NYC Property Sales

Omkar Oka

01/09/2021

Contents

1 Executive Summary:

New York City is repeatedly named as among the most expensive cities in the world to buy real estate. CNBC 15's recent article on the most expensive places in the US to buy a home included three neighborhoods from NYC. With the NYC Property Sales Dataset, the New York City Department of Finance opened up its real estate market for analysis.

The Dataset contains information of all the property sales in NYC from September 1, 2016 to August 31, 2017. With such a recent data set, I was able to analyze trends about NYC real estate market borough and neighborhood-wise.

Analysis Methodology:

My analysis of the NYC Real estate market is broken down into the following sections.

Exploratory Data Analysis - Variables such as Borough, Neighborhood, Age of the building/property, Size of property and type of building are the important ones that I explored majorly. The descriptive statistics section speaks about the distribution of each variable and makes necessary changes for the analysis. I've also tried to isolate any potential outliers for the variable that will need special attention.

Visualization of results - The visualization section is split into a few broad categories for need of clarity. The tabs within the viz section explore one variable at a time with respect to the important numerical fields.

Most In-Demand Borough - Where did New Yorkers buy their properties last year?
Most In-Demand Neighborhood - Which neighborhood do New Yorkers prefer?
The Hottest Buildings - What kind of properties do they buy?
Property sizes in NYC/ Square footage - Does more money mean larger properties in NYC?
Age of the buildings in NYC - Does more money mean newer properties in NYC?
Visualizing each of these plots showed me that many of these variables will be an important predictor of Sales Price, forming the basis for a potential predictive modeling.

We will attempt to predict the prices using Linear Regression and document the results with future plans and recommendations.

2 Data Preparation:

The NYC Property Sales Dataset is a record of every building or apartment unit that was sold in the NYC Property market over a 12 month period.

The dataset was downloaded from Kaggle. As this data set is a relatively cleaned- up version of the original NYC Department of Finance's dataset it has been included as an attachment with the file submitted as part of this project.

Loading all the required libraries and the data frame from the CSV file.

```
if(!require(tidyverse))
  install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret))
  install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table))
  install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(anytime))
  install.packages("anytime", repos = "http://cran.us.r-project.org")
if(!require(lubridate))
  install.packages("lubridate", repos = "http://cran.us.r-project.org")
if(!require(corrr))
  install.packages("corrr", repos = "http://cran.us.r-project.org")
if(!require(knitr))
  install.packages("knitr", repos = "http://cran.us.r-project.org")
if(!require(corrplot))
  install.packages("corrplot", repos = "http://cran.us.r-project.org")
if(!require(randomForest))
  install.packages("randomForest", repos = "http://cran.us.r-project.org")

#Loading Libraries:
library("stringr")
library("tidyverse")
library("caret")
library("anytime")
library("lubridate")
library("corrr")
library("knitr")
library("corrplot")
library("randomForest")
options("scipen" = 10)

nyc <-
  as_data_frame(fread("C:/Users/omkar.oka/Desktop/DataScience/NYC House/nyc-rolling-sales.csv"))
##Replace the path with your local computer path while executing the code.
```

The NYC Property Sales Data has 84548 observations and 22 variables. It has property sales data of each of the 5 boroughs in NYC - Manhattan, the Bronx, Queens, Brookyln and Staten Island.

```
str(nyc)
```

```
## tibble [84,548 x 22] (S3: tbl_df/tbl/data.frame)
##  $ V1                            : int [1:84548] 4 5 6 7 8 9 10 11 12 13 ...
##  $ BOROUGH                       : int [1:84548] 1 1 1 1 1 1 1 1 1 1 ...
##  $ NEIGHBORHOOD                  : chr [1:84548] "ALPHABET CITY" "ALPHABET CITY" "ALPHABET CITY" "ALPHABET CITY" ...
##  $ BUILDING CLASS CATEGORY       : chr [1:84548] "07 RENTALS - WALKUP APARTMENTS" "07 RENTALS - WALKUP APARTMENTS" "07 RENTALS - WALKUP APARTMENT
##  $ TAX CLASS AT PRESENT          : chr [1:84548] "2A" "2" "2" "2B" ...
##  $ BLOCK                         : int [1:84548] 392 399 399 402 404 405 406 407 379 387 ...
##  $ LOT                           : int [1:84548] 6 26 39 21 55 16 32 18 34 153 ...
##  $ EASE-MENT                     : logi [1:84548] NA NA NA NA NA NA ...
##  $ BUILDING CLASS AT PRESENT     : chr [1:84548] "C2" "C7" "C7" "C4" ...
##  $ ADDRESS                       : chr [1:84548] "153 AVENUE B" "234 EAST 4TH   STREET" "197 EAST 3RD   STREET" "154 EAST 7TH STREET" ...
##  $ APARTMENT NUMBER              : chr [1:84548] "" "" "" "" ...
##  $ ZIP CODE                      : int [1:84548] 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 ...
##  $ RESIDENTIAL UNITS             : int [1:84548] 5 28 16 10 6 20 8 44 15 24 ...
##  $ COMMERCIAL UNITS              : int [1:84548] 0 3 1 0 0 0 0 2 0 0 ...
##  $ TOTAL UNITS                   : int [1:84548] 5 31 17 10 6 20 8 46 15 24 ...
##  $ LAND SQUARE FEET              : chr [1:84548] "1633" "4616" "2212" "2272" ...
##  $ GROSS SQUARE FEET             : chr [1:84548] "6440" "18690" "7803" "6794" ...
##  $ YEAR BUILT                    : int [1:84548] 1900 1900 1900 1913 1900 1900 1920 1900 1920 1920 ...
##  $ TAX CLASS AT TIME OF SALE     : int [1:84548] 2 2 2 2 2 2 2 2 2 2 ...
##  $ BUILDING CLASS AT TIME OF SALE: chr [1:84548] "C2" "C7" "C7" "C4" ...
##  $ SALE PRICE                    : chr [1:84548] "6625000" "-" "-" "3936272" ...
##  $ SALE DATE                     : chr [1:84548] "2017-07-19 00:00:00" "2016-12-14 00:00:00" "2016-12-09 00:00:00" "2016-09-23 00:00:00" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
nyc <- nyc %>% select(-V1) ##Removing Row ID column V1
dup <- nyc %>% filter(duplicated(nyc) == TRUE) %>% nrow()
```

Total number of duplicate rows: 765

2.1 Data Cleanup

Based on the number of duplicates its safe to assume that they are just duplicate entries and hold no significance. Hence we can drop these rows.
Also we will split the `SALE DATE` into date and time components and just drop the time component, since time does not affect the price of the property. On further analysis we can see that the `EASE-MENT` column has no data and can be dropped.

2.1.1 Cleaned up Data

```
## tibble [83,783 x 21] (S3: tbl_df/tbl/data.frame)
##  $ BOROUGH                       : int [1:83783] 1 1 1 1 1 1 1 1 1 1 ...
##  $ NEIGHBORHOOD                  : chr [1:83783] "ALPHABET CITY" "ALPHABET CITY" "ALPHABET CITY" "ALPHABET CITY" ...
##  $ BUILDING CLASS CATEGORY NUMBER: chr [1:83783] "07 " "07 " "07 " "07 " ...
##  $ BUILDING CLASS CATEGORY       : chr [1:83783] "RENTALS - WALKUP APARTMENTS" "RENTALS - WALKUP APARTMENTS" "RENTALS - WALKUP APARTMENTS" "RENTA
##  $ TAX CLASS AT PRESENT          : chr [1:83783] "2A" "2" "2" "2B" ...
##  $ BLOCK                         : int [1:83783] 392 399 399 402 404 405 406 407 379 387 ...
##  $ LOT                           : int [1:83783] 6 26 39 21 55 16 32 18 34 153 ...
##  $ BUILDING CLASS AT PRESENT     : chr [1:83783] "C2" "C7" "C7" "C4" ...
##  $ ADDRESS                       : chr [1:83783] "153 AVENUE B" "234 EAST 4TH   STREET" "197 EAST 3RD   STREET" "154 EAST 7TH STREET" ...
##  $ APARTMENT NUMBER              : chr [1:83783] "" "" "" "" ...
##  $ ZIP CODE                      : int [1:83783] 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 ...
##  $ RESIDENTIAL UNITS             : int [1:83783] 5 28 16 10 6 20 8 44 15 24 ...
##  $ COMMERCIAL UNITS              : int [1:83783] 0 3 1 0 0 0 0 2 0 0 ...
##  $ TOTAL UNITS                   : int [1:83783] 5 31 17 10 6 20 8 46 15 24 ...
##  $ LAND SQUARE FEET              : chr [1:83783] "1633" "4616" "2212" "2272" ...
##  $ GROSS SQUARE FEET             : chr [1:83783] "6440" "18690" "7803" "6794" ...
##  $ YEAR BUILT                    : int [1:83783] 1900 1900 1900 1913 1900 1900 1920 1900 1920 1920 ...
##  $ TAX CLASS AT TIME OF SALE     : int [1:83783] 2 2 2 2 2 2 2 2 2 2 ...
##  $ BUILDING CLASS AT TIME OF SALE: chr [1:83783] "C2" "C7" "C7" "C4" ...
##  $ SALE PRICE                    : chr [1:83783] "6625000" "-" "-" "3936272" ...
##  $ SALE DATE                     : chr [1:83783] "2017-07-19" "2016-12-14" "2016-12-09" "2016-09-23" ...
```

After removing the unnecessary columns, we will create a new column called `Building Age` transforming the variable, `Year Built` and `SALE DATE`. Building age is a much clearer metric to understand.

```
nyc <- nyc %>% mutate(`BUILDING AGE` = as.integer(format(as.Date(`SALE DATE`, format="%Y-%m-%d"),"%Y")) - `YEAR BUILT`)
# Creating a new column called 'Building Age' transforming the variable, 'Year Built'
```

2.2 Data Conversion

We will be converting the character and discrete numeric columns to factors. Converting the character columns that have numbers to numeric and some columns to character for further analysis.

```
## tibble [83,783 x 22] (S3: tbl_df/tbl/data.frame)
##  $ BOROUGH                       : Factor w/ 5 levels "Manhattan","Bronx",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ NEIGHBORHOOD                  : chr [1:83783] "ALPHABET CITY" "ALPHABET CITY" "ALPHABET CITY" "ALPHABET CITY" ...
##  $ BUILDING CLASS CATEGORY NUMBER: Factor w/ 47 levels "01 ","02 ","03 ",..: 7 7 7 7 7 7 7 7 8 8 ...
##  $ BUILDING CLASS CATEGORY       : Factor w/ 47 levels " CONDO-RENTALS",..: 34 34 34 34 34 34 34 34 33 33 ...
##  $ TAX CLASS AT PRESENT          : Factor w/ 11 levels "","1","1A","1B",..: 7 6 6 8 7 6 8 6 6 6 ...
##  $ BLOCK                         : chr [1:83783] "392" "399" "399" "402" ...
##  $ LOT                           : chr [1:83783] "6" "26" "39" "21" ...
##  $ BUILDING CLASS AT PRESENT     : Factor w/ 167 levels "","A0","A1","A2",..: 17 22 22 19 17 19 19 22 31 35 ...
##  $ ADDRESS                       : chr [1:83783] "153 AVENUE B" "234 EAST 4TH   STREET" "197 EAST 3RD   STREET" "154 EAST 7TH STREET" ...
##  $ APARTMENT NUMBER              : chr [1:83783] "" "" "" "" ...
##  $ ZIP CODE                      : Factor w/ 186 levels "0","10001","10002",..: 9 9 9 9 9 9 9 9 9 9 ...
##  $ RESIDENTIAL UNITS             : int [1:83783] 5 28 16 10 6 20 8 44 15 24 ...
##  $ COMMERCIAL UNITS              : int [1:83783] 0 3 1 0 0 0 0 2 0 0 ...
##  $ TOTAL UNITS                   : int [1:83783] 5 31 17 10 6 20 8 46 15 24 ...
##  $ LAND SQUARE FEET              : num [1:83783] 1633 4616 2212 2272 2369 ...
##  $ GROSS SQUARE FEET             : num [1:83783] 6440 18690 7803 6794 4615 ...
##  $ YEAR BUILT                    : num [1:83783] 1900 1900 1900 1913 1900 ...
##  $ TAX CLASS AT TIME OF SALE     : Factor w/ 4 levels "1","2","3","4": 2 2 2 2 2 2 2 2 2 2 ...
##  $ BUILDING CLASS AT TIME OF SALE: Factor w/ 166 levels "A0","A1","A2",..: 16 21 21 18 16 18 18 21 30 34 ...
##  $ SALE PRICE                    : num [1:83783] 6625000 NA NA 3936272 8000000 ...
##  $ SALE DATE                     : Date[1:83783], format: "2017-07-19" "2016-12-14" ...
##  $ BUILDING AGE                  : int [1:83783] 117 116 116 103 116 117 96 117 97 96 ...
```

2.3 Data Description

Variable Name	Data Type	Variable Description
BOROUGH	factor	Name of the borough where the property is located
NEIGHBORHOOD	character	Neighbourhood name
BUILDING CLASS CATEGORY NUMBER	factor	Building class category code to identify similar properties
BUILDING CLASS CATEGORY	factor	Building class category title to identify similar properties
TAX CLASS AT PRESENT	factor	Assigned tax class of the property - Classes 1, 2, 3 or 4
BLOCK	character	Sub-division of the borough for property location
LOT	character	Sub-division of a Tax Block for every property location
BUILDING CLASS AT PRESENT	factor	Used to describe a property’s constructive use
ADDRESS	character	Property’s street address
APARTMENT NUMBER	character	Property’s apartment number
ZIP CODE	factor	Property’s postal code
RESIDENTIAL UNITS	integer	Number of residential units at the listed property
COMMERCIAL UNITS	integer	Number of commercial units at the listed property
TOTAL UNITS	integer	Total number of units at the listed property
LAND SQUARE FEET	numeric	Land area of the property listed in square feet
GROSS SQUARE FEET	numeric	Total area of all the floors of a building
YEAR BUILT	numeric	Property’s construction year
TAX CLASS AT TIME OF SALE	factor	Assigned tax class of the property at sale
BUILDING CLASS AT TIME OF SALE	factor	Used to describe a property’s constructive use at sale
SALE PRICE	numeric	Price paid for the property
SALE DATE	Date	Date of property sale
BUILDING AGE	integer	Age of the Building

Variable Name	Data Type	Variable Description

3 Data Analysis

1. Sale Price

```
##          0%         10%         20%         30%         40%
##           1      199000      319410      420810      515000
##         50%         60%         70%         80%         90%
##      628000      753403      935000     1290000     2250000
##        100%
## 2210000000
```

```
## [1] 1125
```

Dropping records with sale price less than $1000 since these are clear anomalies / outliers based on the distribution we see above.

After dropping the records we can see that the distribution has shifted and does not have an extreme low point anymore.

```
##          0%         10%         20%         30%         40%
##        1110      220000      333240      435000      529000
##         50%         60%         70%         80%         90%
##      640000      765000      950000     1300000     2300000
##        100%
## 2210000000
```

2. Land Square Feet

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0    1400    2200    3622    3300 4252327   21000
```

This is the overall distribution:

```
##        0%       10%       20%       30%       40%       50%
##       0.0       0.0       0.0    1709.7    2000.0    2200.0
##       60%       70%       80%       90%      100%
##    2500.0    2955.0    4000.0    5000.0 4252327.0
```

We can clearly deduce that 4252327.0 is an outlier in this dataset for Land Square Feet. Assuming a ball-park value that buildings that have > 500,000 land sq footage must be Commercial Vacant lands, Store Buildings and other large properties.

3. Gross Square Feet

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0     864    1548    3380    2340 3750565   21532
```

This is the overall distribution:

```
##        0%       10%       20%       30%       40%       50%
##       0.0       0.0       0.0    1092.0    1312.0    1548.0
##       60%       70%       80%       90%      100%
##    1816.0    2150.0    2600.0    3486.3 3750565.0
```

Similar to land square foot there are outliers for gross square feet as well which are vacant lands or large properties like warehouses.

4. Residential Units, Commercial Units and Total Units

Residential Units: Summary:

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.000    0.000    1.000    1.702    1.000 1844.000
```

Distribution:

```
## [1] 1376
```

Commercial Units:
Summary:

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##    0.0000    0.0000    0.0000    0.1546    0.0000 2261.0000
```

In the entire data set, there are only 1376 buildings in which more than 5 Residential units were sold in 2016. In the entire data set, there are only 140 buildings in which more than 5 Commercial units were sold in 2016.

As explored earlier, we know that only for 794 properties, Total Units is not equal to Residential, Commercial Units. As Total Units has the least NAs, we will be using this field for further analysis.

5. Building Age

There are 4195 properties with Building age. When we remove properties that don't have a Year Built entry or Year Built = 0, we get 28 property details.

Summary for Building Age:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    51.0    76.0   204.9    96.0  2017.0
```

Summary for Year Built:

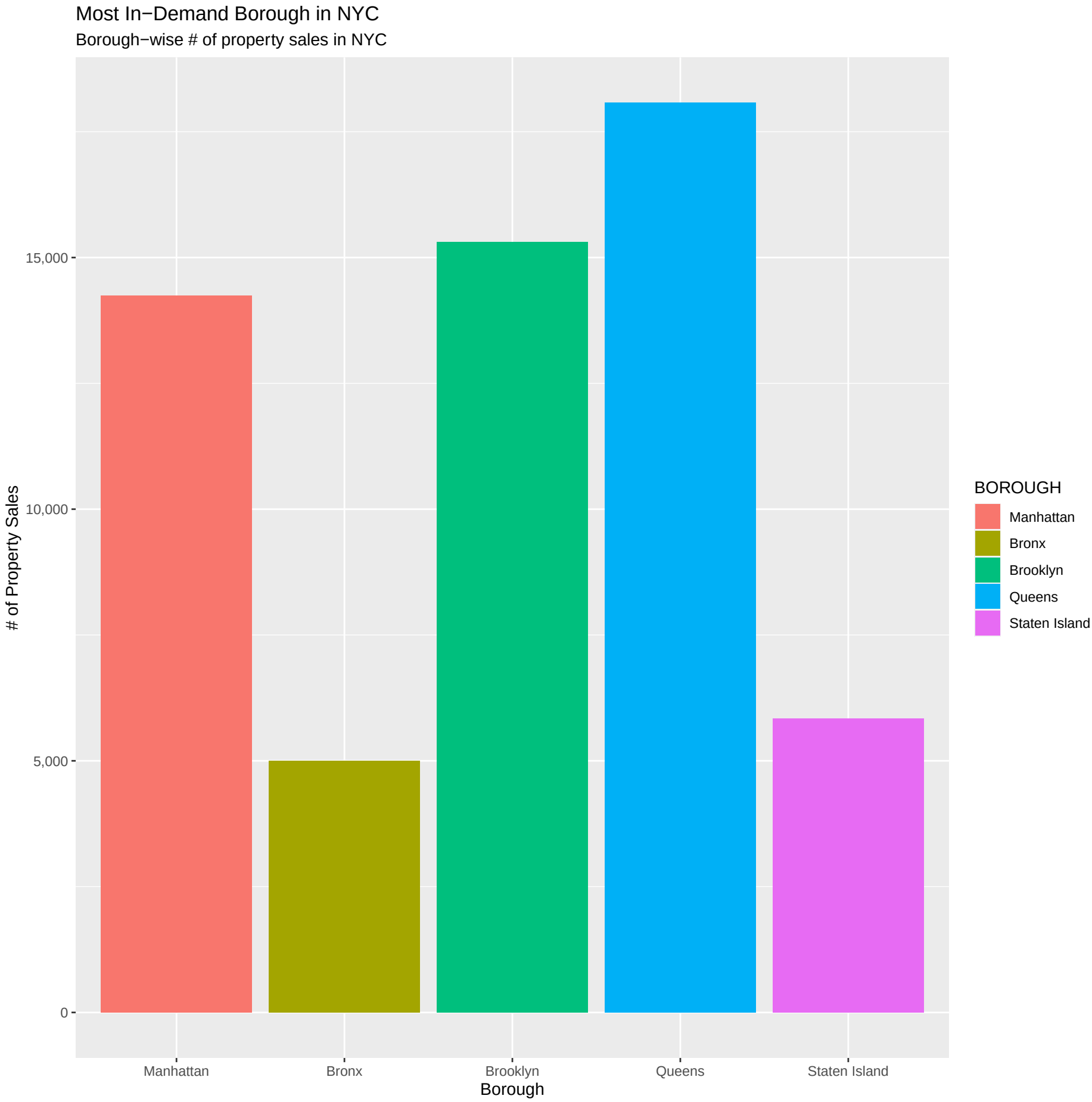
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    1920    1940    1812    1966    2017
```

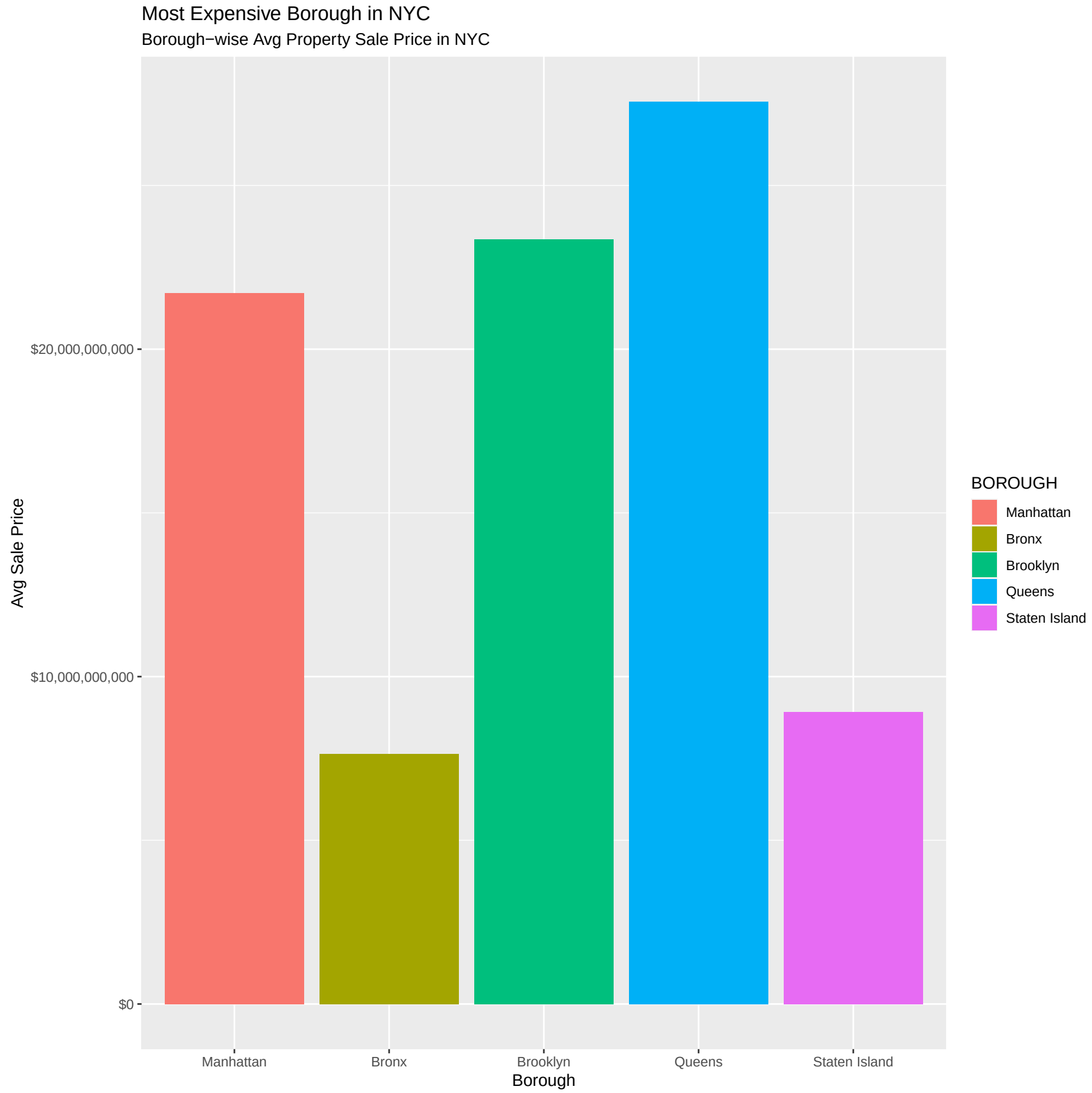
Number of Buildings more than 200 years old:

```
## [1] 28
```

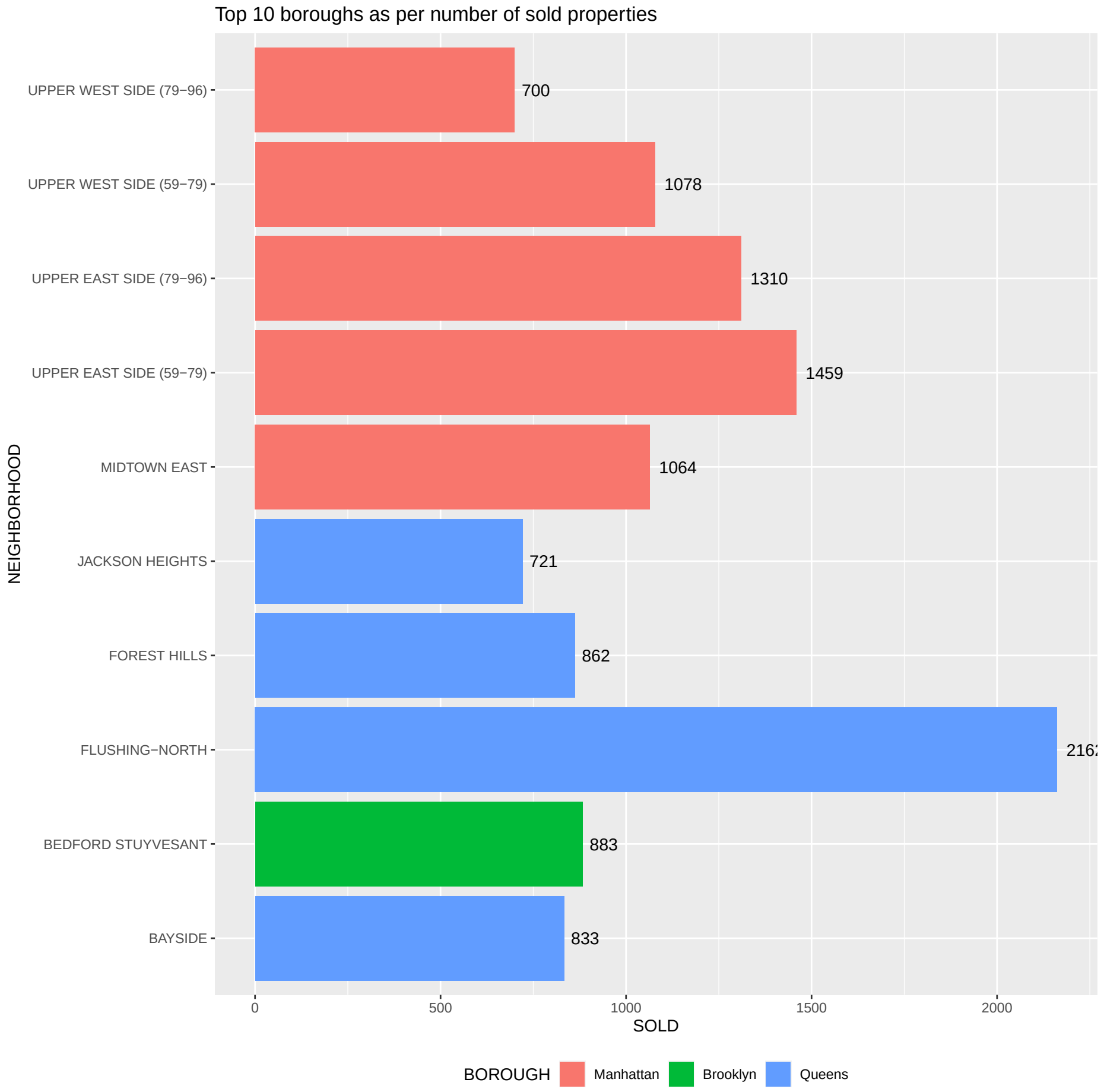
4 Data Visualization

1. NYC Boroughs:





The plots above show that Queens has the most number of property sales, followed by Brookyln. The Average Sale Price of a property in Queens was $27 billion, while in Manhattan was $22 billion. This is surprising as properties in Manhattan are expected to cost more. Let's explore this further with the other fields.

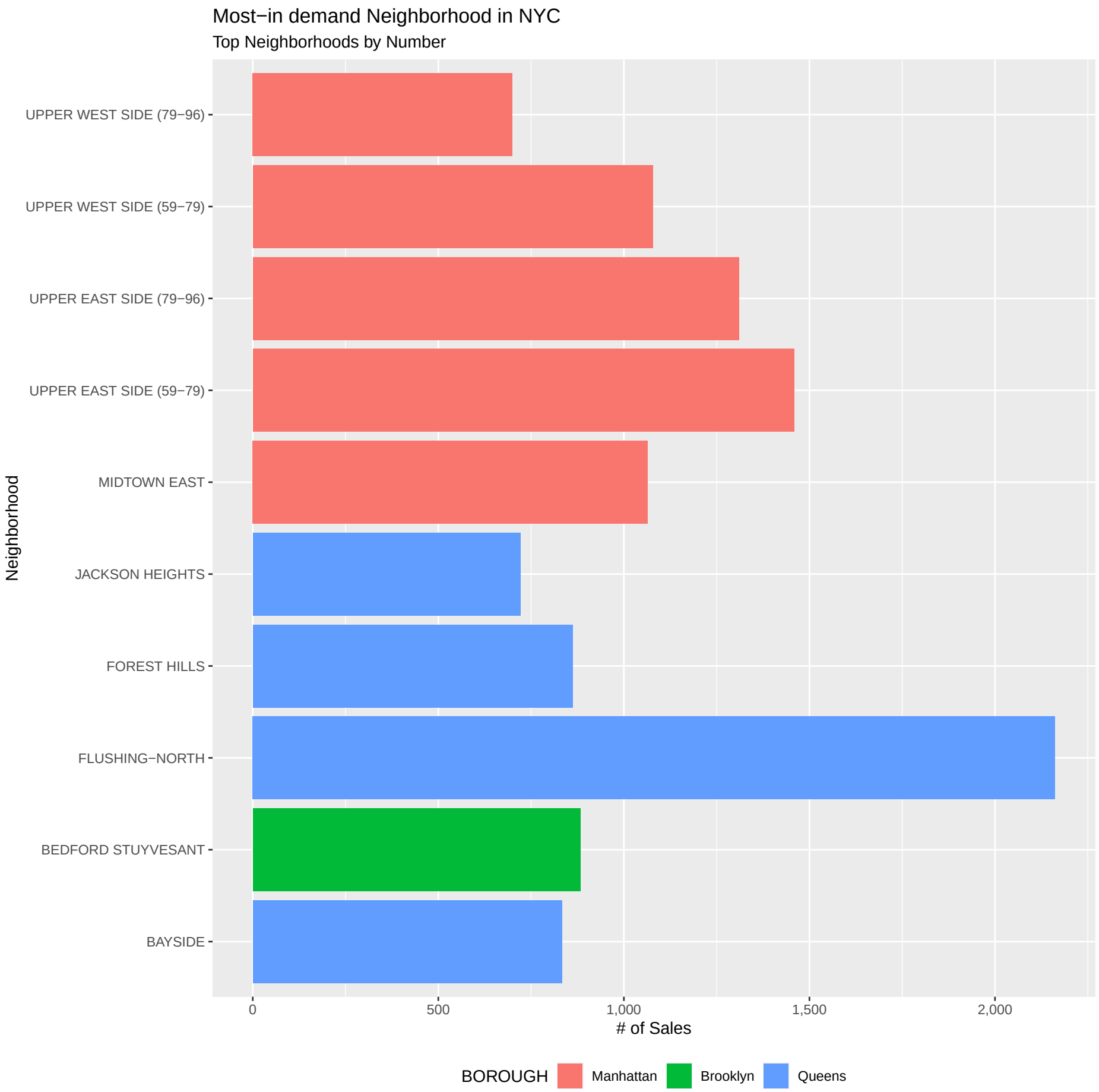


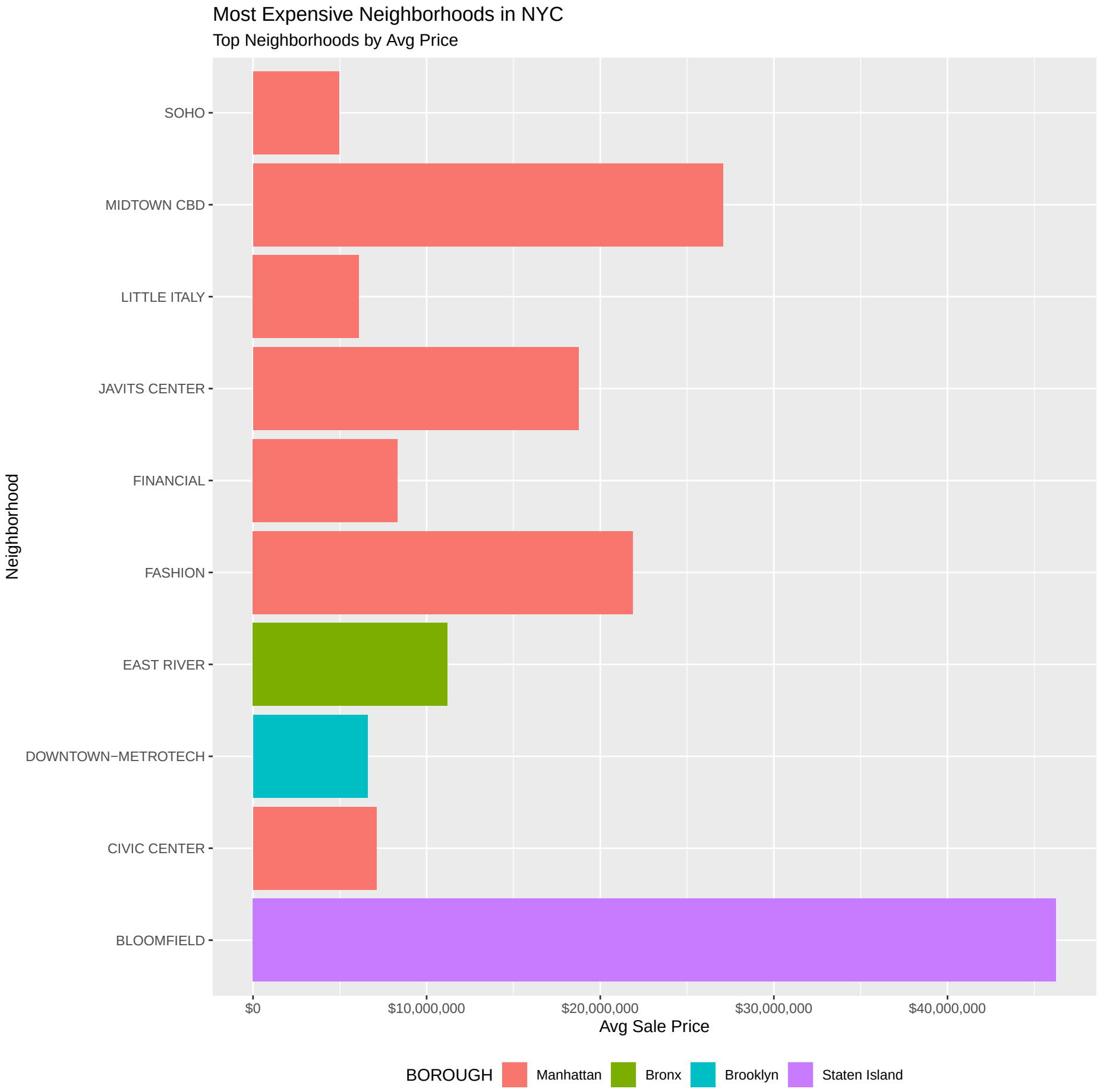
The Flushing North region of the Queens borough has the most number of units sold which might be a new neighborhood promoted by the government civilization program.

2. Neighborhoods:

With the exploration of the property sales and prices across Boroughs in NYC, lets see how the numbers divide up with respect to each Neighborhood. We can start answering this by looking at the Number of Sales and Average property Sales Prices across the most in-demand neighborhoods.

Most-in demand and Expensive Neighborhood in NYC



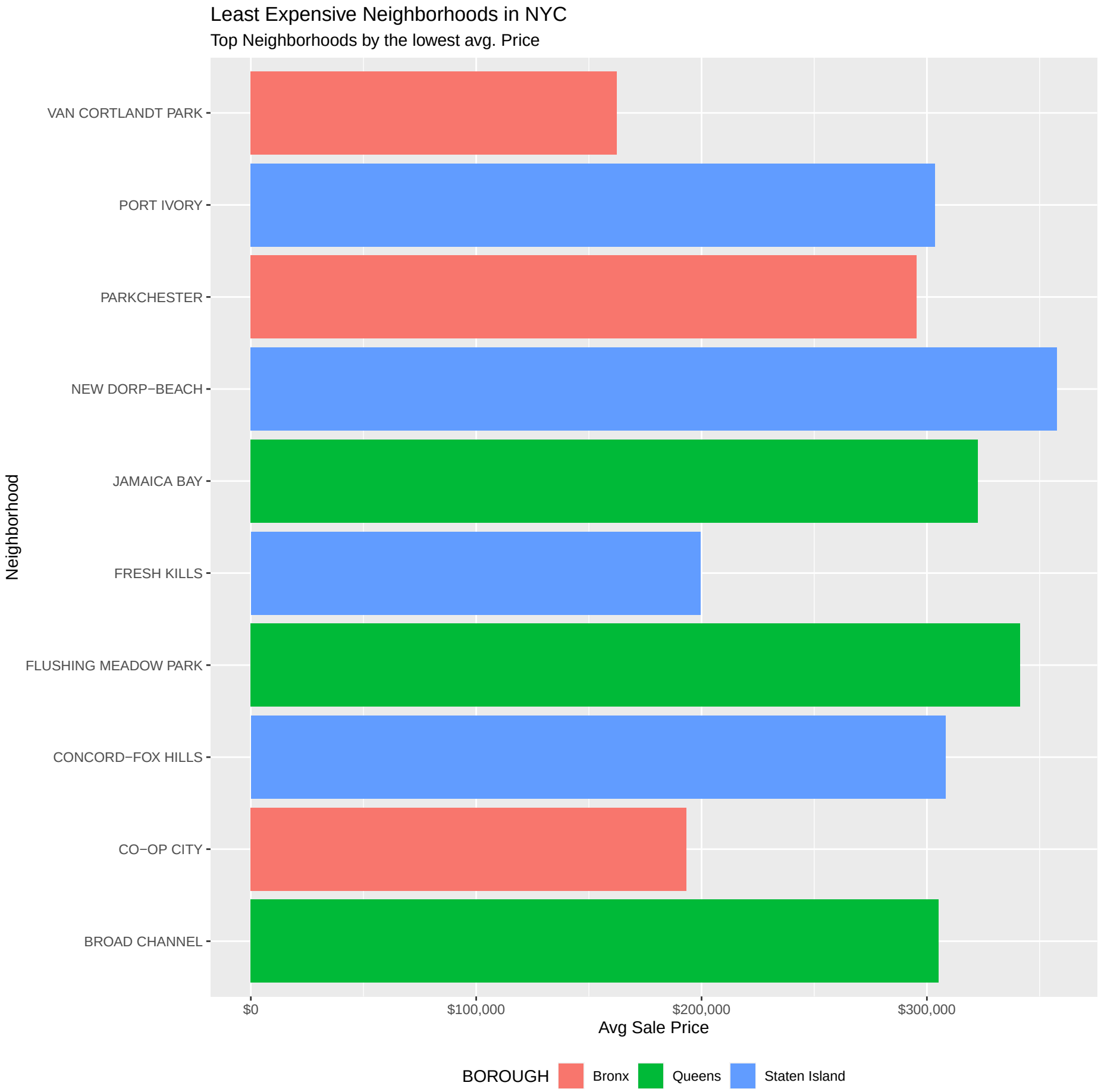


Consistent with the previous plots, the Top Neighborhoods by number of Property Sales in 2016 plot shows that neighborhoods in Queens and Manhattan had the most number of properties sold - this accounted to 7 out of the 10 top neighborhoods.

With respect to the Average value of properties sold in different neighborhoods, Bloomfield in Staten Island was the top. Staten Island, however, was among the lowest when we arranged the Average Sale Price according to Borough. So these properties must have been the top 20% of the Sale Price that we explored earlier. Also interesting to note is that 7 out the 10 top property value neighborhoods are from Manhattan. Clearly, even though no neighborhood in Queens fetched top bucks last year, it sold much more properties than the other boroughs.

Also note that the average price of the Most expensive Neighborhood and the second most expensive by $12.5 billion. Needless to say, the standard deviation of the property prices in NYC is large!

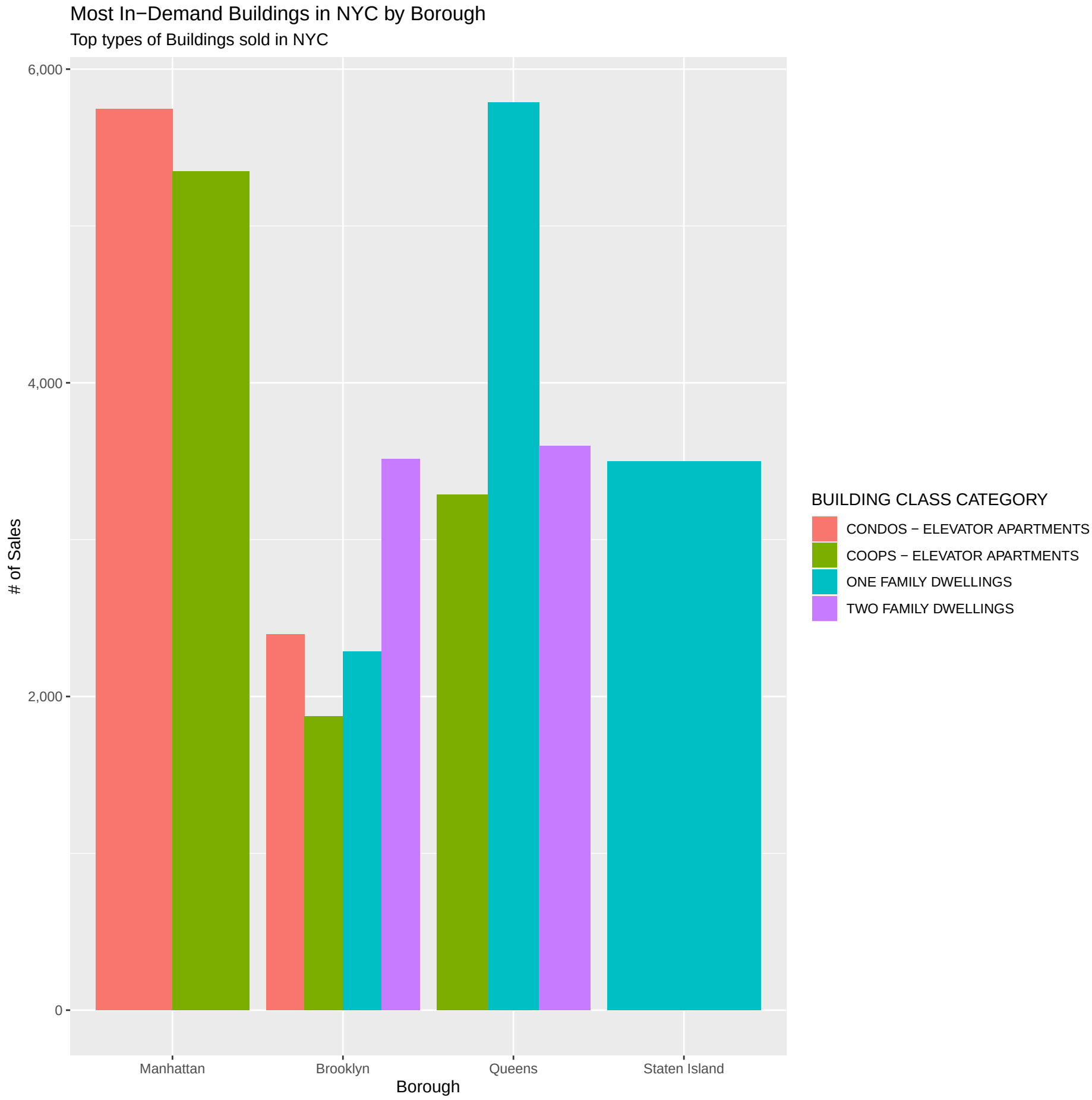
Lets check this by plotting the least expensive neighborhoods.



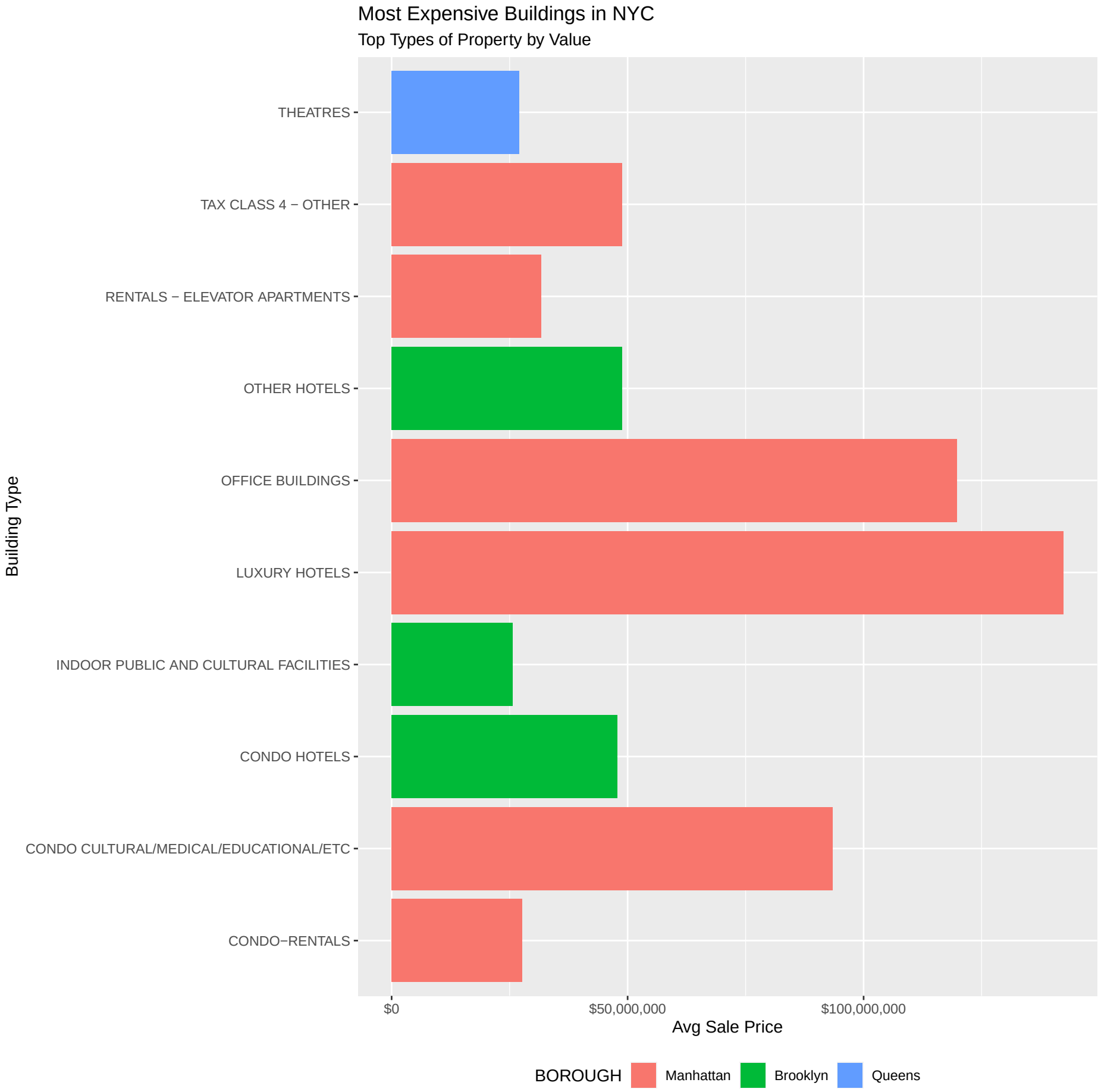
As expected, Staten Island, Queens and Bronx - the Boroughs that didn't feature in the Top Borough list made it here. Interesting to note is the differences in the Average Sale Price scale of the Most and the Least expensive properties in NYC. The average property price in Van Cortlandt Park in the Bronx sold for $160,000, while the most expensive property in Staten Island, Bloomsfield sold for 46 billion dollars.

3. Buildings:

With the knowledge of the demand and prices in neighborhoods across Boroughs, lets understand what kind of buildings get sold across NYC. This will clearly show what the hottest buildings around NYC are and their sale prices.



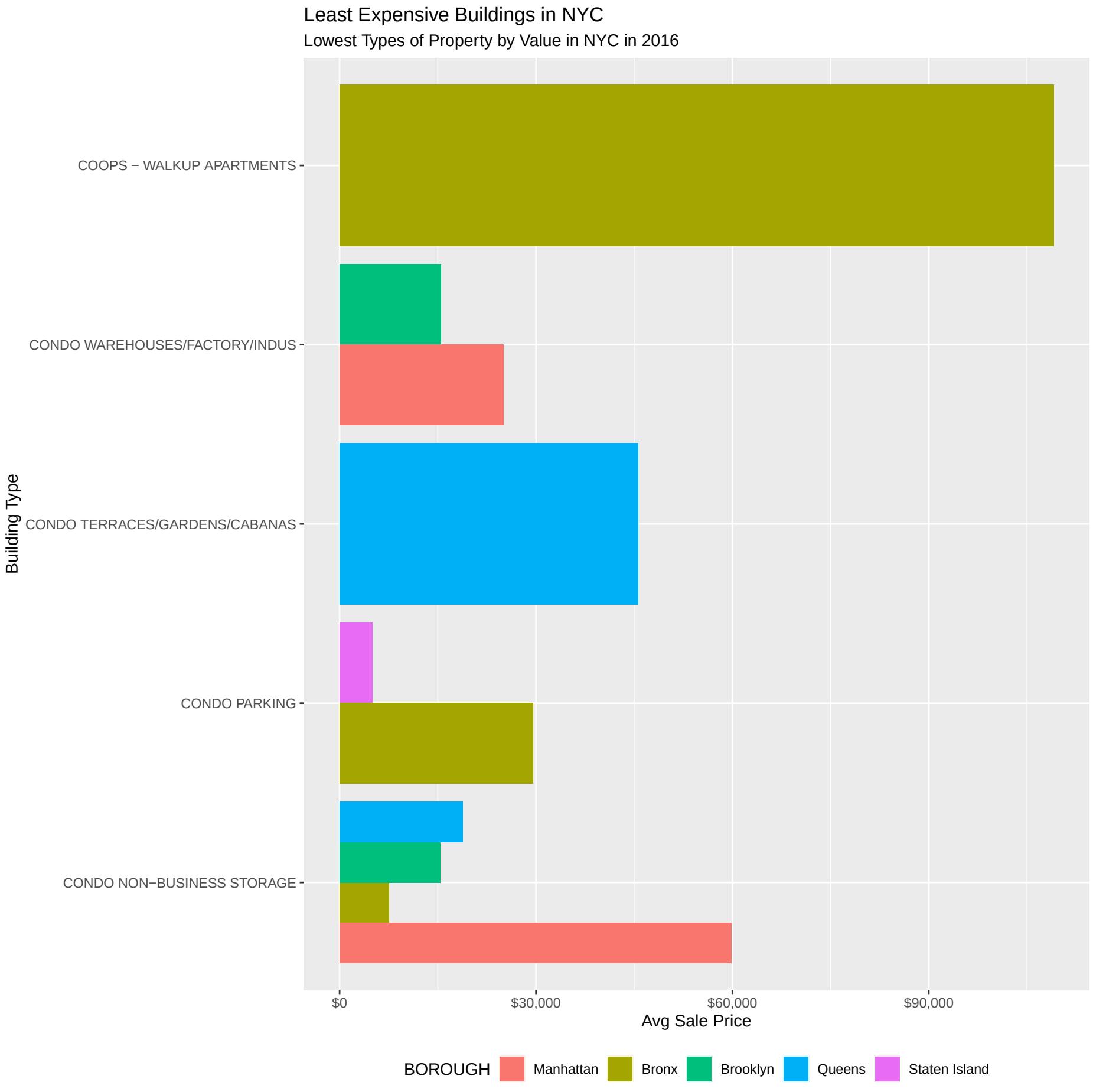
Clearly, the most in-demand buildings in NYC over the years were one family dwellings, across Staten Island, Queens and Brooklyn. Coops in Elevator Apartments were also much wanted over the last year.



Most of the top properties in Manhattan were commercial - Office buildings, Luxury Hotels, and other commercial classes, while the apartments and condos, though expensive, were on the cheaper side for Manhattan.

Another clear pattern is that the most expensive buildings are almost entirely commercial buildings. Also note how the theaters in Queens are as expensive as Rental apartments in Manhattan.

To make the property price variance argument more solid, let’s explore how the least expensive buildings in NYC look.



The least expensive property in NYC is a Condo Parking space in Staten island. Interestingly, the most expensive and the least expensive buildings in NYC are commercial buildings. For $45,000 you could also buy a Condo Terrace in the Queens!

Tax Class of the Properties sold:

Adding another variable to the equation now, let's look at the Tax Class of the Properties sold in NYC. There are 4 tax classes that Property sales are categorized into. Over the last year, there were no Tax Class, 3, property sales.

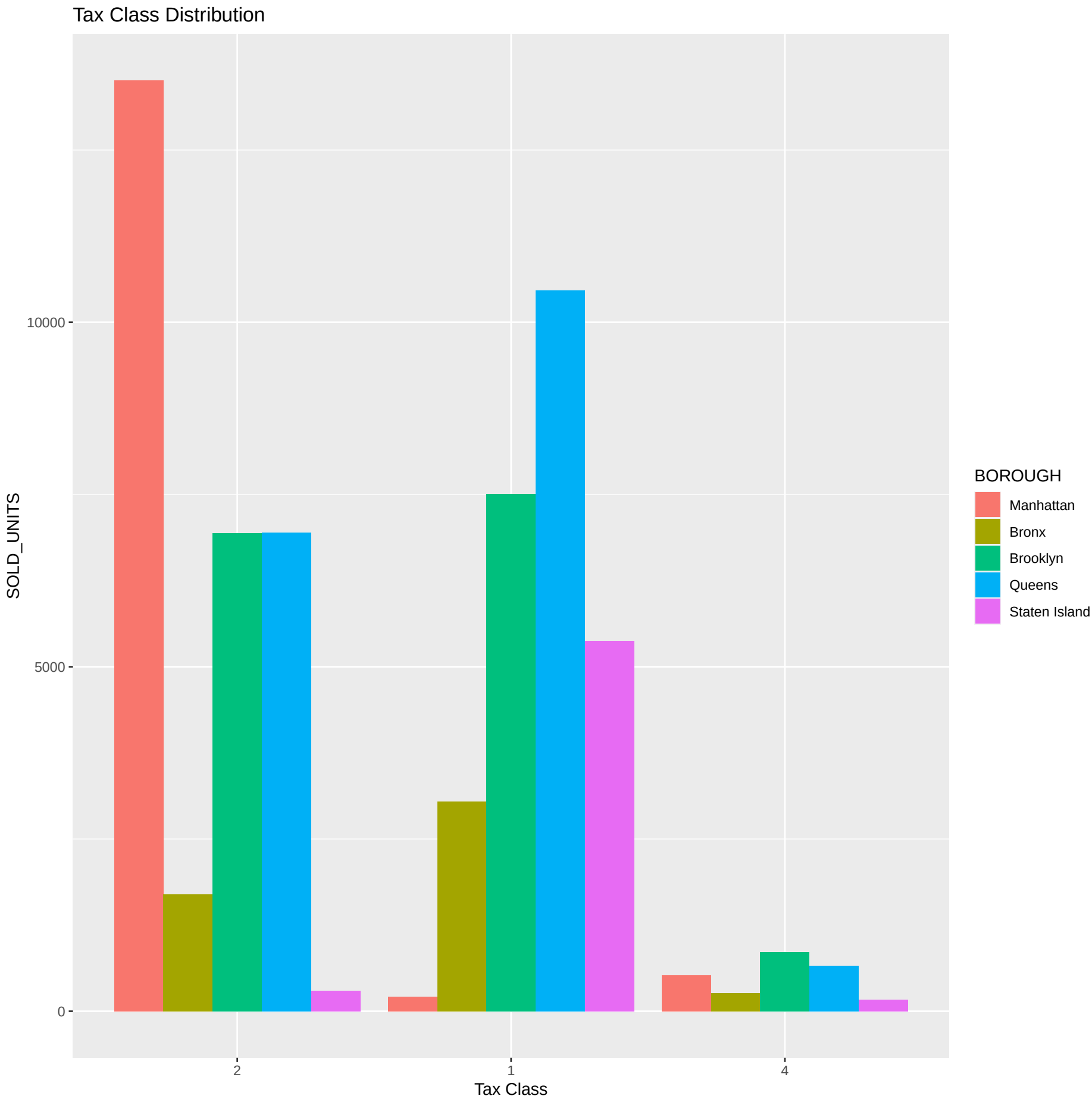
Class 1: Includes most residential property of up to three units

Class 2: Includes all other property that is primarily residential, such as cooperatives and condominiums.

Class 3: Includes property with equipment owned by a gas, telephone or electric company

Class 4: Includes all other properties not included in class 1,2, and 3, such as offices, factories, warehouses, garage buildings, etc.

Here is a distribution of units sold across all tax classes categorized by boroughs:


Tax Class Distribution
SOLD_UNITS
10000
5000
0
2
1
4
Tax Class
BOROUGH
Manhattan
Bronx
Brooklyn
Queens
Staten Island

4. Property Size: We can already guess that the tentative Price/unit area varies with Neighborhood as well. To explore this metric, lets plot Sale Price vs Land square Feet, Sale Price vs Gross square Feet borough-wise.

Sale Price vs Land Square Feet:

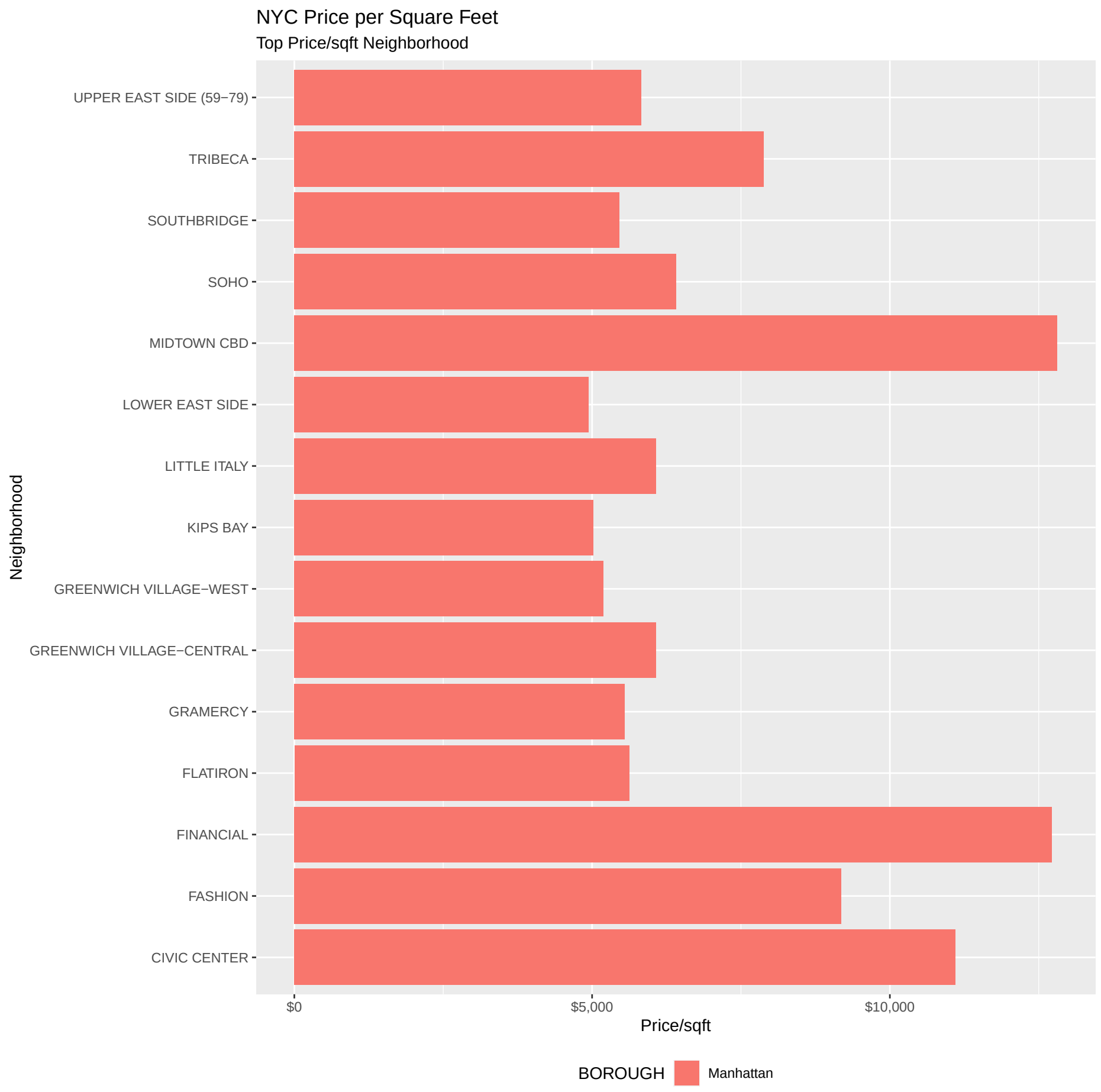
They clearly have different patterns

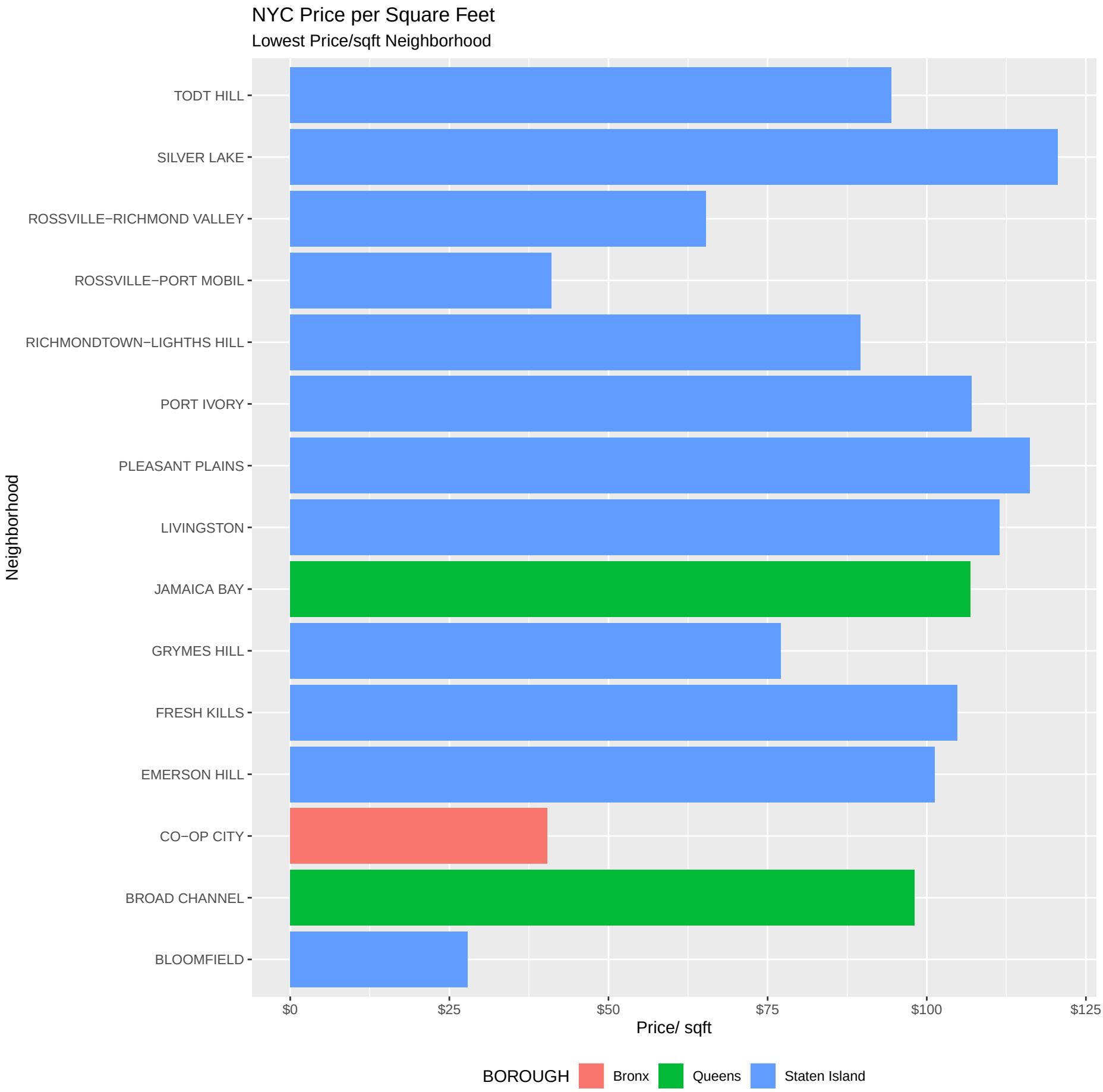
Sale Price vs Gross Square Feet:

The trends seem to largely be similar for Gross Square Footage and Land Square Footage except for the different trend in Manhattan.

To complement the charts above, we can plot a metric ‘Price/unit area’ for all the Boroughs. This will be the clearest indicator of the price of a unit sq foot of space in NYC

Price/sq. Feet in NYC:





Consistent with the analysis above, we can see all top 15 Price/sqft from Manhattan and a mojority of the Lowest Price/sqft from Staten Island. While the most expensive property in NYC sold at 16,000 dollars/sqft in Midtown CBD, the least expensive property was priced at 26 dollars/sqft.

5. Building Age:

Another important variable in the dataset is Building age. Exploring this variable will help us understand how Property prices fluctuate across Boroughs with the age of the building.

We will plot the Distribution of Building age across each borough to figure where the older buildings in NYC are and then also plot the Sale Price vs Building age across each borough to identify if building age impacts the property sale price.

The plot shows that only in Manhattan and the Bronx can we expect property prices to fall as the age of the Building increases. Building age might not even be a good predictor of Sale price for properties in the other Boroughs.

5 Data Preparation and Correlation

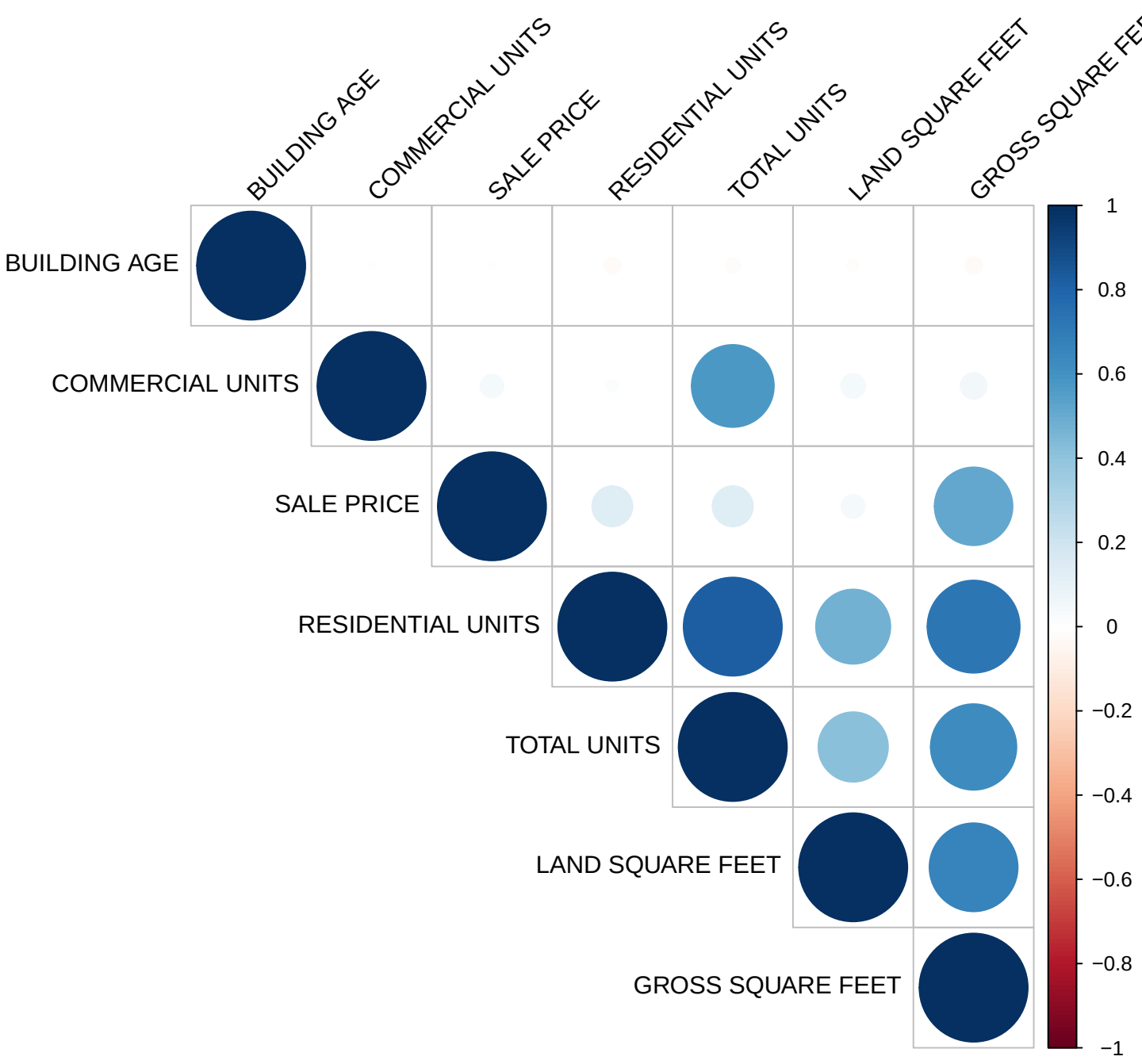
5.1 Data Preparation

To predict the NYC Property Price using this data set, we need to create a new data set for prediction removing all character values - such as Address, etc and retain only the fields that could help in prediction.

We will also transform Sale date into the month in which the sale occurred for better aesthetics.

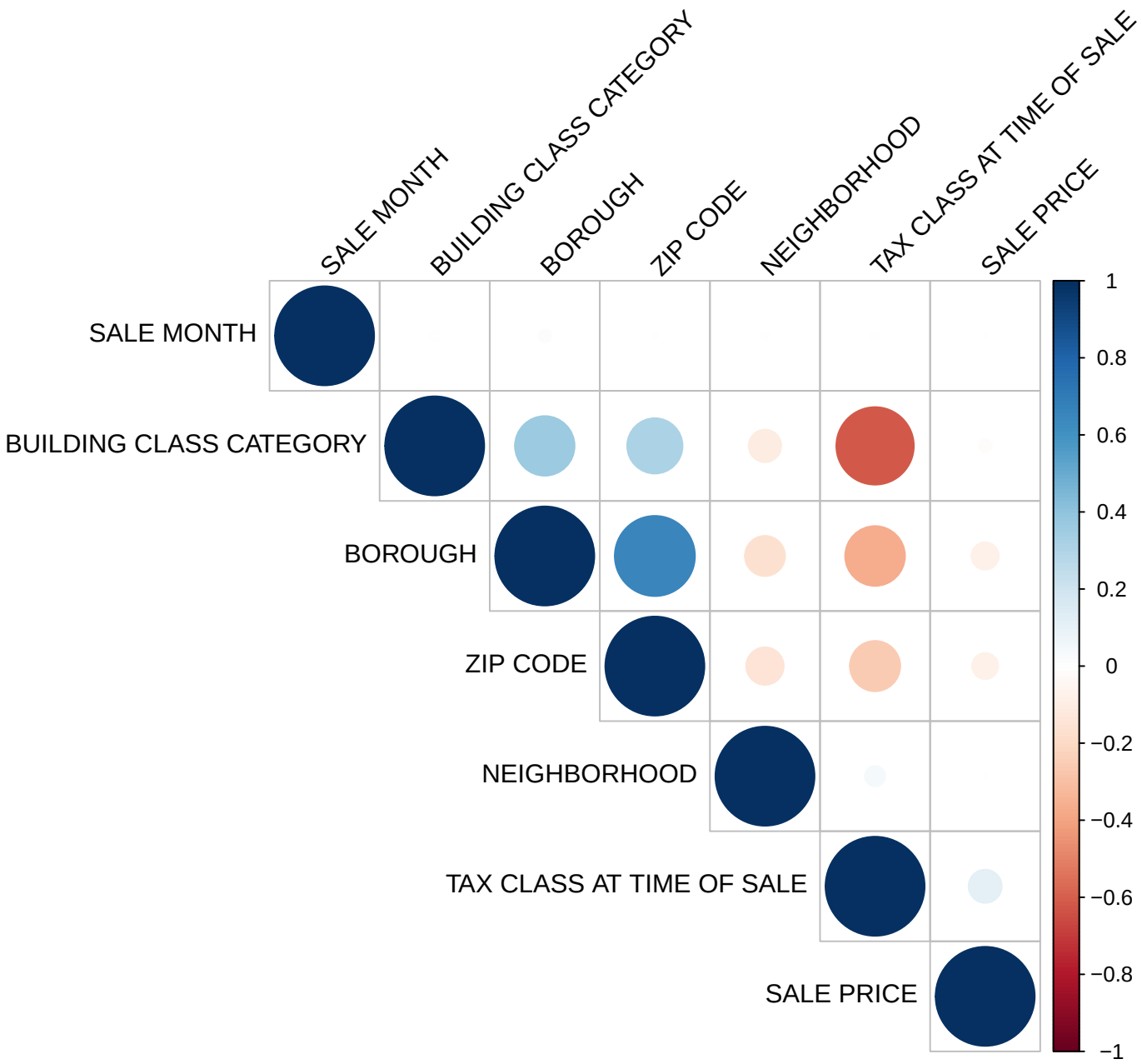
```
nyc$`SALE MONTH` <- as.factor(months(nyc$`SALE DATE`))
nyc$NEIGHBORHOOD <- as.factor(nyc$NEIGHBORHOOD)

nyc_final <- nyc[, -c(3, 5, 6, 7, 8, 9, 10, 17, 19, 21)]
nyc_final <- nyc_final[c(1:10, 12,13,11)]
```

5.2 Correlations



Clearly, Residential units - Total units - Gross Square Feet are highly correlated (> 0.7); Commercial units and Total units are also fairly correlated (0.577) Total units has high correlation with almost all variables.

Land Square Feet and Gross Square Feet are highly correlated as well (0.664). Consider using Total units, instead of Residential units and Commercial units and Land Square Feet instead of Land Square Feet and Gross Square Feet.



Checking the correlation between the categorical variables shows some clear trends. Borough and Zip Code have high correlation (0.65) and Borough Class Category, Tax class has high correlation too (0.613).

6 Predictive Analysis

6.1 Final Data Preparation

To predict the NYC Property Price using this data set, we have to start with creating a new data set for prediction removing all character values - such as Address, etc and retain only the fields that could help in prediction. We will also transform Sale date into the month in which the sale occurred to create another prediction dimension.

```
## tibble [58,470 x 13] (S3: tbl_df/tbl/data.frame)
##  $ BOROUGH                  : Factor w/ 5 levels "Manhattan","Bronx",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ NEIGHBORHOOD             : Factor w/ 253 levels "AIRPORT LA GUARDIA",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ BUILDING CLASS CATEGORY  : Factor w/ 47 levels " CONDO-RENTALS",..: 34 34 34 34 33 33 20 20 20 20 ...
##  $ ZIP CODE                 : Factor w/ 186 levels "0","10001","10002",..: 9 9 9 9 9 9 9 9 9 9 ...
##  $ RESIDENTIAL UNITS        : int [1:58470] 5 10 6 8 24 10 0 0 0 0 ...
##  $ COMMERCIAL UNITS         : int [1:58470] 0 0 0 0 0 0 0 0 0 0 ...
##  $ TOTAL UNITS              : int [1:58470] 5 10 6 8 24 10 0 0 0 0 ...
##  $ LAND SQUARE FEET         : num [1:58470] 1633 2272 2369 1750 4489 ...
##  $ GROSS SQUARE FEET        : num [1:58470] 6440 6794 4615 4226 18523 ...
##  $ TAX CLASS AT TIME OF SALE: Factor w/ 4 levels "1","2","3","4": 2 2 2 2 2 2 2 2 2 2 ...
##  $ BUILDING AGE             : int [1:58470] 117 103 116 96 96 7 97 97 97 92 ...
##  $ SALE MONTH               : Factor w/ 12 levels "April","August",..: 6 12 10 12 10 11 8 7 6 8 ...
##  $ SALE PRICE               : num [1:58470] 6625000 3936272 8000000 3192840 16232000 ...
```

To begin with the modeling exercise, we will split the data set into an 80-20% training and test set.

```
set.seed(1)
index <- sample(nrow(nyc_pred),nrow(nyc_pred)*0.80)
nyc_pred.train <- nyc_pred[index,]
nyc_pred.test <- nyc_pred[-index,]
```

6.2 Single Factor Linear Regression

Running a full model Linear regression for this data set does not seem too appropriate as there are too many categorical predictors that will create lots of dummy variables. To check if they're necessary we will use a single factor regression between the predictor and each response variable.

```
# 1. Single Factor Regression - Borough
model1 <- lm(nyc_pred.train$`SALE PRICE` ~ nyc_pred.train$BOROUGH)
res_borough <- summary(model1)

res_borough
```

```
##
## Call:
## lm(formula = nyc_pred.train$`SALE PRICE` ~ nyc_pred.train$BOROUGH)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
##   -3267200    -797710    -389808     -28353 2206731625
##
## Coefficients:
##                                   Estimate Std. Error
## (Intercept)                        3268375     113964
## nyc_pred.train$BOROUGHBronx       -2408421     224565
## nyc_pred.train$BOROUGHBrooklyn    -2012312     158484
## nyc_pred.train$BOROUGHQueens      -2512022     152560
## nyc_pred.train$BOROUGHStaten Island -2706421     211927
##                                   t value Pr(>|t|)
## (Intercept)                         28.68   <2e-16 ***
## nyc_pred.train$BOROUGHBronx        -10.72   <2e-16 ***
## nyc_pred.train$BOROUGHBrooklyn     -12.70   <2e-16 ***
## nyc_pred.train$BOROUGHQueens       -16.47   <2e-16 ***
## nyc_pred.train$BOROUGHStaten Island  -12.77   <2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12190000 on 46771 degrees of freedom
## Multiple R-squared:  0.007173,   Adjusted R-squared:  0.007088
## F-statistic: 84.48 on 4 and 46771 DF,  p-value: < 2.2e-16
```

```
pf(res_borough$fstatistic[1],res_borough$fstatistic[2],res_borough$fstatistic[3],lower.tail = FALSE)
```

```
##        value
## 1.294664e-71
```

The F statistic for Borough is significant.

```
# 2. Single Factor Regression - Neighborhood

model1 <- lm(nyc_pred.train$`SALE PRICE` ~ nyc_pred.train$NEIGHBORHOOD)
res_neigh <- summary(model1)

# F-statistic p-value
pf(res_neigh$fstatistic[1],res_neigh$fstatistic[2],res_neigh$fstatistic[3],lower.tail = FALSE)
```

```
##         value
## 5.224904e-162
```

Only 4 neighborhoods are significant. Manually creating dummies for these categories and clubbing the rest under a Neighborhood 'Others' variable.

```
nyc_pred.train$N_BLOOMFIELD = ifelse(nyc_pred.train$NEIGHBORHOOD == "BLOOMFIELD", 1,0)
nyc_pred.train$N_FASHION = ifelse(nyc_pred.train$NEIGHBORHOOD == "FASHION", 1,0)
nyc_pred.train$`N_JAVITS CENTER` = ifelse(nyc_pred.train$NEIGHBORHOOD == "JAVITS CENTER", 1,0)
nyc_pred.train$`N_MIDTOWN CBD` = ifelse(nyc_pred.train$NEIGHBORHOOD == "MIDTOWN CBD", 1,0)
nyc_pred.train$`N_OTHERS` = ifelse((nyc_pred.train$NEIGHBORHOOD != "MIDTOWN CBD") &
                                                    (nyc_pred.train$NEIGHBORHOOD != "JAVITS CENTER") &
                                                    (nyc_pred.train$NEIGHBORHOOD != "FASHION") &
                                                    (nyc_pred.train$NEIGHBORHOOD != "BLOOMFIELD"), 1,0)

# Removing the original Neighborhood predictor
nyc_pred.train <- nyc_pred.train[,-2]
```

```
# 3. Single Factor Regression - BCC
model1 <- lm(nyc_pred.train$`SALE PRICE` ~ nyc_pred.train$`BUILDING CLASS CATEGORY`)
res_bcc <- summary(model1)

# F-statistic p-value
pf(res_bcc$fstatistic[1],res_bcc$fstatistic[2],res_bcc$fstatistic[3],lower.tail = FALSE)
```

```
## value
##     0
```

```
# 4. Single Factor Regression - Tax class
model1 <- lm(nyc_pred.train$`SALE PRICE` ~ nyc_pred.train$`TAX CLASS AT TIME OF SALE`)
res_tax <- summary(model1)

res_tax
```

```
##
## Call:
## lm(formula = nyc_pred.train$`SALE PRICE` ~ nyc_pred.train$`TAX CLASS AT TIME OF SALE`)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
##   -8278836    -992264    -348327      29743 2201719164
##
## Coefficients:
##                                          Estimate
## (Intercept)                                748327
## nyc_pred.train$`TAX CLASS AT TIME OF SALE`2   848937
## nyc_pred.train$`TAX CLASS AT TIME OF SALE`4  7532509
##                                          Std. Error
## (Intercept)                                   83209
## nyc_pred.train$`TAX CLASS AT TIME OF SALE`2    114891
## nyc_pred.train$`TAX CLASS AT TIME OF SALE`4    286351
##                                          t value
## (Intercept)                                8.993
## nyc_pred.train$`TAX CLASS AT TIME OF SALE`2   7.389
## nyc_pred.train$`TAX CLASS AT TIME OF SALE`4  26.305
##                                                 Pr(>|t|)
## (Intercept)                                      < 2e-16
## nyc_pred.train$`TAX CLASS AT TIME OF SALE`2 0.00000000000015
## nyc_pred.train$`TAX CLASS AT TIME OF SALE`4         < 2e-16
##
## (Intercept)                                   ***
## nyc_pred.train$`TAX CLASS AT TIME OF SALE`2 ***
## nyc_pred.train$`TAX CLASS AT TIME OF SALE`4 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12150000 on 46773 degrees of freedom
## Multiple R-squared:  0.01465,    Adjusted R-squared:  0.01461
## F-statistic: 347.8 on 2 and 46773 DF,  p-value: < 2.2e-16
```

```
pf(res_tax$fstatistic[1],res_tax$fstatistic[2],res_tax$fstatistic[3],lower.tail = FALSE)
```

```
##         value
## 1.185745e-150
```

All the levels are significant - so maintaining them.

```
# 5. Single Factor Regression - Sale Month
model1 <- lm(nyc_pred.train$`SALE PRICE` ~ nyc_pred.train$`SALE MONTH`)
res_month <- summary(model1)

res_month
```

```
##
## Call:
## lm(formula = nyc_pred.train$`SALE PRICE` ~ nyc_pred.train$`SALE MONTH`)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
##   -2045037   -1102616    -820148    -371727 2207953853
##
## Coefficients:
##                                  Estimate Std. Error
## (Intercept)                     1354670.3   207225.1
```

```
## nyc_pred.train$`SALE MONTH`August     -63054.0   293525.5
## nyc_pred.train$`SALE MONTH`December   397056.9   280298.7
## nyc_pred.train$`SALE MONTH`February   -67678.2   293546.8
## nyc_pred.train$`SALE MONTH`January     61768.2   288377.7
## nyc_pred.train$`SALE MONTH`July        32559.8   288377.7
## nyc_pred.train$`SALE MONTH`June        87046.6   272966.2
## nyc_pred.train$`SALE MONTH`March      105894.3   281388.8
## nyc_pred.train$`SALE MONTH`May        691476.4   280740.0
## nyc_pred.train$`SALE MONTH`November   127007.7   286906.6
## nyc_pred.train$`SALE MONTH`October      -113.9   290572.7
## nyc_pred.train$`SALE MONTH`September  142774.3   279417.4
##                                     t value
## (Intercept)                           6.537
## nyc_pred.train$`SALE MONTH`August    -0.215
## nyc_pred.train$`SALE MONTH`December   1.417
## nyc_pred.train$`SALE MONTH`February  -0.231
## nyc_pred.train$`SALE MONTH`January    0.214
## nyc_pred.train$`SALE MONTH`July       0.113
## nyc_pred.train$`SALE MONTH`June       0.319
## nyc_pred.train$`SALE MONTH`March      0.376
## nyc_pred.train$`SALE MONTH`May        2.463
## nyc_pred.train$`SALE MONTH`November   0.443
## nyc_pred.train$`SALE MONTH`October    0.000
## nyc_pred.train$`SALE MONTH`September  0.511
##                                          Pr(>|t|)
## (Intercept)                          0.0000000000633 ***
## nyc_pred.train$`SALE MONTH`August           0.8299
## nyc_pred.train$`SALE MONTH`December         0.1566
## nyc_pred.train$`SALE MONTH`February         0.8177
## nyc_pred.train$`SALE MONTH`January          0.8304
## nyc_pred.train$`SALE MONTH`July             0.9101
## nyc_pred.train$`SALE MONTH`June             0.7498
## nyc_pred.train$`SALE MONTH`March            0.7067
## nyc_pred.train$`SALE MONTH`May              0.0138 *
## nyc_pred.train$`SALE MONTH`November         0.6580
## nyc_pred.train$`SALE MONTH`October          0.9997
## nyc_pred.train$`SALE MONTH`September        0.6094
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12240000 on 46764 degrees of freedom
## Multiple R-squared:  0.0002945,  Adjusted R-squared:  5.933e-05
## F-statistic: 1.252 on 11 and 46764 DF,  p-value: 0.2457
```

```
pf(res_month$fstatistic[1],res_month$fstatistic[2],res_month$fstatistic[3],lower.tail = FALSE)
```

```
##     value
## 0.2457146
```

None of the levels of the month variable are significant.

6.3 Multi Factor Linear Regression

Running full model based on the variables reduced from the above step.

```
model <- lm(`SALE PRICE` ~ .  -`SALE MONTH`, data = nyc_pred.train)
```

Though the Adjusted R-squared value is quite high and the Model MSE is very large. This is likely due to the large degrees of freedom in the model. **Linear Regression is not the right predictor for this data**

NEXT STEPS:

Variable Reduction to be employed for the Zip Code variable.
Variable selection methods to choose the optimal number of parameters.
Employ cross-validation methods to test the out-of-sample error.
Try other algorithms such as Random Forest.

7 Conclusion

Much of the work with the NYC Property Sales data was data cleaning. After the initial process of data cleaning (predominantly treating missing values), we identified many outliers within the Sales Price and Square footage numerical variables. Isolating these data points and exploring the points individually was valuable with this data set. **Linear regression does not factor the multiple levels of dependencies between the variables to effectively predict the property prices**

Insights:

With this exploratory data analysis of the NYC Property Sale Prices, we found many interesting trends.

Property prices in NYC range from $220,000 (10% percentile of Property prices) all the way to $2.2 billion (95% percentile of Property prices). NYC has a place for everyone!

Price per square footage in Manhattan is as high as $16,000/sqft, while in Bloomfield, Staten Island is $26/sqft. Move to Staten Island, everyone!

Manhattan and Bronx sold the most residential condo apartments in large buildings/ residential societies, while Queens sold the most residential homes.

8 Future Plans

Finally we can conclude that there needs to be further analysis with the data. All the variables in the data set need to be tuned to perfectly complement the Property Sales business.

We need to use K Nearest Neighbors(KNN) to group the records as per the available features and may be try Random Forest to effective factor in all the various avenues of this data and predict the property prices.

Another suggestion would be convert this into a categorical problem rather than regression, by converting the property prices into ranges based on neighborhoods, boroughs and age of buildings and then predict the range to at least narrow it down to a ball park estimate of the real price.