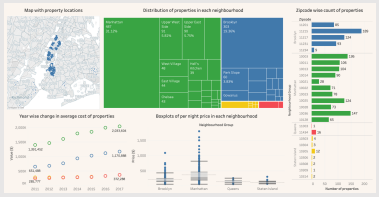
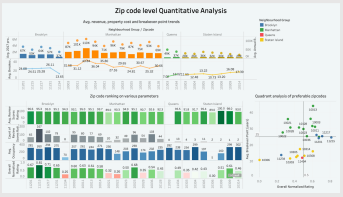
DATA CHALLENGE REPORT

Airbnb & Zillow Data Challenge





RAHUL BHASIN

UCID: 13255512

1. INTRODUCTION

1.1 Problem statement

A real estate company that has a niche in purchasing properties to rent out short-term as part of their business model specifically within New York City. The real estate company has already concluded that two-bedroom properties are the most profitable; however, they do not know which zip codes are the best to invest in.

The real estate company has engaged a firm to build out a data product and we have to provide conclusions to help them understand which zip codes would generate the most profit on short term rentals within New York City.

1.2 Data available

- Cost data: Zillow provides us an estimate of value for two-bedroom properties
- Revenue data: Airbnb is the medium through which the investor plans to lease out their investment property. Fortunately for you, we are able to see how much properties in certain neighborhoods rent out for in New York City

1.3 Assumptions made

- The investor will pay for the property in cash (i.e. no mortgage/interest rate will need to be accounted for).
- The time value of money discount rate is 0% (i.e. $1 today is worth the same 100 years from now).
- All properties and all square feet within each locale can be assumed to be homogeneous (i.e. a 1000 square foot property in a locale such as Bronx or Manhattan generates twice the revenue and costs twice as much as any other 500 square foot property within that same locale.)
- For dimensionality reduction of Zillow dataset, month wise cost price of 1996-04 to 2017-06 period was reduced to year wise cost price for this time period using median of each year. Median was used instead of average because it is prone to outlier values (slow economic growth etc.)

1.4 Platforms used

I have used Python programming language for data importing, data cleaning and exploratory data analysis purpose. Based on cleaned dataset using Python, I have used Tableau for data visualizations.

2. DATA PREPEARATION AND INITIAL EXPLORATION

Some of the steps I have followed in data preparation for this data challenge are files importing, dataset summary, missing value check, duplicate rows check, feature engineering by new variables creation, data cleaning by making zip codes in 5 digits format, datasets merging, data filtering and selecting only required columns.

2.1 Data importing, data cleaning and EDA for Zillow Dataset

Importing Python libraries

```
import pandas as pd
import numpy as np
import missingno as msno        # For checking missing values
import pandas_profiling                # For profiling of each variable
import seaborn as sns                  # For data Visualizations

import matplotlib.pyplot as plt        # For data Visualizations
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')

pd.set_option('display.min_rows',150)
pd.set_option('display.max_rows',150)
pd.set_option('display.max_columns',150)
```

Reading Zillow dataset and displaying its data

```
zillow_v1=pd.read_csv('Zip_Zhvi_2bedroom.csv')
zillow_v1
```

	RegionID	RegionName	City	State	Metro	CountyName	SizeRank	1996-04	1996-05	1996-06	1996-07	1996-08	1996-09	1996-
0	61639	10025	New York	NY	New York	New York	1	NaN	NaN	NaN	NaN	NaN	NaN	Na
1	84654	60657	Chicago	IL	Chicago	Cook	2	167700.0	166400.0	166700.0	167200.0	166900.0	166900.0	16800(
2	61637	10023	New York	NY	New York	New York	3	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	84616	60614	Chicago	IL	Chicago	Cook	4	195800.0	193500.0	192600.0	192300.0	192600.0	193600.0	19550(
4	93144	79936	El Paso	TX	El Paso	El Paso	5	59100.0	60500.0	60900.0	60800.0	60300.0	60400.0	6120(
5	84640	60640	Chicago	IL	Chicago	Cook	6	123300.0	122600.0	122000.0	121500.0	120900.0	120600.0	12090(

➔ There are 8,946 rows and 262 columns in Zillow dataset.

Checking missing values, unique values and data types in Zillow dataset

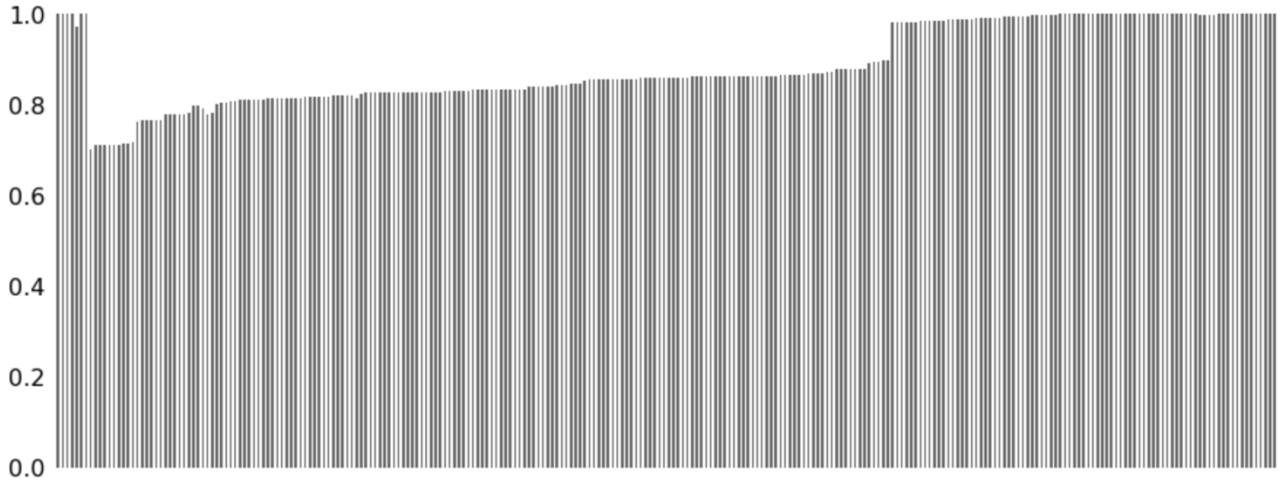
```
def basic_details(df):
    b = pd.DataFrame()
    b['# Missing values'] = df.isnull().sum()
    b['% Missing values'] = round((df.isnull().sum()/df.shape[0])*100,2)
    b['# Unique values'] = df.nunique()
    b['Data type'] = df.dtypes
    return b
basic_details(zillow_v1)
```

	# Missing values	% Missing values	# Unique values	Data type
RegionID	0	0.00	8946	int64
RegionName	0	0.00	8946	int64
City	0	0.00	4684	object
State	0	0.00	48	object
Metro	250	2.79	466	object
CountyName	0	0.00	722	object
SizeRank	0	0.00	8946	int64
1996-04	2662	29.76	1581	float64

- There are no missing values in variables RegionID, RegionName, City, State, CountyName and SizeRank. Only Metro and historical median price variables have missing values.
- All zip code values in RegionName column are unique.

Checking missing values using data visualization

```
p=msno.bar(zillow_v1, figsize=(15, 6))
```



- As, we can see from above data visualization, historical median price variables of 1996-04 to 2017-06 period have missing values and the proportion of these missing values is decreasing with increase in time period. This might be because historical cost price of these properties was not available, and it started becoming available for recent time periods.

Checking if there are duplicate rows in data

```
zillow_v1[zillow_v1.duplicated() == True
```

RegionID	RegionName	City	State	Metro	CountyName	SizeRank	1996-04	1996-05	1996-06	1996-07	1996-08	1996-09	1996-10	1996-11	1996-12	1997-01	1997-02	1997-03

0 rows × 262 columns

- There are no duplicate rows in Zillow dataset.

Displaying summary statistics of Zillow dataset

```
zillow_v1.describe()
```

	RegionID	RegionName	SizeRank	1996-04	1996-05	1996-06	1996-07	1996-08	1996-09	1996-10
count	8946.000000	8946.000000	8946.000000	6284.000000	6364.000000	6364.000000	6369.000000	6370.000000	6370.000000	6370.000000
mean	80671.285938	47494.449027	4473.500000	93754.057925	93616.043369	93642.630421	93609.734652	93646.357928	93722.339089	93849.262166
std	31636.286116	30868.419487	2582.632088	44385.146499	44222.734487	44225.112290	44264.266105	44340.199368	44457.610118	44605.248316
min	58196.000000	1001.000000	1.000000	22400.000000	23500.000000	24500.000000	25400.000000	26200.000000	26700.000000	27200.000000
25%	66819.250000	21125.500000	2237.250000	64100.000000	64000.000000	64000.000000	64000.000000	63900.000000	63900.000000	64000.000000
50%	77191.500000	44404.000000	4473.500000	84500.000000	84500.000000	84600.000000	84600.000000	84700.000000	84700.000000	84850.000000
75%	92251.250000	77357.750000	6709.750000	111000.000000	110800.000000	111000.000000	110900.000000	110900.000000	110975.000000	111100.000000
max	738092.000000	99901.000000	8946.000000	420700.000000	422300.000000	430400.000000	440400.000000	447100.000000	453000.000000	454300.000000

8 rows × 258 columns

Calculating year-wise historical median price

```
years=['1996','1997','1998','1999','2000','2001','2002','2003','2004','2005','2006','2007','2008',
'2009','2010','2011'
       ,'2012','2013','2014','2015','2016','2017']
months=['01','02','03','04','05','06','07','08','09','10','11','12']
for y in years:
    year_month=[]
    for m in months:
        #print(y+'-'+m)
        year_month.append(y+'-'+m)
    #print(year_month)
    zillow_v1['Median-'+y]=zillow_v1.loc[:, zillow_v1.columns.isin(year_month)].median(axis=1)
```

➔ For dimensionality reduction of Zillow dataset, month wise cost price of 1996-04 to 2017-06 period was reduced to year wise cost price for this time period using median of each year. Median was used instead of average because it is prone to outlier values (slow economic growth etc.).

Keeping only required fields and displaying data

```
zillow_v2=zillow_v1[['RegionID','RegionName','City','State','Metro','CountyName','SizeRank','Media
n-1996','Median-1997','Median-1998','Median-1999','Median-2000','Median-2001','Median-
2002','Median-2003','Median-2004','Median-2005','Median-2006','Median-2007','Median-2008',
'Median-2009','Median-2010','Median-2011','Median-2012','Median-2013','Median-2014',
'Median-2015','Median-2016','Median-2017']]

zillow_v2['RegionName']=zillow_v2['RegionName'].astype(str)
zillow_v2
```

	RegionID	RegionName	City	State	Metro	CountyName	SizeRank	Median-1996	Median-1997	Median-1998	Median-1999	Median-2000	Median-2001	Media 20
0	61639	10025	New York	NY	New York	New York	1	NaN	NaN	NaN	NaN	NaN	NaN	Na
1	84654	60657	Chicago	IL	Chicago	Cook	2	167200.0	185350.0	200150.0	233400.0	268050.0	300900.0	313700
2	61637	10023	New York	NY	New York	New York	3	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	84616	60614	Chicago	IL	Chicago	Cook	4	193600.0	213500.0	233200.0	259350.0	296050.0	326950.0	342900
4	93144	79936	El Paso	TX	El Paso	El Paso	5	60800.0	58250.0	57950.0	58600.0	59850.0	58100.0	56900
5	84640	60640	Chicago	IL	Chicago	Cook	6	121500.0	127300.0	137000.0	153350.0	176350.0	208100.0	225800

Data Profiling (Output to an HTML file)

```
profile=pandas_profiling.ProfileReport(zillow_v2[['RegionName','City','State','Metro','CountyName'
,'Median-2017']])
profile.to_file(output_file="output.html")
```

Open 'zillow_data_profiling.html' file to see its detailed data profiling.

Key Insights from Data Profiling of Zillow dataset: -

- ➔ 'City' variable has 4684 (52.4%) distinct values with 'Los Angeles' being the most repeated city followed by Chicago and Indianapolis.
- ➔ For median value of 2017 property price, average, minimum and maximum values are $213753, $29000 and $3212450 respectively.
- ➔ Zillow dataset contains records of 48 US states and maximum for California state. However, later we would be focusing only on the New York city data as Airbnb dataset is of New York city.
- ➔ 'CountyName' variable has 722 (8.1%) distinct values with 'Los Angeles' being the most common CountyName.

2.2 Data importing, data cleaning and EDA for Airbnb Dataset

Reading Airbnb listings dataset and displaying it's top 5 rows

```
listings_v1=pd.read_csv('listings.csv')
listings_v1.head()
```

	id	listing_url	scrape_id	last_scraped	name	summary	space	description	experiences_offered	n
0	2539	https://www.airbnb.com/rooms/2539	20190708031610	2019-07-09	Clean & quiet apt home by the park	Renovated apt home in elevator building.	Spacious, renovated, and clean apt home, one b...	Renovated apt home in elevator building. Spaci...	none	a
1	2595	https://www.airbnb.com/rooms/2595	20190708031610	2019-07-09	Skylit Midtown Castle	Find your romantic getaway to this beautiful, ...	- Spacious (500+ft²), immaculate and nicely fu...	Find your romantic getaway to this beautiful, ...	none	

- ➔ There are 48,8958 rows and 106 columns in Airbnb listings dataset.

Checking missing values, unique values and data types in Airbnb listings dataset

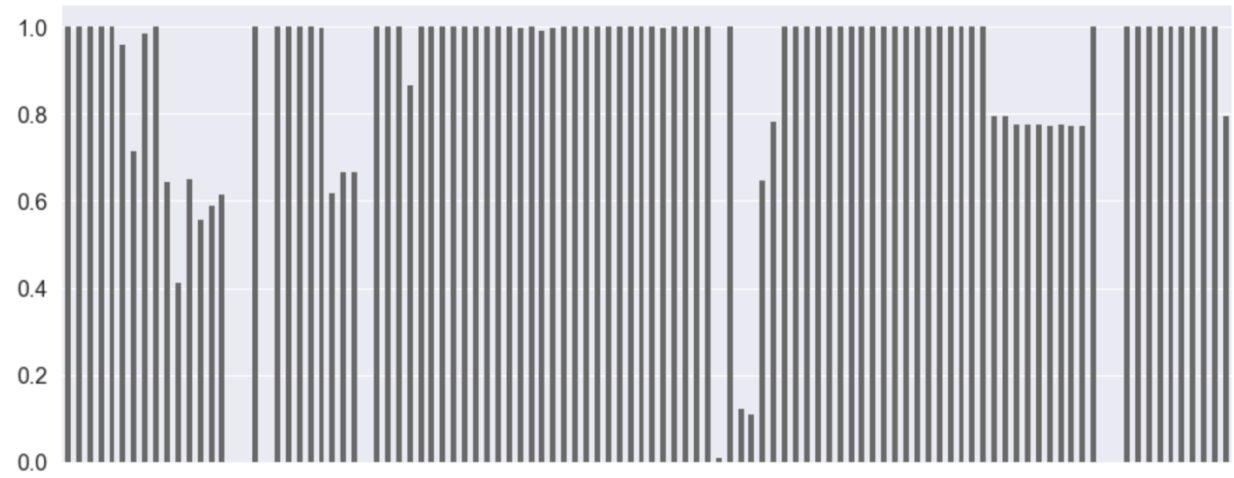
```
def basic_details(df):
    b = pd.DataFrame()
    b['# Missing values'] = df.isnull().sum()
    b['% Missing values'] = round((df.isnull().sum()/df.shape[0])*100,2)
    b['# Unique values'] = df.nunique()
    b['Data type'] = df.dtypes
    return b
basic_details(listings_v1)
```

	# Missing values	% Missing values	# Unique values	Data type
id	0	0.00	48895	int64
listing_url	0	0.00	48895	object
scrape_id	0	0.00	1	int64
last_scraped	0	0.00	2	object
name	16	0.03	47905	object
summary	2041	4.17	43805	object
space	14026	28.69	32170	object
description	781	1.60	46248	object

- ➔ There are missing values for around half of the columns. For zip code column, 1.06% of values are missing.
- ➔ Few of the columns like thumbnail_url, medium_ul and xl_picture_url are completely empty.

Checking missing values using data visualization

```
p=msno.bar(listings_v1, figsize=(15, 6))
```



Checking if there are duplicate rows in data

```
listings_v1[listings_v1.duplicated() == True]
```

id	listing_url	scrape_id	last_scraped	name	summary	space	description	experiences_offered	neighborhood_overview	notes	transit	access	interaction

➔ There are no duplicate rows in Airbnb listing dataset.

Displaying summary statistics of Airbnb listings dataset

```
listings_v1.describe()
```

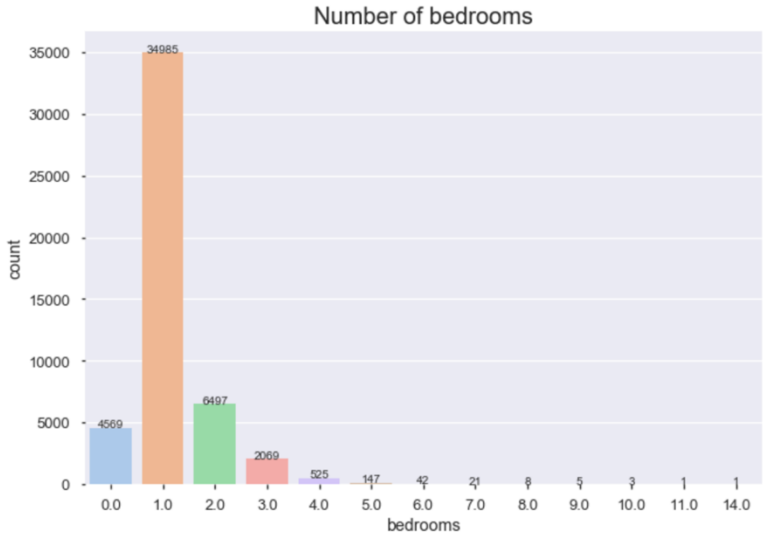
	id	scrape_id	thumbnail_url	medium_url	xl_picture_url	host_id	host_acceptance_rate	host_listings_count	host_total_listings_count
count	4.889500e+04	4.889500e+04	0.0	0.0	0.0	4.889500e+04	0.0	48874.000000	48874.000000
mean	1.901714e+07	2.019071e+13	NaN	NaN	NaN	6.762001e+07	NaN	14.281745	14.281745
std	1.098311e+07	5.449274e+00	NaN	NaN	NaN	7.861097e+07	NaN	84.151375	84.151375
min	2.539000e+03	2.019071e+13	NaN	NaN	NaN	2.438000e+03	NaN	0.000000	0.000000
25%	9.471945e+06	2.019071e+13	NaN	NaN	NaN	7.822033e+06	NaN	1.000000	1.000000
50%	1.967728e+07	2.019071e+13	NaN	NaN	NaN	3.079382e+07	NaN	1.000000	1.000000
75%	2.915218e+07	2.019071e+13	NaN	NaN	NaN	1.074344e+08	NaN	2.000000	2.000000
max	3.648724e+07	2.019071e+13	NaN	NaN	NaN	2.743213e+08	NaN	1070.000000	1070.000000

Cleaning 'Zipcode' variable by cleaning values containing . and -

```
listings_v1['zipcode']=listings_v1['zipcode'].astype(str).str.split(".").apply(lambda x: x[0])
listings_v1['zipcode']=listings_v1['zipcode'].astype(str).str.split("-").apply(lambda x: x[0])
```

Chart showing distribution of number of beds in Zillow dataset

```
plt.rcParams['figure.figsize'] = (15, 7)
plt.style.use('seaborn-talk')
ax=sns.countplot(listings_v1['bedrooms'], palette = 'pastel')
plt.title('Number of bedrooms', fontsize = 20)
for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x()+p.get_width()/2.,height + 5,'{:1}'.format(height),ha="center")
plt.show()
```



➔ The Airbnb dataset contains highest number of apartments with 1 bedroom (34,985) followed by 2 bedrooms (6,497). In this data challenge we will only be considering 2-bedroom apartments for our analysis because Zillow dataset contains only the average property price for 2 bedrooms.

2.3 Joining Zillow dataset and Airbnb listings dataset

Joining Zillow dataset and Airbnb listings dataset on zip codes

```
merged_data_v1=pd.merge(zillow_v2, listings_v1, how='inner', left_on='RegionName',
right_on='zipcode')
merged_data_v1
```

	RegionID	RegionName	City	State	Metro	CountyName	SizeRank	Median-1996	Median-1997	Median-1998	Median-1999	Median-2000	Median-2001	Median-2002	Medi 2
0	61639	10025	New York	NY	New York	New York	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	61639	10025	New York	NY	New York	New York	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	61639	10025	New York	NY	New York	New York	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

➔ Revenue and Cost Data are joined based on common regionName/Zipcode.
➔ There are 1,566 rows in Airbnb listings dataset.

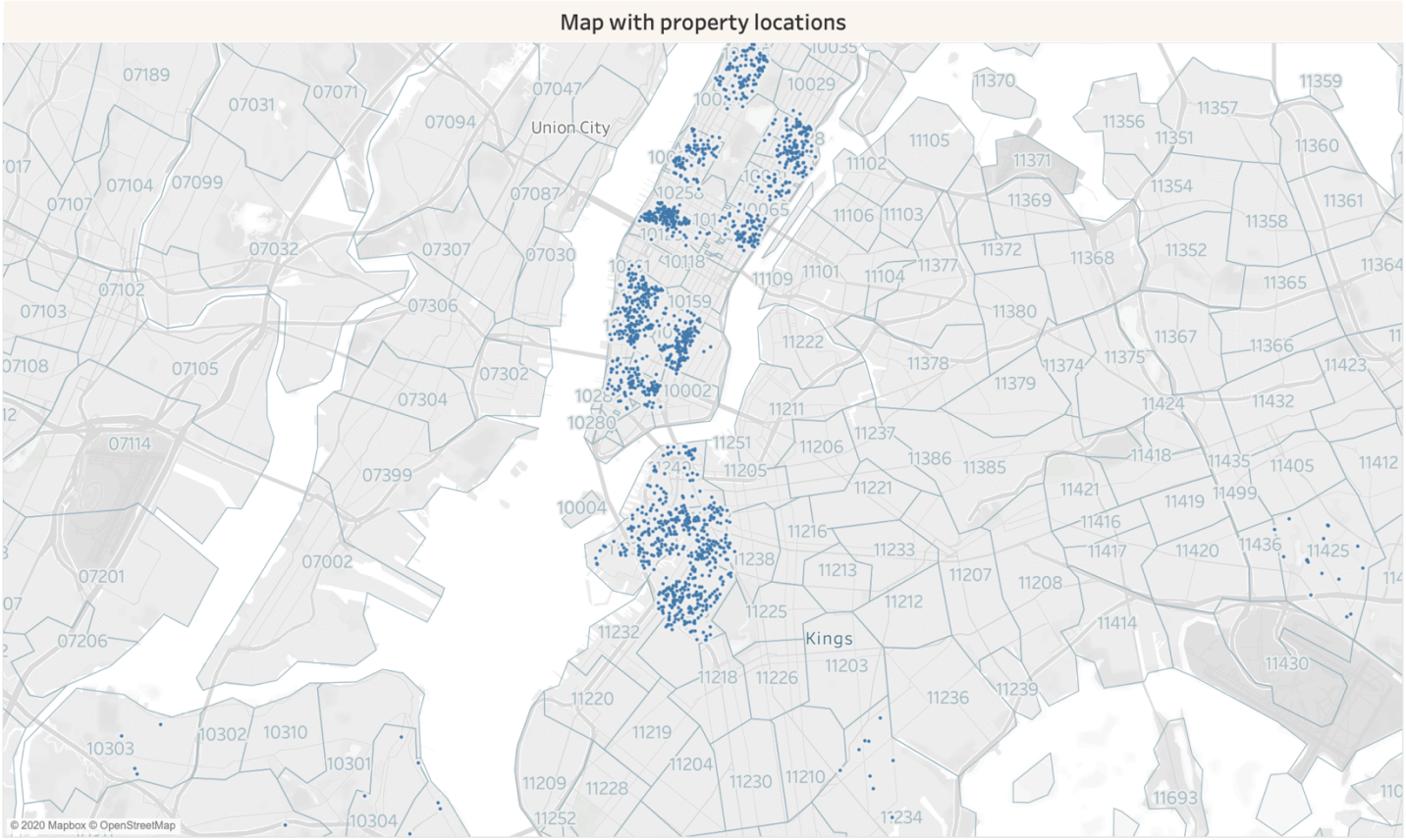
Filtering only 2 bedroom apartments from the combined dataset for further analysis

```
merged_data_v2=merged_data_v1[(merged_data_v1['bedrooms']==2.0)]
```

All the visualizations in the next section will based on this 'merged_data_v2' dataset prepared above.

3. DATA VIZUALIZATIONS AND KEY INSIGHTS

3.1 Map with property locations

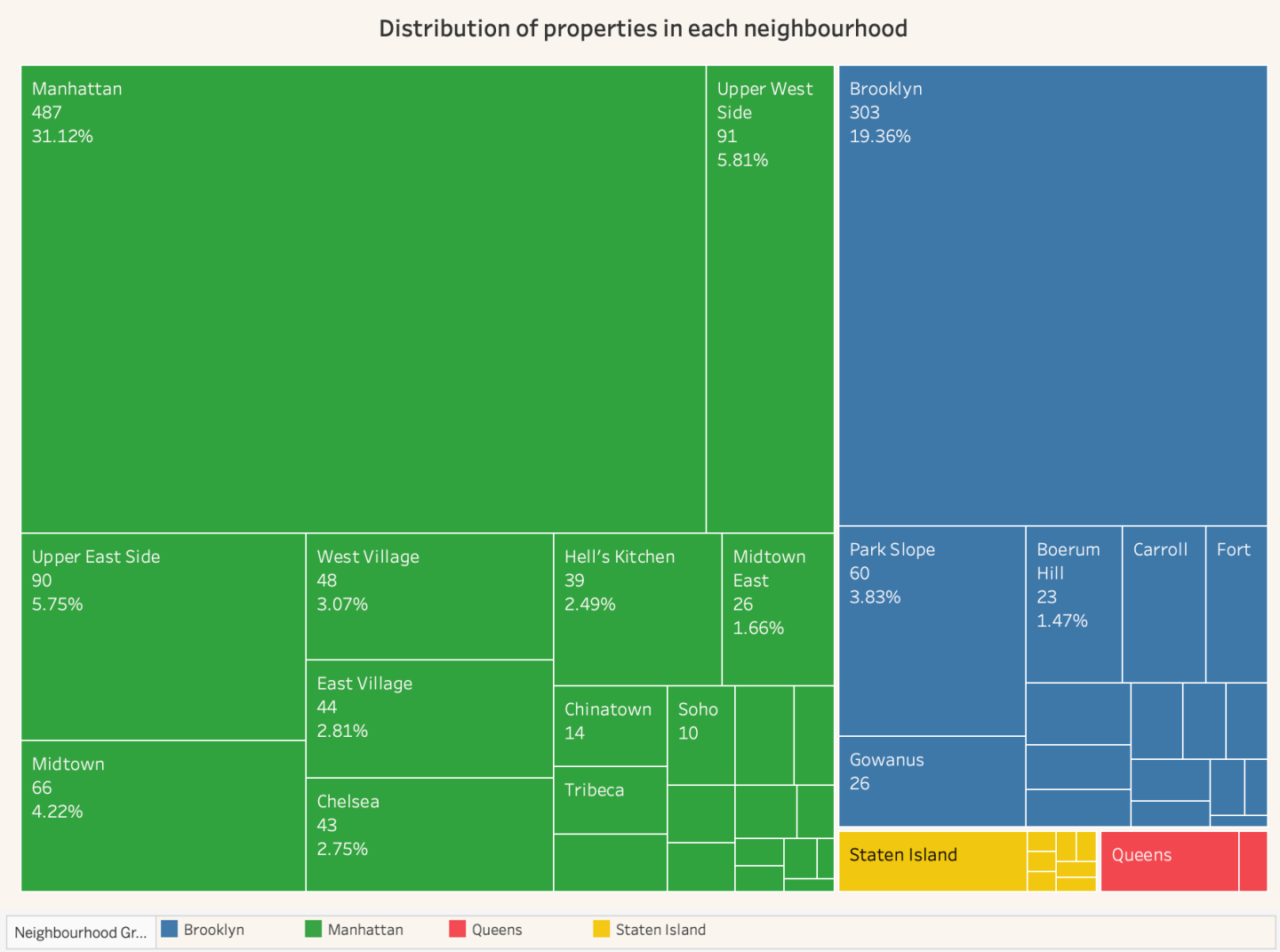


This visualization highlights the locations of properties in merged dataset of Zillow dataset and Airbnb listings on the New York map.

Key Insights from above visualization: -

- ➔ The properties in merged dataset are basically scattered across Manhattan, Brooklyn, Staten Island and Queens.
- ➔ Most of the properties are concentrated in Manhattan area.

3.2 Distribution of properties in each neighborhood

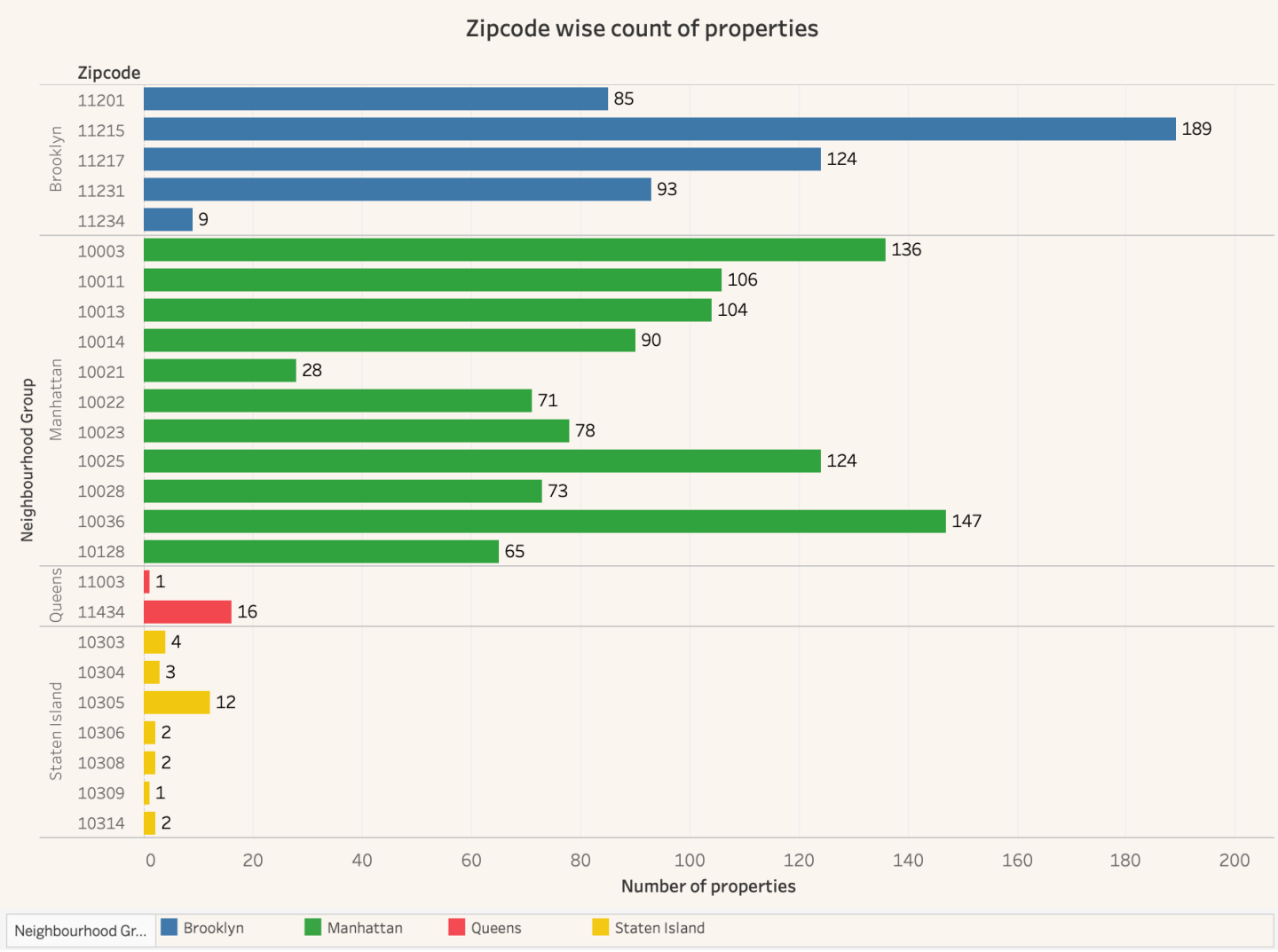


This tree map depicts the number and percentage of properties in various neighborhoods of New York.

Key Insights from above visualization: -

- Manhattan has the highest number of properties 487 (31.12%). Other top neighborhoods in Manhattan area are Upper West (5.81%), Upper East Side (5.75%), Midtown (4.22%) and West Village (3.07%).
- After Manhattan, Brooklyn has the second largest number of properties 303 (19.36%). Within Brooklyn, other top neighborhoods are Park Slope (3.83%), Boerum Hill (1.47%) and Carroll (1.28%).
- This is followed by Staten Island and Queens neighborhood groups with least number of properties.

3.3 Distribution of properties in each neighborhood

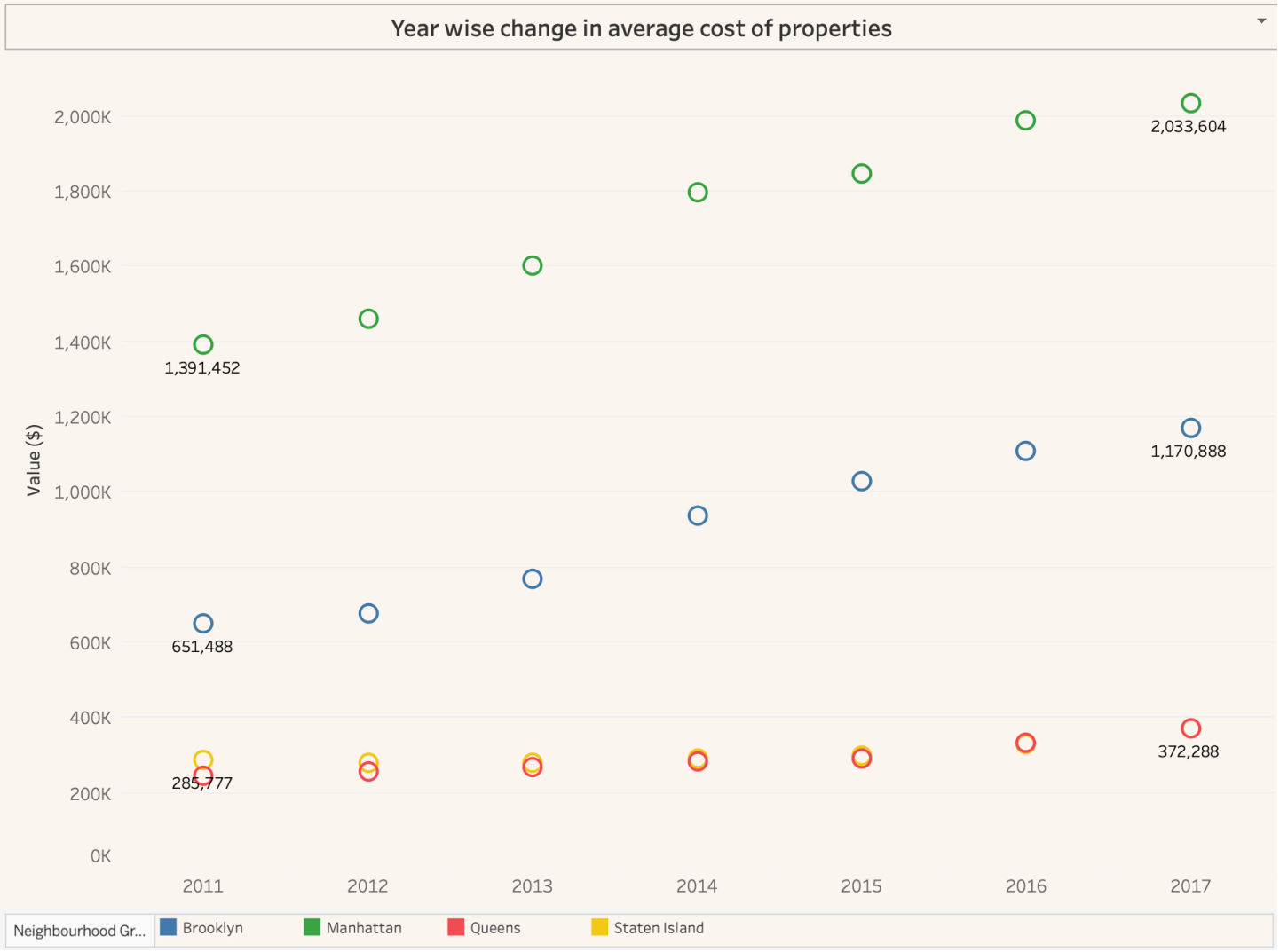


Above bar chart shows the count of properties at zip code level within each neighborhood group.

Key Insights from above visualization: -

- In merged dataset, there are total 30 zip codes with 11 zip codes in Manhattan, 5 zip codes in Brooklyn, 7 zip codes in Staten Island and 2 zip codes in Queens.
- The zip codes with most properties is 11215 in Brooklyn. Other top zip codes within Brooklyn are 11217, 11231 and 11201.
- Within Manhattan, top zip codes count of properties are 10036, 10003, 10025, 10011, and 10013.
- All the zip codes within Staten Island and Queens have quite less properties with maximum being 16 in zip code 11434 and minimum being just 1in zip codes 11003 and 10309.

3.4 Year wise change in average cost of properties

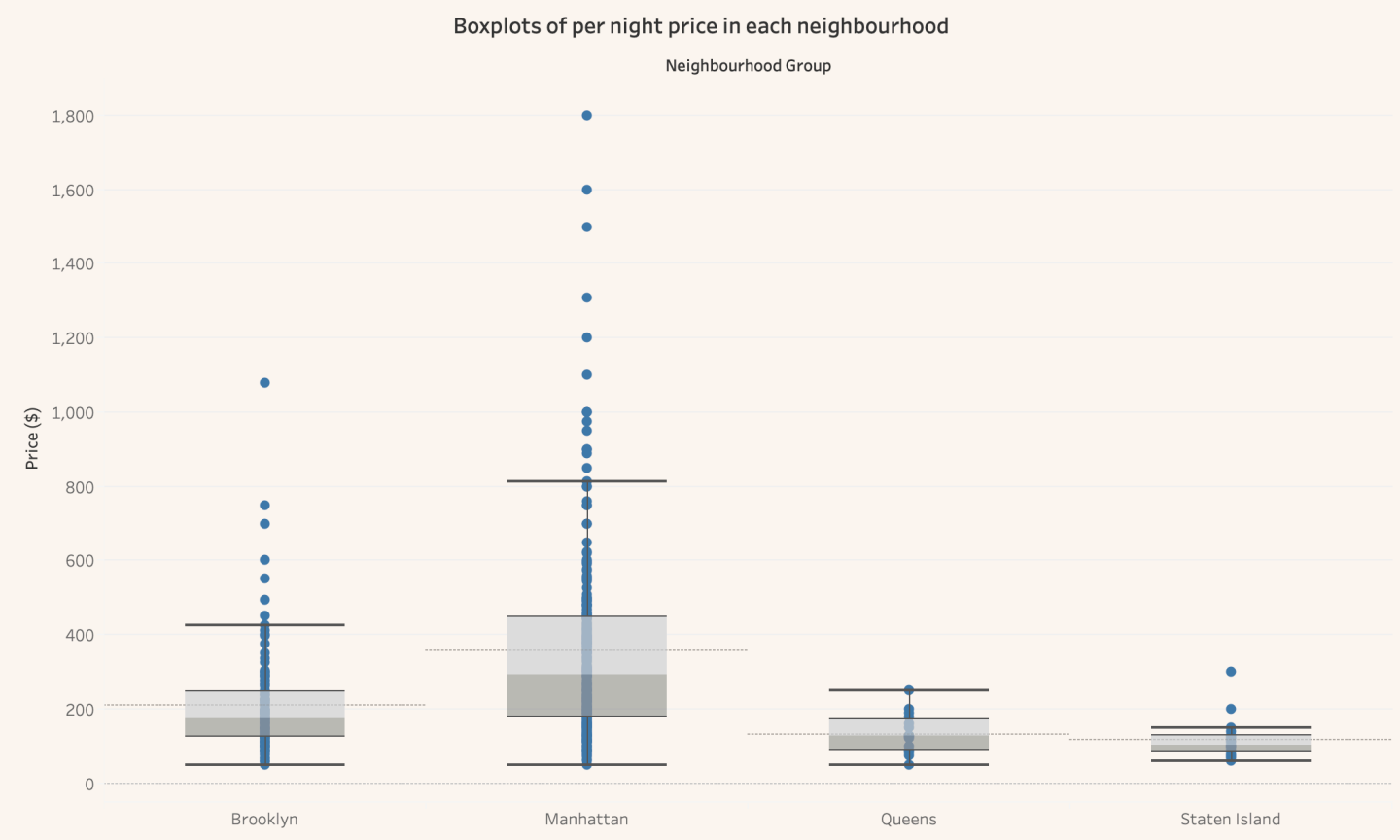


This chart highlights the change in average cost of properties between years 2011 to 2017 with in 4 neighborhood groups.

Key Insights from above visualization: -

- In each year, average cost of properties is highest in Manhattan followed by Brooklyn, Staten Island and then Queens.
- All the 4 neighborhood groups show that after each year the average cost of properties increase.
- Manhattan and Brooklyn had higher growth in average property cost. The average cost of apartment in Brooklyn in 2011 was $651K which went up to as high as $1,170K (79% increase). Similarly, for Manhattan the average cost of apartment in 2011 was $1,391,452 which went up to as high as $2,033,604 (46% growth)

3.5 Boxplots of per night price in each neighborhood



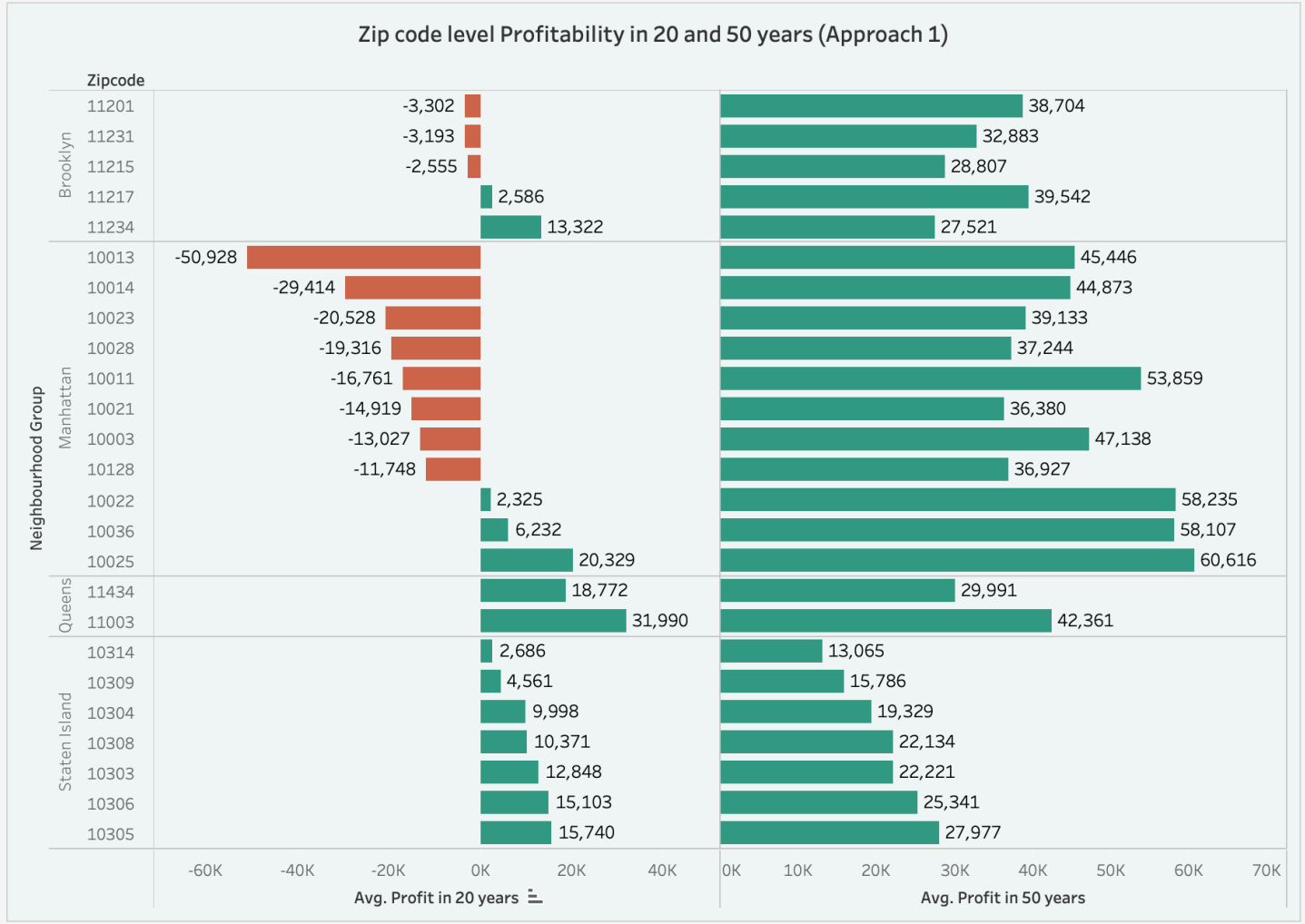
Above boxplots show the variation in per night price of property in each neighborhood group.

Note: In this boxplot, 5 outliers for Manhattan neighborhood group priced at $1,999, $2,000, $2,500, $3,750 and $4,000 have been removed.

Key Insights from above visualization: -

- Average per night property price is highest in Manhattan at $357, followed by $210 in Brooklyn, $133 in Queens and $116 in Staten Island.
- This trend is consistent with neighborhood group wise average cost of properties in last visualization.
- In each year, average cost of properties is highest in Manhattan followed by Brooklyn, Staten Island and then Queens.
- Properties in Manhattan show highest variation with price ranging to ($50-$1799)/Night, with lots of outliers.

3.6 Zip code level Profitability in 20 and 50 years (Approach 1)



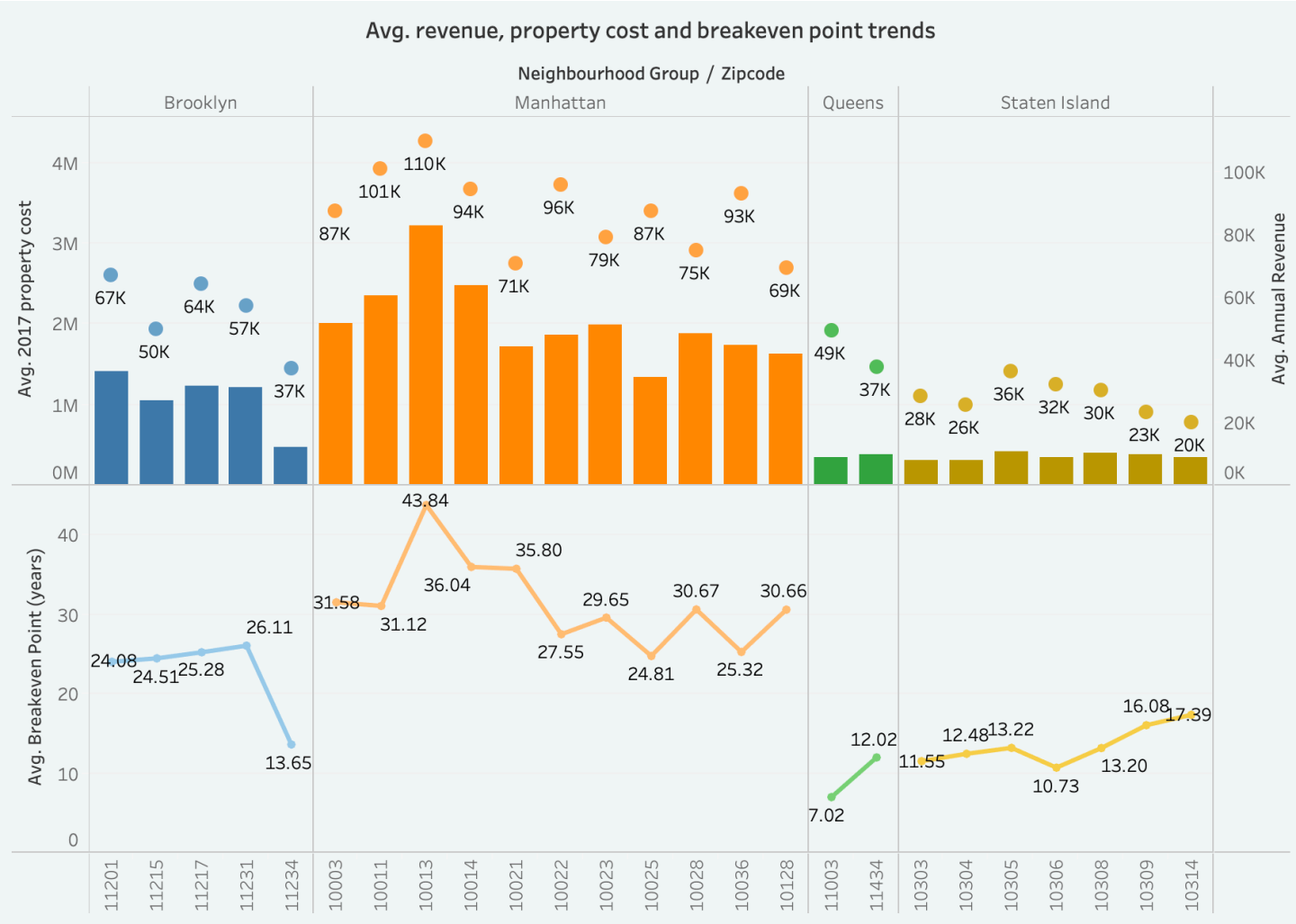
This chart shows whether in long-term (20 years and 50 years) it is profitable to buy property in a particular zip code or not. Bars in green show the avg. annual profit while the bars in red show the avg. annual loss occurred in long term.

Assumption: In this chart I am assuming that company plans to sell all its properties after 20 years (left plot) and 50 years (right plot).

Key Insights from above visualization: -

- ➔ If company has a 20 years long-term plan, it is profitable to buy property in all zip codes of Staten Island and Queens. This is mainly because the properties in these 2 neighborhoods are comparatively cheaper and it will quicker to reach breakeven point in their case. The only profitable zip codes in Brooklyn are 11217 & 11234 and the only profitable zip codes in Manhattan are 10022, 10036 & 10025.
- ➔ If company has a 50 years long-term plan, then all the zip codes are profitable. In that scenario, most profitable zip codes would be mainly in Manhattan like 10025, 10036, 10022 and 10011.

3.6 Average. revenue, property cost and breakeven point trends



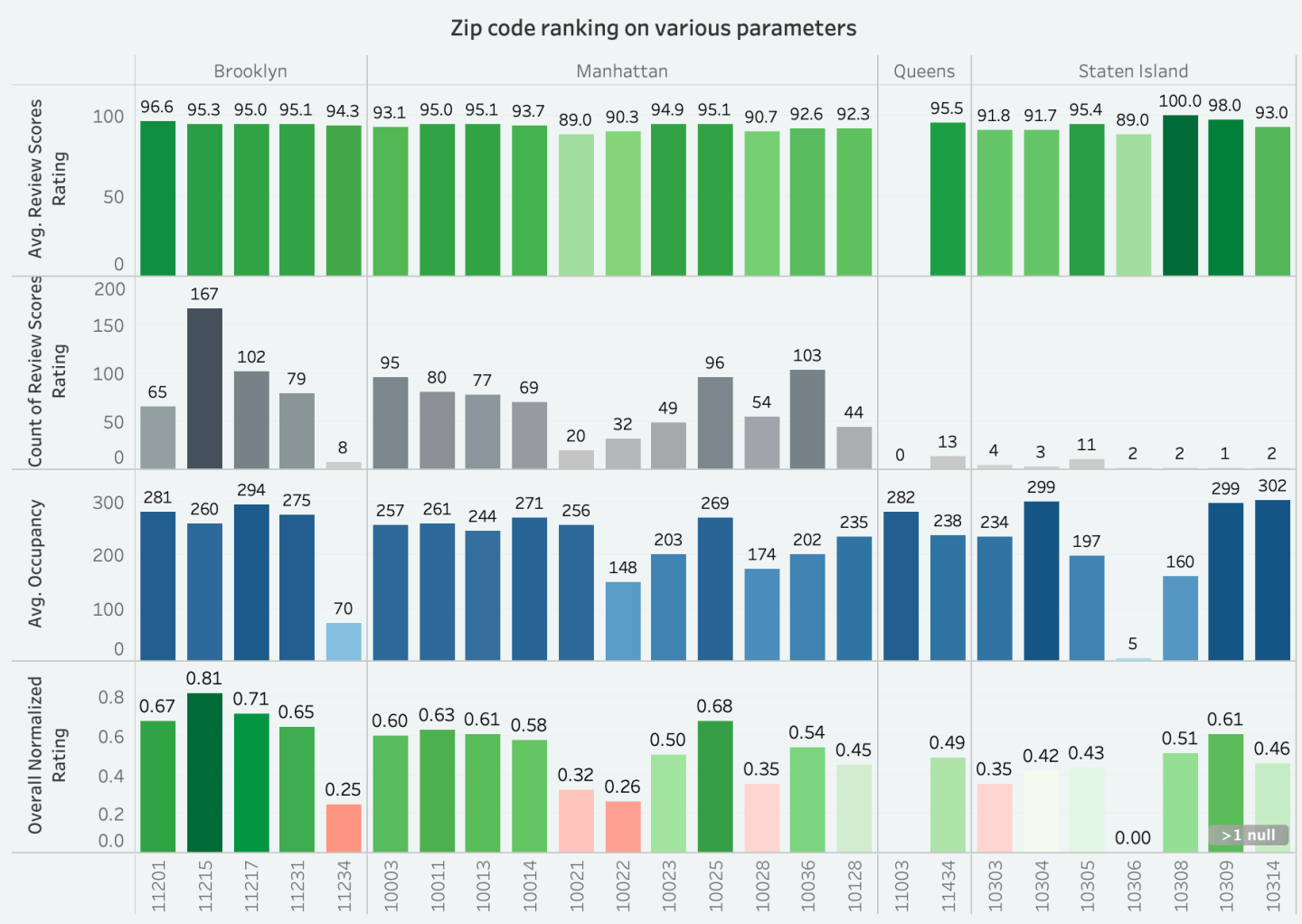
Above graph shows zip code level average revenue, average property cost and breakeven periods.

Revenue = price per night * number of days * occupancy rate (assuming given occupancy rate = 0.75)
Cost = Average price of property in 2017
Breakeven Period = Cost/ Revenue

Key Insights from above visualization: -

- ➔ The zip code wise average revenue, average property cost and breakeven periods have almost the similar trends. The average revenue and average property cost graph patterns are consistent with previous visualizations.
- ➔ Breakeven period is highest for Manhattan neighborhood properties, with highest breakeven period being as high as 43.84 years for zip code 10013. The lowest breakeven period is for 11003 zip code properties of Queens.

3.7 Zip code ranking on various parameters



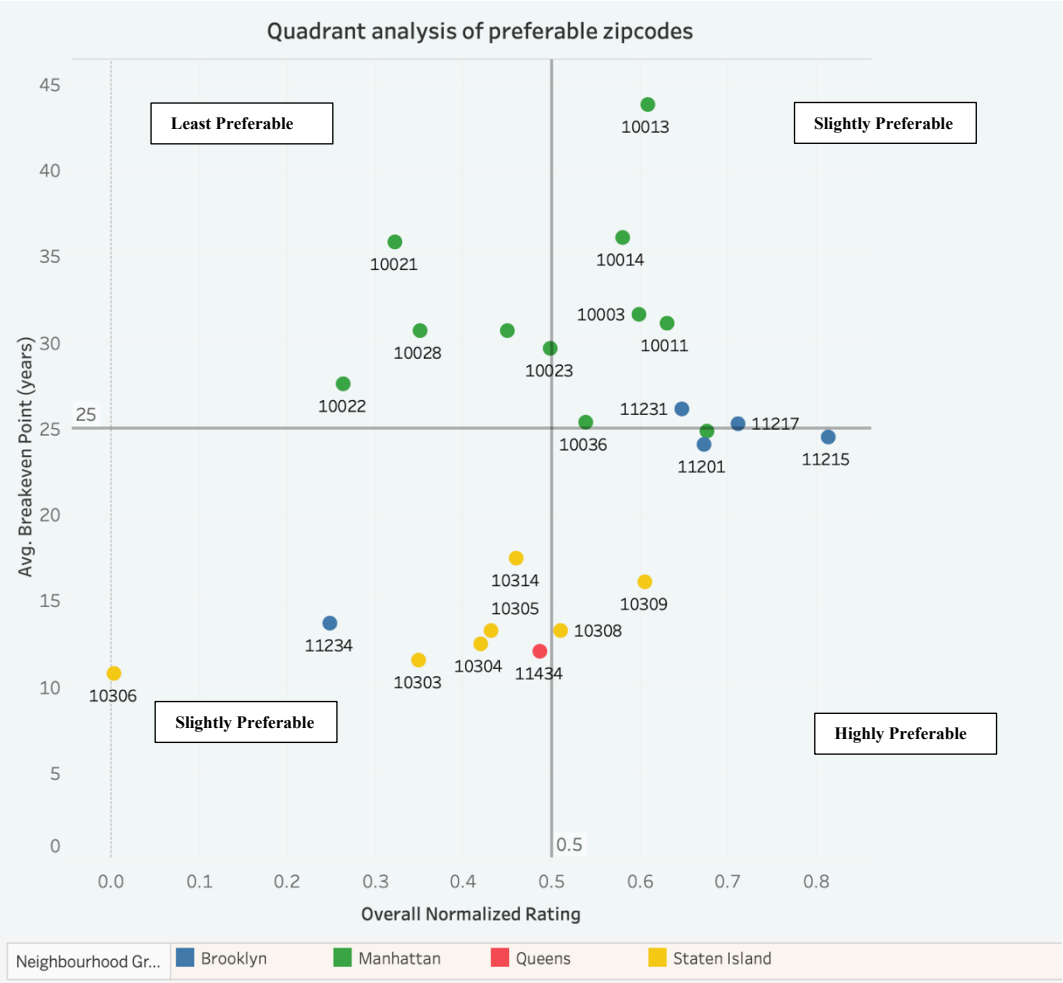
This chart highlights the ranking of all zip codes on various property parameters.

Key Insights from above visualization: -

- (Graph a) Properties of zip code 10308 in Staten Island have highest **Review Scores Rating** (100/100). This rating value (based on accuracy, cleanliness, check-in, communication, location, and value) ranges from 100 to 89 and there is no rating data available for zip code 11003.
- (Graph b) **Number of review rating scores** posted by customers is one of the important factors determining the popularity of properties in a zip code. Most popular property zip code for this parameter is 11215 followed by 10036, 11217 and 10025.
- (Graph c) Average **Occupancy** indicates the number of days the property is occupied within 365 days. The graph shows that on an average property of zip code 10314 in Staten Island is occupied for the highest number of days (302), followed by zip code 10309 and 10304 (299 days).
- (Graph d) This graph is based new parameter named '**Overall Normalized Rating**' which combines normalized values of above 3 parameters and assign them equal weights. Solely based on this parameter, zip codes 11215, 11217, 11201 and 10025 are the best for investment.

Overall Normalized Rating = 1/3 * Normalized Review Scores Rating + 1/3 * Normalized Occupancy Rate + 1/3 * Normalized Count of Review Scores Rating

3.9 Quadrant analysis of preferable zip codes and recommendations



This quadrant analysis takes into consideration Breakeven Period and Overall Normalized Rating (computed in previous visualization) and distinguishes whether it is preferable to invest in a property zip code or not.

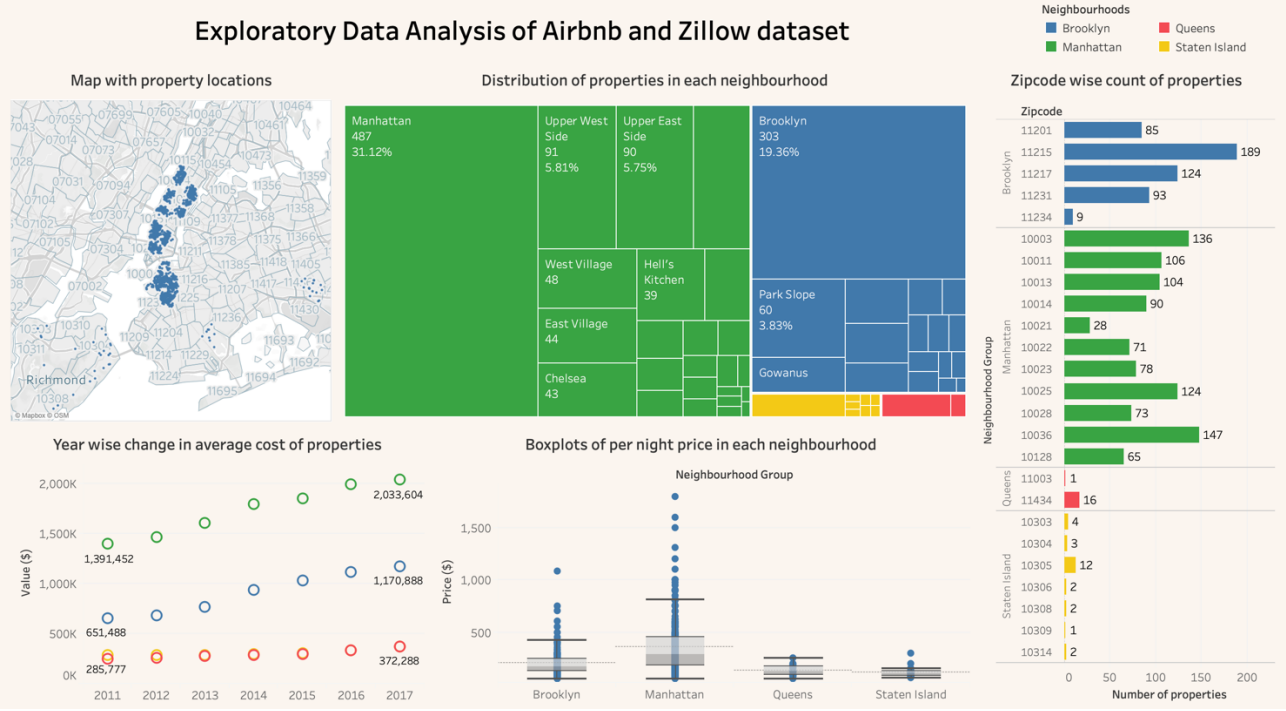
Conclusions from above visualization: -

- ➔ Zip codes 11215 & 11201 of Brooklyn, 10025 of Manhattan and 10308 & 10309 of Staten Island are the most preferable for investment as these zip codes not only have least Breakeven Periods but also highest Overall Normalized Rating.
- ➔ The other zip codes company can plan to invest in are 11217, 11231, 10036, 10314 and 11434 because these lie on the boundary of most preferable zip codes.
- ➔ The least preferable zip codes in which company must not invest are 10022, 10028, 10021 10023 and 10128.

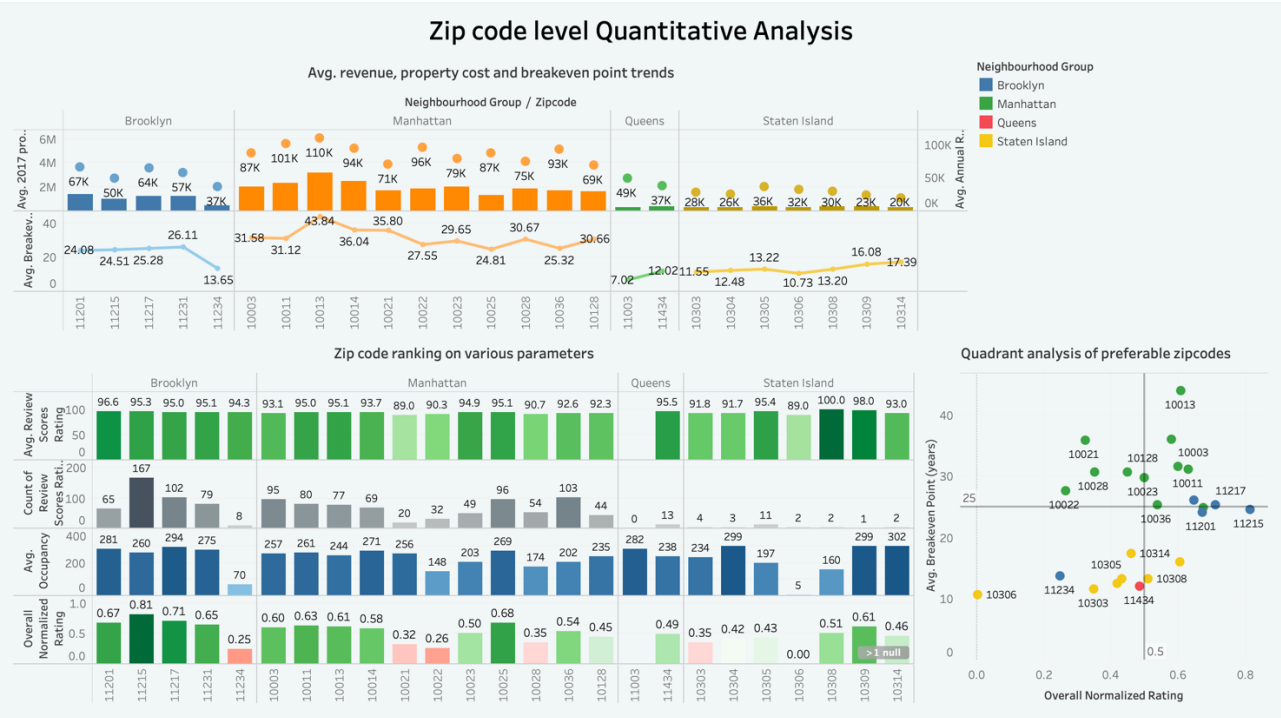
3.10 Final Dashboards

Based on all previous data visualizations, I have designed two dashboards 'Exploratory Data Analysis of Airbnb and Zillow dataset' and 'Zip code level Quantitative Analysis' which can help company to decide and track which zip codes in New York city are best for investment.

Dashboard 1:



Dashboard 2:



Further Steps

Lastly, there are many things that can be done to further this project.

- The current data set is just limited to New York County and 2 bedroom set apartments. It can be extended to remaining counties of NYC and no. of bedrooms.
- Zillow Cost data set is updated only till mid of 2017, it can be augmented with latest data or time series forecasting can be applied for it.
- We have taken 0% discount rate in our assumption, but it is not the case in real life. Necessary calculation changes can be made using its actual value.
- Text Analytics can be applied on description column of Airbnb listing data and that can give us better insights regarding properties, hosts, neighborhoods etc.
- Ratings and reviews data from google reviews and other external sources can be used to augment our dataset for better results.