Vraj Patel - 31st January, 2021.

New York City Airbnb Open Data

Visual and Analytical report on the Airbnb listings and metrics in NYC, NY, USA (2019).

Since 2008, guests and hosts have used Airbnb to expand on traveling possibilities and present more unique, personalized way of experiencing the world. This dataset describes the listing activity and metrics in NYC, NY for 2019.

This public dataset is part of Airbnb, and the original source can be found on this website. The python code for this file can be found at GitHub.

The analytics and visualizations are performed in Python programming language using packages like NumPy, Pandas, Matplotlib and Seaborn.

1. Importing necessary libraries and the dataset.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
```

```
In [2]: df = pd.read_csv('AB_NYC_2019.csv')
```

2. A look at the dataset

```
In [3]: df.head()
```

Out[3]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem

```
In [4]: df.shape
```

```
Out[4]: (48895, 16)
```

Looking at the dataset we find that the data is contained in 48,895 rows and 16 columns. df.head() function is used to visualise any pandas dataframe's first 5 rows. More rows can be visualised by

adding the specific number into the parenthesis i.e., df.head(15) {This will give the first 15 rows of any dataframe}.

3. Finding columns with null values in the data frame.

```
In [5]: df.isnull().sum()
Out[5]: 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
```

df.isnull().sum() is the function which displays the sum of total null or NaN entries in the dataframe. Here, we find that the columns 'name', 'host_name', 'last_review', and 'reviews_per_month' have null or NaN values. These null values need to be removed before we apply the DecisionTree Regressor on the dataset for predictive analytics.

4. Unique values in each column of the dataframe.

```
In [6]: df.nunique()
Out[6]: id                                  48895
        name                                47905
        host_id                             37457
        host_name                           11452
        neighbourhood_group                     5
        neighbourhood                         221
        latitude                            19048
        longitude                           14718
        room_type                               3
        price                                 674
        minimum_nights                        109
        number_of_reviews                     394
        last_review                          1764
        reviews_per_month                     937
        calculated_host_listings_count         47
        availability_365                      366
        dtype: int64
```

Analysis of the Neighbourhood_group column

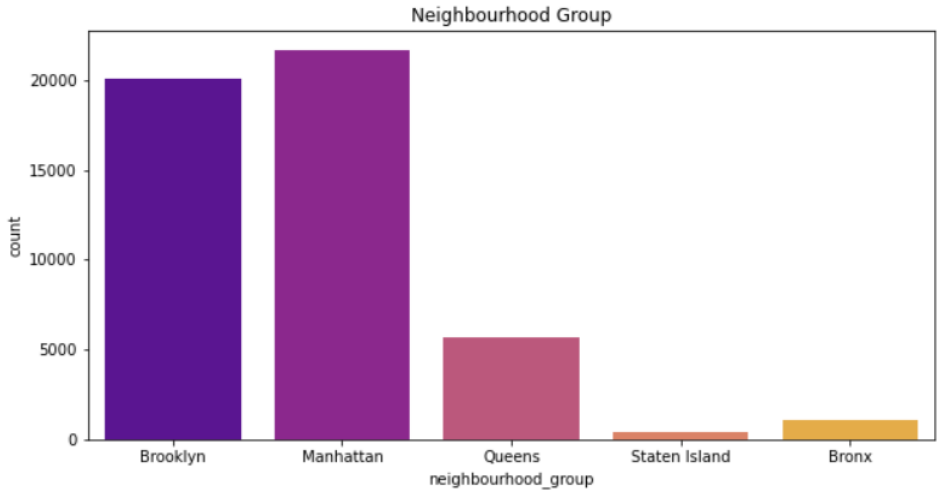
5. The number of neighbourhood groups and their count.

```
In [7]: n = df['neighbourhood_group'].value_counts()
        print(n)
```

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

```
In [9]: sns.countplot(df['neighbourhood_group'], palette="plasma")
        fig = plt.gcf()
        fig.set_size_inches(10,5)
        plt.title('Neighbourhood Group')
```

```
Out[9]: Text(0.5, 1.0, 'Neighbourhood Group')
```



As evident from the count plot, Manhattan has the highest number of Airbnb holdings in New York (21,661) followed by Brooklyn (20,104), Queens (5,666), Bronx (1,091) and Staten Island (373).

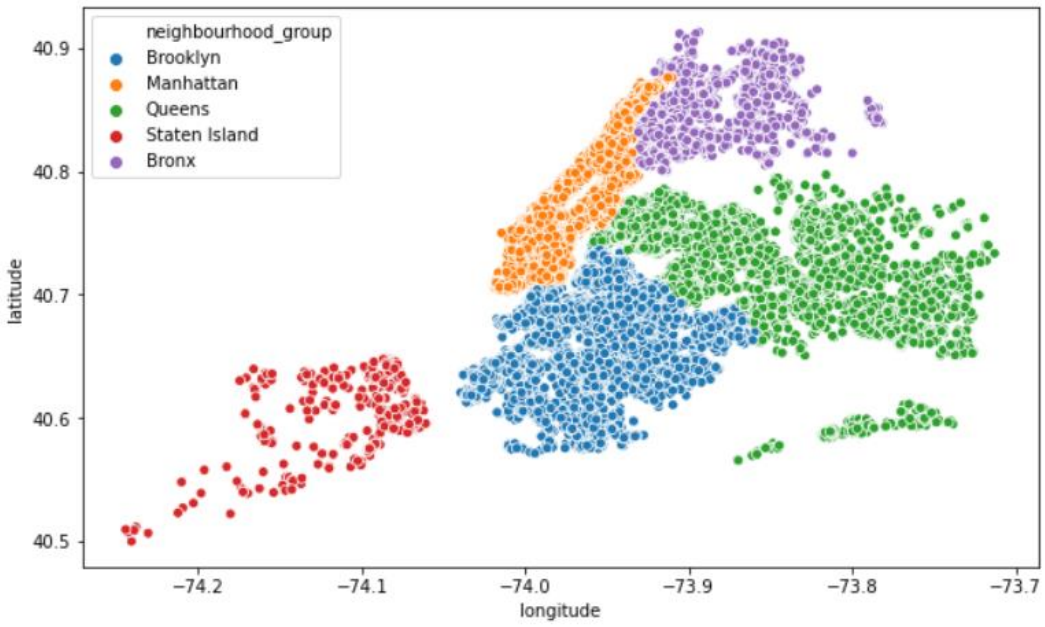
Visualising the neighbourhood groups using libraries like seaborn and matplotlib are very handy and can help in building great dashboards and reports. There are several palette's available for count plot, one should choose it wisely. Also, we get an option to resize and title the plot according to the requirements.

The dataset also provides us with the latitudes and longitudes of the place where each holding is located. This can be used to plot a scatter plot which enables us to visualise the data geospatially.

We infer from the scatter plot that Manhattan is not the largest parts of New York by area but it has the highest number of Airbnb holdings suggesting that it is a popular tourist place and has huge number of visitors round the year.

A reason for Queens having the second largest number of holdings can be its close proximity to the John F Kennedy International airport, one of the busiest airports of the world. People having long duration flight layovers or frequent flyers can be interested in such counties.

```
In [32]: plt.figure(figsize=(10,6))
         sns.scatterplot(df.longitude,df.latitude,hue=df.neighbourhood_group)
         plt.ioff()
```



Analysis of Room types in the Airbnb holdings

6. Types of rooms available all over the New York and their count.

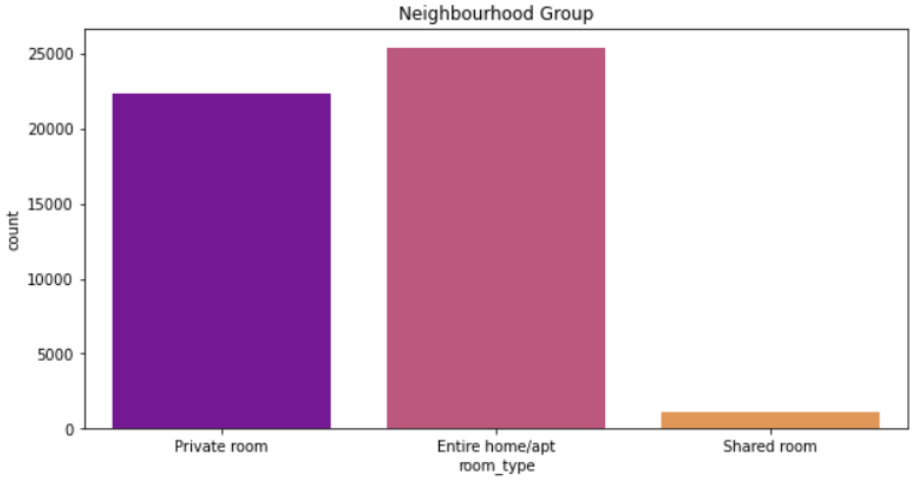
```
In [10]: df['room_type'].value_counts()

Out[10]: Entire home/apt    25409
         Private room       22326
         Shared room         1160
         Name: room_type, dtype: int64
```

The rooms are bifurcated into three types: shared room (1,106), private room (22,326), and entire home/ apartment (25,409).

We find that the hosts prefer offering the entire home and private room rather than shared spaces. This can be due to larger and private spaces can yield better incomes for the hosts. Also, managing the costs of utilities for people in shared spaces and other complexities can be a reason for hosts avoiding shared space offerings.

```
In [11]: sns.countplot(df['room_type'], palette = 'plasma')
         fig = plt.gcf()
         fig.set_size_inches(10,5)
         plt.title('Neighbourhood Group')
```

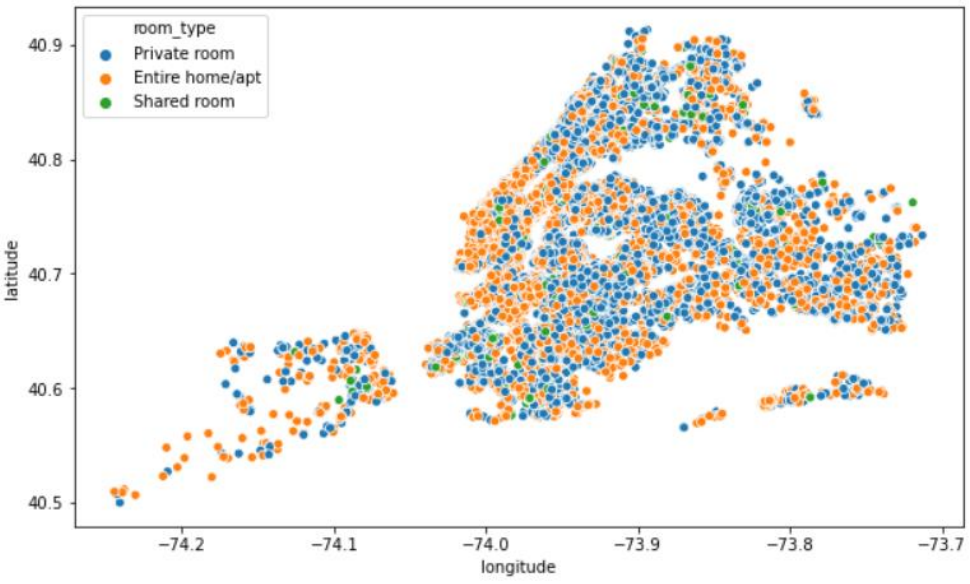
```
Out[11]: Text(0.5, 1.0, 'Neighbourhood Group')
```



7. Types of rooms in various counties.

Room type	Counties					
	Manhattan	Brooklyn	Queens	Bronx	Staten Island	**Total**
Entire house	13199	10132	3372	652	188	25409
Private room	7982	9559	2096	379	176	22326
Shared room	480	413	198	60	9	1160
Total	21661	20104	5666	1091	373	

```
In [156]: plt.figure(figsize=(10,6))
          sns.scatterplot(df.longitude,df.latitude,hue=df.room_type)
          plt.ioff()
```



Analysis of Price of renting the holdings

8. The mean price of Airbnb holdings in New York.

```
In [17]: df['price'].mean()
Out[17]: 152.7206871868289
```

The mean price is 152.72$. The mean is a measure of central tendency for the data but highly dependent on the outliers in any data. The mean is distorted due to high prices of holdings in Manhattan and Queens. The distribution of prices is depicted in the table below.

Also, eleven entries in the price column were zero. This can be due to data entry error or the holdings were used by the owner or owner's acquaintances who might have been charged. To deal with such values, these values have been replaced with the lowest price of holding i.e., 10$.

9. The price of holdings in various counties

Prices	Counties				
	Manhattan	Brooklyn	Queens	Bronx	Staten Island
Mean	196.87	124.38	99.51	87.50	114.81
Maximum	10000	10000	10000	2500	5000
Minimum	10	10	10	10	13

Average mean price = 152.72.

Analysis of neighbourhood of holdings.

10. Distinct neighbourhoods in the dataset.

```
In [24]: df['neighbourhood'].nunique()
Out[24]: 221
```

There are a total of 221 distinct neighbourhoods in the dataset.

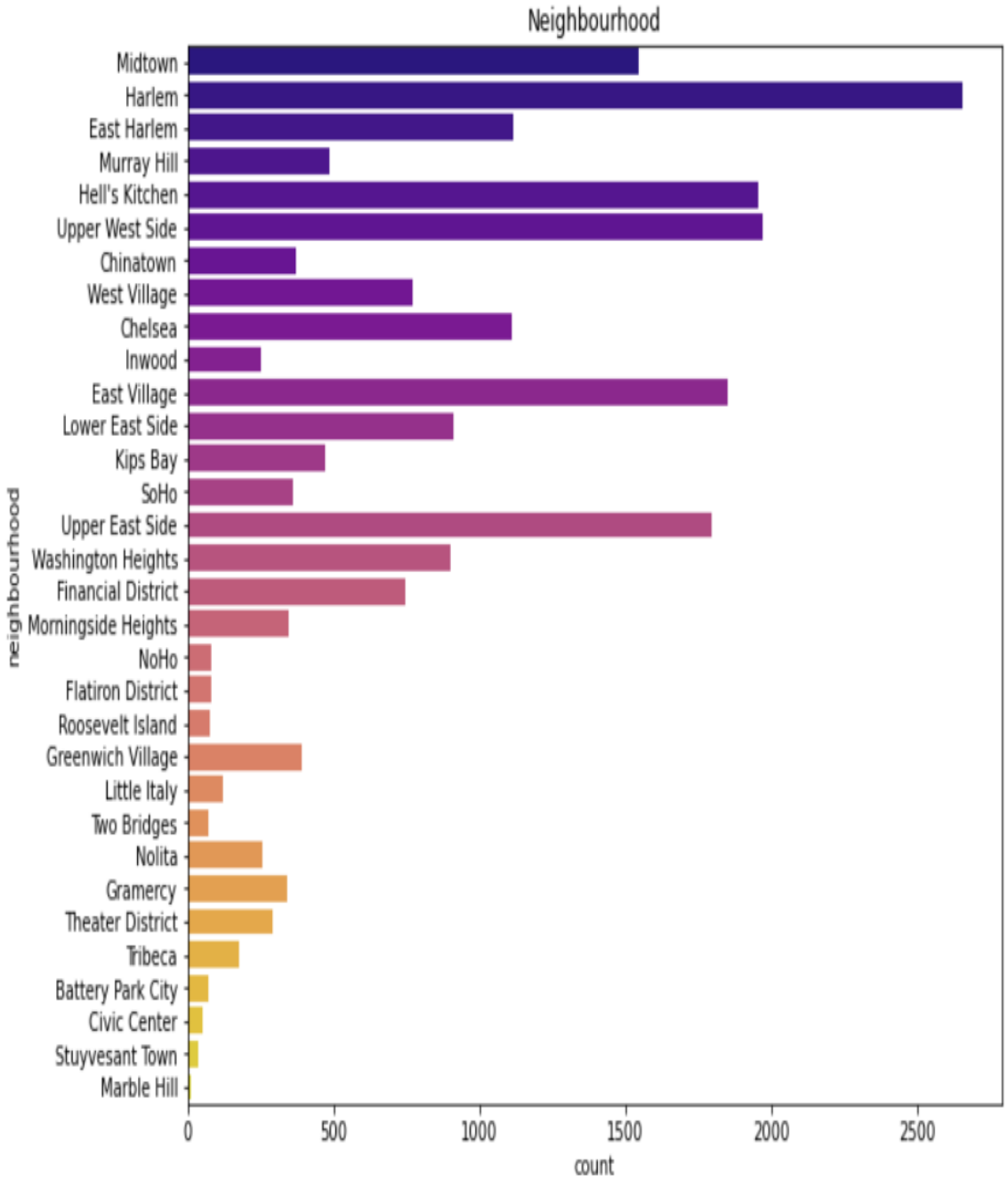
We see that Manhattan has the least number of neighbours among all other counties. This indicates that the holdings in New York are densely located in the neighbourhoods.

Queens has the highest number of neighbours (51).

11. The neighbourhoods with most holdings in each county.

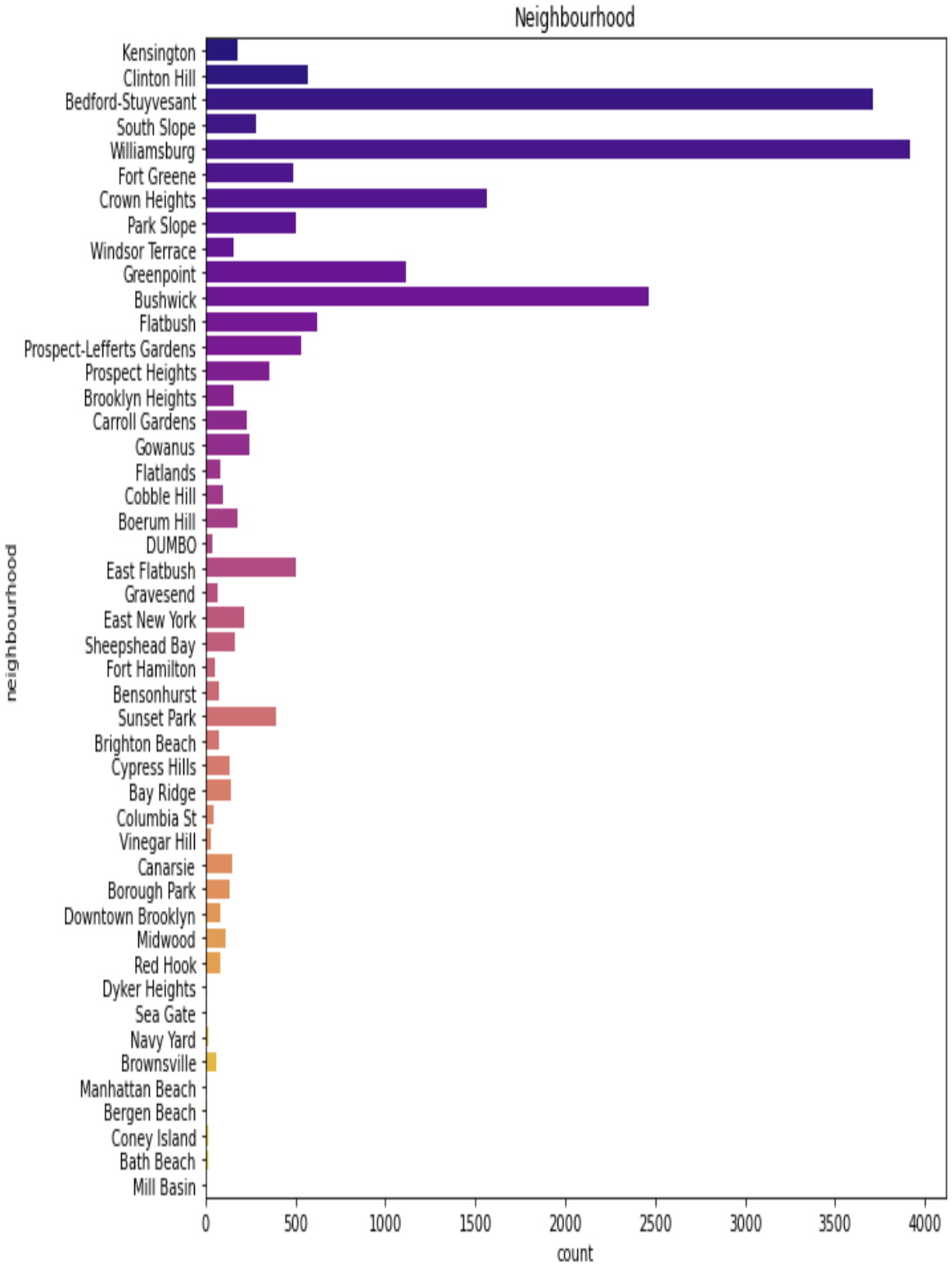
i) Manhattan

- There are a total of 32 neighbourhoods with Airbnb holdings.
- The most popular neighbourhood is Harlem (2568) followed by Upper West Side (1971).
- The least popular neighbourhood are Marble Hill (12) and Stuyvesant Town (37).



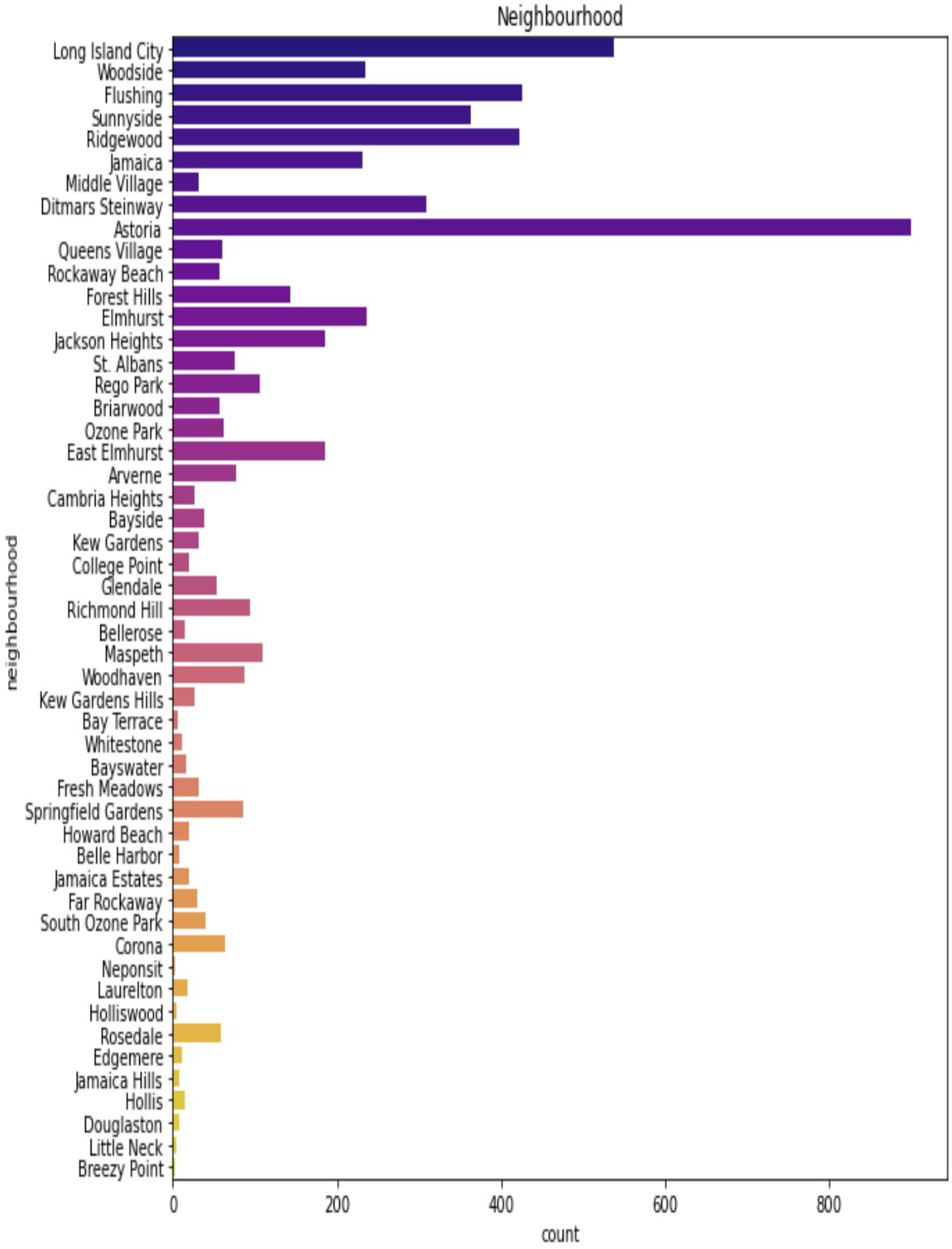
ii) Brooklyn

- There are a total of 47 neighbourhoods with Airbnb holdings.
- The most popular neighbourhood is Williamsburg (3920) followed by Bedford-Stuyvesant (3714).
- The least popular neighbourhood are Mill basin (4) and Sea Gate (7).



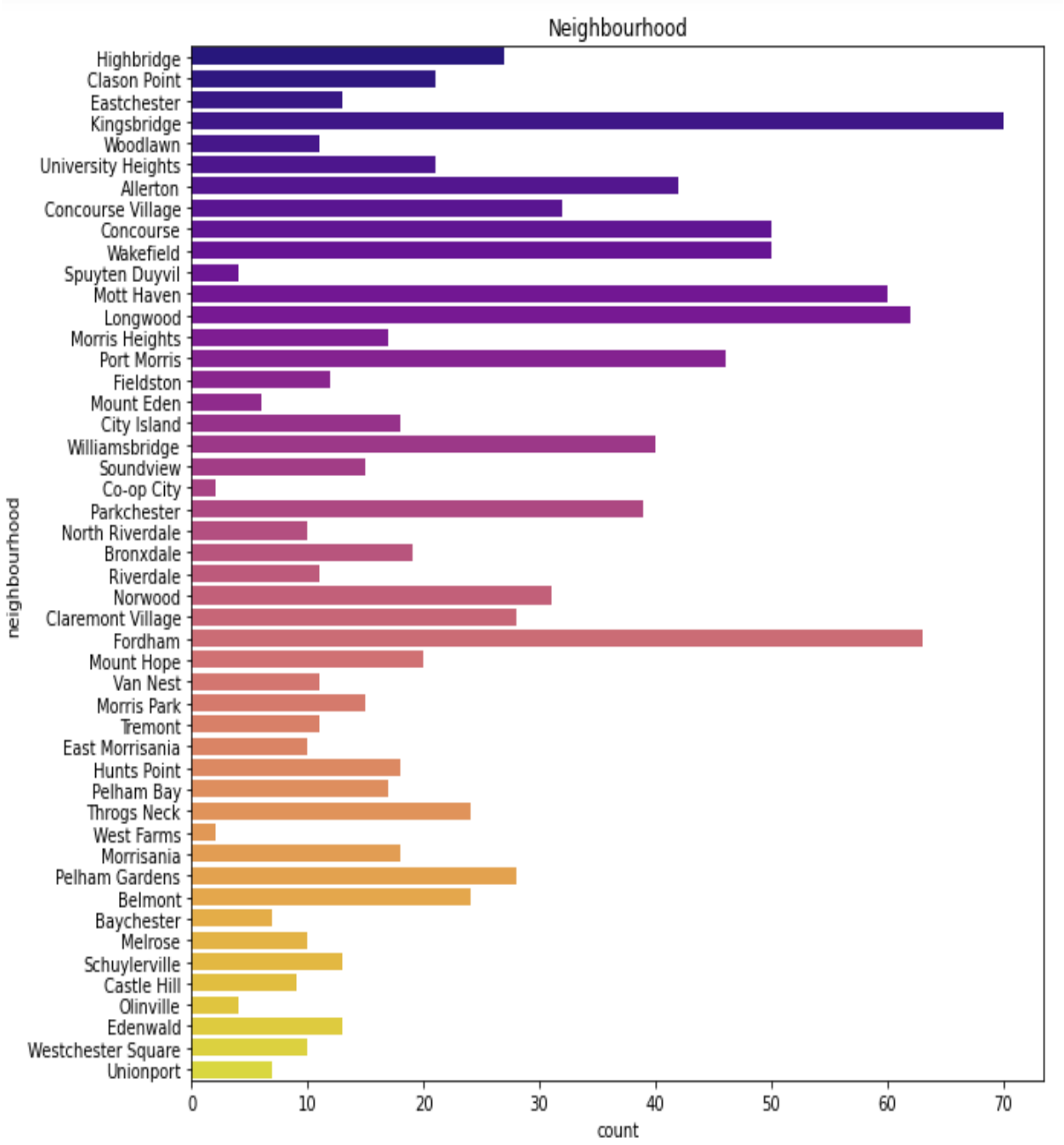
iii) Queens

- There are a total of 51 neighbourhoods with Airbnb holdings.
- The most popular neighbourhood is Astoria (900) followed by Long Island City (537).
- The least popular neighbourhood are Breezy Point (3) and Neponsit (3).



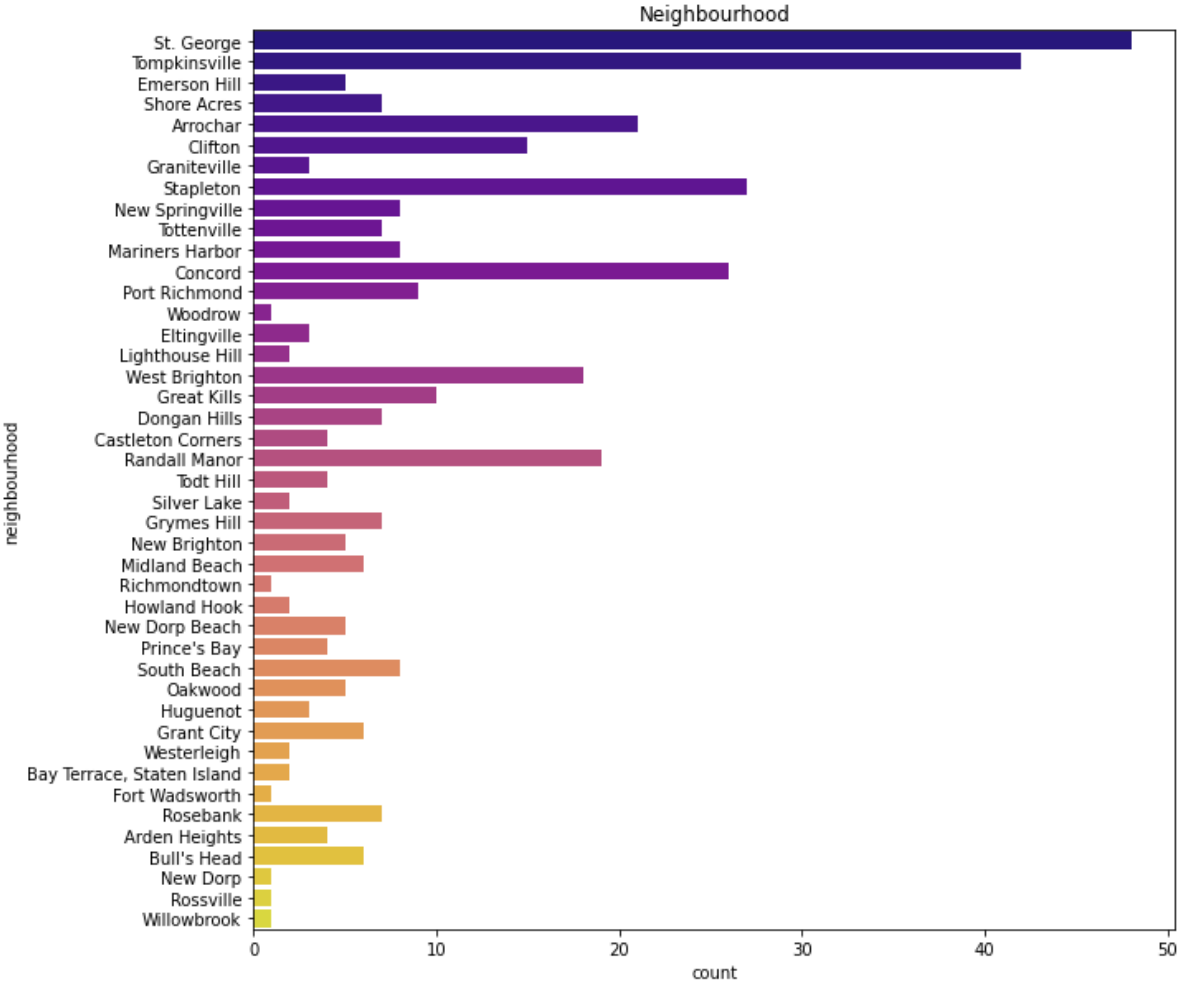
iv) Bronx

- There are a total of 48 neighbourhoods with Airbnb holdings.
- The most popular neighbourhood is Kingsbridge (70) followed by Fordham (63).
- The least popular neighbourhood are West Farms (2) and Co-op City (2).



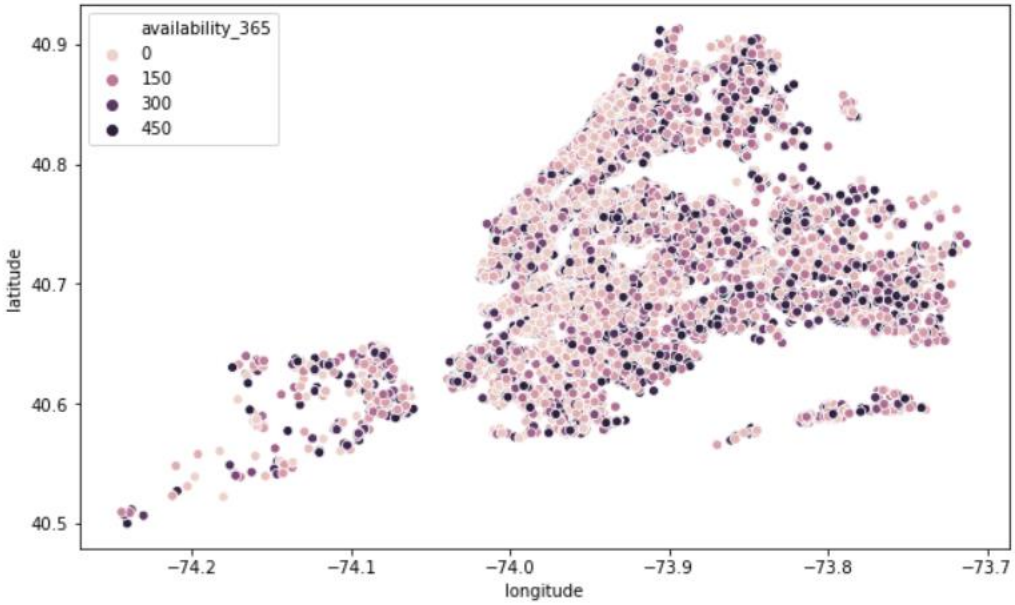
v) Staten Island

- There are a total of 43 neighbourhoods with Airbnb holdings.
- The most popular neighbourhood is St. George (48) followed by Tompkinsville (42).
- The least popular neighbourhood are Willowbrook (1) and Richmond town (1).



Analysis of availability of holdings.

In [157]:
```
plt.figure(figsize=(10,6))
sns.scatterplot(df.longitude,df.latitude,hue=df.availability_365)
plt.ioff()
```



Manhattan being a busy county has least available holdings whereas Queens and Bronx have a lot of available holdings.

Regression analysis for predicting the prices.

The necessary features like neighbourhood_group, room_type, price, holdings count and availability were used for regression analysis. Also, the rows with Nan and Null values from these features were dropped. Only 11 rows are dropped out of 48895 rows, hence it does not make a huge impact.

The neighbourhood_group and room_type features are categorical; hence these are encoded using LabelEncoder from Scikit-learn library.

In [158]:
```
df.drop(['host_id','latitude','longitude','neighbourhood','number_of_reviews',
         'reviews_per_month', 'name','id','host_name','last_review'], axis=1, inplace=True)

df.head(5)

df.dropna(how='any',inplace=True)
```

In [159]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

df['neighbourhood_group'] = le.fit_transform(df['neighbourhood_group'])
df['room_type'] = le.fit_transform(df['room_type'])

df.head()
```

The necessary libraries are imported and the data is split into training and testing.

The decision tree regressor is fit on the training set and its score is calculated on the testing data set.

R^2 score = 0.2474.

The R^2 score should be around 1 if the model fits well, I did not get a good R^2 score suggesting that the model needs tuning and is under fit.

In [163]:
```
from sklearn.tree import DecisionTreeRegressor
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.1,random_state=105)
DTree=DecisionTreeRegressor(min_samples_leaf=.0001)
DTree.fit(x_train,y_train)
y_predict=DTree.predict(x_test)
from sklearn.metrics import r2_score
r2_score(y_test,y_predict)
```

Out[163]:
```
0.24742965608385814
```