Airbnb Case Study – Methodology Overview

Methodology Summary

The analysis for this case study was conducted using **Jupyter Notebook** for data preprocessing and **Tableau** for visualization and data analysis. The dataset used was **AB_NYC_2019.csv**, which contains **48,895 rows** and **16 columns**.

Step 1: Data Cleaning and Preparation

Preprocessing in Jupyter Notebook

- **Columns Removed:** `Id`, `Name`, `Last Review` (as they provided minimal value to the analysis).

```
# Import the necessary libraries
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
# Reading Data from file
air = pd.read_csv("AB_NYC_2019.csv")
air.head(5)
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	9	19-1
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	45	21-0
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	0	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	270	05-0
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	9	19-1

```
availability_365                  0
dtype: int64
```

```
# Now we have the missing values, there are certain columns that are not efficient to the dataset
air.drop(['id','name','last_review'], axis = 1, inplace = True)
```

- **Duplicate Data Check**: No duplicate rows were found in the dataset.

- **Handling Missing Values**:
    - Columns such as `name`, `host-name`, `last review`, and `review-per-month` contained missing values.

```
# Checking for missing values
air.isnull().sum()
```

```
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
dtype: int64
```

    - The `name` column was dropped since the number of missing values was negligible, making its removal insignificant to the analysis.

```
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
dtype: int64
```

```
# Now we have the missing values, there are certain columns that are not efficient to the dataset
air.drop(['id','name','last_review'], axis = 1, inplace = True)
```

- **Formatting and Outlier Identification**: The dataset was checked for inconsistencies and outliers.

```
air.reviews_per_month.isnull().sum()
```

```
10052
```

```
# Now reviews per month contains more missing values which should be replaced with 0 respectively
air.fillna({'reviews_per_month':0},inplace=True)
```

```
air.reviews_per_month.isnull().sum()
```

```
0
```

```
# There are no missing values present in reviews_per_month column
# Now to check the unique values of other columns'
air.room_type.unique()
```

```
array(['Private room', 'Entire home/apt', 'Shared room'], dtype=object)
```

```
len(air.room_type.unique())
```

```
3
```

```
air.neighbourhood_group.unique()
```

```
array(['Brooklyn', 'Manhattan', 'Queens', 'Staten Island', 'Bronx'],
      dtype=object)
```

```
len(air.neighbourhood_group.unique())
```

```
5
```

```
len(air.neighbourhood.unique())
```

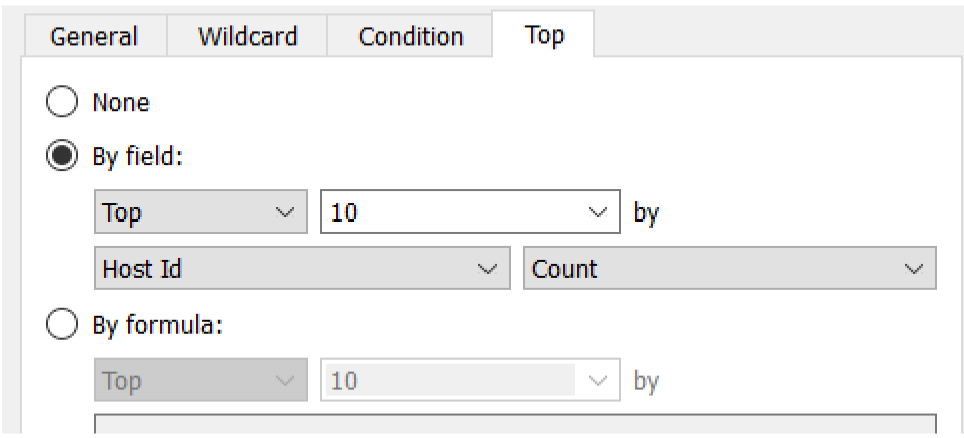
```
221
```

Step 2: Data Analysis & Visualization Using Tableau

Key Insights and Visualizations

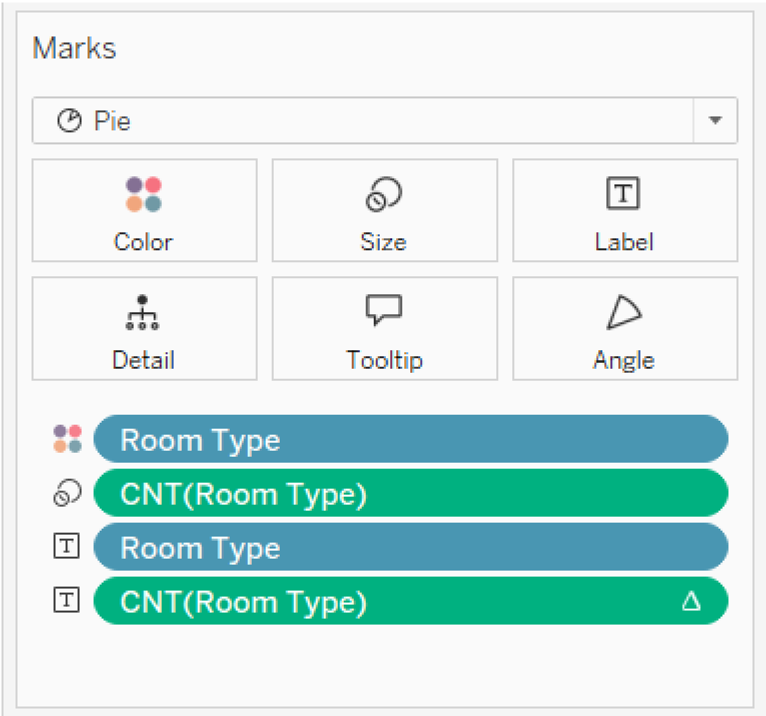
1. Top 10 Hosts Analysis

- A **Tree Map** was created to visualize the **Top 10 Hosts** by **Host ID count**.



2. Room Type Preferences by Neighborhood Group

- A **Pie Chart** was generated to show the **percentage distribution** of room types across different **neighborhood groups**.
- The **Room Type** attribute was assigned different colors to distinguish each type, and **Host ID count** was used for size representation.

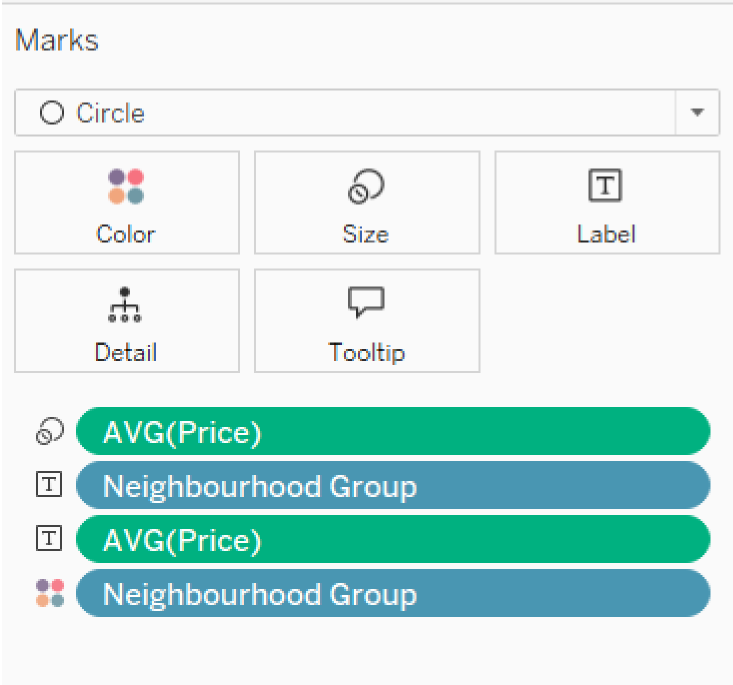


3. Price Variance by Neighborhood Group

- A **Box-and-Whisker Plot** was used, placing **Neighborhood Groups** on the x-axis and **Price** on the y-axis.
- Instead of using the **sum of prices**, the **median price** was calculated for better representation.

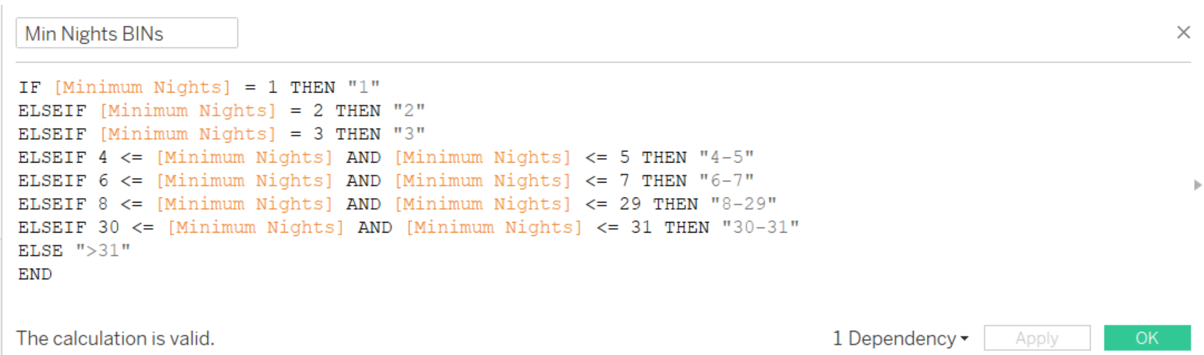
4. Average Price by Neighborhood Group

- A **Bubble Chart** was created with **Neighborhood Groups** as categories and **Price** as the numeric variable.
- The **Average Price** was displayed using labels, and different colors were assigned to each **Neighborhood Group**.



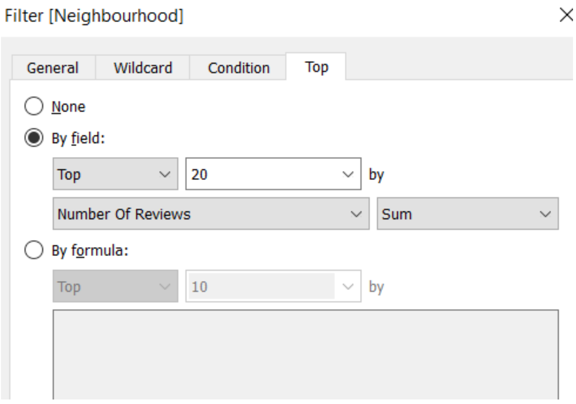
5. Customer Booking Trends by Minimum Nights

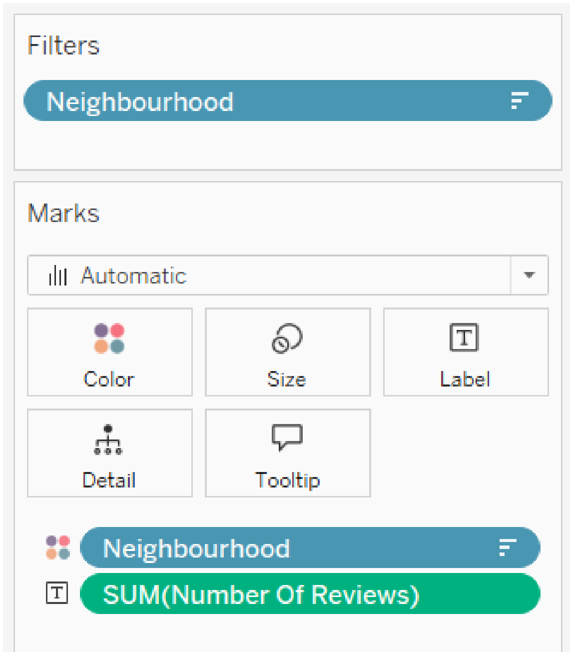
- **Bins were created** for the `Minimum Nights` column to visualize the **distribution of bookings** based on the duration of stays across different neighborhoods.



6. Most Popular Neighborhoods

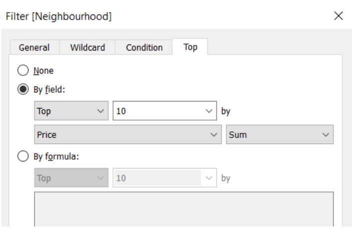
- A **Bar Chart** was developed using **Neighborhood names** on the y-axis and **Total Review Count** on the x-axis.
- The **Top 20 neighborhoods** were filtered based on the highest number of reviews.

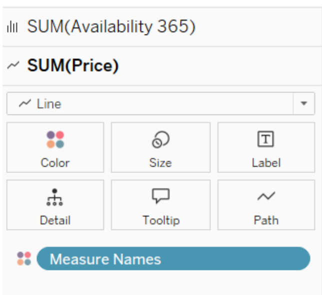




7. Neighborhood vs. Availability

- A **Dual-Axis Chart** was created:
    - A **Bar Chart** represented the **Availability (365 days)** for top neighborhoods.
    - A **Line Chart** overlaid the **Price variation** for the **top 10 neighborhoods** sorted by price.

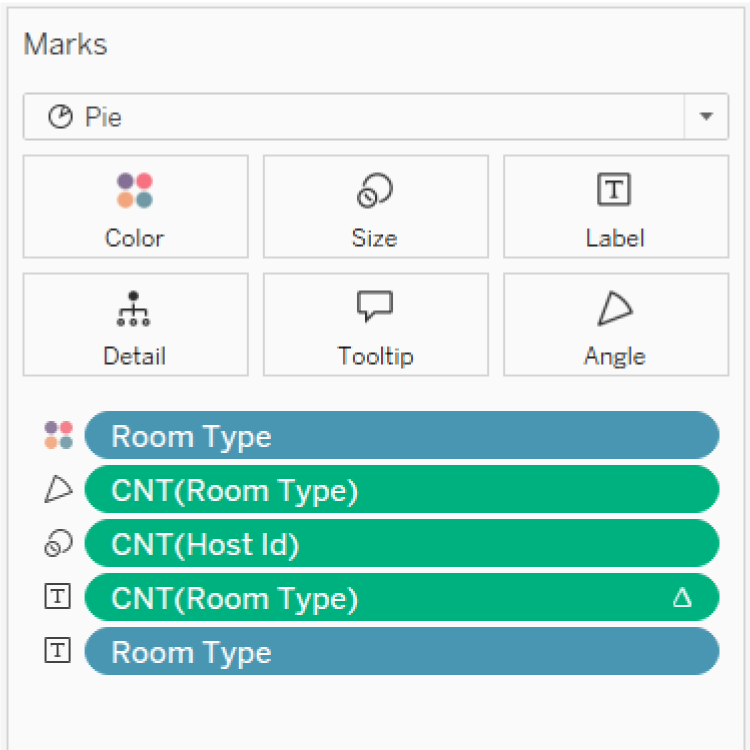




Step 3: Additional Visualizations (Methodology PPT 2)

1. Room Type Preferences by Neighborhood Group (Revisited)

- The **Pie Chart** from the previous analysis was replicated for cross-validation.



2. Customer Booking Trends by Minimum Nights (Revisited)

- The **binning approach** was re-examined to further refine **booking distribution trends** across **neighborhoods**.

3. Neighborhood vs. Availability (Revisited)

- The **dual-axis chart** was revisited for further insights.

4. Price Range Preferences by Customers

- A **Bar Chart** was created to analyze customer pricing preferences.

- **Bins** were generated for the `Price` column at **$20 intervals** to understand how price influences booking volume.

Price Bins

```
IF [Price] >= 0 AND [Price] < 20 THEN "$0 - $19"
ELSEIF [Price] >= 20 AND [Price] < 40 THEN "$20 - $39"
ELSEIF [Price] >= 40 AND [Price] < 60 THEN "$40 - $59"
ELSEIF [Price] >= 60 AND [Price] < 80 THEN "$60 - $79"
ELSEIF [Price] >= 80 AND [Price] < 100 THEN "$80 - $99"
ELSEIF [Price] >= 100 AND [Price] < 120 THEN "$100 - $119"
ELSEIF [Price] >= 120 AND [Price] < 140 THEN "$120 - $139"
ELSEIF [Price] >= 140 AND [Price] < 160 THEN "$140 - $159"
ELSEIF [Price] >= 160 AND [Price] < 180 THEN "$160 - $179"
ELSEIF [Price] >= 180 AND [Price] < 200 THEN "$180 - $199"
```

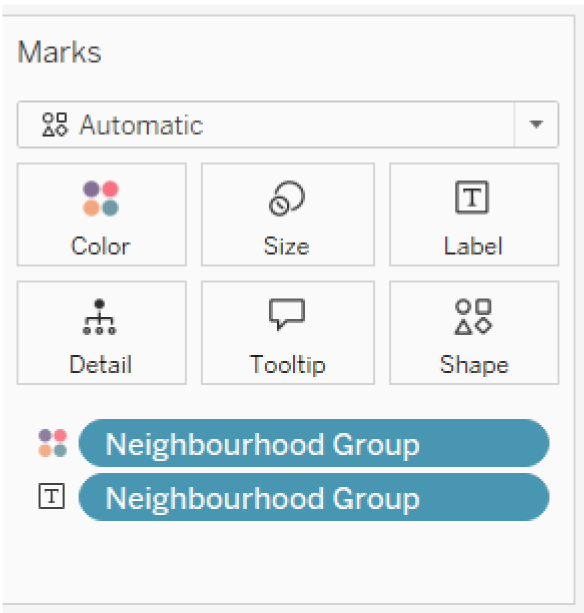
The calculation is valid. 1 Dependency Apply OK

5. Price Variation by Room Type & Neighborhood

- A **Heatmap** was created using a **Highlights Table**:
  - `Room Type` on the y-axis.
  - `Neighborhood Group` on the x-axis.
  - **Average price** was color-coded to reveal variations.

6. Price Variation by Geography

- A **Geo-Location Map** was used to plot neighborhoods, allowing for a **geographical representation of price differences** across different areas.



7. Most Popular Neighborhoods (Revisited)

- The **Bar Chart with review counts** was revisited to verify the **Top 20 most-reviewed neighborhoods**.

Conclusion

This study utilized **Jupyter Notebook** for initial data processing and **Tableau** for advanced analysis and visualization. Key insights included identifying **top hosts, room type preferences, price variances, popular neighborhoods, and customer booking behaviors**. The **dual-axis charts, pie charts, heatmaps, and geo-location maps** provided an in-depth understanding of the dataset.