Airbnb Methodology

1. I have used Excel and Python for the initial analysis of the dataset.

- Read the data.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

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

- This dataset has around 48,895 observations in it with 16 columns and it is a mix of categorical and numeric values.

In [6]:
```
df
```

Out[6]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	numb
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	
...	...	...	...	...	...	...	...	...	...	...	...	...
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73.94995	Private room	70	2	
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73.93317	Private room	40	4	
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73.94867	Entire home/apt	115	10	
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73.99112	Shared room	55	1	
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73.98933	Private room	90	7	

48895 rows × 16 columns

In [4]:
```
df.info()
```

```
<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
```

- Identify outliers of the dataset.

```
In [5]: 1 df.describe()
```

Out[5]:

host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	38843.000000	48895.000000	48895.000000
3.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	1.373221	7.143982	112.781327
7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	1.680442	32.952519	131.622289
2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.010000	1.000000	0.000000
7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	0.190000	1.000000	0.000000
3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	0.720000	1.000000	45.000000
1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	2.020000	2.000000	227.000000
2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	58.500000	327.000000	365.000000

As we can of the above picture, the following columns (Price, minimum_nights, number_of_reviews, reviews_per_month, calculated_host_listing_count) have outliers because the maximum is much larger than the 75% percentile.

- Identified missing values and missing values percentage.

```
In [25]: 1 df.isnull().sum()
```

```
Out[25]: 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
```

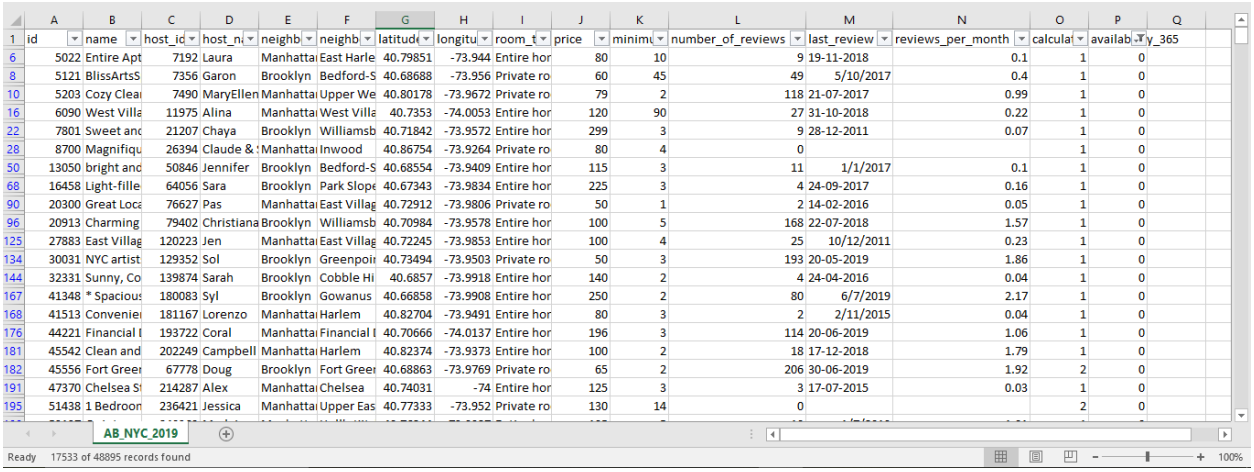
```
In [27]: 1 df.isnull().sum()*100/len(df)
```

```
Out[27]: id                                0.000000
         name                             0.032723
         host_id                          0.000000
         host_name                        0.042949
         neighbourhood_group              0.000000
         neighbourhood                    0.000000
         latitude                         0.000000
         longitude                        0.000000
         room_type                        0.000000
         price                            0.000000
         minimum_nights                   0.000000
         number_of_reviews                0.000000
         last_review                     20.558339
         reviews_per_month               20.558339
         calculated_host_listings_count   0.000000
         availability_365                 0.000000
         dtype: float64
```

- Last_review and reviews_per_ month columns are blank when number_of_reviews column is 0

L	M	N
number_of_reviews	last_review	reviews_per_month
0		
0		
0		
0		
0		
0		
0		
0		
0		
0		
0		
0		
0		
0		
0		
0		

- There are 17,533 out of 48,895 records (35% of the dataset) that have zero day of available for booking

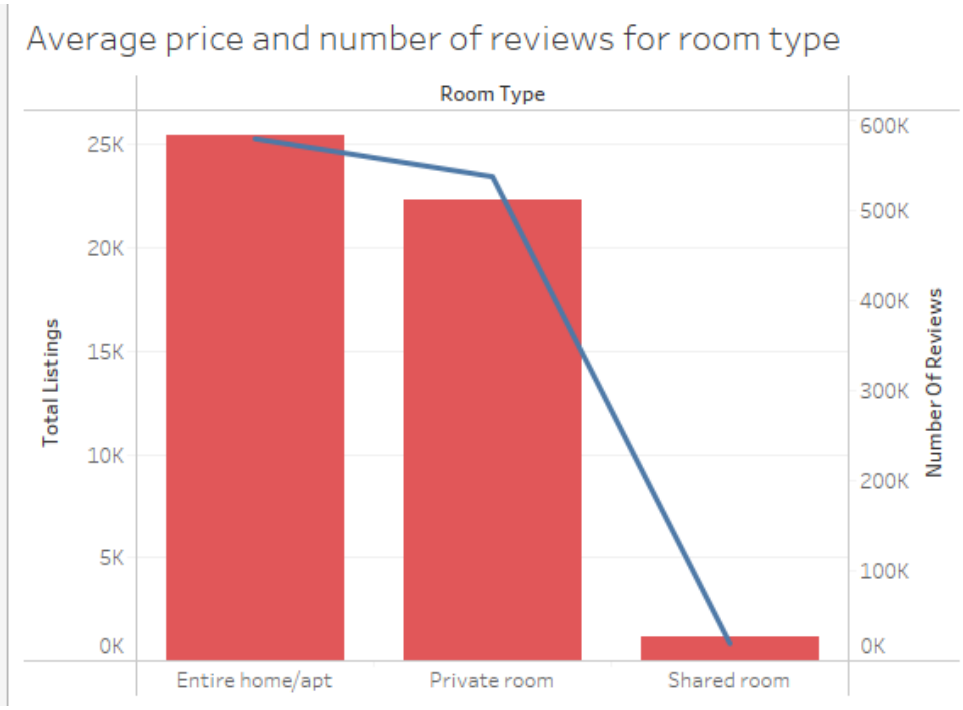


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	id	name	host_id	host_n	neighb	neighb	latitud	longitu	room_t	price	minim	number_of_reviews	last_review	reviews_per_month	calcula	availab	y_365
6	5022	Entire Apt	7192	Laura	Manhatta	East Harle	40.79851	-73.944	Entire hor	80	10	9	19-11-2018	0.1	1	0	
8	5121	BlissArtsS	7356	Garon	Brooklyn	Bedford-S	40.68688	-73.956	Private ro	60	45	49	5/10/2017	0.4	1	0	
10	5203	Cozy Clea	7490	MaryEllen	Manhatta	Upper We	40.80178	-73.9672	Private ro	79	2	118	21-07-2017	0.99	1	0	
16	6090	West Villa	11975	Alina	Manhatta	West Villa	40.7353	-74.0053	Entire hor	120	90	27	31-10-2018	0.22	1	0	
22	7801	Sweet and	21207	Chaya	Brooklyn	Williamsb	40.71842	-73.9572	Entire hor	299	3	9	28-12-2011	0.07	1	0	
28	8700	Magnifiqu	26394	Claude &	Manhatta	Inwood	40.86754	-73.9264	Private ro	80	4	0			1	0	
50	13050	bright and	50846	Jennifer	Brooklyn	Bedford-S	40.68554	-73.9409	Entire hor	115	3	11	1/1/2017	0.1	1	0	
68	16458	Light-fille	64056	Sara	Brooklyn	Park Slope	40.67343	-73.9834	Entire hor	225	3	4	24-09-2017	0.16	1	0	
90	20300	Great Loca	76627	Pas	Manhatta	East Villag	40.72912	-73.9806	Private ro	50	1	2	14-02-2016	0.05	1	0	
96	20913	Charming	79402	Christiana	Brooklyn	Williamsb	40.70984	-73.9578	Entire hor	100	5	168	22-07-2018	1.57	1	0	
125	27883	East Villag	120223	Jen	Manhatta	East Villag	40.72245	-73.9853	Entire hor	100	4	25	10/12/2011	0.23	1	0	
134	30031	NYC artist	129352	Sol	Brooklyn	Greenpoir	40.73494	-73.9503	Private ro	50	3	193	20-05-2019	1.86	1	0	
144	32331	Sunny, Co	139874	Sarah	Brooklyn	Cobble Hi	40.6857	-73.9918	Entire hor	140	2	4	24-04-2016	0.04	1	0	
167	41348	* Spacious	180083	Syl	Brooklyn	Gowanus	40.66858	-73.9908	Entire hor	250	2	80	6/7/2019	2.17	1	0	
168	41513	Convenie	181167	Lorenzo	Manhatta	Harlem	40.82704	-73.9491	Entire hor	80	3	2	2/11/2015	0.04	1	0	
176	44221	Financial I	193722	Coral	Manhatta	Financial I	40.70666	-74.0137	Entire hor	196	3	114	20-06-2019	1.06	1	0	
181	45542	Clean and	202249	Campbell	Manhatta	Harlem	40.82374	-73.9373	Entire hor	100	2	18	17-12-2018	1.79	1	0	
182	45556	Fort Greer	67778	Doug	Brooklyn	Fort Greer	40.68863	-73.9769	Private ro	65	2	206	30-06-2019	1.92	2	0	
191	47370	Chelsea S	214287	Alex	Manhatta	Chelsea	40.74031	-74	Entire hor	125	3	3	17-07-2015	0.03	1	0	
195	51438	1 Bedroon	236421	Jessica	Manhatta	Upper Eas	40.77333	-73.952	Private ro	130	14	0			2	0	

2. I have used Tableau for further analysis and visualization.

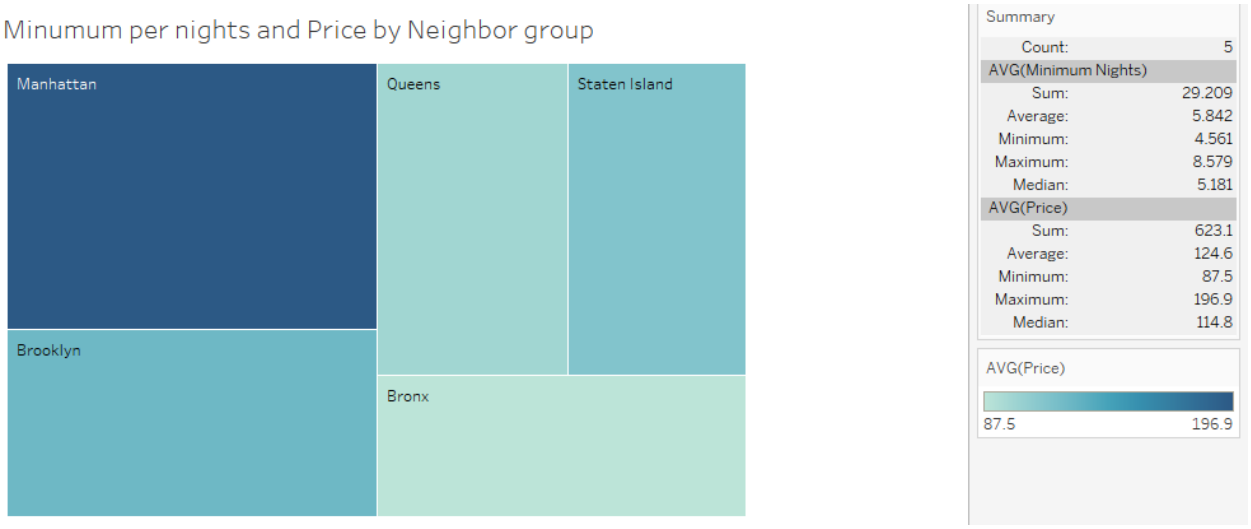
a. Average price and number of reviews for room type

I used a vertical bar graph with dual axis. The trends of distinct count of Id and Number Of Reviews for Room Type. Color shows details about distinct count of Id and Number Of Reviews.



b. Minimum per night and Price by Neighborhood Group

I used tree map to show the average Price (by color) and average Minimum Nights (by size). The marks are labeled by Neighborhood Group.

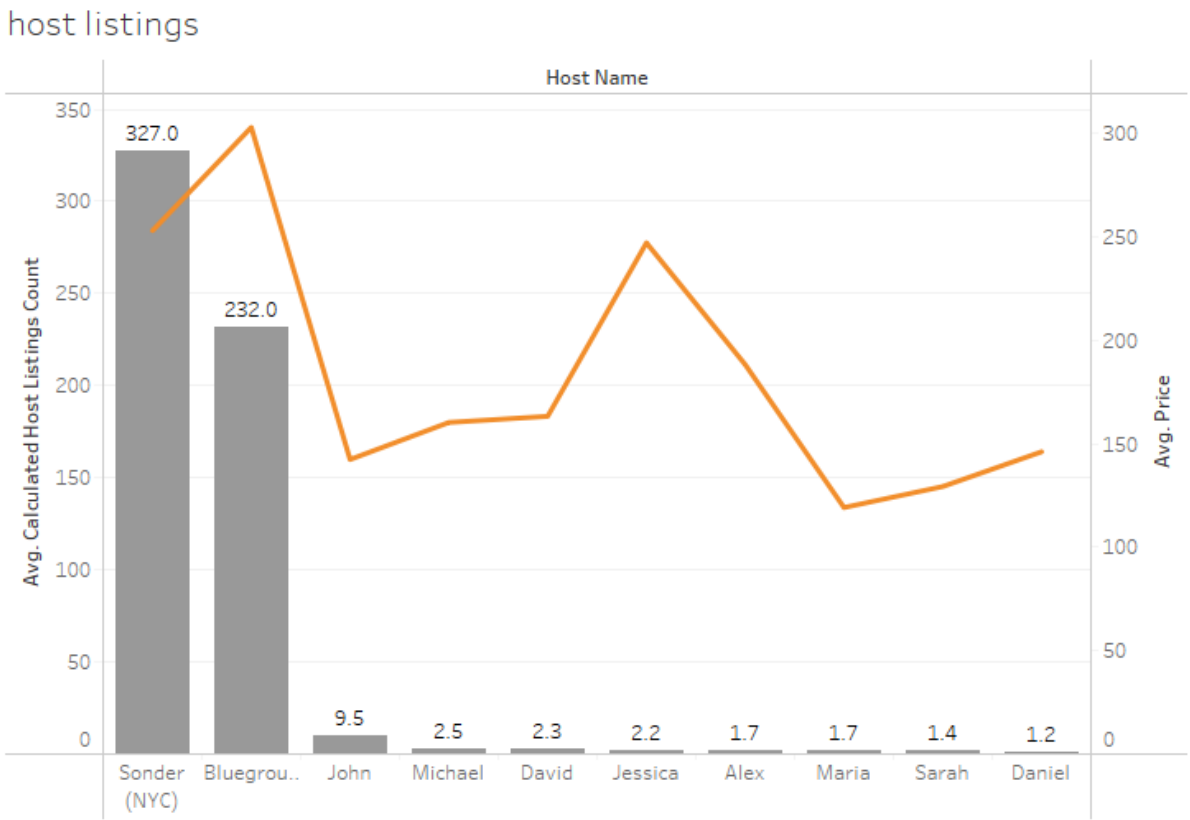


c. Map

Map based on Longitude and Latitude. Color shows details about Neighbourhood Group. Size shows sum of Price.

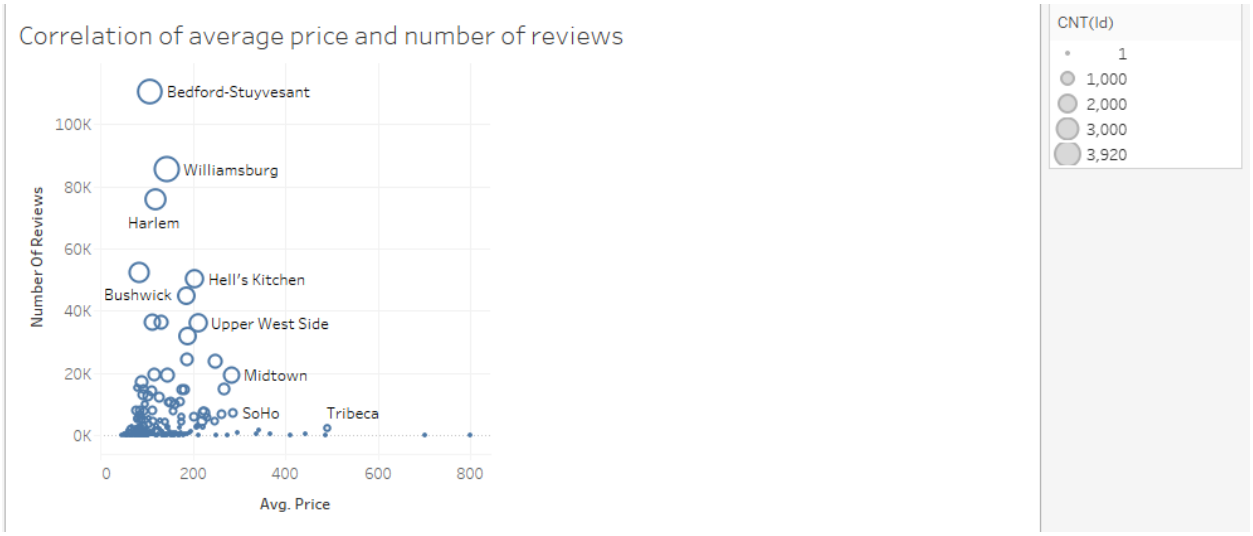
d. Top 10 host listings

Dual axis bar chart. The trends of the average of Calculated Host Listings Count and Avg. Price for Host Name. For pane Average of Price: Color shows details about Avg. Price. The view is filtered on Host Name, which keeps 10 of 11,428 members.

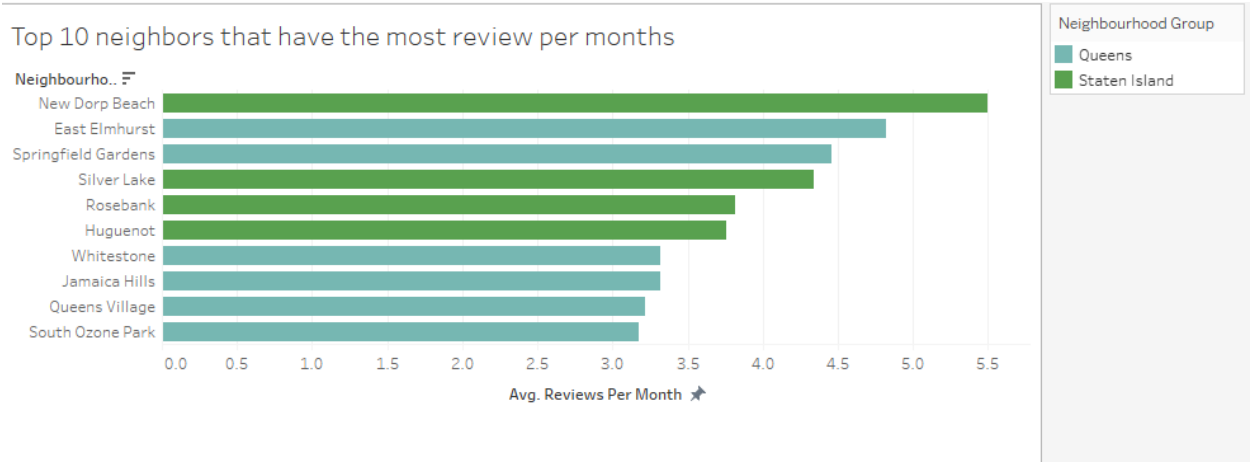


e. Correlation of average price and number of reviews

Scatter plot. Average of Price vs. sum of Number Of Reviews. Size shows count of Id. The marks are labeled by Neighbourhood.



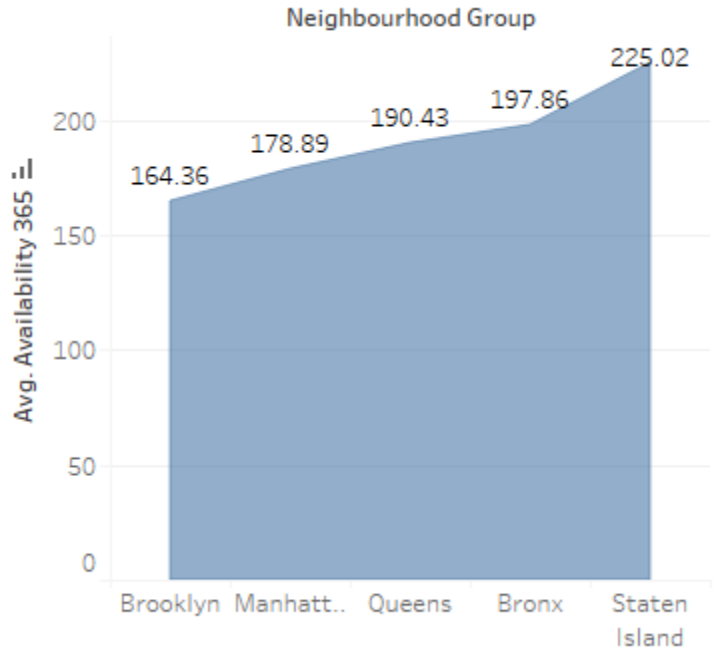
f. Top 10 neighborhood have the most review per month



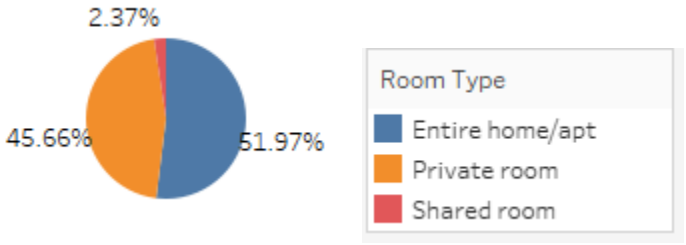
Horizontal bar chart. Average of Reviews Per Month for each Neighbourhood. Color shows details about Neighbourhood Group. Details are shown for the Neighbourhood Group. The view is filtered on Neighbourhood, which keeps 10 of 221 members.

g. Average Availability 365 per neighborhood group

Area chart. Average of Availability 365 for each Neighbourhood Group. The data is filtered on Availability 365, which ranges from 1 to 365.



h. Pie chart shows proportion of number of listings per room type.



3. I used PowerPoints to conduct the data storytelling to present to (1) Data Analysis Managers and Lead Data Analyst and (2) Head of Acquisitions and Operations, NYC and Head of User Experience, NYC