

Airbnb Case study

Ppt 2

Objective

- ▶ Airbnb is an online platform using which people can rent their unused accommodations.
- ▶ During the covid time, Airbnb incurred a huge loss in revenue.
- ▶ People have now started travelling again and Airbnb is aiming to bring up the business again and e ready to provide services to customers.

Background

- ▶ For the past few months, Airbnb has seen a major decline in revenue.
- ▶ Now that the restrictions have started lifting and people have started to travel more, Airbnb wants to make sure that it is fully prepared for this change.
- ▶ So, analysis has been done on a dataset consisting of various Airbnb listings in New York.

Appendix- Data sources

Column	Description
id	listing ID
name	name of the listing
host_id	host ID
host_name	name of the host
neighbourhood_group	location
neighbourhood	area
latitude	latitude coordinates
longitude	longitude coordinates
room_type	listing space type
price	
minimum_nights	amount of nights minimum
number_of_reviews	number of reviews
last_review	latest review
reviews_per_month	number of reviews per month
calculated_host_listings_count	amount of listing per host
availability_365	number of days when listing is available for booking

Data methodology

- ▶ Conducted a thorough analysis of Airbnb Dataset
- ▶ Cleaned the dataset using python
- ▶ Dropped insignificant columns.
- ▶ Identified outliers
- ▶ Created charts and visualizations

Appendix - data assumptions

Categorical Variables:

- room_type
- neighbourhood_group
- neighbourhood

Continuous Variables(Numerical):

- Price
- minimum_nights
- number_of_reviews
- reviews_per_month
- calculated_host_listings_count
- availability_365
- Continuous Variables could be binned in to groups too

Location Variables:

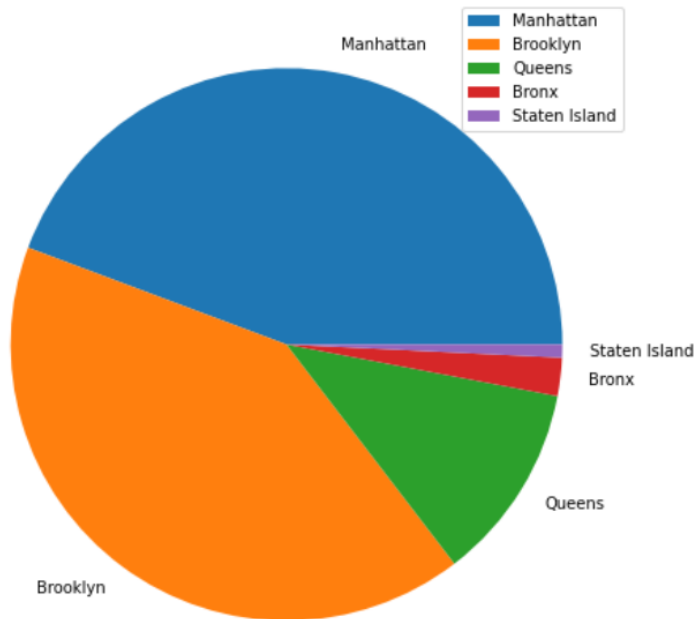
- latitude
- longitude

Time Variable:

- last_review

What are the neighbourhoods they need to target?

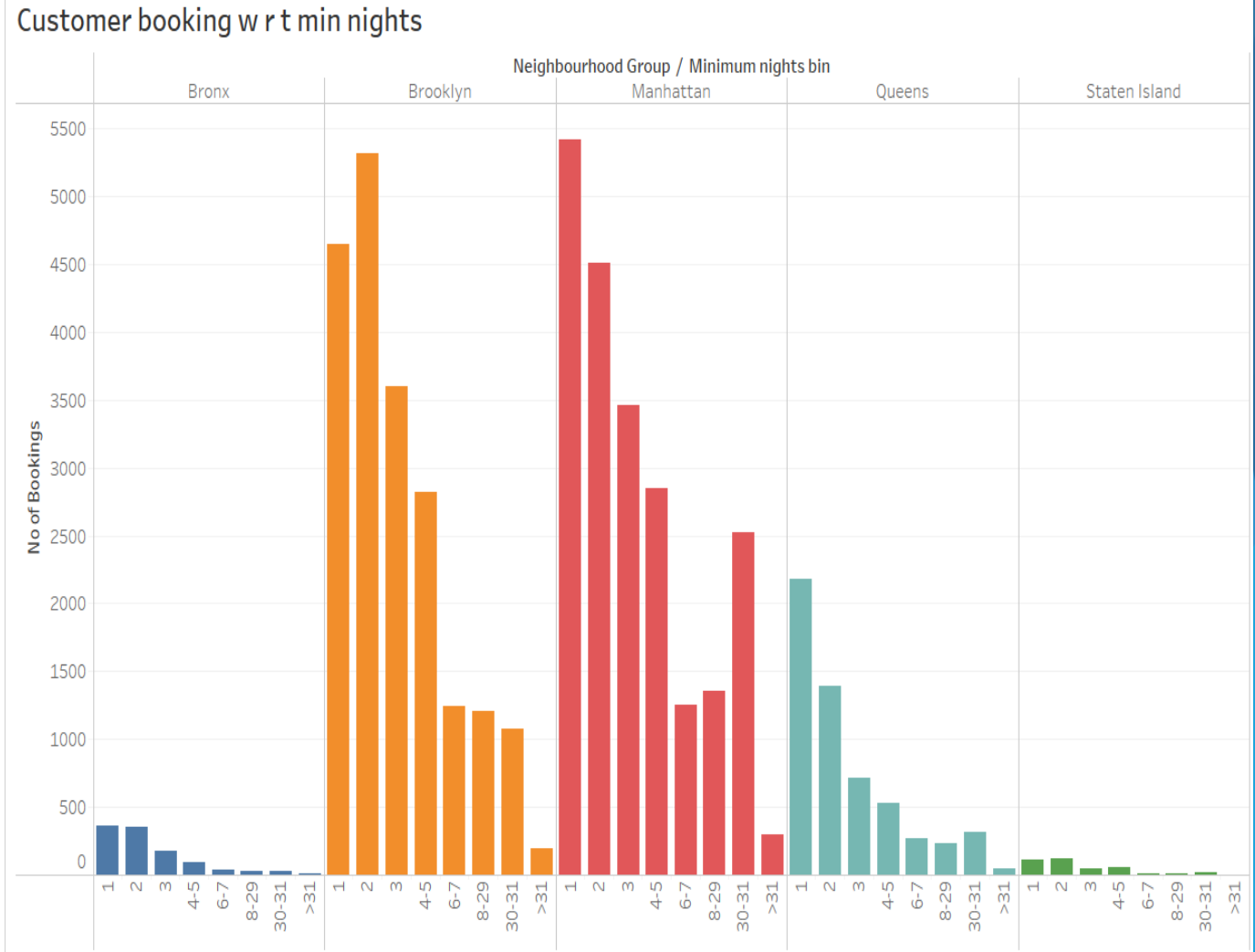
81 % of the listing are Manhattan and Brooklyn
neighbourhood_group



```
Manhattan    21661
Brooklyn     20104
Queens       5666
Bronx        1091
Staten Island 373
Name: neighbourhood_group, dtype: int64
```

Customer Booking with respect to minimum nights

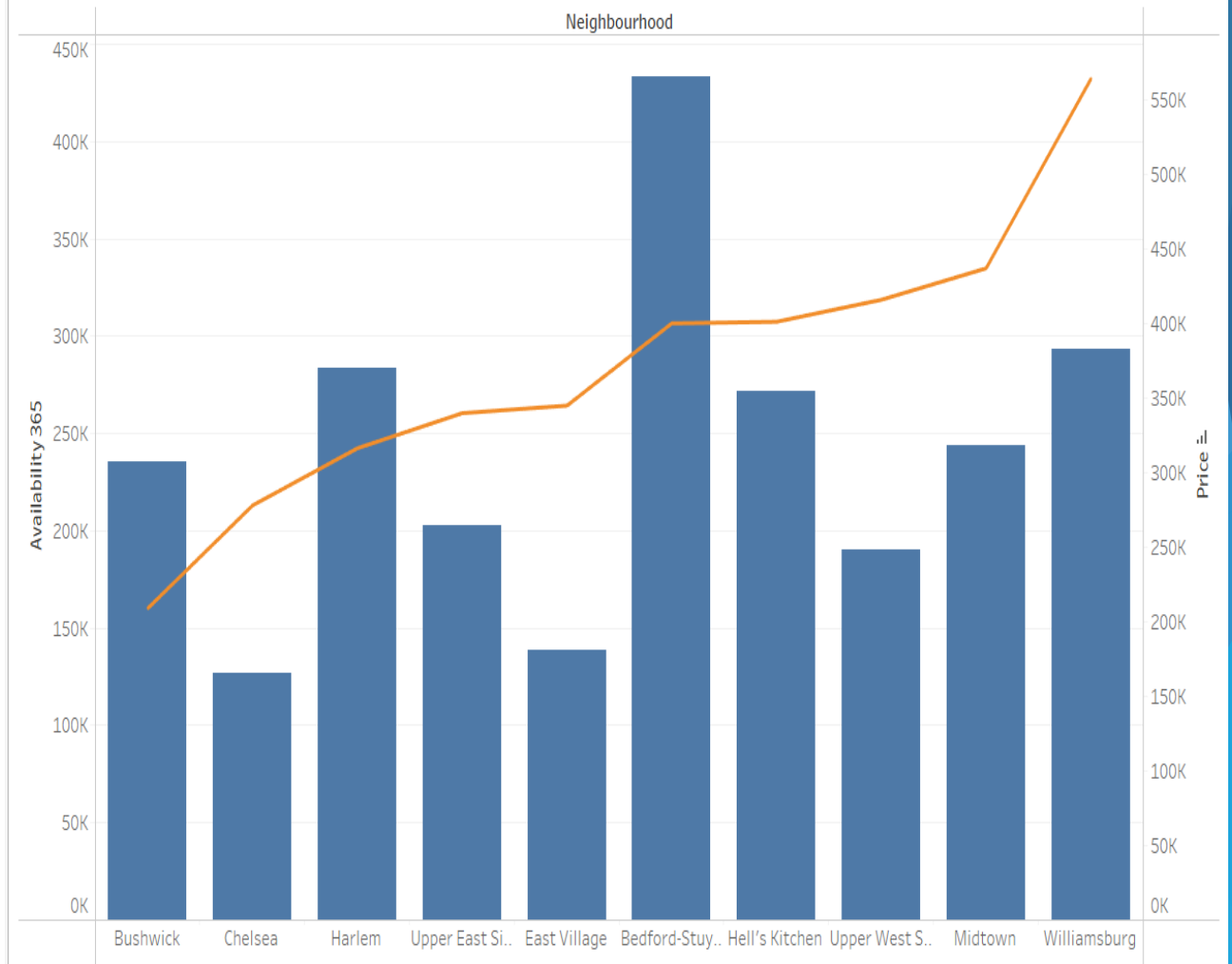
- The listings with Minimum nights 1-5 have the most number of bookings. We can see a prominent spike in 30 days, this would be because customers would rent out on a monthly basis.
- After 30 days, we can also see small spikes, this can also be explained by the monthly rent taking trend.
- Manhattan & Queens have higher number of 30 day bookings compared to the others. The reason could be either tourists booking long stays or mid-level employees who opt for budget bookings due company visits



Neighbourhood vs Availability

- Availability of Bedford is highest and its price is on the lower side. It is a good choice for customers.
- After Bedford, Harlem follows the same trend.
- Chelsea's availability low but it is costly.
- On the other hand, Williams's price is high and has average availability.

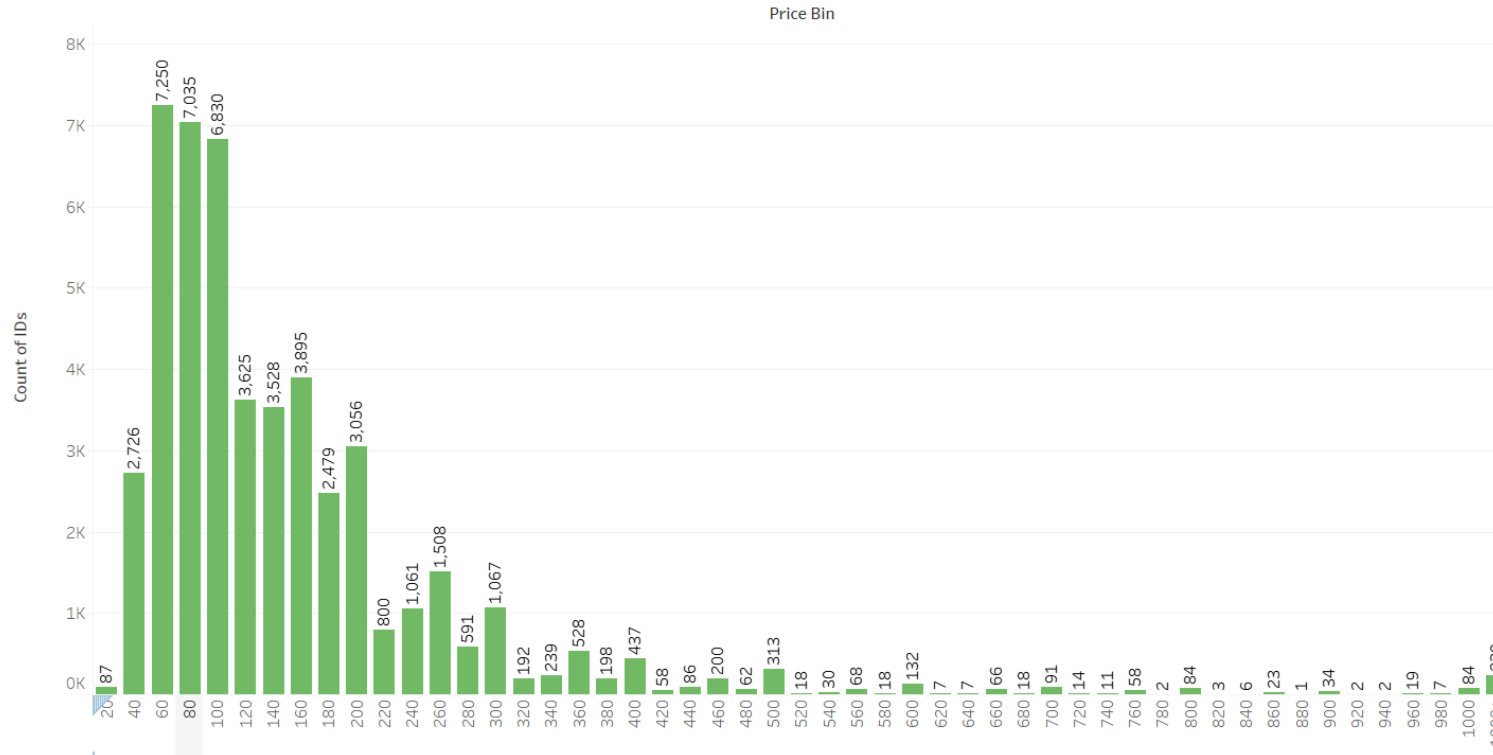
Neighbourhood vs Availability



• Price range preferred by Customers

We have taken pricing preference based on volume of bookings done in a price range. From both the graphs, the favourable price range is \$60 - \$200. This is the price range most preferred by most customers.

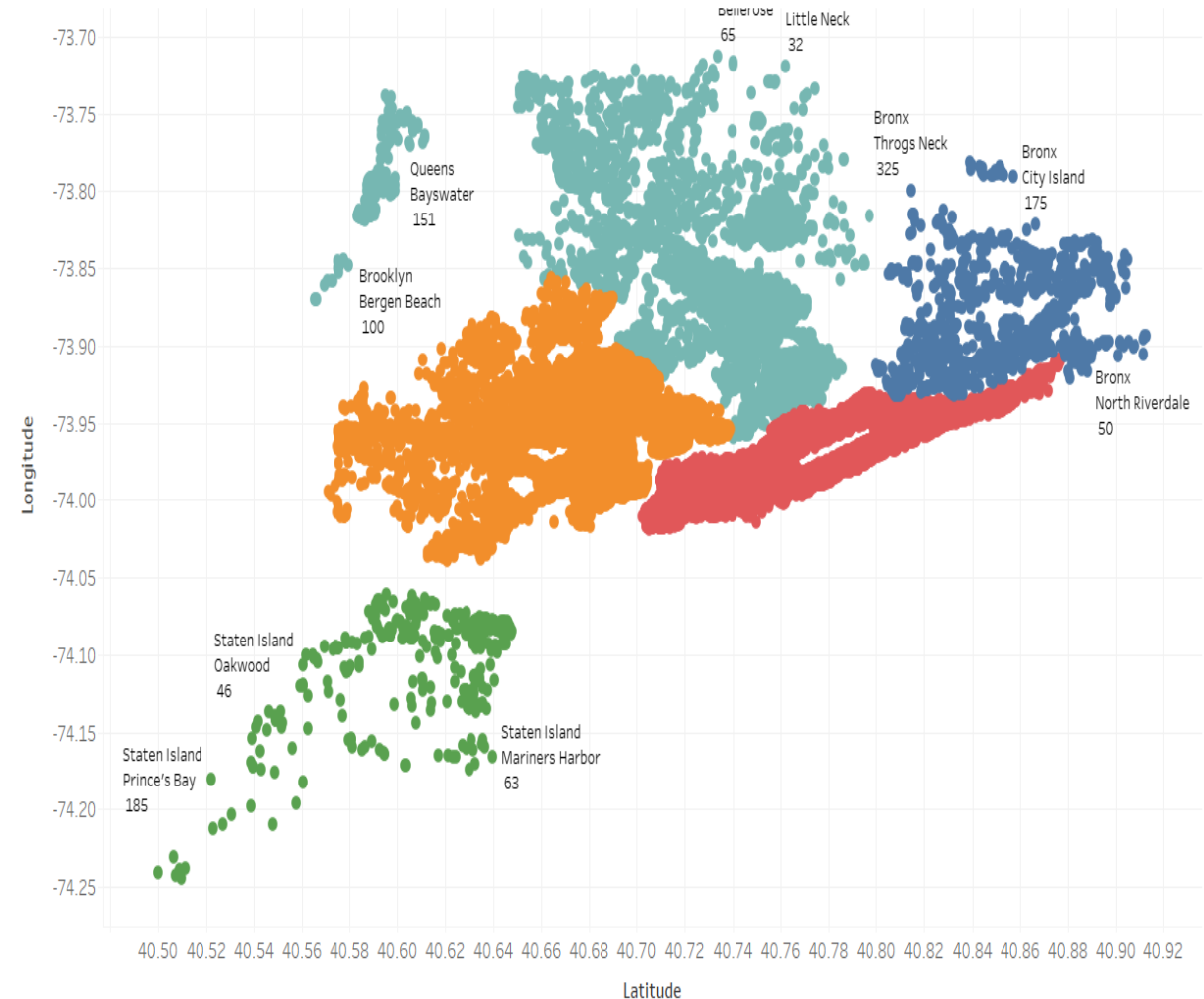
Price range preferred by customers



Price variation with respect to Geography

- We see that, Airbnb has good presence in Manhattan, Brooklyn & Queens.
- Listings are maximum in Manhattan & Brooklyn owing to the high population density and it being the financial and tourism hub of NYC. Staten Island has the least number of listings, due to its low population density and very few tourism destinations.

Price variation w.r.t Geography



Popular Neighborhoods

- We see that Bedford-Stuyvesant from Brooklyn is the highest popular with 1,10,352 no of reviews in total followed by Williamsburg.
- Harlem from Manhattan got the highest no of reviews followed by Hell's kitchen.
- The higher number of customer reviews imply higher satisfaction in these localities.

Popular Neighbourhoods

