NYC Property Price Prediction

NYC Property Price Prediction Project:

INTRODUCTION:

The objective of this project is to build a model to predict the propery prices in NYC. The dataset is intially downloaded from kaggle(https://www.kaggle.com/new-york-city/nyc-property-sales). The same data set has been uploaded to my github page as well(https://raw.githubusercontent.com/YazeedMoosafi/Covid-Dataset/main/nyc-rolling-sales.csv) which we'll be using for this project.

We split the data set into train_set(training set) and test_set(validation set) for developing/training and testing the model respectively.

There are ~84,000 records in the data set which will be split into training set and test set after cleansing the data.

METHODS/ANALYSIS:

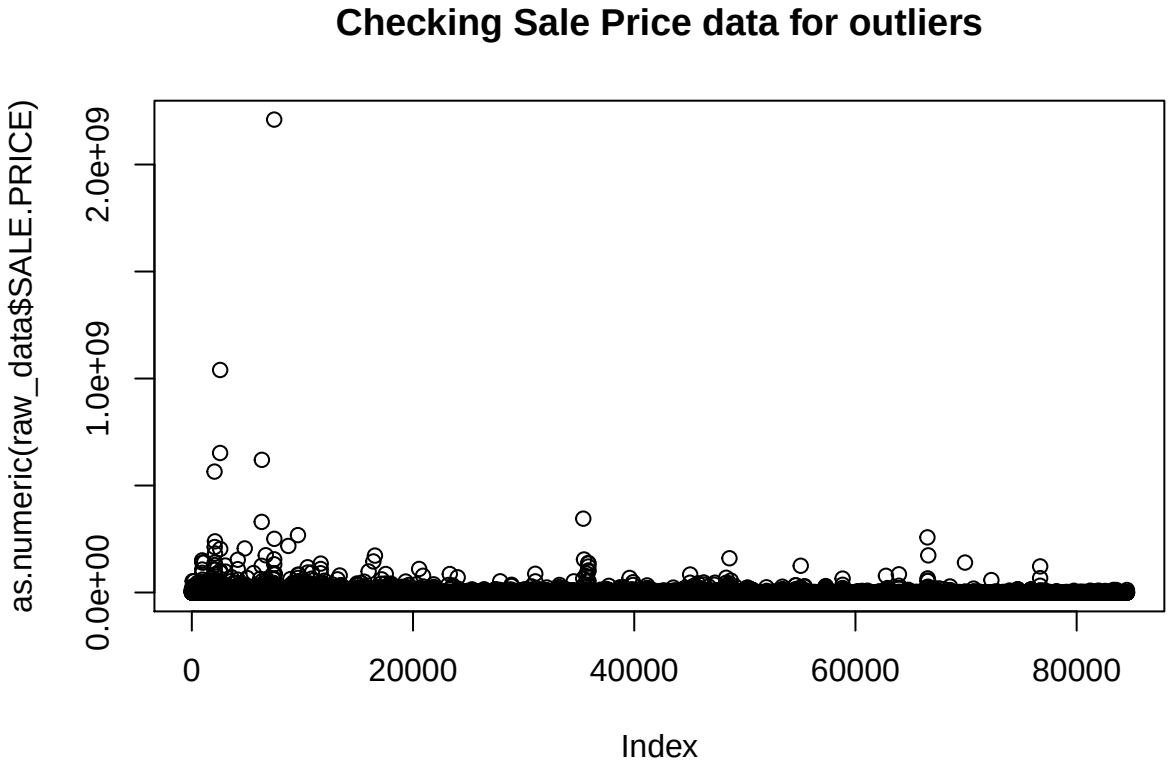
Upon inspection of the data set we see that the different predictors that can be used to predict the property prices are GROSS.SQUARE.FEET ,YEAR.BUILT, BOROUGH, ZIP.CODE, LAND.SQUARE.FEET, TAX.CLASS.AT.TIME.OF.SALE and BLOCK. But we see that there are lot of invalid data and outliers present in the data set. We remove the invalid data and outliers before proceeding with developing a model.

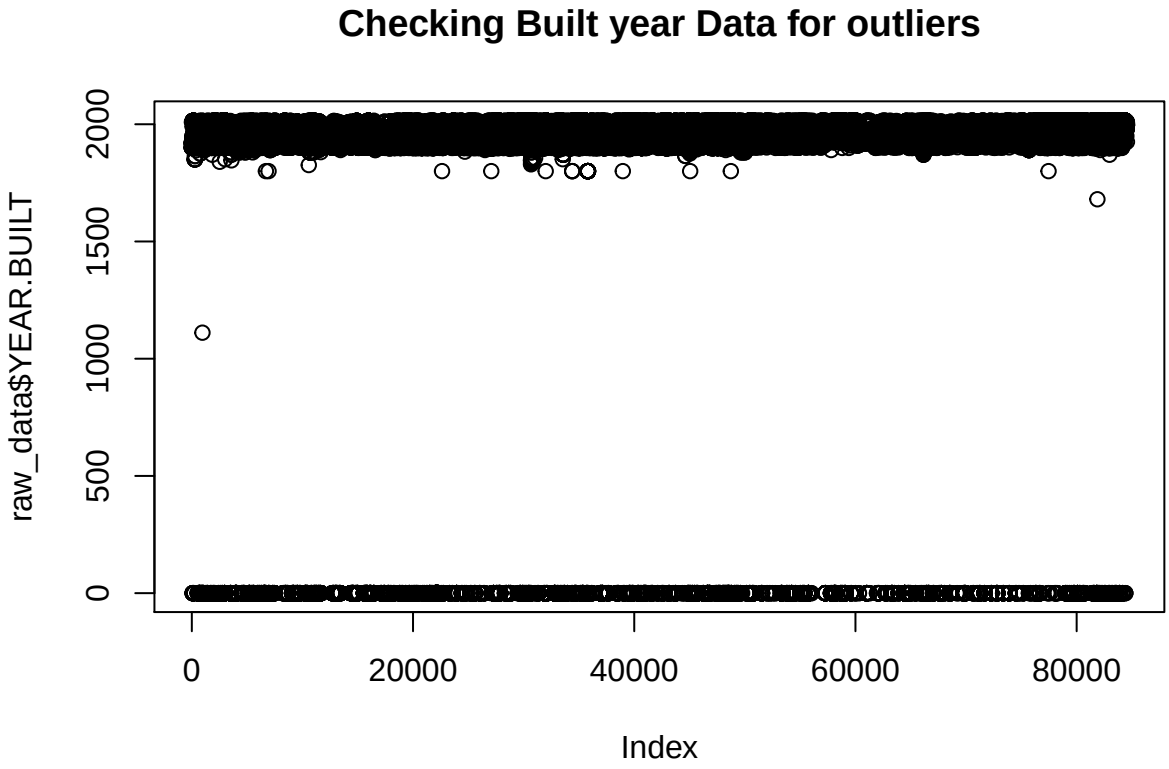
Once the data cleansing is done, we'll be going ahead and training different models to predict the property prices.

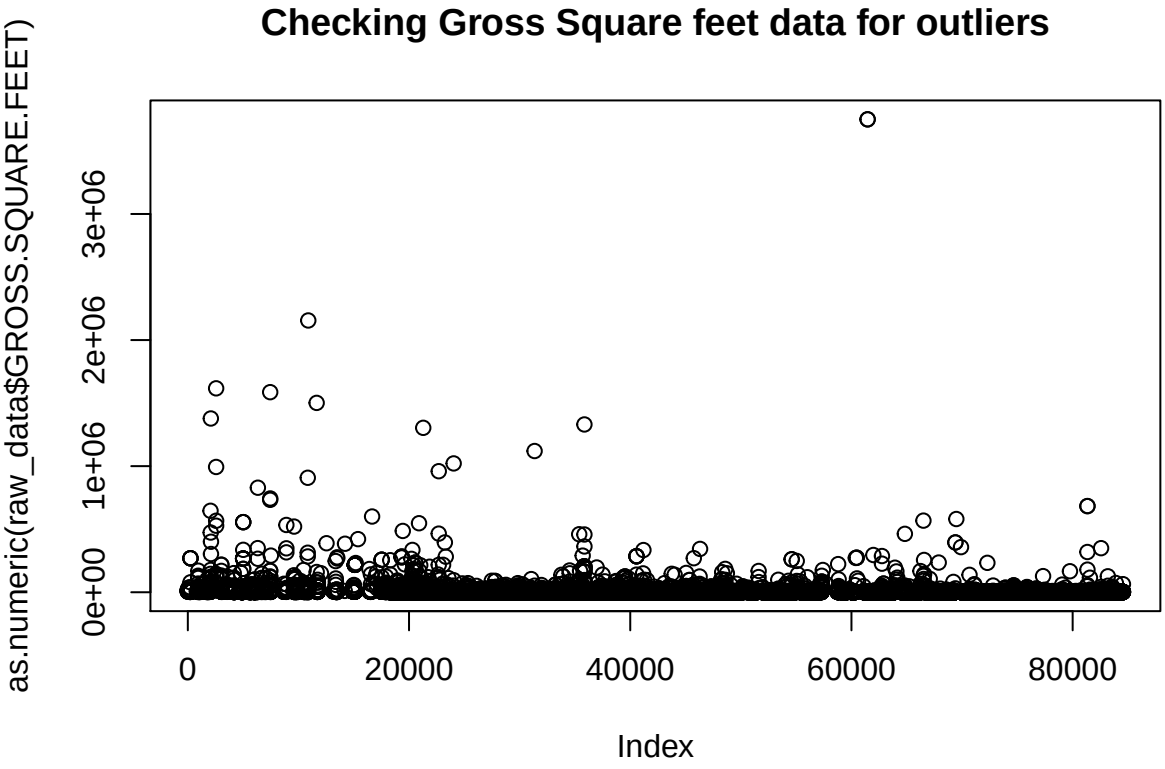
Download the data set from github

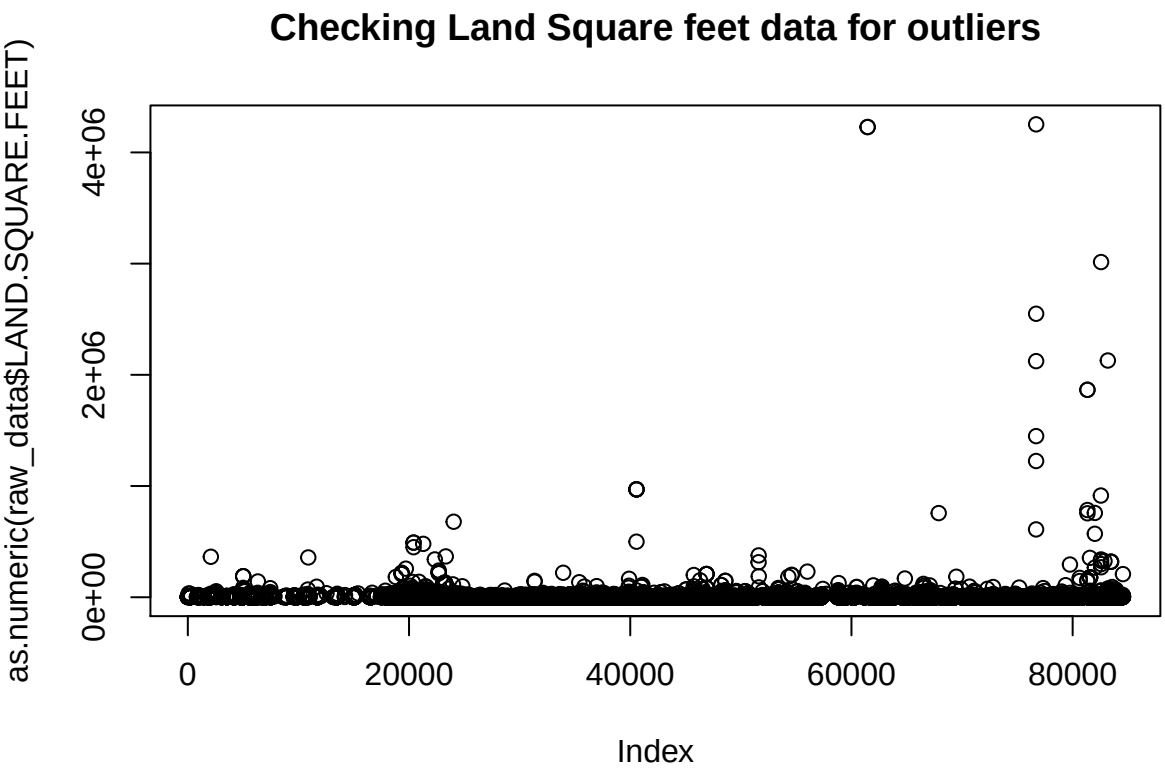
The first step is to dlwonload the raw data set from github. After that we'll be proceeding with data exploration.

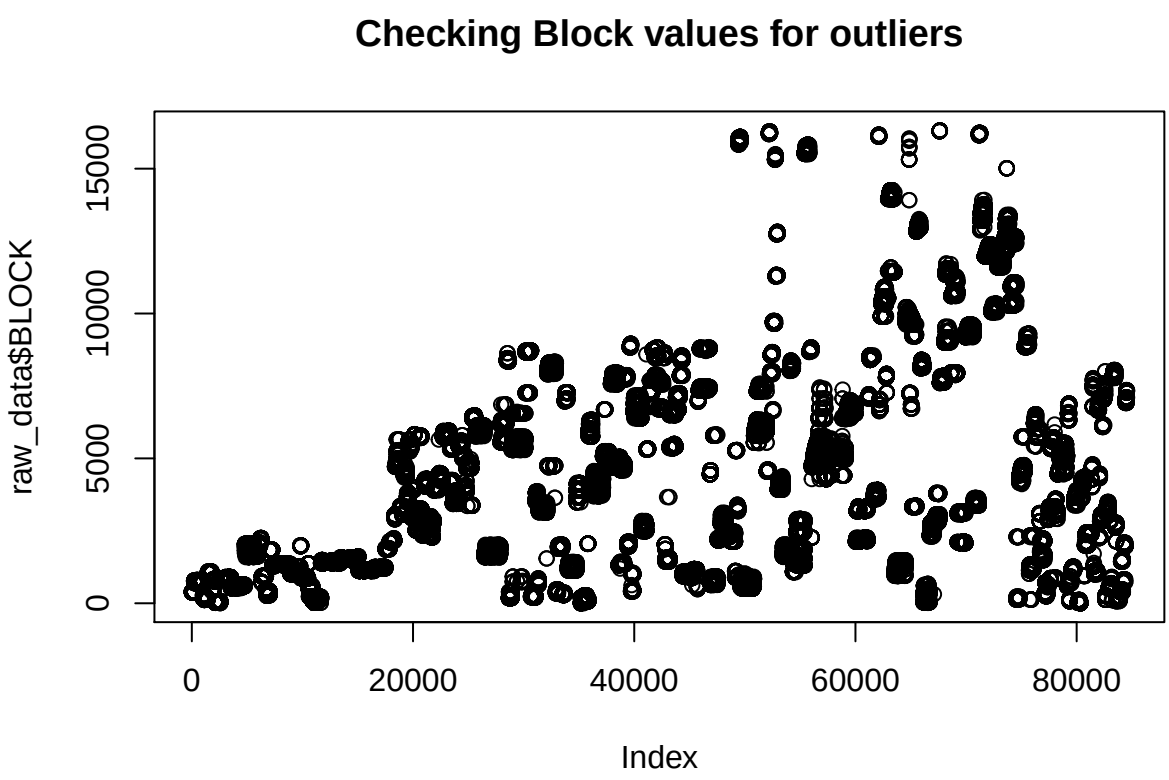
Data Exploration

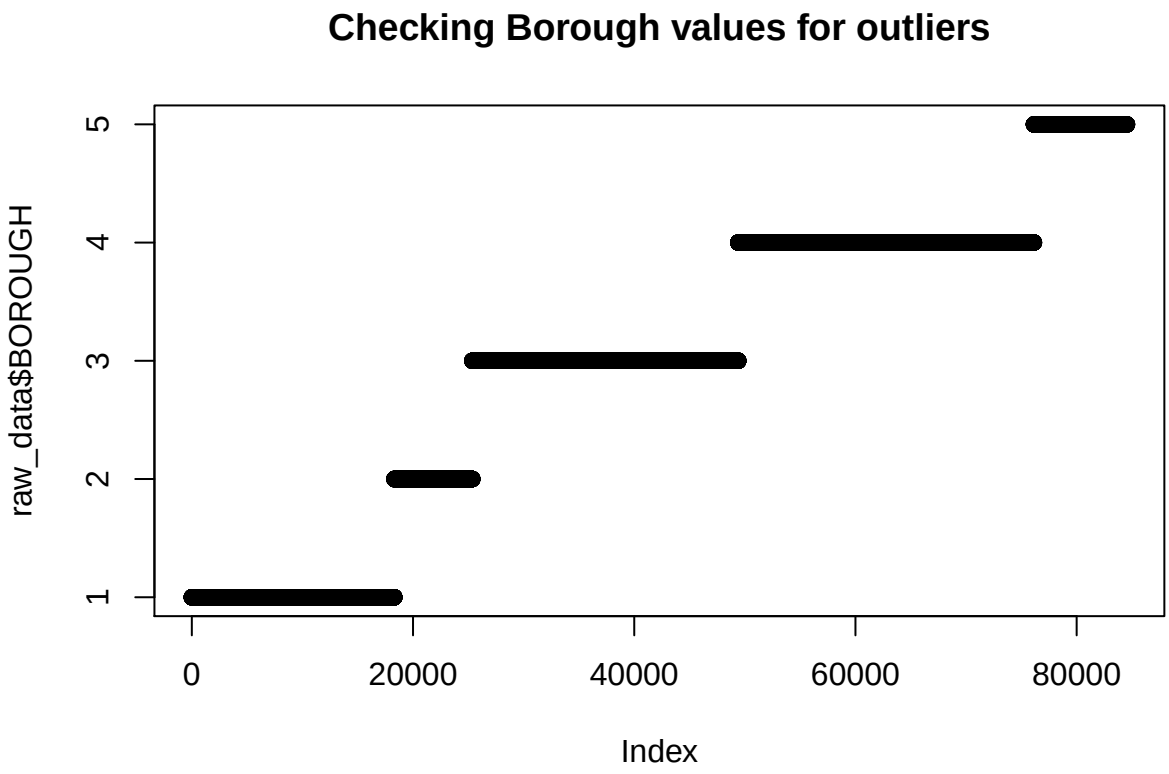


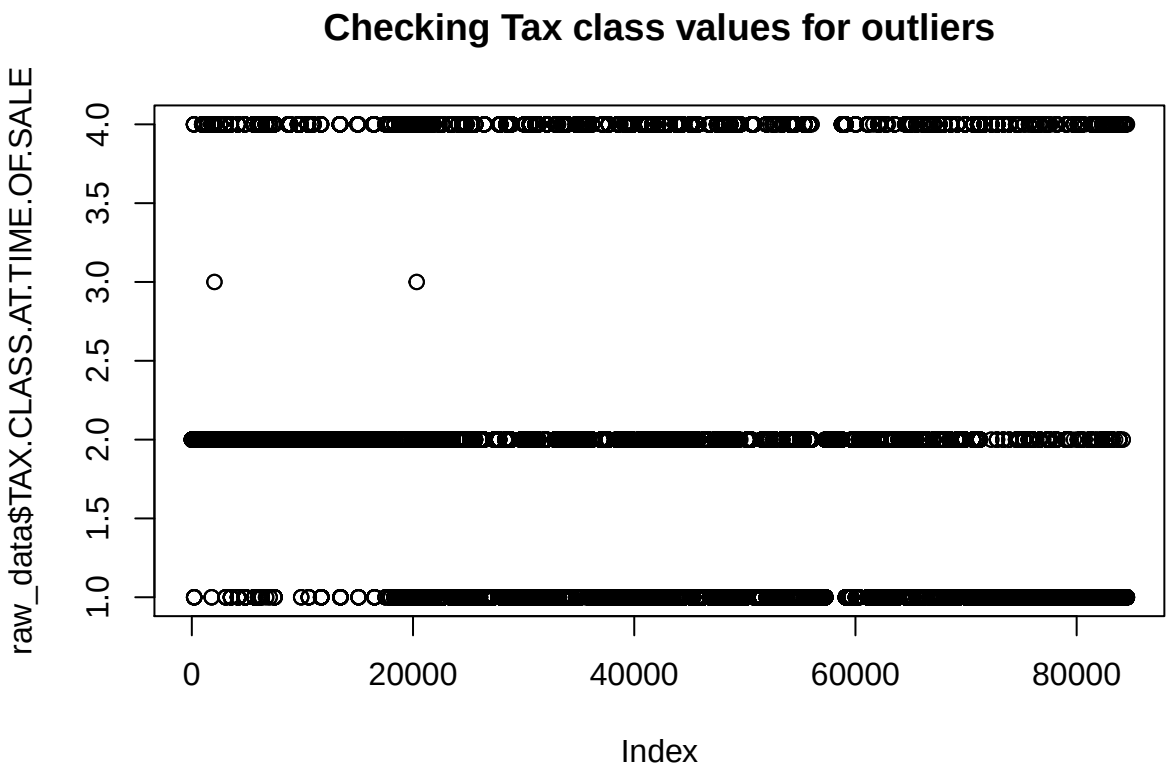


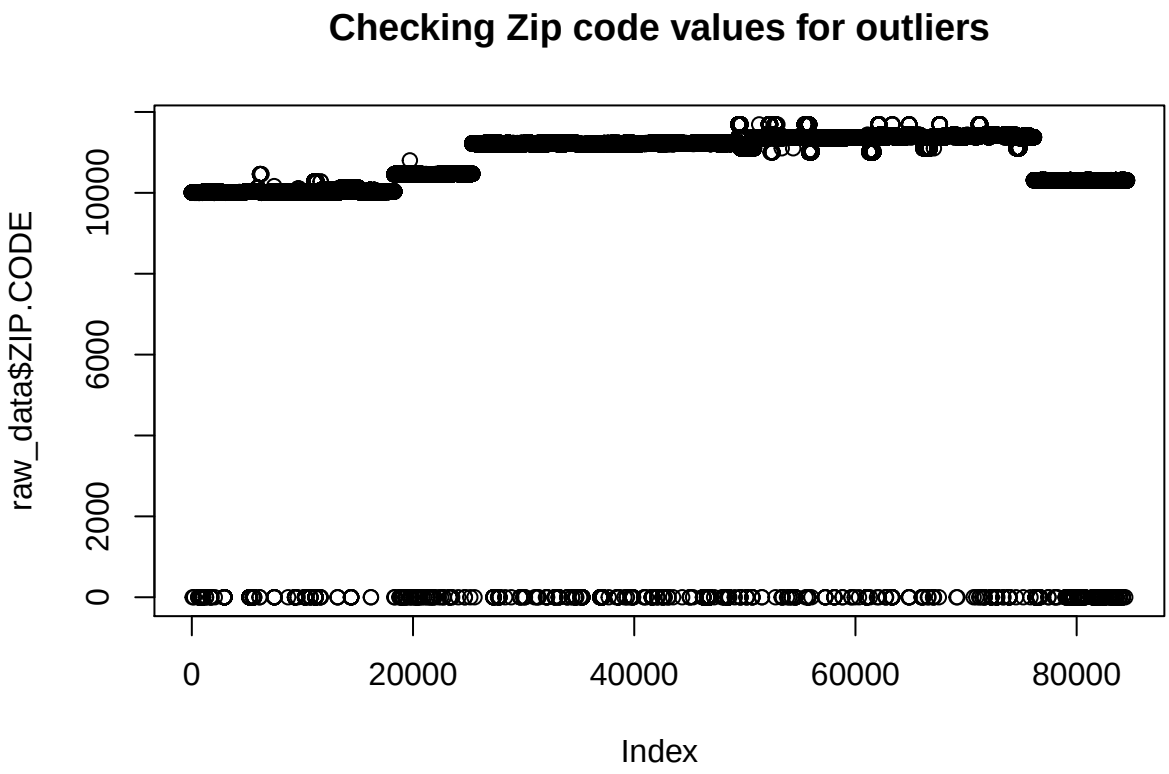

Checking Gross Square feet data for outliers
as.numeric(raw_data$GROSS.SQUARE.FEET)
0e+00
1e+06
2e+06
3e+06
0
20000
40000
60000
80000
Index


Checking Land Square feet data for outliers
as.numeric(raw_data$LAND.SQUARE.FEET)
0e+00
2e+06
4e+06
0
20000
40000
60000
80000
Index


Checking Block values for outliers
raw_data$BLOCK
0
5000
10000
15000
0
20000
40000
60000
80000
Index


Checking Borough values for outliers
raw_data$BOROUGH
1
2
3
4
5
0
20000
40000
60000
80000
Index


Checking Tax class values for outliers
raw_data$TAX.CLASS.AT.TIME.OF.SALE
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
20000
40000
60000
80000
Index



Checking for invalid values

```
head(sort(unique(raw_data$SALE.PRICE)))
```

```
## [1] " -  " "0"    "1"    "10"   "100"  "1000"
```

```
tail(sort(unique(raw_data$SALE.PRICE)))
```

```
## [1] "999980"  "999988"  "99999"   "999990"  "999999"  "9999999"
```

```
head(sort(unique(raw_data$YEAR.BUILT)))
```

```
## [1]    0 1111 1680 1800 1826 1829
```

```
tail(sort(unique(raw_data$YEAR.BUILT)))
```

```
## [1] 2012 2013 2014 2015 2016 2017
```

```r
head(sort(unique(raw_data$GROSS.SQUARE.FEET)))
```

```
## [1] " -  "   "0"      "100"    "1000"   "10000"  "100000"
```

```r
tail(sort(unique(raw_data$GROSS.SQUARE.FEET)))
```

```
## [1] "9972" "9975" "998"  "999"  "9990" "9992"
```

```r
head(sort(unique(raw_data$LAND.SQUARE.FEET)))
```

```
## [1] " -  "   "0"      "100"    "1000"   "10000"  "100000"
```

```r
tail(sort(unique(raw_data$LAND.SQUARE.FEET)))
```

```
## [1] "9977" "998"  "9980" "999"  "9992" "9996"
```

```r
head(sort(unique(raw_data$ZIP.CODE)))
```

```
## [1]     0 10001 10002 10003 10004 10005
```

```r
tail(sort(unique(raw_data$ZIP.CODE)))
```

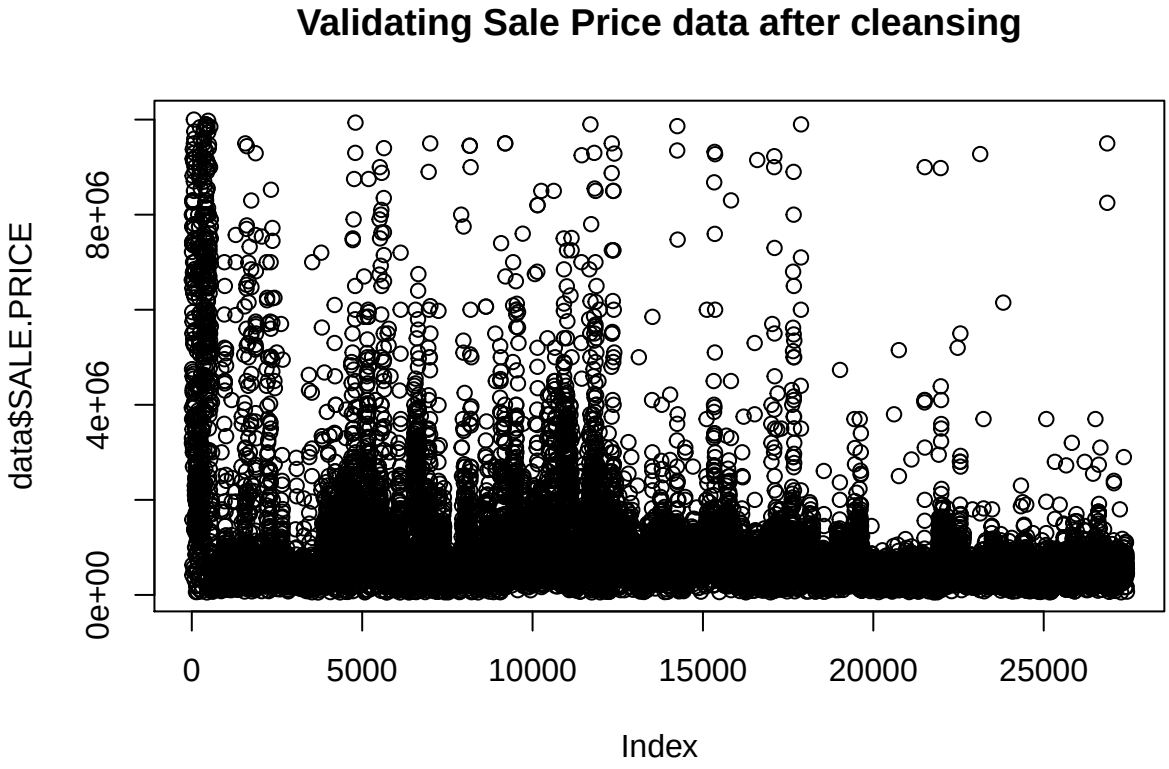
```
## [1] 11435 11436 11691 11692 11693 11694
```

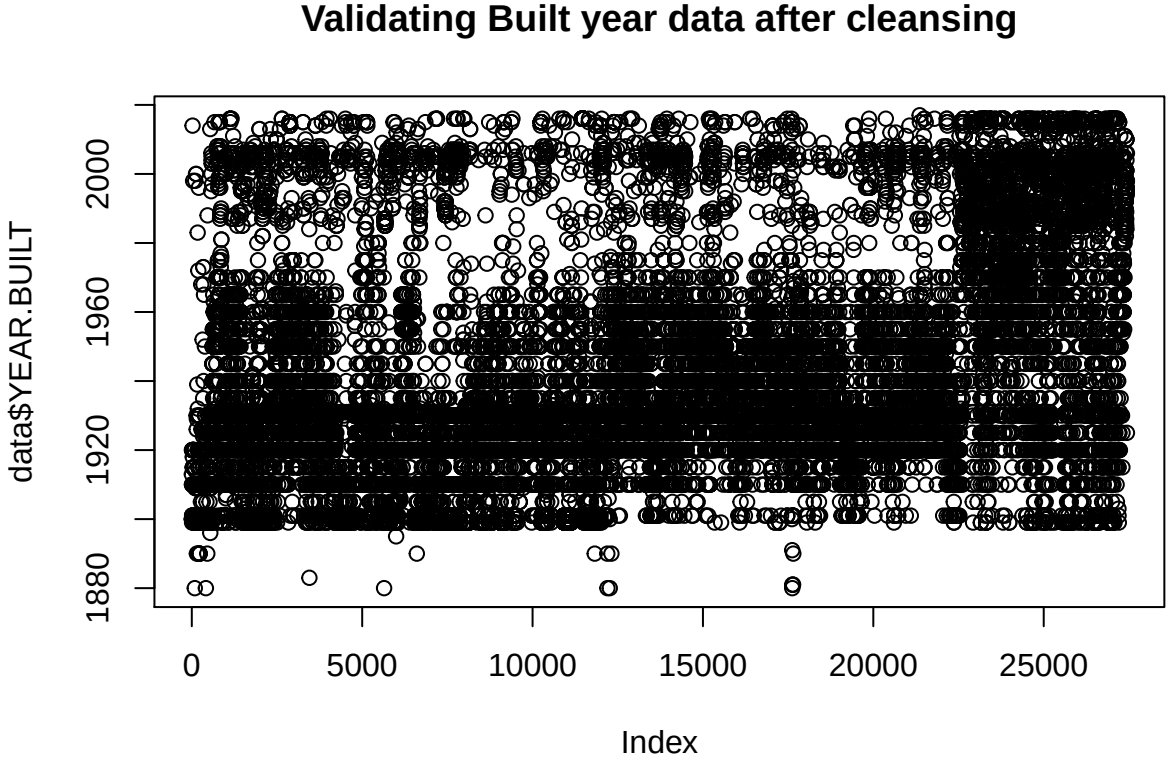
From data exploration we understand that we need to do the following as part of data cleansing inorder to predict the prices with better accuracy

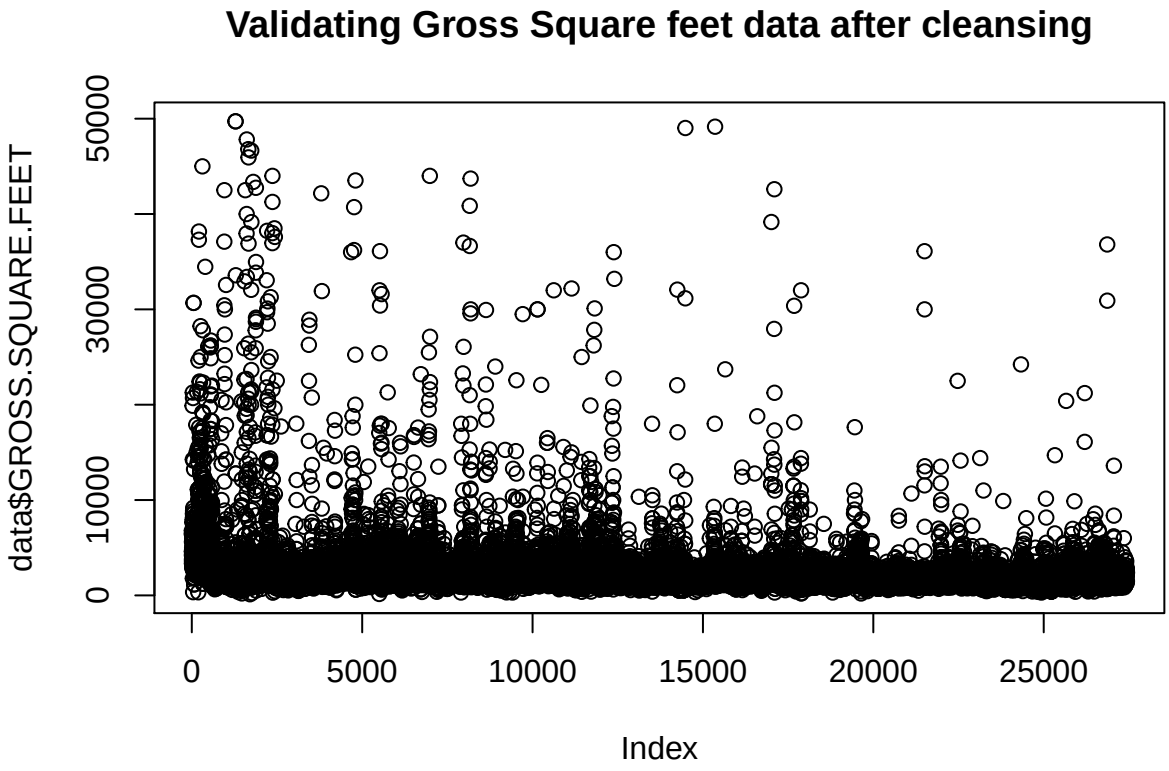
1. Choose the SALE price between 50,000 and 10 million
2. Remove invalid values of sale price, built year, gross square feet, land square feet and zip code
3. Gross and land square feet should be less than 50,000
4. Built year should be grated than 1875.

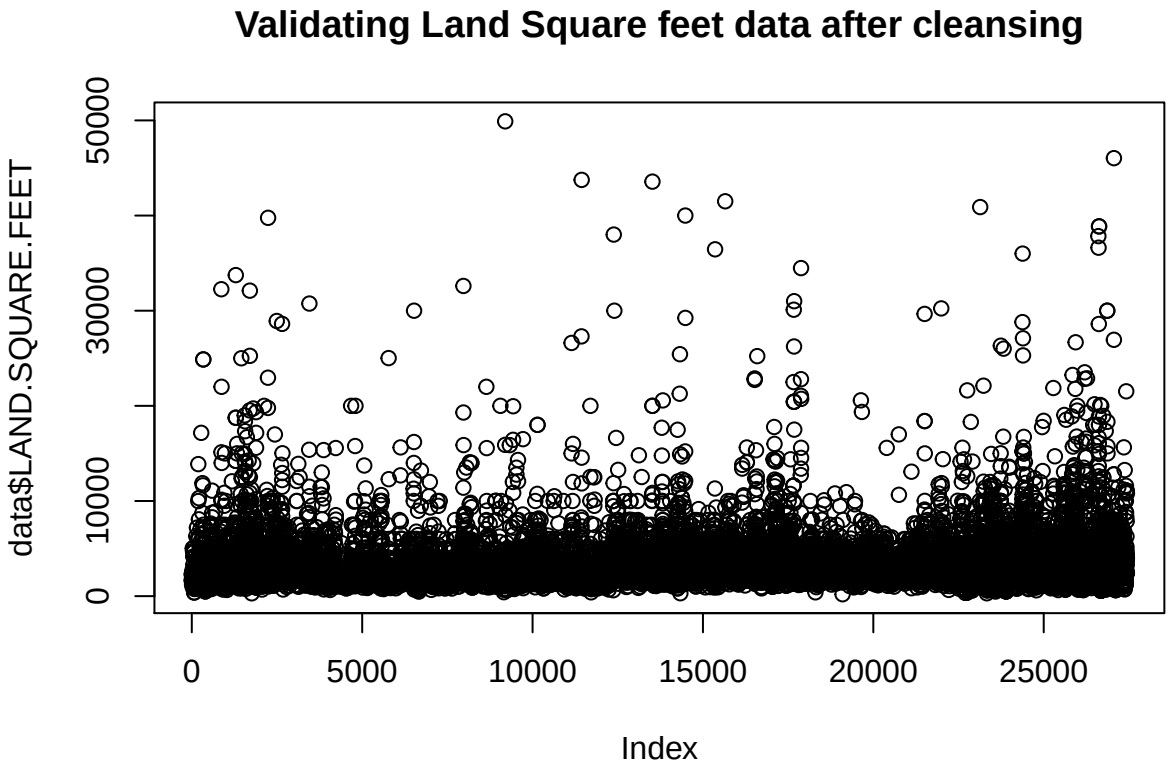
In the next part of the code will be doing the data cleansing and we'll be validating the data after cleansing

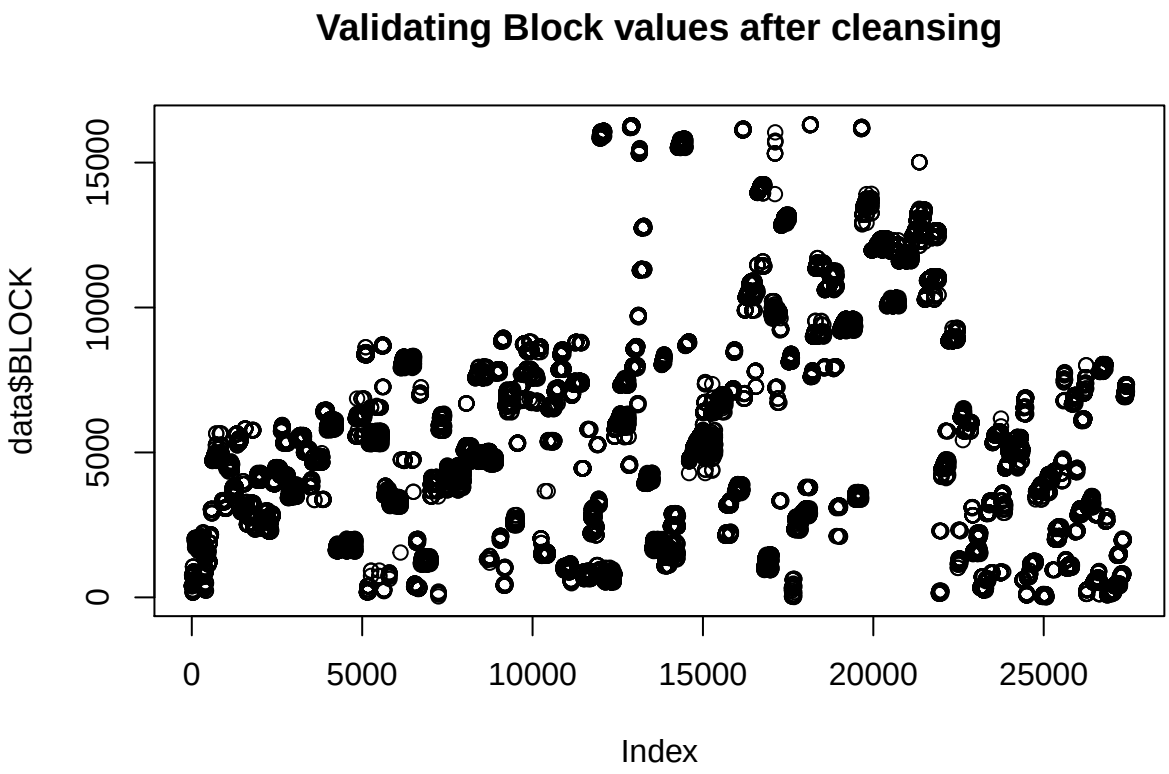
Check for invalid values and outliers after data cleansing

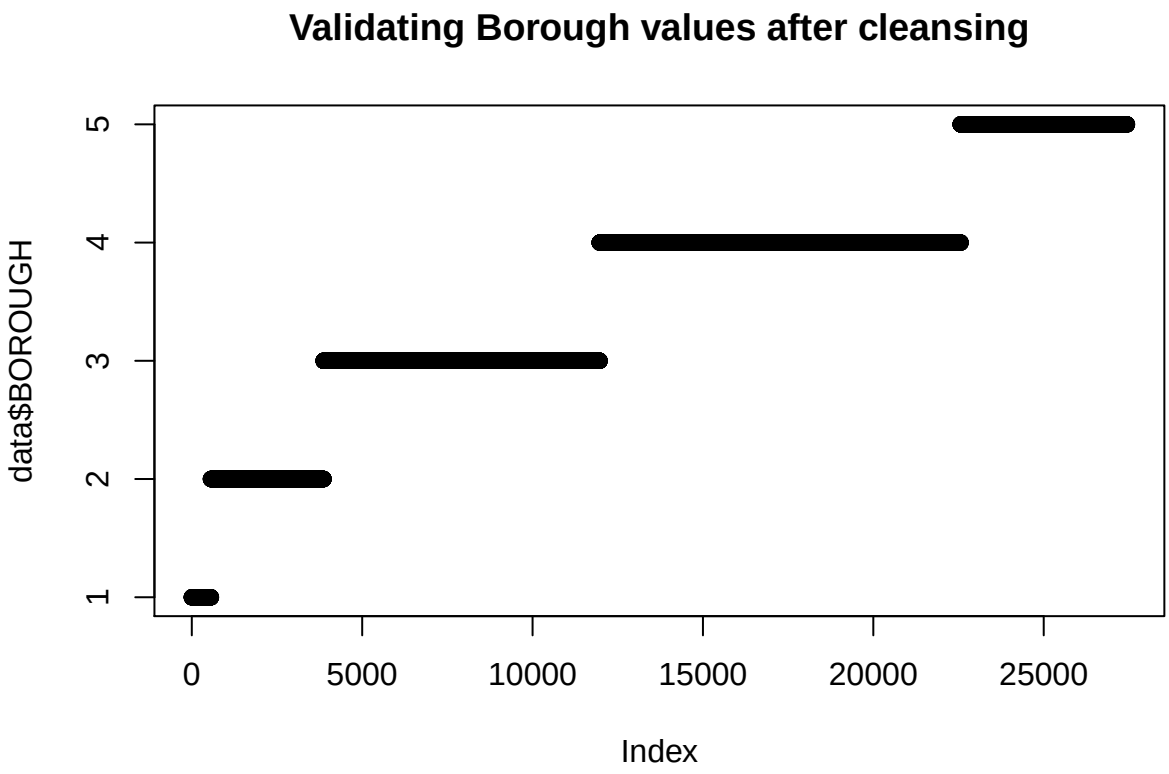


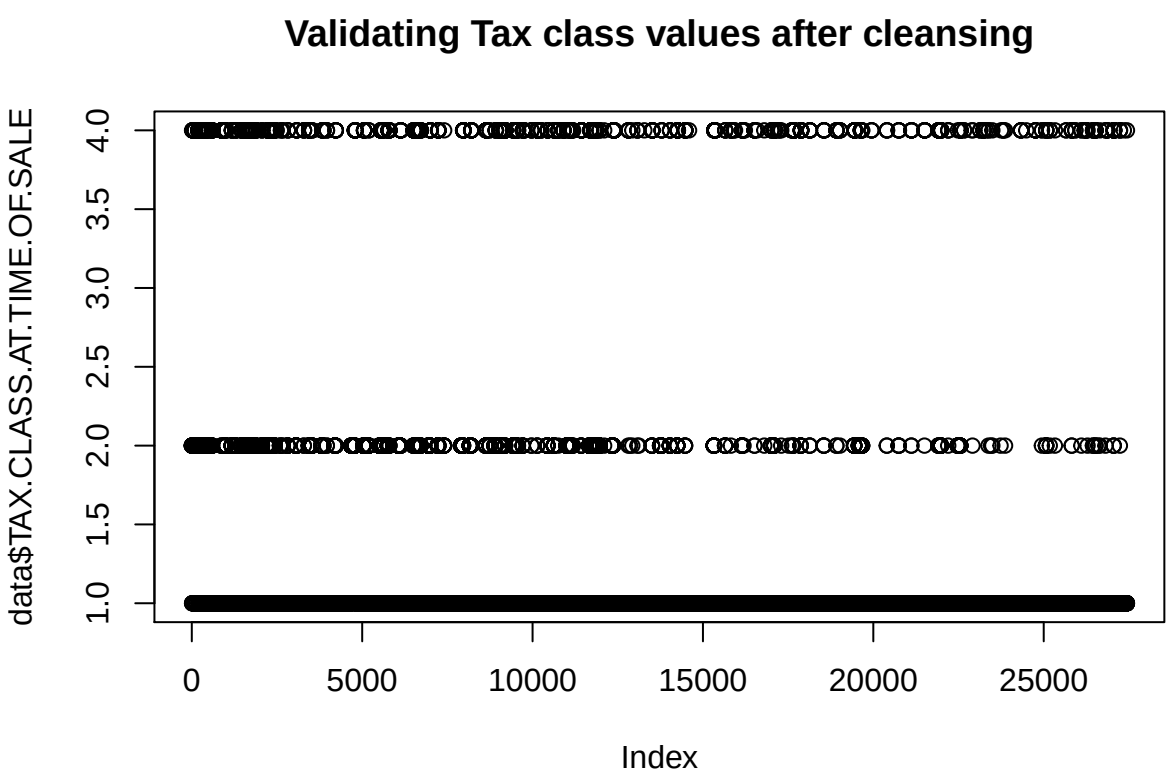


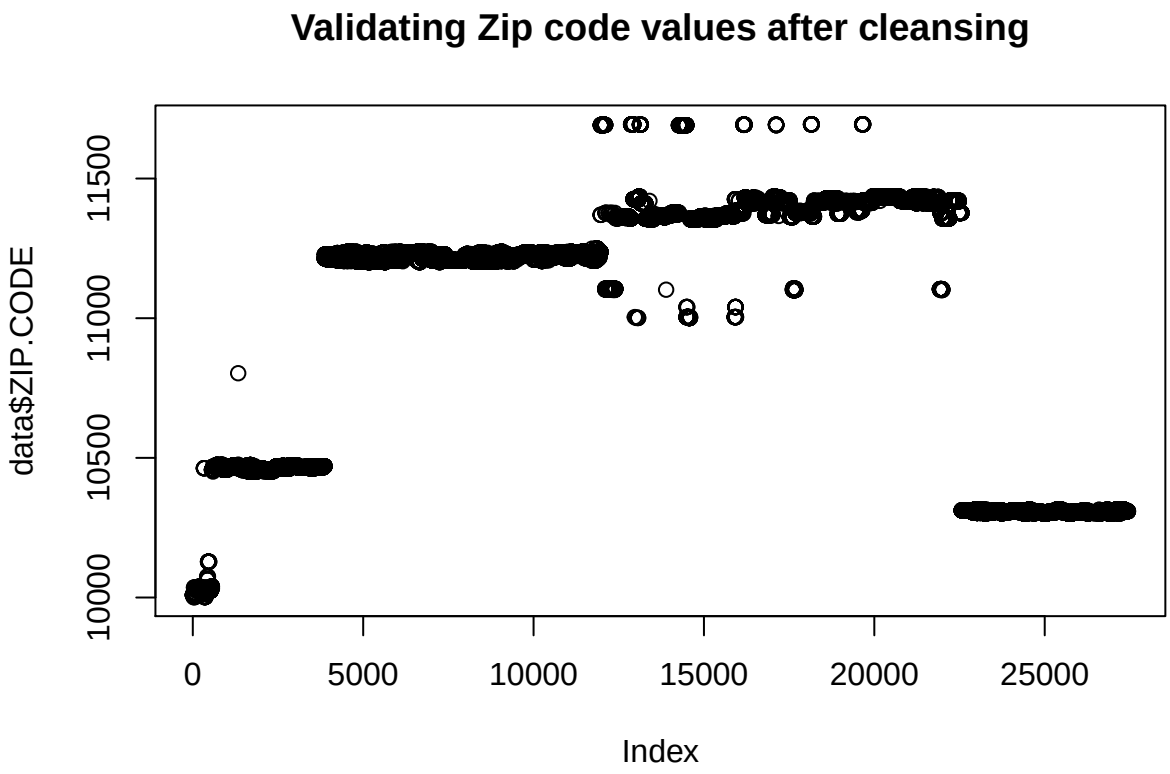

Validating Gross Square feet data after cleansing
data$GROSS.SQUARE.FEET
0
10000
30000
50000
0
5000
10000
15000
20000
25000
Index


Validating Land Square feet data after cleansing
data$LAND.SQUARE.FEET
0
10000
30000
50000
0
5000
10000
15000
20000
25000
Index


Validating Block values after cleansing
data$BLOCK
0
5000
10000
15000
0
5000
10000
15000
20000
25000
Index


Validating Borough values after cleansing
data$BOROUGH
1
2
3
4
5
0
5000
10000
15000
20000
25000
Index


Validating Tax class values after cleansing
data$TAX.CLASS.AT.TIME.OF.SALE
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
5000
10000
15000
20000
25000
Index



Check for invalid values after data cleansing

```
head(sort(unique(data$SALE.PRICE)))
```

```
## [1] 51000 51500 52000 52210 52500 52650
```

```
tail(sort(unique(data$SALE.PRICE)))
```

```
## [1]  9862640  9890400  9900000  9936760  9975000 10000000
```

```
head(sort(unique(data$YEAR.BUILT)))
```

```
## [1] 1880 1881 1883 1890 1891 1895
```

```
tail(sort(unique(data$YEAR.BUILT)))
```

```
## [1] 2012 2013 2014 2015 2016 2017
```

```r
head(sort(unique(data$GROSS.SQUARE.FEET)))
```

```
## [1] 120 150 200 205 240 247
```

```r
tail(sort(unique(data$GROSS.SQUARE.FEET)))
```

```
## [1] 46655 46800 47820 49021 49160 49720
```

```r
head(sort(unique(data$LAND.SQUARE.FEET)))
```

```
## [1] 200 297 300 305 353 384
```

```r
tail(sort(unique(data$LAND.SQUARE.FEET)))
```

```
## [1] 40890 41500 43563 43750 46034 49900
```

```r
head(sort(unique(data$ZIP.CODE)))
```

```
## [1] 10001 10002 10003 10009 10010 10011
```

```r
tail(sort(unique(data$ZIP.CODE)))
```

```
## [1] 11435 11436 11691 11692 11693 11694
```

Now that the data cleansing is done, we'll be proceeding with splitting the data set into train_set and test_set. After that we'll be training different models to predict the sale price. We'll be using the lm, glm, Rborist, svmLinear, knn, lasso and ridge methods for model training.

Model Training

After training the lm, glm, Rborist, svmLinear, knn, lasso and ridge models, we get the following RMSE values corresponding to each model. From that, we'll be picking the model with the best RMSE. We find that Rborist model gives the best RMSE value.

Method	RMSE
lm	802949.2
glm	802949.2
Rborist	550459.4
svmLinear	831955.9
knn	639923.3
lasso	803868.4
ridge	802951.4

```r
## [1] 550459.4
```

Fine tuning the model

We will now be doing repeated cross-validation using the traincontrol option to fine tune the Rborist model to further improve the RMSE. The RMSE achieved is shown below.

```
## [1] 549211.6
```

Final RMSE:

Below is the final RMSE value

```
## [1] 549211.6
```

RESULTS:

We tried to use mutliple models to predict the property prices in NYC. The model with the best RMSE(Rborist) was picked at the end and fine tuned it to improve the RMSE.

CONCLUSION:

We were able to develop a model to successfully predict the property prices in NYC. The accuracy of the model can still be improved by using advanced algorithms and by fine tuning the models.