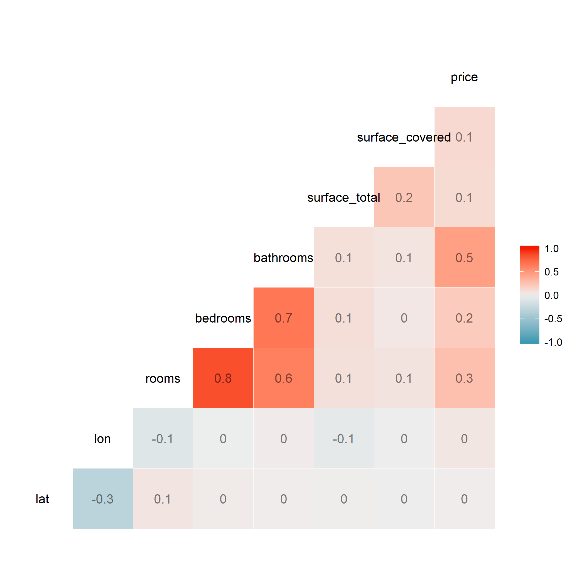
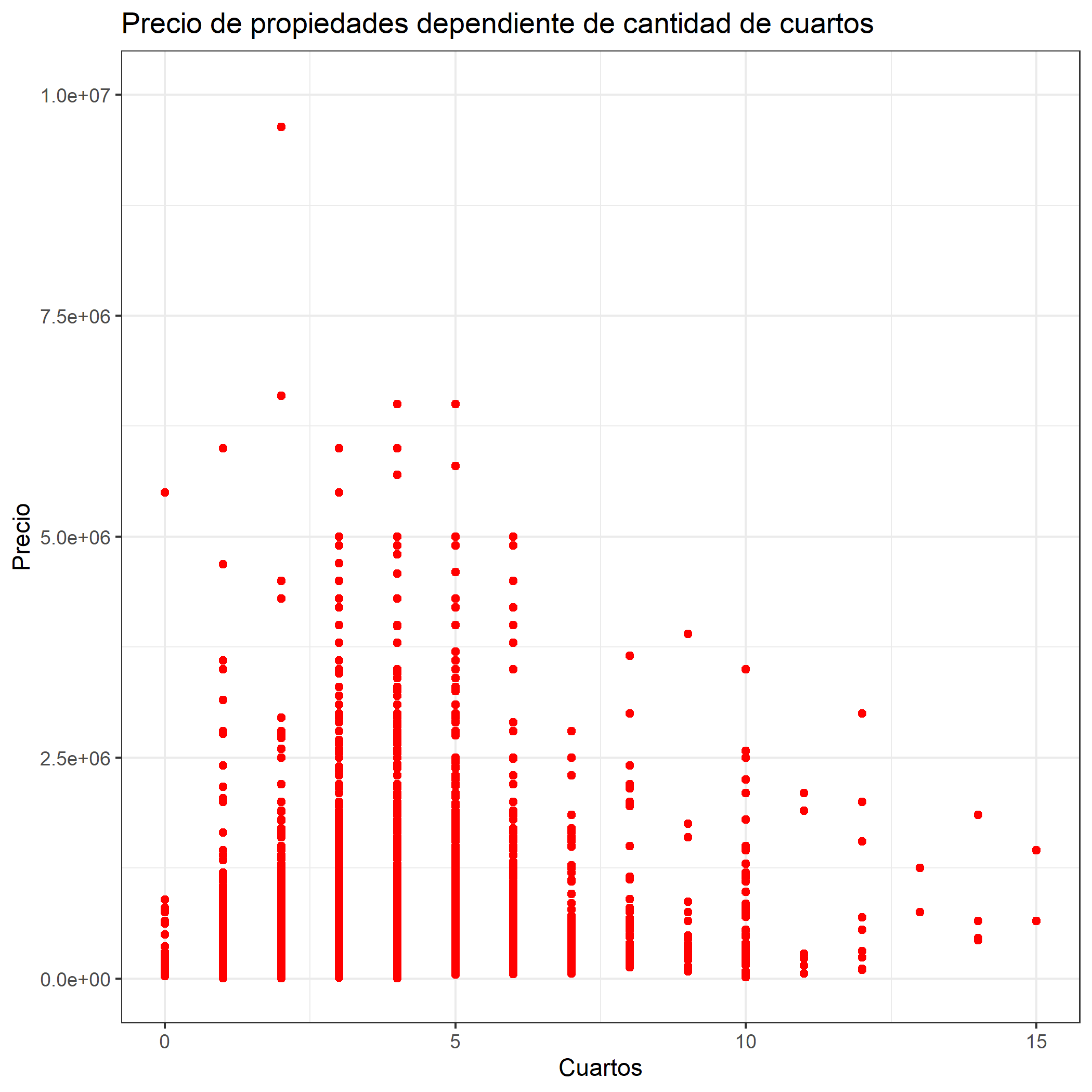
## Exploratory Analisis

We start by making a simple correlogram of the variables to get an idea of ​​the relationship between the variables. We observed that the correlations between the variables (numerical at least), were not very strong. The strongest correlations were with the number of bathrooms and the number of rooms in the house (and they weren't even that high, 0.5 and 0.3 respectively).

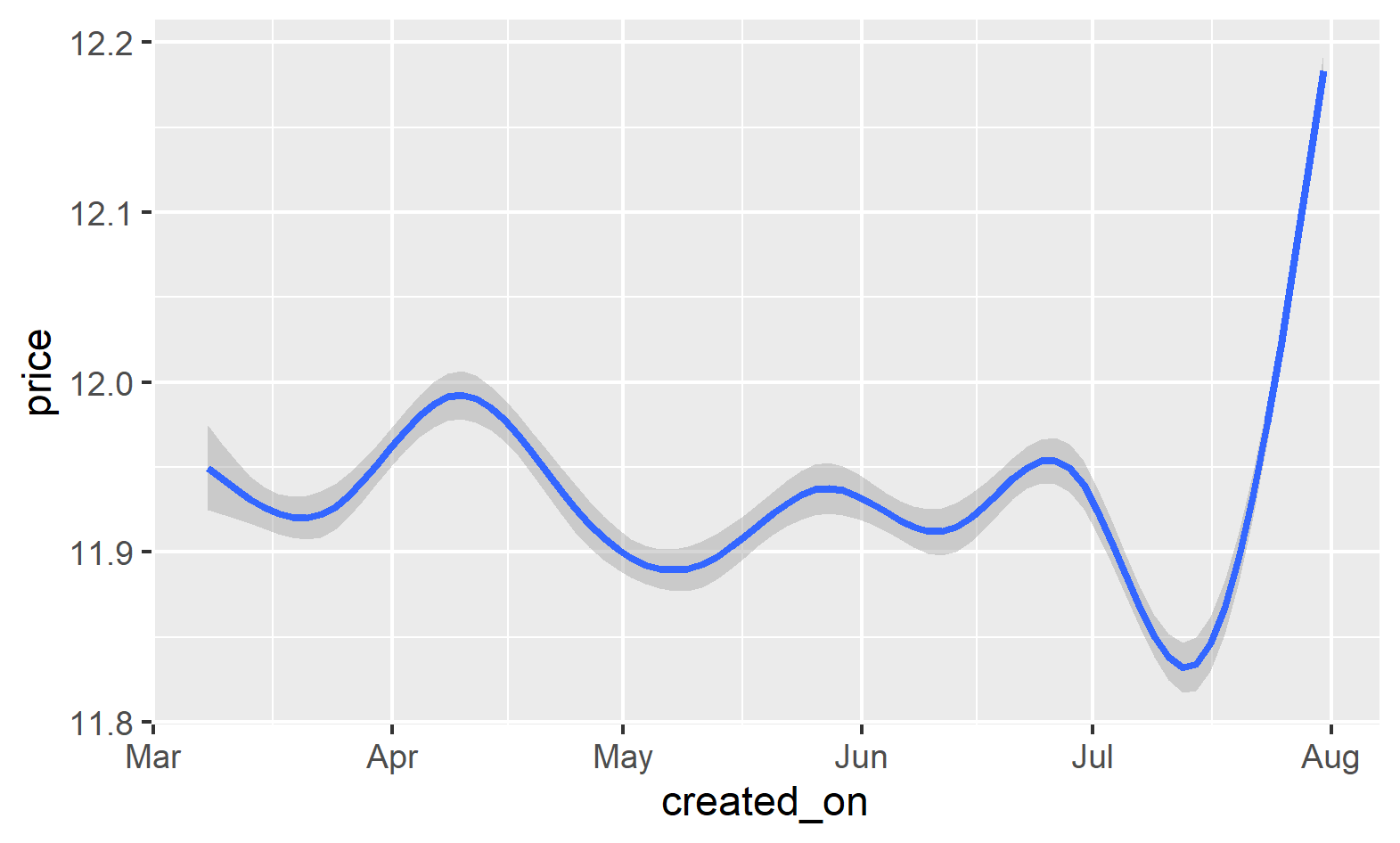
 Then we make a table that shows the number of missings in each variable in the data set. We find that there are variables, which we consider important, that have several NAs. As a result of this, we decided later that we would impute these data with means or modes dependent on other variables (mostly location “l2” and “l3”).

|  |  |
| --- | --- |
| "lat" | 36255 |
| "lon" | 36255 |
| "l1" | 0 |
| "l2" | 0 |
| "l3" | 7615 |
| "l4" | 183312 |
| "l5" | 267032 |
| "l6" | 269224 |
| "rooms" | 100356 |
| "bedrooms" | 143302 |
| "bathrooms" | 64239 |
| "surface\_total" | 41270 |
| "surface\_covered" | 60975 |
| "price" | 48755 |
| "price\_period" | 160558 |
| "description" | 5 |
| "property\_type" | 0 |
| "operation\_type" | 0 |

We find that column l6, for example, has only missings, and almost the same happens with l5 and l4, so we then decided to exclude them from our model. Seeing that rows l3, bedrooms and bathrooms had missings, and we consider them to be important variables, we decided to impute them. In the case of l3, we impute the missings with the most frequent location (the mode) according to the category of l2, that is, if the most frequent neighborhood in Maldonado was Punta del Este, it would impute that value in the column if the property was in Maldonado. It happened that some l2 locations did not have categories in l3, therefore, in that case, we imputed the same value of l2 in l3 (it happened with the properties in Miami, where the district was not specified). Then, for bedrooms and bathrooms we impute the average value depending on the location l3, that is, if the average number of bathrooms in the San Isidro neighborhood was 3, it would impute that value if the property was in San Isidro. As we can see in the table, several variables have many missings, but we believed that these three were the most important and we could not find consistent ways to impute the variable "price period", for example. However, it would not be consistent to impute all the missings in the table, as it would ruin the data set too much.

Then we make some graphs to understand a little more the distributions between the variables. We note that when we plot the price in terms of the number of rooms, it has a similar distribution to the normal with the mean shifting to the left. It is logical that with more rooms, higher priced properties begin to appear since it implies (generally) a larger size. When imputing the data, we verified that the distribution did not change too much and it did not happen.

Training set separation and validation

Regarding the choice of the validation set, we think the best policy was that you take random data from the data set. In our case, we chose to choose 2% of the rows in the dataset to separate for the validation set. We decided this because, as we can see in the graph below, the distribution of the price according to the date of creation of the publication, from mid-July there is a sharp rise in prices. As we did not want it to validate only with this data (and train with the rest, therefore, giving an erroneous value of RMSE) or vice versa, that these data are not in the validation and thus causing the same problem, we believed it prudent to make a selection when random so that this does not happen.

## Create variables

## We add variables through the "bag of words" process, which searches for words within the description and title text, and creates a variable for each word that appears, removing the "stopwords" (words like "and", "is", " here ”) that are not considered important to predict the price.

## Also, we removed the word "no" from R's set of stopwords because we thought it was not trivial.

## The most important variables of words were: "bedrooms, lot, dependency, route, kitchen, service, neighborhood, garage and land". It should be recognized that without bag of words, grepl would have taken us much longer and we would have ignored many words that did not seem important and ended up being. For example: Route.

Hyperparameters and predictive model

The predictive model we used was XGboost and we didn't try other models because we thought it was a waste of time. Not so with hyperparameters.

To find the best hyperparameters, we use a random grid, several times with different maximums and minimums. The strategy we used was to start with very separate highs and lows and we narrowed and moved them based on the results obtained.

Also, we saw that hyperparameters that were good for one validation data set were not good for others.

For the final code the best hyperparameters we found were:

Max\_depth = 24; eta = 0.0546; gamma = 0.0992; colsample\_bytree = 0.451; subsample = 0.4803; min\_child\_weight = 0.6184; nrounds = 1400.

Broadly speaking, we saw that it was more efficient for us to add more trees to our model (nrounds) but learn less from each tree (eta).