SALES PRICE FOR PROPERTIES IN

- NOTE: MY COWORKER (KARIM GAMAL) AND I DECIDED TO SOLVE THE SAME PROBLEM, BUT WE EACH HAVE OUR OWN WAY OF CLEANING, AND WE ALSO HAVE DIFFERENT RESEARCH QUESTIONS.

GOAL

- FOR THIS PROJECT, I'M GOING TO BUILD A MACHINE LEARNING MODEL THAT COULD PREDICT THE PROPERTY SALES PRICE FOR PROPERTIES IN NYC. I'M DIVIDING MY PROJECT INTO THE FOLLOWING SECTIONS:
    - EDA
    - DATA PREPARATION
    - MODELLING

EXPLORATORY DATA ANALYSIS (EDA)

- THE EDA IS DIVIDED IN THE FOLLOWING STEPS:

  - BASIC DATA EXPLORATION
  - IDENTIFICATION OF VARIABLES AND DATA TYPES
  - MISSING VALUES
  - UNIVARIATE ANALYSIS
  - BIVARIATE ANALYSIS
  - CORRELATION MATRIX
  - CONCLUSION

  NOTE :

  - THE DATA SIZE BEFORE EDA WAS 26,118 RECORDS X 21 FEATURES.
  - AND AFTER EDA IT WAS 13,196 RECORDS X 19 FEATURES.

DATA PREPROCESSING

	BOROUGH	NEIGHBORHOOD	BUILDING CLASS CATEGORY	TAX CLASS AT PRESENT	BLOCK	LOT	EASE-MENT	BUILDING CLASS AT PRESENT	ADDRESS	APARTMENT NUMBER	...	RESIDENTIAL UNITS
0	4.0	AIRPORT JFK	39 TRANSPORTATION FACILITIES	4	14260.0	80.0	NaN	V7	N/A ROCKAWAY BLVD	NaN	...	0.0
1	4.0	AIRPORT LA GUARDIA	01 ONE FAMILY DWELLINGS	1	949.0	41.0	NaN	A5	19-62 81ST STREET	NaN	...	1.0
2	4.0	AIRPORT LA GUARDIA	12 CONDOS - WALKUP APARTMENTS	2	949.0	1003.0	NaN	R2	19-37 80TH STREET, 1	1	...	1.0
3	4.0	ARVERNE	01 ONE FAMILY DWELLINGS	1	15830.0	20.0	NaN	A5	3-20 BEACH 41 STREET	NaN	...	1.0
4	4.0	ARVERNE	01 ONE FAMILY DWELLINGS	1	15830.0	27.0	NaN	A5	336 BEACH 41ST STREET	NaN	...	1.0

DATASET

Data Description

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
BOROUGH	20357.0	NaN	NaN	NaN	4.0	0.0	4.0	4.0	4.0	4.0	4.0
NEIGHBORHOOD	20357	61	FLUSHING-NORTH	2006	NaN	NaN	NaN	NaN	NaN	NaN	NaN
BUILDING CLASS CATEGORY	20357	40	01 ONE FAMILY DWELLINGS	7330	NaN	NaN	NaN	NaN	NaN	NaN	NaN
TAX CLASS AT PRESENT	20356	10	1	12812	NaN	NaN	NaN	NaN	NaN	NaN	NaN
BLOCK	20357.0	NaN	NaN	NaN	6914.861866	4482.172669	15.0	2977.0	6598.0	10587.0	16350.0
LOT	20357.0	NaN	NaN	NaN	209.179594	441.273336	1.0	17.0	40.0	83.0	7113.0
EASE-MENT	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
BUILDING CLASS AT PRESENT	20356	111	A1	3481	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ADDRESS	20357	19776	21-36 28TH STREET	10	NaN	NaN	NaN	NaN	NaN	NaN	NaN
APARTMENT NUMBER	2608	1089	3B	42	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ZIP CODE	20357.0	NaN	NaN	NaN	11374.850371	119.554457	11001.0	11361.0	11375.0	11420.0	11697.0
RESIDENTIAL UNITS	16414.0	NaN	NaN	NaN	1.761971	5.982168	0.0	1.0	1.0	2.0	314.0
COMMERCIAL UNITS	14784.0	NaN	NaN	NaN	0.135417	1.065624	0.0	0.0	0.0	0.0	89.0
TOTAL UNITS	16910.0	NaN	NaN	NaN	1.828681	5.995685	0.0	1.0	1.0	2.0	314.0
LAND SQUARE FEET	14288	2770	4,000	1777	NaN	NaN	NaN	NaN	NaN	NaN	NaN
GROSS SQUARE FEET	14288	2626	0	446	NaN	NaN	NaN	NaN	NaN	NaN	NaN
YEAR BUILT	19498.0	NaN	NaN	NaN	1949.336599	29.627932	1018.0	1930.0	1945.0	1960.0	2021.0
TAX CLASS AT TIME OF SALE	20357.0	NaN	NaN	NaN	1.46048	0.776176	1.0	1.0	1.0	2.0	4.0
BUILDING CLASS AT TIME OF SALE	20357	111	A1	3486	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SALE PRICE	20357	2711	0	7159	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SALE DATE	20357	359	2020-09-09	174	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
Shape of the Dataframe: (26118, 21)
```

DATA DESCRIPTION

DATA CLEANING

- **DATA DUPLICATE:** WE HAVE '5774' DUPLICATE RECORDS SO I WILL REMOVE ALL THE DUPLICATES

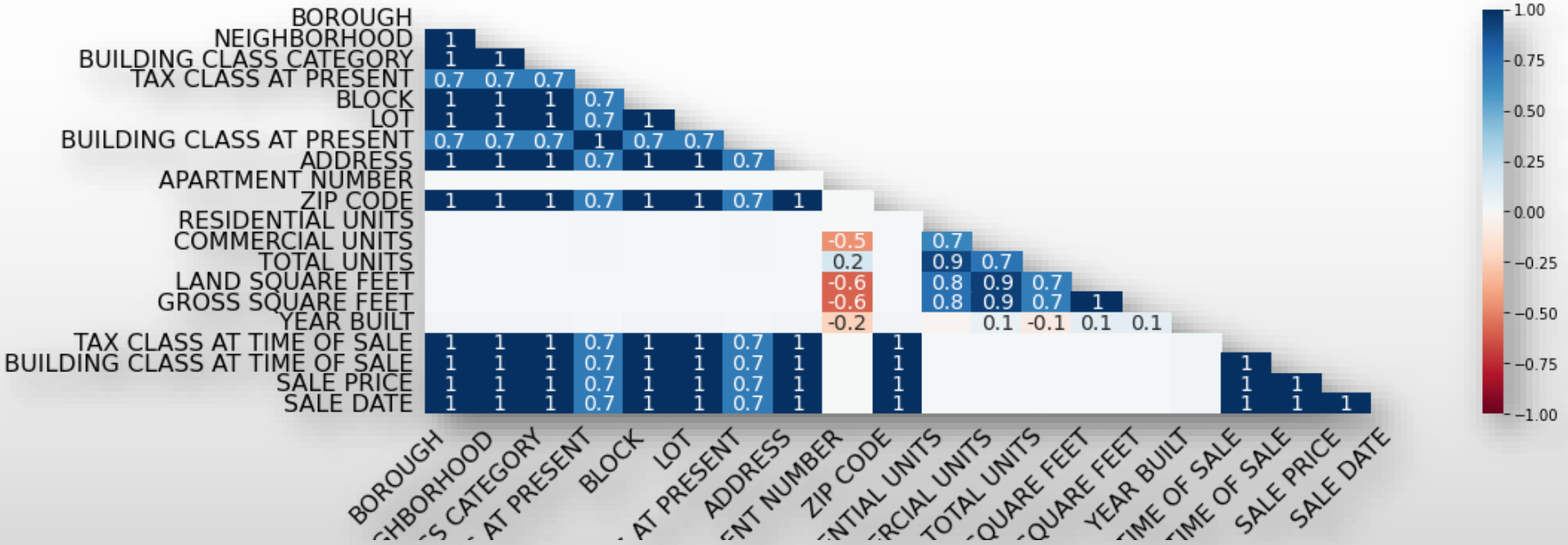
- **THE SHAPE OF THE DATA AFTER REMOVE DUPLICATES**

```
print('Shape of the Dataframe:', df.shape,'\n')
```

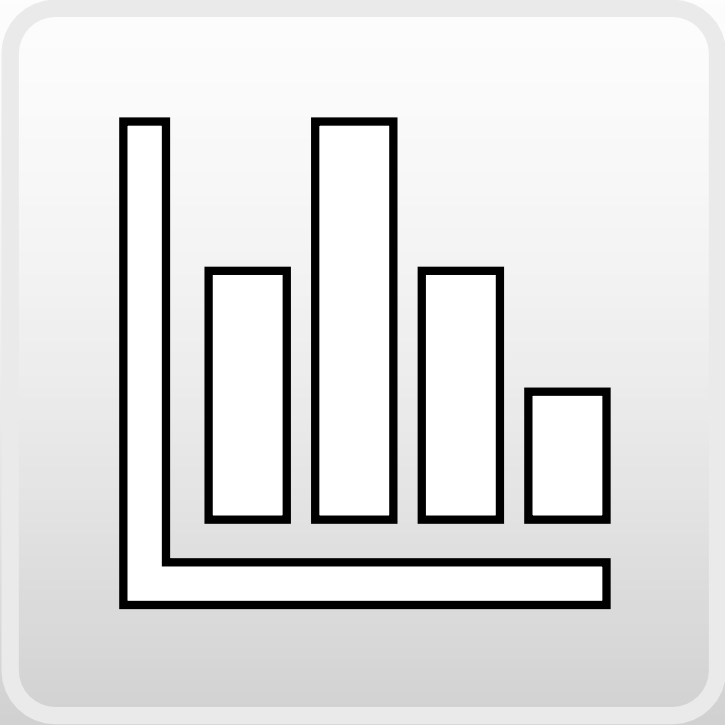
```
Shape of the Dataframe: (20344, 21)
```

MISSING VALUES

	Total missing	% missing
EASE-MENT	20344	100.000000
APARTMENT NUMBER	17736	87.180495
LAND SQUARE FEET	6066	29.817145
GROSS SQUARE FEET	6066	29.817145
COMMERCIAL UNITS	5570	27.379080
RESIDENTIAL UNITS	3940	19.366890
TOTAL UNITS	3444	16.928824
YEAR BUILT	860	4.227291
TAX CLASS AT PRESENT	2	0.009831
BUILDING CLASS AT PRESENT	2	0.009831
BOROUGH	1	0.004915
SALE PRICE	1	0.004915
BUILDING CLASS AT TIME OF SALE	1	0.004915
TAX CLASS AT TIME OF SALE	1	0.004915
ZIP CODE	1	0.004915
NEIGHBORHOOD	1	0.004915
ADDRESS	1	0.004915
LOT	1	0.004915
BLOCK	1	0.004915
BUILDING CLASS CATEGORY	1	0.004915
SALE DATE	1	0.004915



MISSING VALUES CORRELATION



DATA EXPLORATION

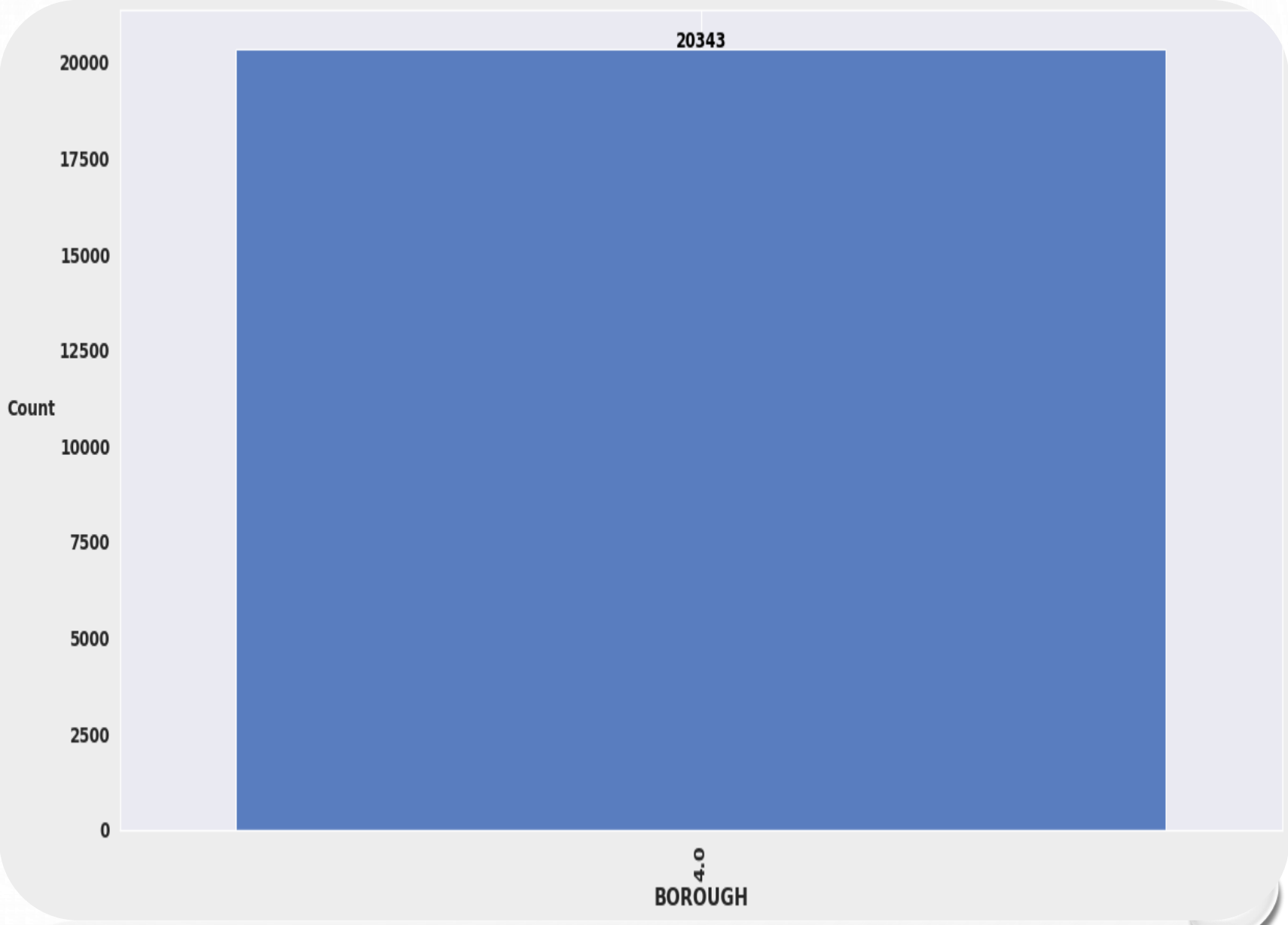
EASE-MENT & APARTMENT NUMBER *COLUMNS*

	Total missing	% missing
EASE-MENT	20344	100.000000
APARTMENT NUMBER	17736	87.180495

As we can see, these two columns contains more than 50% missing values.

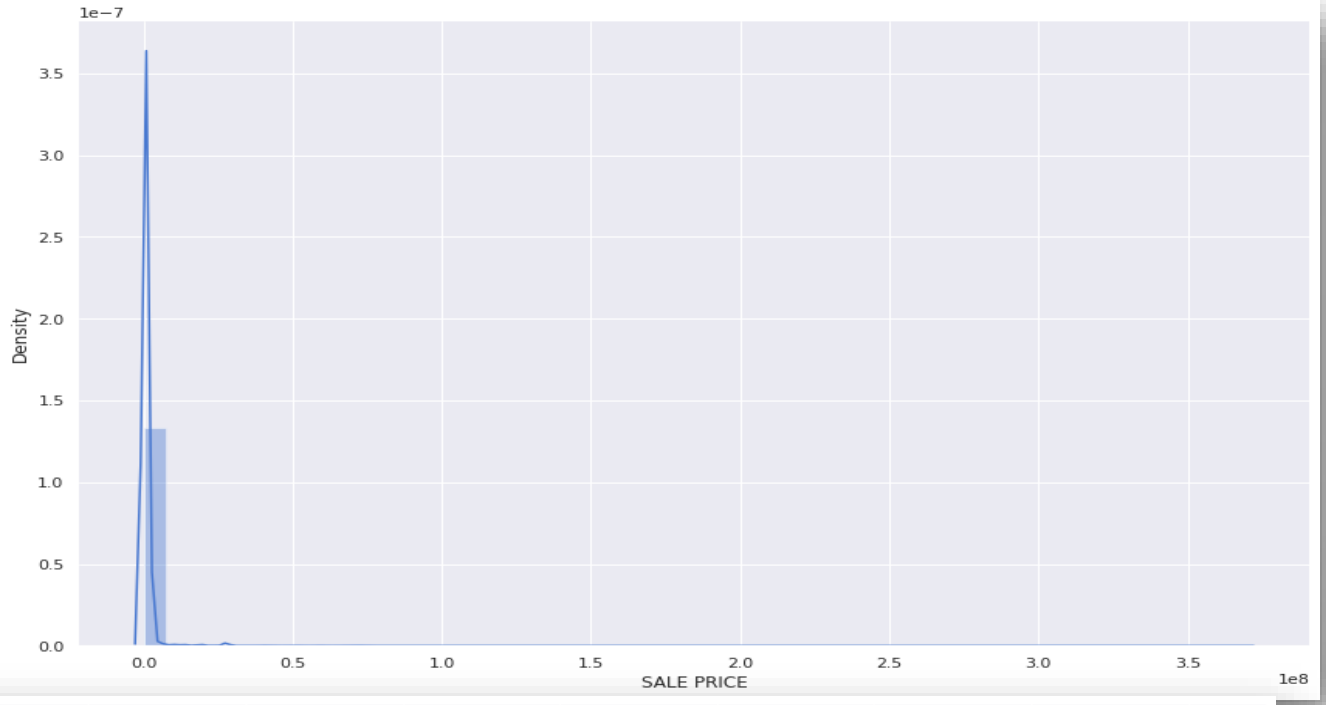
BOROUGH COLUMN

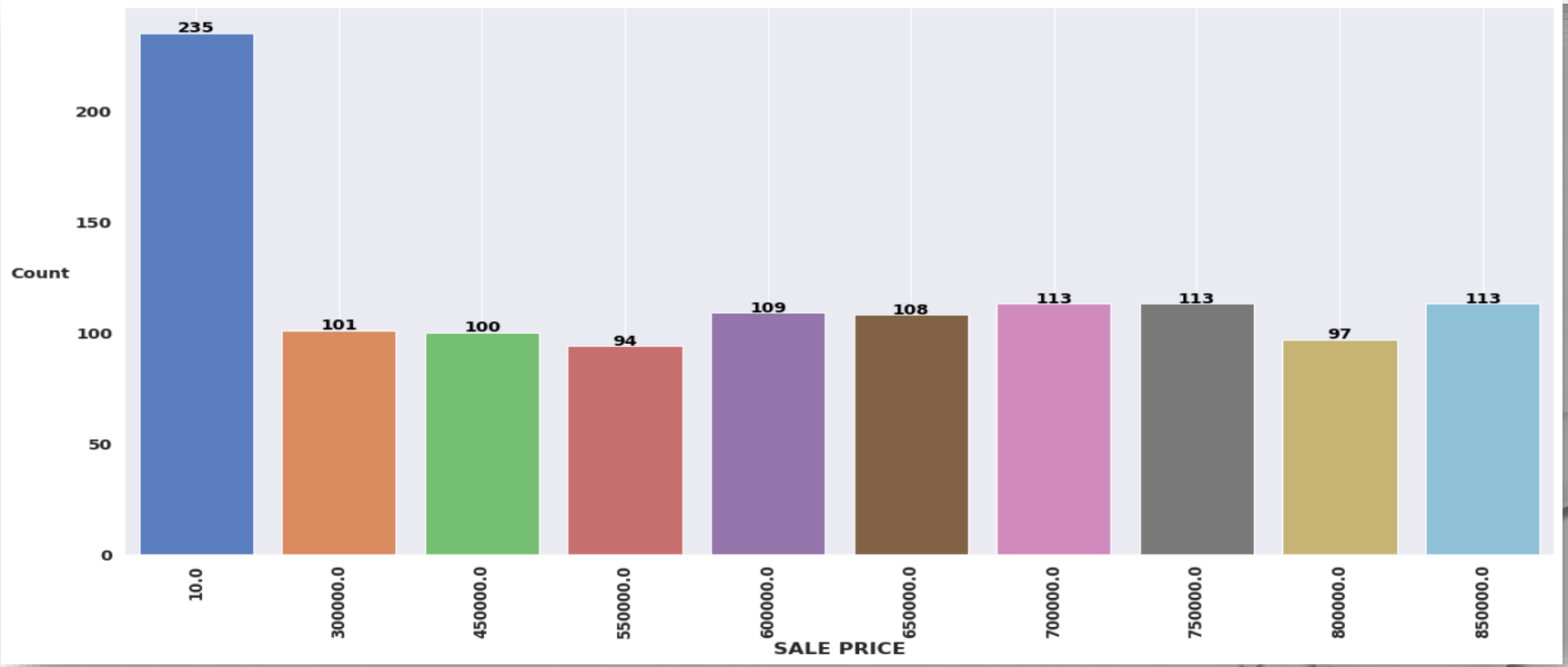
As we can see, this column contains only one value, So we will drop it



SALE PRICE COLUMN

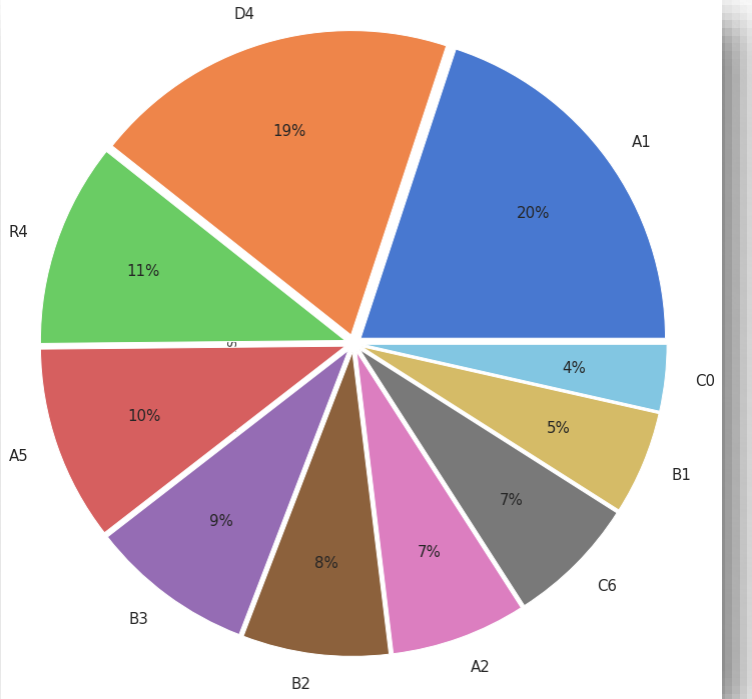
In this column, I dropped all records which have a ‘0’ value

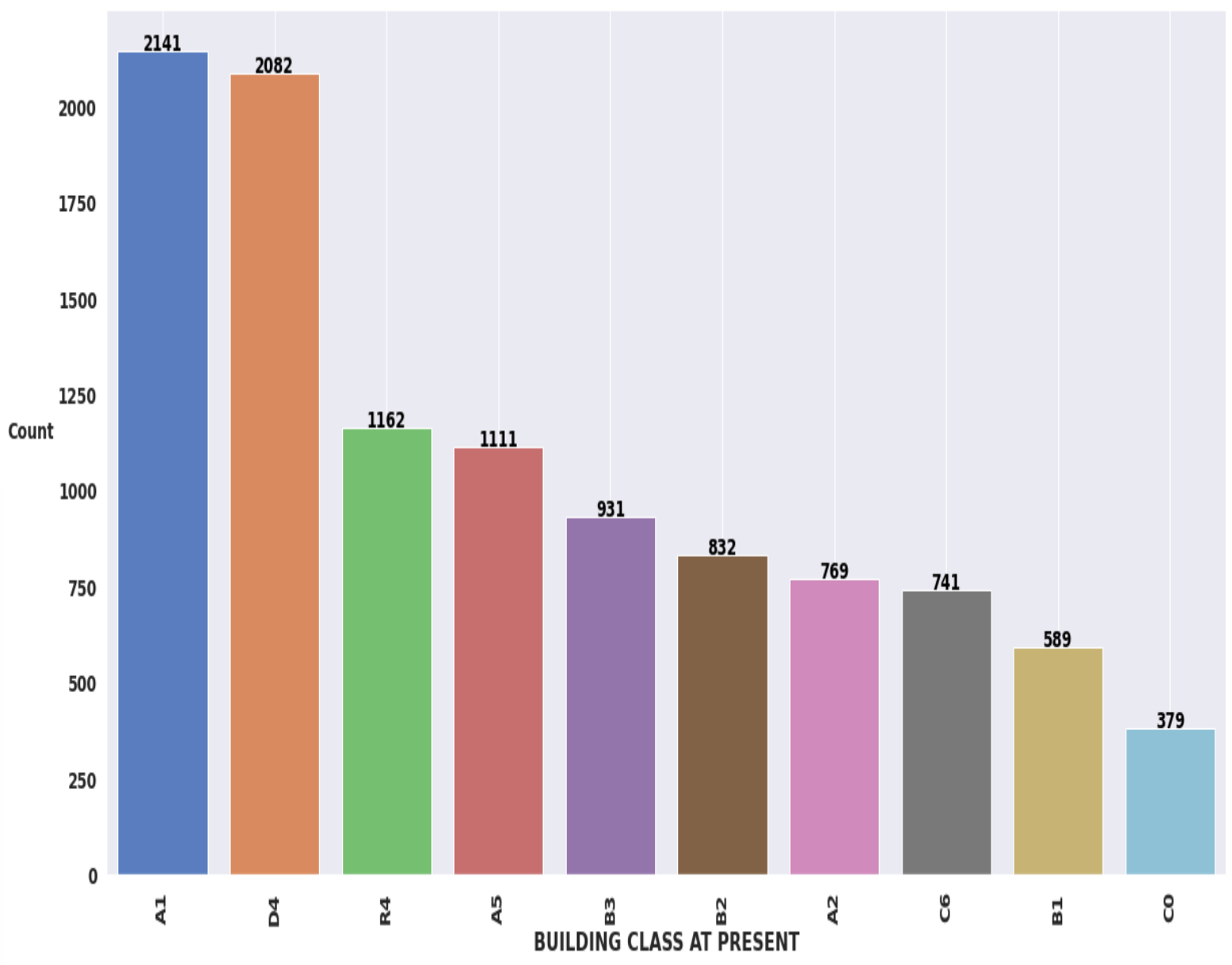




BUILDING CLASS AT PRESENT *COLUMN*

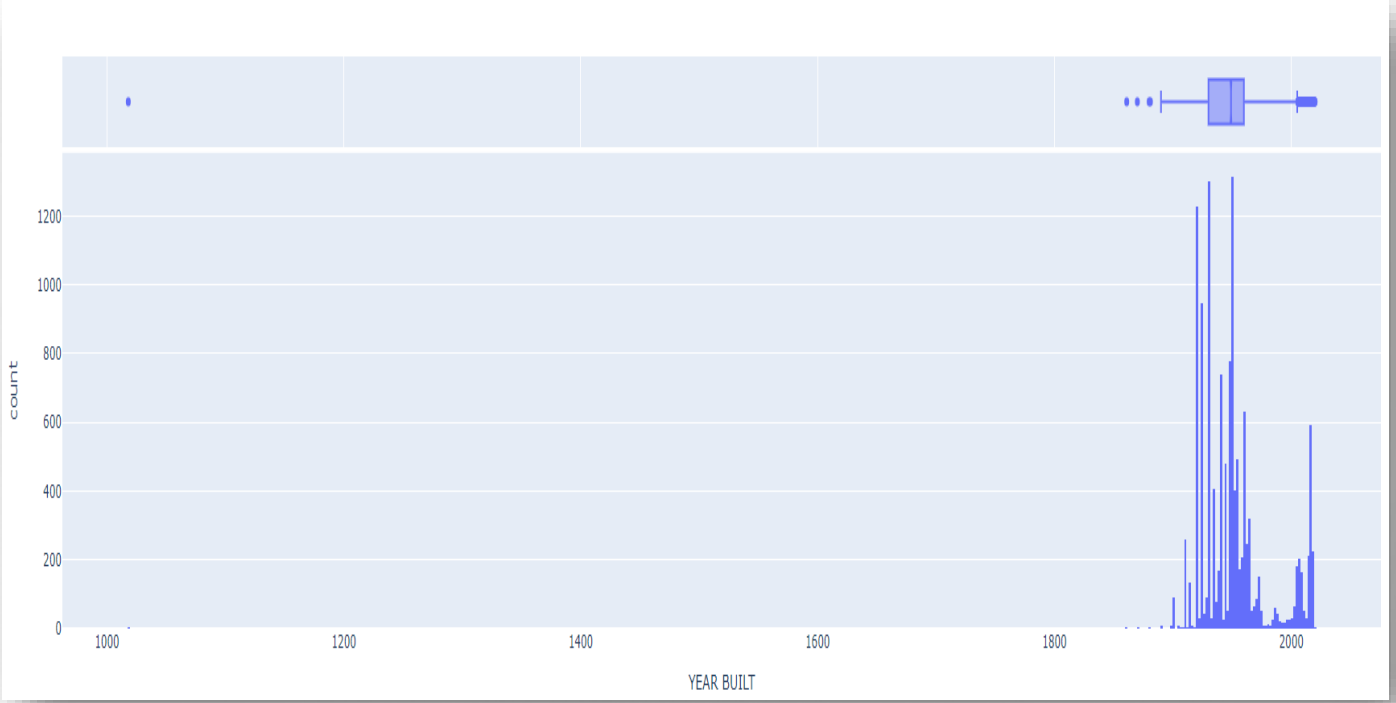
This column have ’97’ unique values

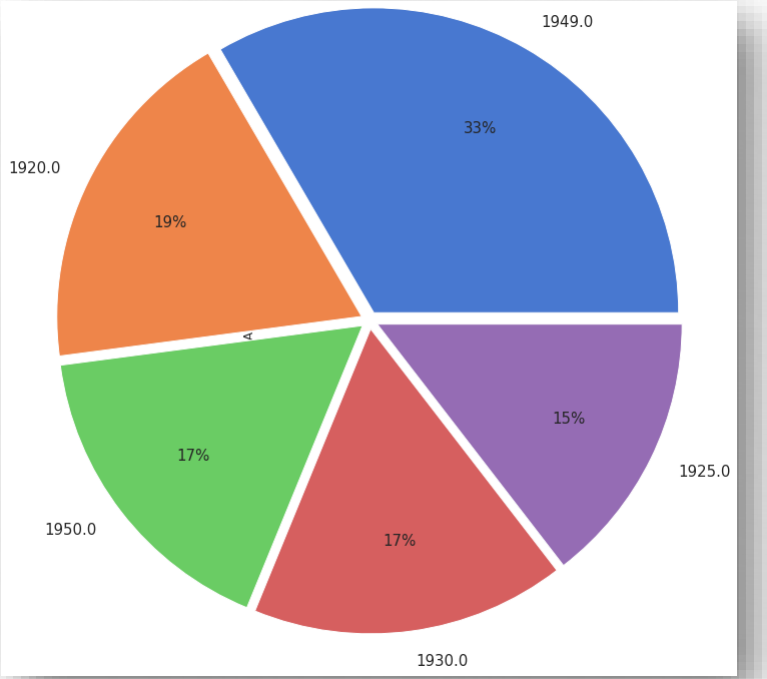




YEAR BUILT *COLUMN*

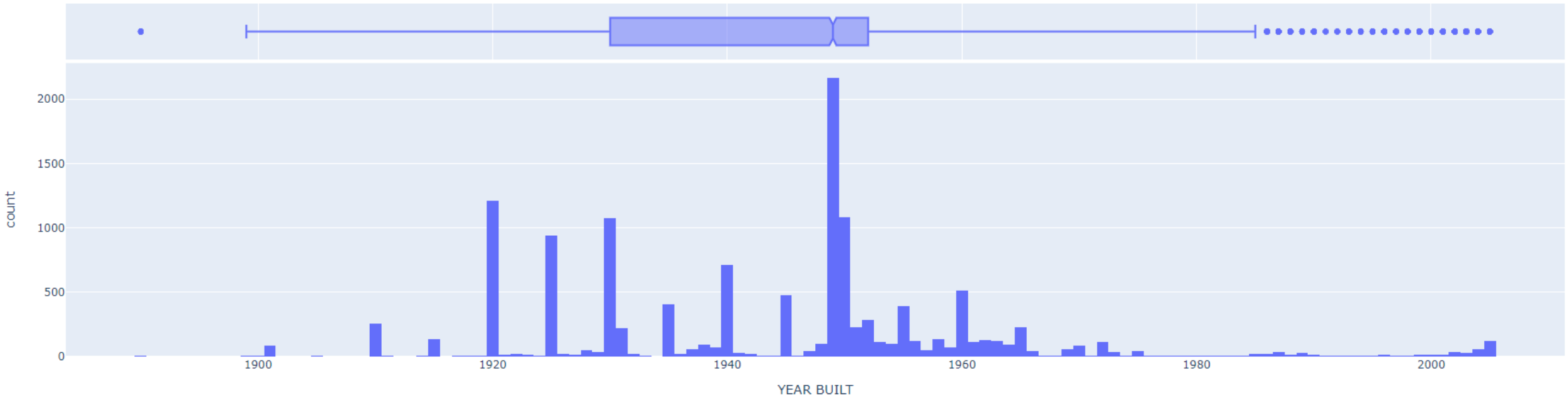
This column have ’123’ unique values





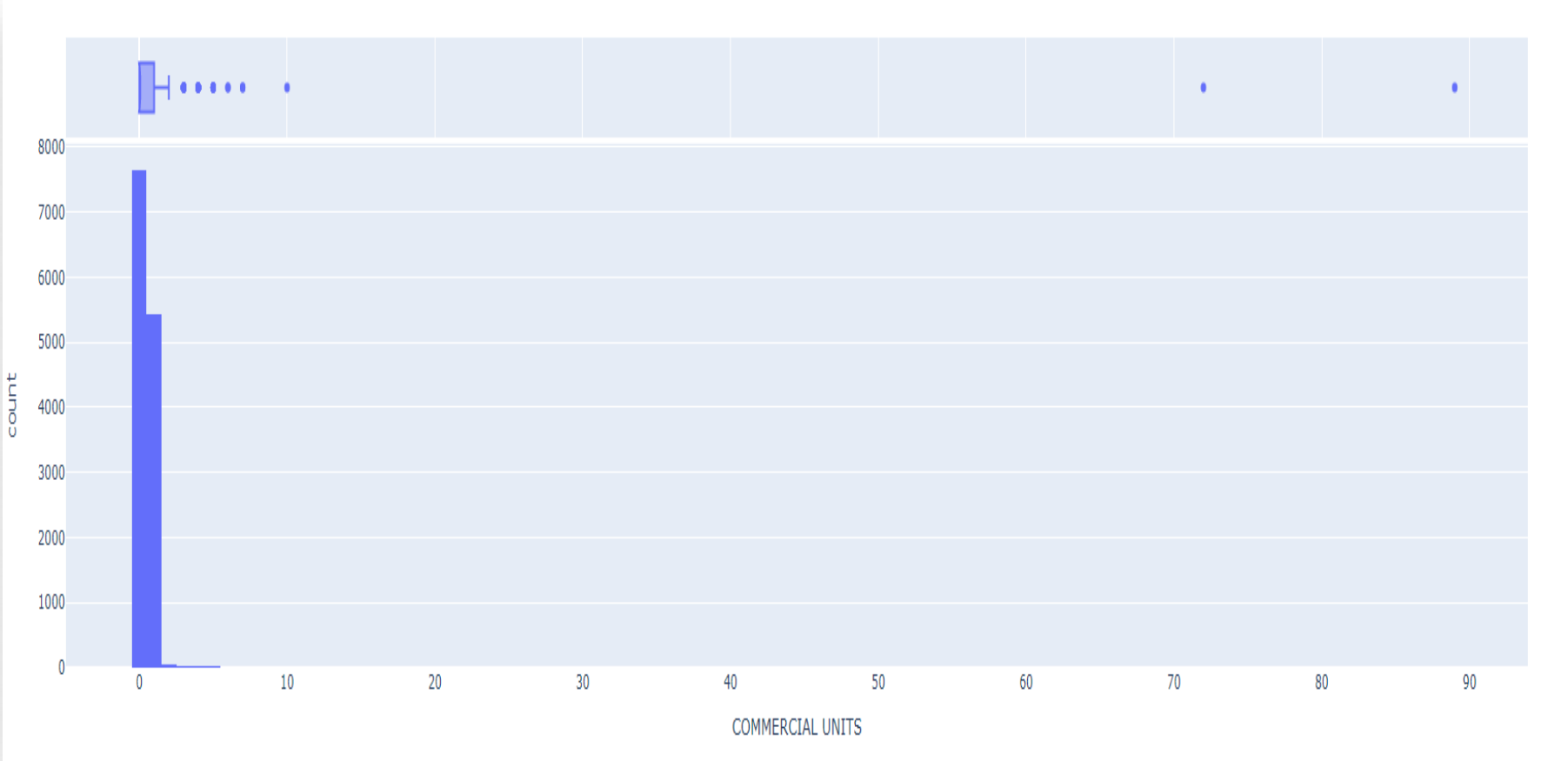
YEAR BUILT *COLUMN*

After Removing Outliers



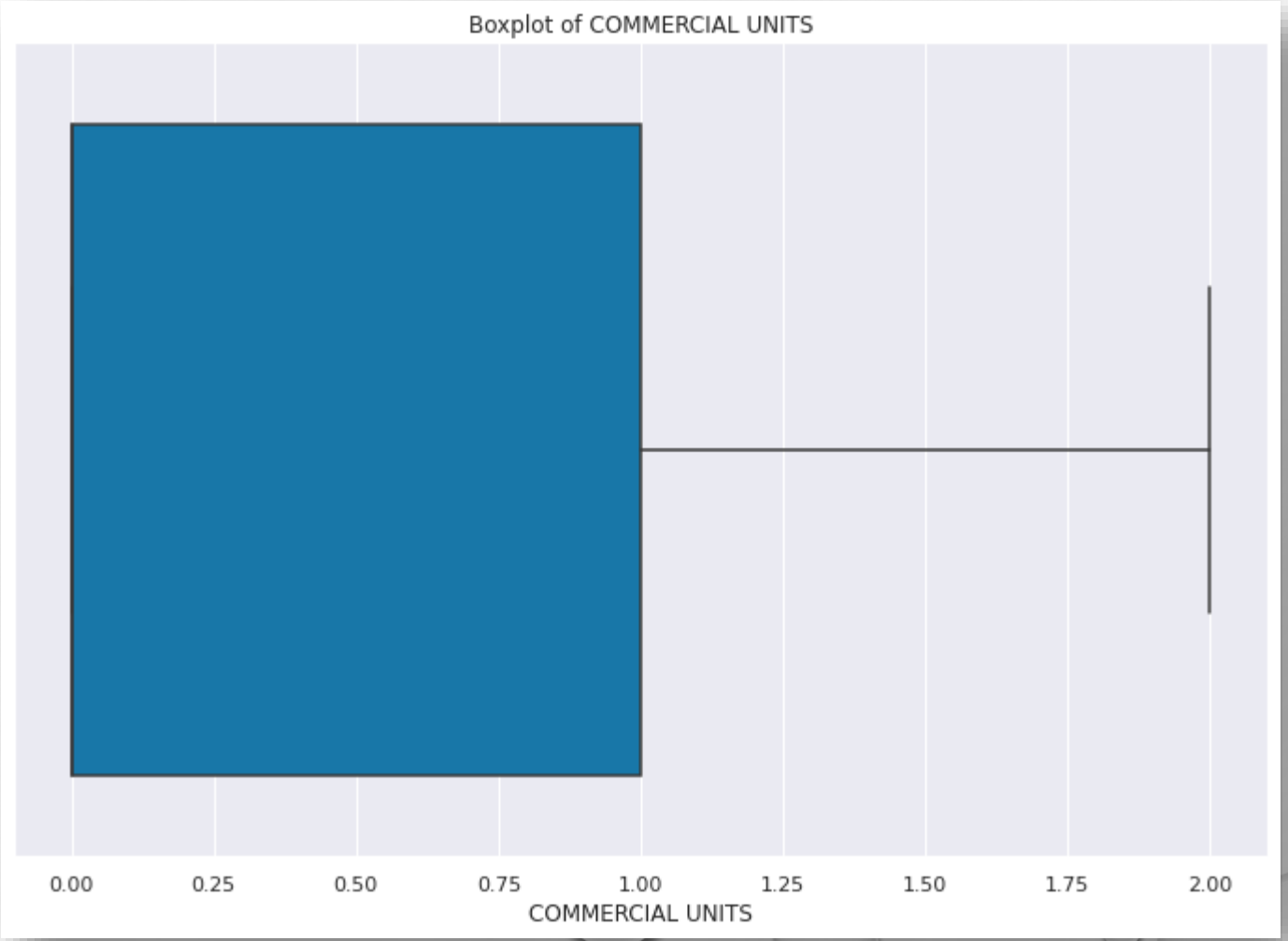
COMMERCIAL UNITS *COLUMN*

This column have ’11’ unique values



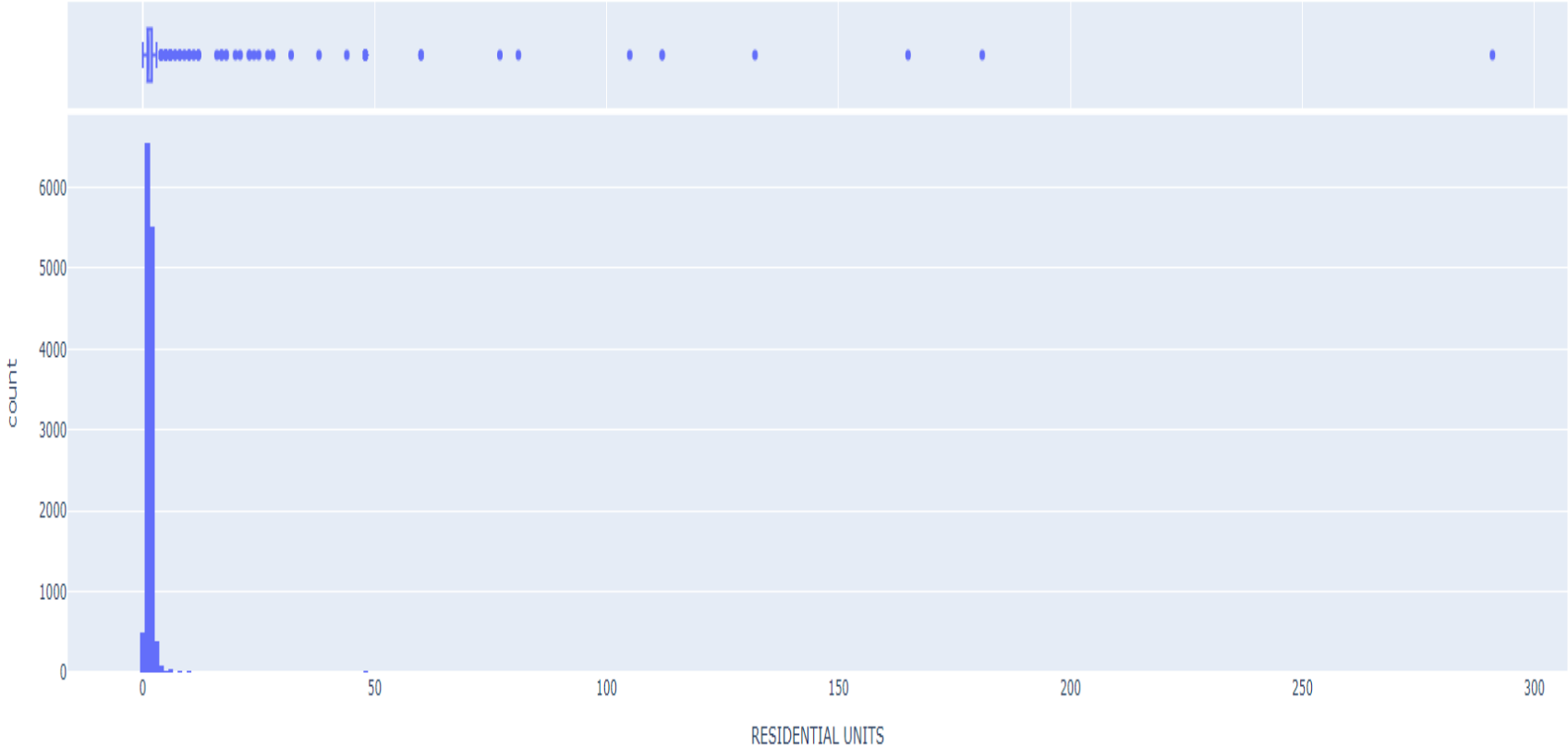
COMMERCIAL UNITS *COLUMN*

After Removing Outliers



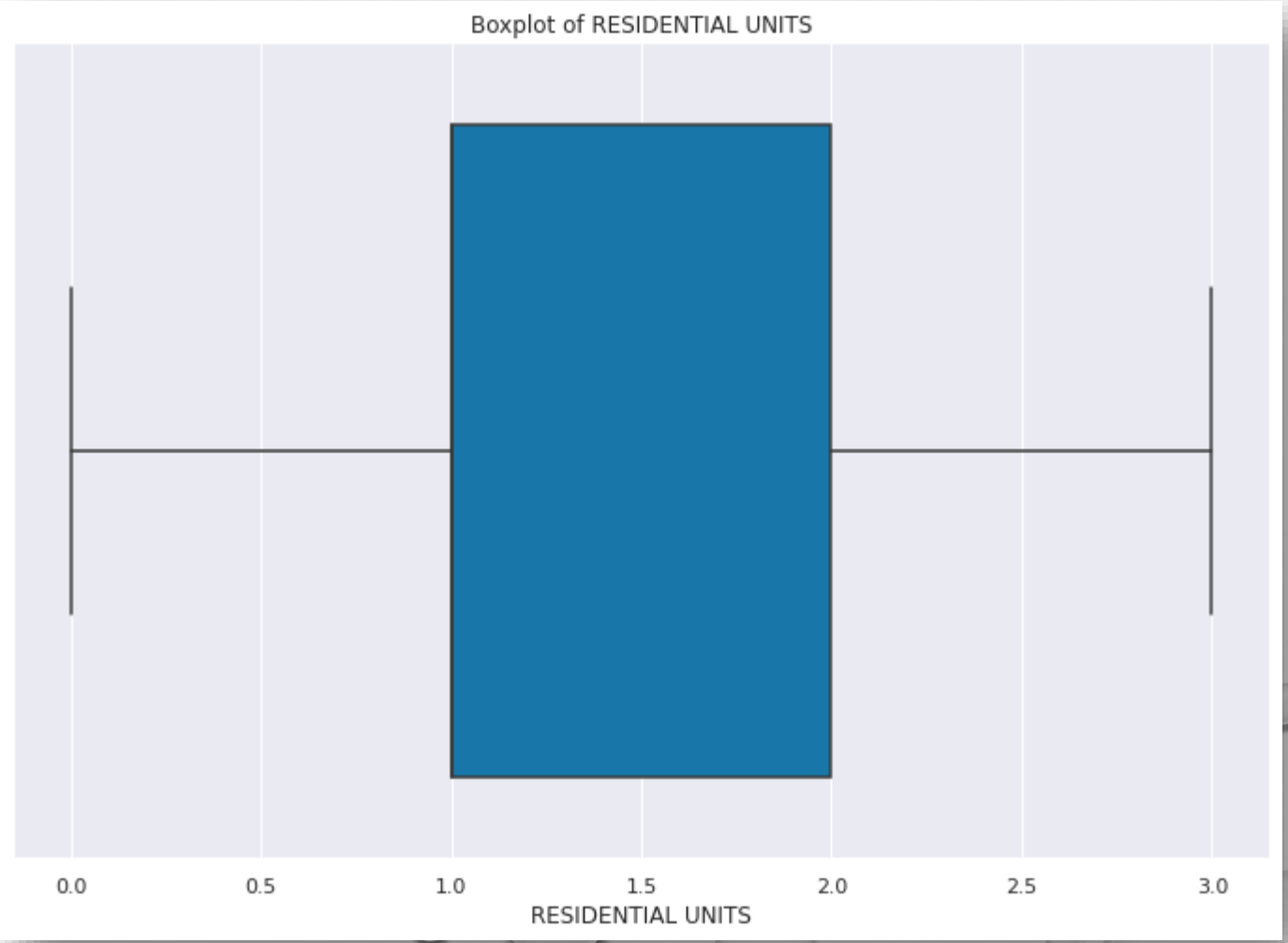
RESIDENTIAL UNITS *COLUMN*

This column have ’36’ unique values



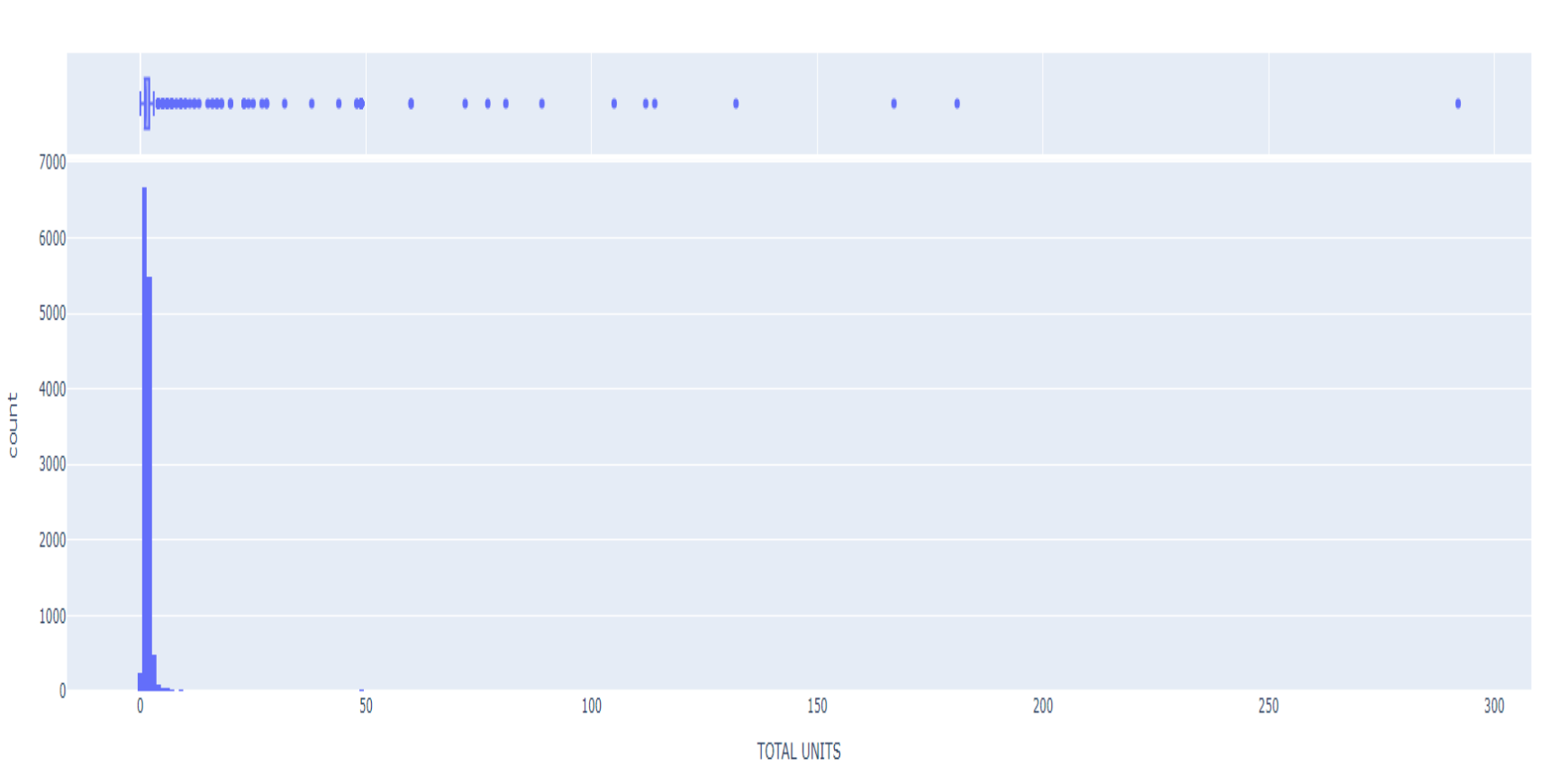
RESIDENTIAL UNITS *COLUMN*

After Removing Outliers



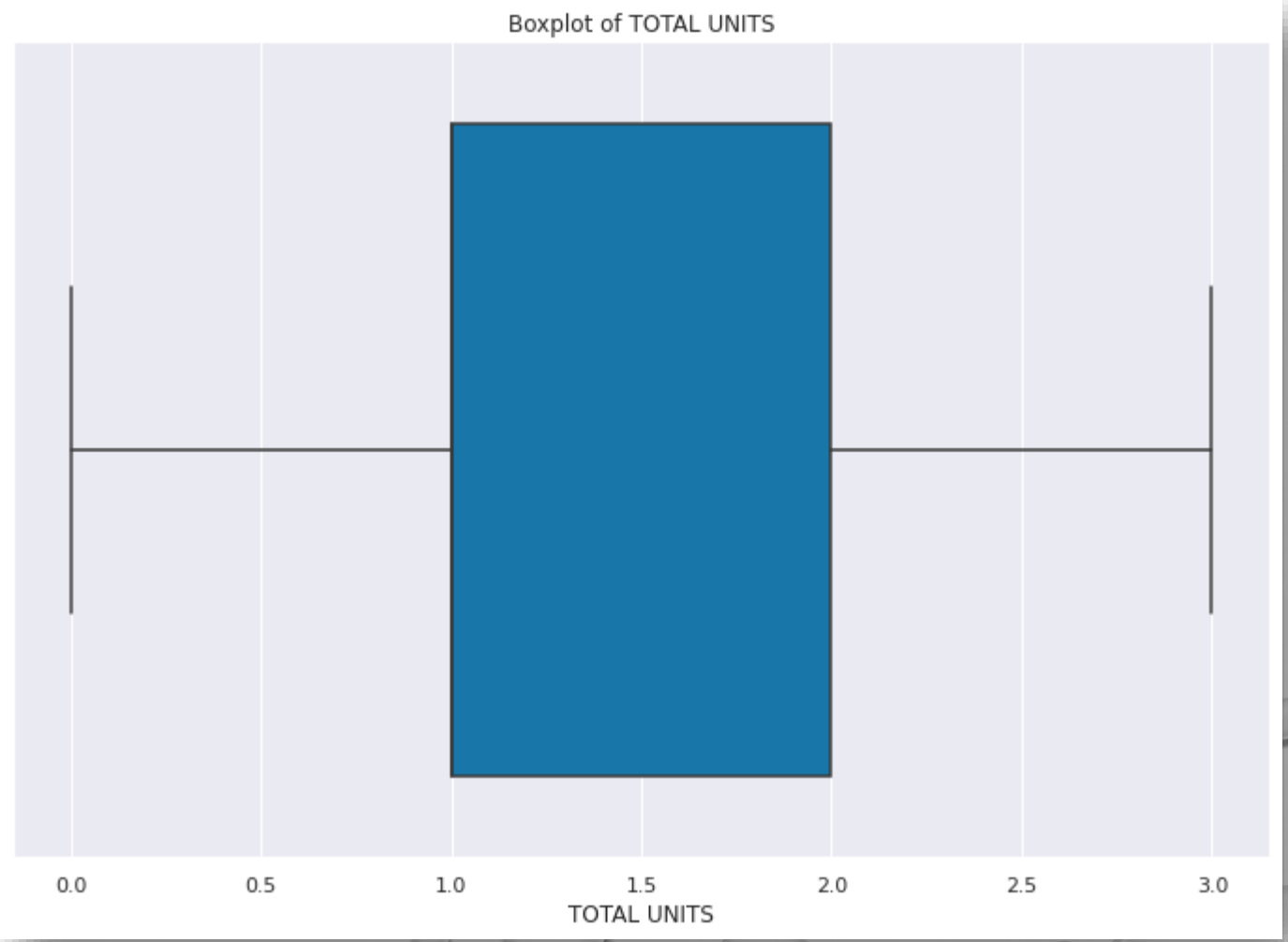
TOTAL UNITS
COLUMN

This column have ’41’ unique values



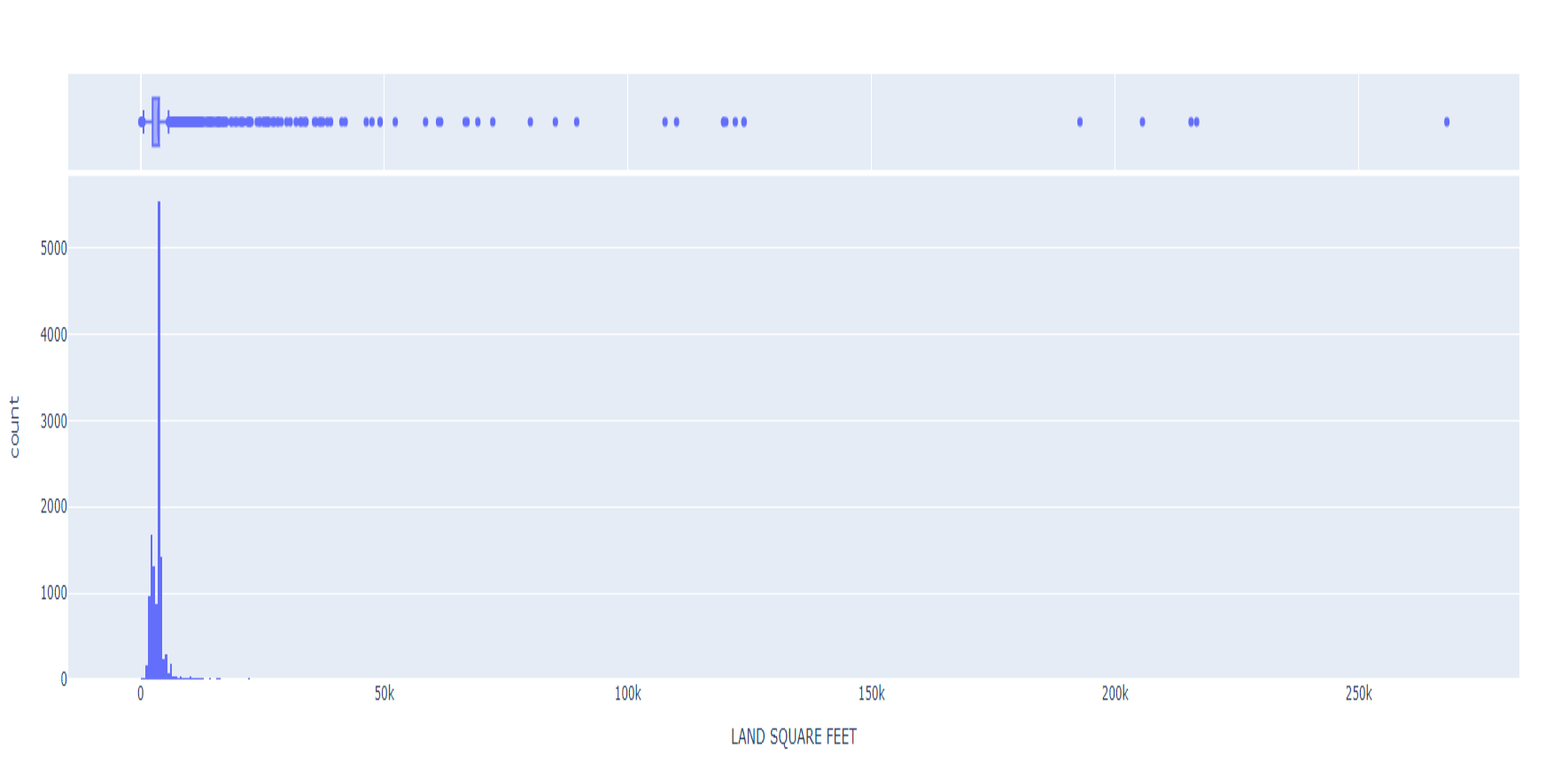
TOTAL UNITS *COLUMN*

After Removing Outliers



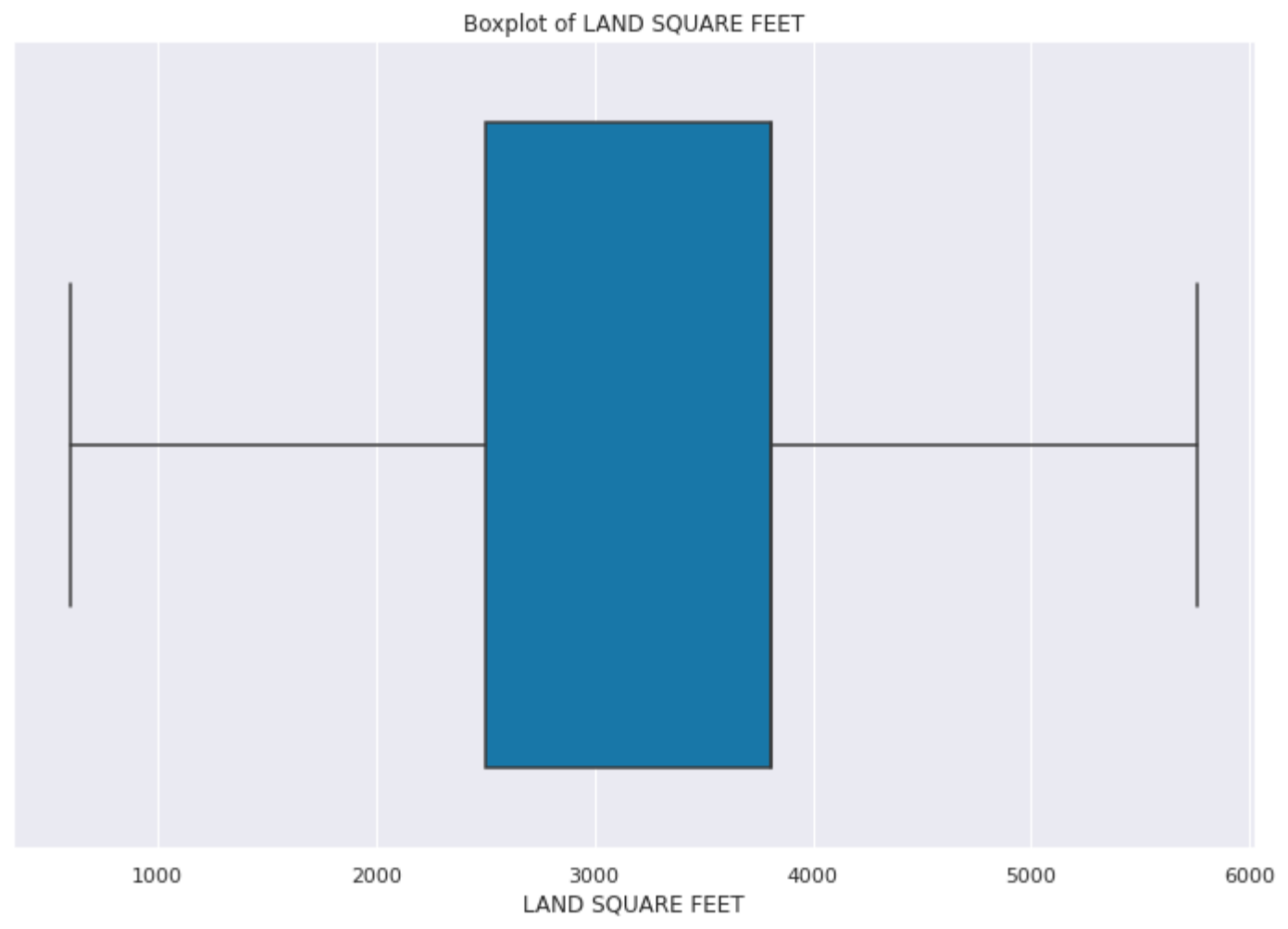
LAND SQUARE FEET *COLUMN*

This column have ’ 1939’ unique values



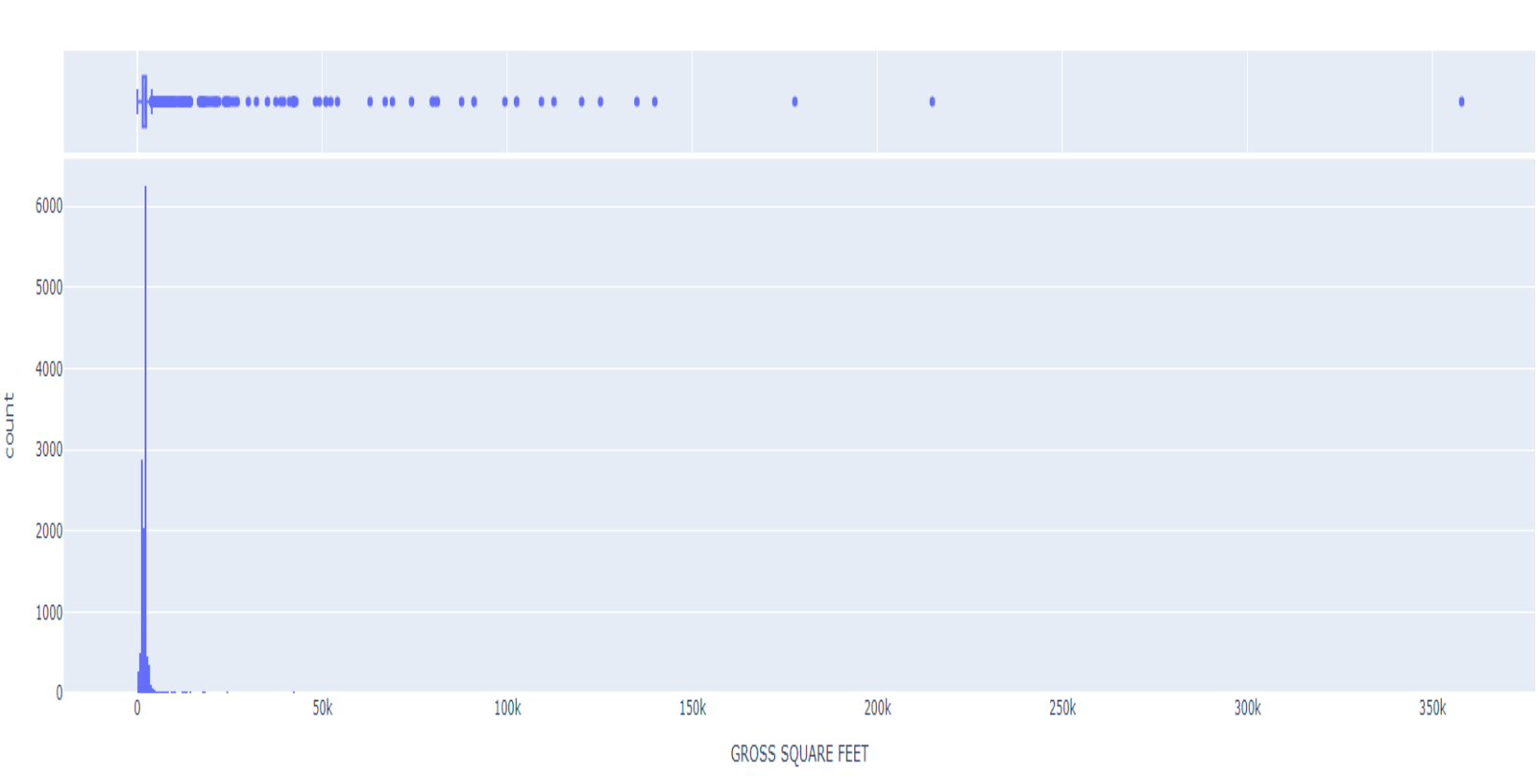
LAND SQUARE FEET *COLUMN*

After Removing Outliers



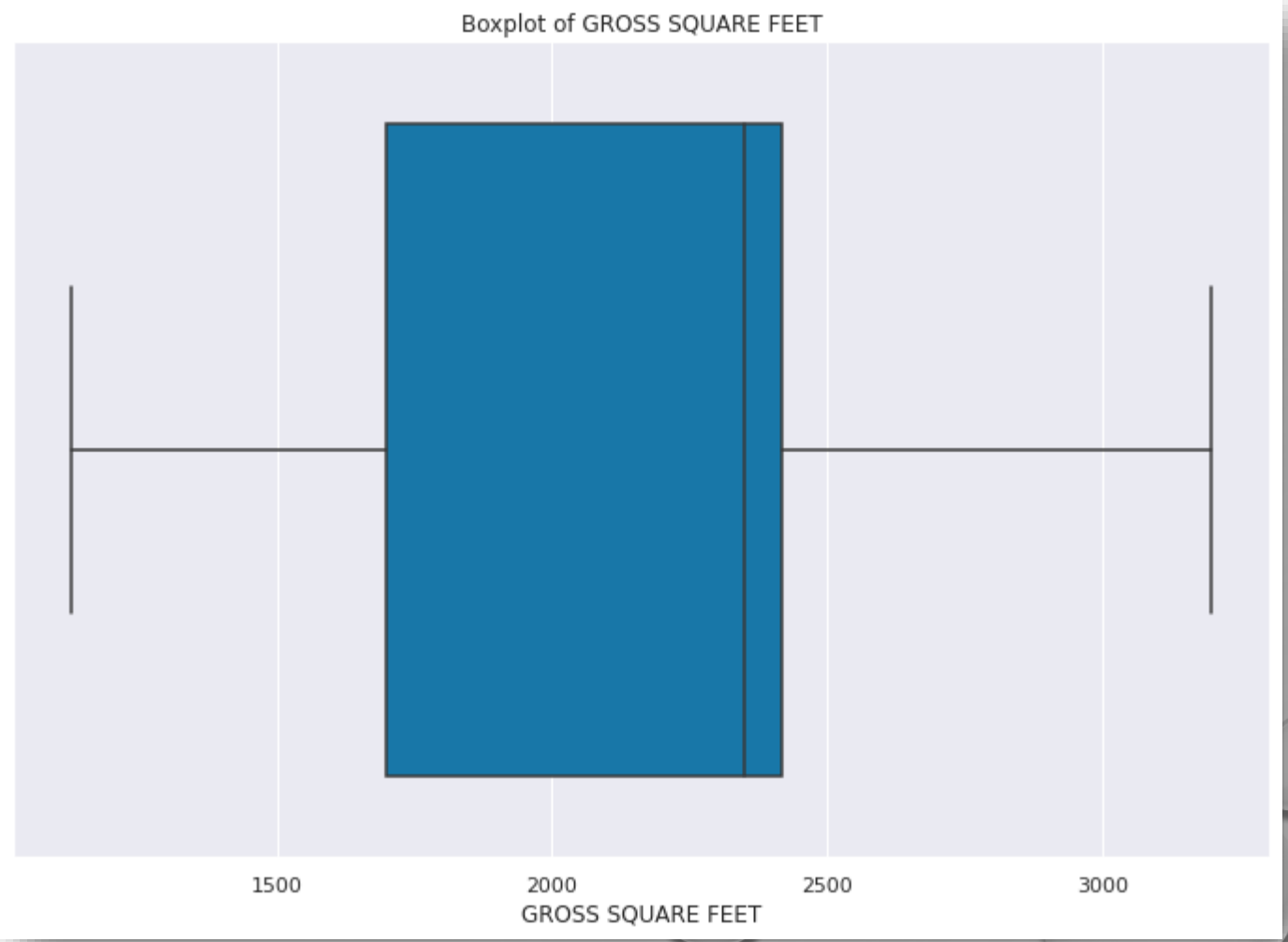
GROSS SQUARE FEET *COLUMN*

This column have ’ 1995’ unique values



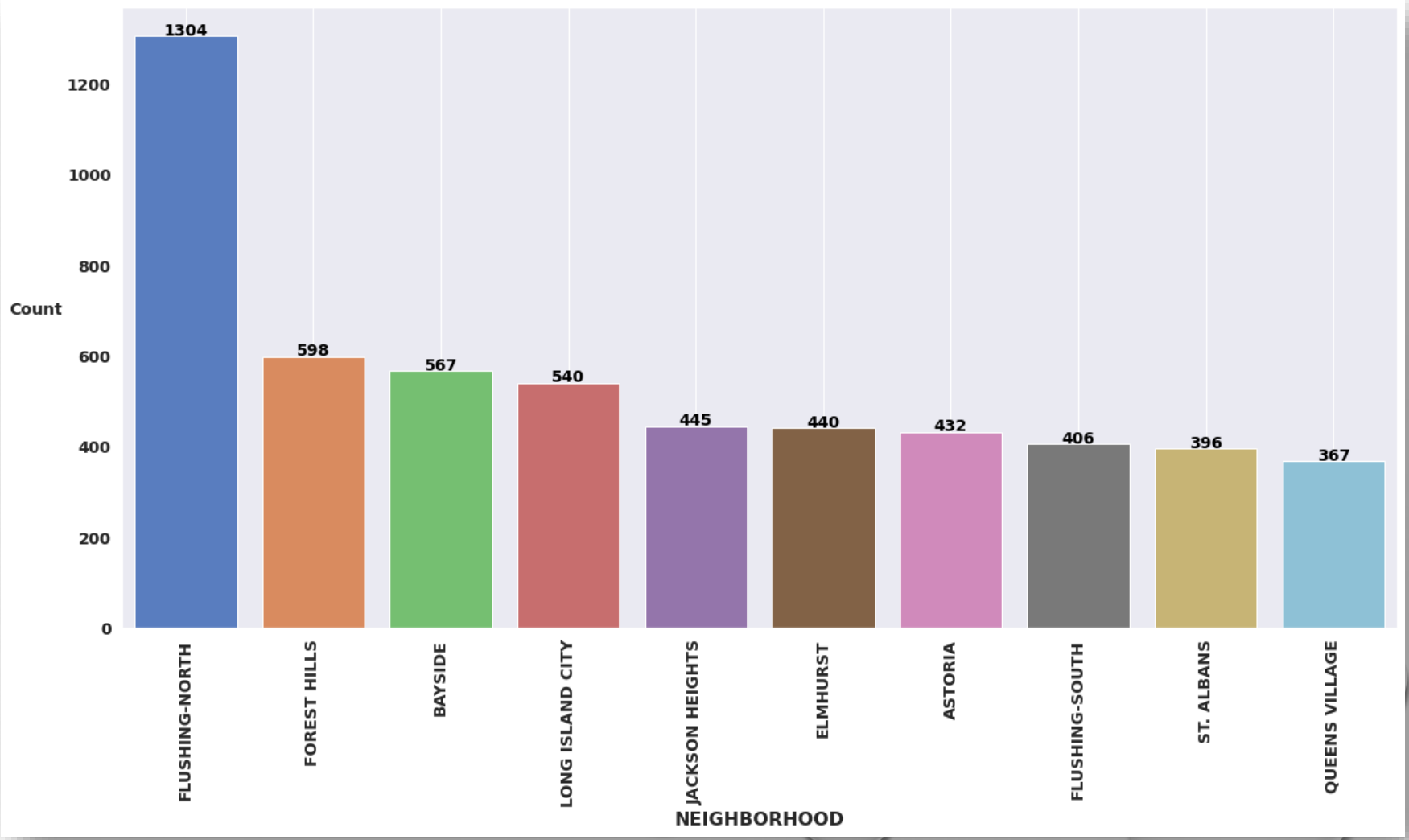
GROSS SQUARE FEET *COLUMN*

After Removing Outliers



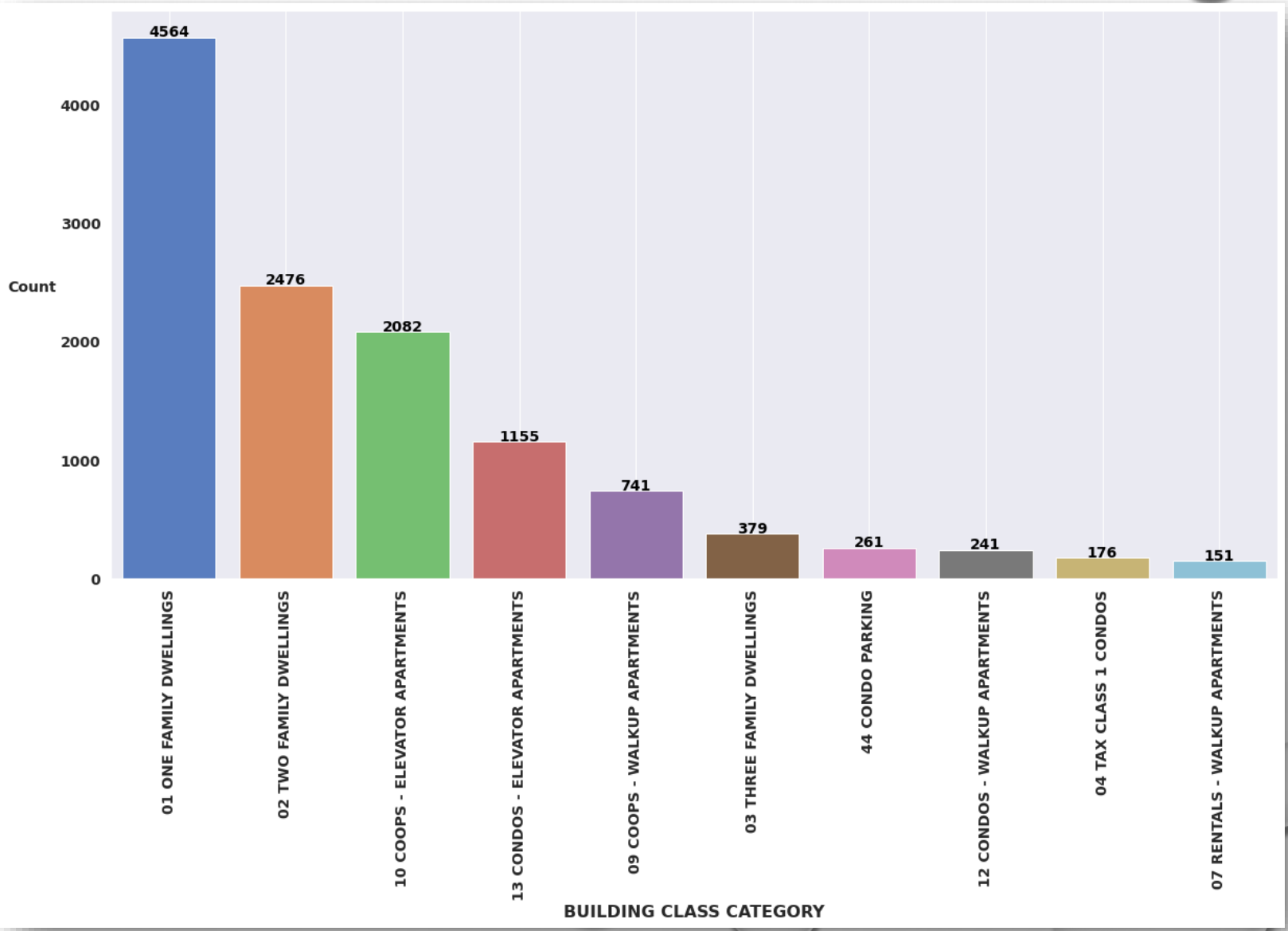
NEIGHBORHOOD
COLUMN

This column have ’ 61’ unique values



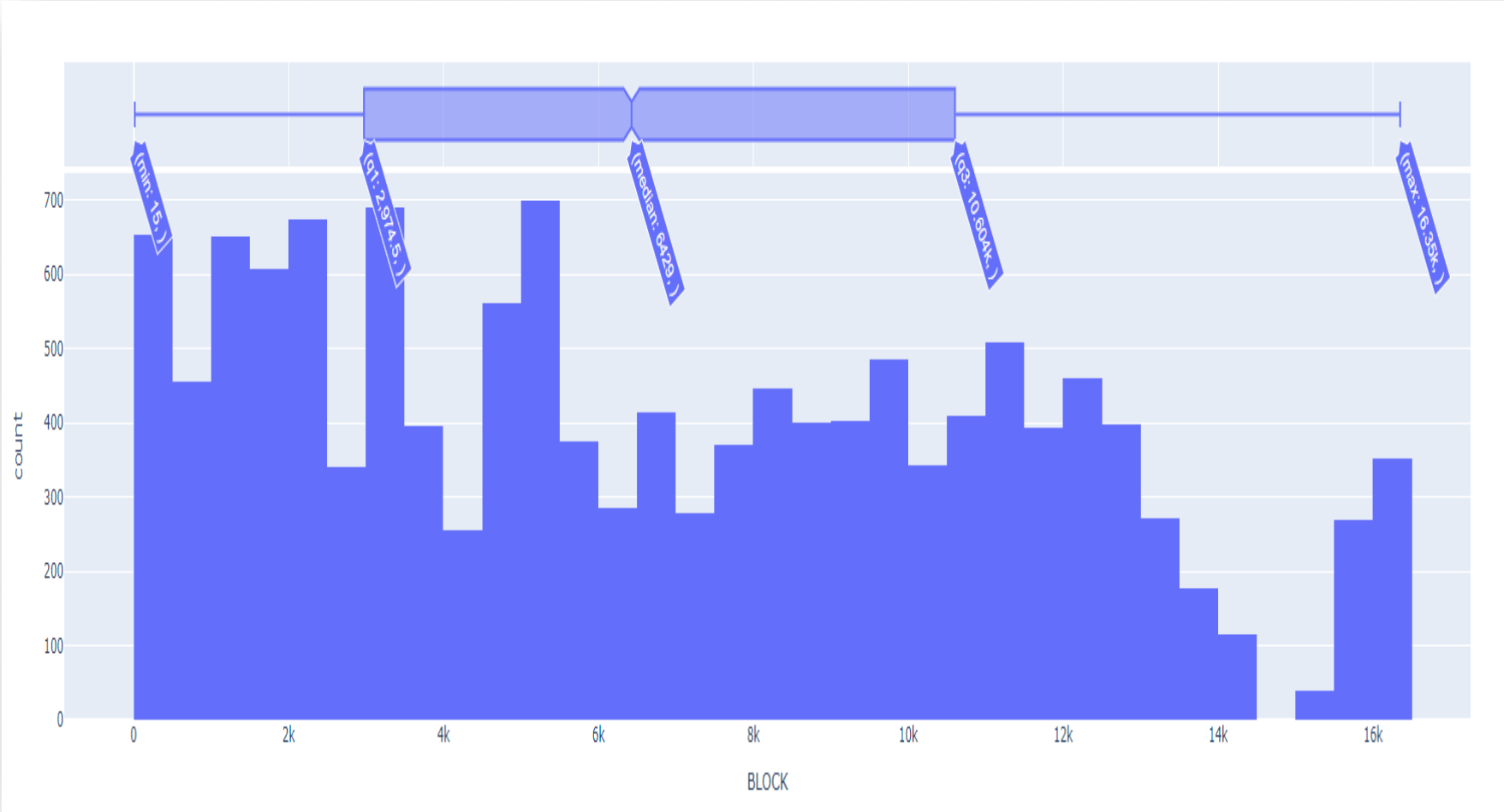
BUILDING CLASS CATEGORY *COLUMN*

This column have ’ 40’ unique values



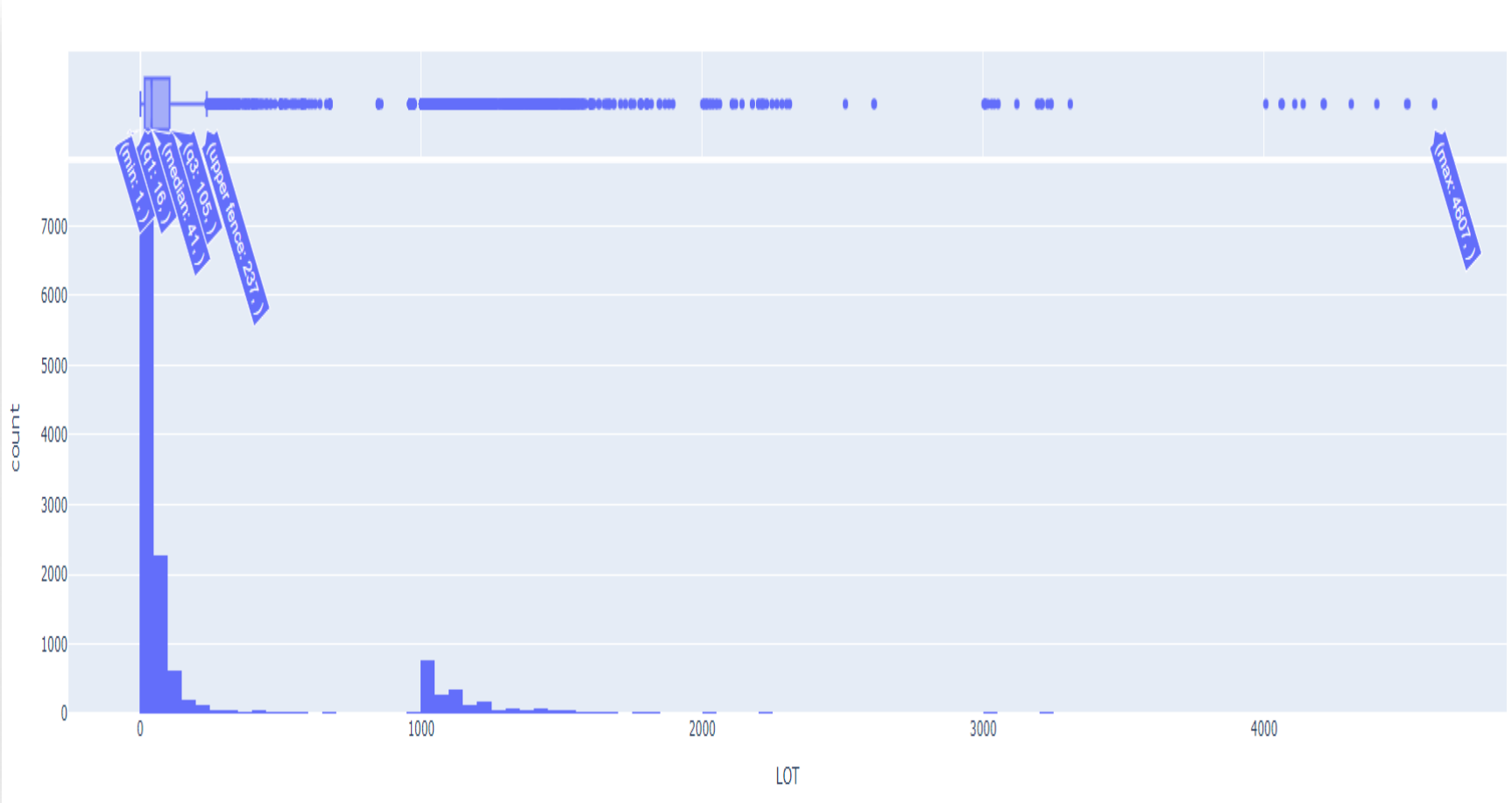
BLOCK
COLUMN

This column have ' 5640' unique values



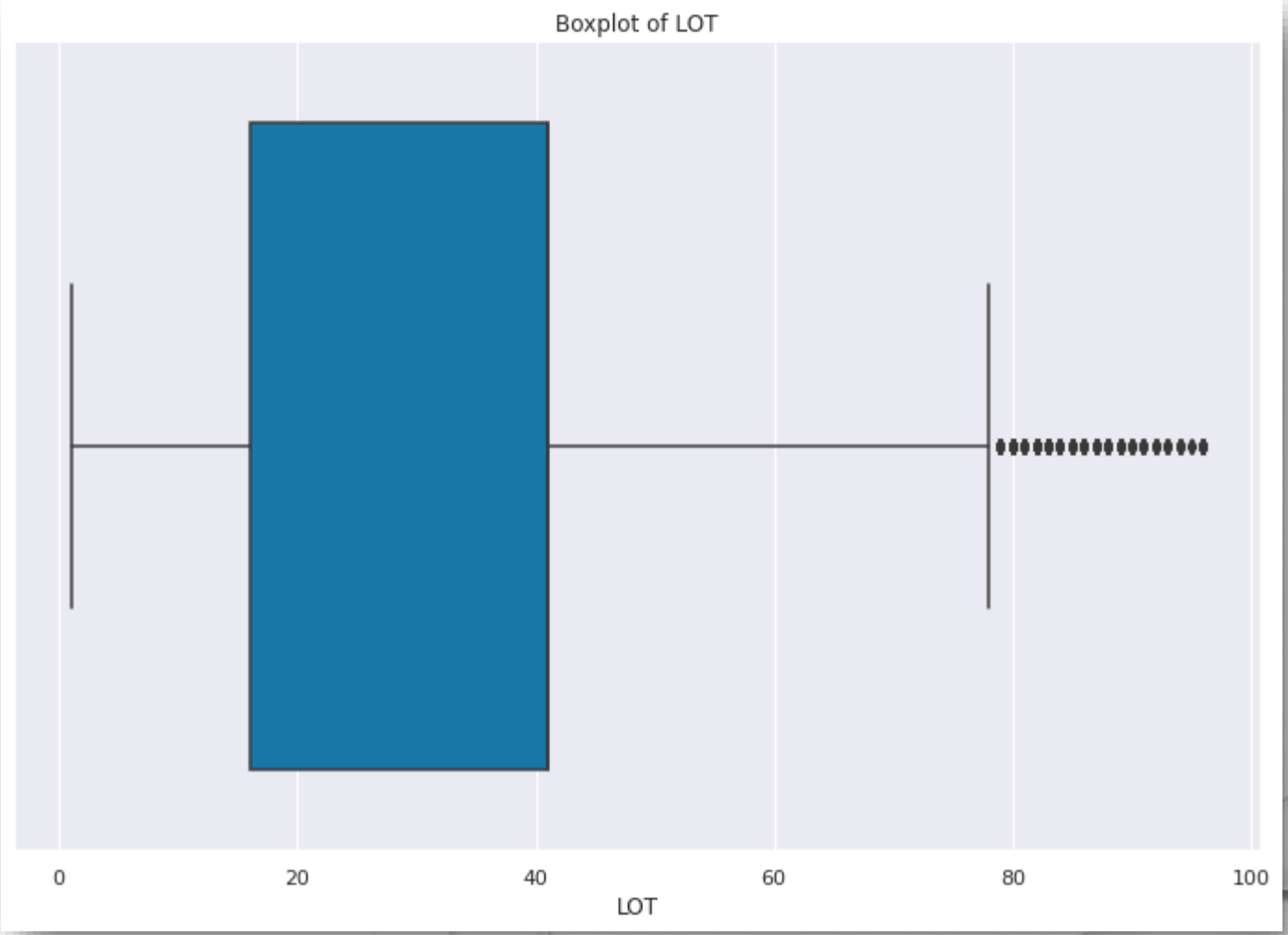
LOT
COLUMN

This column have ’936’ unique values



LOT
COLUMN

After Removing Outliers



AFTER DATA CLEANING

After cleaning the data shape is (13196 X 18) wit zero missing data and zero duplicates

DATA SHAPE AND DATA DUPLICATE

```
df.shape
```

```
(13196, 18)
```

```
#checking for number of duplications
df.duplicated().sum()
```

```
0
```

MISSING VALUES

	Total missing	% missing
NEIGHBORHOOD	0	0.0
BUILDING CLASS CATEGORY	0	0.0
SALE PRICE	0	0.0
BUILDING CLASS AT TIME OF SALE	0	0.0
TAX CLASS AT TIME OF SALE	0	0.0
YEAR BUILT	0	0.0
GROSS SQUARE FEET	0	0.0
LAND SQUARE FEET	0	0.0
TOTAL UNITS	0	0.0
COMMERCIAL UNITS	0	0.0
RESIDENTIAL UNITS	0	0.0
ZIP CODE	0	0.0
ADDRESS	0	0.0
BUILDING CLASS AT PRESENT	0	0.0
LOT	0	0.0
BLOCK	0	0.0
TAX CLASS AT PRESENT	0	0.0
SALE DATE	0	0.0

CONVERT CATEGORICAL TO NUMERICAL

```python
df['NEIGHBORHOOD'].unique()
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
       51, 52, 53, 54, 55, 56, 57, 58, 59, 60])
```

NEIGHBORHOOD COLUMN

WE HAVE 61 UNIQUE VALUES SO I WILL USE LABEL ENCODING

```
df['BUILDING CLASS CATEGORY'].unique()
```

```
array([1.07119106e+06, 6.80915232e+05, 7.05560320e+05, 8.04024502e+05,
       1.46681696e+06, 5.43624875e+05, 7.36760568e+05, 8.15593848e+05,
       2.87128080e+06, 1.30617085e+07, 2.93046773e+05, 3.78286535e+06,
       1.64690666e+06, 4.35025074e+06, 3.63751888e+05, 3.65811051e+05,
       2.03148668e+06, 4.79897241e+06, 7.85031617e+05, 9.95269369e+05,
       3.44457108e+05, 4.05539062e+06, 4.29079622e+06, 1.29618308e+07,
       4.89014638e+06, 2.21389886e+07, 6.74808525e+05, 3.79724360e+06,
       8.66689568e+05, 2.82681818e+06, 1.43824172e+06, 9.88441766e+05,
       4.21752751e+06, 4.77072733e+06, 2.35699527e+06, 8.20588678e+05,
       2.21765594e+08])
```

BUILDING CLASS CATEGORY COLUMN

WE HAVE 40 UNIQUE VALUES SO I WILL USE TARGET ENCODING

	NEIGHBORHOOD	BUILDING CLASS CATEGORY	BLOCK	LOT	BUILDING CLASS AT PRESENT	ADDRESS	ZIP CODE	RESIDENTIAL UNITS	COMMERCIAL UNITS	TOTAL UNITS	...	TAX CLASS AT PRESENT_1	TAX CLASS AT PRESENT_1A	TAX CLASS AT PRESENT_1B	TAX CLASS AT PRESENT_1C	TAX CLASS AT PRESENT_1D	TAX CLASS AT PRESENT_2	TAX CLASS AT PRESENT_2A	TAX CLASS AT PRESENT_2B	TAX CLASS AT PRESENT_2C	TAX CLASS AT PRESENT_4
0	0	1.071191e+06	14260.0	80.0	V7	N/A ROCKAWAY BLVD	11434.0	0.0	0.0	0.0	...	0	0	0	0	0	0	0	0	0	1
1	1	6.809152e+05	949.0	41.0	A5	19-62 81ST STREET	11370.0	1.0	0.0	1.0	...	1	0	0	0	0	0	0	0	0	0
2	1	7.055603e+05	949.0	41.0	R2	19-37 80TH STREET, 1	11370.0	1.0	1.0	1.0	...	0	0	0	0	0	1	0	0	0	0
3	2	6.809152e+05	15830.0	20.0	A5	3-20 BEACH 41 STREET	11691.0	1.0	0.0	1.0	...	1	0	0	0	0	0	0	0	0	0
4	2	6.809152e+05	15830.0	27.0	A5	336 BEACH 41ST STREET	11691.0	1.0	0.0	1.0	...	1	0	0	0	0	0	0	0	0	0

5 rows × 27 columns

TAX CLASS AT PRESENT COLUMN

WE HAVE 10 UNIQUE VALUES SO I WILL USE ONE HOT ENCODING

```
array([1.07119106e+06, 6.55017699e+05, 7.03467908e+05, 6.54164930e+05,
       5.75475624e+05, 3.18420526e+05, 6.71584223e+05, 7.81046031e+05,
       7.19246705e+05, 8.05870131e+05, 9.36104270e+05, 1.46681696e+06,
       5.40346045e+05, 7.30948638e+05, 7.68483069e+05, 2.61721500e+06,
       1.58352370e+07, 2.93046773e+05, 5.00979523e+06, 8.77905444e+05,
       1.11270324e+06, 3.15834054e+06, 1.37213978e+06, 3.99740453e+06,
       1.03768005e+06, 3.14667285e+06, 9.54398516e+06, 3.63751888e+05,
       3.65811051e+05, 2.02510568e+06, 1.92317696e+06, 1.01145249e+07,
       1.38100718e+06, 1.46589008e+06, 7.83750744e+05, 9.95269369e+05,
       3.44457108e+05, 2.33788177e+06, 3.33571537e+06, 5.93594138e+06,
       3.70643700e+06, 2.45606809e+06, 6.25185085e+06, 6.05738465e+06,
       4.33528057e+07, 5.93092766e+06, 5.56084765e+06, 1.21906721e+06,
       6.30029412e+06, 1.58849393e+07, 4.85844685e+05, 7.27420080e+05,
       5.02311623e+05, 3.79724360e+06, 8.66689568e+05, 7.41318519e+05,
       1.47037090e+06, 3.21066489e+06, 5.29618042e+05, 1.43824172e+06,
       1.15307046e+06, 7.21414357e+05, 1.65618946e+07, 1.55645450e+07,
       9.88441766e+05, 1.47605784e+06, 8.14379277e+05, 2.65963407e+06,
       2.86403203e+06, 3.24120044e+07, 2.27268132e+06, 1.71727218e+06,
       1.27523605e+06, 3.13898399e+06, 2.50584122e+06, 2.48126338e+06,
       8.20588678e+05, 2.21765594e+08, 1.09225208e+06, 7.80154219e+06,
       4.26449938e+06, 7.67734982e+05])
```

BUILDING CLASS AT PRESENT COLUMN

WE HAVE 97 UNIQUE VALUES SO I WILL USE TARGET ENCODING

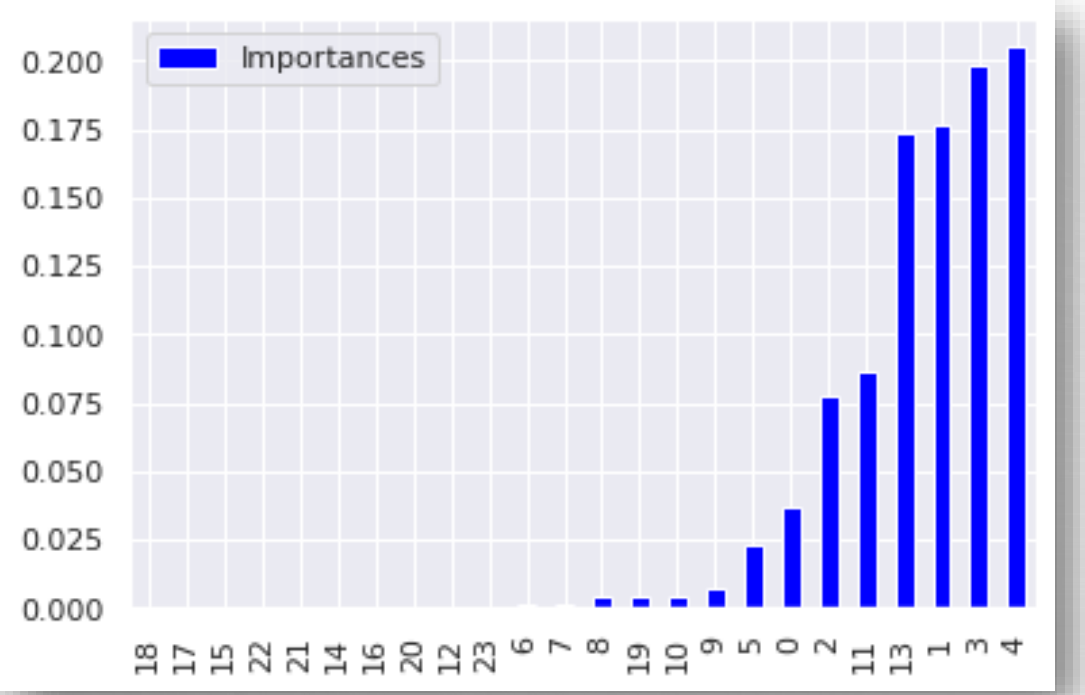
```python
df['BUILDING CLASS AT TIME OF SALE'].unique()
```

```
array([1.07119106e+06, 6.54428664e+05, 7.05560320e+05, 6.57431545e+05,
       5.75475624e+05, 3.18420526e+05, 6.73466723e+05, 7.78490083e+05,
       7.12768355e+05, 8.08469000e+05, 9.37696267e+05, 1.46681696e+06,
       5.41028446e+05, 7.37006657e+05, 7.68483069e+05, 2.87128080e+06,
       1.58352370e+07, 2.93046773e+05, 3.79976187e+06, 8.81205270e+05,
       1.11526593e+06, 3.15834054e+06, 1.37213978e+06, 3.99740453e+06,
       1.03768005e+06, 3.14667285e+06, 9.54398516e+06, 3.63751888e+05,
       3.65811051e+05, 2.03148668e+06, 1.92317696e+06, 1.01145249e+07,
       1.38100718e+06, 1.46589008e+06, 7.85031617e+05, 9.95269369e+05,
       3.44457108e+05, 2.33788177e+06, 3.33571537e+06, 5.84092362e+06,
       3.70643700e+06, 2.45606809e+06, 6.25185085e+06, 6.05738465e+06,
       4.33528057e+07, 5.93092766e+06, 5.56084765e+06, 1.21906721e+06,
       6.30029412e+06, 1.58849393e+07, 4.85844685e+05, 7.27420080e+05,
       5.02311623e+05, 3.79724360e+06, 8.66689568e+05, 7.47527103e+05,
       1.47037090e+06, 2.93294097e+06, 5.29618042e+05, 1.43824172e+06,
       1.15307046e+06, 7.21414357e+05, 1.65618946e+07, 1.55645450e+07,
       9.88441766e+05, 1.47605784e+06, 8.14379277e+05, 4.77072733e+06,
       2.65963407e+06, 2.86403203e+06, 3.24120044e+07, 2.27268132e+06,
       1.71727218e+06, 1.27523605e+06, 3.13898399e+06, 2.50584122e+06,
       2.48126338e+06, 8.20588678e+05, 2.21765594e+08, 1.09225208e+06,
       7.80154219e+06, 4.26449938e+06, 7.67734982e+05])
```

BUILDING CLASS AT TIME OF SALE COLUMN

WE HAVE 98 UNIQUE VALUES SO I WILL USE TARGET ENCODING

FEATURE SELECTION




```
#take the names of the best features
imp_features = final_df['Features'][14:].values
imp_features
```

```
array(['GROSS SQUARE FEET', 'LAND SQUARE FEET', 'ZIP CODE',
       'NEIGHBORHOOD', 'BLOCK', 'YEAR BUILT',
       'BUILDING CLASS AT TIME OF SALE', 'BUILDING CLASS CATEGORY', 'LOT',
       'BUILDING CLASS AT PRESENT'], dtype=object)
```

RANDOMFORESTREGRESSOR FOR SELECTION

```python
#splite the data into Train and Test by 70% for training to 30% for testing
xTrain, xTest, yTrain, yTest = train_test_split(X, Y, test_size = .3, random_state = 42, shuffle = True)
print(xTrain.shape)
print(xTest.shape)
print(yTrain.shape)
print(yTest.shape)
```

```
(9237, 13)
(3959, 13)
(9237,)
(3959,)
```

TRAIN , TEST SPLIT

WE SPLIT THE DATA TO 70% FOR TANNING AND 30% FOR TESTING

PREDICTIONS AND EVALUATION

<u>Mean Absolute Error (MAE)</u>: Every residual (distance between fitted line and observation) needs to be calculated for every data point, taking only the absolute value of each. Then the average of the residuals is the MAE.

<u>Mean Squared Error (MSE):</u> The MSE is just like the MAE, but squares the difference before summing them all instead of using the absolute value. MSE is more popular because it punishes larger which tend to be useful in the real world.

<u>R2:</u> In regression, R2 a statistical measure of how well the regression predictions approximate the real data points.

MODELS RESULTS

Algorithm	MAE	MSE	RMSE	R^2
RandomForestRegressor Model	**0.044**	**0.0225**	**0.4749**	**0.782**
Linear Regression Model	0.1526	1.481	1.217	-0.4303
KNN MODEL	0.0937	1.068	1.0336	-0.031
SVR MODEL	**0.1035**	**0.971**	**0.9852**	**0.0638**
DecisionTreeRegressor MODEL	0.1058	1.77	1.33	-0.71
GradientBoostingRegressor MODEL	0.1108	1.673	1.293	-0.615
AdaBoostRegressor MODEL	0.295	1.925	1.387	-0.817
XGBRegressor MODEL	0.095	1.632	1.277	-0.576

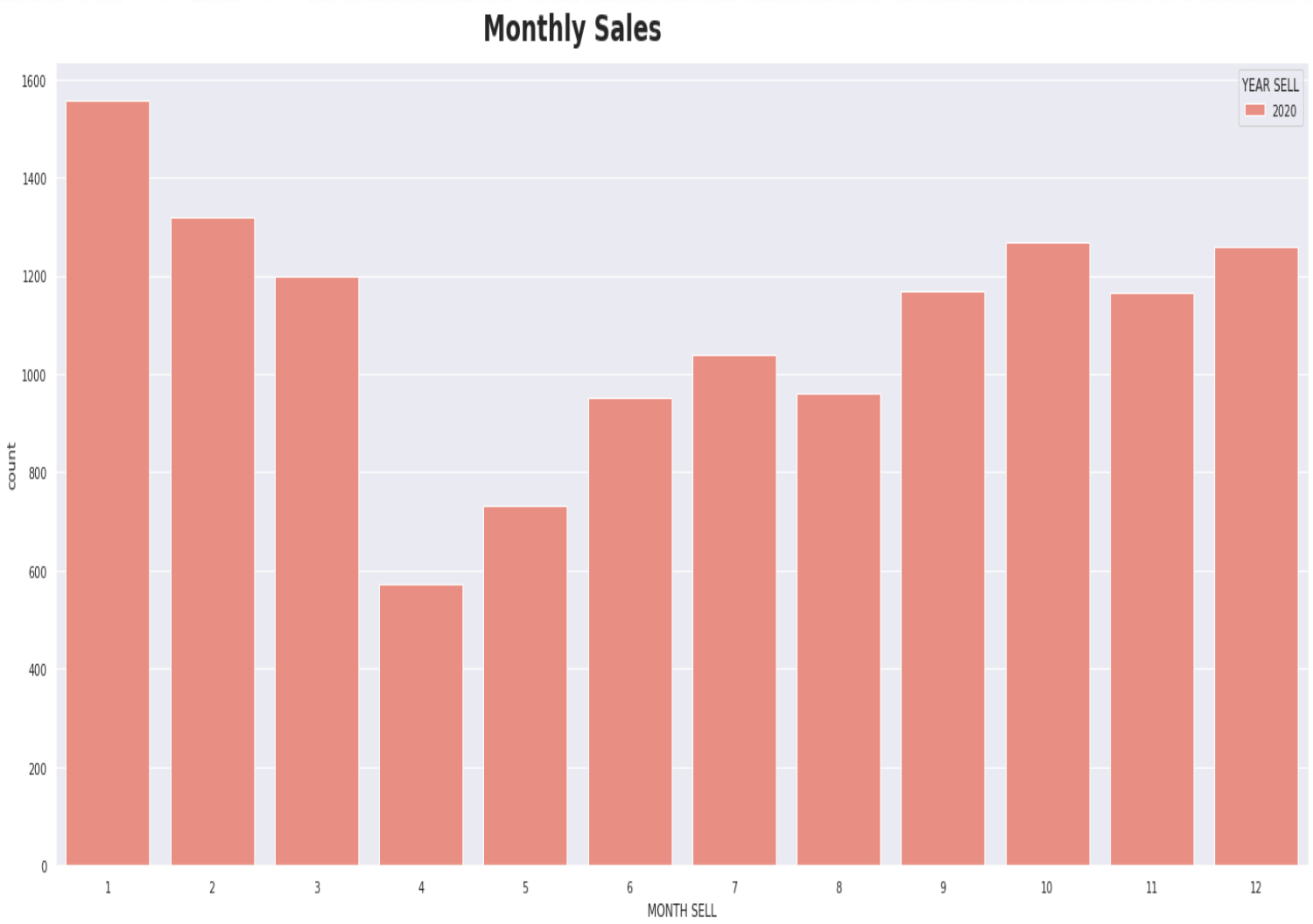
- AS WE CAN SEE, THE BEST TWO MODELS ARE RANDOM FOREST AND SVR

RESEARCH QUESTIONS:

what are the number of units are sold for each month?

What is the new building at the lowest selling price?

WHAT ARE THE NUMBER OF UNITS ARE SOLD FOR EACH MONTH?



So, As we see from the previous results, we can get that:

- The Maximum value is around ‘1600’ Data records ==> [Month: 1]

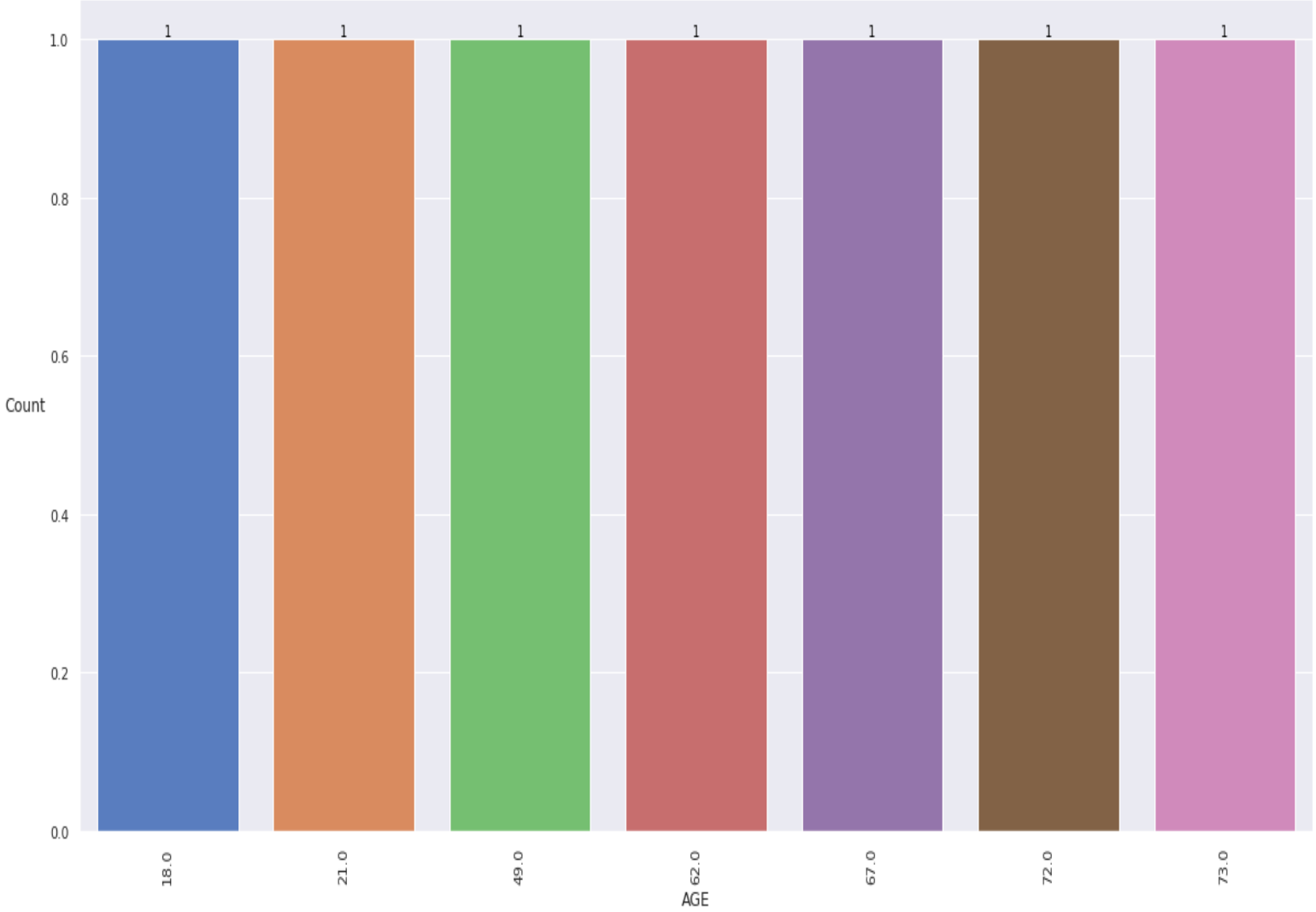
- The Minimum value is around ‘600’ Data ==> [Month: 4]

This question is important for real estate companies to know the best time of the year to buy buildings to save more money and to know the best time of the year to sell buildings to make a lot of money

And our audience are any company, or any person wants to buy an apartment

WHAT ARE THE NUMBER OF UNITS ARE SOLD FOR EACH MONTH?

Also, We can see that the newest building with the lowest selling price is the one that was built “18” years ago with a selling price of “1”



This question is very important for everyone because all people are looking for new homes with low budget

And our audience is the one who wants to buy a new building or apartment

THANK YOU