Airbnb Case Study

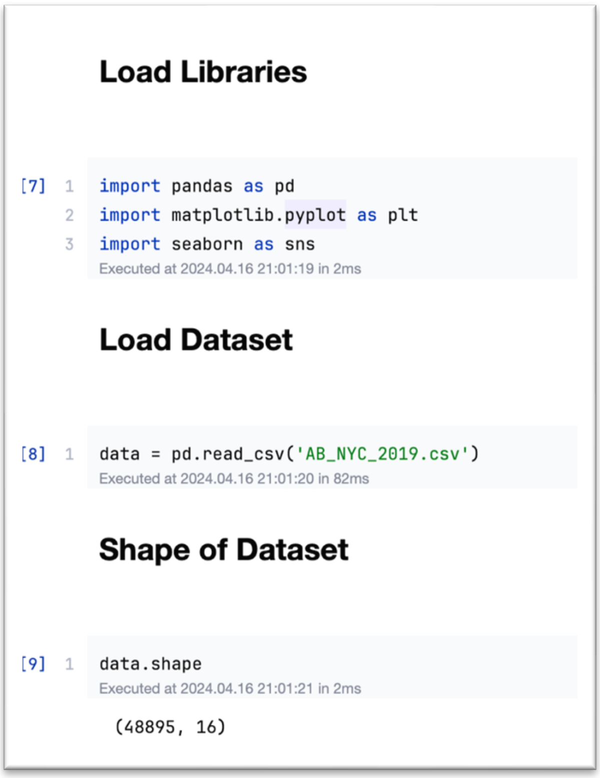
New York | 2019

Tools

- **Analysis**: Jupyter Notebook
- **Visualization**: Tableau

Dataset

- **File**: AB_NYC_2019.csv
- **Size**: 48,995 Rows and 16 columns



Load Libraries

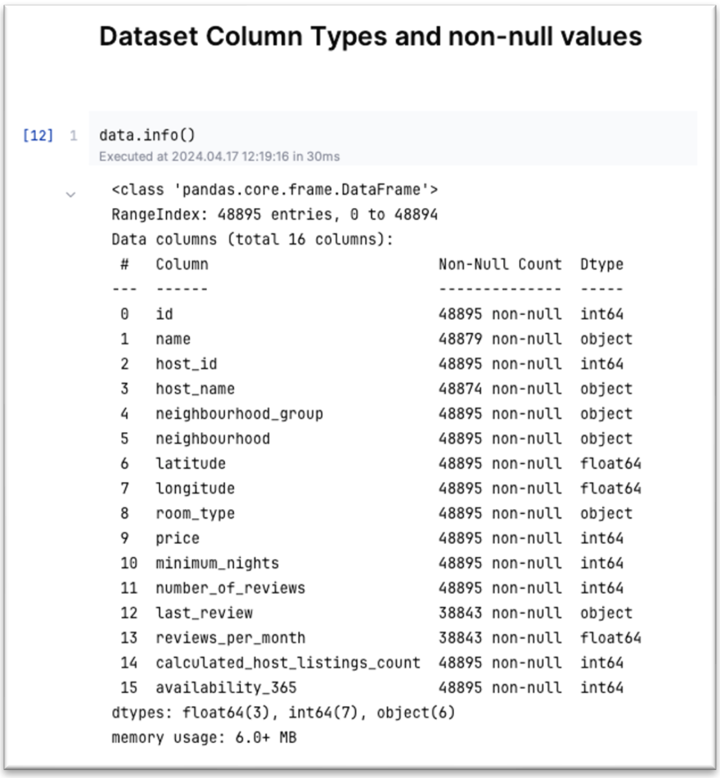
```
[7]
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
Executed at 2024.04.16 21:01:19 in 2ms
```

Load Dataset

```
[8]
data = pd.read_csv('AB_NYC_2019.csv')
Executed at 2024.04.16 21:01:20 in 82ms
```

Shape of Dataset

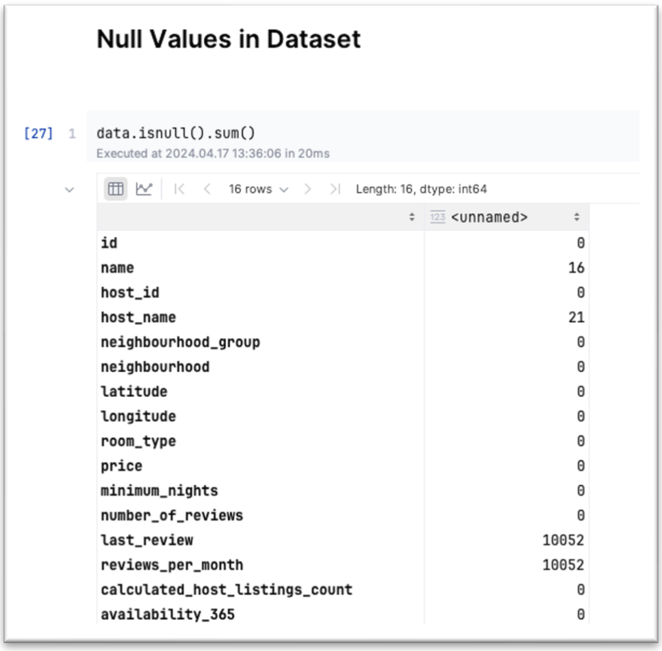
```
[9]
data.shape
Executed at 2024.04.16 21:01:21 in 2ms

(48895, 16)
```



Dataset Column Types and non-null values

```
[12]
data.info()
Executed at 2024.04.17 12:19:16 in 30ms

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              48895 non-null  int64
 1   name                            48879 non-null  object
 2   host_id                         48895 non-null  int64
 3   host_name                       48874 non-null  object
 4   neighbourhood_group             48895 non-null  object
 5   neighbourhood                   48895 non-null  object
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object
 9   price                           48895 non-null  int64
 10  minimum_nights                  48895 non-null  int64
 11  number_of_reviews               48895 non-null  int64
 12  last_review                     38843 non-null  object
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64
 15  availability_365                48895 non-null  int64
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

- **Null Values**



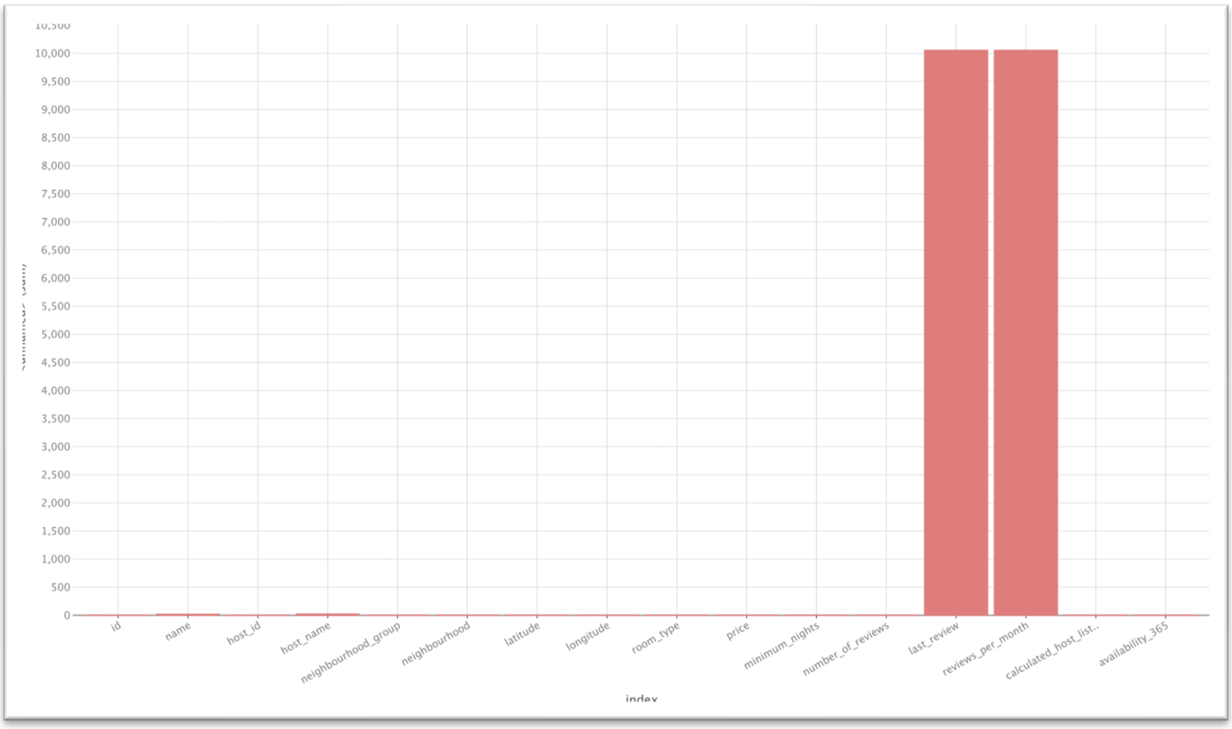
Null Values in Dataset

```
[27] data.isnull().sum()
```

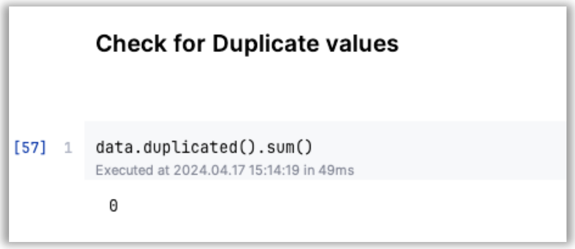
Executed at 2024.04.17 13:36:06 in 20ms

16 rows Length: 16, dtype: int64

	<unnamed>
id	0
name	16
host_id	0
host_name	21
neighbourhood_group	0
neighbourhood	0
latitude	0
longitude	0
room_type	0
price	0
minimum_nights	0
number_of_reviews	0
last_review	10052
reviews_per_month	10052
calculated_host_listings_count	0
availability_365	0



- **Duplicate Values**



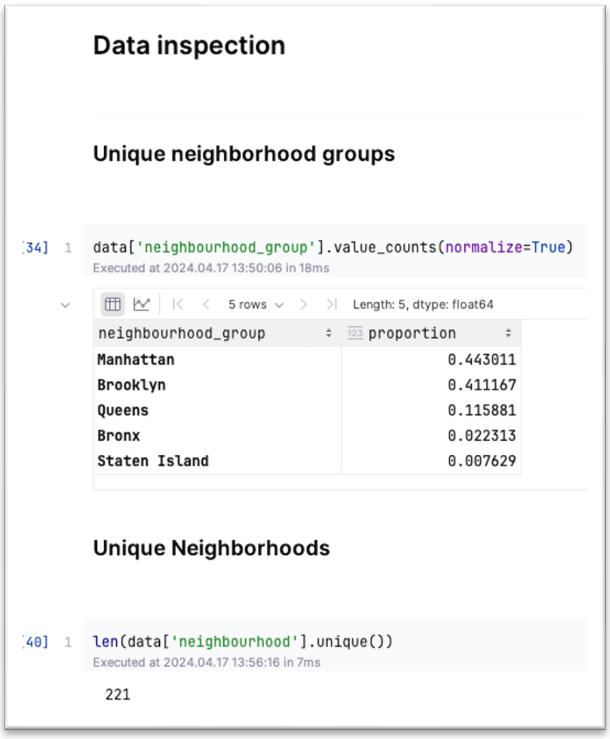
Check for Duplicate values

```
[57] data.duplicated().sum()
```

Executed at 2024.04.17 15:14:19 in 49ms

```
0
```

- **Unique Values** (Categorical variables)



Data inspection

Unique neighborhood groups

```
[34]  data['neighbourhood_group'].value_counts(normalize=True)
```

Executed at 2024.04.17 13:50:06 in 18ms

5 rows   Length: 5, dtype: float64

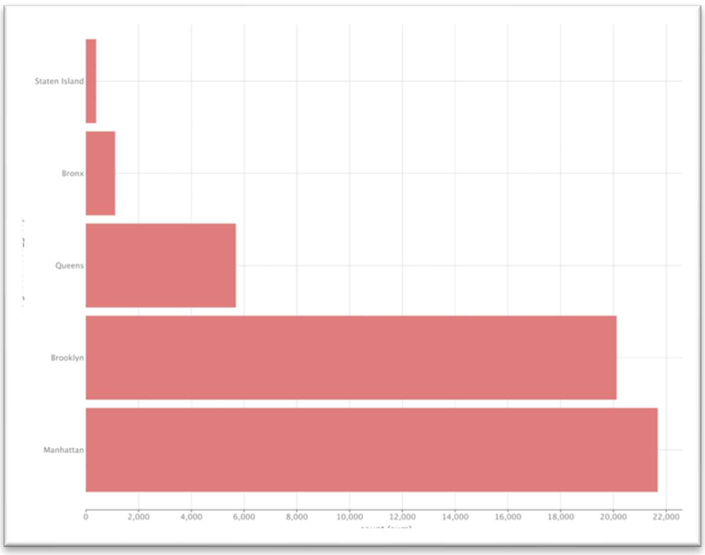
neighbourhood_group	proportion
Manhattan	0.443011
Brooklyn	0.411167
Queens	0.115881
Bronx	0.022313
Staten Island	0.007629

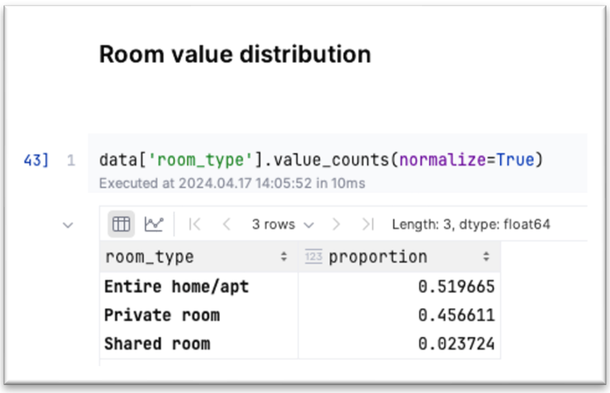
Unique Neighborhoods

```
[40]  len(data['neighbourhood'].unique())
```

Executed at 2024.04.17 13:56:16 in 7ms

```
221
```





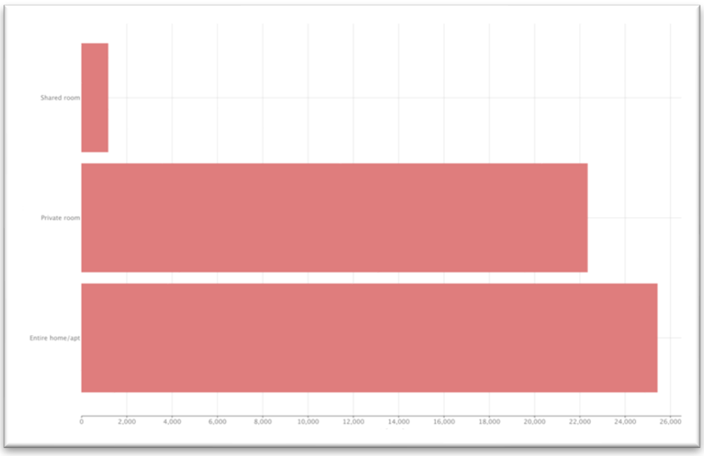
Room value distribution

```
[43]  data['room_type'].value_counts(normalize=True)
```

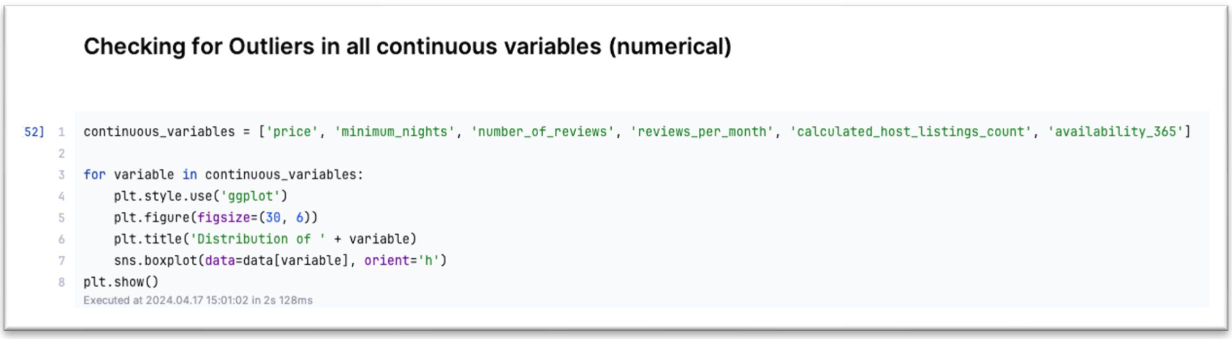
Executed at 2024.04.17 14:05:52 in 10ms

3 rows   Length: 3, dtype: float64

room_type	proportion
Entire home/apt	0.519665
Private room	0.456611
Shared room	0.023724



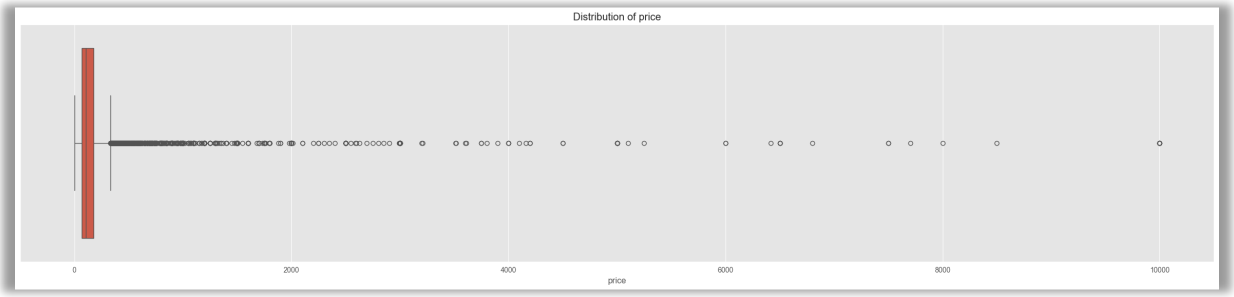
- **Checking Outliers** (Continuous variables)

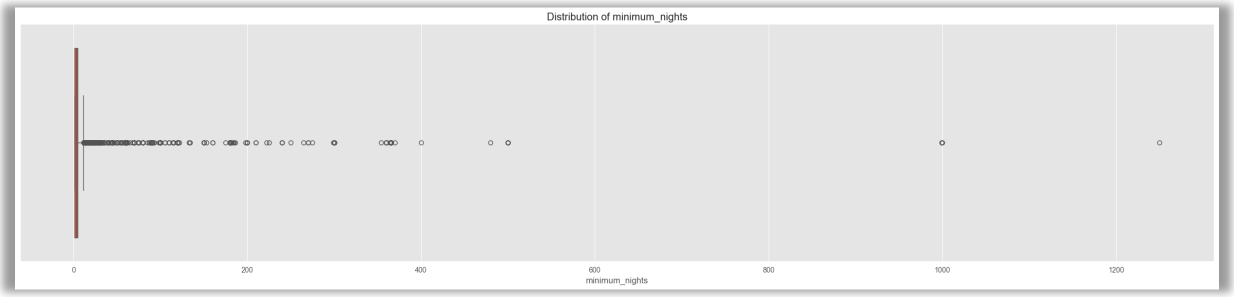


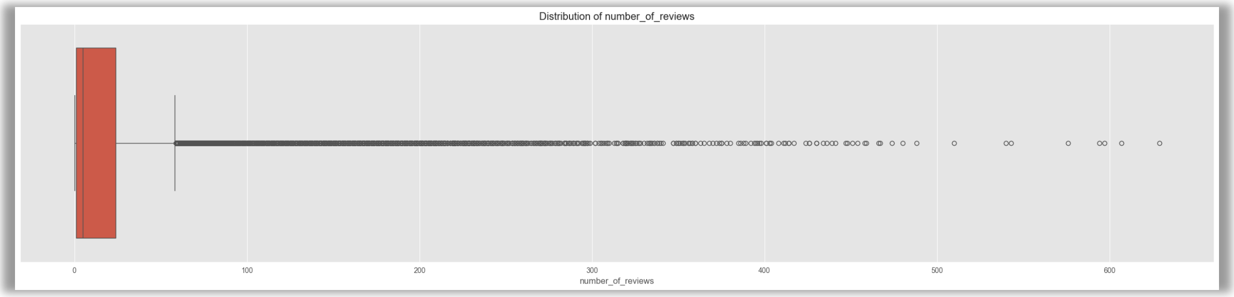
Checking for Outliers in all continuous variables (numerical)

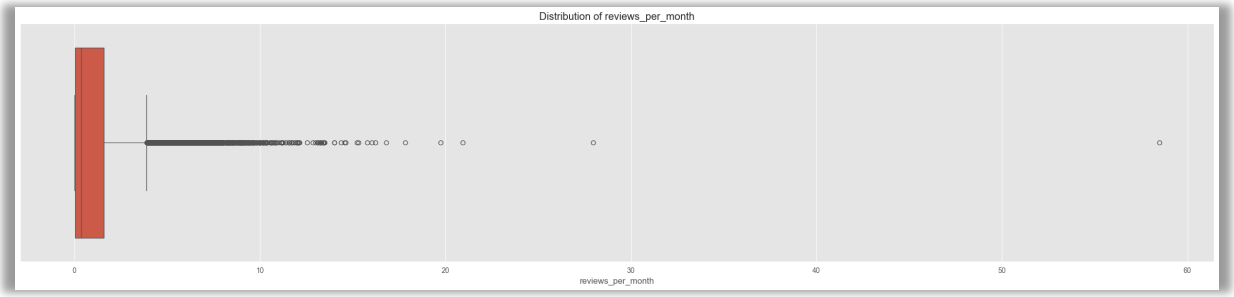
```
[52]  continuous_variables = ['price', 'minimum_nights', 'number_of_reviews', 'reviews_per_month', 'calculated_host_listings_count', 'availability_365']

for variable in continuous_variables:
    plt.style.use('ggplot')
    plt.figure(figsize=(30, 6))
    plt.title('Distribution of ' + variable)
    sns.boxplot(data=data[variable], orient='h')
plt.show()
```

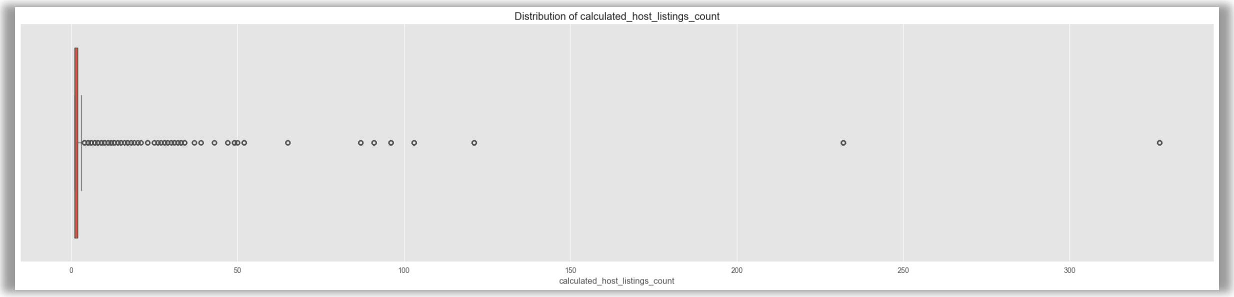
Executed at 2024.04.17 15:01:02 in 2s 128ms

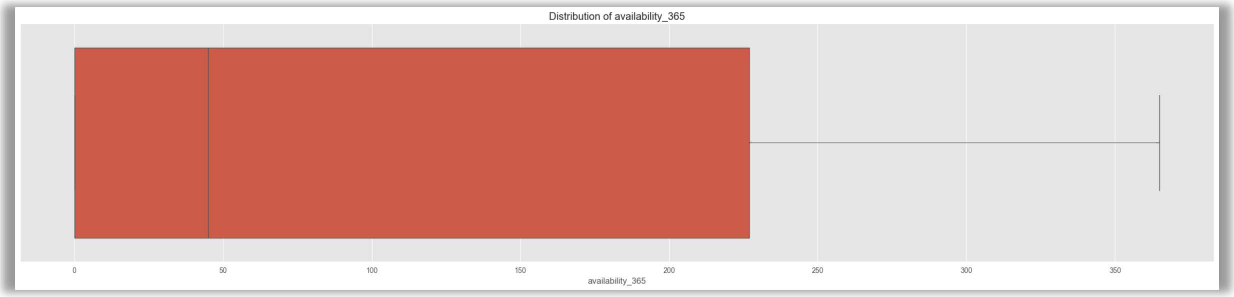

Distribution of price
0
2000
4000
6000
8000
10000
price


Distribution of minimum_nights
0
200
400
600
800
1000
1200
minimum_nights


Distribution of number_of_reviews
0
100
200
300
400
500
600
number_of_reviews


Distribution of reviews_per_month
0
10
20
30
40
50
60
reviews_per_month


Distribution of calculated_host_listings_count
0
50
100
150
200
250
300
calculated_host_listings_count

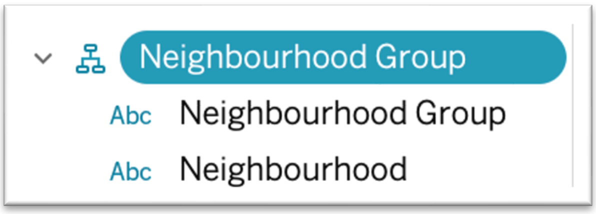

Distribution of availability_365
0
50
100
150
200
250
300
350
availability_365

Data Wrangling

- First, we checked the shape (columns and rows present) and data types of columns in the dataset.
- Then we checked for null values in the dataset. Columns: name, host_name, last review and reviews_per_month had null values.
- After that, we checked for duplicate rows in the dataset and no duplicate data was found.
- Lastly, we identified and reviewed outliers. All columns except for availability_365 had outliers. For now, we are not removing the outliers since it's a part of the population that we are studying.
- The data was then loaded to Tableau.

Data Analysis and Visualization using Tableau

- **Neighbourhood Distribution**
    - Created Hierarchy between Neighbourhood Groups and Neighbourhood



    - Visualised the distribution using Packed bubbles, Pie Charts, Treemaps and Symbol maps

- **Room Type Distribution**
    - Visualised the distribution using Pie Charts, and side-by-side bars across each neighbourhood group.

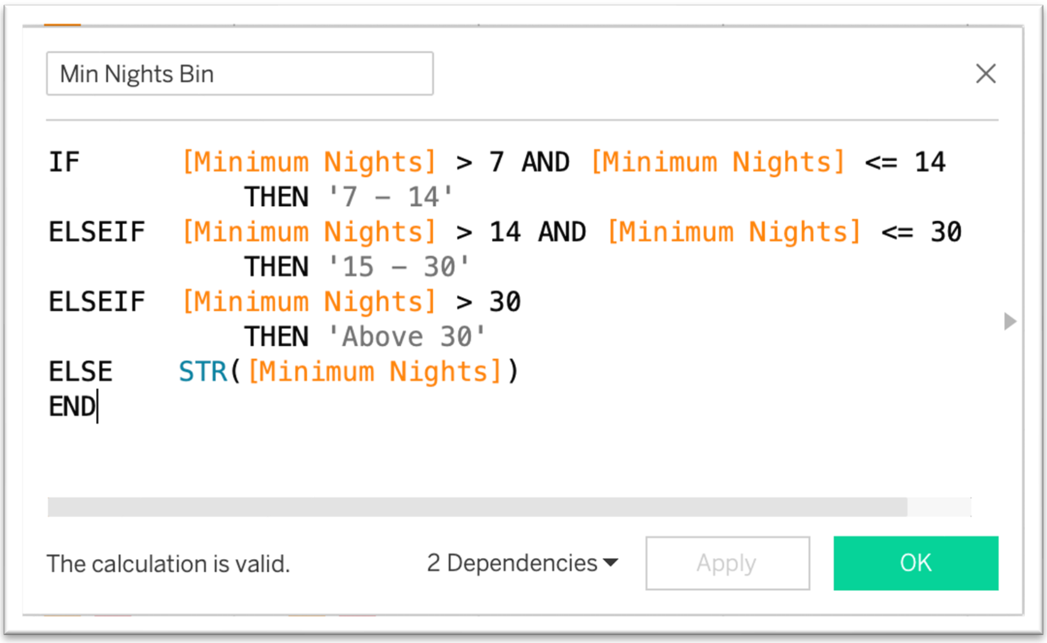
- **Price Analysis**
    - Checked average prices of properties across neighbourhood groups.

- **Bookings (Listings) across the neighbourhood**
    - Checked the count of properties for each neighbourhood bar chart.

- **Review analysis**
    - Checked for the relationship between Total reviews and the average price of the property in each neighbourhood using side-by-side bars.
    - Checked for the relationship between Total reviews per month and the average price of the property in each neighbourhood using side-by-side bars.

- Checked for the relationship between Total reviews and the average minimum nights required in each neighbourhood using side-by-side bars.

- **Minimum Nights**
  - Created bins for Minimum nights in the range of 1, 2, 3, 4, 5, 6, 7, 7-14, 15-30 and Above 30 as shown below,



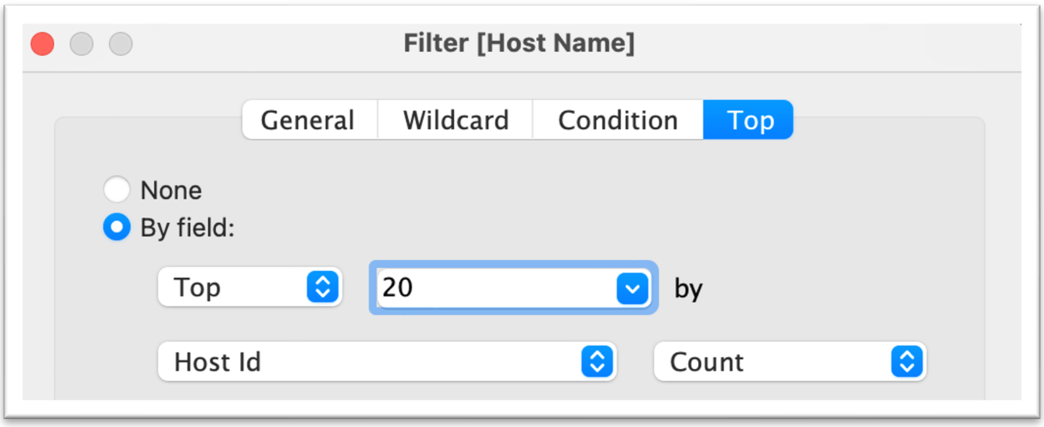
Min Nights Bin

```
IF      [Minimum Nights] > 7 AND [Minimum Nights] <= 14
            THEN '7 - 14'
ELSEIF  [Minimum Nights] > 14 AND [Minimum Nights] <= 30
            THEN '15 - 30'
ELSEIF  [Minimum Nights] > 30
            THEN 'Above 30'
ELSE    STR([Minimum Nights])
END
```

The calculation is valid.

  - Visualised the bookings/listings for each bin across each neighbourhood group using side-by-side charts.

- **Top 20 hosts**
  - Filtered top 20 hosts by count across all neighbourhoods as shown below,



Filter [Host Name]

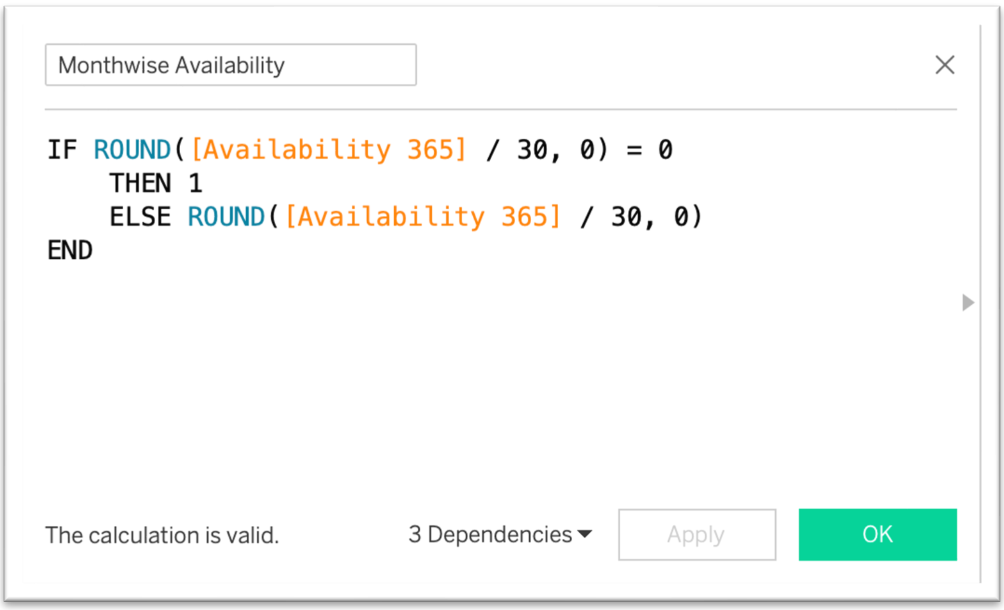
None
[✓ By field:]

Top 20 by

Host Id Count

  - Visualised the data using Treemaps.
  - Added filter for Neighbourhood group to get Top 20 hosts of each neighbourhood.

- **Availability 365 Analysis**
    - Created bins to segregate the availability into the number of months, as shown below,



Monthwise Availability

```
IF ROUND([Availability 365] / 30, 0) = 0
    THEN 1
    ELSE ROUND([Availability 365] / 30, 0)
END
```

The calculation is valid.

    - Converted the attribute to a continuous one.
    - Created an area chart to visualise the availability across each neighbourhood group.
    - Created an area chart to visualise the availability across each room type
    - Created a dual combination chart to visualise the number of properties available across the number of months.