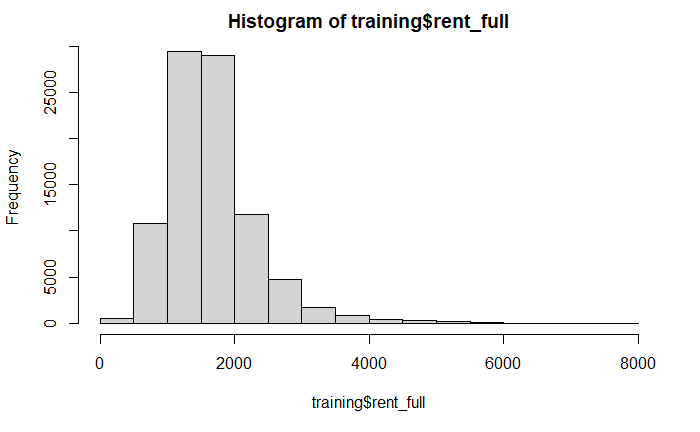
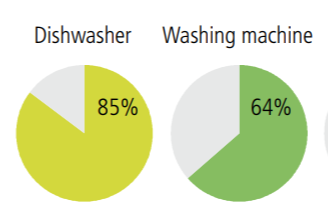
## Training dataset

The dataset we used to train our models was created by using a web crawler across multiple listing website in Switzerland. The dataset contains 100 variables, including our outcome variable: rental.

From a first exploration we can see that most values are distributed between 500 CHF and 4000 CHF, with an average listing price of 1678 CHF and a median value of 1580 CHF. The desired outcome of this exercise is to train our models by using the “training.csv” dataset and test our models on “X\_test.csv”. The final outcome will only show the ID of the listing and the predicted rental price. Let’s explore now the predictors used in our models. To give an overview, we can group them in:

* Geographical identifier: they provide information on where the listing is located around Switzerland (GDENAMK,GDENR,KTKZ,address, lat, lon)
* Features of the listing: they provide additional information about the characteristic of the listing. The predictors in this group provide essential information on the equipment and services available in the listing, more details about the building in which the listing is located, and information regarding the listing itself, like the area.   
  ( *appartments, area, area\_useable, balcony, basement, bath, bath\_tube, bright, building\_plot, cabletv, ceiling, cheminee, date, descry, dishwasher, dryer, elevator, floors, furnished, garden\_m2, gardenshed, heating\_air, heating\_earth, heating\_electro, heating\_far, heating\_gas, heating\_oil, heating\_pellets, home\_type, kids\_friendly, laundry,manlift, middle\_house, minergie, month,msregion, new\_building, newly\_built, oldbuilding, oven, parking\_indoor, parking\_outside, pets, playground, pool, public\_transport, quarter\_general, quarter\_specific, quiet, raised\_groundfloor, rooms, shared\_flat, shopping, shower, size\_land, sunny, terrace, toilets, topstorage, veranda, water, wheelchair*)
* Micro ratings: they provide additional information about the desirability of the listing, when external factors are taken in consideration. Example of external elements are the noise level of the area, the sun exposure of the listing and the distance to the services. Micro ratings help to describe and categorize the area where the listing is located. The first value “Micro\_rating” sum up all the other variables in this group. *( Micro\_rating, Micro\_rating\_NoiseAndEmission, Micro\_rating\_Accessibility, Micro\_rating\_DistrictAndArea, Micro\_rating\_SunAndView, Micro\_rating\_ServicesAndNature,*
* Descriptive variables about the surrounding area: similar to the micro ratings, they provide more information about the desirability of a listing, when calculating external factors. They provide a more detailed view, compared to the micro\_ratings. (*wgh\_avg\_sonnenklasse\_per\_egid, Anteil\_auslaend, Avg\_age, Avg\_size\_household, Noise\_max, anteil\_efh, apoth\_pix\_count\_km2, avg\_anzhl\_geschosse, avg\_bauperiode, dist\_to\_4G, dist\_to\_5G, dist\_to\_haltst, dist\_to\_highway, dist\_to\_lake, dist\_to\_main\_stat, dist\_to\_school\_1, dist\_to\_train\_stat, geb\_wohnnutz\_total, max\_guar\_down\_speed, restaur\_pix\_count\_km2, superm\_pix\_count\_km2, wgh\_avg\_sonnenklasse\_per\_egid, dist\_to\_river)*

If we have a further look at the predictors, we encounter a limitation of the crawler. Most of the columns contains missing values. By cross-validating common knowledge with the data from Federal Statistical Office (2015), our assumptions appear to be correct: the missing value is a limitation of the dataset and doesn’t translate in a boolean “False”. Below an example of the availiabilty of dishware in the Swiss households. In the training data set no listing present a dishwasher.

****Figure 1: Figure 1: Availability of selected consumer goods, 2012 (FSO 2015)

For this reason we are going to reduce the dataset to drop the predictors which have too many empty values, to retain only meaningfull predictors.

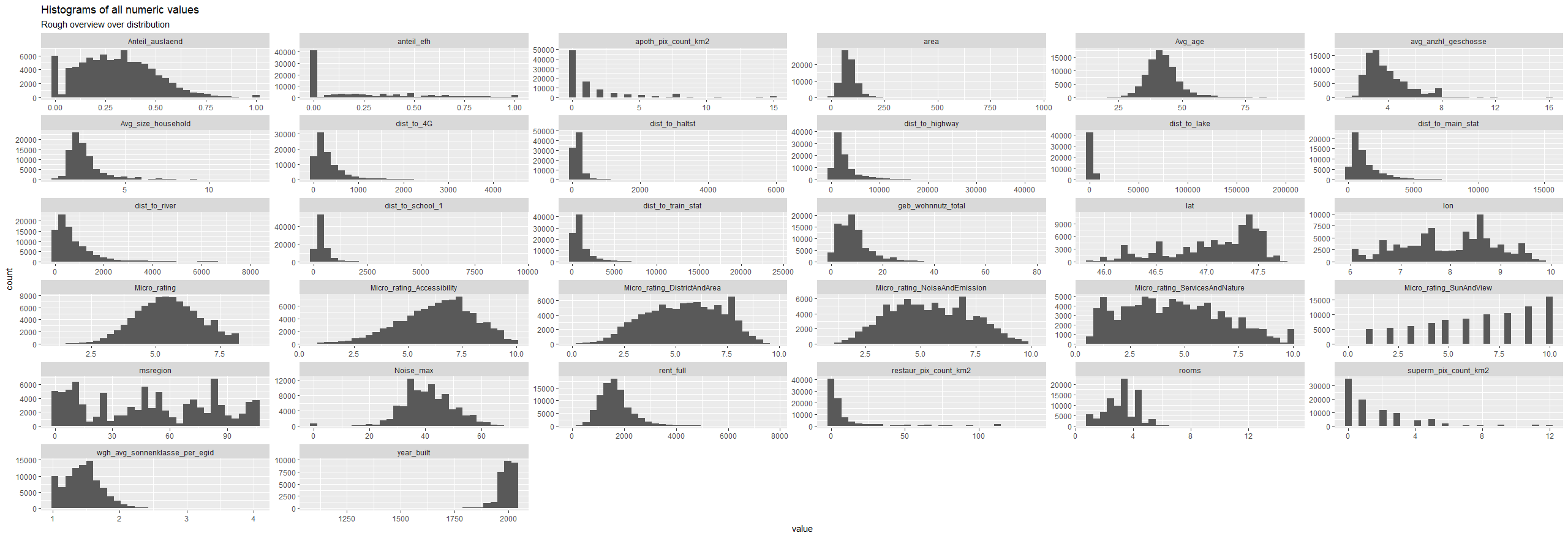
We split the data into 3 datasets:

* **Training\_reduced\_large**: which includes both micro ratings and descriptive variables about the surrounding area. k = 41 variables.
* **Training\_microrating**: with the micro ratings and data about the listing. k = 34 variables.
* **Training\_no\_microrating**: without the micro-ratings to surpass possible covariance in linear regression.

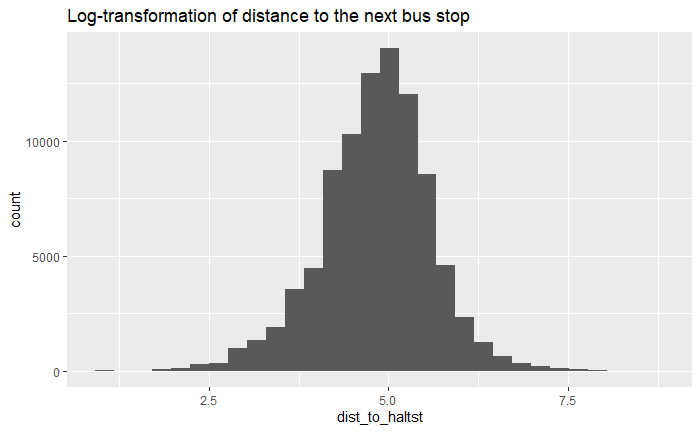
k = 21 variables

### Variables Exploration

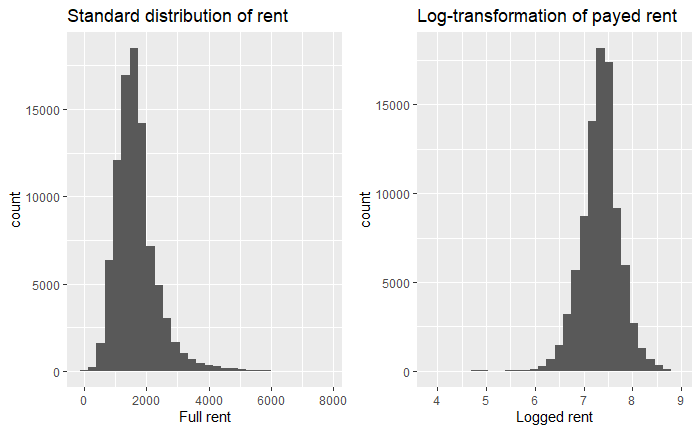
We start the exploration process by preparing the data. To start this process, we check the largest dataset, **Training\_reduced\_large**, to have a first overview of all variables. We first look at the distribution of the variables. The first plot only shows numeric values.

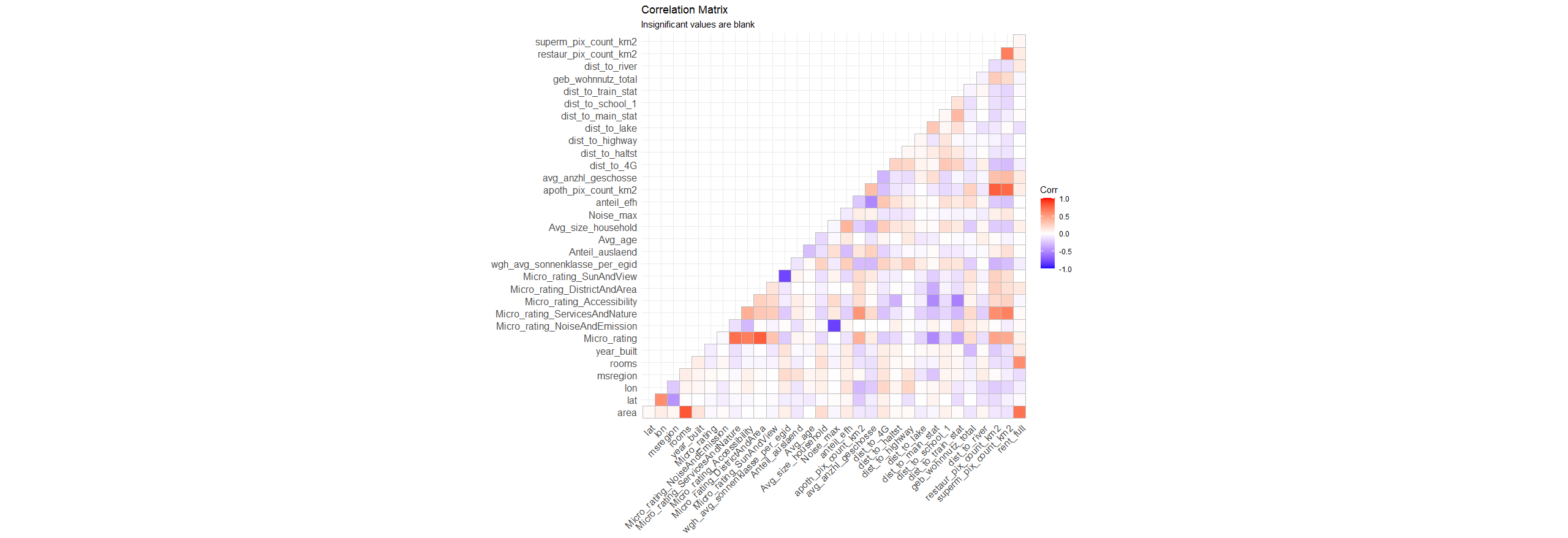


With a quick examination, we can spot that only the group of micro ratings are evenly distributed. Many of the other predictors are not evenly distributed, and they are either left or right skewed. Our first approach is to apply a log-transformation. We start with the variable “dist\_to\_halts” (the distance to the next bus stop), which is heavily skewed to the left. After a log transformation we obtain the following result:

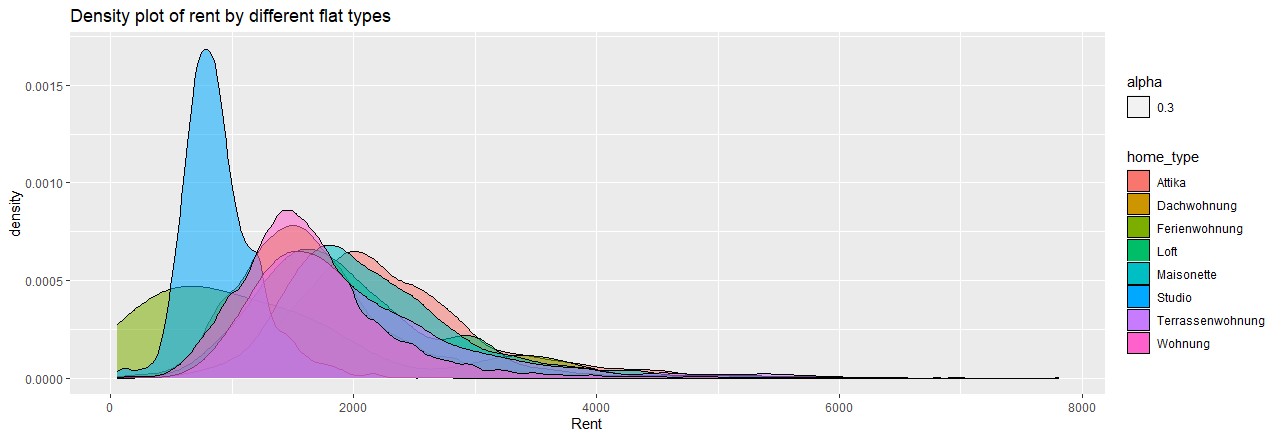


We take a similar approach with the predictors: *“area”, “Avg\_size\_household”, “dist\_to\_haltst”, “dist\_to\_highway”, “dist\_to\_lake”, “dist\_to\_main\_stat”, “dist\_to\_river”, “dist\_to\_school\_1”, “dist\_to\_train\_stat”.*

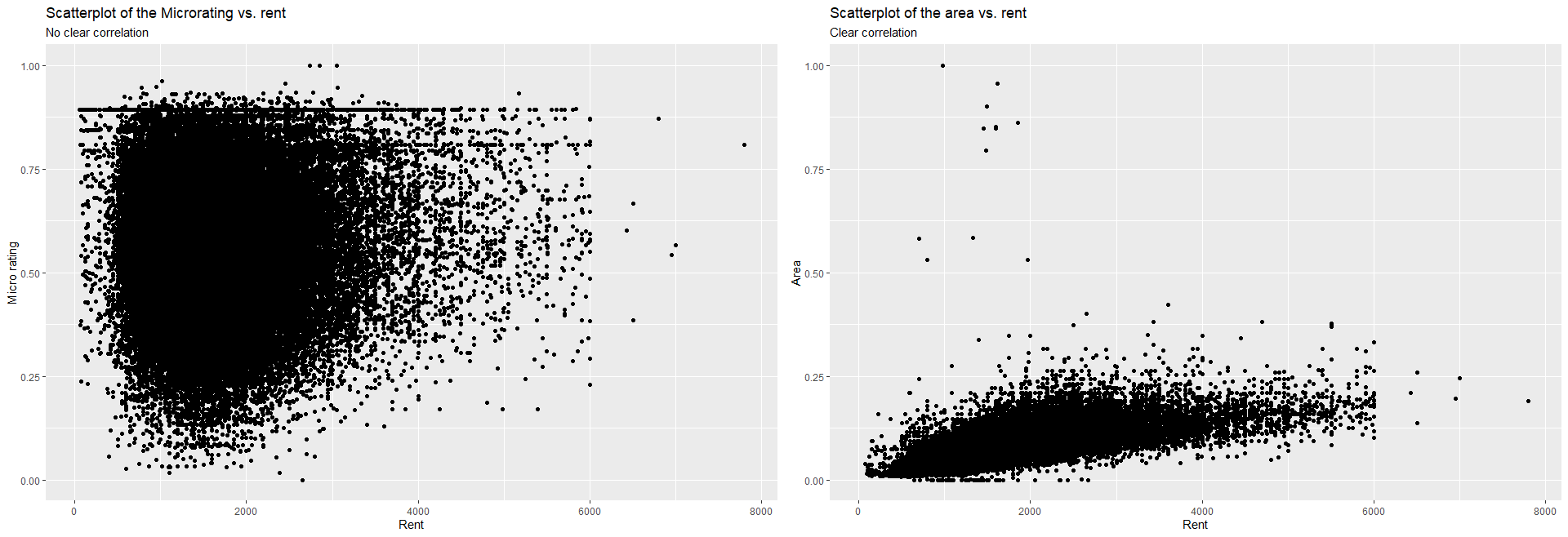


And we apply a log transformation to our outcome variable (rental price) as well. Next, we are going to normalize our data and scaling it so that every numeric value (expect the rent) takes a value between 0 and 1. The normalization of data will allow to implement several models, like Random Trees. After, we look at the correlation plot of our numerical values. 

What we see are some correlation between the regressors, but limited correlation between regressors and the desired outcome: rent. The highest correlation with the rent appears to be with the quantity of rooms and the area of the listing. We can improve if we group the regressors by their **flat type**.



We tried a similar approach, by grouping by **Quarter type** but the result was not significant, with an almost perfect overlap.

Lastly, we create a scatterplot of the micro rating, as predictor for rent and a scatterplot of the area in relationship for the rent. We can already see a clear relationship between area and rent, with only few outliers. 

To sum up, we decided that we will prioritize area and home type as regressors.

## Linear Regression

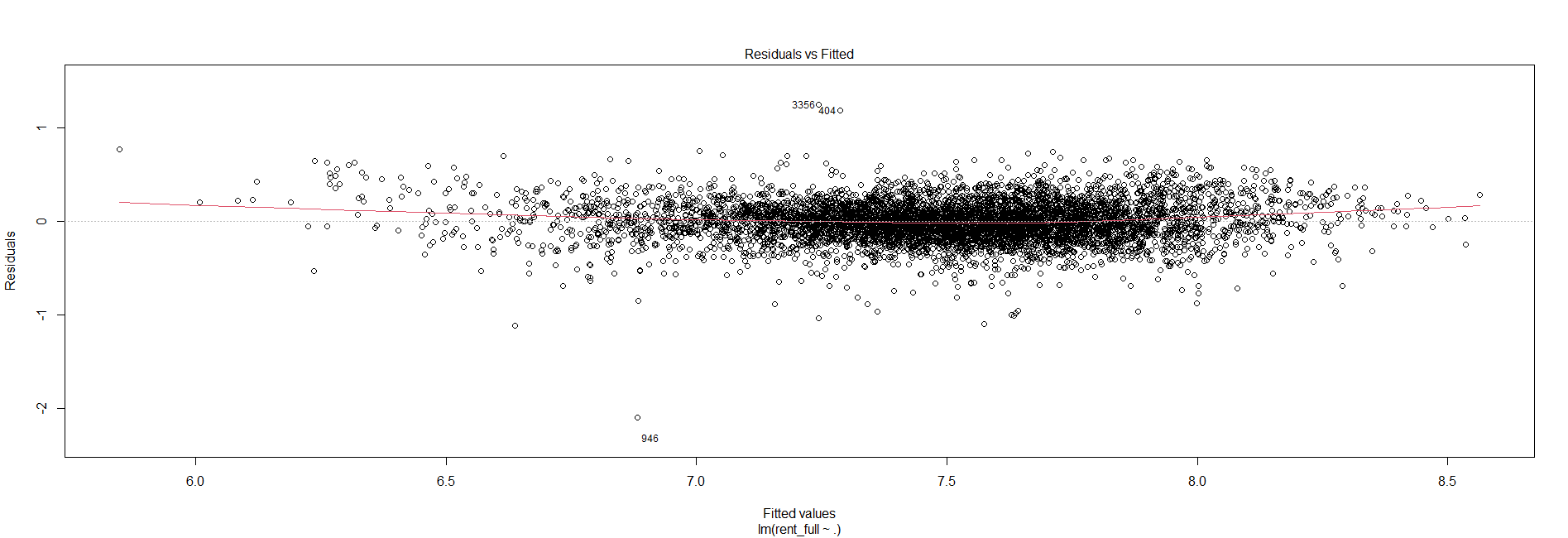
The first method we want to implement is the multiple linear regression. We decided to start with a multiple linear regression because it is fairly easy to both implement and to interpret. However, we expect we might obtain better predictions with more advanced algorithms. We start by comparing the logged and the non-logged dataset. This will provide a benchmark to decide which dataset to utilize and after we will try to improve the accuracy. Both datasets are going to be cross validated, using a 80/20 split and then we will compare the Rooted Square Mean Errors (RSME).

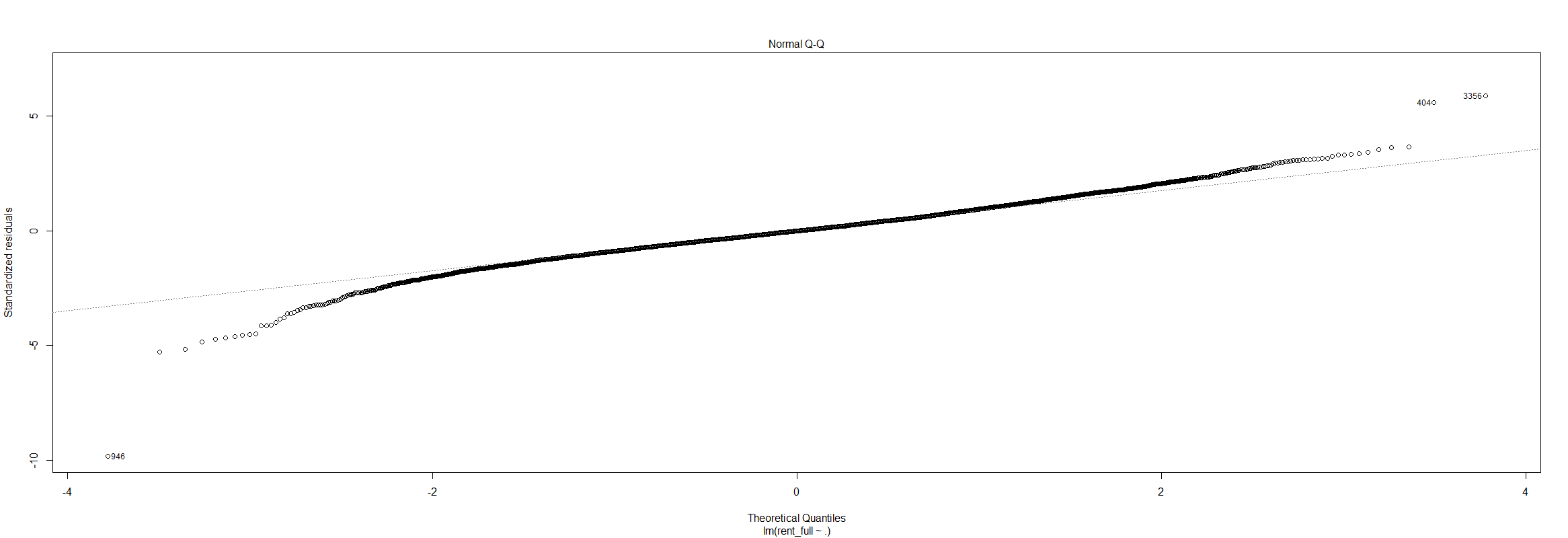
As our model cannot handle missing values, we are first just working with a sample that omits all missing values. This significantly reduces our dataset but should work as a baseline. Because we work with factors that have many possible outcome values like the canton and floors, we are going to leave them out in this first try.

We obtain two linear regression models:

1. With standard values and a RSME of 466
2. With log value and a RSME of 450

We proceed by plotting the log value regression model, to have a visual representation.





From the Normal Q-Q we can assess that the model overpredicts cheap listing and underpredicts the most expensive ones. The next step we are going to take is to include in our model only the predictors which do not contain any missing values.

The dimensions here are dim(no\_na). But as we see from the new RSMEs, this approach kicked out most of the important columns, which left us with a significant decrease of prediction power:

* With standard values and a RSME of 575
* With log value and a RSME of 590

In this scenario, the standard values predicted better, but this is due to model fluctuations; both the logged and the standard values are sometimes better, sometimes worse.

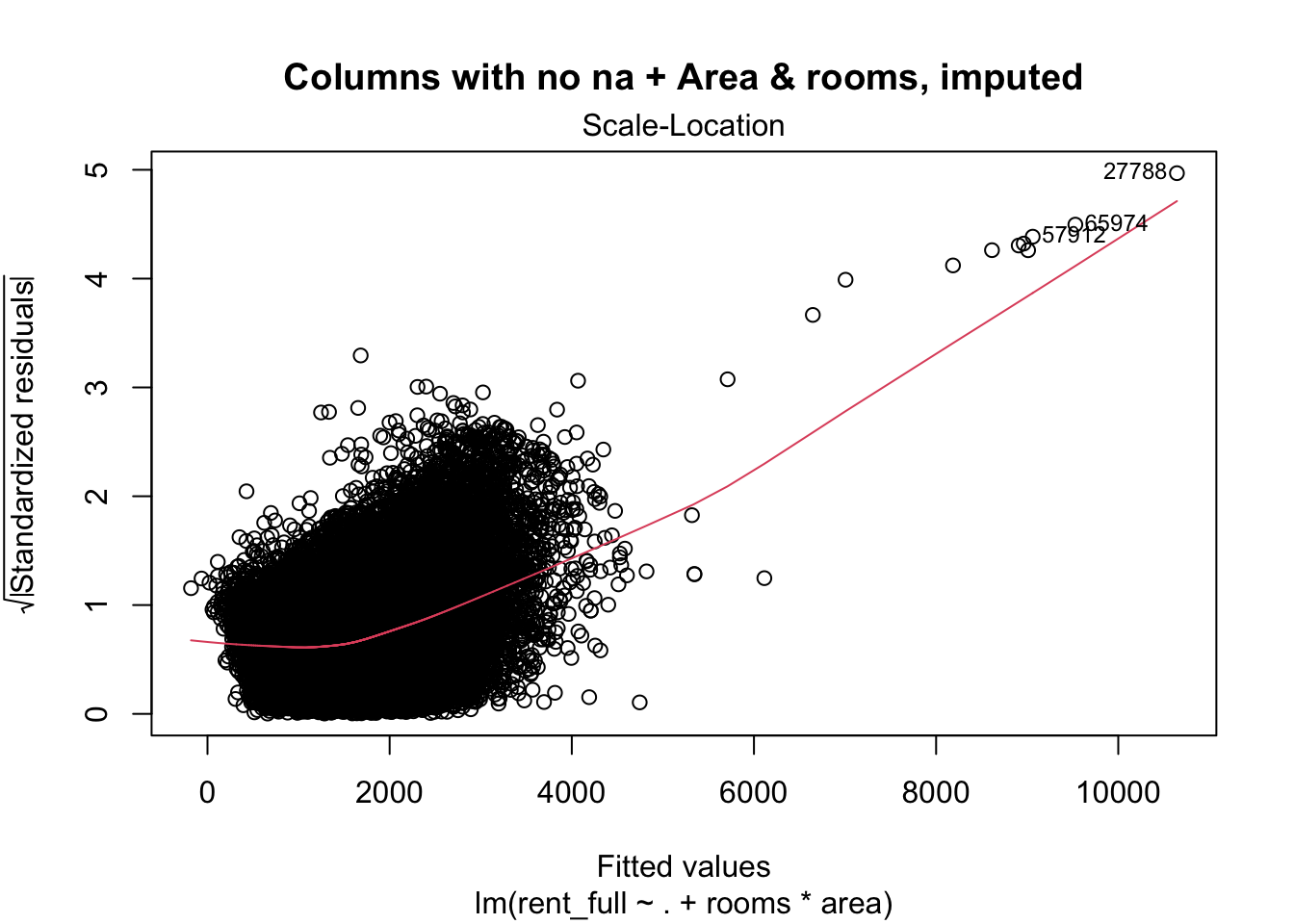
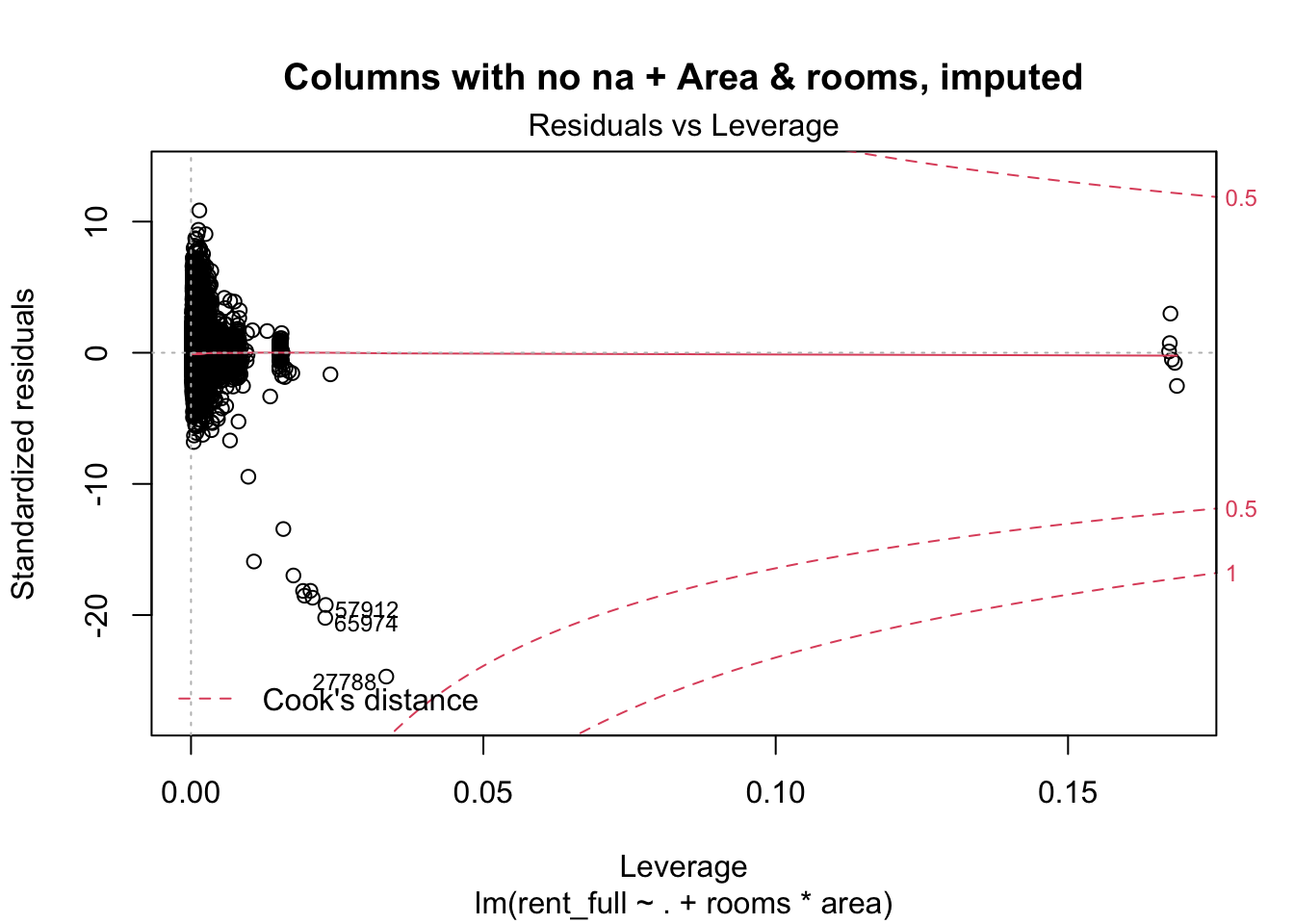
### Linear Regression - Imputation

The learning database contains multiple predictors with many missing values. We are now trying to impute the NAs by using multiple imputation and the mice-package. A full explanation of how the mice-package works can be found [here](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074241/).

First, we are going to impute the full base set with no feature selection. This approach should decrease our prediction power, as our dataset has too many values that do not correlate with each other and therefore generate a high bias. What we see is that an imputation works well for the aggregate level - the coefficients (median, mean & quantile) stay the same, but we lose interpretability on a row by row-level.

We train again our model, by using the new dataset, after imputation. The RMSE for standard values increased from 466 to 1157. We conclude that the imputation of the whole dataset was not effective.

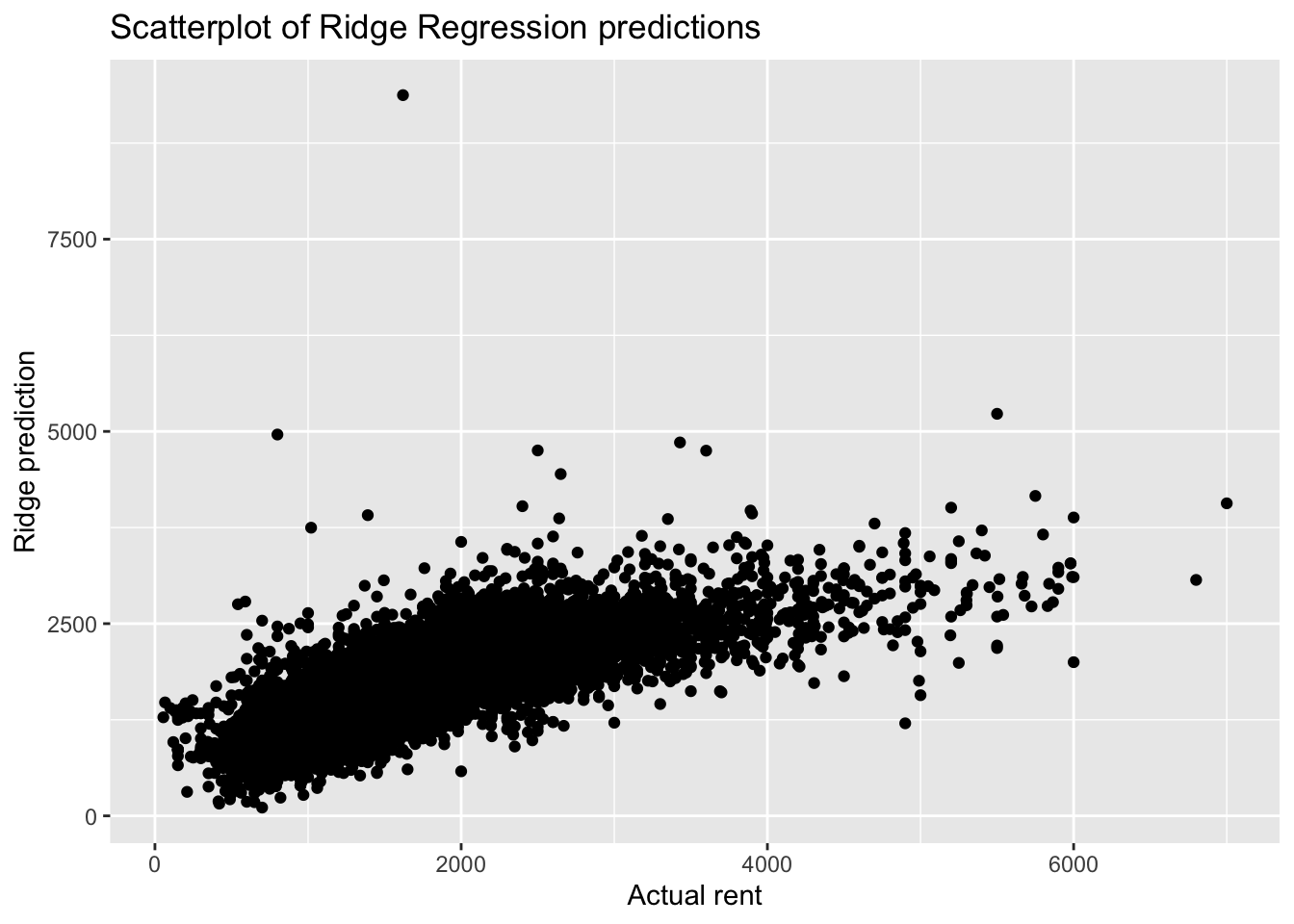
As a next step, we decide to use as regressors columns without missing values (like the Micro ratings) and to only impute critical columns like area and rooms. After this we train again our model and we see a significant improvement with a reduction of RMSE from 466 to 391. This is probably the best result we can obtain by using a linear regression model.



## Ridge and Lasso

As a second model we decide to train a Ridge and Lasso model. The rationality behind this choice is that we want to see if we can improve on the feature selection. We use the normalized data set we gathered in the last computation. The method used was a resampling method that searches for the best lambda-value.

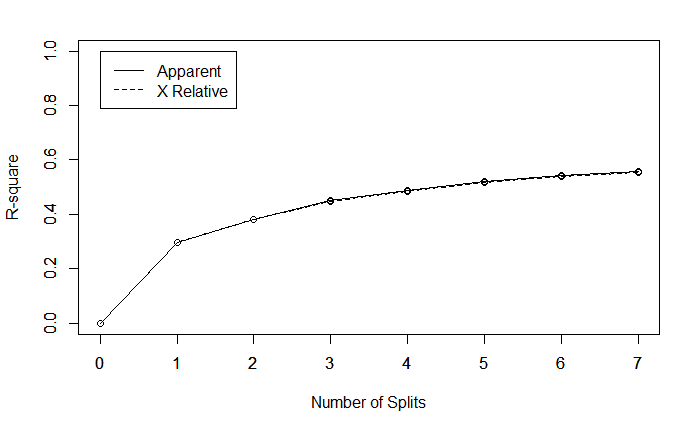
Compared to the previous model we obtain a RMSE of 435 and an R2 error of 0.58.  
All the three models come to similar results using the imputed dataset.

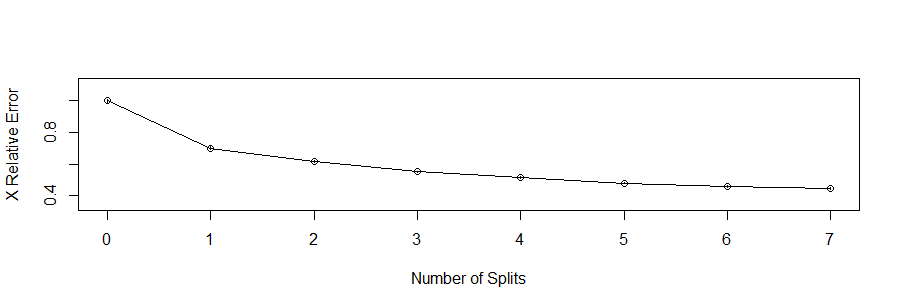


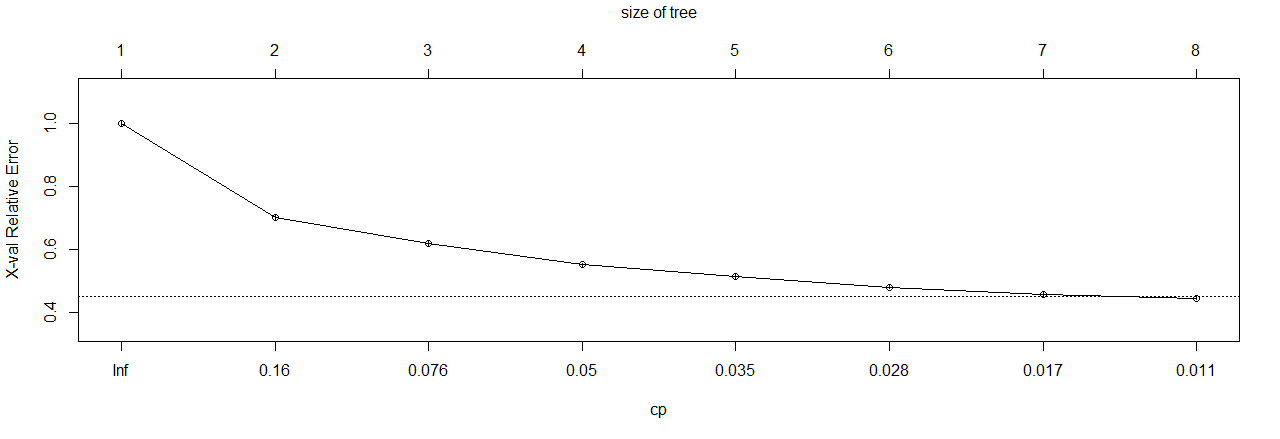
## Random Trees

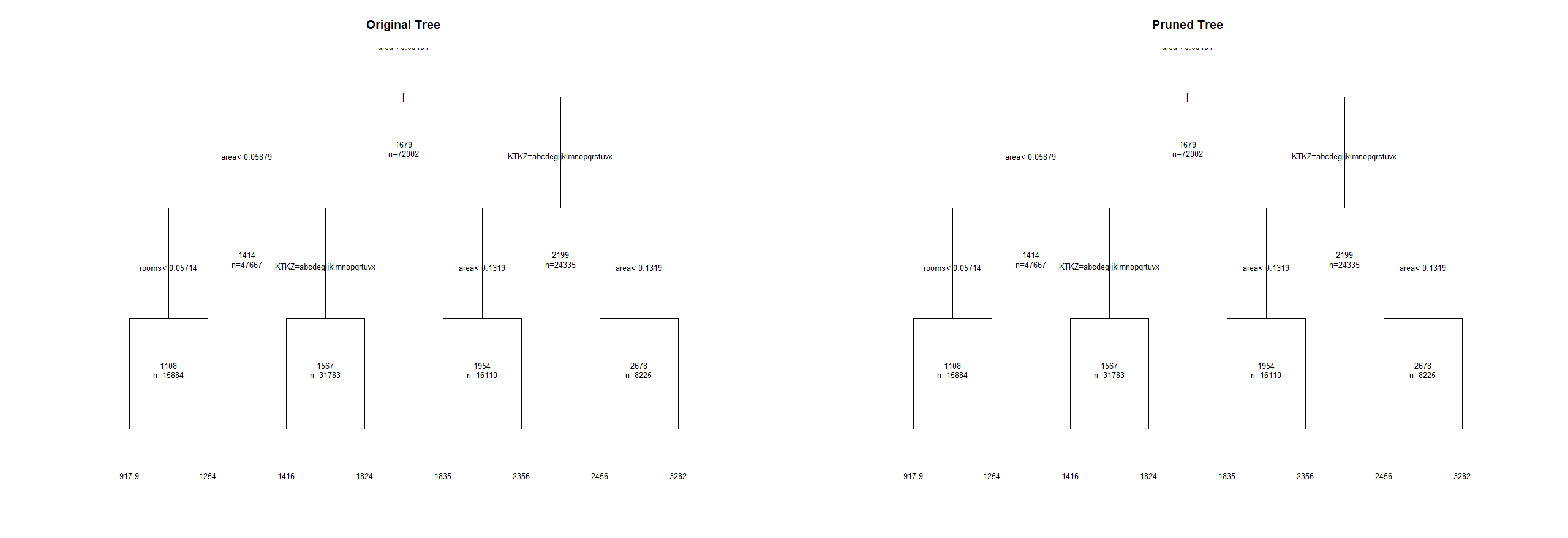
We did not detect any perfect linearity between our regressors and the rent while some categorical values seem to have a strong effect on the dependent variable. Therefore, we switched approach. As a third model we used random trees. Our goal is to build basic regression tree, prune it and then fit a prediction model.

##   
## Regression tree:  
## rpart(formula = rent\_full ~ ., data = X\_train\_tree, method = "anova")  
##   
## Variables actually used in tree construction:  
## [1] area KTKZ rooms  
##   
## Root node error: 3.3387e+10/72002 = 463692  
##   
## n= 72002   
##   
## CP nsplit rel error xerror xstd  
## 1 0.297313 0 1.00000 1.00002 0.0099029  
## 2 0.085528 1 0.70269 0.70274 0.0072257  
## 3 0.066836 2 0.61716 0.61802 0.0061745  
## 4 0.036913 3 0.55032 0.55119 0.0060384  
## 5 0.033072 4 0.51341 0.51431 0.0058879  
## 6 0.023125 5 0.48034 0.48134 0.0052588  
## 7 0.013212 6 0.45721 0.45920 0.0048157  
## 8 0.0100 00 7 0.44400 0.44599 0.0048046









We can see that the RMSE of 451 is comparable to the one from the Ridge and Lasso method. Until now the best model would appear to be the linear regression model based on the imputed dataset.

### Conditional Inference Trees

To improve the result, we will try to use a more sophisticated regression tree method: conditional interference trees.

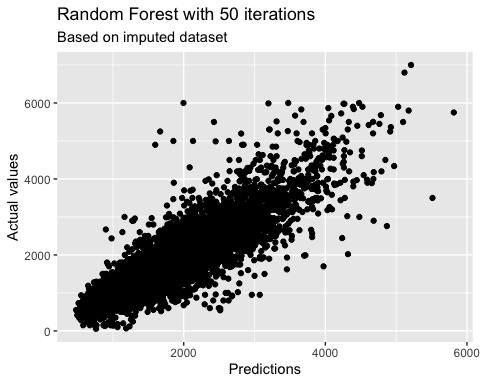
The conditional trees work as follows:

*“ctree, according to its authors (see chl’s comments) avoids the following variable selection bias of rpart (and related methods): They tend to select variables that have many possible splits or many missing values. Unlike the others, ctree uses a significance test procedure in order to select variables instead of selecting the variable that maximizes an information measure (e.g. Gini coefficient).”*

With a RMSE of 375 they are better at predicting the housing prices compared to all the previous models.

## Random Forest

As a last method we decided to use the random forest. With the possibility of handling large data sets with high dimensionality, we expect to obtain good results.



With a RMSE of only 309, the random forest of n = 50 trees got by far the best result! The output values are nearly linearly distributed in comparison to the actual values.

As the random forest produced the best results we’ve seen so far, we are going to predict the rent of the test set using random forest regression and:

* Recreating the column selection by using only the data with no NAs plus floors and area.
* Normalizing the data, so that it follows a 0/1 scale.
* Imputing the missing values in area and floors using Multiple Imputation

## Conclusions