



# New York - 2019

Understanding the past and planning the Future !!

By - Aman Nehra



# Problem Areas...



## Clarity on Data.

Details of the Data, background & clean Data is not provided.



## Problem statement.

It is very important to know what exactly the problem statement.



## Presenting Data.

Deciding on what type of information is to be added.



# Solution

## EDA & Domain Knowledge:

Following Exploratory Data analysis and knowledge of the business model can help solve these problems.



**Clean the Data.**

**Look for insights.**

**Present as per audience.**





# Clean the Data!!

```
1 bnb = bnb.drop(['latitude', 'longitude', 'host_name'], axis = 1)
2 bnb.head()
```

Drop Unwanted Columns.

```
1 ## Dropping 'Name' null record as they comprise of only 0.03% of data.
2 bnb1 = bnb[~bnb.name.isnull()].copy()
3 bnb1.shape
```

Drop Blank Rows.

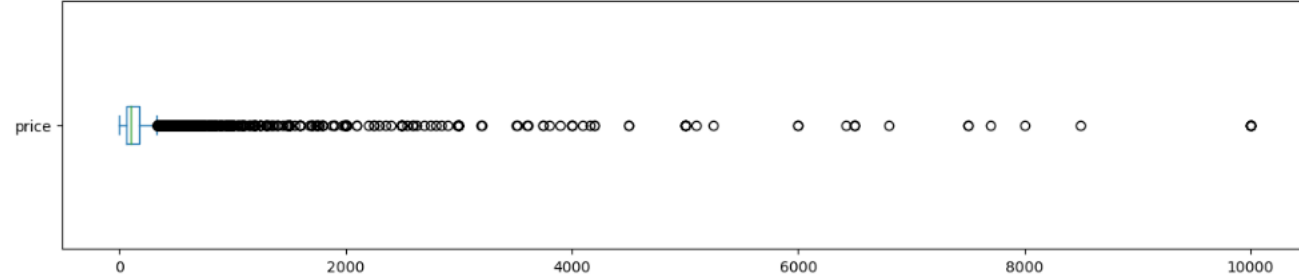
```
1 ## Filling '0' in place of 'null' in 'newviews_per_month'
2 bnb1['reviews_per_month'] = bnb1['reviews_per_month'].replace(np.nan, 0)
3 bnb1['reviews_per_month'].isnull().sum()
```

Filling null values.



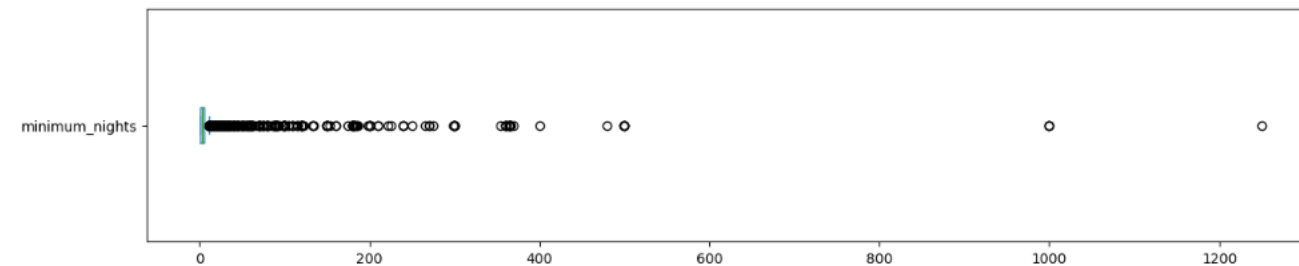
# Addressing Outliers.

Checking for outliers in Price and minimum\_nights. Removing the outlier values from the data set.



Values over \$500.00:  
1044  
2.1359%

Values equal to \$0:  
11  
0.0225%



Values over 30 nights:  
746  
1.5262%



# Understanding Data.

Understanding the data and figuring out what aspects can be compared and what are important to the decision makers.

```
1 ##Let's find number of unique values in each Columns
2 bnb1.nunique(axis = 0)
```

id	47110
name	46157
host_id	36245
neighbourhood_group	5
neighbourhood	219
room_type	3
price	454
minimum_nights	30
number_of_reviews	393
reviews_per_month	938
calculated_host_listings_count	47
availability_365	366
dtype:	int64

```
1 bnb1.room_type.unique()
array(['Private room', 'Entire home/apt', 'Shared room'], dtype=object)

1 bnb1.neighbourhood_group.unique()
array(['Brooklyn', 'Manhattan', 'Queens', 'Staten Island', 'Bronx'],
      dtype=object)
```

# Comparing vectors.

- Comparing Average price of stay of different neighbourhood's.
- Comparing Average price of stay in different room types in different neighbourhood's.

```

1 # Lets now compare the price of diffrent neighborhood.
2 df = bnb1[['neighbourhood_group', 'price']]
3 df = bnb1.groupby(['neighbourhood_group'], as_index=False)[['price']].mean()
4
5 plt.figure(figsize=(12, 6))
6 df = sns.barplot(x="neighbourhood_group", y="price", data=df)
7 for p in df.patches:
8     df.annotate(format(p.get_height(), '.2f'),
9                 (p.get_x() + p.get_width() / 2., p.get_height()),
10                ha = 'center', va = 'center', size = 12,
11                xytext = (0, -12), textcoords = 'offset points')
12
13 plt.xlabel('Neighbourhood's')
14 plt.ylabel('Average Price')
15 plt.title('Average Price by Neighbourhood')
16 plt.show()
17

```

```

1 # Checking price of diifrent room type at diffrent Neighbourhood_Groups
2
3 df3 = bnb1[['neighbourhood_group', 'room_type', 'price']]
4 df3 = df3.groupby(['neighbourhood_group', 'room_type'], as_index=False)[['price']].mean()
5
6 plt.figure(figsize=(12, 6))
7 sns.set_theme(style='white')
8 df = sns.barplot(x="neighbourhood_group", y="price", data=df3, hue='room_type')
9 for p in df.patches:
10     df.annotate(format(p.get_height(), '.2f'),
11                 (p.get_x() + p.get_width() / 2., p.get_height()),
12                ha = 'center', va = 'center', size = 11,
13                xytext = (0, -12), textcoords = 'offset points')
14
15 plt.xlabel('Neighbourhood_Group')
16 plt.ylabel('Average Price')
17 plt.title('Average Price by Room Type')
18 plt.show()

```



# Selection of Problem.

```
1 df6 = df5[['neighbourhood_group', 'availability_365']].copy()
2 df6 = df5.groupby(['neighbourhood_group'], as_index=False)[['availability_365']].count()
3
4 plt.figure(figsize=(12, 6))
5 df = sns.barplot(x="neighbourhood_group", y="availability_365", data=df6)
6 for p in df.patches:
7     df.annotate(format(p.get_height(), '.2f'),
8                 (p.get_x() + p.get_width() / 2., p.get_height()),
9                 ha = 'center', va = 'center', size = 12,
10                xytext = (0, -12), textcoords = 'offset points')
11
12 plt.xlabel('Neighbourhood_Group')
13 plt.ylabel('Number of Outlets')
14 plt.title('Number of Zero Days Available Units')
15 plt.show()
```

We see that a major number of units are not available for even a single day throughout the year.

This shows that they are registered but not renting out the space available or may be renting out on different platform.

```
1 # Share of Zero days available Airbnb in each Neighbourhood_Group
2 #assuming 46175(count of unique name) as total number of listings in Airbnb
3 #data for 0 night Airbnb
4 bronx = 177/(46175*.02)*100
5 brooklyn = 7733/(46175*.41)*100
6 manhattan = 7870/(46175*.44)*100
7 queens = 1353/(46175*.12)*100
8 staten = 40/(46175*.01)*100
9
10 print('Following is the Percentage of listings which are showing available for 0 Nights')
11 print('Bronx = ', bronx)
12 print('Brooklyn = ', brooklyn)
13 print('Manhattan = ', manhattan)
14 print('Queens = ', queens)
15 print('Staten Island = ', staten)
```

Following is the Percentage of listings which are showing available for 0 Nights

Bronx = 19.16621548456957

Brooklyn = 40.84672573850806

Manhattan = 38.73603386326722

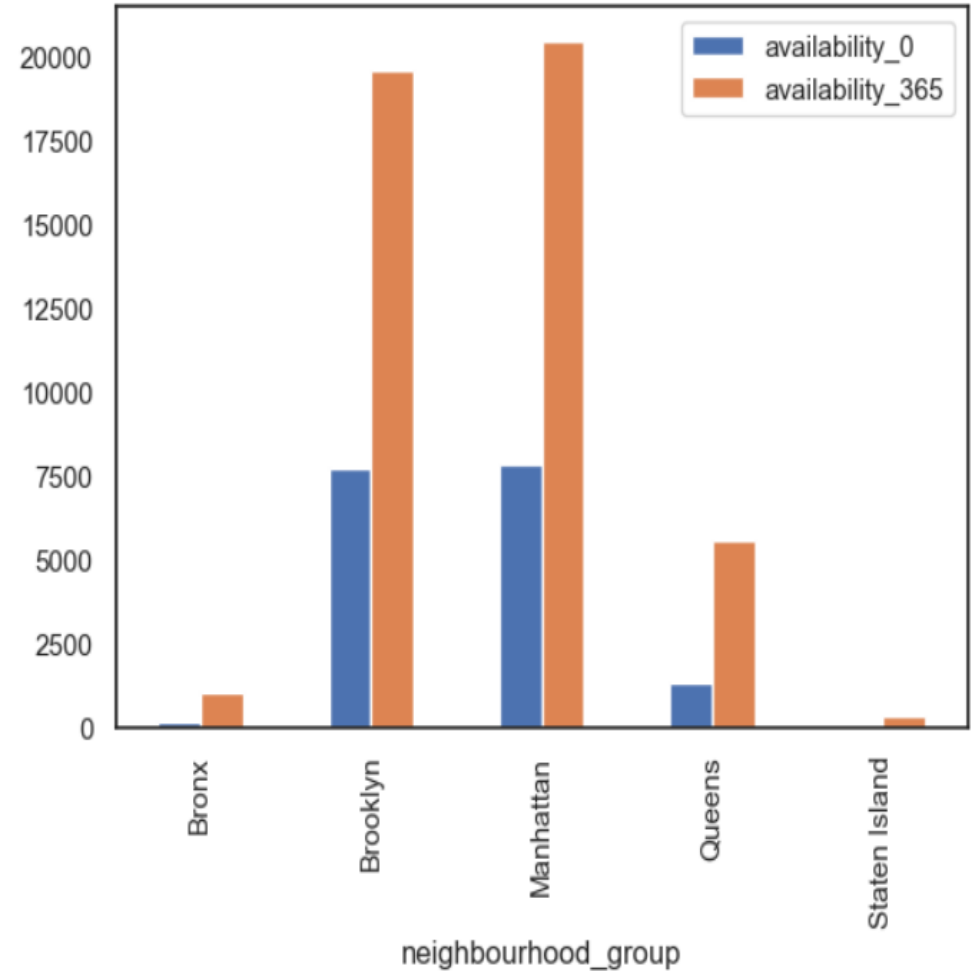
Queens = 24.41797509474824

Staten Island = 8.662696264212236



# Presenting the findings.

Simple bar charts have been used in the presentation for the ease of perception and comparison by the decision makers.





# Method Followed

- ❖ Cleaning and understanding the Data.
- ❖ Figuring out the problem.
- ❖ Presenting the problem with solution to stake holders.

**EDA**

**Understanding the Data.**

Clean the data and figure out what is given.

**Problem ?**

**Present findings & Solution.**

**Output !!**

Present the Data .

**Focus on cause of problem.**

With domain knowledge find problem.



# Questions Please ??<sup>#</sup>

# Separate PDF File is attached for the Methodology and assumptions.