Prediciting House Rental Price

Eline Morais

19/06/2020

INTRODUCTION

The aim of this data analysis project is to create an automated model of prediction to property rental values without human bias using regression machine learning algorithms.

For the owner of a property, determining the rental price is very important to make his investment worthwhile. If you charge less you will be losing money and if you charge too much, you may not find a tenant willing to pay and the property will be empty for a long period becoming more of a cost, and extending the period of payback.

In the same way, the person who is going to rent, is always looking for the best cost-benefit trying to find a property well located, comfortable and at the lowest possible price.

Knowing that both, "buyers and sellers" have contrary and biased interests, it is necessary to develop an independent technique to calculate the fair price of a house according to several factors that affect the price, such as the physical condition, location, size, etc.

The data source used for this project is obtained from https://www.kaggle.com/rubenssjr/brasilian-houses-to-rent?select=houses_to_rent_v2.csv, also available at Github https://raw.githubusercontent.com/emoraiss/brazilian_rent/master/datasets_554905_1035602_houses_to_rent_v2.csv".

The data files above consists of information from about 10.692 houses to rent in different cities in Brazil and 13 different features. According to Rubens Junior, who published this dataset, the data was collected from a rental website, on 3/20/20.

The dataset was randomly partitioned into two tables by a 90%$$10% ratio. The first table with 90% partition was used to build the model while the second table was used as a validation dataset for the would-be developed algorithm.

To measure the assertiveness of the models and define the best approach, the RMSE will be used. RMSE - Residual Mean Squared Error is the square root of the average of squared errors. Error, in turn, is the difference between the prediction and the actual outcome, therefore the lower RMSE the better.

This project is a part of the final evaluation of students on the Data Science Professional Certificate of Harvardx - Casptone, where each student must choose a dataset on internet and use tools they learned during the course, applying machine learning.

EXPLORATORY DATA ANALYSIS - EDA

Data extraction

From this project we are going to use the following packages. Specific machine learning packages will be loaded at the time of training data later.

```
library(tidyverse)
library(caret)
library(readr)
library(httr)
library(dplyr)
library(ggplot2)
library(ggthemes)
library(gridExtra)
library(stringr)
library(knitr)
library(lubridate)
library(rvest)
library(matrixStats)
library(purrr)
```

Files will be downloaded from url above, and translating them into a structured tidy format is illustrated as shown below:

```
#Accessing the data - Github
tmp<- tempfile()
download.file("https://raw.githubusercontent.com/emoraiss/brazilian_rent/master/datasets_554905_1035602_
data<-read_csv(tmp)
file.remove(tmp)
```

```
## [1] TRUE
```

Data structure

We can start confirming dimensions of dataset:

```
## [1] 10692    13
```

And having a look on features:

```
##  [1] "city"                "area"                "rooms"
##  [4] "bathroom"            "parking spaces"      "floor"
##  [7] "animal"              "furniture"           "hoa (R$)"
## [10] "rent amount (R$)"    "property tax (R$)"   "fire insurance (R$)"
## [13] "total (R$)"
```

The first step in building an algorithm is to better understand the outcomes and features. Below follows a brief description of them:

#Outcome:

-rent amount (R$) - Rent amount

#Features:

-city - City where the property is located

-area - Property area in square meters

-rooms - Quantity of rooms

-bathroom - Quantity of bathrooms

-parking spaces - Quantity of parking spaces

-floor - In which floor the apartment is located, in case of apartment

-animal - Accept or not accept animals?

-furniture - Is furbished or not?

-hoa (R$) - Homeowners association tax - condominium that makes and enforces rules for the properties and their residents, monthly paid for common-area and facilities upkeep.

-property tax (R$) - By definition, property tax is the tax paid on property owned by an individual or other legal entity, such as a corporation. The tax is usually calculated by a local government where the property is located and paid by the owner of the property, based on the value of the owned property, including land.

-fire insurance (R$) - is a property coverage that pays for damages to property and other losses you may suffer from a fire.

-total (R$) - Total amount of 4 previous values

As a good practice and to facilitate the coding process we will remove spaces, punctuation and curly brackets from titles, resulting on these news titles:

```
##  [1] "city"           "area"           "rooms"          "bathroom"
##  [5] "parking_spaces" "floor"          "animal"         "furniture"
##  [9] "hoa"            "rent_amount"    "property_tax"   "fire_insurance"
## [13] "total"
```

Looking at the summary we can observe there is a small variability in the number of rooms, bathrooms and parking spaces and, on the other hand there is a wide variation in areas, values of rent amount, and other taxes and fees.

```
##      city                area           rooms          bathroom
##  Length:10692       Min.   :   11   Min.   : 1.00   Min.   : 1.00
##  Class :character   1st Qu.:   56   1st Qu.: 2.00   1st Qu.: 1.00
##  Mode  :character   Median :   90   Median : 2.00   Median : 2.00
##                     Mean   :  149   Mean   : 2.51   Mean   : 2.24
##                     3rd Qu.:  182   3rd Qu.: 3.00   3rd Qu.: 3.00
##                     Max.   :46335   Max.   :13.00   Max.   :10.00
##  parking_spaces     floor              animal           furniture
##  Min.   : 0.00   Length:10692       Length:10692       Length:10692
##  1st Qu.: 0.00   Class :character   Class :character   Class :character
##  Median : 1.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 1.61
##  3rd Qu.: 2.00
##  Max.   :12.00
##       hoa           rent_amount    property_tax     fire_insurance
##  Min.   :      0   Min.   :  450   Min.   :     0   Min.   :  3.0
##  1st Qu.:    170   1st Qu.: 1530   1st Qu.:    38   1st Qu.: 21.0
##  Median :    560   Median : 2661   Median :   125   Median : 36.0
##  Mean   :   1174   Mean   : 3896   Mean   :   367   Mean   : 53.3
##  3rd Qu.:   1238   3rd Qu.: 5000   3rd Qu.:   375   3rd Qu.: 68.0
##  Max.   :1117000   Max.   :45000   Max.   :313700   Max.   :677.0
##      total
##  Min.   :    499
##  1st Qu.:   2062
##  Median :   3582
##  Mean   :   5490
##  3rd Qu.:   6768
##  Max.   :1120000
```

We also see that "floor" is classified as "character" and as we can see below for most properties this field is filled with "-". Once we do not have one field to identify the type of property (house or apartment), we could suppose, observations where floor= "-", are houses, what means "do not apply". So, let us change this "-" to "0", and change this field to numeric.

floor	n
-	2461
1	1081
2	985
3	931
4	748
5	600

We see that there is none "NA".

```
sum(is.na(data))
```

```
## [1] 0
```

Our dataset contains houses in 5 different cities in Brazil, with mean area about 150m2, mean price about R$3.900,00, 11 different number of rooms and bathrooms, and 35 different options of floor.

Cities	Mean_area	Room_options	Parking_options	Floor_options	Mean_rental
5	149.2	11	11	35	3896

A more detailed look at the data reveals that most of properties are concentrated in São Paulo, which is responsible for 55% of available houses. The number of properties by cities can be seen on the table below.

city	n	%
São Paulo	5887	55.060
Rio de Janeiro	1501	14.039
Belo Horizonte	1258	11.766
Porto Alegre	1193	11.158
Campinas	853	7.978

Training/Test data set

Before starting the exploratory analysis of the data it is necessary to split the data set into training and validation set.

To create our train and test set we will use "createDataPartition" from caret package, setting seed = 1 to be reproducible. We will take a random sample of 10% as a validation data while the remaining 90% is allocated to build the model.

```
# Validation set will be 10% of data
# if using R 3.5 or earlier, use ‘set.seed(1)‘ instead
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(y = data$rent_amount, times = 1, p = 0.1, list = FALSE)
build<- data[-test_index,]
validation <- data[test_index,]
```

It is important to note that the validation dataset will be considered as an unknown basis to evaluate the accuracy of our final model, trained and tested only with the data from the build dataset. Therefore, the validation set will not be considered in the exploratory analysis.

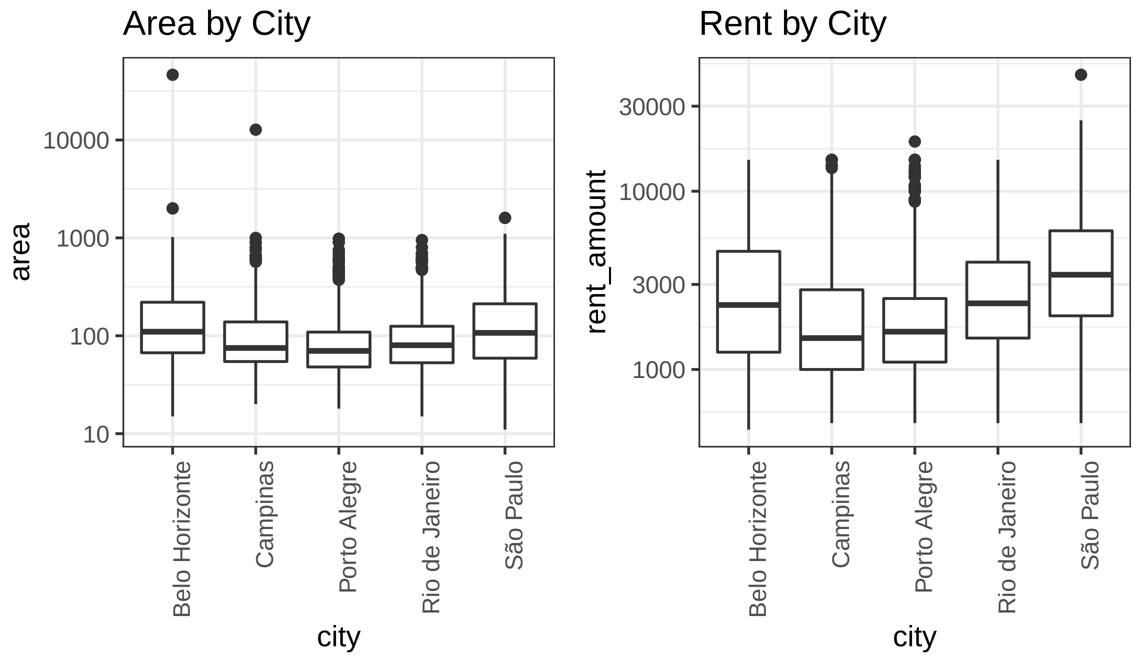
The high ratio 90 $$ 10 was defined due to the relatively small size of our dataset. So, now our “build” dataset contains 9.621 houses and 13 features.

```
## [1] 9621   13
```

Data Visualization

We are going to examine the outcome and its relations with each feature starting by area.

- Area



Comparing area by city, we can see that São Paulo and Belo Horizonte have larger ranges, but in general 50% of properties have areas between 50m2 and 200m2.

Since by definition a mansion is around 2.000 m2, we will consider properties that have an area equal or larger than that as errors or outliers and will remove them. Our dataset has 4 of them.

city	area	rent_amount
Belo Horizonte	46335	8500
Campinas	12732	1600
Belo Horizonte	2000	4956
Belo Horizonte	2000	5000
São Paulo	1600	7600
São Paulo	1600	6900

New dimension:

```
remove_index<- which(build$area>=2000)

build<- build[-remove_index,]

dim(build)
```

```
## [1] 9617   13
```

At the same time we observe that there are 60 properties smaller than 20m2 or 695 smaller than 30m2. But we are going to keep them once its quantity is relevant and because it is normal to find studios or even single rooms to rent at this size.

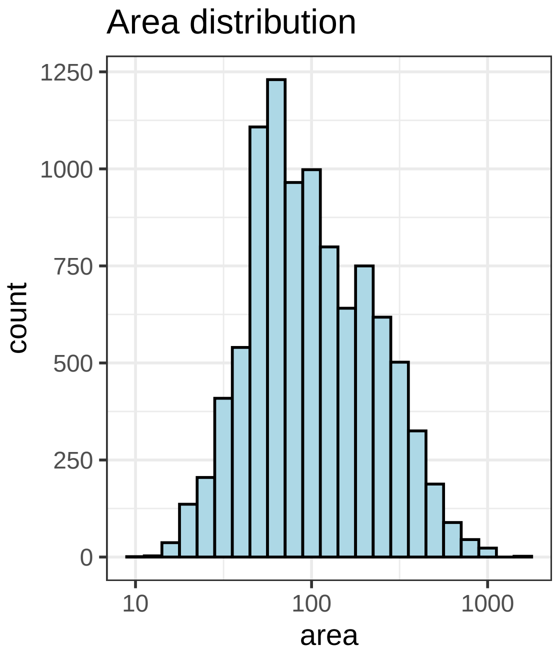
Comparing rent amount by city, we can see that São Paulo presents the most expensive properties, followed by Belo Horizonte and Rio de Janeiro, which have similar behavior, and after Porto Alegre and Campinas presenting the cheapest prices.

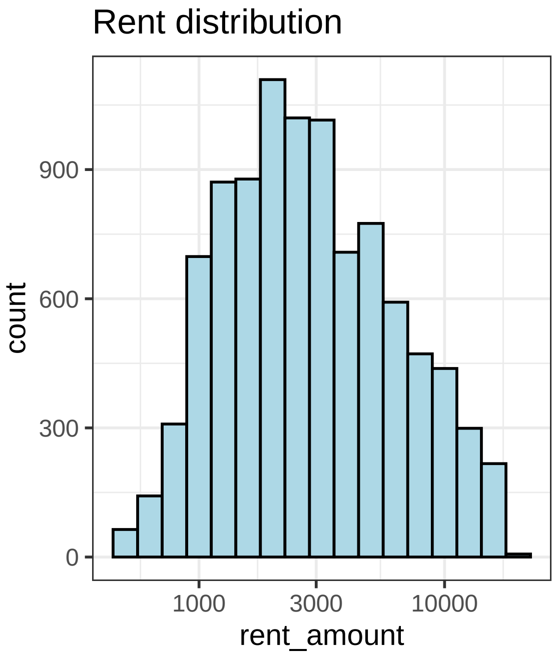
Let us have a look on the highest rental prices.

city	area	rooms	bathroom	parking_spaces	property_tax	rent_amount
São Paulo	700	4	7	8	8750	45000
São Paulo	486	8	4	6	2200	25000
São Paulo	80	2	1	1	0	24000
São Paulo	660	4	5	5	1750	20000
São Paulo	410	4	5	5	0	20000
São Paulo	285	4	5	4	1834	20000
São Paulo	900	3	4	8	3813	20000
São Paulo	455	4	5	4	3334	19500
Porto Alegre	318	4	3	0	384	19000
São Paulo	303	3	4	4	0	18000

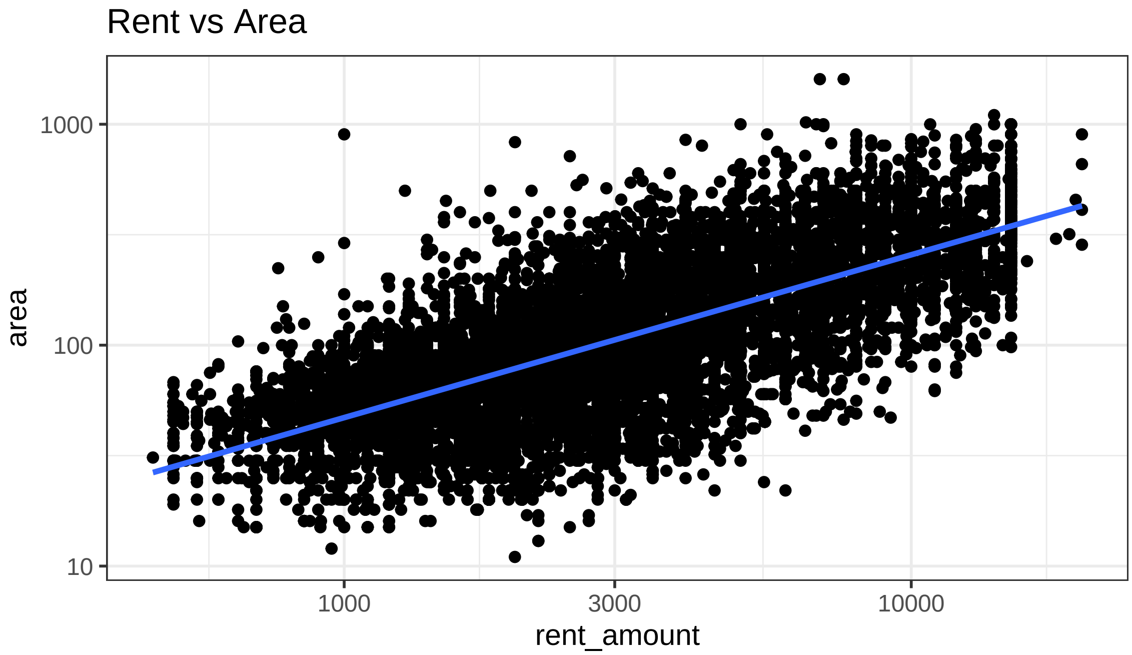
We see a small house in disagreement with the rest of the “select” group. Let us take rent values higher than 20k as outliers.

Now let us see their distributions:

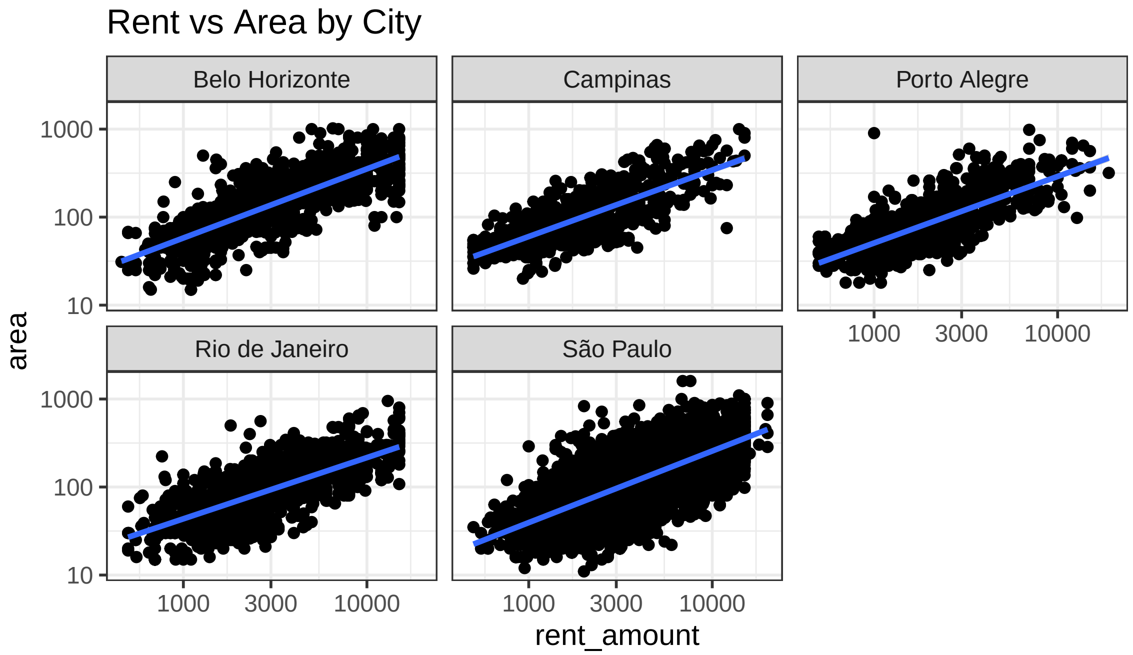




Assuming that the properties have the same pattern, one could expected that the larger the property, higher the rent would be, once a common unit of measure is value per square meter. On the graphic below we can see if there is a positive correlation between them.

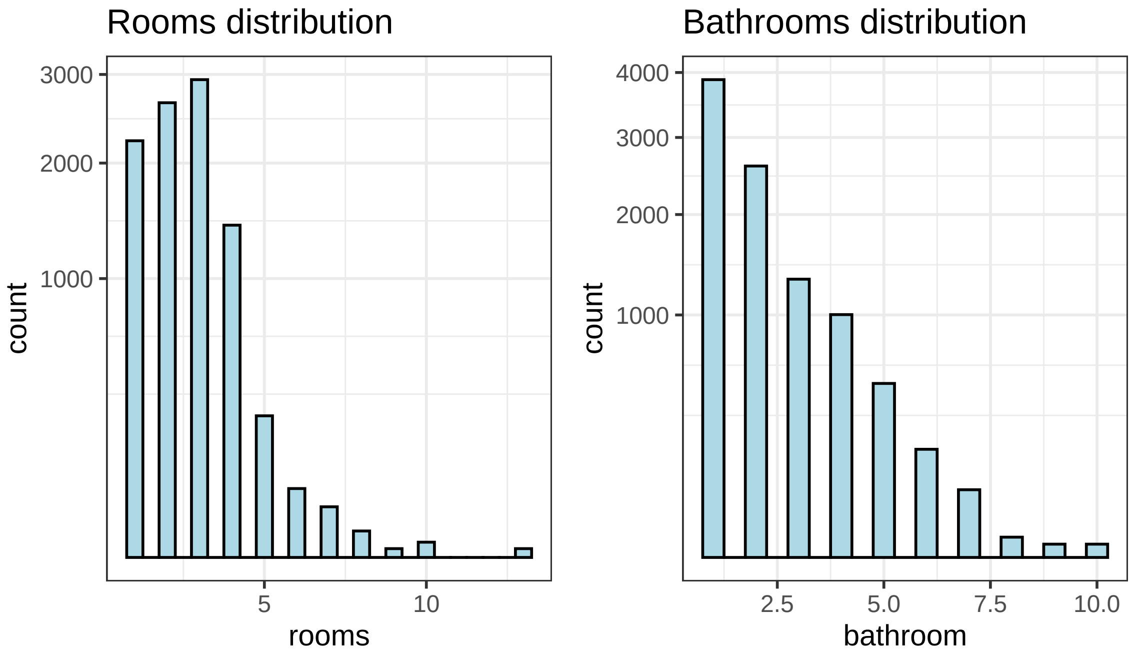


We can see below that all cities show a strong correlation, larger than 0.6.

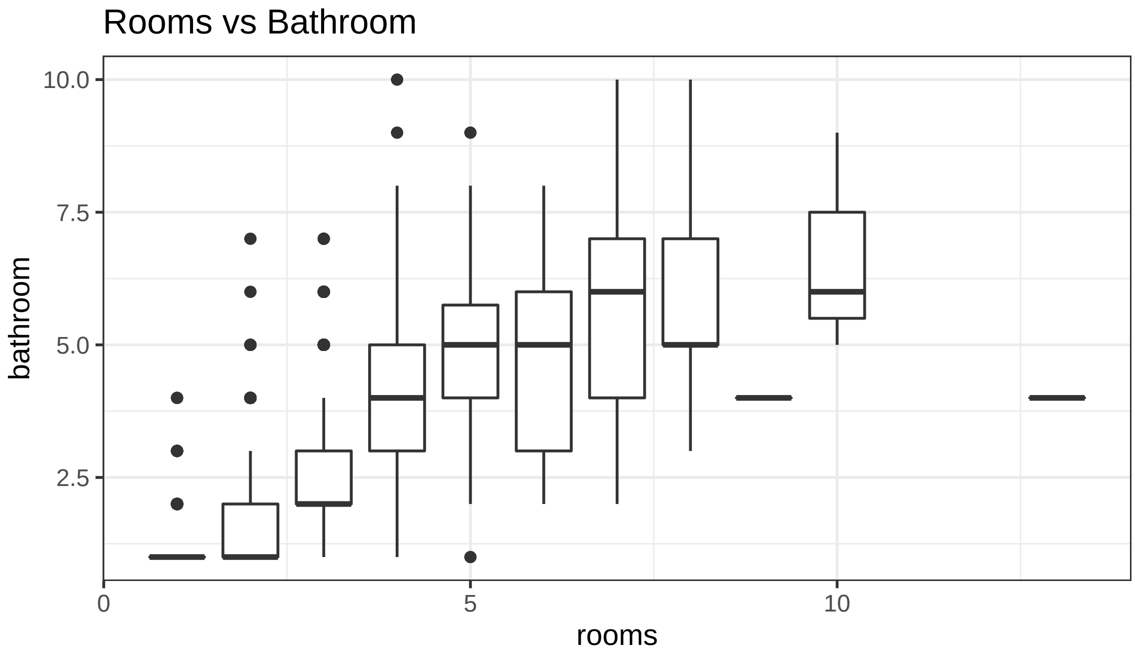


- Rooms and Bathrooms

We have seen at the first summary that the minimum and maximum number of rooms (between 1 and 13) and bathrooms (between 1 and 10) are quite similar. Let us check how they are distributed and correlated.



Only about 2% of houses has more than 5 rooms or bathrooms. In a boxplot we can see the range of bathrooms for each quantity of room.

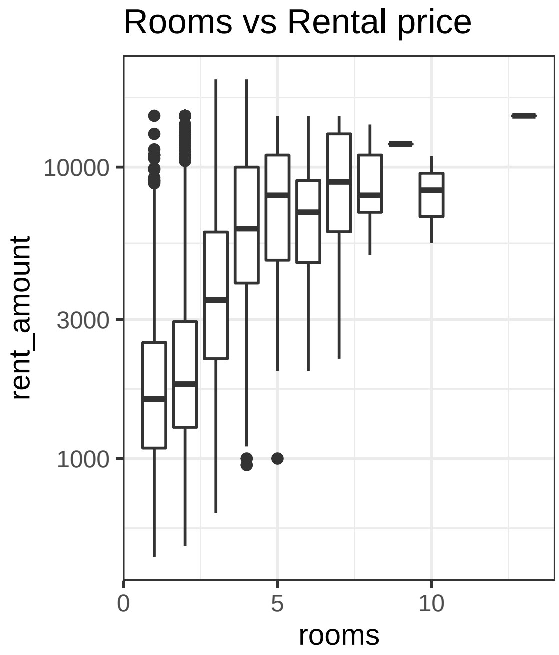


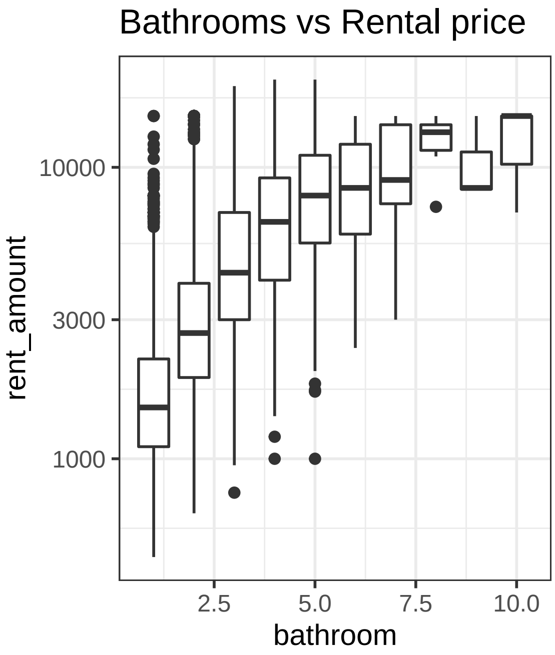
We can think that a low standard apartment/house may have fewer bathrooms than rooms, in spite of high standard properties, in addition to having one bathroom per room, there may be a toilet, a maid's bathroom, a leisure area bathroom, among others.

Looking closer at some points in the graphic we can find other outliers. Small houses (area) with a big quantity of bathroom or rooms, making no sense. Let us remove them.

city	area	rooms	bathroom	parking_spaces	floor	rent_amount
São Paulo	35	5	1	0	0	2500
São Paulo	43	2	7	2	11	2270

Now let us see the relation of room and bathroom with rental price.

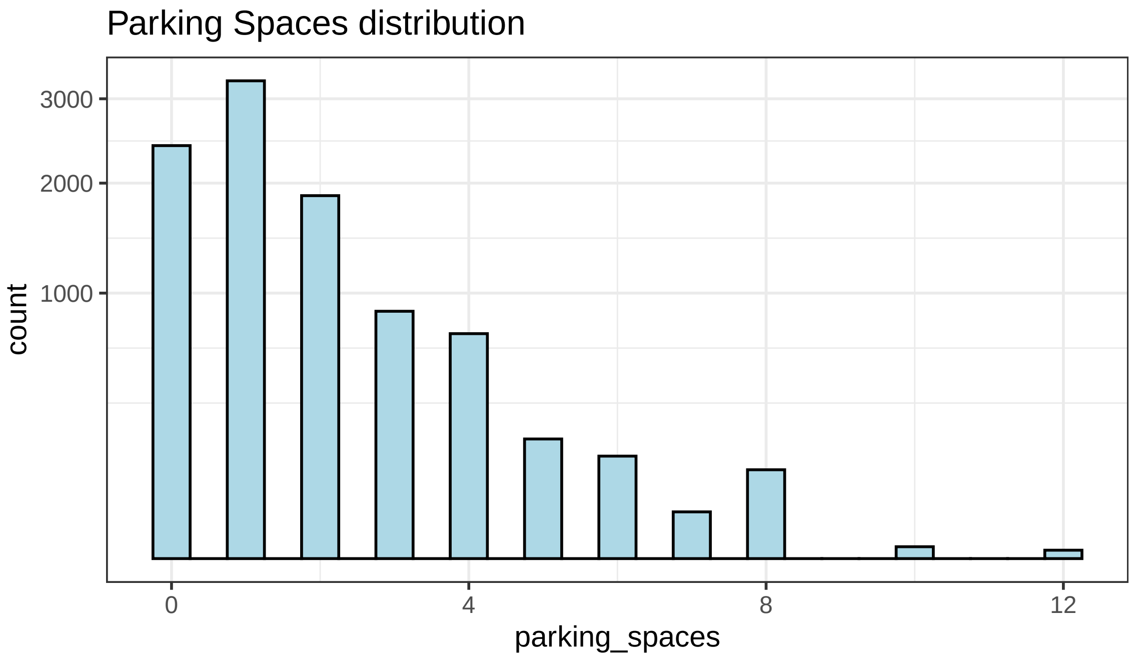




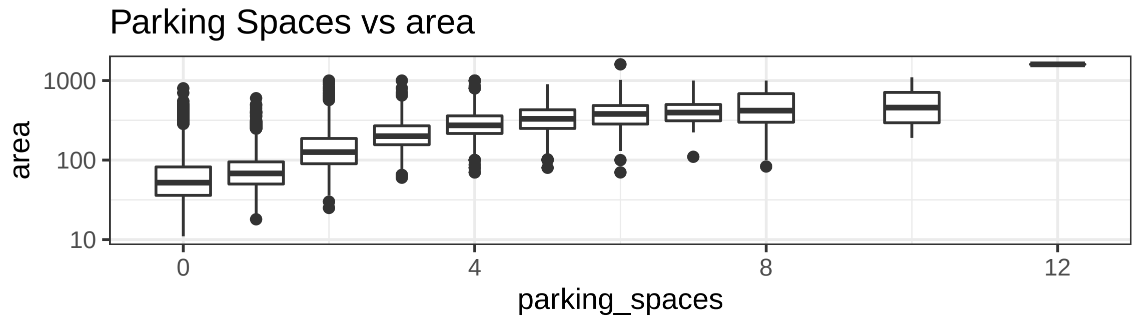
By these plots we confirm that rental price is highly related to the number of rooms and mainly to the number of bathrooms.

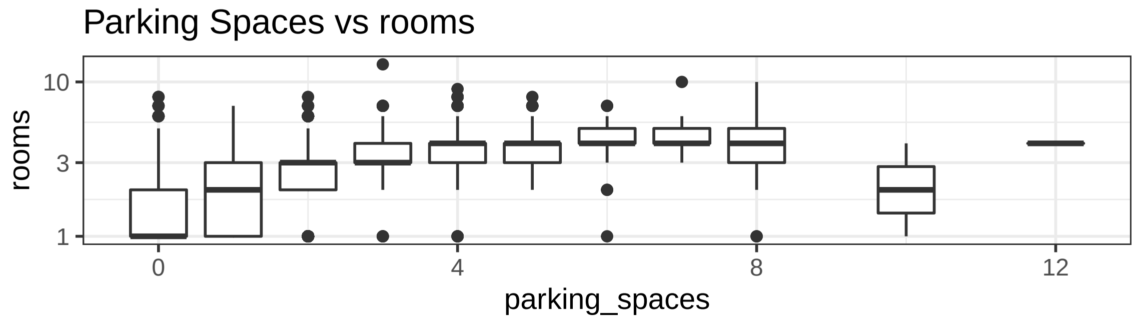
- Parking Spaces

We believe they are quite related to area and rental price as well, since highest pattern houses may have more parking spaces.



By the plot we can see most properties has less than 4 parking spaces. Now let us have a look at boxplot to see the range of area and rooms to each quantity of parking space.

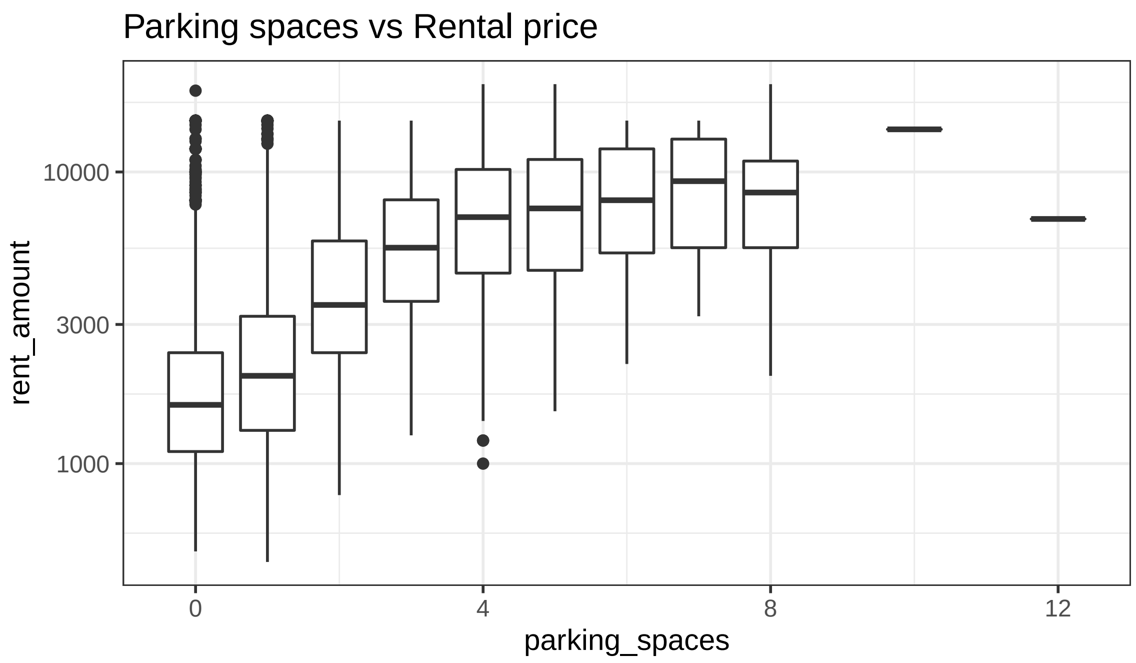




Looking closer to some outliers we can confirm they should be removed. There are 8 houses with only one room and 4 or more parking spaces. Also, there are 3 other houses with less than 100m2, 2 rooms and more than 4 parking spaces. Let us remove these outliers as well.

city	area	rooms	bathroom	hoa	parking_spaces	rent_amount
São Paulo	190	1	2	0	10	3900
Campinas	83	2	1	0	8	4550
São Paulo	330	1	3	0	8	3147
Belo Horizonte	360	1	1	0	8	2190
São Paulo	70	1	2	0	6	3290
São Paulo	80	2	2	0	5	3000
São Paulo	70	2	2	0	4	3000
São Paulo	440	1	2	1	4	9800
São Paulo	70	1	1	0	4	1300
São Paulo	148	1	2	1845	4	9900
São Paulo	80	1	1	0	4	1000

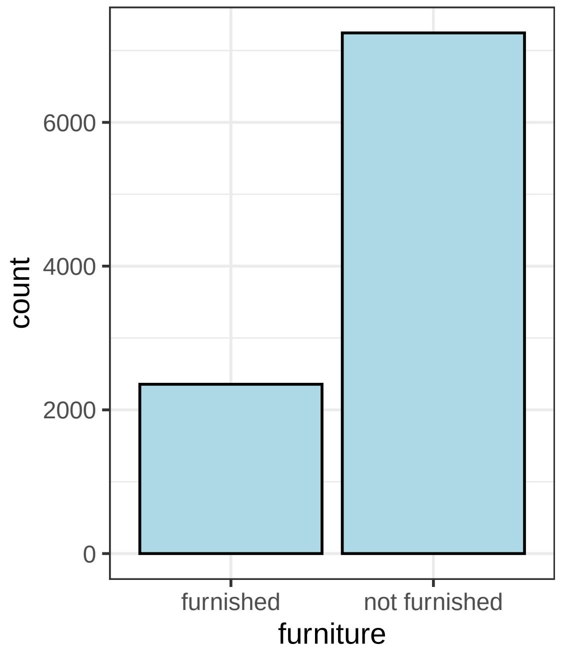
On the graphic below we can see how parking spaces are related to rental price.

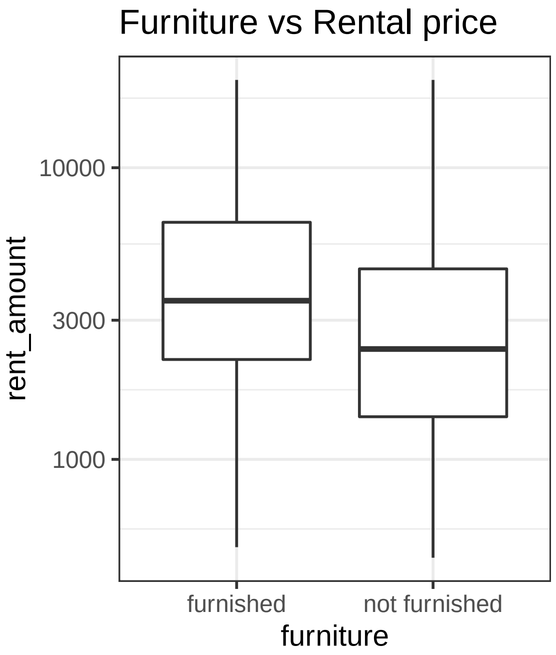


We see an increasing price variation with the increase in the number of parking until reaching 4. The same behave is found even when detailed by city.

- Furniture

We can think that prices of furnished houses are more expensive due to the investment made. But let us see what data shows. Furnished properties represent about 25% of our dataset.

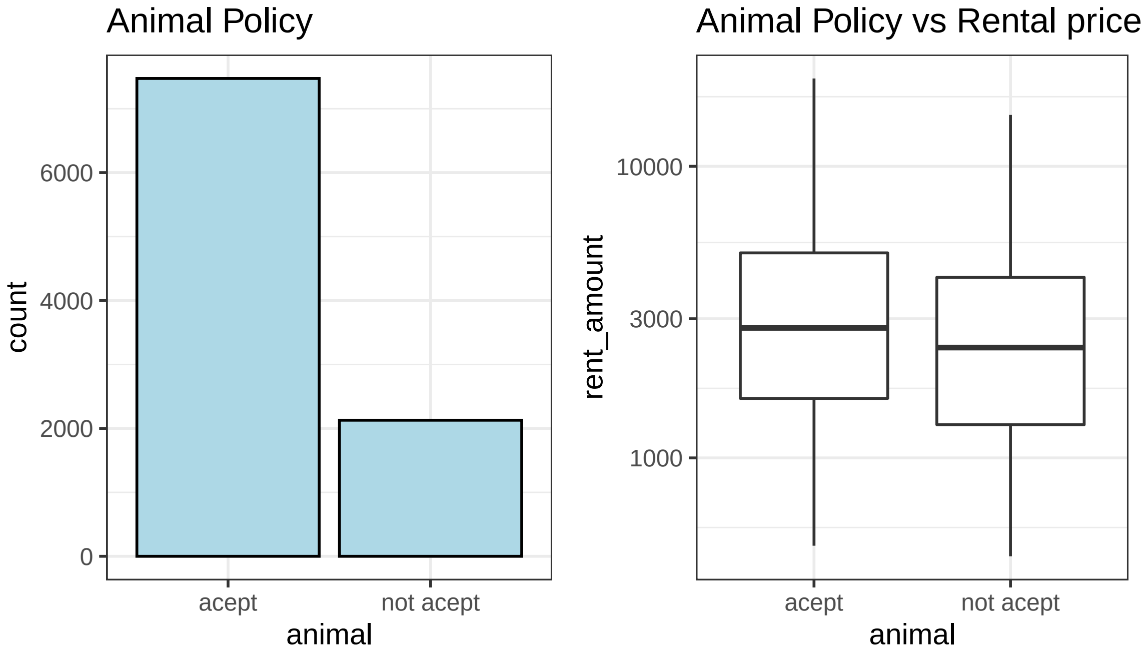




Despite knowing that the rental price varies based on many other variables, we can confirm that the furnished properties have, on average, a higher price, although the range for furnished and unfurnished is practically the same.

- Animal

Although it is a very relevant factor in the decision to choose a home for those who have a pet, we do not expect the price to be impacted. Let us see the plots.



Most properties accept animals and by the chart we can confirm that there is not such variation in relation to the rental price.

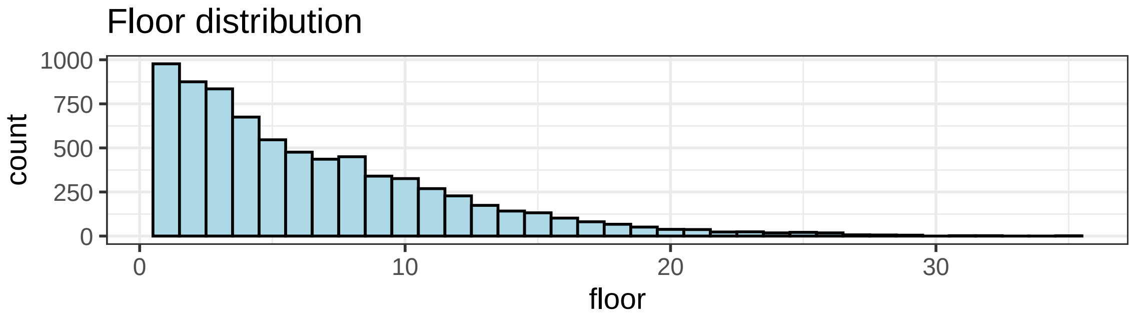
- Floor

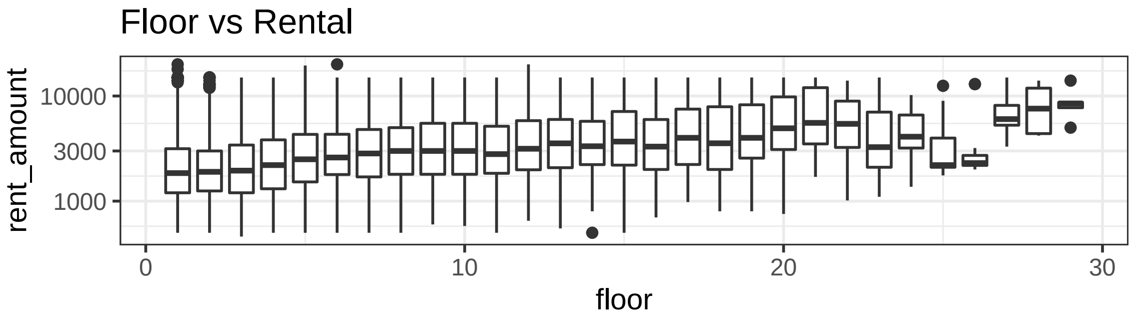
First let us see the highest floor available.

city	area	floor
Belo Horizonte	80	301
Campinas	64	51
São Paulo	250	35
São Paulo	84	32
São Paulo	51	32
São Paulo	77	29

It is unlikely that this information is true once the highest buildind in brazil has 46 floors. Some taller buildings are expected to be completed in 2020. So for now we will change floors higher than 46 (301 and 51) to 31.

Let us see the distribution. Once we have changed floor with“-” to “zero”, assuming they are houses or that there is a lack of information, let us filter the floor > 0 and see the distribution and relation to price.





As samples of floor taller than 30 is not representative, we will take them off from the second part of plot. We can observe rental prices are quite similar in every range of floor.

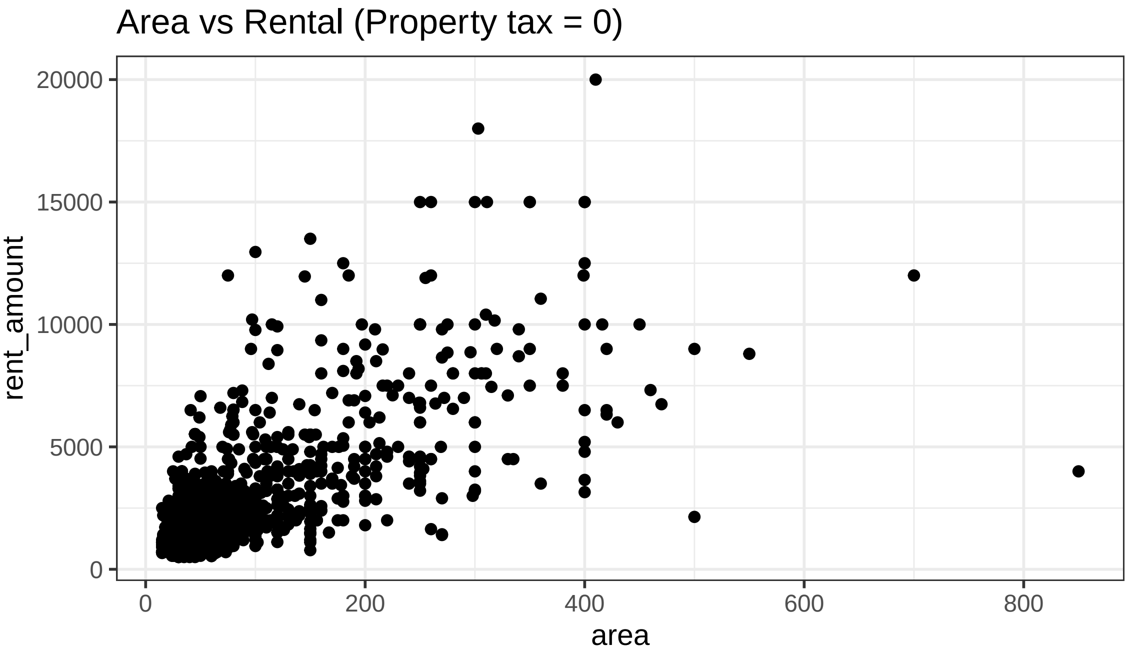
- Property tax

Let us have a look at the highest values:

city	area	property_tax	hoa	rent_amount
São Paulo	42	313700	690	2500
Rio de Janeiro	95	28120	1024	3000
São Paulo	700	21880	0	10000
Belo Horizonte	260	12500	3200	11900
Rio de Janeiro	35	9900	81150	4500
São Paulo	890	9500	0	11000
São Paulo	700	8750	0	15000
São Paulo	884	5917	9000	12750

We can see that the first five values seem off putting, since that or the area is small in comparison to the thers, or the property tax is much higher than other properties with similar area and rent value of the same city. So, we will take these 5 properties off.

And now let us analyze the lowest values. There are 1.448 properties where the tax is equal to zero.



We can see that most of those houses which do not have property taxes are located in the lower left corner of the graph, which means they are buildings with a smaller area and lower value of rent.

Let us understand a little more about this tax.

In Brazil this tax is called IPTU and each city has a specific law that defines the forms of calculations and exemption cases. In general, we see that the amount due is related to the venal value of property which is determined by the venal value of the land, for territorial properties and, by adding the venal values of the land and construction, for building properties. The venal value of the land is obtained by multiplying the lot's area by the value of the land's m^2; correction factors may be applied to this value.

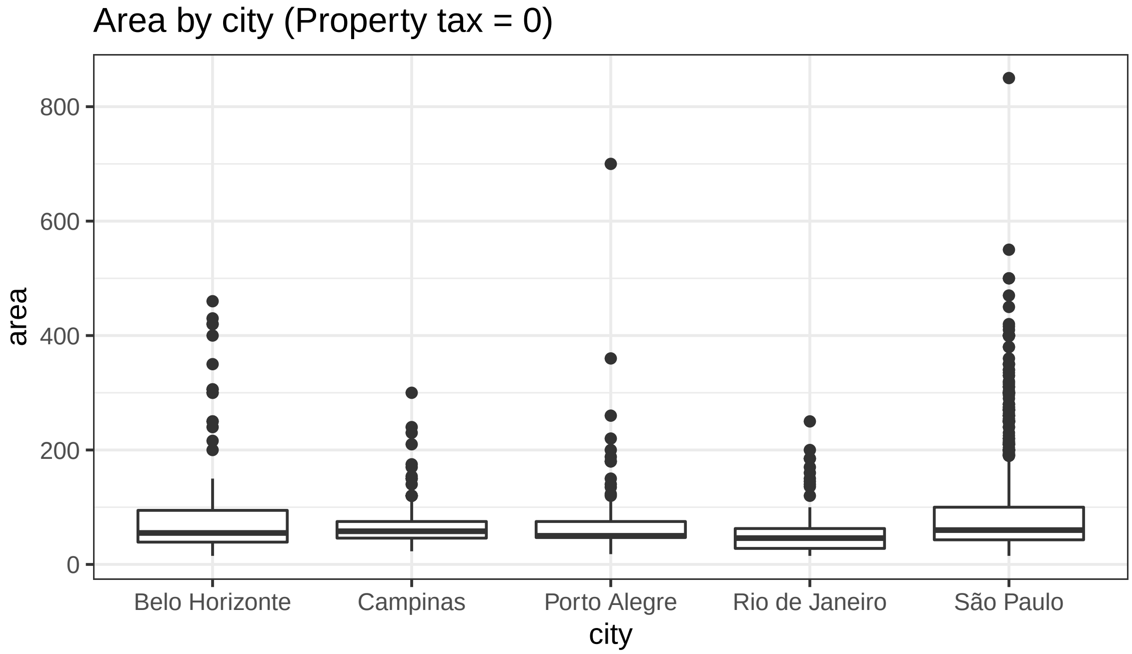
Doing a quick search on the laws of these 5 cities, we found:

- Rio de janeiro - With the validity of Law 6,250 / 2017, residential units with a venal value of up to R $ 58,802.00 are exempt from IPTU.
- São Paulo - Exemptions and discounts for the property's market value are automatically applied the value of the property is up to R $ 160 thousand. - Retirees and pensioners, cultural entities, sports associations and Societies Friends of Neighborhoods, among others, may apply for exemption from IPTU if they prove the requirements determined by law. In this case the property's value of up to R $ 1,310,575.00. - The tendency is for properties with a smaller built area to be exempt, but the exemption is not related to the construction footage, but to the property's market value (which depends on several factors, in addition to the built area).
- Porto Alegre - The exemption is integral for IPTU for properties with a venal value of up to 100,000 (one hundred thousand). - Retired, inactive, pensioners and people with disabilities are exempt from paying the Porto Alegre Property and Land Tax (IPTU).
- Belo Horizonte - Properties with a value of up to R $ 66,601.98 are exempt from tax.
- Campinas - Properties for popular housing registered in the Horizontal Residential (RH) category with a total built area not exceeding 80.00m^2 (eighty square meters) or in the Vertical Residential (RV) category with a total built area not exceeding 50.00m^2 (fifty square meters), without territorial area surplus, the venal value of which does not exceed 60,000. - Retirees, pensioners and beneficiaries of

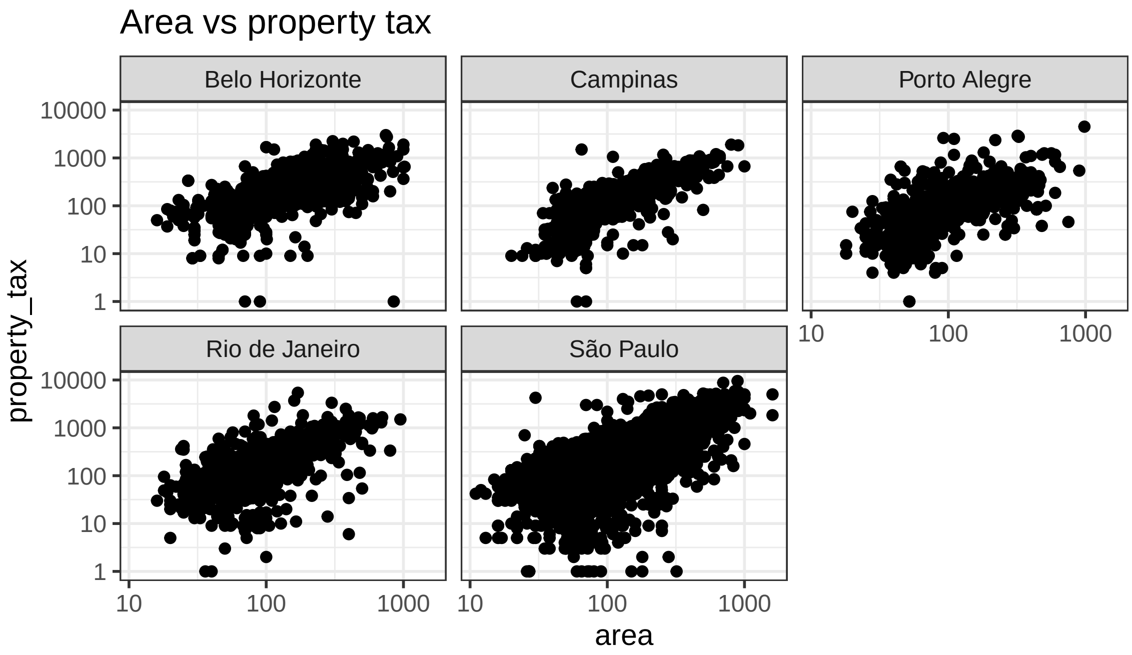
> Social Protection for the Elderly, Social Protection for Persons with Disabilities and Lifetime Monthly Income, in relation to the property belonging to their assets classified in the strictly residential category and where they actually reside, conditioning the person legally benefited to attend some rules.

So in summary, we can see that there is a convergence in exemption from fees for lower value properties, sometimes considering the area as well, however there are some other cases to exempt, like for elderly people that meet certain income criteria and for philanthropic and government support entities. Which means our data shown in the last graph does make sense.

Another way to confirm that is looking to graph below, where we can see most of properties that have property tax = 0 has area less than 150m2 in all cities.



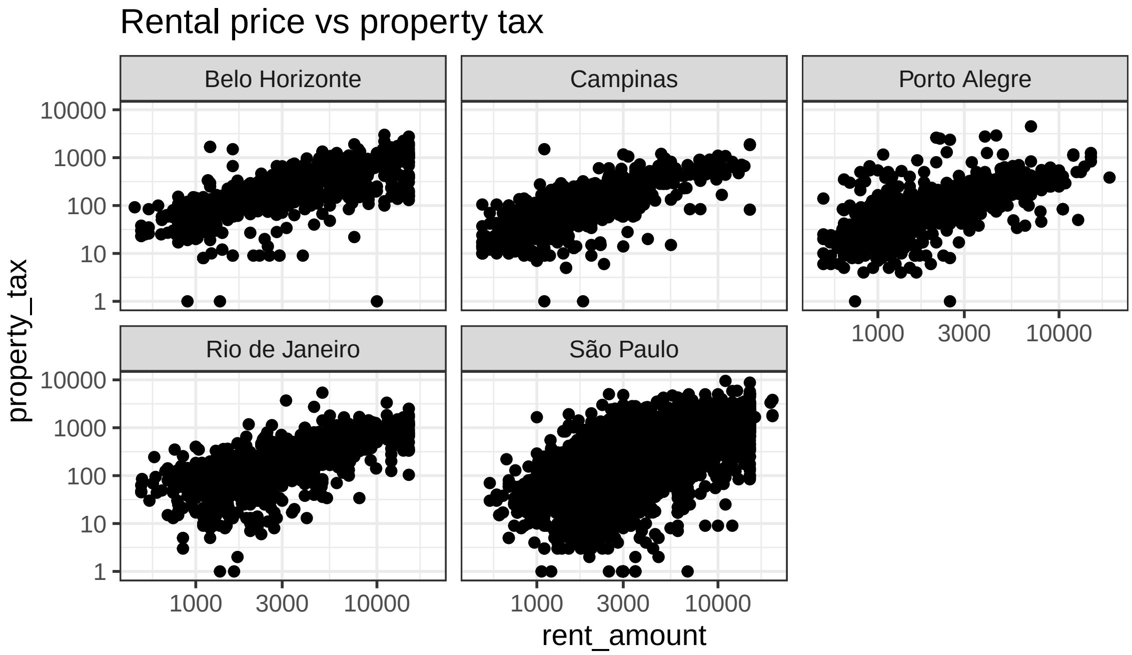
Now let us see what happens to properties where the property tax is due.



The basic and most used rule to calculate rent is to apply a value between 0.5% and 1%, per month, on the market value of the property. This means that if your property is worth 100 thousand the rent should be between R 500 and R $ 1,000 per month.

Therefore, as there is no feature with the market value of the property, we can infer that the market value is related to the venal value, which in turn impacts the value of the tax. That is, the higher the amount of tax to be paid, the more valuable the property is and consequently its rental price.

Let us see if data confirms it looking at how they behave graphically.



- Hoa - homeowner's association

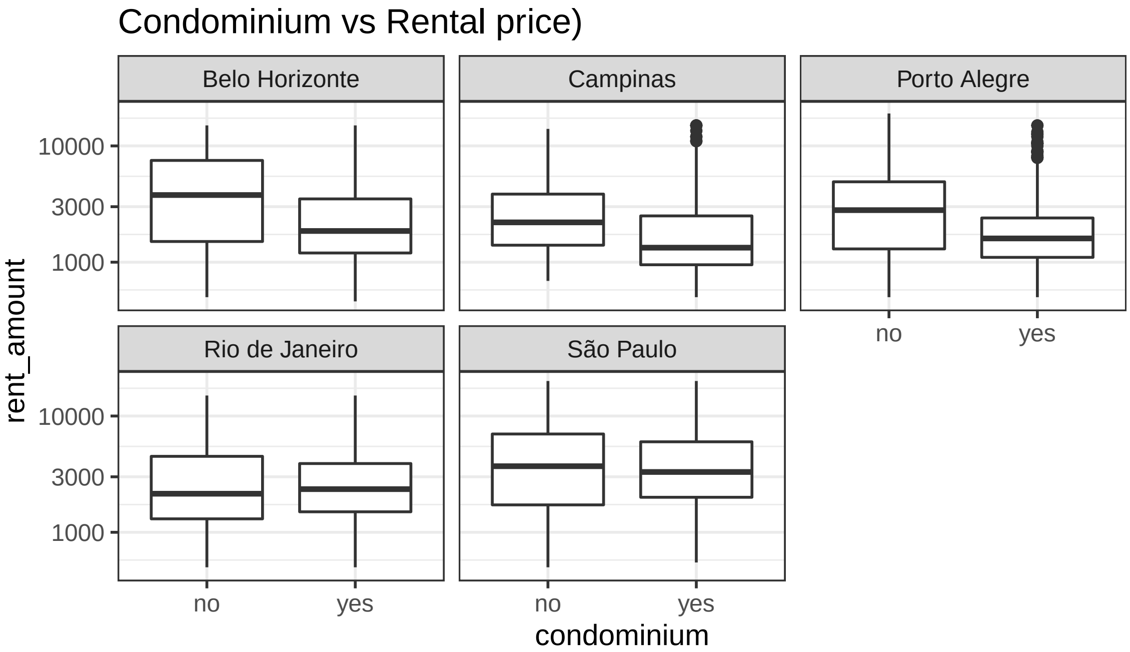
As the previous features let us see the lowest and highest values.

There are 2.138 properties with hoa= 0, what can be used to define if a property is an independent house or whether it is an apartment or a condominium house, once we do not have the field about the kind of home is about.

Looking at the highest values we see that it is necessary to take the first 3 values off, once they do not make any sense, they are too high to be paid monthly if compared with area, property tax or rental price.

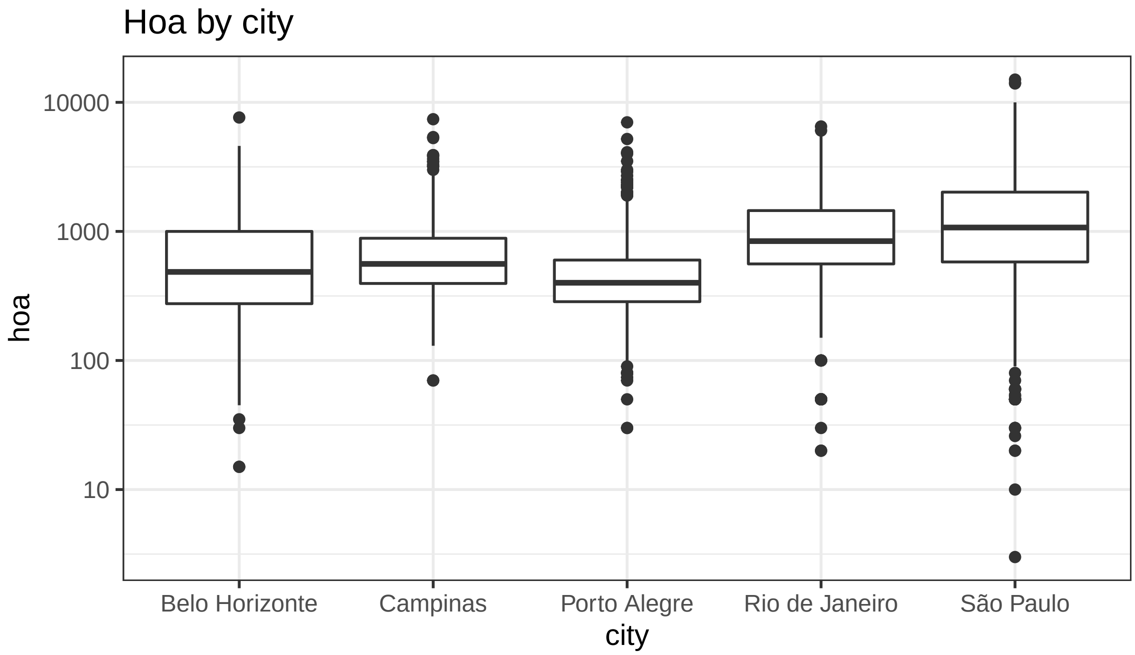
city	area	property_tax	hoa	rent_amount
Belo Horizonte	155	64	1117000	2790
São Paulo	285	1834	200000	20000
Porto Alegre	42	40	32000	700
São Paulo	850	2465	15000	13000
São Paulo	488	1214	14130	6400
São Paulo	850	0	14000	4000
São Paulo	600	84	10000	15000
São Paulo	800	209	10000	8500

We will create a new feature “condominium” to classify properties in two classes, those who do pay the fee and those who do not.

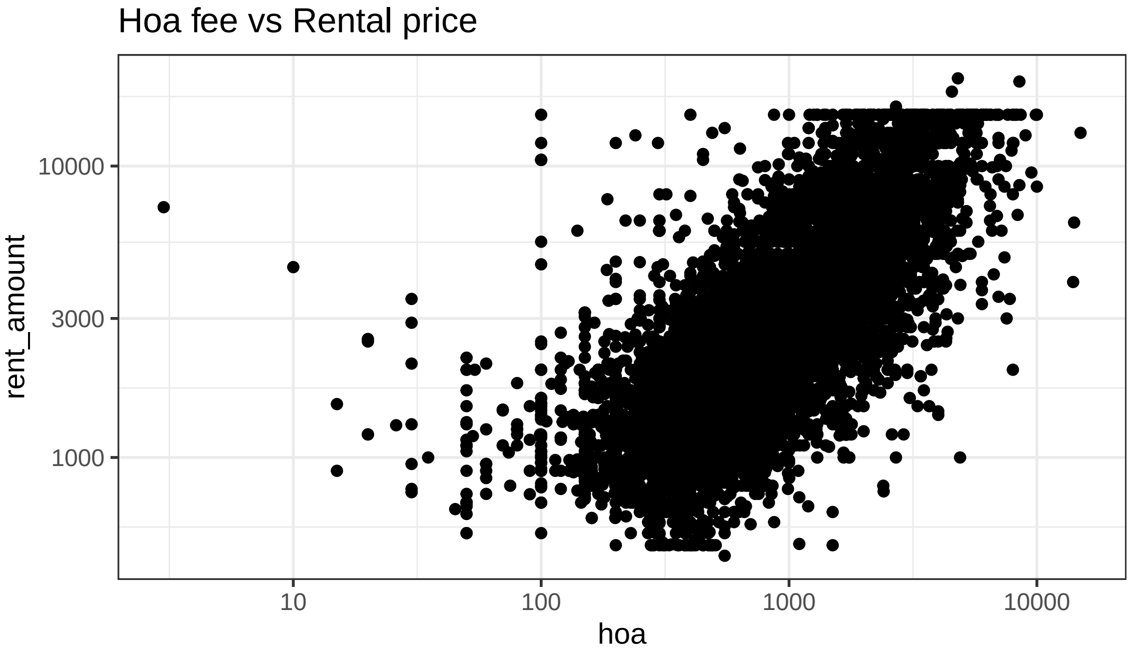


Properties that do not have a condominium fee have higher median rental values in Belo Horizonte, Campinas and Porto Alegre. This is justified because for the tenant what really matters is the total monthly amount to be paid, which includes the condominium fee.

For those who have hoa fee, we can see that São Paulo and Rio de Janeiro have highest values of hoa and highest standard deviation.

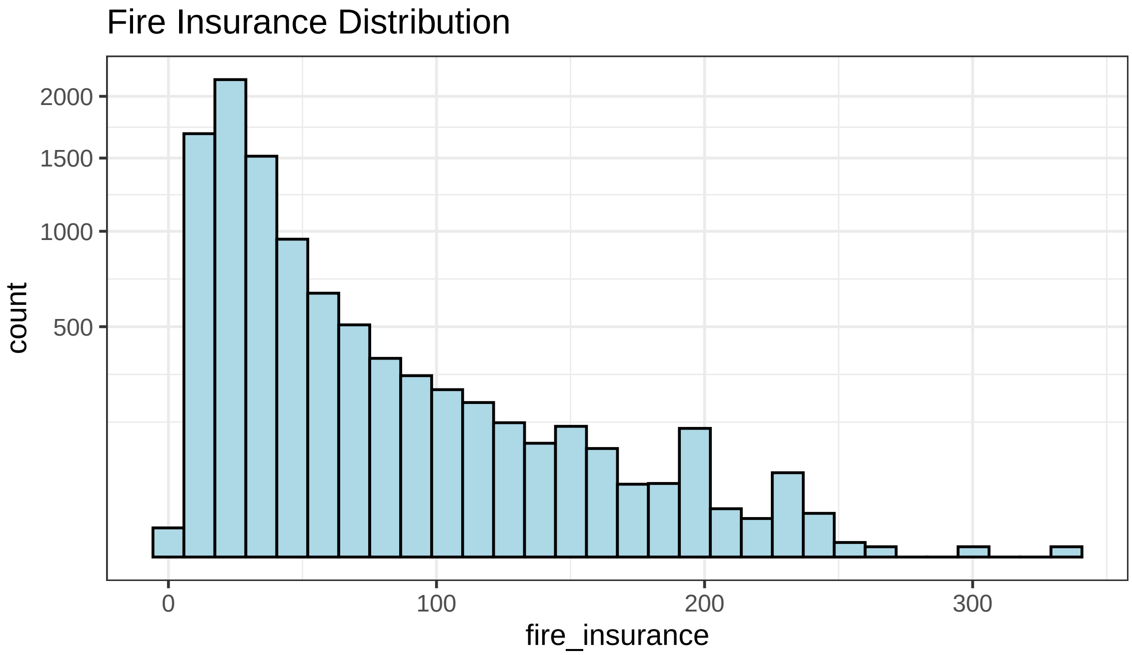


Now let us see the relation between hoa and rental price.

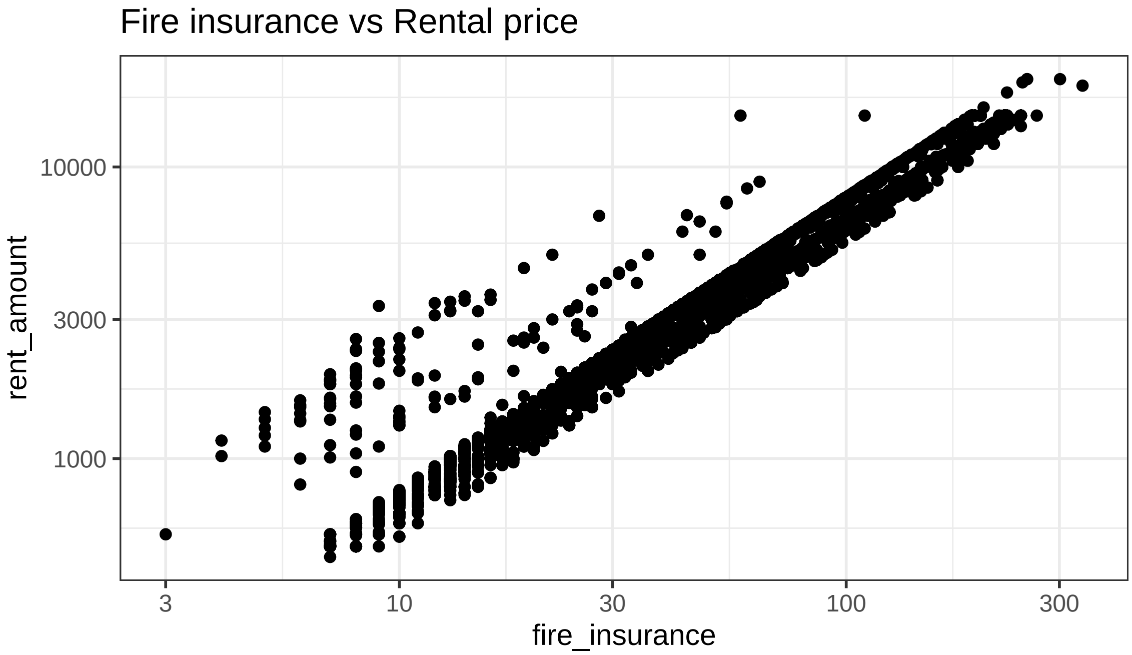


- Fire Insurance

In relation to fire insurance we can observe there is no such a variation like the previous taxes. For most of properties (98,7%) this fee is less or equal to R$200 which means its value is not too related to area or value of property.



We can see also, the ones which have the highest values are the ones that have the highest rent amounts in a perfect correlation. Which implies that this feature is calculated based on the rent amount, therefore is not an independent feature.



- Total

Considering that this feature is also a dependable variable, equal to the sum of rental price and all previous presented tax and fees, we are going to remove it, leaving the following features remaining:

```
build<- select(build,- c(total, fire_insurance))
names(build)
```

```
##  [1] "city"           "area"           "rooms"          "bathroom"
##  [5] "parking_spaces" "floor"          "animal"         "furniture"
##  [9] "hoa"            "rent_amount"    "property_tax"
```

Now that we have seen all features we can start to build our model.

Modelling

We are going to use different regression algorithms to see which one performs better. Models will be evaluated and compared by using the function "postResample" that estimates the root mean squared error (RMSE), simple R2, and the mean absolute error (MAE). However we will choose the model of smaller RMSE.

Some definitions: - RMSE - Residual Mean Squared Error, the square root of the average of squared errors. - Rsquared - The R^2 indicates how much that model explain the observations well. - MAE - The mean absolute error is calculated using mean(abs(pred-obs)). We can use the model to predict a "fair" rent price for a property and use the MAE as a tolerance. This would help a buyer to find good opportunities and run away from bad ones.

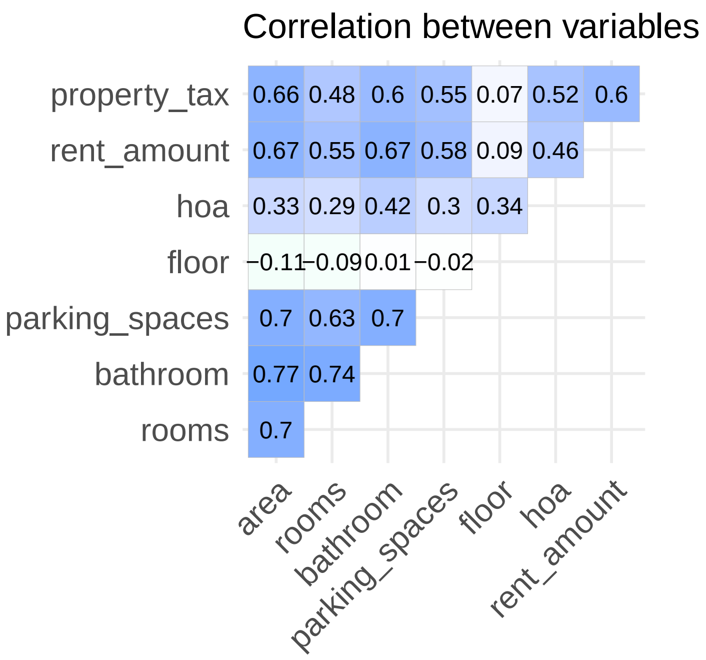
Pre processing

First we will see if there is some predictor zero-variance. Predictors when the data are split into cross-validation/bootstrap sub-samples or that a few samples may have an undue influence on the model. These "near-zero-variance" predictors may need to be identified and eliminated prior to modeling.

```
nzv<- nearZeroVar(build)
nzv
```

```
## integer(0)
```

Correlation Most models may benefit from reducing the level of correlation between the predictors. Let us take a look into the correlation between all numeric features.



We see the highest correlation, higher than 0.75 correlation between area and bathrooms. We can see below the correlation summary.

```
vars_cont <- build %>% select_if(is.numeric)
buildCor <- cor(vars_cont)
summary(buildCor[upper.tri(buildCor)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -0.106   0.298   0.534   0.436   0.664   0.768
```

The code below shows the highest correlation after removing predictors with absolute correlations above 0.75 and the remaining predictor names.

```
highlyCor <- findCorrelation(buildCor, cutoff = .75)
vars_cont <- vars_cont[,-highlyCor]
names(vars_cont)
```

```
## [1] "rooms"          "bathroom"       "parking_spaces" "floor"
## [5] "hoa"            "rent_amount"    "property_tax"
```

```
buildCor2 <- cor(vars_cont)
summary(buildCor2[upper.tri(buildCor2)])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.0892  0.2903  0.4756  0.4037  0.5998  0.7355
```

So we are going to take the predictor “area” off.

We do not find any linear dependency among features

```
comboInfo <- findLinearCombos(vars_cont)
comboInfo
```

```
## $linearCombos
## list()
##
## $remove
## NULL
```

Creating train and test set

After removing outliers and the high correlated features, 9.593 houses remaining at our build dataset with 9 features and 1 outcome.

```
## [1] 9593   10
```

Now we are going to split the build dataset in two parts in a ratio 90/10 called "train" and "test" datasets.

```
#Create train and test set
set.seed(27, sample.kind = "Rounding")
test_index <- createDataPartition(y =build$rent_amount, times = 1, p = 0.1, list = FALSE)
train<- build[-test_index,]
test <- build[test_index,]
```

Before training models with machine learning algorithms we will set seed = 35 so that they all use the same sample and results can be compared.

To reduce the risk of overtraining and randomness we are going to use Cross validation technique, which have as general idea to randomly generate smaller datasets that are not used for training, and instead used to estimate the true error.

```
control <- trainControl(method = "cv",
                              number = 10, p=0.9)
```

Machine Learning Algorithms

1- Linear Regression Sometimes it can be too rigid to be useful and but for some challenges it works rather well. We will define its results as our baseline to compare to other models.

```
ini_time<- Sys.time()
set.seed(35, sample.kind = "Rounding")
train_lm<- train(rent_amount~., method= "lm", data = train, preProcess= "scale",trControl = control)
end_time<- Sys.time()
train_lm
```

```
## Linear Regression
##
## 8632 samples
##    9 predictor
##
```

```
## Pre-processing: scaled (12)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 7769, 7768, 7768, 7768, 7768, 7771, ...
## Resampling results:
##
##   RMSE  Rsquared  MAE
##   2208  0.5667    1447
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
print(end_time-ini_time)
```

```
## Time difference of 1.214 secs
```

Using the model to predict on the test set we obtain a RMSE = 2079.

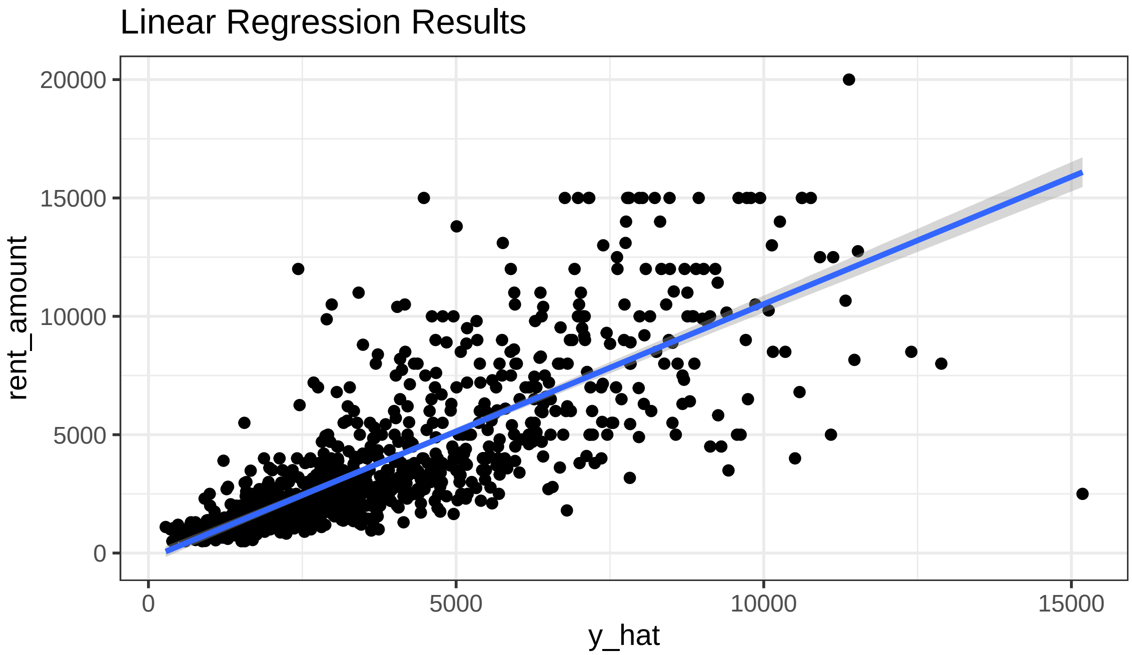
```
y_hat_lm<- predict(train_lm, test)
lm_results<- postResample(y_hat_lm, test$rent_amount)
lm_results
```

```
##      RMSE  Rsquared       MAE
## 2079.0791    0.6134 1373.2298
```

Predictions could be easily calculated by a linear function with the following coefficients:

	x
(Intercept)	702.87
cityCampinas	-136.04
cityPorto Alegre	-67.30
cityRio de Janeiro	160.89
citySão Paulo	284.50
rooms	397.21
bathroom	1042.01
parking_spaces	453.53
floor	48.83
animalnot acept	26.04
furniturenot furnished	-472.13
hoa	354.84
property_tax	656.73

And now we can see what were our biggest mistakes, comparing prediction to actual data.



We can see that a few outliers are responsible for much of our error, impacting the RMSE. Let's have a look in the biggest mistakes.

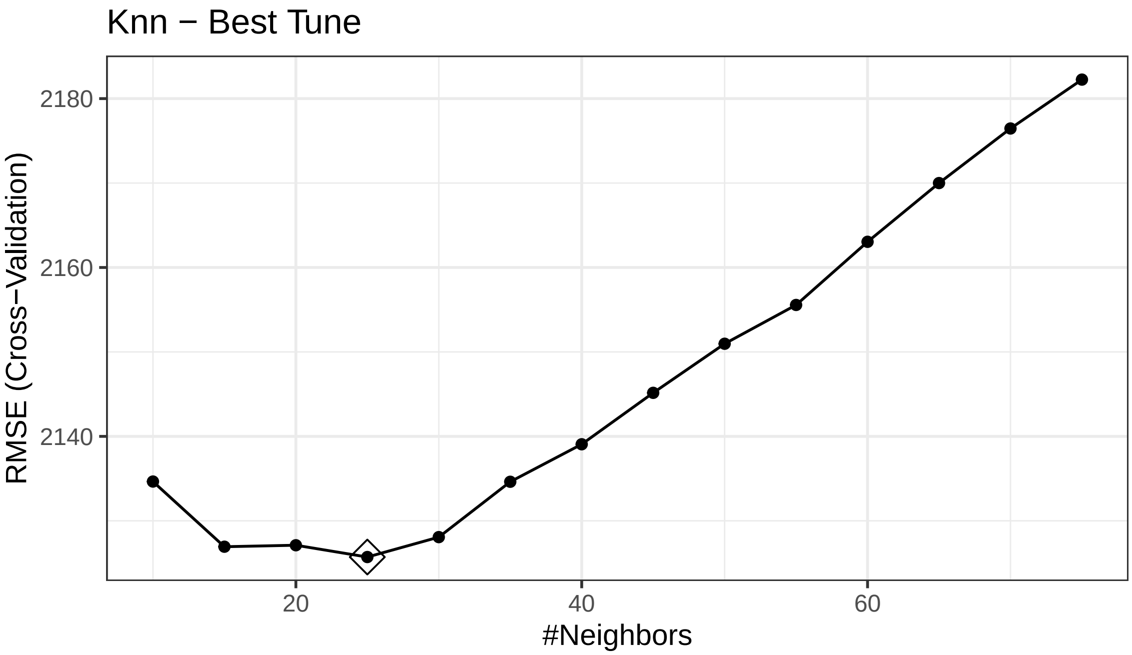
city	bathroom	rooms	property_tax	rent_amount	y_hat	error
São Paulo	5	5	5032	2500	15181	12681
São Paulo	2	1	130	15000	4476	10524
Belo Horizonte	2	3	236	12000	2434	9566
São Paulo	3	3	375	13800	5009	8791
São Paulo	5	4	1750	20000	11385	8615
Belo Horizonte	5	4	153	15000	6767	8233

Let us try to improve using others models.

2- K-Nearest Neighbors *method = "knn"*

We can control the flexibility of our Knn estimate through the k parameter: larger ks result in smoother estimates, while smaller ks result in more flexible and more wiggly estimates.

We pick k = 25 as shown on the plot below.



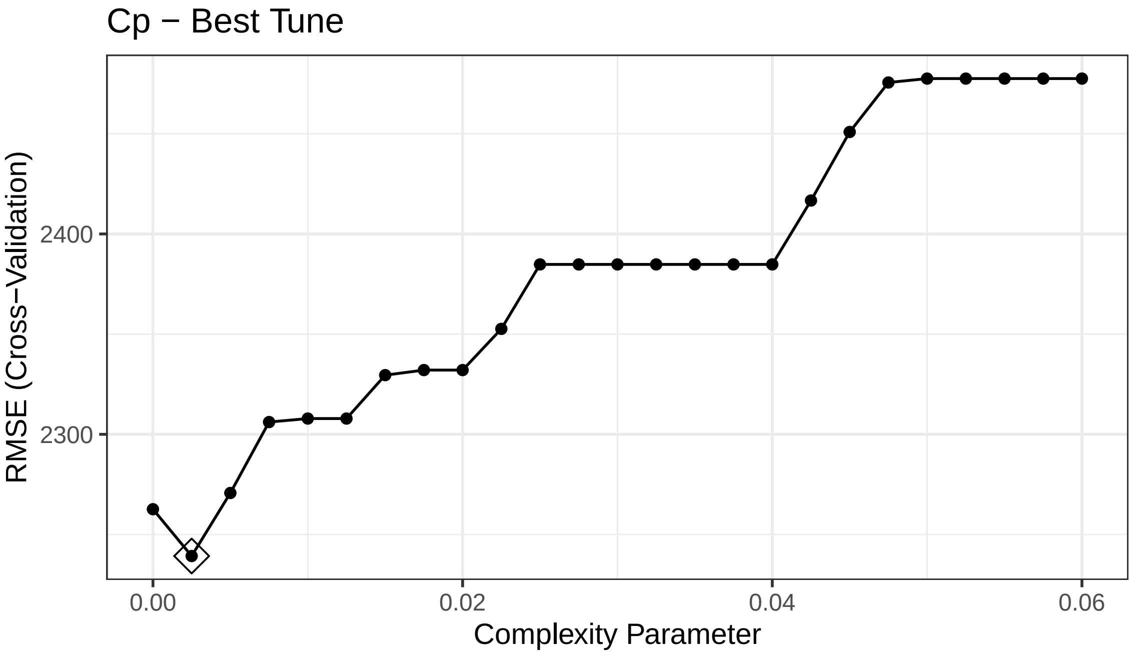
Now we can predict on the test dataset to see the results.

```
##      RMSE  Rsquared       MAE
## 1982.1343    0.6581 1233.5457
```

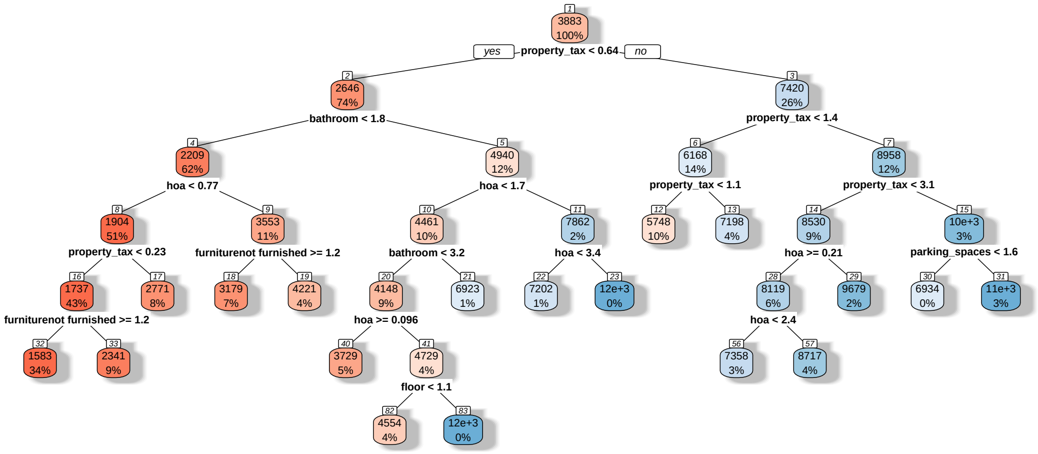
There was an improvement about 5% if compared to linear regression. Let's see if we can do better.

3- Regression tree *method = "rpart"*

On this model we have the tuning parameter: cp (Complexity Parameter).



Cp was defined = 0.0025 but RMSE resulted in 2239, higher than previous models. Let us have a look on the final model of Regression Tree.

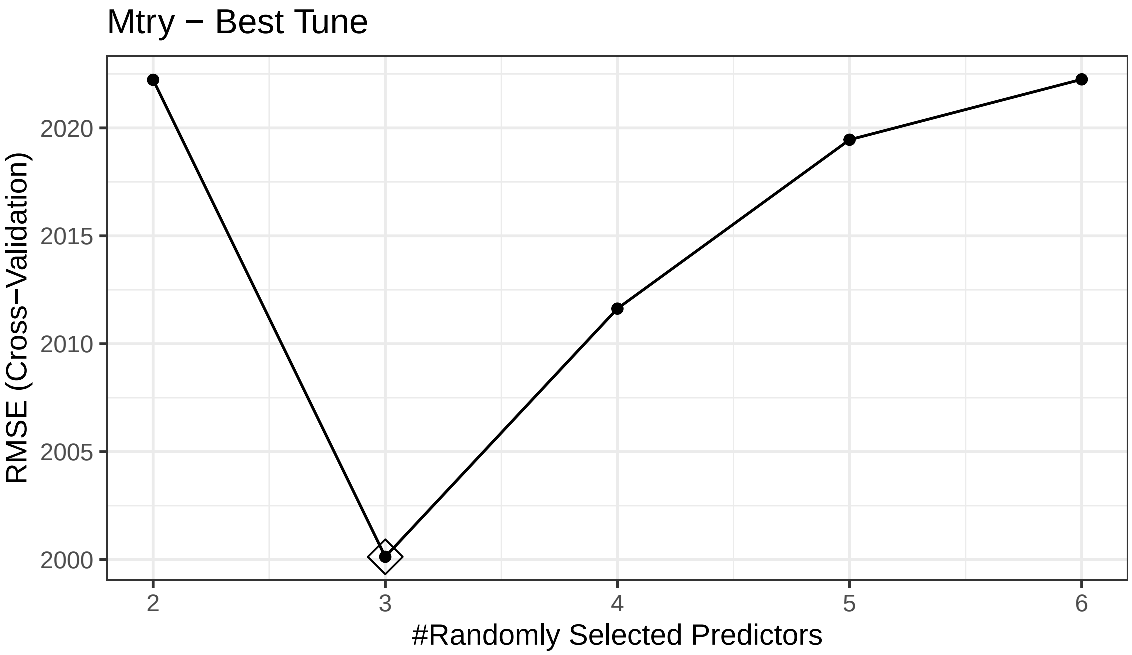


Although results on training was worse let us save the RMSE on the test set.

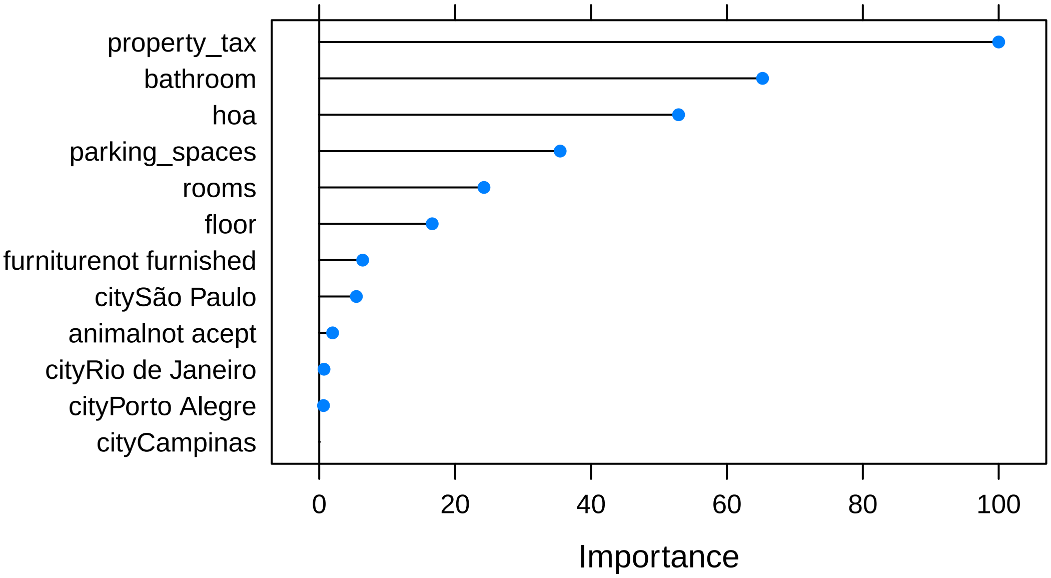
```
##      RMSE  Rsquared       MAE
## 2100.6106    0.6026 1379.0609
```

4- Random Forest   *method = "rf"*

We have tested some values for ntree and mtry. The best results were obtained with ntree=100, mtry=3 and without pre-processing features.



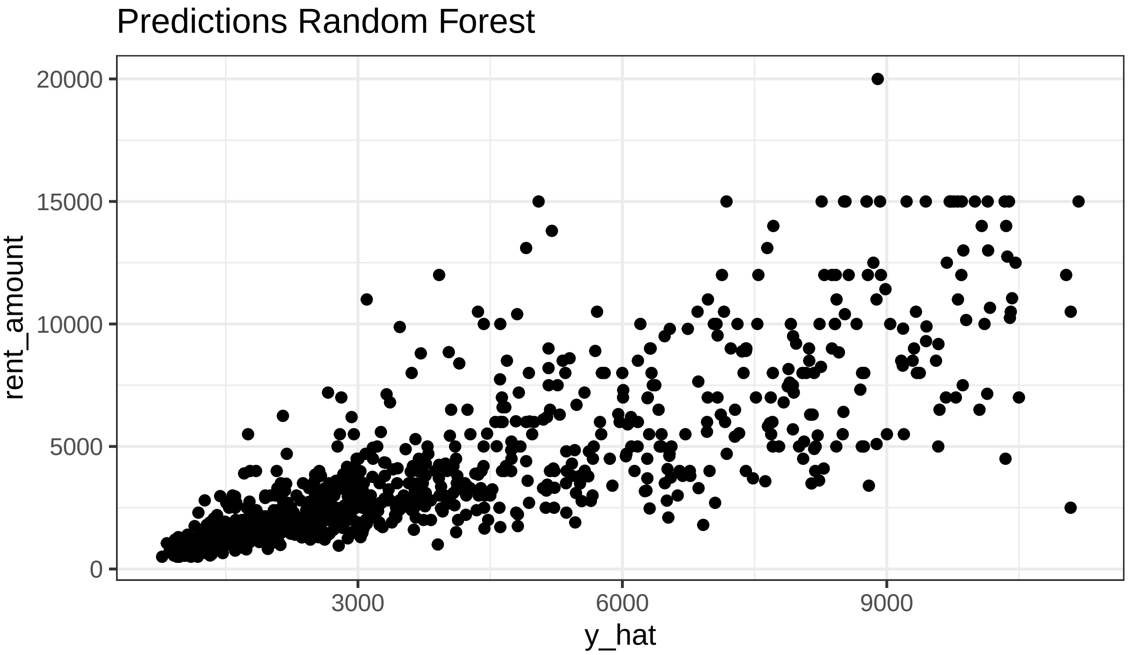
One disadvantage of random forests is that we lose interpretability. An approach that helps with interpretability is to examine variable importance. To define variable importance we count how often a predictor is used in the individual trees.



So far Random Forest presents the best results. Let us calculate RMSE on the test set.

```
##      RMSE  Rsquared       MAE
## 1902.2055    0.6756 1204.2884
```

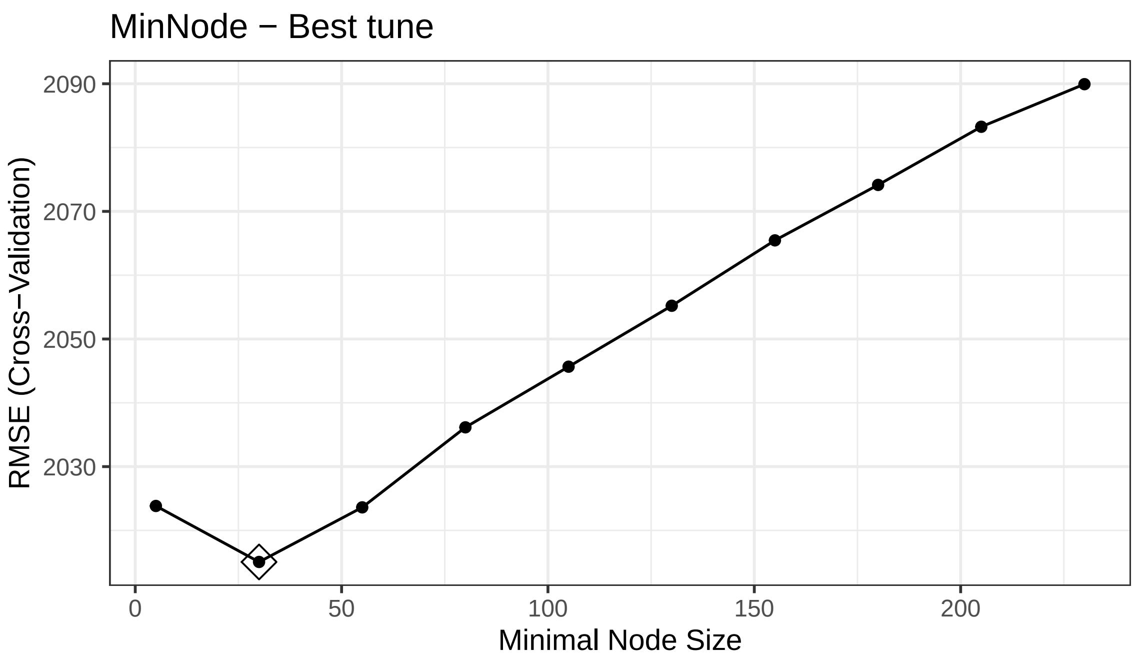
We had an improvement of more 4% in relation to previous best model Knn, or 8% in relation to Linear Regression. Let us plot the results.



We still have a few outliers responsible for most of variance, but they are smaller than the previous models as shown below.

city	property_tax	bathroom	hoa	parking_spaces	rent_amount	y_hat	error
Belo Horizonte	236	2	0	0	12000	1394	10606
São Paulo	1750	5	4800	5	20000	9470	10530
São Paulo	130	2	3177	0	15000	5745	9255
São Paulo	5032	5	4300	8	2500	11678	9178
São Paulo	375	3	1500	2	13800	4697	9103
Belo Horizonte	153	5	2405	4	15000	6510	8490

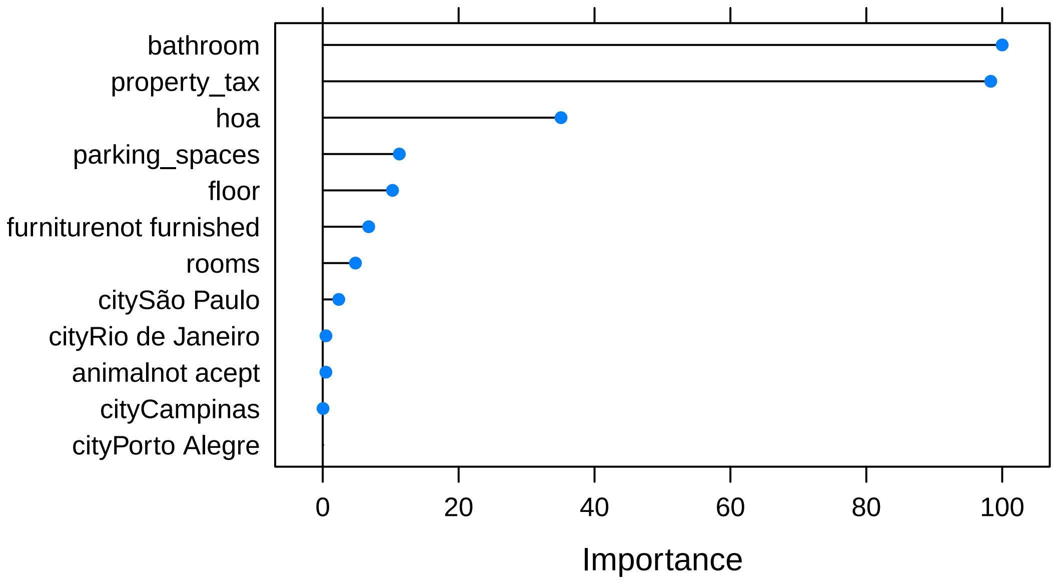
5- Random Forest *method = "Rborist"*



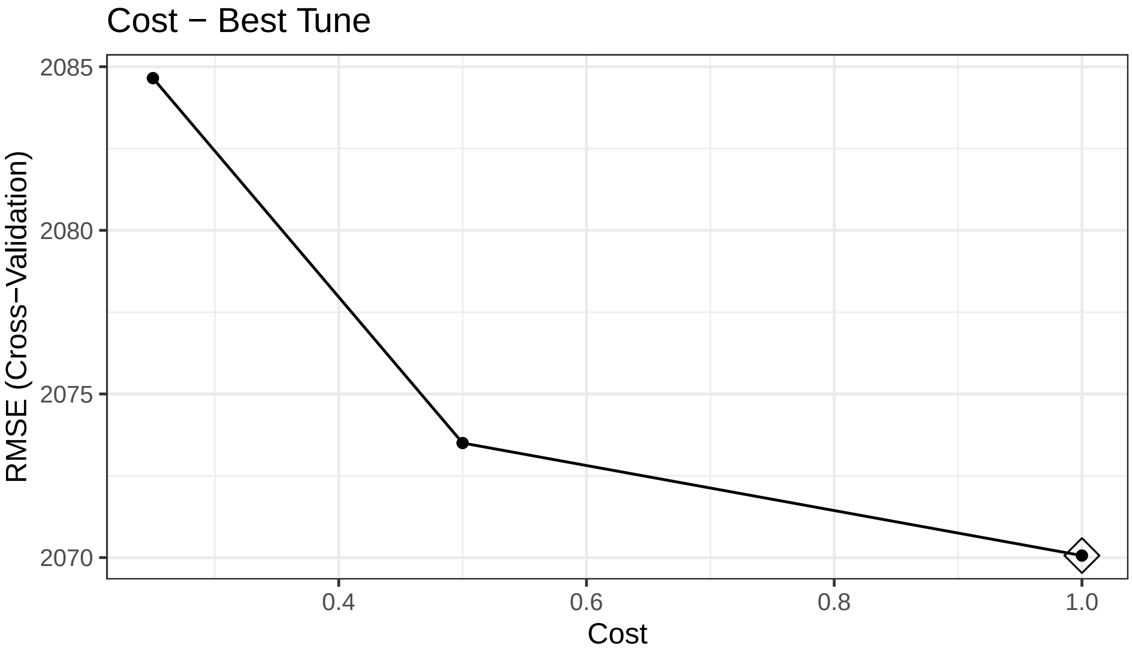
Although the performance on the training data was very similar but inferior to the previous model, on the test dataset, Rborist got better results.

```
##      RMSE  Rsquared       MAE
## 1893.2269    0.6783 1202.7562
```

We can see one change into the features importance.



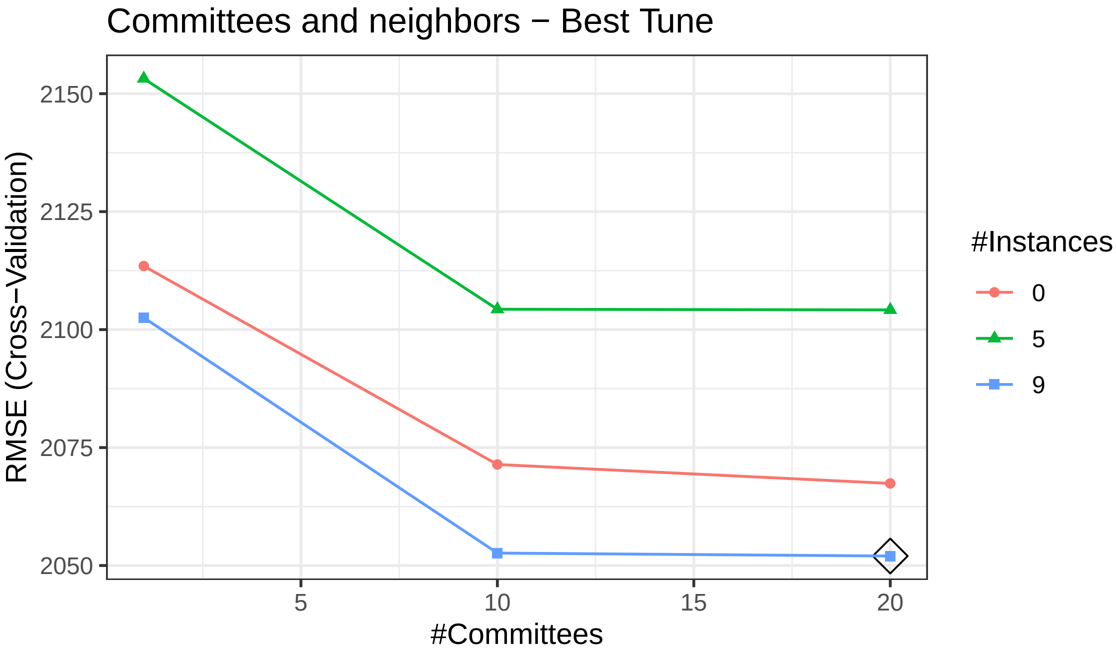
6- Support Vector Machines with Radial Basis Function Kernel *method = "svmRadial"*

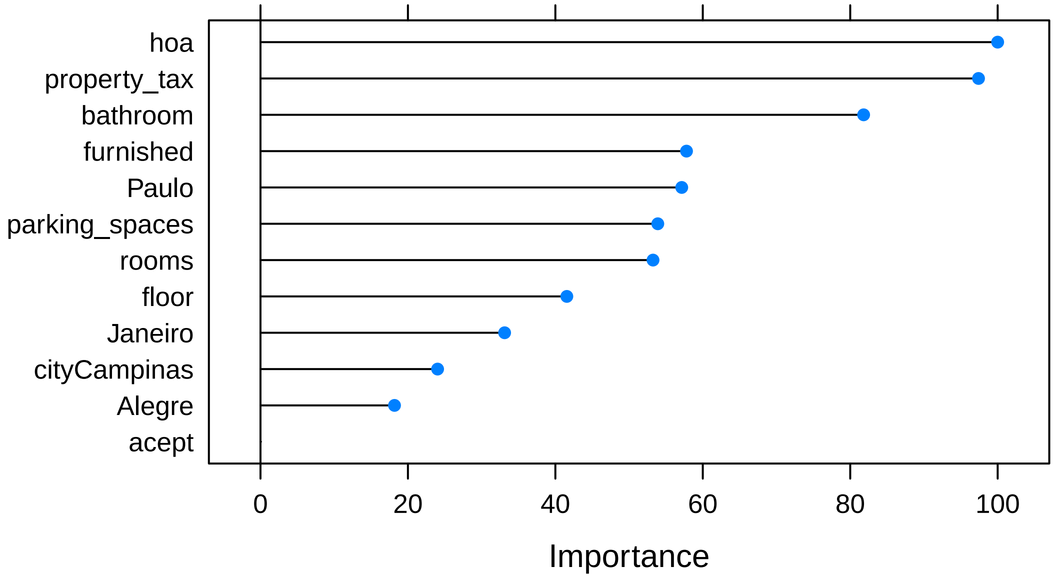


Resulting in a RMSE on the test set equal to 1962 on the test set.

```
##      RMSE  Rsquared       MAE
## 1962.5858    0.6677 1181.5953
```

7- Model Tree *method = "cubist"*

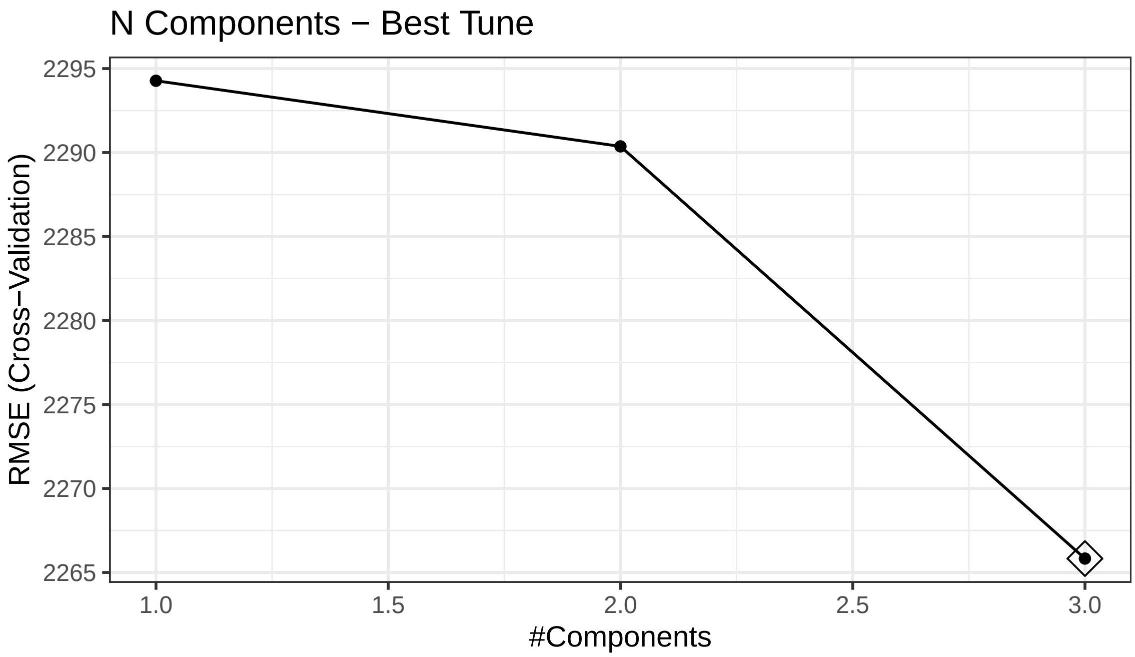




Resulting in:

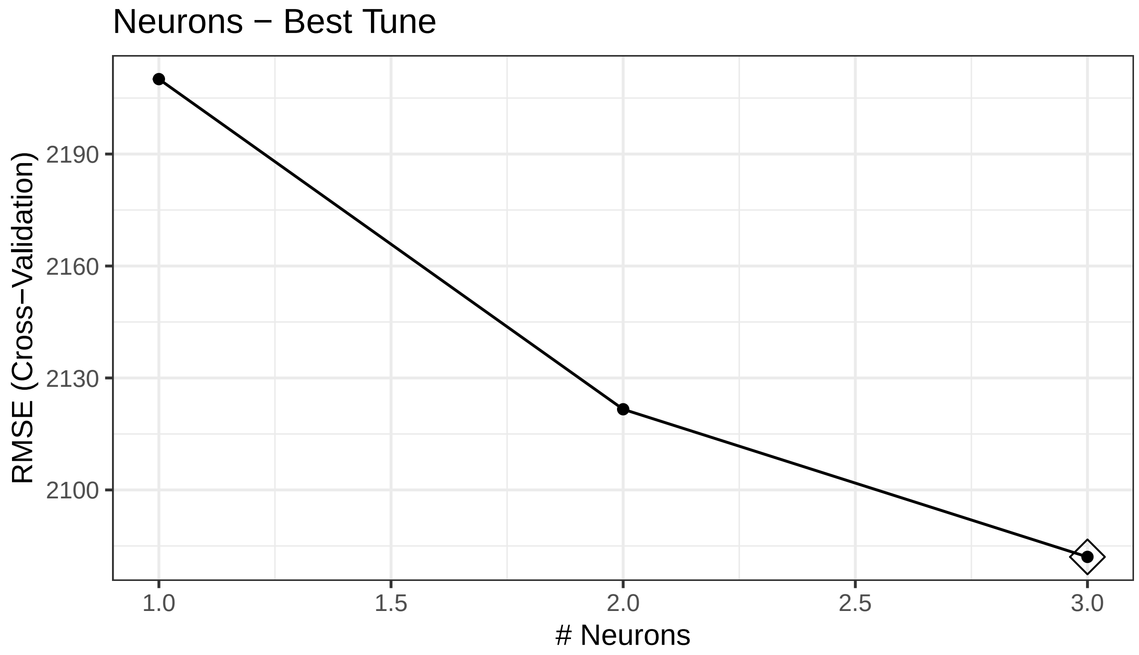
```
##      RMSE  Rsquared       MAE
## 1948.9552    0.6579 1241.8137
```

8- Principal Component Analysis *method = "pcr"*



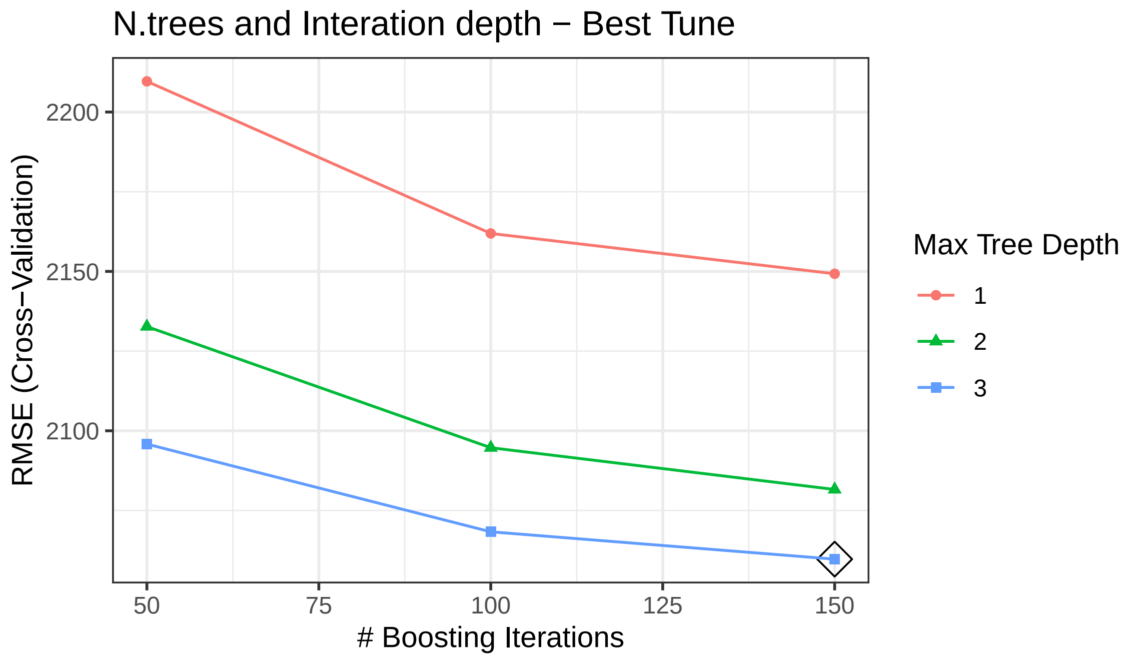
```
##      RMSE  Rsquared       MAE
## 2127.9688    0.5949 1440.4238
```

9- Bayesian Regularized Neural Networks *method= “brnn”*



```
##      RMSE  Rsquared       MAE
## 1946.0140    0.6589 1250.2272
```

10- Stochastic Gradient Boosting *method= “gbm”*



```
##      RMSE  Rsquared       MAE
## 1969.2314    0.6517 1276.0881
```

Now let us compare the results of our models on the test set.

```
##                               RMSE Rsquared  MAE
## Linear_Regression             2079   0.6134 1373
## KNearest_Neighbors            1982   0.6581 1234
## Regression_tree               2101   0.6026 1379
## Random_Forest                 1902   0.6756 1204
## Random_Rborist                1893   0.6783 1203
## SVM_Radial                    1963   0.6677 1182
## Model_tree                    1949   0.6579 1242
## Principal_Component_Analysis 2128   0.5949 1440
## Bayesian_Reg_Neural_Networks 1946   0.6589 1250
## Stochastic_Gradient_Boosting 1969   0.6517 1276
```

11- Ensemble Now we are going to combine the results of the top 3 best RMSE results on the test set and predict with their average.

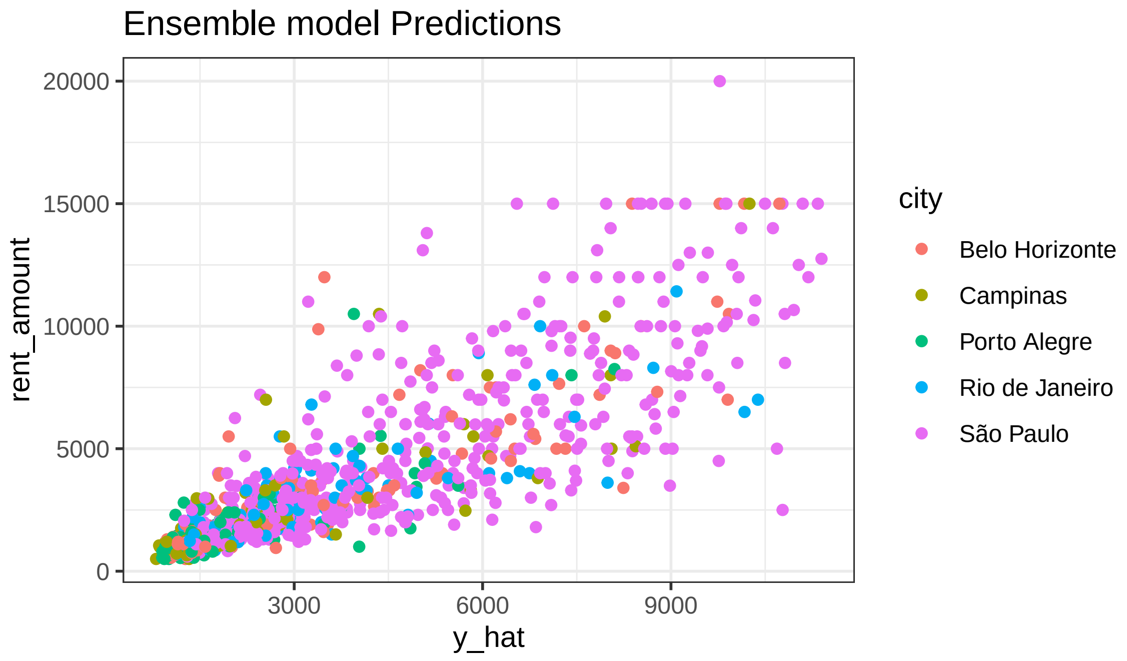
```
##               Random_Rborist                Random_Forest
##                         1893                         1902
## Bayesian_Reg_Neural_Networks
##                         1946
```

Let us make an ensemble model that predicts the mean value of the best 3 previous models and calculate the RMSE.

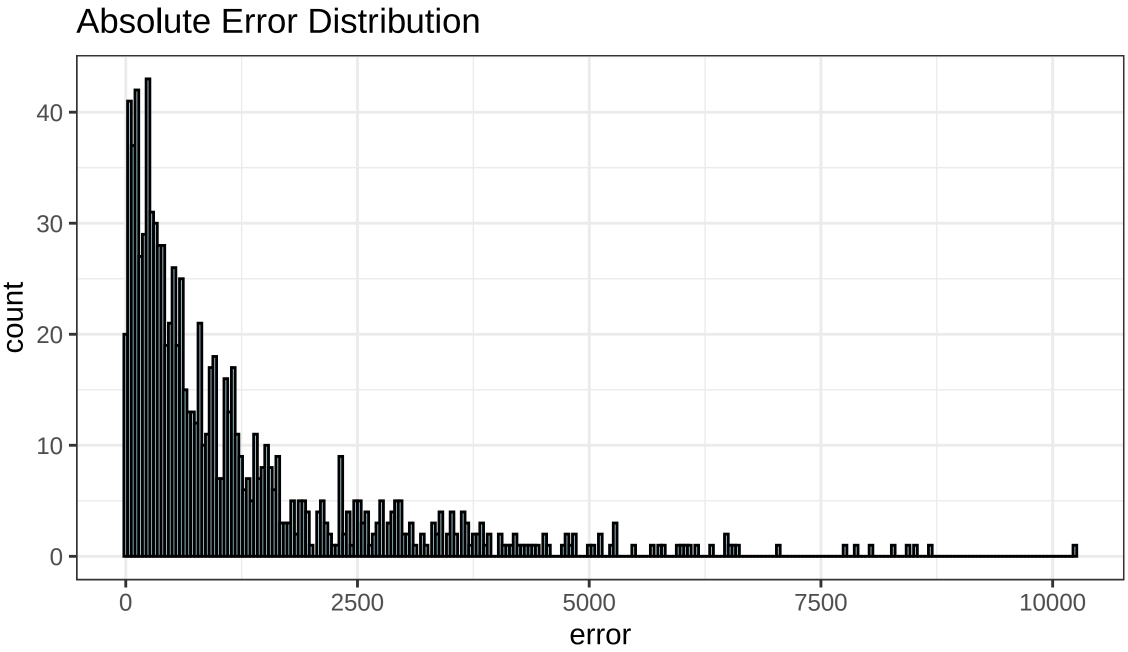
```
##      RMSE  Rsquared       MAE
## 1882.4003    0.6827 1198.6898
```

We have a small improvement. So let us decide to use the ensemble model equal to the mean prediction of Random Forest, Random Forest Rborist and Bayesian Regularized Neural Networks as our final model.

Let us plot our final predictions and compare to actual values.



Although São Paulo is the city with the highest prevalence in the database, it is also the city that contains the largest ranges in features, requiring additional information to improve the accuracy of models.



RMSE is sensitive to outliers. We have some very precise predictions. About 35% of predictions had an absolute error smaller than R$300. We can see most absolute errors were small but the effect of each error on RMSE is proportional to the size of the squared error, thus larger errors have a disproportionately large effect on RMSE.

RESULTS

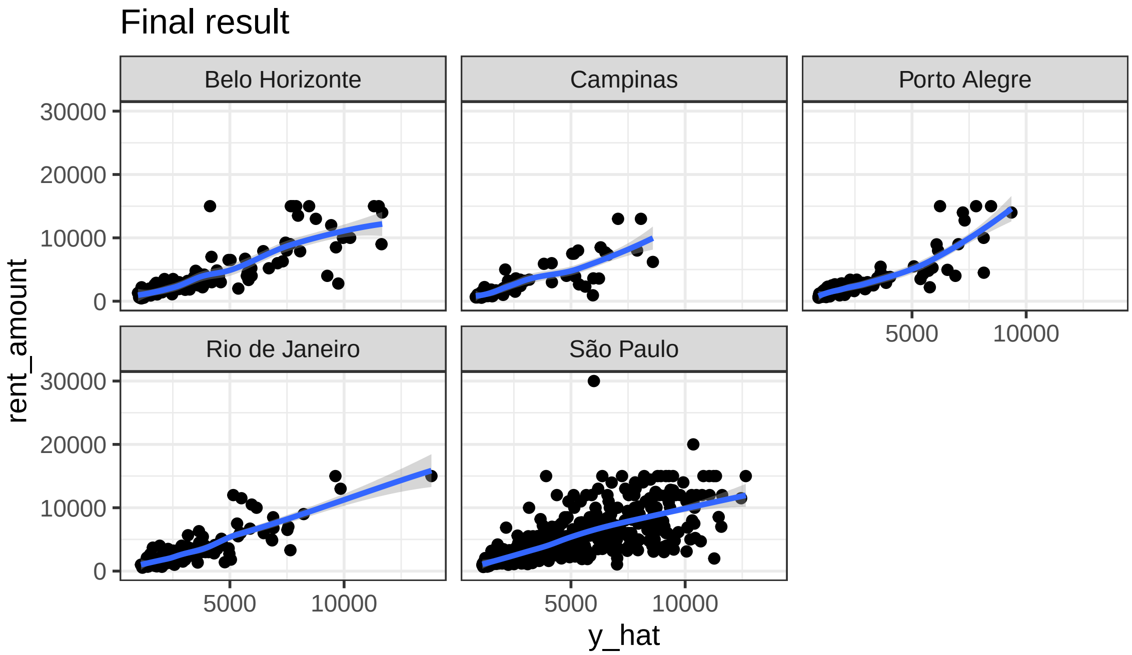
As result we are going to apply our best model - ensemble - on the validation set and see how precise we are.

Validation data has 1071 observations and 13 features. As we did earlier on the build set we must remove features which we found high correlation or dependents variables. Resulting in 10 columns.

Applying the ensemble model to validation set we obtain a RMSE equal to 2171.

```
##      RMSE  Rsquared       MAE
## 2171.5180    0.6266 1286.0880
```

Let us see the predictions by city.



We were able to get very precise results on smaller rentals, and lost precision on higher ones.

The model proved to be satisfactory mainly if we took into consideration the small sample and the lack of relevant features like type of place (studio, apartment, house, country house, etc), neighborhood characteristics as comercial/residential area, average income, population, points of interest (public transportation, scholls, malls, banks, or others).

The higher the data quality the better the accuracy of the algorithms.

CONCLUSION

This project is one part of the final evaluation of the Data Science Professional Certificate from Harvardx-Edx, PH125.9x:Data Science: Capstone, where students should use the tools they learned during the course to solve a problem on your own and write a report explaining the whole process: the techniques used, including data cleaning, data exploration and visualization, the information obtained and modeling approaches.

For this project, we have applied machine learning techniques to solve a problem of our choice, using a dataset available at Kaggle or The UCI Machine Learning Repository. The chosen problem was the prediction of renting houses with a dataset containing information about 10.692 houses in 5 different cities in Brazil and 13 different features. Data was collected from a rental website, on 3/20/20, and it was available at Kaggle.

We tested 10 machine learning models and finally we have chosen the ensemble model, equal to the mean prediction of Random Forest, Random Forest Rborist and Bayesian Regularized Neural Networks as our final model. After applying it in an unknown dataset (validation) we have the results: RMSE = 2171, Rsquared = 0.6266 and MAE = 1286.

Among the available features, the most important were the Property Tax, Bathroom and Hoa on most of machine learning models. We got good results but limited by the quality of the data and the absence of important information, since the economic and population characteristics of the neighborhood where the properties are located that must also be taken into account in consideration to the physical characteristics of the properties. A few outliers are responsible for much of the model error.

Some variables suggested to improve the results are: neighborhood, zipcode or adress information, type of location (studio, apartment, house, country house, bedroom), distance to nearby points of interest, such as:

supermarket, pharmacy, public transportation, hospitals, schools, shopping malls, etc., existence of leisure area, gym, swimming pool, barbecue, etc. Normally this kind of information is described in an open field “Description”.

It is also suggested to review the data periodically to ensure that the information provided is true. Data quality is a key factor to predict the house prices or any other machine learning problem.