NYC Land price prediction

planckie_vx

29/12/2020

Introduction

This project is part of the capstone project of HarvardX's Data Science Professional Certificate[1] program. This part of capston project consists of picking a prediction problem with data collected from the web. In this case, a price prediction problem has been chosen. The aim of the project is an algorithm which predicts the price of real state in NYC.

Instalation of packages

We install the requuired packages and clean the environment:

Data downloading

The data used for the project has been downloadad from kaggle. The dataset is called "NYC Property Sales. A year's worth of properties sold on the NYC real state market.

NYC Property sales Dataset descripction. Research

Location

Firstly, the dataset includes the borough, neighbourhood, zip code and address of each property. This data allows us to locate the property.The borough is an important aspect in NYC as there are high difference of standard of living in each borough. We expect higher prices in borough number 1 (MANHATTAN) which concentrates the richest buildings both residential and commercial. The other districts are Bronx (2), Brookly (3), Queens (4) and Staten Island (5) [2]

Type of building

Secondly, the dataset provides us with information regarding the type of building, such as the NYC classfication (building class) at present and at time of sale; and number of commercial, residential and total units saled in the lot. The building class

---

[1] https://www.edx.org/professional-certificate/harvardx-data-science

[2] https://en.wikipedia.org/wiki/Boroughs_of_New_York_City

classification of NYC[3] is a complex system with two levels of classification with a letter and an number. The letter is the general classification and the number is the subset. In general, it can be said that the first letter classes (A-E) correspond to residential despite having some other residential classes with latter letters. Rest of letters classify for commercial, equipment or industrial. It seems that this information would be usful for our study. Nevertheless, the classfication seems to be really complex to get insights.

We could include in this information, the tax class.Tax class may be related to prices. This correlation must be checked once we work with the real data[4]

Dates.

The dates provided are the date of the sale (between 2016 and 2017) and the year of construction of the building (from 1800 to 2016). These dates may permit us to know the evolution of prices during the years 2016 and 2017 and reagarding year built, this information may be misleading. One could think that newer states may be more expensive. However, NYC is an special city with problems to develop new buildings in really crowded areas. This means that in some areas as Manhattan (1) the sale of old buildings could be really expensive as there is no more availabe lands in the sorrounds. Then, the evolution of price in relation with year of building must be interpreted carefully.

Sizes and prices

The information of size is provided with two variables: gross square feet and land square feet. Land square feet is the direct measurement of the area of the state whereas Gross square feet is an abstract measure of the usable area from bottom to top. When gross square feet is close to land square feet, the building is low and the usable are will be close to the area of the state. Nevertheless, when gross square feet is much higger than land square feet, this means that the building is high and each floor has usael area. For our study, the relation between gross square feet and land square feet is really usfeul because the usability o the land may be a really important variable to predict the prices. Those states with low gross square feet, then, low usability, may be chepaer. We will define the variable floor area ratio (FAR)[5] which is the ratio between gross square feet and land square feet. Now, the higher the FAR, the higher the price of the building.

The price is defined with a simple variable called price of sale.

---

[3] https://www1.nyc.gov/assets/finance/jump/hlpbldgcode.html

[4] http://www.cynthiaramirez.nyc/uploads/1/1/8/7/118704813/class_1_guide__1_.pdf

[5] https://en.wikipedia.org/wiki/Floor_area_ratio

Goals

As already explained, the main aim of the project is the development of an algorithm that allows the prediction of real state prices in NYC. In spite of predicting the price itself, the prediction is going to be based in price per square feet. We have choosen price per square feet because the data set includes residencial, commecial and equipment sales whit really diferent sizes. Therefore, the comparison then, will be done more properly with price per square feet, which is besides, the reference value for real state prices. The case would have been different if we had had just residencial or just commercial. In that case we would have been able to compare direclty the prices.

Then, the three variables GROSS SQUARE FEET, LAND SQUARE FEET AND PRICE OF SALE will be summarized in two variables. the predictor FAR (GROSS SQUARE FEET/LAND SQUARE FEET) and the target variable price per square feet (PRICE/LAND SQUARE FEET)

RMSE

We will split the data between train and test data. We will use test data to tune the parameters of the different models. The accuracy of the prediction will be measured with the RMSE. We will compare the prediction with the real prices from the test data.

$$RMSE = \sqrt{\frac{1}{N}\sum(\hat{y} - y)^2}$$

Proccess and workflow

1. The first step will be the preparation of data: download, parse, import and prepare the data to be processed and analysed.
2. The second step will be the exploration of the data. In this part we analyse the relation between the possible predictors and the variable we want to predict. Diffrent type of graphics and some calculation may permit us the identification of the most correlated variables.
3. The next step will be the preparation of the dat focused on the prediction, once we know the relations and the possible models that are going to be implemented.
4. After that, we develop the models based on the previous exploration. We choose or calculate the best tuning parameters for each algorithm and then we test it and validate it.
5. Finally, we analyze the results, which means analyzing the accuracy of each model and the meaning of the predictions. Furthermore, we discuss the utilization of other type of models that are usually used for price prediction challenges.

Data preparation

The data downloaded from kaggle is a csv file that we must convert to data frame:

```
nyc_clean<-as.data.frame(nyc_rolling_sales)
```

Missing data

Once we can work with a tidy data frame, we must assure that prices, land square feet and gross square feet are provided for the sales reproted in the data frame. We delete the sales that do not include any of these variables:

```
nyc_clean$`SALE PRICE`<-as.numeric(nyc_clean$`SALE PRICE`)
nyc_clean<-nyc_clean%>%filter(!is.na(`SALE PRICE`))
nyc_clean$`GROSS SQUARE FEET`<-as.numeric(nyc_clean$`GROSS SQUARE FEET`)
nyc_clean<-nyc_clean%>%filter(!is.na(`GROSS SQUARE
FEET`))%>%filter(!`GROSS SQUARE FEET`==0)
nyc_clean$`LAND SQUARE FEET`<-as.numeric(nyc_clean$`LAND SQUARE FEET`)
nyc_clean<-nyc_clean%>%filter(!is.na(`LAND SQUARE FEET`))%>%filter(!`LAND
SQUARE FEET`==0)
```

The variables ease-ment and apartment number are not provided. So, we just eliminate the column

```
nyc_clean<-nyc_clean%>%select(-`EASE-MENT`,-`APARTMENT NUMBER`)
```

Cleaning repeated data

Until now, we have 19 variables in our data frame. Nevertheless, many of 22 variables mean the same. We are going to eliminate the repeated data. Building class is included as “building class category”, “building class at present” and “building class at thime of sale”. We consider that “building class at thime of sale” is the variable that best suits our aim. We eliminate the other two variables. However, the double classification (letter and number) is too complex for our purpose. Once we check again the official clasification[6] we notice we can take just the general classification (letter).

```
nyc_clean<-nyc_clean%>%select(-`BUILDING CLASS CATEGORY`,-`BUILDING CLASS
AT PRESENT`)
nyc_clean<-nyc_clean%>%mutate(building_class=str_extract(`BUILDING CLASS
AT TIME OF SALE`,"^[A-Z]"))
nyc_clean<-nyc_clean%>%select(-`BUILDING CLASS AT TIME OF SALE`)
```

Apart from building class itself, there are three other variables related: “commercial units”, “residential units” and “total units”. Firstly we should calculate the proportion of commercial or residential units for each sale. Nevertheless, for some of the sales this information is not reported. Furthermore, this classification does not include

---

[6] https://www1.nyc.gov/assets/finance/jump/hlpbldgcode.html

neither industrial nor equipment sales. We have considered that the type of building is better accounted with building_class than with the two hypothetic variables proportion of commercial and proportion of residential. The advantage of building class is a wider representation. The advantage of the proportions is a continuous variable instead of working with a categorical one.

```
sum(is.na(nyc_clean$`TOTAL UNITS`))

## [1] 0

sum(nyc_clean$`TOTAL UNITS`==0)

## [1] 68
```

Some of "total units" are 0. The calculation of the precentages would be problematic (divisor would be 0).

```
nyc_clean_0<-nyc_clean%>%filter(nyc_clean$`TOTAL
UNITS`==0)%>%group_by(building_class)%>%summarize(n=n())
nyc_clean_0

## # A tibble: 15 x 2
##    building_class     n
##    <chr>          <int>
##  1 E                  3
##  2 F                  3
##  3 G                 14
##  4 H                  1
##  5 I                  3
##  6 J                  1
##  7 K                  1
##  8 L                  2
##  9 M                 11
## 10 N                  2
## 11 O                  4
## 12 P                  3
## 13 R                  1
## 14 W                  2
## 15 Z                 17

rm(nyc_clean_0)
```

We analyze that many of these sales are type of building G,M,Z. If we check the classification [7], these buildings correspond mainly to garages, religious buildings and other equipments. So, we can avoid these data o tranform it to percentage 0.00% for both residential and commercial which is close to reality. We are just commiting error in some of these 68 sales which could correspond to commercial types. These types

[7] https://www1.nyc.gov/assets/finance/jump/hlpbldgcode.html

are less than 3 per each class. We decide to continue with this transformation and to keept these sales.

```
nyc_clean<-nyc_clean%>%mutate(com_perce=ifelse(`TOTAL
UNITS`==0,0,`COMMERCIAL UNITS`/`TOTAL UNITS`))
nyc_clean<-nyc_clean%>%mutate(res_perce=ifelse(`TOTAL
UNITS`==0,0,`RESIDENTIAL UNITS`/`TOTAL UNITS`))
nyc_clean<-nyc_clean%>%select(-`RESIDENTIAL UNITS`,`RESIDENTIAL
UNITS`,`TOTAL UNITS`)
nyc_clean<-nyc_clean%>%filter(res_perce<=1)
nyc_clean<-nyc_clean%>%filter(com_perce<=1)
```

We can follow the same reasoning with tax class. We have “tax class at present” and “tax class at time of sale”. We must pick “tax class at time of sale”.

```
nyc_clean<-nyc_clean%>%select(-`TAX CLASS AT PRESENT`)
nyc_clean<-nyc_clean%>%mutate(tax_class=`TAX CLASS AT TIME OF SALE`)
nyc_clean<-nyc_clean%>%select(-`TAX CLASS AT TIME OF SALE`)
```

Unuseful data

The vectors index X1, Neighbourhood, Block, Lot, address, and zip code cannot be related to the price. Therefore we eliminate them.

```
nyc_clean<-nyc_clean%>%select(-ADDRESS,-`ZIP CODE`,-LOT,-BLOCK,-X1, -
NEIGHBORHOOD)
```

Dates

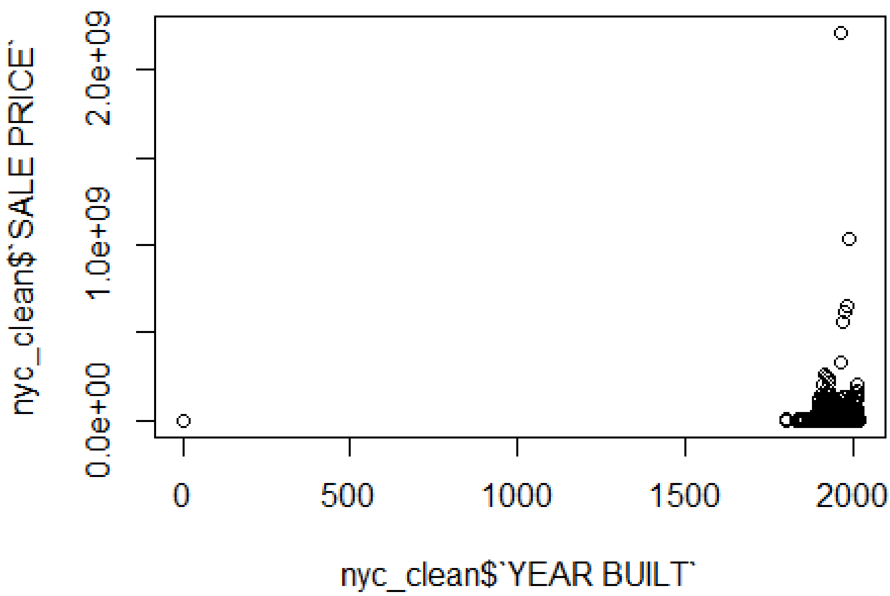
There are two possible date variables to consider the possibe effect of time in price. We must check if data is complete and reasonable.

Sale of date: check if any of the dates are previous to 2016 or NA.

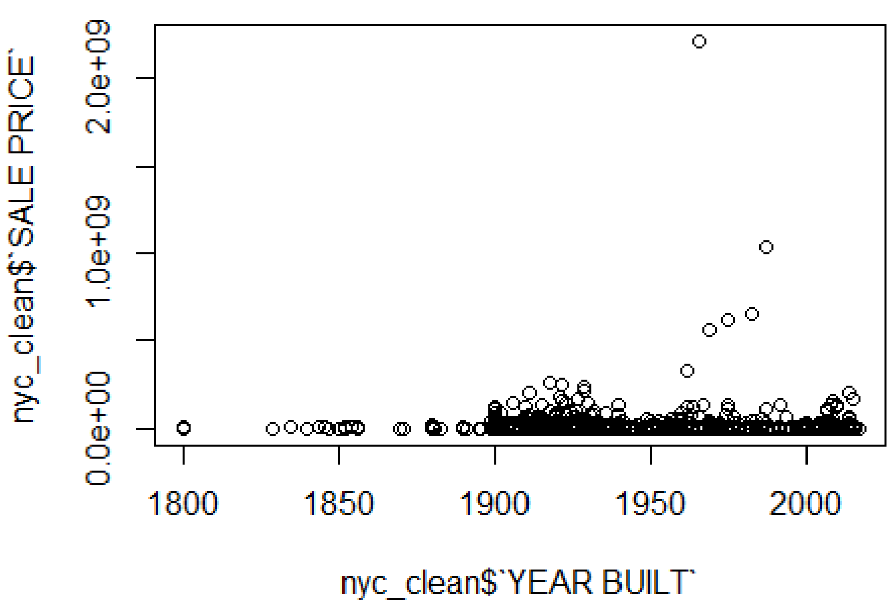
```
which(is.na(nyc_clean$`SALE DATE`))

## integer(0)

which(nyc_clean$`SALE DATE`<2016-09-01)

## integer(0)

nyc_clean<-nyc_clean%>%mutate(sale_date=`SALE DATE`)%>%select(-`SALE
DATE`)
```

Year built: explore the data.

```
plot(nyc_clean$`YEAR BUILT`,nyc_clean$`SALE PRICE`)
```

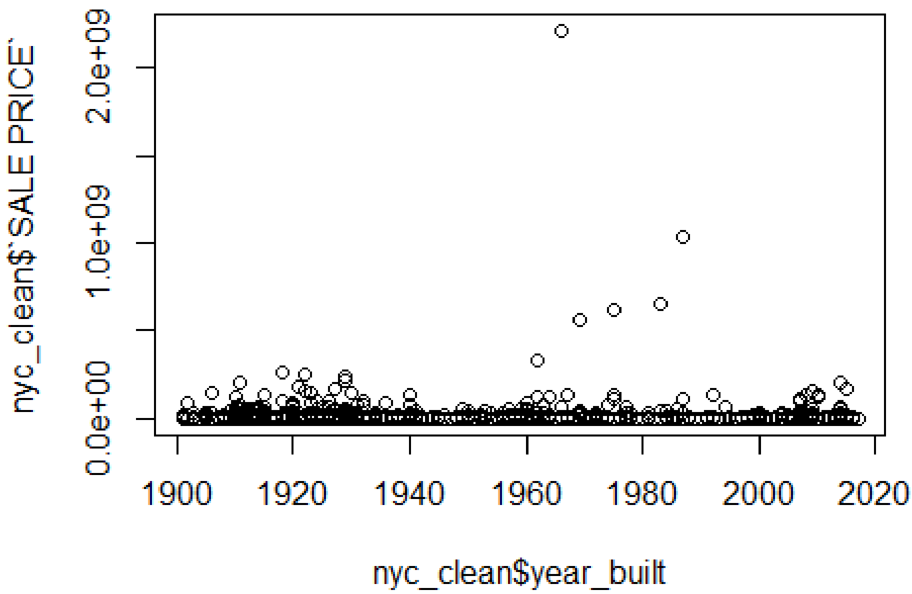



```
nyc_clean<-nyc_clean%>%filter(`YEAR BUILT`>0)
plot(nyc_clean$`YEAR BUILT`,nyc_clean$`SALE PRICE`)
```



From the plot we obtain that some "year built" are 0, which cannot be considered. We also see that some of them are from 19th century. These data from 19th is not representative because we have a number of sales that is not enough to use it statistically. Then, we just take data beginning at 1900.

```
nyc_clean<-nyc_clean%>%filter(`YEAR BUILT`>1900)
nyc_clean<-nyc_clean%>%mutate(year_built=`YEAR BUILT`)%>%select(-`YEAR
BUILT`)
plot(nyc_clean$year_built,nyc_clean$`SALE PRICE`)
```



Land and prices

Firstly, we are going to analyze the distribution of prices as it seems that many sales have reported price 0.

```
nyc_clean_price<-nyc_clean%>%group_by(`SALE
PRICE`)%>%summarize(n=n())%>%arrange(-n)
nyc_clean_price

## # A tibble: 4,540 x 2
##    `SALE PRICE`     n
##           <dbl> <int>
##  1            0  6880
##  2           10   541
##  3       700000   285
##  4       450000   275
##  5       600000   271
```

```
## 6        650000   271
## 7        550000   269
## 8        400000   264
## 9        500000   250
## 10       750000   230
## # ... with 4,530 more rows

rm(nyc_clean_price)
```

We consider that we should eliminate all prices below 10 dollars as they are not representative. We cannot work neither with price 0 nor with any price below 10.

```
nyc_clean<-nyc_clean%>%filter(`SALE PRICE`>10)
```

Transformation of variables: As previously explained, we are going to use price per square feet and F.A.R. (floor area ratio):

```
nyc_clean<-nyc_clean%>%mutate(price_lsf=`SALE PRICE`/`LAND SQUARE FEET`)
nyc_clean<-nyc_clean%>%mutate(FAR=`GROSS SQUARE FEET`/`LAND SQUARE
FEET`)%>%select(-`LAND SQUARE FEET`,-`GROSS SQUARE FEET`,-`SALE PRICE`)
```

At this moment, our initial data frame has been reduced to 27272 sales and 12 variables.

```
head(nyc_clean)

##   BOROUGH COMMERCIAL UNITS TOTAL UNITS building_class com_perce
res_perce
## 1       1                0          10              C      0.00
1.00
## 2       1                0           8              C      0.00
1.00
## 3       1                0          24              D      0.00
1.00
## 4       1                0          10              D      0.00
1.00
## 5       1                0          24              C      0.00
1.00
## 6       1                1           4              S      0.25
0.75
##   RESIDENTIAL UNITS tax_class  sale_date year_built price_lsf      FAR
## 1                10         2 2016-09-23       1913  1732.514 2.990317
## 2                 8         2 2016-09-23       1920  1824.480 2.414857
## 3                24         2 2016-11-07       1920  3615.950 4.126309
## 4                10         2 2016-10-17       2009  2784.504 3.322572
## 5                24         2 2017-06-21       1928  2880.658 4.061002
## 6                 3         2 2016-11-15       1910  2171.053 2.210526
```

Data splitting

train and test

We are going to take a 10% of data to test our models:

```
set.seed(2)
test_index <- createDataPartition(y = nyc_clean$price_lsf, times = 1, p =
0.1, list = FALSE)
nyc_train <- nyc_clean[-test_index,]
nyc_temp <- nyc_clean[test_index,]
```

We must assure that classes present in test set are included in train set. Otherwise code cannot operate. We must move these classes that are in test set from test to train.

```
# Make sure classes(borough, tax, building) in 'test' set are also in
'train' set
nyc_test <- nyc_temp %>%
  semi_join(nyc_train, by = "building_class") %>%
  semi_join(nyc_train, by = "tax_class")%>%
  semi_join(nyc_train, by = "BOROUGH")

# Add rows removed from 'test' set back into 'train' set
removed <- anti_join(nyc_temp, nyc_test)

## Joining, by = c("BOROUGH", "COMMERCIAL UNITS", "TOTAL UNITS",
"building_class", "com_perce", "res_perce", "RESIDENTIAL UNITS",
"tax_class", "sale_date", "year_built", "price_lsf", "FAR")

nyc_train <- rbind(nyc_train, removed)
rm(test_index, nyc_temp, removed)
```

From now on, we work with nyc_train to explore the data. Once we know how the models will be, we test and tune them with the nyc_test. As it is not compulsory, in this case we are not going to split in validation data.

Splitting justification:

In [8] we have notices that the split ratio has been used in some cases as 50/50 and in others 20/80 or 10/90. We have researched and according to [9], the split ratio depends mainly on 2 aspects: the total of number of samples and the paramters to be tuned in the models. In [10] we have seen that a size between 100 and 100000 can be splitted in

[8] https://www.edx.org/professional-certificate/harvardx-data-science

[9] https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

[10] https://towardsdatascience.com/data-splitting-technique-to-fit-any-machine-learning-model-c0d7f3f1c790

80/20 or 90/10. A dataset with less than 100 samples must be splitted with a larger test set. Otherwise, test set would not be representative. Meanwhile, a dataset of more than 1 million can be splitted with a training set higher than 95% because test set would be representative enough. Regarding the difficulty of our models, apart from linear models, we are going to tune 1 paramter. We are not going to use many parameters. We must tune span for loess (1), k for KNN (1) and cp for trees (1). Following the advices indicated in the links, with a sample close to 30000 and a paramter to be tuned in each model, we could take a ratio of 60/20/20 or 80/10/10. As it is not compulsory to split in validation data, we the ratios would be 80/20 or 90/10. Besides, caret packages (the one that is used) internally makes cross-validation with the train set. Summarizing we have chosen a ratio of 90/10. We consider that a test set with close to 3000 reported sales may be representative enough even considering the presenc of outliers.

we would like to pinpoint tha for time varying datasets, the slicing should be different [11]. Neverthless, we have noticed that both time variables are not correlated enough to be considered. See data exploration year built and sale date. In other works in which time is more correlated, the splitting must be different and must follow the rules diven in the link provided.

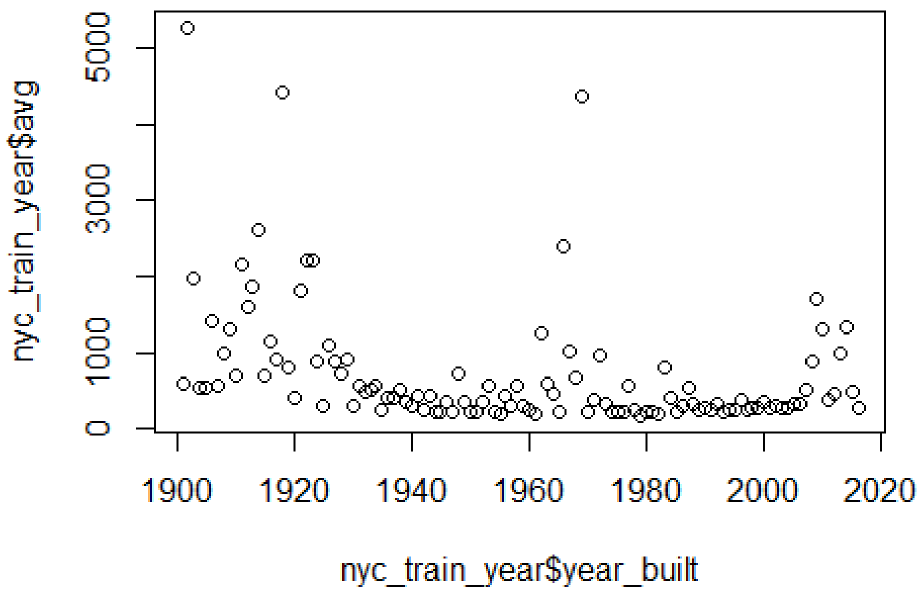
Data exploration

Year built

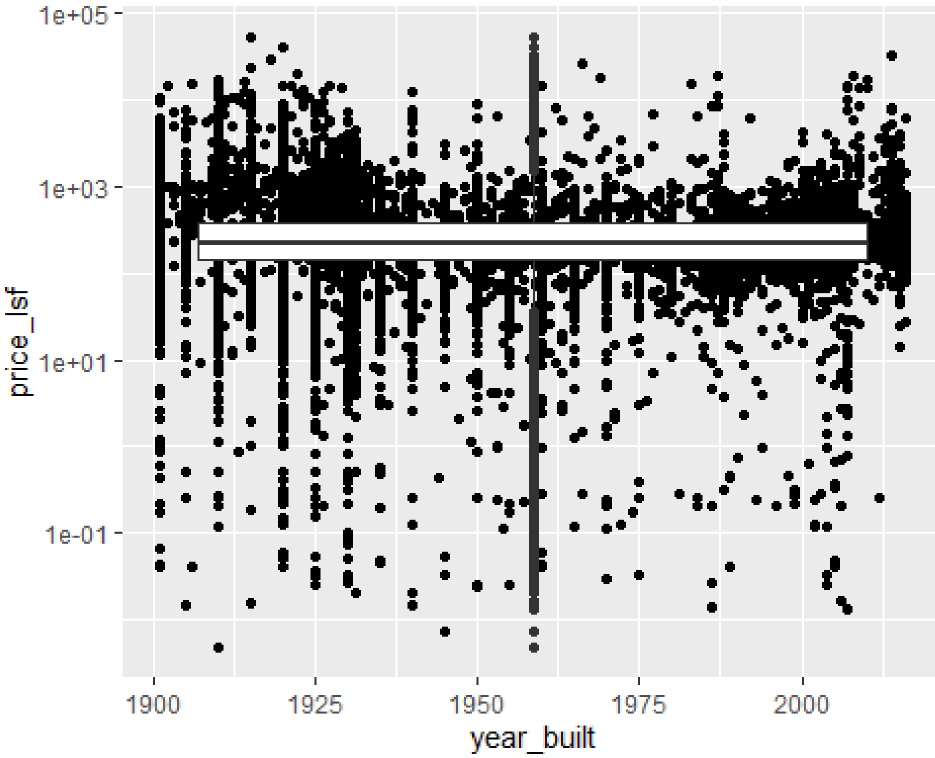
We want to analyze if there is a relation between the year of construction of the building and the variation of price As there are many sales, we analyze the relation between the average of prices of sales for each year built. One may consider that new buildings are more expensive:

```
nyc_train_year<-
nyc_train%>%group_by(year_built)%>%mutate(avg=mean(`price_lsf`))
plot(nyc_train_year$year_built,nyc_train_year$`avg`)
```

[11] https://topepo.github.io/caret/data-splitting.html#time




```
rm(nyc_train_year)
cor(nyc_train$year_built, nyc_train$price_lsf)

## [1] -0.0694182

nyc_train%>%ggplot(aes(x=year_built,y=price_lsf))+geom_point()+geom_boxpl
ot()+scale_y_log10()
```



Nevertheless, from the graphics and the correlation calculation we notice:

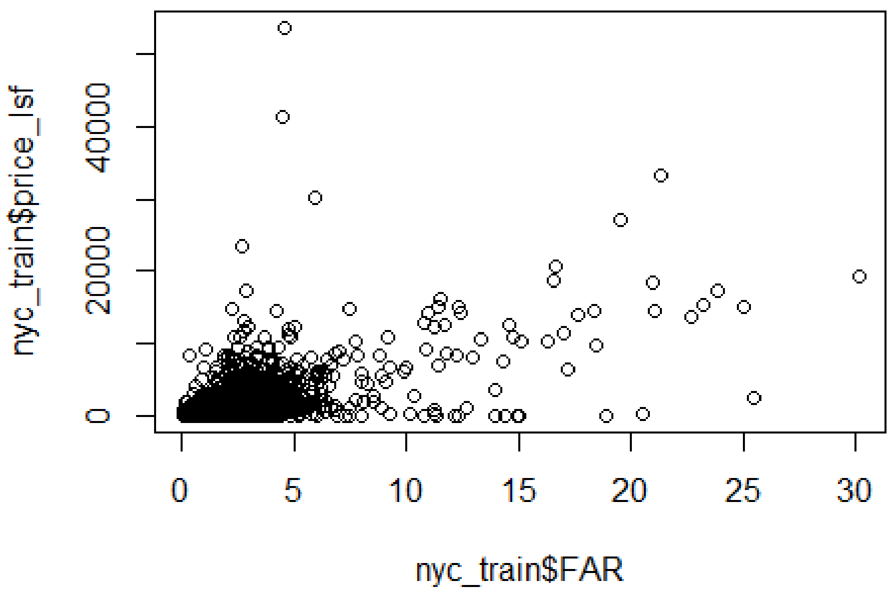
- The older the building the more expensive it is.
- The correlation is negative and low. We should not consider it.
- We need logarithmic scales for the prices are there are high differences between prices.

FAR

We expect that the correlation between FAR, the use of the land, and the price per square feet is high. We are going to check it with the standard calculation and then according to [12], we calculate it with spearman method. We suppose that there are many outliers and they may have influence in teh calculation:

```
plot(nyc_train$FAR,nyc_train$price_lsf)
```

[12] https://learning.edx.org/course/course-v1:HarvardX+PH125.7x+2T2020/block-v1:HarvardX+PH125.7x+2T2020+type@sequential+block@fec4dd1cf6a0417c8ff836b29a2064b3/block-v1:HarvardX+PH125.7x+2T2020+type@vertical+block@163263ef7a5b4336a00a3a15295119ef

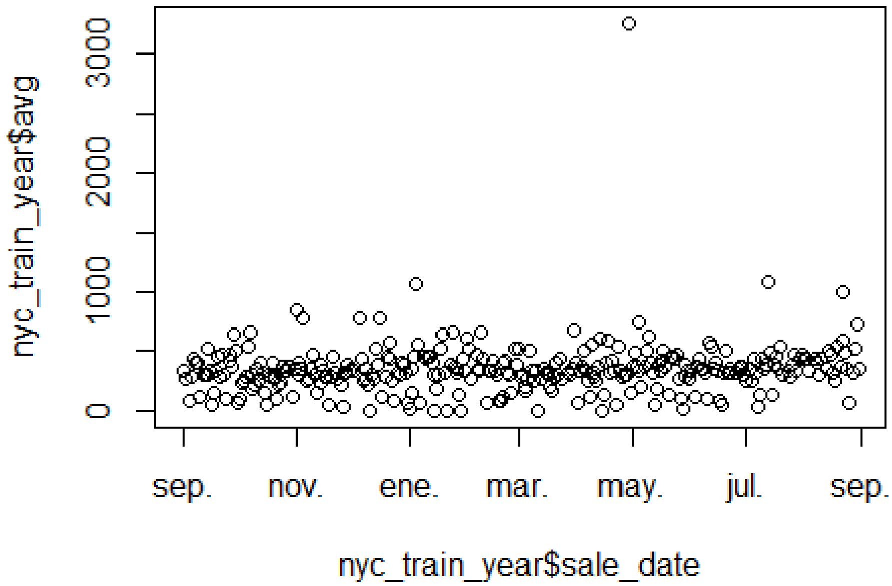



```
cor(nyc_train$FAR,nyc_train$price_lsf)

## [1] 0.505402

cor(nyc_train$FAR,nyc_train$price_lsf,method = "spearman")

## [1] 0.6383351
```

From the graphic we notice that there are many outliers and it is difficult to obtain conclusions. The default (pearson) calculation establish a correlation of 0.505402 which is high and above 0.5. However, with Spearmn we can avoid the distortoion of "outliers" and we get 0.6383351 which is enough to consider the variable FAR really useful for the study.

Sale date.

```
nyc_train_year<-
nyc_train%>%group_by(sale_date)%>%mutate(avg=mean(`price_lsf`))
plot(nyc_train_year$sale_date,nyc_train_year$`avg`)
```

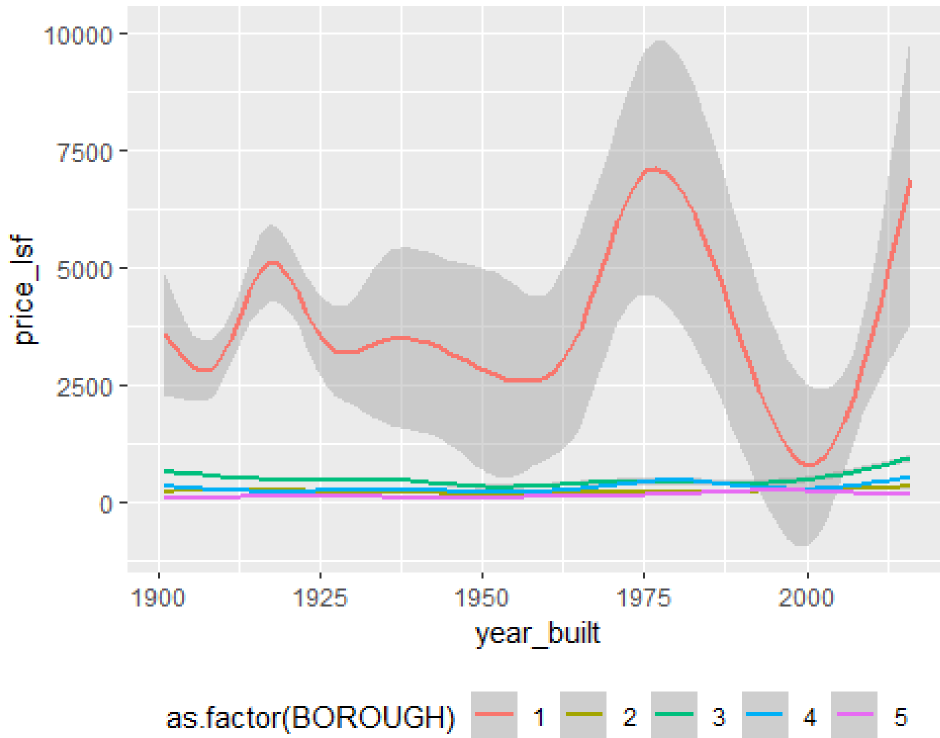



```
rm(nyc_train_year)
nyc_train_year<-nyc_train%>%mutate(sale_date = as.numeric(sale_date))
cor(nyc_train_year$sale_date, nyc_train_year$price_lsf)

## [1] 0.007782703

rm(nyc_train_year)
```

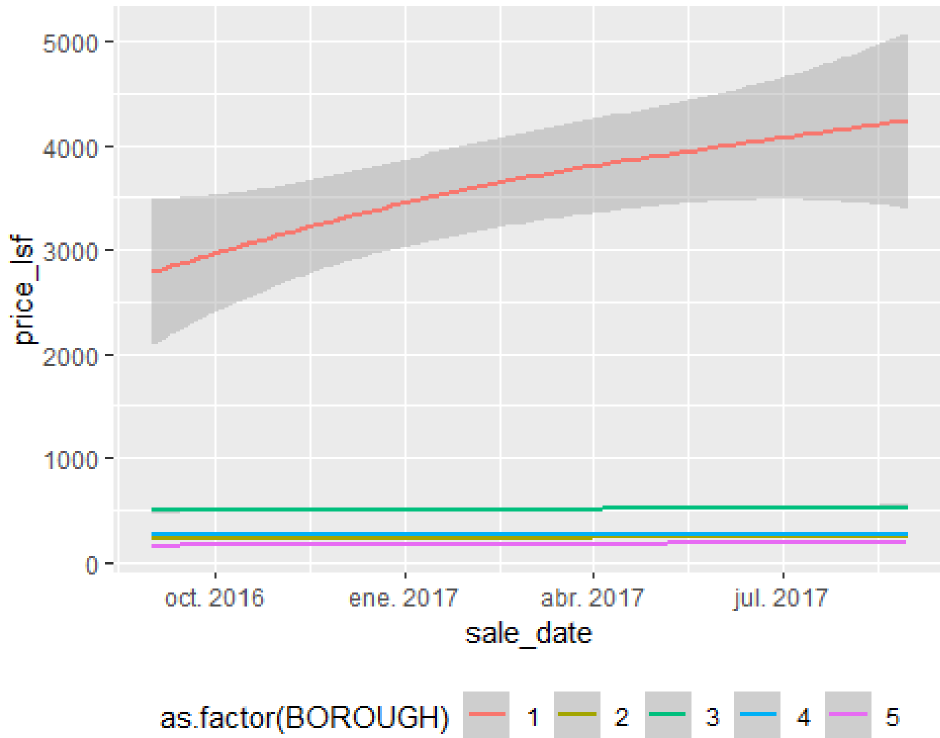
The conclusion is that sale date (between september 2016 and september 2017) is not a significant variable.

Borough

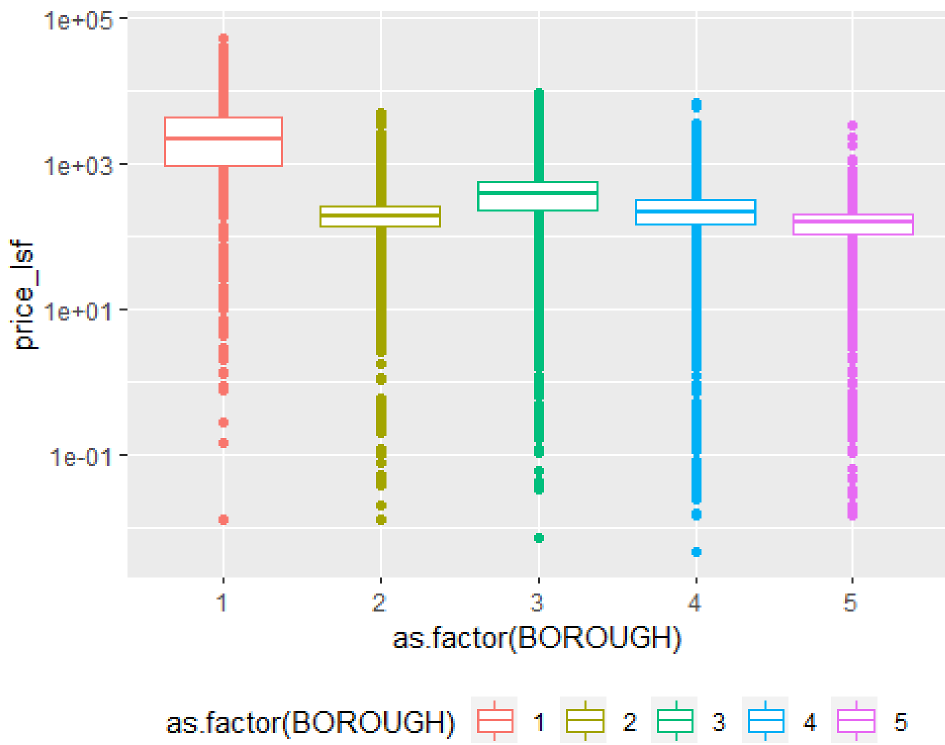
```
ggplot(data=nyc_train)+geom_smooth(mapping=aes(x=year_built,y=price_lsf,group=as.factor(BOROUGH),color=as.factor(BOROUGH)))+theme(legend.position="bottom")
```




```
ggplot(data=nyc_train)+geom_smooth(mapping=aes(x=sale_date,y=price_lsf,gr
oup=as.factor(BOROUGH),

color=as.factor(BOROUGH)))+theme(legend.position="bottom")
```




```
nyc_train%>%group_by(BOROUGH)%>%
  ggplot(aes(x=as.factor(BOROUGH),y=price_lsf,
group=as.factor(BOROUGH),color=as.factor(BOROUGH)))+

geom_point()+geom_boxplot()+scale_y_log10()+theme(legend.position="bottom
")
```



We can notice that the variation of sale prices is minimun in boroughs 2,3,4 and 5 during the year of sale. The variation of price in relation to the construction of the building is higher in borough 1 than in the others. This fact is surely due to the higher variability in general in this borough. From the boxplot graphics we notice that the range is higher in district 1 than in the others (same conclusion).

```
nyc_train_bor<-nyc_train%>%group_by(BOROUGH)%>%summarize(r =
cor(as.numeric(sale_date), price_lsf,method="spearman"))
nyc_train_bor

## # A tibble: 5 x 2
##   BOROUGH      r
##     <dbl>  <dbl>
## 1       1 0.210
## 2       2 0.0799
## 3       3 0.0366
## 4       4 0.0296
## 5       5 0.0861

rm(nyc_train_bor)
nyc_train_bor<-nyc_train%>%group_by(BOROUGH)%>%summarize(r =
cor(as.numeric(year_built), price_lsf,method="spearman"))
nyc_train_bor
```

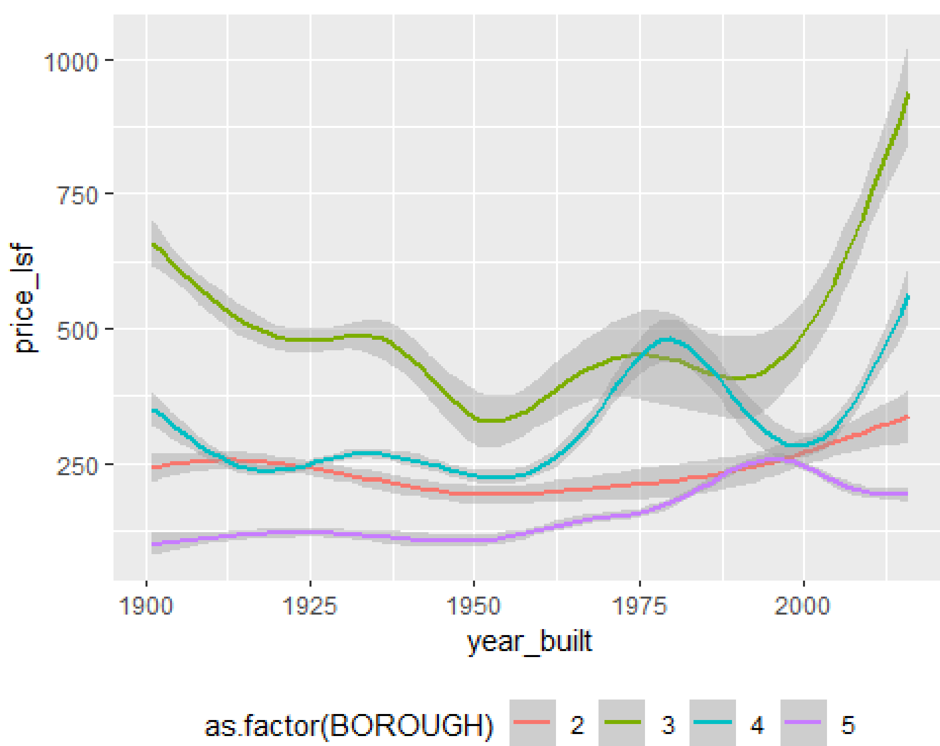
```
## # A tibble: 5 x 2
##   BOROUGH       r
##     <dbl>   <dbl>
## 1       1 -0.188
## 2       2  0.0913
## 3       3 -0.124
## 4       4  0.0494
## 5       5  0.524

rm(nyc_train_bor)
```

Although we cannot get conclusions from correlation (because boroughs are not continuous variables), we notice that the difference of data, and of correlation factors is high between boroughs. So, we must consider the difference of boroughs for our models either by categorical variables or by decission trees. Anyway, we are interested in analyzing the data separating borough 1 and the others:

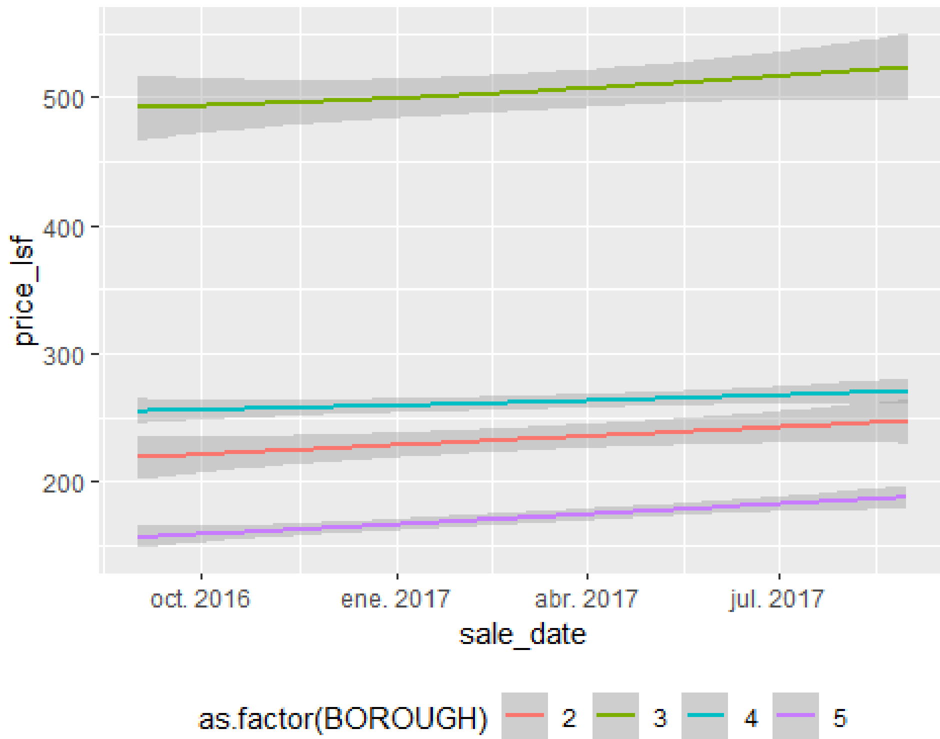
```
nyc_train_25<-nyc_train%>%filter(BOROUGH!=1)
nyc_train_1<-nyc_train%>%filter(BOROUGH=="1")
```

Once separated Manhattan (1) from the others, we try to make the same plots:

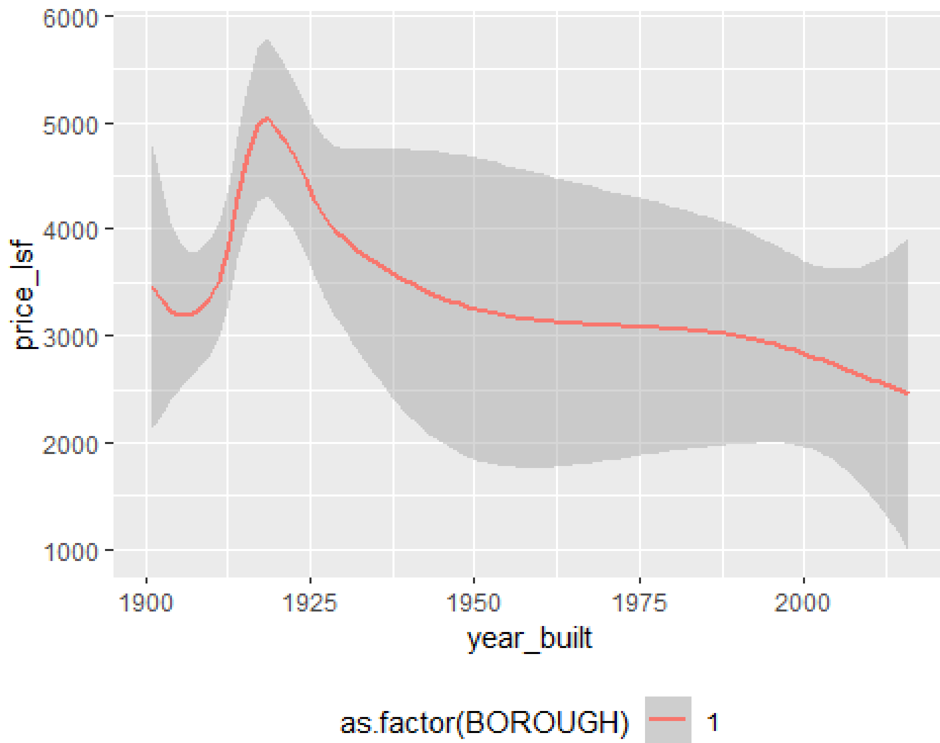
```
ggplot(data=nyc_train_25)+geom_smooth(mapping=aes(x=year_built,y=price_ls
f,group=as.factor(BOROUGH),

color=as.factor(BOROUGH)))+theme(legend.position="bottom")
```



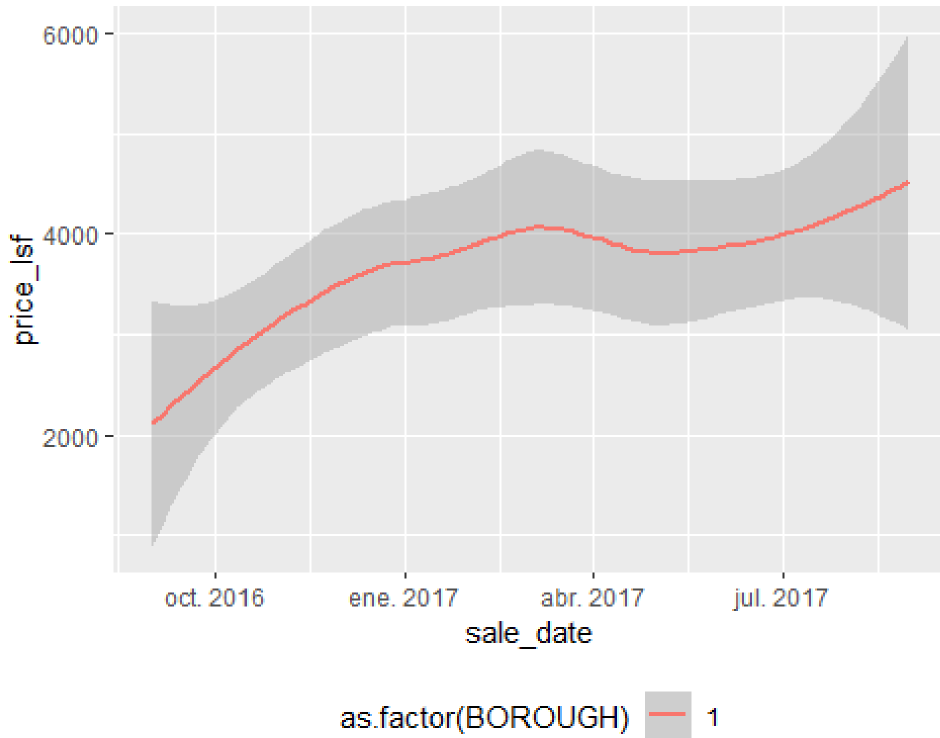
```
ggplot(data=nyc_train_25)+geom_smooth(mapping=aes(x=sale_date,y=price_lsf
,group=as.factor(BOROUGH),

color=as.factor(BOROUGH)))+theme(legend.position="bottom")
```




```
ggplot(data=nyc_train_1)+geom_smooth(mapping=aes(x=year_built,y=price_lsf
,group=as.factor(BOROUGH),

color=as.factor(BOROUGH)))+theme(legend.position="bottom")
```




```
ggplot(data=nyc_train_1)+geom_smooth(mapping=aes(x=sale_date,y=price_lsf,
group=as.factor(BOROUGH),

color=as.factor(BOROUGH)))+theme(legend.position="bottom")
```




```
rm(nyc_train_1)
rm(nyc_train_25)
```

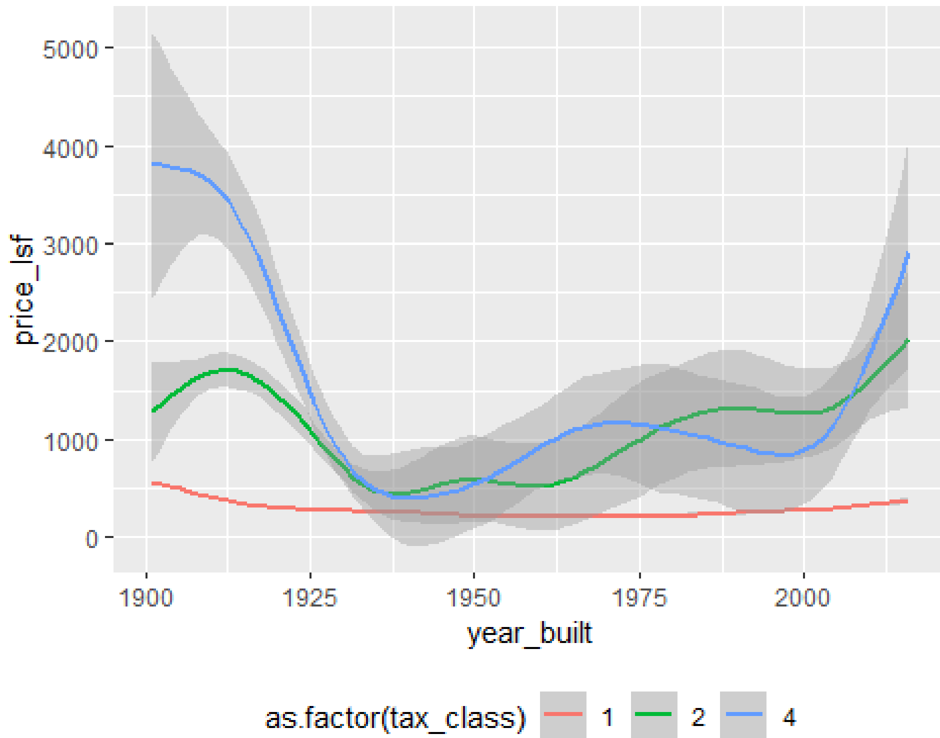
By separating borough 1, we can confirm that the price does not vary between 2016 and 2017 for boroughs 2 to 5 and lightly in borough 1. Price does vary in all boroughs depending of the year of the building with higher variations in borough 1. Finally, it is clear that the variability of the data in borough 1 es important as the buffer (standard deviation) shows. Therefore, providing that each borough has different behaviour, we must take BOROUGH'S division into account.

Tax class

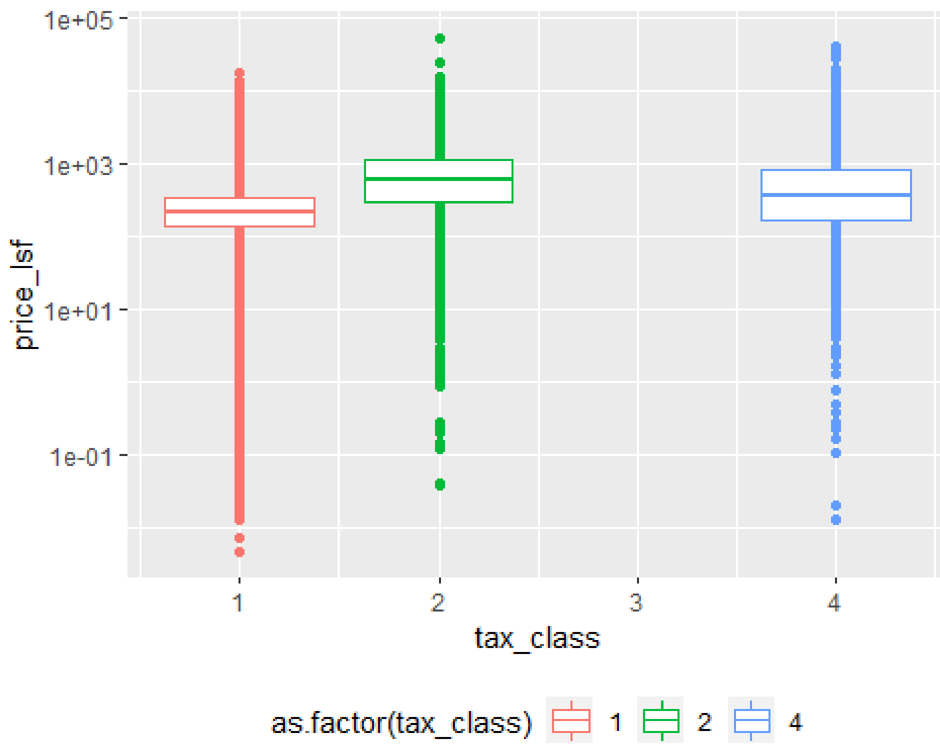
Tax class may have an influence in prices. we expect that higher taxes imply higher prices. Nevertheless we have a categorical variable, nor a continuous one. [13] We can try to study the influence like we have done with boroughs. This means studying the variation of prices along time depending of tax class:

```
ggplot(data=nyc_train)+geom_smooth(mapping=aes(x=year_built,y=price_lsf,group=as.factor(tax_class),

color=as.factor(tax_class)))+theme(legend.position="bottom")
```

13 http://www.cynthiaramirez.nyc/uploads/1/1/8/7/118704813/class_1_guide__1_.pdf




```
nyc_train_tax<-nyc_train%>%group_by(tax_class)%>%summarize(r =
cor(as.numeric(year_built), price_lsf,method="spearman"))
nyc_train_tax

## # A tibble: 3 x 2
##   tax_class      r
##       <dbl>  <dbl>
## 1         1 -0.101
## 2         2 -0.236
## 3         4 -0.311

rm(nyc_train_tax)
nyc_train%>%group_by(tax_class)%>%
  ggplot(aes(x=tax_class,y=price_lsf,
group=as.factor(tax_class),color=as.factor(tax_class)))+

geom_point()+geom_boxplot()+scale_y_log10()+theme(legend.position="bottom
")
```



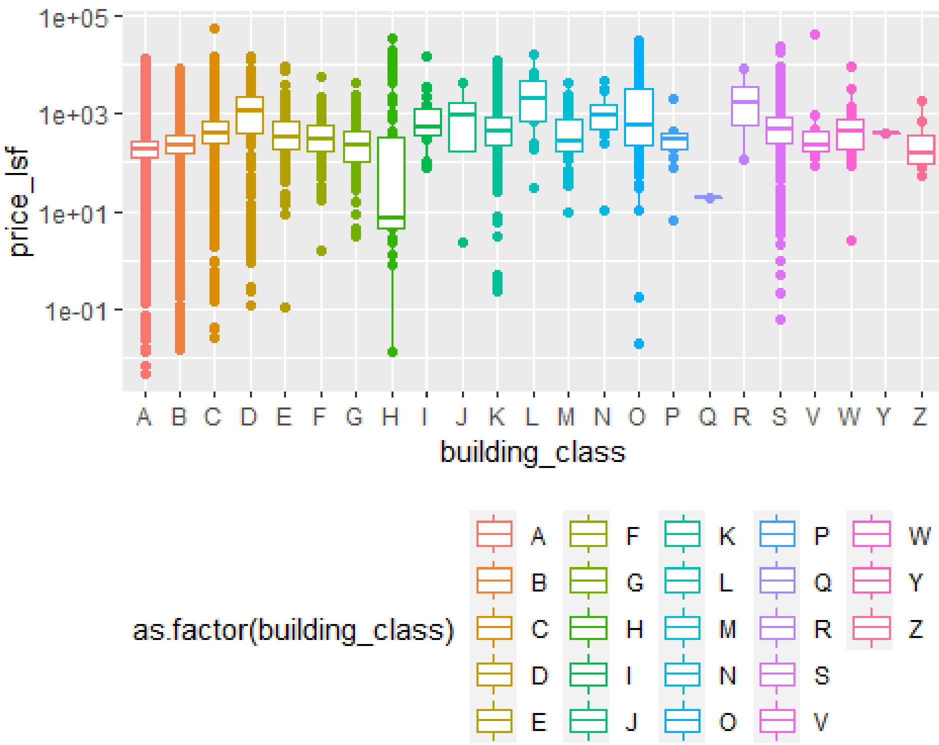
From graphics we see that:

- The variability of prices for tax class 4 (industrial and commercial) is high (see boxplot).
- The variability of prices for tax class 1 (residential of 1-3 units) is minimum.
- The prices of tax class 4 and 2 dcrease from 1900 to 1950 and increase since 200.The increase is lowert than the initial decrease.
- The correlations are low. neverthless, the correlations are between time and prices. Here, we just pursued if there is any difference between tax classes. We notice that -0.1 (tax class 1) and -0.3 (tax class 4) are similar. Problably, we will not be able to obtain significative information from tax class.

Building class (categorical)

Given the great amount of building classes, the smmoothing graphic does not provide useful information in this case. We prefer to analyze the boxplot.

```
nyc_train%>%group_by(building_class)%>%
  ggplot(aes(x=building_class,y=price_lsf,
group=as.factor(building_class),color=as.factor(building_class)))+

geom_point()+geom_boxplot()+scale_y_log10()+theme(legend.position="bottom
")
```



From the boxplot graphic we can see that despite the quartile range is not great for many of the classes, the variability of the data is high. There are many outliers. Besides, in clases H,J and O the quartile range is high. which could mean that there are few and really different reported sales. In the next step, we calculate the number of reported sales for each type of building:

```
nyc_train_bc<-nyc_train%>%group_by(building_class)%>%summarize(number=
n())
nyc_train_bc

## # A tibble: 23 x 2
##    building_class number
##    <chr>           <int>
##  1 A               11017
##  2 B                7948
##  3 C                3307
##  4 D                 192
##  5 E                 149
##  6 F                  90
##  7 G                 153
##  8 H                  89
##  9 I                  23
## 10 J                   4
## # ... with 13 more rows
```

We have 89 reported sales with building class H, 4 with building class J and 192 with building class O. in J case, the variability is explained by the low number of reported sales meanwhile for H and O the differente between prices must be enormous. The great amount of sales correspond to A (11017),B(7948),C(3307),K (407) and S (825). According to [14], A,B ,C and S are residential. K is commercial, whereas O is office buildings, J is leisure facilities and H is hotels.

```
nyc_train_bc<-nyc_train%>%group_by(building_class)%>%summarize(r =
cor(as.numeric(year_built), price_lsf,method="spearman"))
nyc_train_bc%>%arrange(r)

## # A tibble: 23 x 2
##    building_class      r
##    <chr>           <dbl>
##  1 V              -0.609
##  2 P              -0.523
##  3 O              -0.374
##  4 M              -0.291
##  5 N              -0.269
##  6 W              -0.221
##  7 H              -0.217
##  8 K              -0.200
##  9 B              -0.179
## 10 F              -0.177
## # ... with 13 more rows

nyc_train_bc%>%arrange(-r)

## # A tibble: 23 x 2
##    building_class        r
##    <chr>             <dbl>
##  1 Z               0.465
##  2 J               0.316
##  3 L               0.0825
##  4 A               0.00840
##  5 R               0
##  6 E              -0.0624
##  7 G              -0.0649
##  8 S              -0.0807
##  9 I              -0.0872
## 10 C              -0.145
## # ... with 13 more rows
```

Like with boroughs and tax classes, we cannot obtain really useful conclusions from these correlations. Nevertheless, the evolution of prices in time is really different depending on the building class. The r varies from (0) (R) to (0.0824689) (L). The summary is:

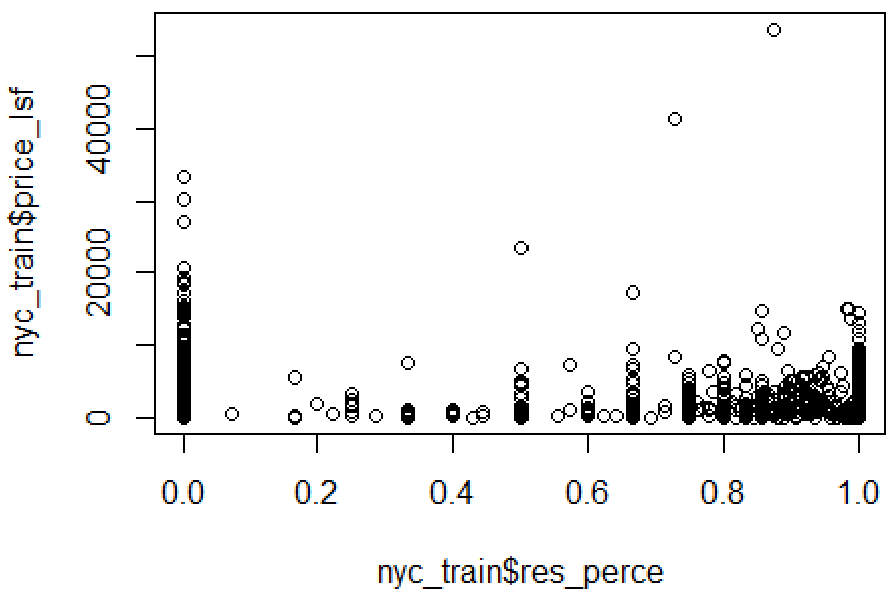
[14] https://www1.nyc.gov/assets/finance/jump/hlpbldgcode.html

- Studying the categorical variable building class is difficulty (too many classes and too much variability).
- The differences between classes are high.
- The most reported classes are A, B, C, S and K which are either residential or commercial.
- We must consider building class.
- The percentage of commercial or residencial units may be useful.

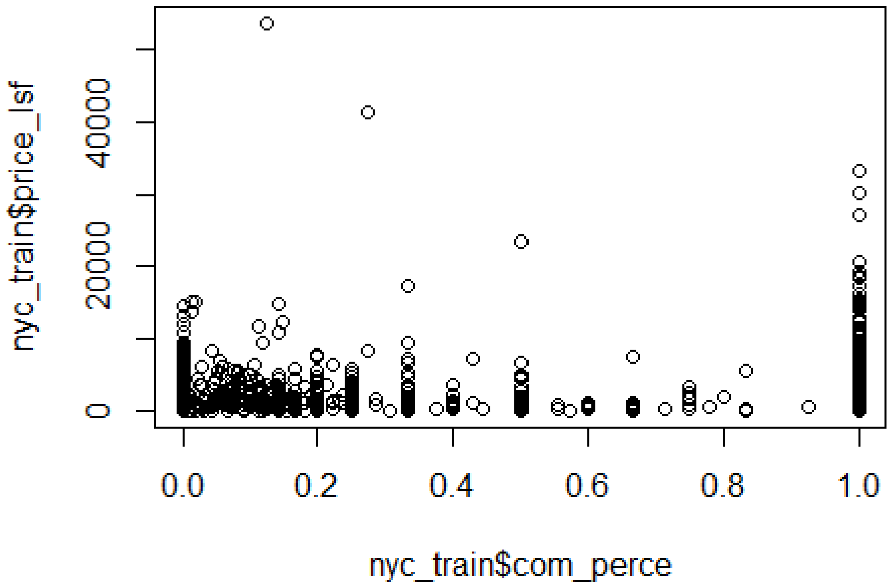
Building class (continuous)

We must study these percentage in the same way we studied FAR.Given the great amount of outliers reported, we directly study correlation by spearman method in this case:

```
plot(nyc_train$res_perce,nyc_train$price_lsf)
```




```
cor(nyc_train$res_perce,nyc_train$price_lsf,method = "spearman")

## [1] -0.1925943

plot(nyc_train$com_perce,nyc_train$price_lsf)
```




```
cor(nyc_train$com_perce,nyc_train$price_lsf,method = "spearman")

## [1] 0.1949046
```

In the graphic it is difficult to see any relation. Regarding correlations, the higher proportion of residential units decreases the prices opposite to a higher proportion of commercial units. Nevertheless, the correlation is close to 0.2 which is not significant. We can use this variables but they are not going to improve our results as the correlation is low.

Summary

The main variable is FAR which is highly correlated to price per square feet. The correlation for sale date is inexistent. The correlation for year of building is really low. The correlation of continuous variables residential and commercial are low but could be useful. In relation to categorical variables, we are not sure the efficiency of considering borough, tax class and building class. Firstly, we should take all into account and then discart those which lower relation.

Models

Models’ definition

The most intuitive model for price prediction challanges is linear regression. There are approaches such as ARIMA for time depending series which may be really usfeul for this case. Nevertheless, as ARIMA is not included in [15], we are not going to use it. We can develop different linear models. With tecnhiques such as AIC, BAS, BIC or ANOVA we would have been able to select the best linear model. Again, we are not going to use them because they are not included in the program. Given the easiness of linear models, we can check quickly some possibilites. Withoug a great effort we can compare the different models an select the best one.

We suppose that we can achieve a better prediction with models more complicated than linear regression, all of them included in the program. We have considered KNN model and Loess model. As explained in [16], loess model may be a good aproximation for a time varying variable. However, in our study we have stated that the time variable is not as significative as we expected.

We cannot procede with these models as with linear models are time of calculation here could be a problem. Therefore, we must select the variables we are going to use for each model with the help of the linear models. Once we pick the varibales to be used, we can tune the loess and the knn parameters with iterations. We cannot use too many variables because the optimization of these parameters would be a really long process. Surely, the variable that must be considered in any model is FAR. Summarizing, we select the variables to be used in KNN and LOESS by means of the linear models interpretation.

Finally, given that we have three possible categorical variables to use, and the difference of behaviour of prices in the different categories we consider that a regression tree could be a good combination of continuaous and categorical variables.

The procedure for the models would be as follows:

- Calculation or optimization of the model.
- Selection of the best model (lineal) or the best parameters (rest of models).
- Calculation of the prediction.

---

[15] https://www.edx.org/professional-certificate/harvardx-data-science

[16] https://learning.edx.org/course/course-v1:HarvardX+PH125.8x+2T2020/block-v1:HarvardX+PH125.8x+2T2020+type@sequential+block@5e2f559f1188441fa6bd972e356994c3/block-v1:HarvardX+PH125.8x+2T2020+type@vertical+block@af43f54714794279877b1296c61a45a0

– Calculation of the reference parameter RMSE.

Linear models

Firstly, we are going to run two extreme models: only FAR model and all-included model. After that, we add or eliminate variables and see the influence. We also, analyze the p-values.

Only FAR lineal model (11)

```
fit_11<-lm(price_lsf~FAR,data=nyc_train)
summary(fit_11)

##
## Call:
## lm(formula = price_lsf ~ FAR, data = nyc_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8174   -130    -38     39  51898
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.952      6.999   0.565    0.572
## FAR          403.572      4.398  91.759   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 866.2 on 24543 degrees of freedom
## Multiple R-squared:  0.2554, Adjusted R-squared:  0.2554
## F-statistic:  8420 on 1 and 24543 DF,  p-value: < 2.2e-16

y_test_11<-predict(fit_11,nyc_test)
RMSE_test_n11<-sqrt(mean((y_test_11 - nyc_test$price_lsf)^2))
RMSE_test_n11

## [1] 793.1704

RMSE_linear<-tibble(Method="Linear FAR",RMSE=RMSE_test_n11)
```

As expected, p-value for FAR is correct. The RMSE is (793.1703664).

All-included lineal model (12)

Directly, we do not include sale_date. It is not going to be useful as we have stated previously.

```
fit_12<-lm(price_lsf~FAR+as.factor(BOROUGH)+as.factor(building_class)+

as.factor(tax_class)+year_built+com_perce+res_perce,data=nyc_train)
summary(fit_12)
```

```
##
## Call:
## lm(formula = price_lsf ~ FAR + as.factor(BOROUGH) +
as.factor(building_class) +
##     as.factor(tax_class) + year_built + com_perce + res_perce,
##     data = nyc_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -10861    -98     -5     71  50053
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4088.3189   393.5549  10.388  < 2e-16 ***
## FAR                           392.1082     6.8529  57.217  < 2e-16 ***
## as.factor(BOROUGH)2         -2251.6755    40.3376 -55.821  < 2e-16 ***
## as.factor(BOROUGH)3         -2027.8353    38.5597 -52.589  < 2e-16 ***
## as.factor(BOROUGH)4         -2104.2045    39.7355 -52.955  < 2e-16 ***
## as.factor(BOROUGH)5         -2149.1245    42.1277 -51.015  < 2e-16 ***
## as.factor(building_class)B   -49.3290    11.7578  -4.195 2.73e-05 ***
## as.factor(building_class)C  -140.5228    19.7852  -7.102 1.26e-12 ***
## as.factor(building_class)D  -883.6730    65.4698 -13.497  < 2e-16 ***
## as.factor(building_class)E -2607.1676   238.4626 -10.933  < 2e-16 ***
## as.factor(building_class)F -2762.3417   244.0478 -11.319  < 2e-16 ***
## as.factor(building_class)G -2522.1464   224.8991 -11.215  < 2e-16 ***
## as.factor(building_class)H -7160.5033   254.2765 -28.160  < 2e-16 ***
## as.factor(building_class)I -2343.6837   279.7453  -8.378  < 2e-16 ***
## as.factor(building_class)J -3159.8445   452.8136  -6.978 3.06e-12 ***
## as.factor(building_class)K -2536.6860   230.8491 -10.989  < 2e-16 ***
## as.factor(building_class)L -3255.7182   294.5649 -11.053  < 2e-16 ***
## as.factor(building_class)M -3089.1863   254.2808 -12.149  < 2e-16 ***
## as.factor(building_class)N -2736.9836   320.1721  -8.548  < 2e-16 ***
## as.factor(building_class)O -1881.4585   235.7494  -7.981 1.52e-15 ***
## as.factor(building_class)P -2646.1913   314.0088  -8.427  < 2e-16 ***
## as.factor(building_class)Q -2623.1059   808.4598  -3.245 0.001178 **
## as.factor(building_class)R   308.8403   391.1923   0.789 0.429836
## as.factor(building_class)S   -23.5282    47.9141  -0.491 0.623396
## as.factor(building_class)V  2165.8513   210.1751  10.305  < 2e-16 ***
## as.factor(building_class)W -2653.4818   281.0474  -9.441  < 2e-16 ***
## as.factor(building_class)Y -2802.5125   808.4336  -3.467 0.000528 ***
## as.factor(building_class)Z -2379.4845   309.0219  -7.700 1.41e-14 ***
## as.factor(tax_class)2       -117.4167    25.5426  -4.597 4.31e-06 ***
## as.factor(tax_class)4       2635.0444   232.7305  11.322  < 2e-16 ***
## year_built                    -0.9221     0.1897  -4.860 1.18e-06 ***
## com_perce                    -95.9830   127.6570  -0.752 0.452129
## res_perce                   -173.2180   147.9641  -1.171 0.241741
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 774.9 on 24512 degrees of freedom
```

```
## Multiple R-squared:  0.4048, Adjusted R-squared:  0.4041
## F-statistic:   521 on 32 and 24512 DF,  p-value: < 2.2e-16
```

```
y_test_12<-predict(fit_12,nyc_test)
RMSE_test_n12<-sqrt(mean((y_test_12 - nyc_test$price_lsf)^2))
RMSE_test_n12
```

```
## [1] 703.0657
```

```
RMSE_linear<-bind_rows(RMSE_linear,tibble(Method="Linear all-
included",RMSE=RMSE_test_n12))
```

Now, we can compare the RMSE. We have decrease the initial (793.1703664) to (703.0657221). We should wonder which of the added variables has increased the accuracy:

- p-value of com_perce and res_perce is not corrected. We will avoid them in next step.
- some of building classes are not corrected but we cannot use some of them. We will keep them.
- year built p-value shows us that it may not be useful. At this moment we keep it.
- for the same reasons as with building classes we keep tax classes at this moment.

Commercial and residential not included. (13)

```
fit_13<-lm(price_lsf~FAR+as.factor(BOROUGH)+as.factor(building_class)+
             as.factor(tax_class)+year_built,data=nyc_train)
y_test_13<-predict(fit_13,nyc_test)
RMSE_test_n13<-sqrt(mean((y_test_13 - nyc_test$price_lsf)^2))
RMSE_test_n13
```

```
## [1] 702.8983
```

```
RMSE_linear<-bind_rows(RMSE_linear,tibble(Method="Linear without com &
res",RMSE=RMSE_test_n13))
```

We obtain almost the same RMSE which means our decision of eliminating commercial and residential proportions is correct. Nowe, we eliminate year_built.

FAR, BOROUGH and classes.(14)

```
fit_14<-lm(price_lsf~FAR+as.factor(BOROUGH)+as.factor(building_class)+
             as.factor(tax_class),data=nyc_train)
y_test_14<-predict(fit_14,nyc_test)
RMSE_test_n14<-sqrt(mean((y_test_14 - nyc_test$price_lsf)^2))
RMSE_test_n14
```

```
## [1] 703.4175
```

```
RMSE_linear<-bind_rows(RMSE_linear,tibble(Method="Linear FAR, BOROUGH,
tax, building",RMSE=RMSE_test_n14))
```

Again, the decision is correct, RMSE does not change. We now will develop a model with just BOROUGH and FAR.

FAR, BOROUGH. (15)

```
fit_15<-lm(price_lsf~FAR+as.factor(BOROUGH),data=nyc_train)
y_test_15<-predict(fit_15,nyc_test)
RMSE_test_n15<-sqrt(mean((y_test_15 - nyc_test$price_lsf)^2))
RMSE_test_n15

## [1] 707.809

RMSE_linear<-bind_rows(RMSE_linear,tibble(Method="Linear FAR and
BOROUGH",RMSE=RMSE_test_n15))
```

Now, we have increased RMSE by (4.7433137) which means that FAR and BOROUGH are the most important variables. Now we examine which of building class or tax class is more useful.

FAR, BOROUGH, building class (16)

```
fit_16<-
lm(price_lsf~FAR+as.factor(BOROUGH)+as.factor(building_class),data=nyc_tr
ain)
y_test_16<-predict(fit_16,nyc_test)
RMSE_test_n16<-sqrt(mean((y_test_16 - nyc_test$price_lsf)^2))
RMSE_test_n16

## [1] 703.8785

RMSE_linear<-bind_rows(RMSE_linear,tibble(Method="Linear FAR BOROUGH and
building",RMSE=RMSE_test_n16))
```

We obtain a even better prediction than with the all-included. therefore, building class is better variable than tax class. The three important variables to be taken into account are FAR (continuous), BOROUGH (categorical) and building class (categorical)

Summary linear models

we can now compare all results of linear models:

```
RMSE_linear

## # A tibble: 6 x 2
##   Method                              RMSE
##   <chr>                              <dbl>
## 1 Linear FAR                          793.
## 2 Linear all-included                 703.
## 3 Linear without com & res            703.
## 4 Linear FAR, BOROUGH, tax, building  703.
```

```
## 5 Linear FAR and BOROUGH              708.
## 6 Linear FAR BOROUGH and building     704.
```

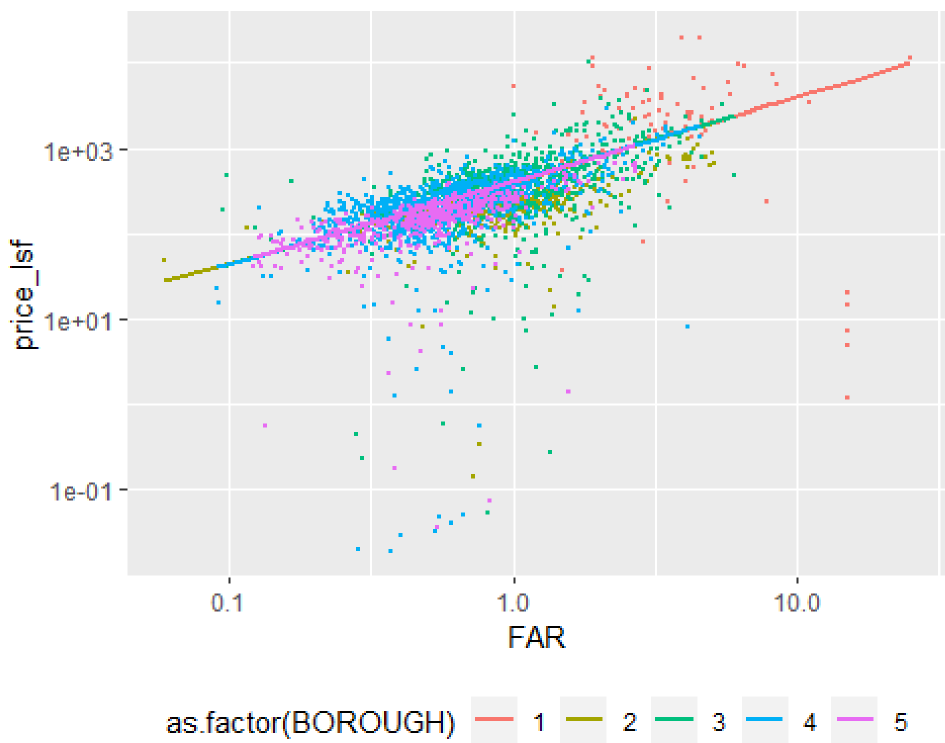
It is a bit surprising that the last model (FAR, borough and building class) provides a similar prediction that all-included model. Anyway, we can choose both models as our final linear model. We choose the three variables model. Furthermore, for the rest of models we are aware that we must develop just one of these three possibilities:

- Only FAR.
- FAR and BOROUGH.
- FAR, BOROUGH and building class.

We will select one of this three type of models depending on the difficulty of the model (KNN, LOESS and TREES). We will try to use the three variables. Just in case the time of calculation is really exagerated we avoid firstly building class and secondly borough and we see if the predicition is better than the best linear prediction.

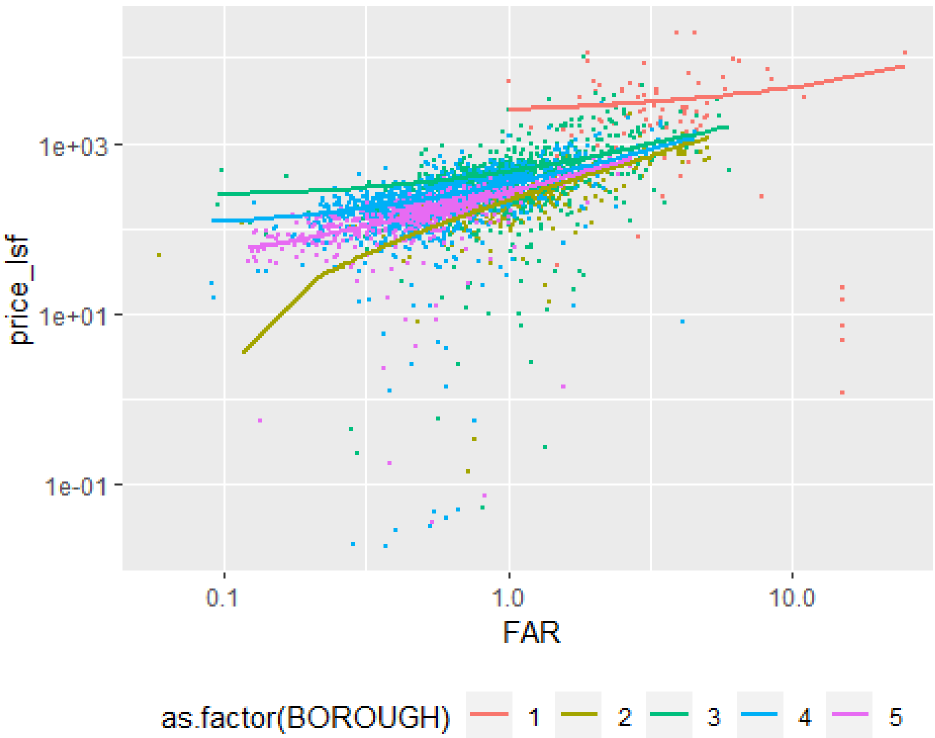
We plot, the three solutions with FAR in x axis and price per square feet in Y axis. We add a logarithmic scale for Y axis. Points represent test dataset and the line represents the prediction. We wanted to differenciate boroughs by color. In all graphics we see that borough has higher FAR and prices. Indeed, there are no reported sales under FAR 1.0. Furthermore, in borough 1 the dispersion of points (right part of graphic) is important.

```
nyc_test%>%mutate(model_1=y_test_11,model_2=y_test_15,model_3=y_test_16)%
>%
  ggplot(aes(x=FAR, y=price_lsf, group=as.factor(BOROUGH),
             color=as.factor(BOROUGH)))+geom_point(size=0.5)+
  scale_y_log10()+scale_x_log10()+theme(legend.position="bottom")+
  geom_line(aes(y=y_test_11),size=1)
```



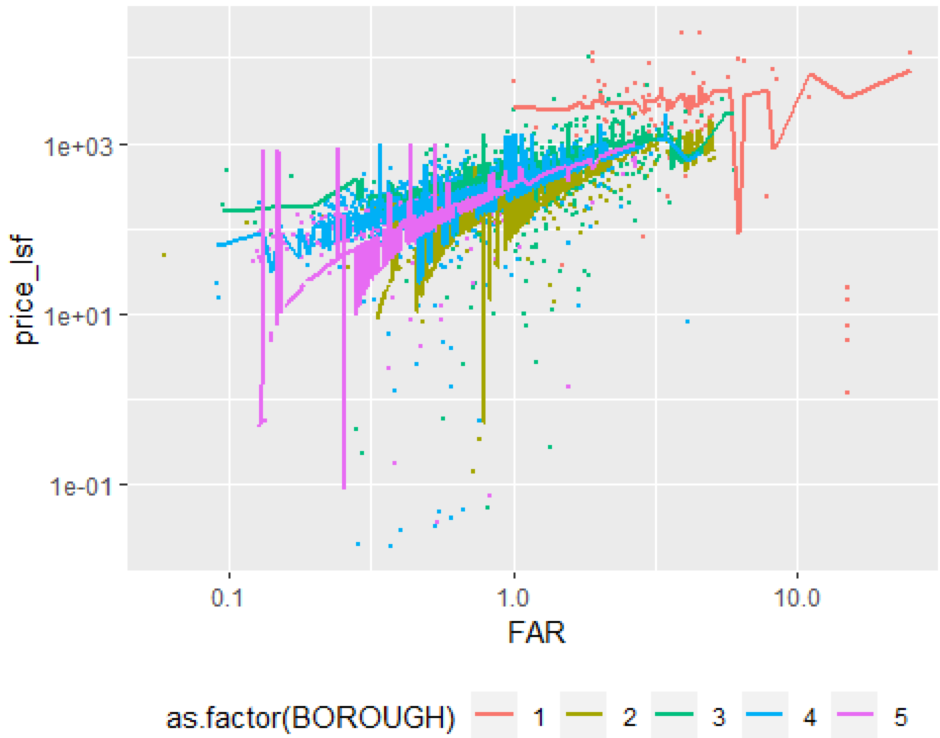
The first solution is the only far model. The solution is a line.

```
nyc_test%>%mutate(model_1=y_test_11,model_2=y_test_15,model_3=y_test_16)%
>%
  ggplot(aes(x=FAR, y=price_lsf, group=as.factor(BOROUGH),
             color=as.factor(BOROUGH)))+geom_point(size=0.5)+
  scale_y_log10()+scale_x_log10()+theme(legend.position="bottom")+
 geom_line(aes(y=y_test_15),size=1)
```



The second solution is the FAR and borough model.

```
nyc_test%>%mutate(model_1=y_test_11,model_2=y_test_15,model_3=y_test_16)%
>%
  ggplot(aes(x=FAR, y=price_lsf, group=as.factor(BOROUGH),
             color=as.factor(BOROUGH)))+geom_point(size=0.5)+
  scale_y_log10()+scale_x_log10()+theme(legend.position="bottom")+
  geom_line(aes(y=y_test_16),size=1)
```

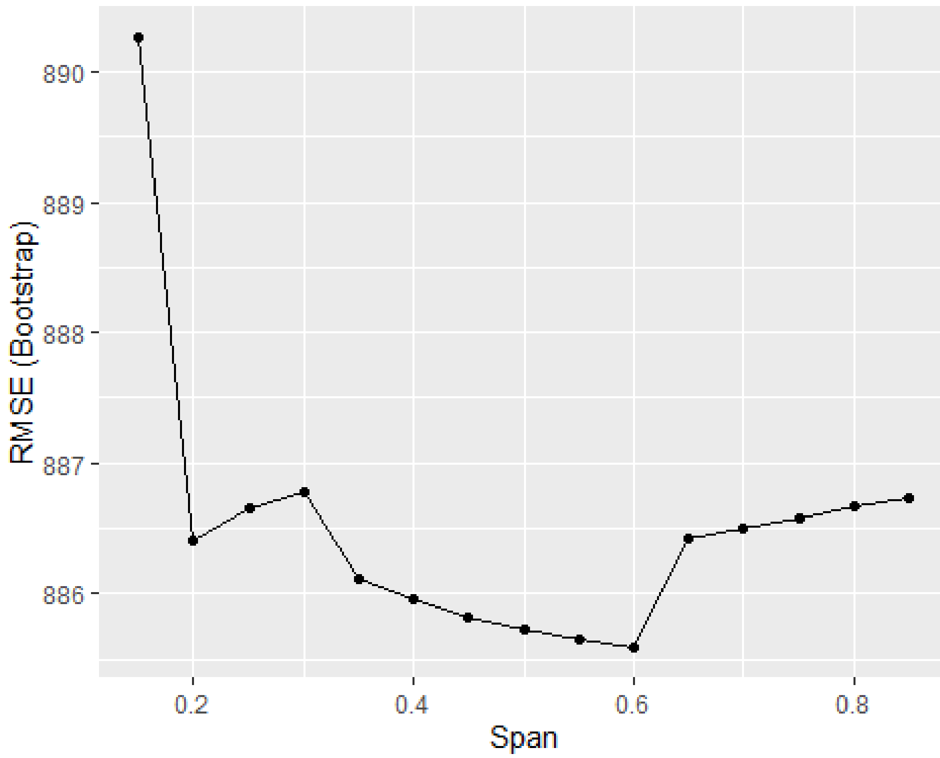


The third solution is the linear model with FAR, borough and building classes. It shows irregularities due to the addition of classes.

Loess model (case 2)

Loess models are recommended for time depending regressions. We are going to use the packe caret to directly tune the parameters. In case of loess models the parameter we must tune is the lengh of the span we use for the approximations. We are going to use search for the best span in an interval between 0.15 and 0.85 with 15 values (we have tried unsuccesfully with other intervals). Firstly, we try a model with 1 variable (FAR). After that we will add BOROUGH and building class if the computer tolerates it. In the next chunk we include the code of loess model, its parameters, the selection of the best parameter, and the RMSE.

```
grid <- expand.grid(span = seq(0.15, 0.85, len = 15), degree = 1)
fit_2<-train(price_lsf ~ FAR, method = "gamLoess",
             data = nyc_train, tuneGrid = grid)
y_test_2<-predict(fit_2,nyc_test)
ggplot(fit_2)
```

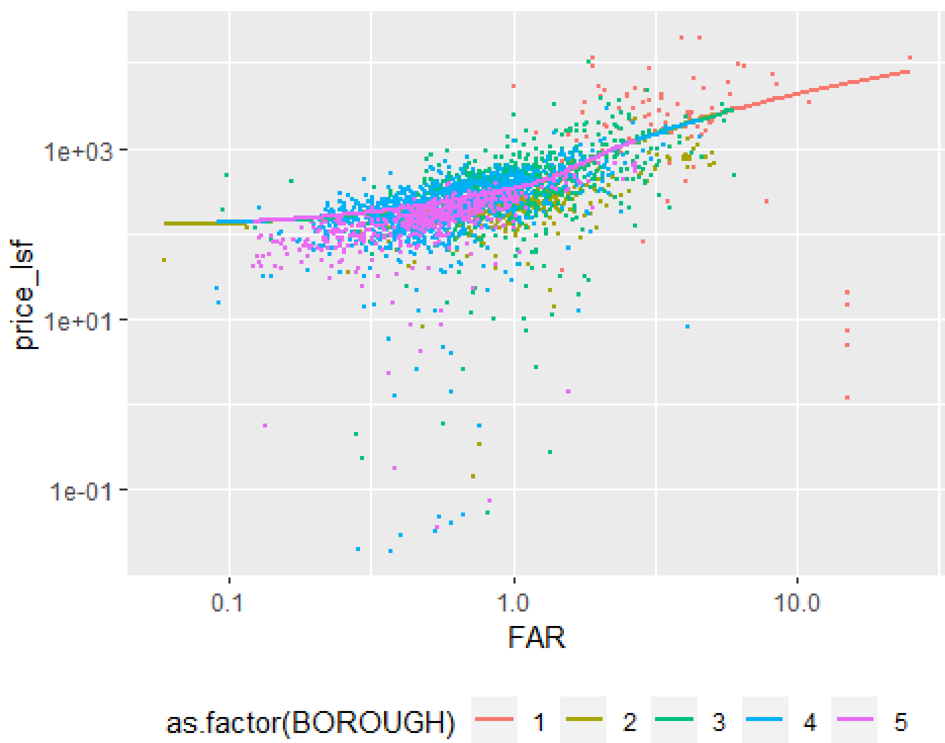



```r
fit_2$bestTune

##    span degree
## 10  0.6      1

RMSE_test_2<-sqrt(mean((y_test_2 - nyc_test$price_lsf)^2))
RMSE_test_2

## [1] 782.1079
```

We reported computing problems with loess model with 2 variables (FAR and borough). Therefore, our loess model will be the only FAR model, whose accuracy is low. Its RMSE (782.1079449) is comparible with RMSE of the only FAR linear model (793.1703664), clearly above the best linear model achieved.

We include the graphic:

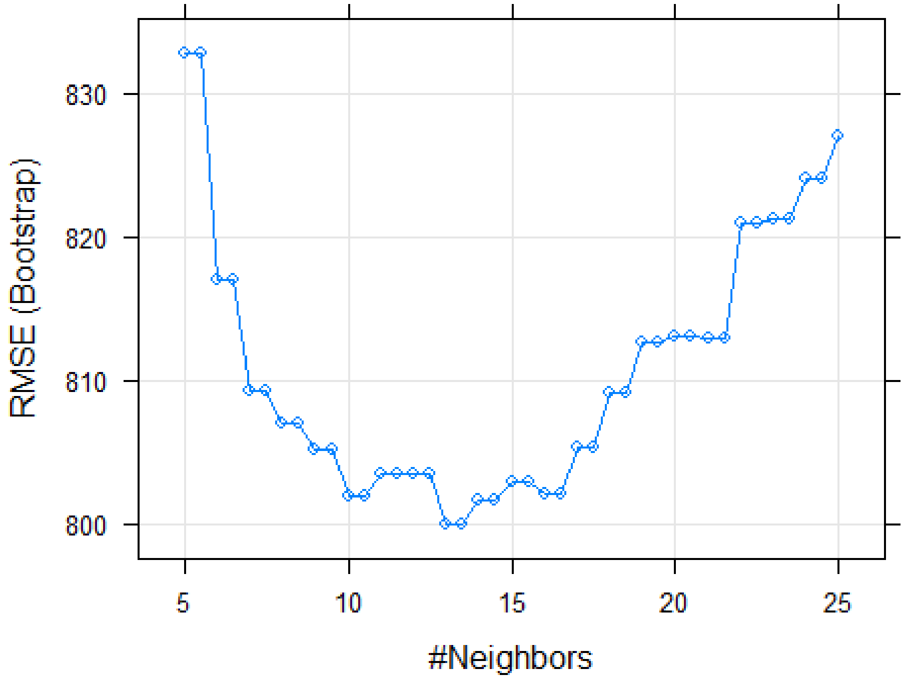
```r
nyc_test%>%mutate(model_loess=y_test_2)%>%
  ggplot(aes(x=FAR, y=price_lsf, group=as.factor(BOROUGH),
             color=as.factor(BOROUGH)))+geom_point(size=0.5)+
  scale_y_log10()+scale_x_log10()+theme(legend.position="bottom")+
  geom_line(aes(y=y_test_2),size=1)
```



Knn model (case 3)

In Knn parameters we must optimize the value of k which is the number of neighbors. We have tried with different intervals. We just show one of them that provides us with the minimum obtained with tuning. In the code, we can see the tuning of the model (fit_3), the optimization in plot, the prediction using the model fit_3 and the accuracies.

```
fit_3<-train(price_lsf ~
FAR+as.factor(BOROUGH)+as.factor(building_class), method = "knn",
             data = nyc_train, tuneGrid = data.frame(k = seq(5, 25,
0.5)))
plot(fit_3)
```

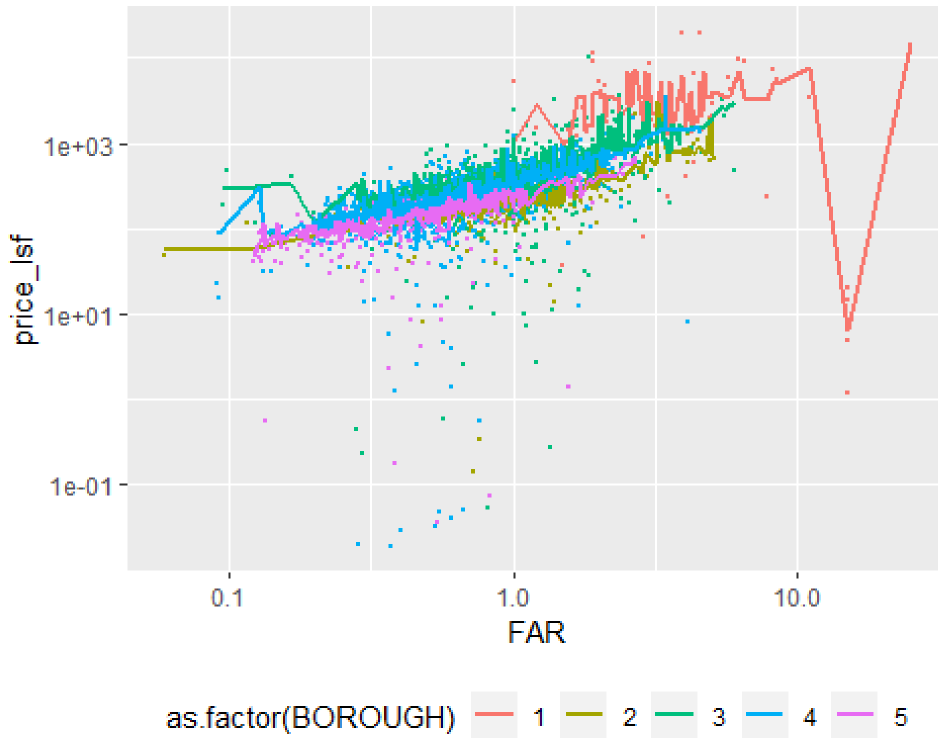



```
y_test_3<-predict(fit_3,nyc_test,type="raw")
RMSE_test_3<-sqrt(mean((y_test_3 - nyc_test$price_lsf)^2))
RMSE_test_3

## [1] 632.9971
```

In this case, our RMSE is (632.9970864), which is a better prediction, (70.8814303), than the best linear model. Knn models require more calculation time but they provide more accurate solutions because the surroundings correct the approximation (neighbors). With caret package we directly select the amount of neighbors (see plot). An elevated k supposes that we take too many neighbors and the prediction may differ from the data itself. Whereas a low k offers a too wriggly approximation because each sale is approximated by itself.

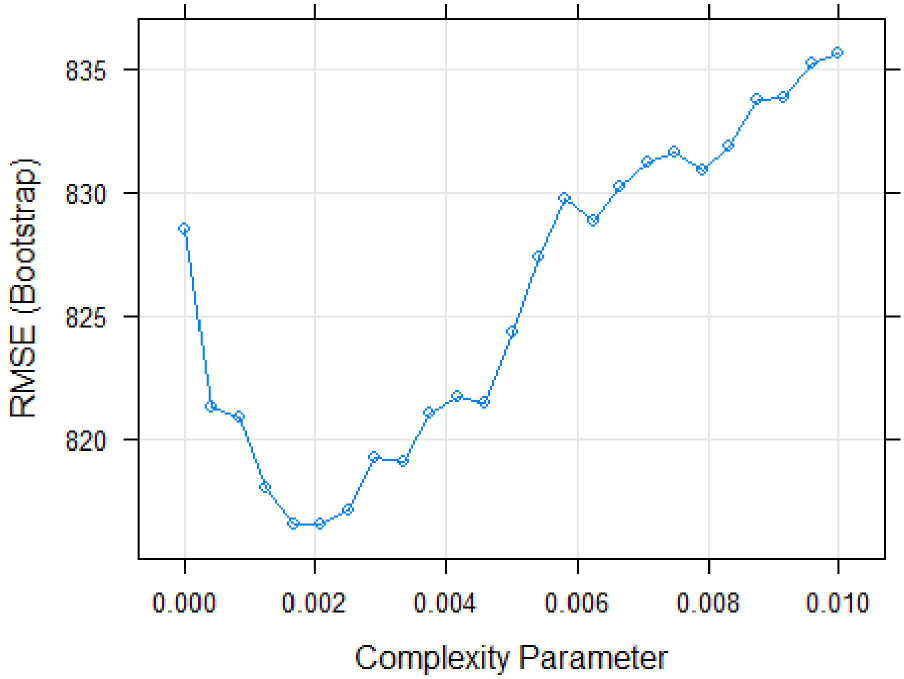
We include the graphic:

```
nyc_test%>%mutate(model_knn=y_test_3)%>%
  ggplot(aes(x=FAR, y=price_lsf, group=as.factor(BOROUGH),
             color=as.factor(BOROUGH)))+geom_point(size=0.5)+
  scale_y_log10()+scale_x_log10()+theme(legend.position="bottom")+
  geom_line(aes(y=y_test_3),size=1)
```

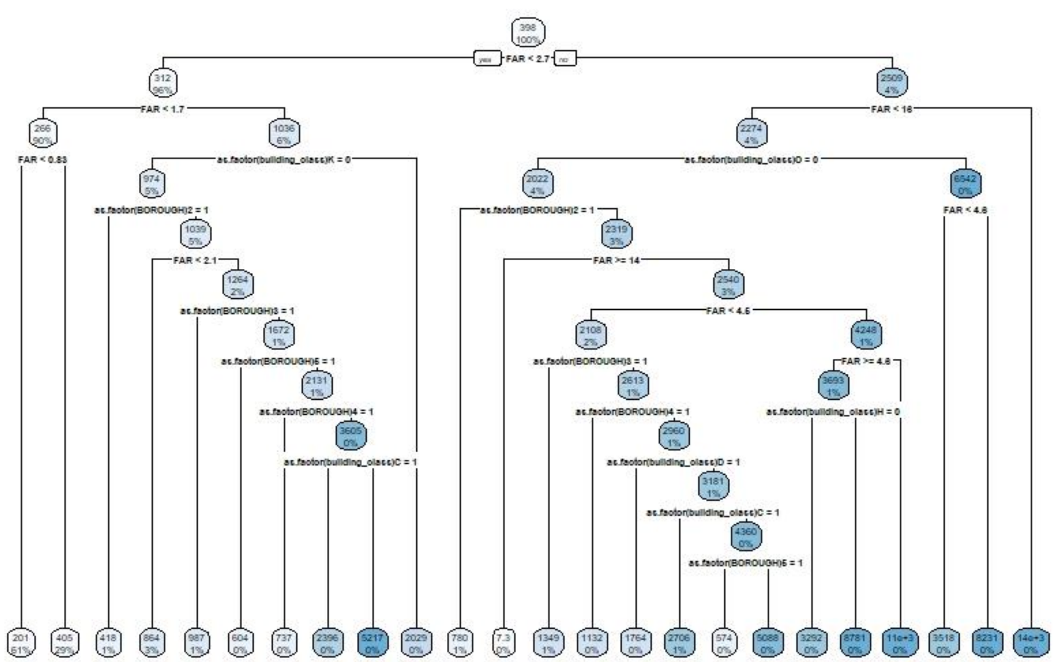


Tree model (case 4)

```
fit_4 <- train(price_lsf ~
FAR+as.factor(BOROUGH)+as.factor(building_class), method = "rpart",
                     tuneGrid = data.frame(cp = seq(0, 0.01, len = 25)),
data = nyc_train)
plot(fit_4)
```




```
rpart.plot(fit_4$finalModel)
```

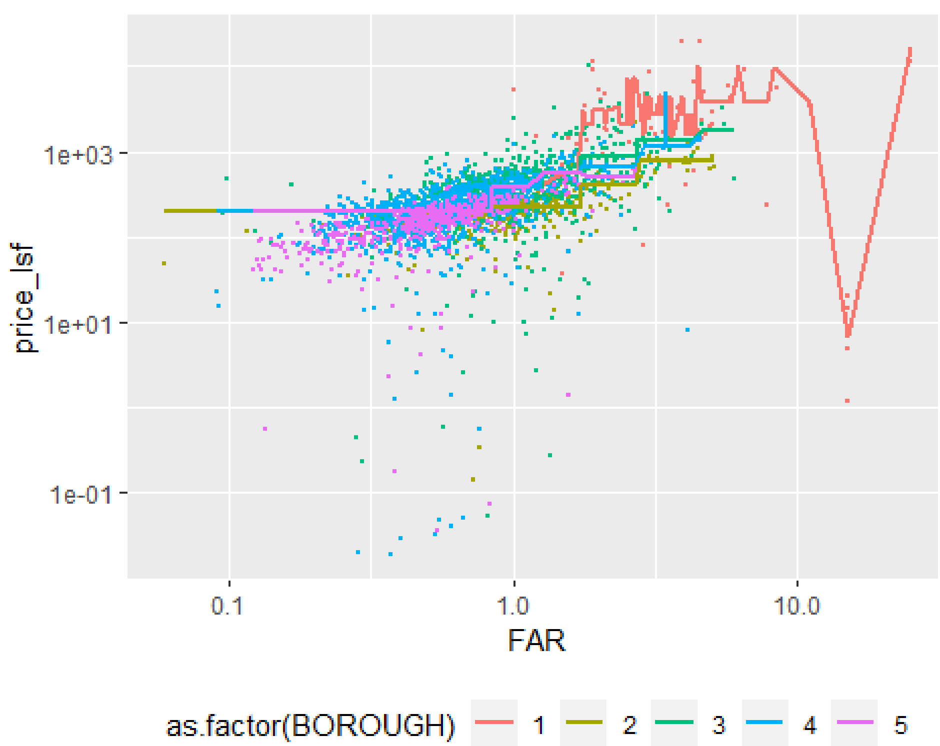


```
y_test_4<-predict(fit_4,nyc_test,type="raw")
RMSE_test_4<-sqrt(mean((y_test_4 - nyc_test$price_lsf)^2))
RMSE_test_4

## [1] 662.0038
```

In this case, our RMSE is (662.0037944), which is a better prediction, (41.8747223), than the best linear model, but worse than Knn model.

We include the graphic:

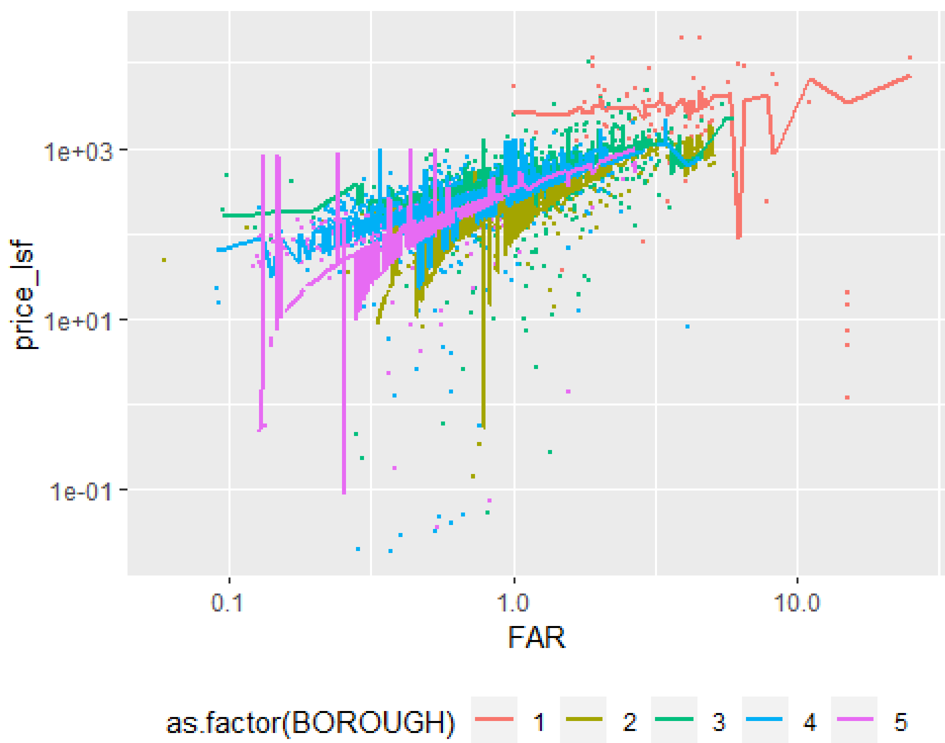
```
nyc_test%>%mutate(model_tree=y_test_4)%>%
  ggplot(aes(x=FAR, y=price_lsf, group=as.factor(BOROUGH),
              color=as.factor(BOROUGH)))+geom_point(size=0.5)+
  scale_y_log10()+scale_x_log10()+theme(legend.position="bottom")+
  geom_line(aes(y=y_test_4),size=1)
```



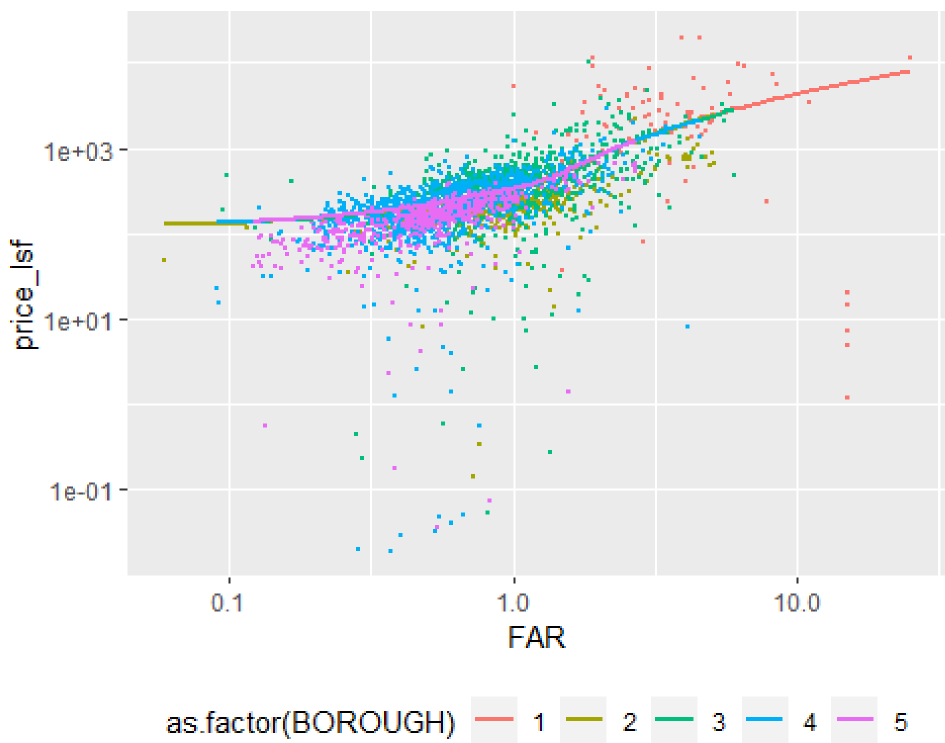
Results

We include a summary of results (graphics):

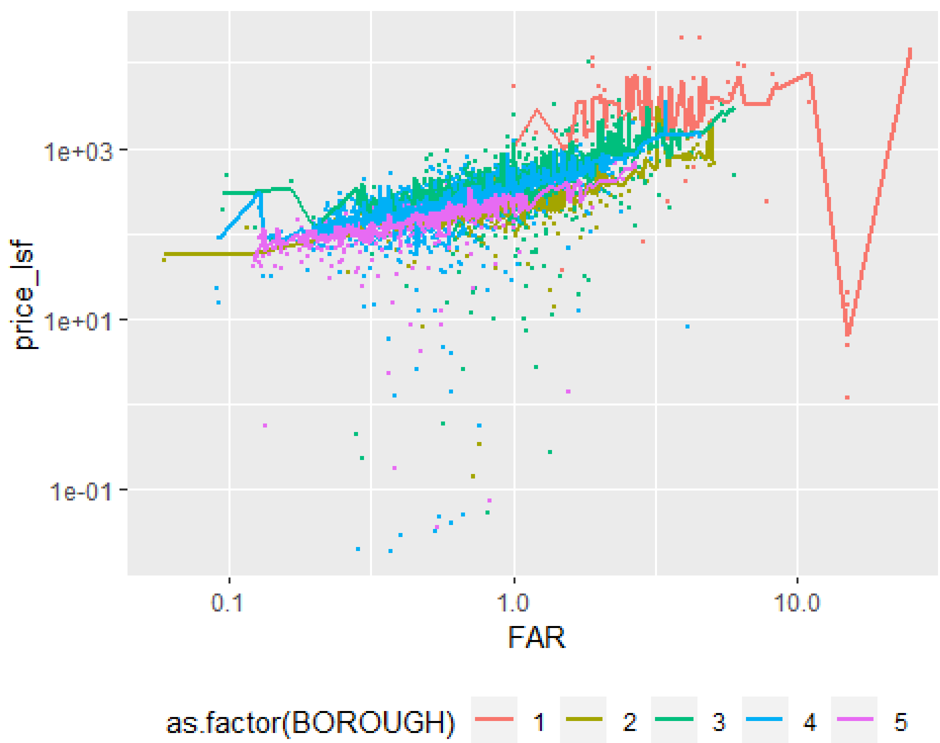
```
# Graphic. Linear. Selected (3x)
nyc_test%>% ggplot(aes(x=FAR, y=price_lsf, group=as.factor(BOROUGH),
              color=as.factor(BOROUGH)))+geom_point(size=0.5)+
  scale_y_log10()+scale_x_log10()+theme(legend.position="bottom")+
  geom_line(aes(y=y_test_16),size=1)
```




```
# Graphic. Loess. FAR (1x)
nyc_test%>%  ggplot(aes(x=FAR, y=price_lsf, group=as.factor(BOROUGH),
             color=as.factor(BOROUGH)))+geom_point(size=0.5)+
  scale_y_log10()+scale_x_log10()+theme(legend.position="bottom")+
  geom_line(aes(y=y_test_2),size=1)
```




```
# Graphic. Knn. (3x)
nyc_test%>% ggplot(aes(x=FAR, y=price_lsf, group=as.factor(BOROUGH),
              color=as.factor(BOROUGH)))+geom_point(size=0.5)+
  scale_y_log10()+scale_x_log10()+theme(legend.position="bottom")+
  geom_line(aes(y=y_test_3),size=1)
```




```
# Graphic. Tree. (3x)
nyc_test%>%  ggplot(aes(x=FAR, y=price_lsf, group=as.factor(BOROUGH),
             color=as.factor(BOROUGH)))+geom_point(size=0.5)+
  scale_y_log10()+scale_x_log10()+theme(legend.position="bottom")+
  geom_line(aes(y=y_test_4),size=1)
```

We can see that there are three different type of graphics. As explained in linear models, the inclusion of building class implies that the representation of the prediction is more wriggly as the prediction tries to be "more local" and to adapt more to the variability of data (Knn,linear selected). In models with just one variable (loess) the graphic is represented by a line because it represents only the relation between FAR and price. Finally, trees are completely different. They try to cover the prediction of a span (tuned). They performance is logic with decisions. As a result, the graphic is staggered.

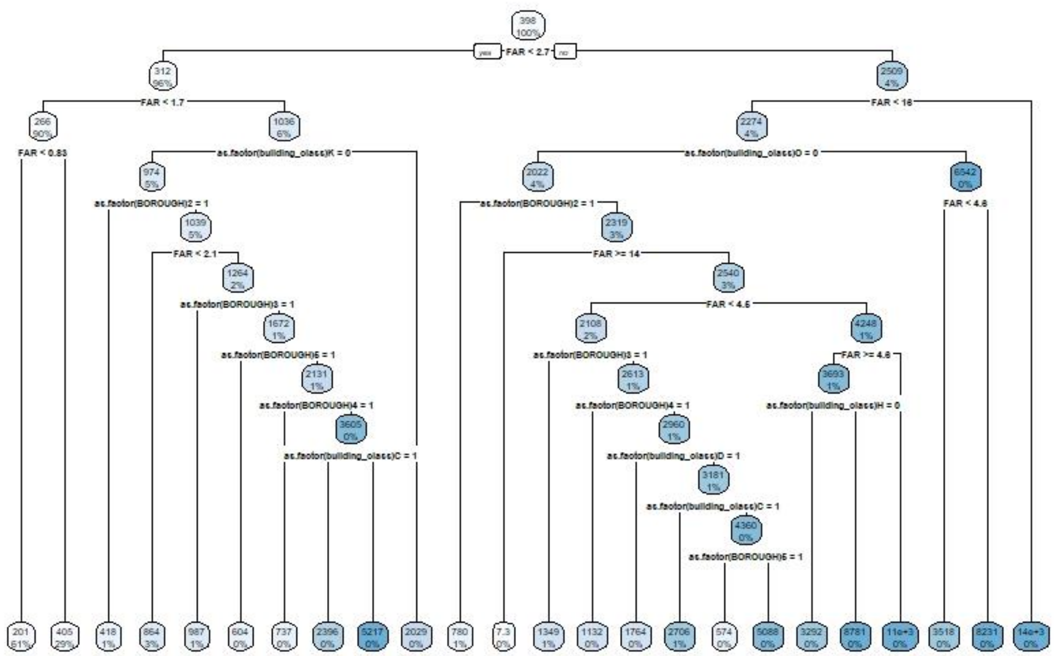
Conclusion

We include a summary of accuracies (RMSE by model)

We think that our initial goal has been mainly accomplished. We have obtained some models that offer us a prediction. Furthermore, we tried to use the linear models to project models to improve results obtained from them. We have developed a tree model that is better than the linear model and a Knn that is better than the linear model.

Nevertheless, we consider the RMSE is too high to be satisfied. Taking into account all data, it may be satisfactory. However, we cannot forget that the great amount of data has low prices, even comparible with the proper RMSE. Then, we must wonder if the prediction is so bad. The answer to this question is the inclucence of the segment of data with really variable and high prices that distort the overall results. This segment of data is really related to the inclusion of data coming from Borough 1. We explain this issue in detail in the limitations section: borough. The problem is also related to the building classes that do not pertain neither to residential not to commercial which are difficult to predict.

Our sales, once summaryzed the main variables is based on FAR (continuous variable) and Borough and building class (categorical). Besides, we detected the high influence of borough. Data is really different in borough 1 than in the others. So, we tried to create a double model. One for borough 1 and one for the others. Nevertheless, we did not find a way to reflect this effect easily. The most simple way of considering this division was a tree of decission that may be able also to include the importance of building classes by separating the most problematic. We are satisfied that our tree model has achieved a better accuracy than the best lineal model.



Limitations.

Computing limitations

We thought that the capacity of the computer would be an issue. We had not have major problems with it. We could not use all the accuracy that a loess model could have provided. We had to stop at Only FAR model. This is the main issue reported. Nevertheless, the tuning ofknn and tree models although has been accomplished, has required too much time. The time has prevent us from improving other aspects of the code but has not have a direct infuence on results.

Available data.

First of all, we were obliged to eliminate many reported sales with NA's in important variables such as price itself or land. Some variables were nor reported (just the name) and others were not useful. We want to explain difficulties in the variables that were indeed used.

Year built.

First of all we noticed that there were reported sales for 19th century. They could have been used but we thought that there were not representative enough as there were just a few sales for some years where the great amount of sales concentrated on the 20th century. Besides, the data was irregular even in 20th century, with many sales building in a specific year and a few for the rest of the years of the century.

Units

We were interested in using the proportion of commercial and residential units but we did not count on a efficient methodology to use it. Our calculated percentage was unuseful. Besides, many units did not pertain to commercial or residential because they were equipment or industrial or other type of buildings. They were reported as TOTAL UNTIS 0.

Price

We eliminated prices reported as 0 and below 10. Analyzint the data we considered that these data were erroneuos. It was not a low prices but an error. In the other hand, we have noticed that prices were too variable. We can see this characteristic of the data in the boxplots. We have many sales with a low prices and some very high prices. Nevertheless, we have not eliminated these expensive units despite it seems they are outliers. We explain why in the next parts:

Borough

This variable may the one of the most intresting of our study. We have verified that there were important differences in the data between Borough 1 (Manhattan) and the others. The variability of the data in Manhattan was greater than in the others. Apart from allocating the richest people in the city it allocates many headquarters of important companies, industrial facilites, stadium and others. The more expensive prices has been found in BOROUGH 1. Many of them correspond to neighet commercial nor residential buildings. So the prediction was more difficult even. Summarizing, in taking into account Manhattan's data, we include the most difficult data to predict: many outliers, the most expensive sales, variability and many building classes. We have two possibilites: either avoiding this data (borugh 1) or assuming our prediction will be worse. In other words the two possibillites will be: either predicting boroughs 2 to 5 with a better precision and giving up the prediction of borough 1 or predicting all with a worse accuracy. (We have verified that the inclusion of borough 1 supposes the worsening of the accuracy. We have not included this calculation because it divers from the general goal.It was made with a linear model and the comparison of general RMSE, borough 1 RMSE and borough 2to5 RMSE).

Building class

We have a similar problem as the explained in BOROUGH. We have many building classes different from residential and commercial. The train dataset includes many residential and commercial prices. Therefore, we can predict there building classes easily. Nevertheless there are some others classes with a few of reported sales. There is a relation between building clases different from residential and commercial and borough 1. Again we did have a choice: either avoiding the prediction of this building classes and getting a better accuracy in the main ones or trying to predict all. As with boroughs we have decided not giving up our intial goal.

Given the influence of clases (Building and Borough) we decided to try to use a tree model that has not been as efficient as expected. We did not know a technique that could have enhanced the prediction of borough 1 and building clases neither commercial not residential.

Program escope

In our research we have found some models, methodologies and concepts that may improve our prediction. Nevertheless we have limited our study to the concepts of the program. Anyway, we want to briefly mention some of them here:

- Splitting: for time depending studies the splitting is based on windows of time instead of our simpler splitting.
- ARIMA: "AutoRegressive Integrated Moving Average" for time series whose standard deviation do not vary in time. It tries to search for patterns in the pass to use them in the future.
- ANOVA: it is a useful technique that studies the variance of the groups and their average. This could be an interesting approach for our problem with boroughs and building classes.
- AIC, BIC: techniques to regularize and select the best linear model.

R Markdown

R Markdown cache. We have confirmed that with the same seed, parameters and code, R Markdown provides us with different results than R script. This has been a great difficulty because there were some differencies of RMSE and the analysis was different (different approach after linear models, different results and different conclusions). We have tried as much as possible to mantain the decissions, approaches and conclussions of our initial work with R script in the final report. We have researched and included a comment in the discussion of the forum regarding this issue. We have found that it is related to the internal working system of R markdown. The elimination of the environment (which would be the solution in R script) is not enough. We have found that the solution is related to the "cache" of R Markdown but we have not been able to solve it.

Improvements and future works.

Considering conclusions of the RMSE results and limitations we have thought of different ways to improve the performance with the same data.

Percentages

We were interested in using the percentages of units (comercial and residential). This tool could be useful for enhancing the prediction of the segment of data with less variability and lower prices. The correlation obtained was not succesful. We would have liked to develop a more complex rule to take this into account. Firtly, a unique continuous variable instead of two variables.

Outliers and incomplete data.

We had to eliminate some sales because the year of building was 0 or 19th century. At first we thought that this was an important variable. Once we know that year built is not used in models we could go back and not eliminate these sales as they have no influence in our three final variables. The sales reported as below 10 neccesarily must be eliminated. These data may have distorted the model even more. In relation to other possible outliers they are related mainly to borough or building class.

Borough

As we have explained we would have liked to develop to different models (borough 1 and the others) and then unit them. The difficulty of this approach is highly over our knowledge. We just tried to repeat a linear model separate to verify that borough 1 was a problem (already explained).

Building classes.

A possibility not explored was the gathering of classes in a more reduced number. Residentials, commercials, industrial, equipment. We do not know if this approach may enhance the result. The gathering will be difficult itself. It is not easy to mix all the building classes apart from residential and commercial in a few types because they are important concept differences among them. One must study carefully[17] to make this division. We have already supposed that we could gather the subgroups (numbers) in the same group. A quick glance in the link could justificate the assumption.

[17] https://www1.nyc.gov/assets/finance/jump/hlpbldgcode.html

ANNOVA

Considering the variability we have noticed among groups (borughs or building classes), ANNOVA would be useful. This technique is based indeed the variability among groups. Nevertheless, this techniques are over our scope.

ARIMA

We could not use the time variability. The price oscillates depending on the year of building. It is difficlt to stablish the relation but there is a relation. The prices is not a constant. On the opposite, the price is almost constant during the year of study. Given that there is a time variable, and there is a relation, ARIMA could find the tendency and use it to improve the prediction of sales. Nevertheless, we are not using data from the past to predict the future. So, the utility of ARIMA would not be so high. Year built is not exaclty a continuous time variable. The continuous time variable is sale date and it does not show tendency. The use of ARIMA is indicated in case we had larger series of sale dates with clear tendencies and we wanted to predict sales for the future.

References and sources

Some of these links include information in Spanish. We can speak english and Spanish so they have been useful. They include information of splitting ratio, modeling techniques, R Markdown, patterns for machine learning workflows, data exploration techniques and graphics, and information of NYC.

1. Machine Learning with caret. (https://rpubs.com/Joaquin_AR/383283)
2. Multiple linear regression guide. (https://rpubs.com/MStenroos/385153)
3. Linear regression on Car Price Prediction. (https://rpubs.com/Argaadya/531140)
4. Price Forecasting Using Time Series Analysis, Machine Learning and single layer neural network Models (https://rpubs.com/kapage/523169)
5. Correlation and regression. (https://www.cienciadedatos.net/documentos/24_correlacion_y_regresion_lineal)
6. ANOVA (https://www.cienciadedatos.net/documentos/19_anova)
7. Predicting Housing Price with R. ARIMA. (https://towardsdatascience.com/predicting-housing-prices-with-r-c9ec0821328d)
8. R Markdown cheat seet (https://rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf)
9. knitr with R Markdown (https://kbroman.org/knitr_knutshell/pages/Rmarkdown.html)
10. Data Splitting. (https://towardsdatascience.com/data-splitting-technique-to-fit-any-machine-learning-model-c0d7f3f1c790)
11. ARIMA. (https://rpubs.com/valeamasso/386527)

12. Boroughs of NYC. (https://en.wikipedia.org/wiki/Boroughs_of_New_York_City)
13. Building classes of NYC (https://www1.nyc.gov/assets/finance/jump/hlpbldgcode.html).
14. Tax classes of NYC 2017 (http://www.cynthiaramirez.nyc/uploads/1/1/8/7/118704813/class_1_guide__1_.pdf)
15. Definition of FAR ratio (https://en.wikipedia.org/wiki/Floor_area_ratio).