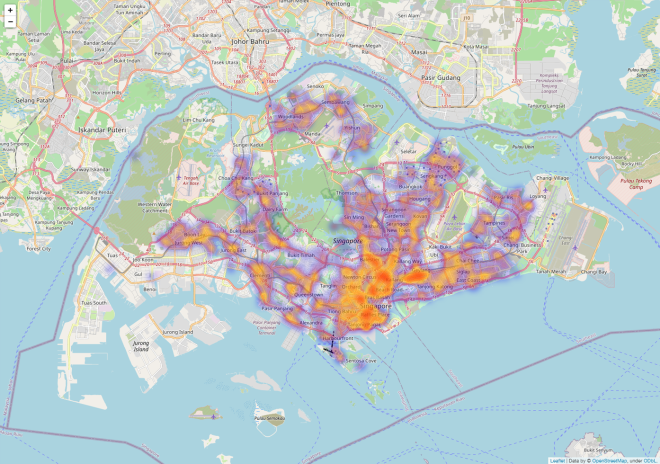
plt.figure(figsize=(18,12))  
plt.style.use('fivethirtyeight')BBox = (103.5935, 104.0625, 1.1775, 1.5050)sg\_map = plt.imread('map\_bnw.png')  
plt.imshow(sg\_map,zorder=0,extent=BBox)  
ax = plt.gca()groups = airbnb.groupby('neighbourhood\_group')  
for name,group in groups :  
 plt.scatter(group['longitude'],group['latitude'],label=name,alpha=0.5, edgecolors='k')plt.xlabel('Longitude')  
plt.ylabel('Latitude')  
plt.legend()



Listing Map

Now we can see how the listings are plotted into a map. For a better understanding of the listings density, we can use the folium heat map

**import folium  
from folium.plugins import HeatMap**map\_folium = folium.Map([1.35255,103.82580],zoom\_start=11.4)HeatMap(airbnb[['latitude','longitude']].dropna(),radius=8,gradient={0.2:'blue',0.4:'purple',0.6:'orange',1.0:'red'}).add\_to(map\_folium)  
display(map\_folium)



Listing Density Map

From the map above, we can see clearly where the**densest listing** is located, shown by the red color in the **southern area of the Central Region**. The listing density increasingly declining the more it’s farther away from the Central Region.

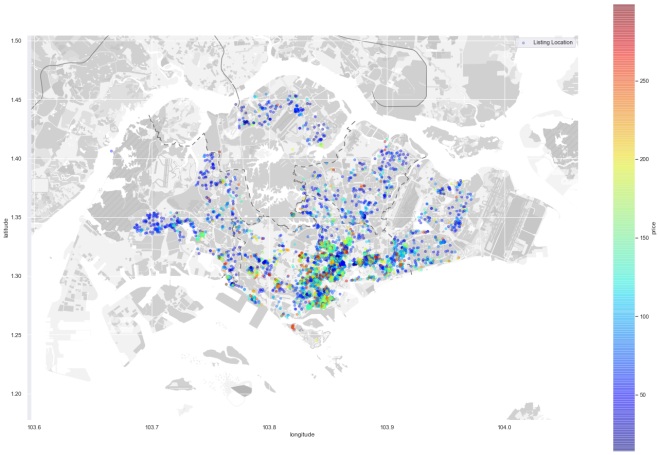
**e. Price Map**

Before we visualize the price map, we need to update the dataset by removing some of the outlier data as some data prices have value far from the IQR (interquartile range).

airbnb\_1 = airbnb[airbnb.price < 300]

Next, we visualize the scatter plot map of every listing and the difference in price range using longitude and latitude points with a price heat map.

plt.figure(figsize=(18,12))sg\_map = plt.imread('map\_bnw.png')  
plt.imshow(sg\_map,zorder=0,extent=BBox)  
ax = plt.gca()airbnb\_1.plot(kind='scatter',x='longitude',y='latitude',label='Listing Location', c='price', ax=ax, cmap=plt.get\_cmap('jet'), colorbar=True, alpha=0.4, zorder=5)**plt.legend()  
plt.show()**



Listing Price Map

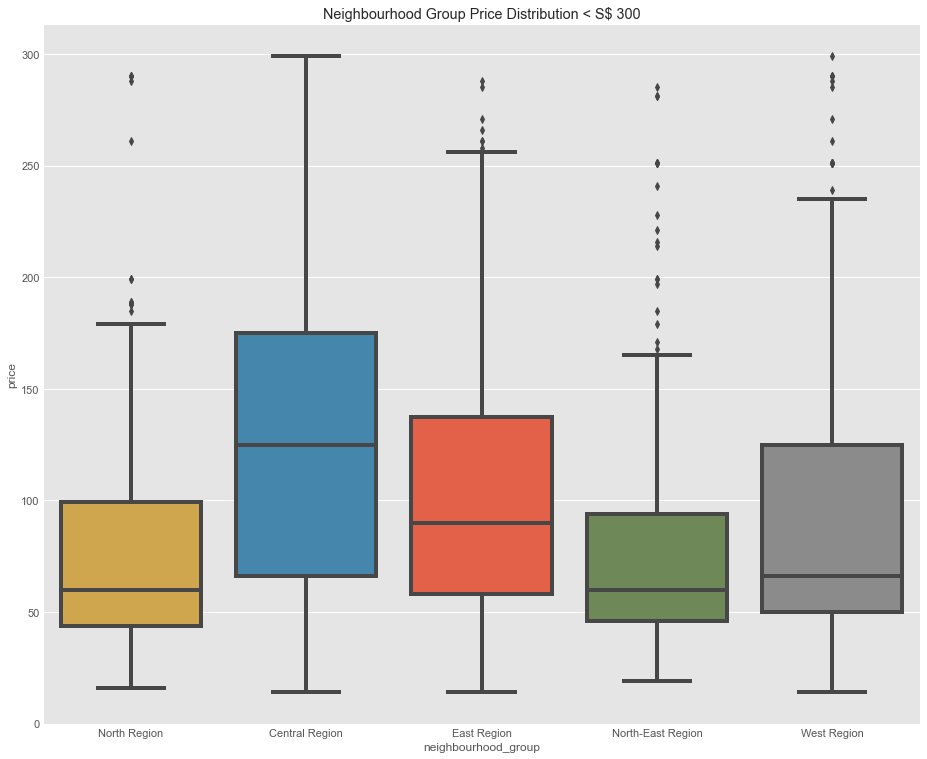
From the map above, we observe the **price relatively going up towards the center part of the Central Region** as this region is the CCR Area in Singapore.

By looking at two maps above, we could argue that the Airbnb listing price is related to the real estate market segments. But to conclude that, we need more data to do further analysis.

**f. Price Distribution**

Based on our price heat map observation, we need to visualize the price distribution using a box plot to understand more on the listing price range grouped by the ‘neighbourhood\_group’ /region area.

plt.style.use('fivethirtyeight')  
plt.figure(figsize=(14,12))  
sns.boxplot(y='price',x='neighbourhood\_group',data = airbnb\_1)  
plt.title('Neighbourhood Group Price Distribution < S$ 300')  
**plt.show()**



From the data above, we see the **Central Region has the most expensive price per night** with a median**S$ 130**.

**g. Top listing words**

Next, we will explore deeper on the property detail by finding out what the most used word in the listing name. The most used word could represent the selling value of their property for the prospective guests. First, we will create a function to collect the words.

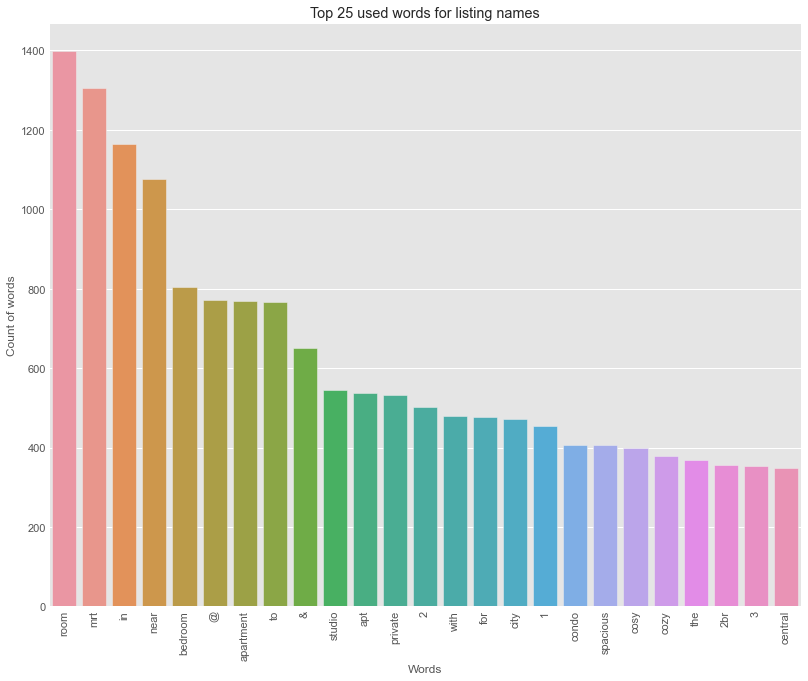
**#Crete empty list where we are going to put the name strings**  
names=[]**#Getting name string from 'name' column and appending it to the empty list**  
for name in airbnb.name:  
 names.append(name)**#Setting a function to split name strings into seperate words**  
def split\_name(name):  
 s = str(name).split()  
 return s**#Create empty list where we are going to count the words**  
names\_count = []**#Getting name string to appending it to the names\_count list**  
for n in names:  
 for word in split\_name(n):  
 word = word.lower()  
 names\_count.append(word)

We need to import **counter** library to count and generate raw data which contains the top 25 words used by the host

**from collections import Counter**top\_25 = Counter(names\_count).most\_common()  
top\_25 = top\_25[:25]

Then, we convert the data into DataFrame and visualize our findings

word\_count\_data = pd.DataFrame(top\_25)  
word\_count\_data.rename(columns={0:'Words',1:'Counts'},inplace=True)viz\_count = sns.barplot(x='Words',y='Counts', data = word\_count\_data)  
viz\_count.set\_title('Top 25 used words for listing names')  
viz\_count.set\_ylabel('Count of words')  
viz\_count.set\_xlabel('Words')  
viz\_count.set\_xticklabels(viz\_count.get\_xticklabels(),rotation = 90)



Top 25 used words for listing names

From the chart above, we see the top 25 words used in the listing name. We can use the word cloud visualization method to help us better understandthe chart.

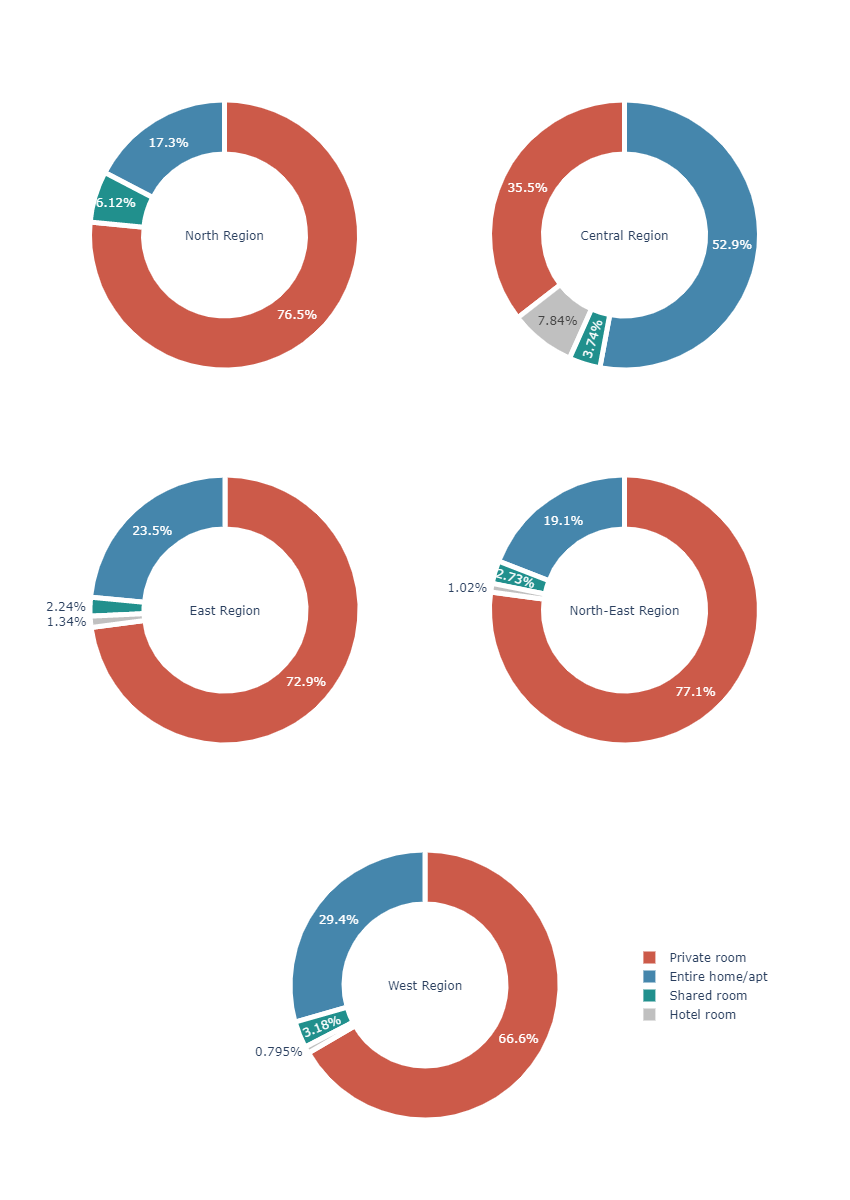
**from wordcloud import WordCloud, ImageColorGenerator**text = ' '.join(str(n).lower() for n in airbnb.name)**#Generate wordcloud image**  
wordcloud = WordCloud(max\_words=200, background\_color = 'white').generate(text)  
plt.figure(figsize=(25,20))**#Display the image**  
plt.imshow(wordcloud, interpolation='bilinear')  
plt.axis('off')  
plt.show()

As we can see, most of the listing selling values are related to the proximity or connection to public facilities such as MRT and center of activities, shown by ‘mrt’, ‘near’, ‘to’, ‘city’ , ‘walk to’ keyword. Interesting to see how the room condition falls behind those values, shown by the ‘spacious’, ‘cosy’, ’cozy’ on the lower rank of the chart.

**h. Room type details**

Next, we will visualize all listing’s room type proportions from each region area using Plotly API library for graph visualization

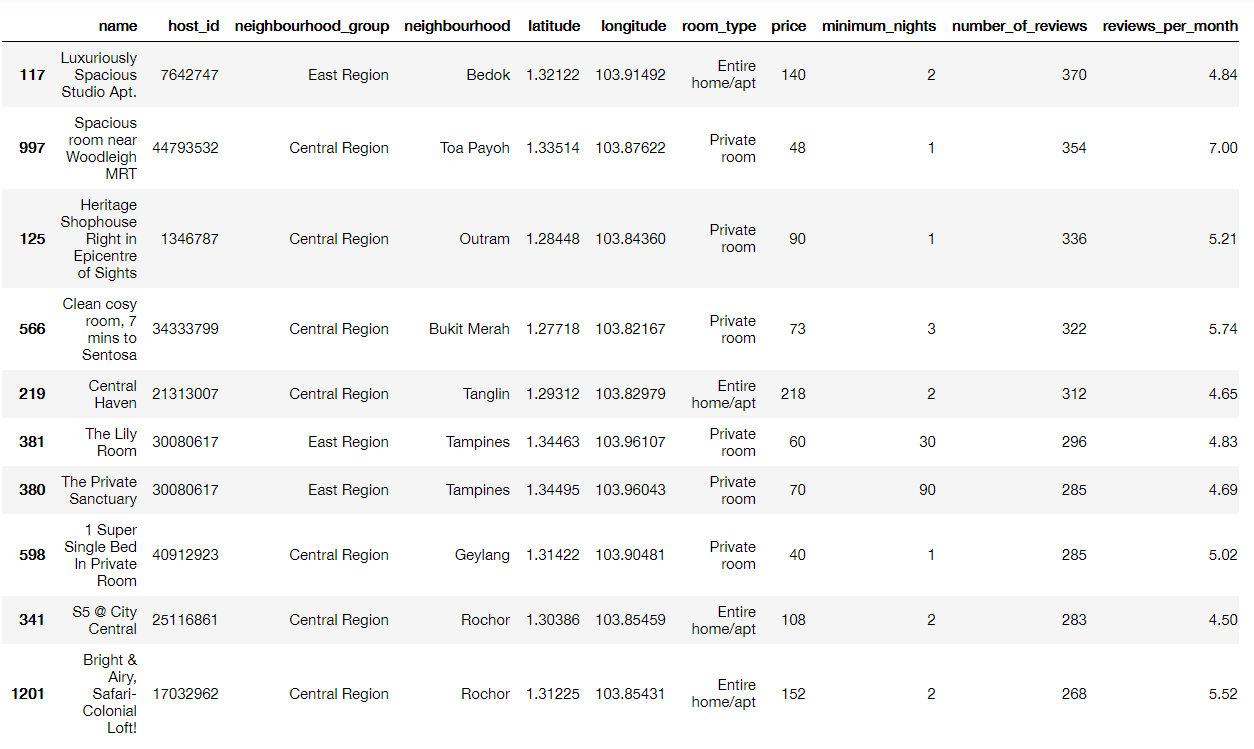
**import plotly.offline as pyo  
import plotly.graph\_objs as go#Setting up the color pallete**  
color\_dict = {'Private room': '#cc5a49', 'Entire home/apt' : '#4586ac', 'Shared room' : '#21908d', 'Hotel room' : '#C0C0C0' }**#Group the room type using 'neighbourhood\_group' as an index**  
airbnb\_types=airbnb.groupby(['neighbourhood\_group', 'room\_type']).size()**#Create function to plot room type proportion on all region area**for region in airbnb.neighbourhood\_group.unique():  
   
 plt.figure(figsize=(24,12))  
   
 airbnb\_reg=airbnb\_types[region]  
 labels = airbnb\_reg.index  
 sizes = airbnb\_reg.values  
   
 colors = [color\_dict[x] for x in labels]  
   
 plot\_num = 321  
 plt.subplot(plot\_num)  
 reg\_ch = go.Figure(data = [go.Pie(labels = labels, values = sizes, hole = 0.6)])  
 reg\_ch.update\_traces(title = reg, marker=dict(colors=colors))  
 reg\_ch.show()  
   
 plot\_num += 1



Room type

We can see the Central Region is the only region that dominated by the entire home/apt type, with the rest of the region is dominated by private room type. Overall, the hotel type is the least listing on each region, since Airbnb has only been accepting hotel listing on 2018.

airbnb.nlargest(10, 'number\_of\_reviews')



Voila! Those are the 10 most popular listings. Again, we found out the majority of the most reviewed listing is located in the Central Region, with 7 out of 10 listings.

**j. Average price per night**

Lastly, we will calculate the average price per night of the 10 most popular listings

price\_avg = top\_review.price.mean()  
print('Average price per night: S$ {}'.format(price\_avg))

https://miro.medium.com/max/672/1*_lFzJobEn15EFsG3V1ZJGA.png

From this output, we can observe that the **top 10 most popular listings on Airbnb Singapore have a price average of S$ 99.9** with most of the listings under S$ 90, and 6/10 of them are ‘Private Room’ type, top reviewed listings have a total 3111 reviews.

# **Conclusion**

The properties have large differences in prices. Separating the dataset by price categories is useful for the analysis. The most interesting variables regarding price prediction are: Location Room type calculated\_host\_listings\_count Number of review Price prediction models are not performing well Best score is 0.55 Prediction are nore accurate for price under $175 (75% of the dataset) Using categorical ecnoded data did not improve the model Possible next steps The next step could be to transform continuous variables into categorical variables as it can help capture non-linear relations. I doubt a neural network would be useful here because the number of observations is limited.

CodeText

**Reference:**

1. <https://www.channelnewsasia.com/news/business/airbnb-records-30-growth-rate-in-first-quarter-on-booking-11817260>
2. <https://www.airbnb.com/help/article/5/what-does-the-room-type-of-a-listing-mean>