Report

A predictive model of house prices in King County, USA

Nicolas Carmona, Niklas Tillenburg, Janek Teders

January 21, 2019

Contents

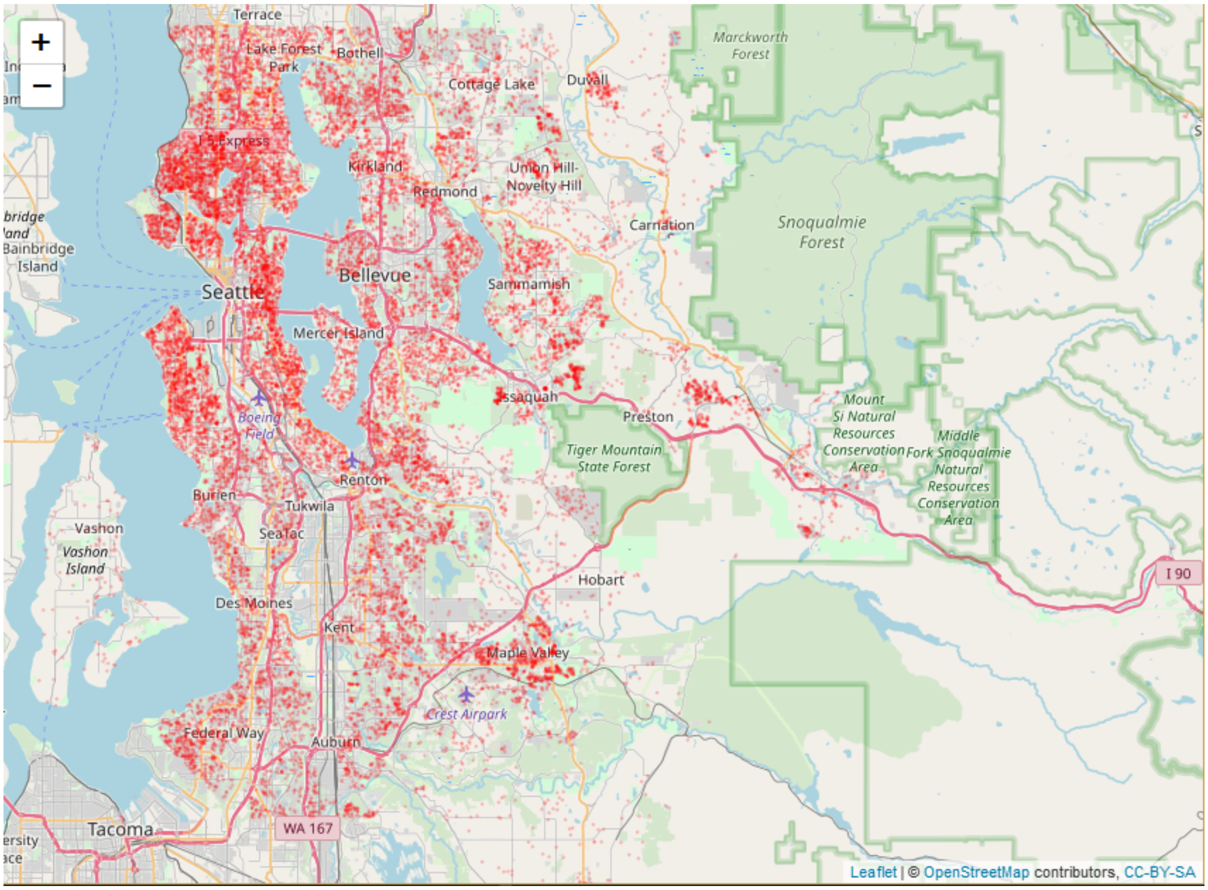
Disclaimer: All members of this group contributed equally to this project.

The goal

Our aim in this project is the construction and refinement of a predictive model which predicts with the highest accuracy possible the price of future house sales in King County, USA, which may be applicable to other counties as well.

The dataset

The data consists of house sale prices for King County area, Washington State, which includes Seattle. It contains houses sold between May 2014 and May 2015. The dataset was obtained from https://www.kaggle.com/harlfoxem/housesalesprediction on December 2018 and consists of 21613 observations in 21 variables. The following map displays the geographical distribution of the data points. We can observe the data points being spread around most of the urban areas with smaller amounts in the rural areas.



The variables

Below is a description of each one of the variables available in the data:

- id: Unique ID for each home sold
- date: Date of the home sale
- price: Price of each home sold
- bedrooms: Number of bedrooms
- bathrooms: Number of bathrooms, where .5 accounts for a room with a toilet but no shower
- sqft_living: Square footage of the apartments interior living space
- sqft_lot: Square footage of the land space

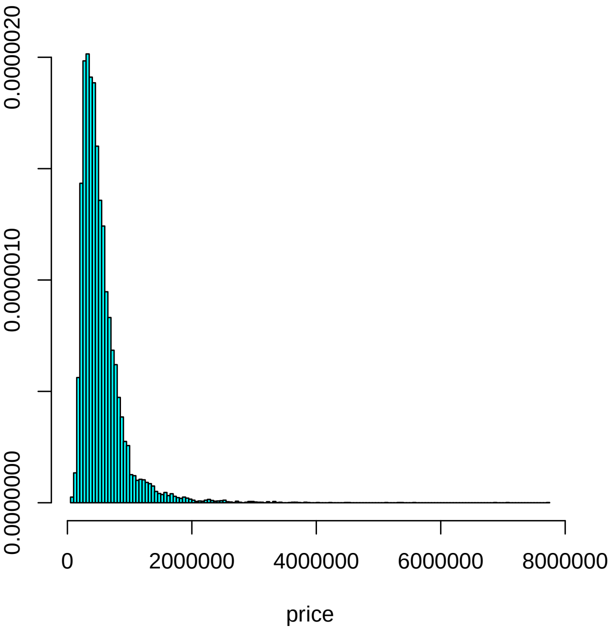
- floors: Number of floors
- waterfront: A dummy variable for whether the apartment was overlooking the waterfront or not
- view: An index from 0 to 4 of how good the view of the property was
- condition: An index from 1 to 5 on the condition of the apartment,
- grade: An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design.
- sqft_above: The square footage of the interior housing space that is above ground level
- sqft_basement: The square footage of the interior housing space that is below ground level
- yr_built: The year the house was initially built
- yr_renovated: The year of the house's last renovation
- zipcode: What zipcode area the house is in
- lat: Latitude
- long: Longitude
- sqft_living15: The square footage of interior housing living space for the nearest 15 neighbors
- sqft_lot15: The square footage of the land lots of the nearest 15 neighbors

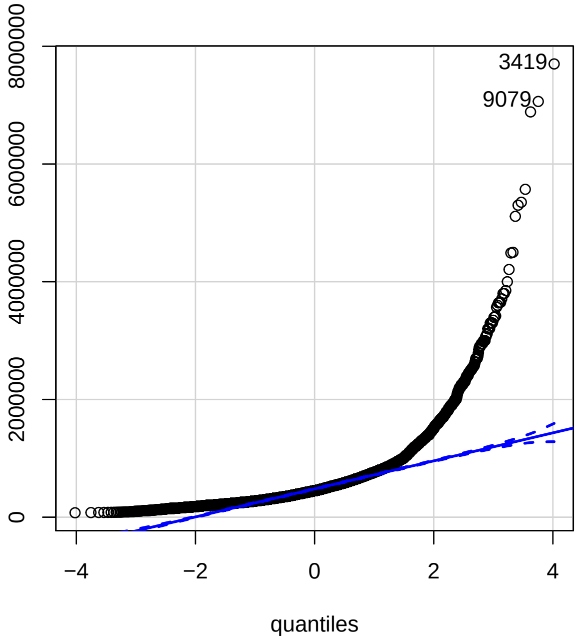
The response variable: numeric and visual inspection

Here we observe some descriptive statistics about the sale prices:

minimum	q1	median	mean	q3	maximum
75000	320000	450000	538212.5	641100	7700000

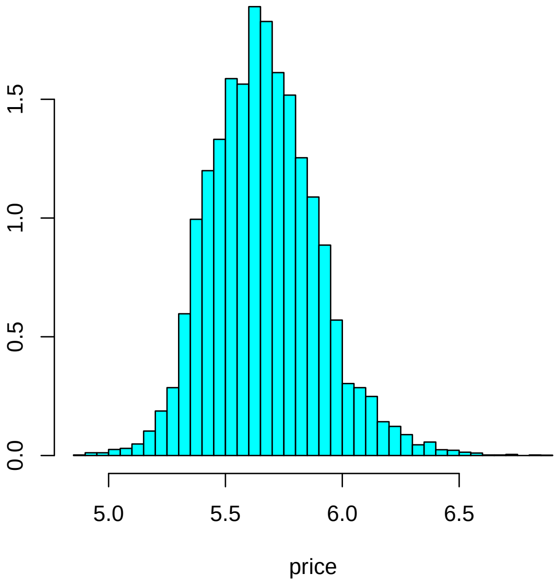
By looking at the median and mean, we suspect skewness to the right in the data. We verify this in the following visual inspection of the data:

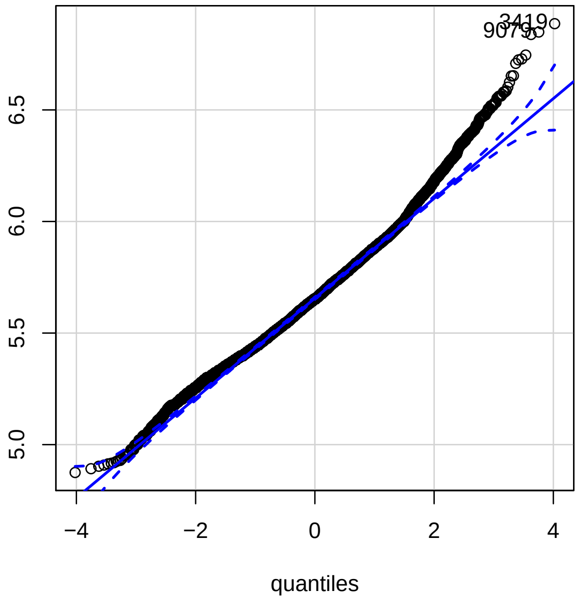




Additionally to confirming our suspicion of skewness to the right, we observe in the qq-plot (the one on the right) that the distribution does not fit the theoretical quantiles of the normal distribution. This goes against the "utopian" assumption of normality for the response variable when attempting linear regression analysis.

In order to improve this, we apply a simple transformation to the response variable. The data is transformed to it's corresponding logarithm with base 10. The following are the histogram and qq-plot of the transformed variable:





Although a perfect fit is difficult to achieve, the transformation proves to be a huge improvement in terms of the normality assumption.

The model selection

In order to pick the best model, criteria must be specified. The p-value presents the problem of multiple testing, and, since we will be conducting several statistical tests, will only serve as a supporting criteria.

Given this, the following metrics will be used for model comparison instead:

- AIC
- BIC
- $RMSE$
- $T - Test\ (p - value)$

The testing procedure

The first model criteria we are considering are the AIC and the BIC. Those are defined as:

$$AIC = -2logL(\hat{\theta}) + 2p \qquad BIC = -2logL(\hat{\theta}) + log(n)p$$

The AIC is a model selection method which punishes the amount of variables by adding two times the amount of variables to two times the negative log likelihood of $\hat{\theta}$, a lower value being the desired outcome. Considering our data sets great amount of observations (n = 21613) the punishment might be negligible. A better criterion might therefore be the BIC which penalizes the amount of variables by multiplying them with the log of the amount of observations. The BIC will therefore be our preferred metric of those two.

Given that we are developing a predictive model, the most important metric would still be the RMSE, the rooted mean square error, which is a measure of the mean prediction error:

$$RMSE = \left(\sum_{i=1}^{N} (y_{pi} - y_{oi})^2 \right)^{\frac{1}{2}}$$

with:

- y_{pi} being the i-th predicted response value
- y_{oi} being the i-th observed response value

This criterion gives us an actual predictive performance measurement to compare different models against each other in a meaningful an easy to interpret format, the amount of dollars for which a house was sold. To make this metric and its comparisons even more precise we will use two more different methods.

The first one will be a repeated cross-validation procedure using the k-fold strategy. For this method the data set will be randomly divided into k different folds. Leaving one fold out a model will be fitted against the remaining folds. The model will then attempt to predict the values of the fold previously left out and the RMSE will be calculated. This procedure will be repeated for all the different folds. This whole procedure of calculating the RMSE for all k folds will then again be repeated for B times. In our case we will use $k = 10$ and $B = 100$.

Using all those repeated measurements of the RMSE as a sample distribution we are now able to use the second method of comparison, a one sided two means t-test of the two RMSE distribution of two different model. This gives us a measure of confidence, a p-value, about the difference between those models.

The a priori data formating

It was necessary to do some initial formatting in order to make sense of the data for a linear model. The modifications are the following:

- transforming square feet into square meters
- converting waterfront, renovated and zipcode into factor variables
- splitting date into week and month of the year, removing the original date variable
- removing id because it is not informative
- splitting dataset into two parts, 80% and 20% (the reserved part, more on this later)

```r
full_data <- raw_data %>%
  mutate_at(
    vars(starts_with("sqft")),
    function(x) x * 0.092903
  ) %>%
  rename_at(
    vars(starts_with("sqft")),
    function(x) str_replace(x, "sqft", "sqm")
  ) %>%
  mutate(
    week_of_year = week(date),
    month_of_year = month(date),
    renovated = factor(ifelse(yr_renovated > 0, "yes", "no")),
    wasViewed = factor(ifelse(view > 0, 1, 0)),
    waterfront = factor(waterfront)
  ) %>%
  select(-id, -date, -starts_with("sqm"), starts_with("sqm"))
```

```r
first_fifth_avg <- data_20p %>%
  group_by(zipcode) %>%
  summarise(mean_price_zip = mean(price))

full_data <- full_data %>%
  left_join(., first_fifth_avg, key = zipcode) %>%
  mutate(
    price = log10(price),
    mean_price_zip = log10(mean_price_zip)
  )
```

The first model

In the first model only the useful original variables will be included with minimal formatting to provide a baseline for comparison.

```r
data_wo_new_vars <- full_data %>%
  select(-wasViewed, -renovated, -mean_price_zip)

model_1 <- lm(price ~ ., data = data_wo_new_vars)
summary(model_1)
```

```
##
## Call:
## lm(formula = price ~ ., data = data_wo_new_vars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79545 -0.06984  0.00137  0.06877  0.51844
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)   -1.0227807971  1.7821374985  -0.574             0.56604
## bedrooms      -0.0065796282  0.0011934235  -5.513    0.000000035733833 ***
## bathrooms      0.0304110205  0.0019792560  15.365 < 0.0000000000000002 ***
## floors         0.0334927857  0.0021716861  15.422 < 0.0000000000000002 ***
## waterfront1    0.1657845101  0.0104082177  15.928 < 0.0000000000000002 ***
## view           0.0269147354  0.0013000536  20.703 < 0.0000000000000002 ***
## condition      0.0276373112  0.0014200196  19.463 < 0.0000000000000002 ***
## grade          0.0678112380  0.0013095874  51.781 < 0.0000000000000002 ***
## yr_built      -0.0014655209  0.0000439524 -33.343 < 0.0000000000000002 ***
## yr_renovated   0.0000165792  0.0000022486   7.373    0.000000000000174 ***
## zipcode       -0.0002920170  0.0000200183 -14.588 < 0.0000000000000002 ***
## lat            0.6089333776  0.0065181396  93.421 < 0.0000000000000002 ***
## long          -0.0682539439  0.0079122055  -8.626 < 0.0000000000000002 ***
## week_of_year  -0.0018685591  0.0006489327  -2.879             0.00399 **
## month_of_year  0.0057473400  0.0028245154   2.035             0.04189 *
## sqm_living     0.0007156784  0.0000287739  24.872 < 0.0000000000000002 ***
## sqm_lot        0.0000021379  0.0000003206   6.668    0.000000000026751 ***
## sqm_above     -0.0000568279  0.0000285409  -1.991             0.04649 *
## sqm_basement             NA            NA      NA                   NA
## sqm_living15   0.0004488352  0.0000224319  20.009 < 0.0000000000000002 ***
## sqm_lot15     -0.0000012124  0.0000004785  -2.534             0.01130 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1092 on 17271 degrees of freedom
## Multiple R-squared:  0.7717, Adjusted R-squared:  0.7715
## F-statistic:  3073 on 19 and 17271 DF,  p-value: < 0.00000000000000022
```

There are two peculiarities we can observe. In the first place, we noticed the variable sqm_basement was not taken into account and consists solely of NAs. The second observation provides the reason for this behavior: "A singularity was discovered". This means that sqm_basement is a linear combination of two or more other variables, sqm_living and sqm_above in this case. The decision to remove those variables will be dealt with during the experimentation phase.

The experimentation procedure

Given that we now have both a baseline model and defined our model selection criteria, using a trial and error approach, we can start testing different approaches seeking to improve the model. Among those approaches we will attempt outlier detection, creation of new and meaningful variables, correlation analysis and different kinds of variable transformations. Let us start with outlier detection.

The outliers

Outliers, depending on their impact, may have a detrimental effect on the capacity of a model to effectively fit itself to the general patterns of the data and not just the present sample, which decreases its predictive ability. To find those divergent data points we will have a look at their leverage and the cook's distance.

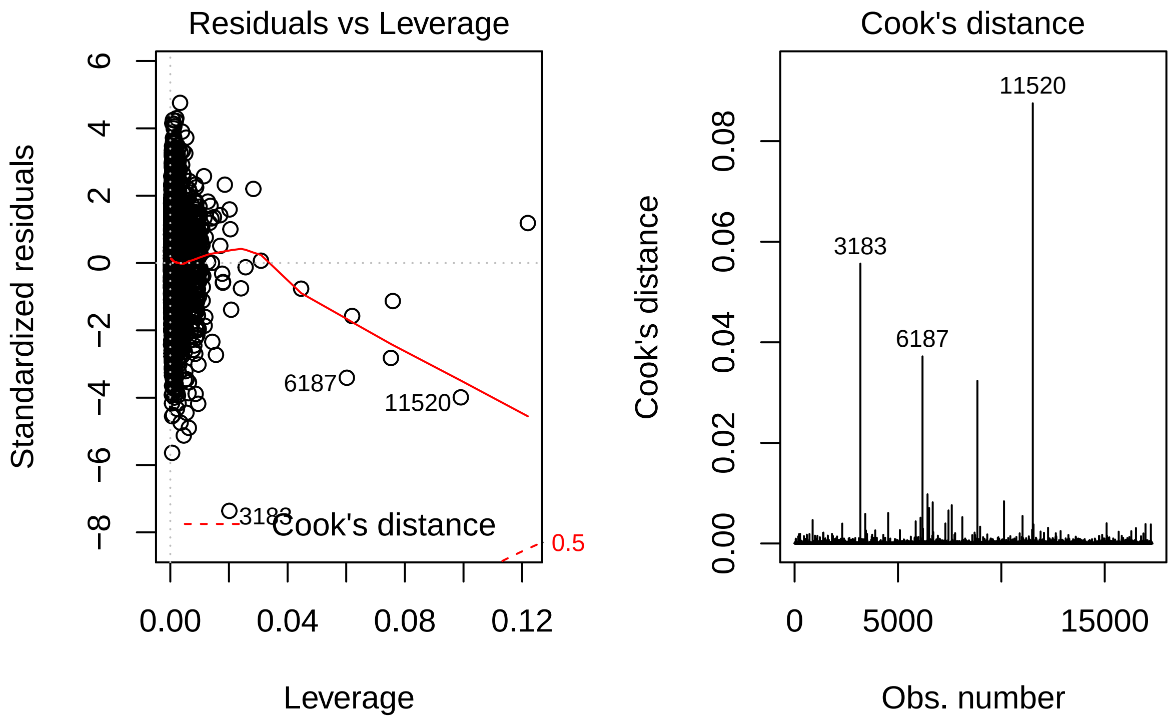
- Leverage: The power to shift the model towards that specific data point

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i'=1}^{n} (x_{i'} - \bar{x})^2}$$

- Cook's distance: Measures the aggregated influence of that observation on the fitted values

$$D_i = \frac{E_i^2}{k+1} \cdot \frac{h_i}{1-h_i}$$

```r
par(mfrow = c(1,2))
plot(model_1, which=5)
plot(model_1, which=4)
```



Inspecting the cook's distance plot we find that there are 4 highly influential data points. For the following test we will fit a linear model without these aforementioned observations and compare it to our baseline model. Note that in the following tables the p-value refers to a one sided t-test of the model it is next to with the model directly above.

```
full_data_woo <- full_data %>%
  #select(-wasViewed, -renovated, -mean_price_zip) %>%
  slice(c(-11520, -3183, -6187, -8844))

cross_val <- function(data, B = 100, k = 10){
  n <- nrow(data)
  folds <- sample(rep(c(1:k), length.out = nrow(data)))
  results <- foreach(icount(B), .combine = cbind) %dopar% {
    res <- numeric(k)
    for (i in 1:k) {
      training <- data[folds != i,]
      test <- data[folds == i,]

      lmo <- lm(data = training, price ~ .)
      predict <- predict(lmo, test)

      res[i] <- sum((10^test$price - 10^predict)^2)/length(test$price)
    }
    res
  }
  m_rmse <- mean(sqrt(results))
  sd_rmse <- sd(sqrt(results))
  l <- list(distr = sqrt(results), m_rmse = m_rmse, sd_rmse = sd_rmse)
  return(l)
}
```

```r
intermediate_1 <- full_data_woo %>%
  select(-wasViewed, -renovated, -mean_price_zip)

intermediate_2 <- data_wo_new_vars

model_2 <- intermediate_1 %>%
  lm(data = . , price ~ .)

cv_2 <- cross_val(intermediate_1)
cv_1 <- cross_val(intermediate_2)

p_value_1 <-
  tidy(t.test(cv_1$distr, cv_2$distr, var.equal = F, alternative = "greater")) %>%
  select(p.value) %>%
  unlist() %>%
  formatC(format = "e") %>%
  rbind(" ",.)

Mean_RMSE <- round(rbind(cv_1$m_rmse, cv_2$m_rmse), 0)
SD_RMSE <- round(rbind(cv_1$sd_rmse, cv_2$sd_rmse), 0)

cbind(BIC(model_1, model_2), Mean_RMSE, SD_RMSE, p_value_1) %>%
  kable() %>%
  kable_styling()
```

	df	BIC	Mean_RMSE	SD_RMSE	p.value
model_1	21	-27338.06	203777	56684	
model_2	21	-27420.96	186945	30079	1.1816e-16

Out of the resulting summary table, we can draw some conclusions regarding the removal of these outliers:

- By solely looking at the three first metrics (AIC, mean of RMSE and s.d. of RMSE) we can see that removal of outliers proved to be successful.
- When looking only at the mean RMSE, we observe it was reduced by about 20.000$. This, in terms of usefulness as a prediction tool in real life scenarios, is very desirable, given that it provides the end user with a more accurate estimation of the actual values.
- The standard deviation of RMSE errors was cut almost by half, meaning that the new models are way more stable.
- The last metric of the table, the p-value, verifies the significance of the conclusions made above. Since the resulting number is very low (highly significant), we proceed to conclude with enough confidence (>99%) that the previous statements are valid.
- The BIC was reduced by 83.

In terms of data points (observations) there will be no further additions or removals from now on. So, the next phases of experimentation, are rather column-wise, and focus on the creation, deletion or edition of the variables as a whole.

The creating of new variables

In this phase we decided to create some additional variables by using some the reserved part of the data in the current ones or by creating a simplified version of an existing one.

The “Average price by zipcode” variable:

To understand the creation process of this variable and why it is a valid thing to do, it is necessary to explain the 80%-20% data splitting mentioned before in the *Preliminary data formating* section.

It goes as follows: Since the split of the data, made at the very beginning, we have only worked with the 80% chunk, which comes to represent the only usable data for our experiments, model comparison, conclusion making and everything else. Given this, we could see the rest of the 20% data as “preconceived”. By doing so, we intend to pretend as if the information (including the sale prices) provided by this usable set of observations is available throughout all the study. It is not considered data leakage since the rows themselves are not being used to train or evaluate any model at any point.

So, how are we utilizing this data exactly? By using them as a summary of prices by zipcode. First we select the zipcode and the price columns in this 20% chunk, then we group the observations by their zipcode and compute the average price within each zipcode. Finally in the 80% data chunk, we create a new column containing the average price found in the corresponding zipcode-price pairs obtained with the 20% chunk. In this way, we have obtained additional data to the model, which is related to the prices, without falling prey to data leakage.

```
intermediate_3 <- full_data_woo %>%
  select(-wasViewed, -renovated)

model_3 <- lm(price ~ ., data = intermediate_3)

cv_3 <- cross_val(intermediate_3)

p_value_2 <-
  tidy(t.test(cv_2$distr, cv_3$distr, var.equal = F, alternative = "greater")) %>%
  select(p.value) %>%
  unlist() %>%
  formatC(format = "e")

Mean_RMSE <- round(rbind(cv_2$m_rmse, cv_3$m_rmse))
SD_RMSE <- round(rbind(cv_2$sd_rmse, cv_3$sd_rmse))
p_vals <- rbind(" ", p_value_2)

cbind(BIC(model_2, model_3), Mean_RMSE, SD_RMSE, p_vals) %>%
  kable() %>%
  kable_styling()
```

	df	BIC	Mean_RMSE	SD_RMSE	p.value
model_2	21	-27420.96	186945	30079	
model_3	22	-35341.87	155384	35346	1.6393e-92

Once again, let us analyze the resulting summary:

- In terms of our comparison criteria, introducing this new variables improves the model.
- The mean RMSE is reduced by about 30.000$
- The standard deviation of the RMSE remains similar to the previous one. So adding this variable does not destabilizes the model considerably.
- Once again, the p-value obtained suggests that the difference in the distribution of RMSEs is highly significant when using this additional variable.
- Considerably large improvement in the BIC metric.

The factorization:

The second and third additional columns are less complicated, being only a simplification of two current columns. The general reasoning is rather than using all the values provided in the column, we are only concerned on whether or not there is a non-zero value present. The specific transformations are:

- From the *view* variable, create a "yes or no" variable indicating if the properties has been viewed or not.
- From the *year renovated* variable, create a variable that specifies if the property has ever been renovated or not.

For the following experiment, we will test if adding the transformed variables will improve the model as well as whether we should keep the untransformed original ones on which the new ones are based or not. Note that in the following output the p-value always refers to a comparison against the uppermost model within the table.

```
intermediate_3.1 <- full_data_woo
model_3.1 <- lm(price ~ ., data = intermediate_3.1)

cv_3.1 <- cross_val(intermediate_3.1)

p_value_2.1 <-
  tidy(t.test(cv_3$distr, cv_3.1$distr, var.equal = F, alternative = "greater")) %>%
  select(p.value) %>%
  unlist() %>%
  round(., 2)

intermediate_3.2 <- full_data_woo %>%
  select(-view, -yr_renovated)

model_3.2 <- lm(price ~ ., data = intermediate_3.2)

cv_3.2 <- cross_val(intermediate_3.2)

p_value_2.2 <-
  tidy(t.test(cv_3$distr, cv_3.2$distr, var.equal = F, alternative = "greater")) %>%
  select(p.value) %>%
  unlist() %>%
  round(., 2)

Mean_RMSE <- round(rbind(cv_3$m_rmse, cv_3.1$m_rmse, cv_3.2$m_rmse))
SD_RMSE <- round(rbind(cv_3$sd_rmse, cv_3.1$sd_rmse, cv_3.2$sd_rmse))
p_vals <- rbind(" ", p_value_2.1, p_value_2.2)

cbind(BIC(model_3, model_3.1, model_3.2), Mean_RMSE, SD_RMSE, p_vals) %>%
  kable() %>%
  kable_styling()
```

	df	BIC	Mean_RMSE	SD_RMSE	p.value
model_3	22	-35341.87	155384	35346	
model_3.1	24	-35391.64	154127	28181	0.19
model_3.2	22	-35296.07	155212	25355	0.45

As the tests reveal, keeping the factorized variables while removing the variables they are based on (model 3.2) proved to be of no improvement at all. If anything it decreases the accuracy in terms of the BIC.

On the other hand factorizing while keeping the original variables (model 3.1) improves the model performance slightly as indicated by the drop in BIC of about 50 points. The mean RMSE was reduced to some extent as well, although the p-value indicates a low chance of this difference being significant.

Now that we have attempted and tested creations of new variables, we'll proceed with correlation analysis in the next section. Although our experiments so far suggest that at least one of the recently added variables will stay present in the final model, given the improvements obtained when adding the three of them, they are still subject to testing.

The correlation analysis

In the following table, we present the correlation coefficients between each of the predictors and the target variable (price). The idea is to determine how linearly related are each of those variables to the response and based on this, do a sub selection of the most correlated ones to determine if there is an improvement.

month_of_year	0.0174	sqm_lot	0.0962	bathrooms	0.5508
week_of_year	0.0197	yr_renovated	0.1074	sqm_above	0.602
zipcode	0.0372	floors	0.314	sqm_living15	0.6187
condition	0.0396	sqm_basement	0.3191	sqm_living	0.6945
long	0.0485	bedrooms	0.3489	grade	0.7022
yr_built	0.0826	view	0.3504	mean_price_zip	0.7172
sqm_lot15	0.0909	lat	0.4482	price	1

Looking at the table, we suggest the following hypothesis: Given that we observe a big jump in correlation from 0.107 to 0.314 in between the variable yr_renovated and floor, dropping all variables with a correlation below 0.314 might decrease prediction error and improve the model.

```r
intermediate_4 <- full_data_woo %>%
  dplyr::select(one_of(good_corr), waterfront)

model_4 <- lm(price ~ ., data = intermediate_4)

cv_4 <- cross_val(intermediate_4)

Mean_RMSE <- round(rbind(cv_3$m_rmse, cv_4$m_rmse), 0)
SD_RMSE <- round(rbind(cv_3$sd_rmse, cv_4$sd_rmse), 0)

p_value_3 <-
  tidy(t.test(cv_3$distr, cv_4$distr, var.equal = F, alternative = "greater")) %>%
  select(p.value) %>%
  unlist() %>%
  round(., 2) %>%
  rbind(" ",.)

cbind(BIC(model_3, model_4), Mean_RMSE, SD_RMSE, p_value_3) %>%
  kable() %>%
  kable_styling()
```

	df	BIC	Mean_RMSE	SD_RMSE	p.value
model_3	22	-35341.87	155384	35346	
model_4	13	-33124.77	159525	23983	1

As we can see, although the variability of the predictions is reduced, the actual mean of the RMSEs increases, which is not a desirable result. This together with the huge increase of the BIC, dropping the less correlated variables decreases the model performance. Looking at the p-value, we realize that there is no evidence at all that the reduced model performs better.

The variable selection

As we saw earlier the variable sqm_living is a linear combination of sqm_above and sqm_basement. Thus it might be reasonable to select the best performing model with a certain subset of those variables. To figure that out we first tried to find the best transformation of the all of those variables under the assumption that doing so might improve the RMSE. We used an array of different transformations for each variable separately and evaluated the impact through the change in the BIC compared to a model containing the untransformed variable.

```r
intermediate_5 <- full_data_woo %>%
  mutate(
    sqm_above = sqrt(sqm_above),
    sqm_basement = sqrt(sqm_basement)
    ) %>%
  dplyr::select(-sqm_living)

model_5 <- lm(data = intermediate_5, price ~ .)

intermediate_6 <- full_data_woo %>%
  mutate_at(vars(sqm_living), sqrt) %>%
  dplyr::select(-sqm_above, -sqm_basement)

model_6 <- lm(data = intermediate_6, price ~ .)

s_model_5 <- summary(model_5)
s_model_6 <- summary(model_6)

cv_5 <- cross_val(intermediate_5)
cv_6 <- cross_val(intermediate_6)

p_value_4 <- tidy(t.test(cv_5$distr, cv_6$distr, var.equal = F)) %>%
  select(p.value) %>%
  unlist()

row_names <- c("above & basement", "living")
col_names <- c("R^2 adjust.", "AIC", "BIC", "Mean_RMSE", "SD_RMSE", "P-value")
a <- matrix(
  c(
    round(s_model_5$adj.r.squared, 4),
    round(AIC(model_5), 0),
    round(BIC(model_5), 0),
    round(cv_5$m_rmse, 0),
    round(cv_5$sd_rmse, 0),
    " ",
    round(s_model_6$adj.r.squared, 4),
    round(AIC(model_6), 0),
    round(BIC(model_6), 0),
    round(cv_6$m_rmse, 0),
    round(cv_6$sd_rmse, 0),
    round(p_value_4, 2)),
  ncol = 6,
  byrow = T,
  dimnames = list(row_names, col_names)
)

a %>%
  kable() %>%
  kable_styling()
```

	R^2 adjust.	AIC	BIC	Mean_RMSE	SD_RMSE	P-value
above & basement	0.8606	-36049	-35863	133703	9328	
living	0.86	-35976	-35798	134038	6253	0.35

Comparing both models with each other the result is quite inconclusive. On the one hand, the p-value indicates that there is no significant difference between their distributions of RMSEs. According to the principle of parsimony one should stick with the second model, containing sqm_living instead of sqm_basement and sqm_above, due to a reduced amount of variables.

However the BIC, AIC and the adjusted R^2 suggest the first model to be the more precise one and that's why we have chosen to stick to this particular model.

The reduced model

We are now going to check if the model can be improved by dropping some combination of variables by using a step-wise selection approach based on the BIC as the selection metric.

```r
intermediate_reduced <- full_data_woo %>%
  select(-zipcode, -sqm_lot15, -month_of_year, -wasViewed)

cv_reduced <- cross_val(intermediate_reduced)

p_value_red <-
  tidy(
    t.test(cv_5$distr, cv_reduced$distr, var.equal = F, alternative = "greater")) %>%
  select(p.value) %>%
  unlist() %>%
  formatC(format = "e") %>%
  rbind(" ",.)

Mean_RMSE <- round(rbind(cv_5$m_rmse, cv_reduced$m_rmse))
SD_RMSE <- round(rbind(cv_5$sd_rmse, cv_reduced$sd_rmse))


cbind(BIC(model_5, reduced_model), Mean_RMSE, SD_RMSE, p_value_red) %>%
  kable() %>%
  kable_styling()
```

	df	BIC	Mean_RMSE	SD_RMSE	p.value
model_5	24	-35862.85	133703	9328	
reduced_model	20	-35881.30	155089	27968	1.0000e+00

Removing certain variables was advised by the BIC. Using the step-wise selection approach zipcode, sqm_lot15, month_of_year and wasViewed were removed. Testing this model against the previous one yielded disagreeing results. Although the BIC improved slightly, the mean RMSE increased considerably as well as the standard deviation of the mean RMSE. We will therefore stick to the previous complete model.

The transformations

Applying the previously mentioned method of trying different transformations on variables we are now going to use it for all the other variables as well. We will then compare this last model with the untransformed previous one as well as the very first model to have an overview of progress we made as well as to provide us with a last comparison to draw our final conclusion.

```r
int_col_names <- full_data_woo %>%
  select_if(function(col) is.integer(col) | is.double(col)) %>%
  select(-long) %>%
  colnames()

results <- character(length(int_col_names)-1)
names(results) <- int_col_names[-1]

for (i in 2:length(int_col_names)) {
  no_trans <- full_data_woo %>%
  lm(price ~ ., data = .)

  square_rt <- full_data_woo %>%
    modify_at(int_col_names[i], sqrt) %>%
    lm(price ~ ., data = .)

  loga <- full_data_woo %>%
    modify_at(int_col_names[i], function(x) log_x(x, a = 1)$x.t) %>%
    lm(price ~ ., data = .)

  power_2 <- full_data_woo %>%
    modify_at(int_col_names[i], function(x) x^2) %>%
    lm(price ~ ., data = .)

  power_3 <- full_data_woo %>%
    modify_at(int_col_names[i], function(x) x^3) %>%
    lm(price ~ ., data = .)

  power_4 <- full_data_woo %>%
    modify_at(int_col_names[i], function(x) x^4) %>%
    lm(price ~ ., data = .)

  power_5 <- full_data_woo %>%
    modify_at(int_col_names[i], function(x) x^5) %>%
    lm(price ~ ., data = .)

  power_6 <- full_data_woo %>%
    modify_at(int_col_names[i], function(x) x^6) %>%
    lm(price ~ ., data = .)

  power_10 <- full_data_woo %>%
    modify_at(int_col_names[i], function(x) x^10) %>%
    lm(price ~ ., data = .)

  bics <- BIC(
    no_trans, square_rt, loga, power_2, power_3,
    power_4, power_5, power_6, power_10
  )
  bics_names <-c(
    "no_trans", "square_rt", "loga", "power_2",
    "power_3", "power_4", "power_5", "power_6", "power_10"
  )
  results[i - 1] <- bics_names[which(bics[, 2] == min(bics[, 2]))]}

transformed_data <- full_data_woo %>%
  select(-sqm_living) %>%
  mutate(
    sqm_above = sqrt(sqm_above),
    sqm_basement = sqrt(sqm_basement),
    floors = log10(floors + 1),
    bathrooms = log10(bathrooms + 1),
    sqm_living15 = log10(sqm_living15 + 1),
    mean_price_zip = log10(mean_price_zip + 1),
    grade = log10(grade + 1),
    condition = log10(condition + 1),
    yr_built = log10(yr_built + 1),
    lat = log10(lat + 1),
    week_of_year = log10(week_of_year + 1),
    month_of_year = log10(month_of_year + 1),
    sqm_lot15 = log10(sqm_lot15 + 1),
    bedrooms = (bedrooms)^4,
    view = (view)^4,
    yr_renovated = (yr_renovated)^10,
    zipcode = (zipcode)^10
  )
```

```
model_7 <- transformed_data %>%
  lm(data = ., price ~ .)

cv_7 <- cross_val(transformed_data)

p_value_4 <-
  tidy(t.test(cv_1$distr, cv_5$distr, var.equal = F, alternative = "greater")) %>%
  select(p.value) %>%
  unlist() %>%
  formatC(format = "e")

p_value_5 <-
  tidy(t.test(cv_5$distr, cv_7$distr, var.equal = F, alternative = "greater")) %>%
  select(p.value) %>%
  unlist() %>%
  round(., digits = 2)


Mean_RMSE <- rbind(cv_1$m_rmse, cv_5$m_rmse, cv_7$m_rmse)
```

```r
SD_RMSE <- rbind(cv_1$sd_rmse, cv_5$sd_rmse, cv_7$sd_rmse)
p_values <- c(" ", p_value_4, p_value_5)
stopCluster(cl)


cbind(BIC(model_1, model_5, model_7), Mean_RMSE, SD_RMSE, p_values) %>%
  kable() %>%
  kable_styling()
```

	df	BIC	Mean_RMSE	SD_RMSE	p_values
model_1	21	-27338.06	203777.1	56683.881	
model_5	24	-35862.85	133703.5	9328.486	6.0291e-204
model_7	24	-35920.63	134460.4	5657.831	0.99

The conclusion

Our best model in the end turned out to be model 5. Its mean RMSE decreased by 70.000$, the standard deviation of the RMSE decreased to one fifth and the BIC dropped by about 8.500 compared to the very first model. We achieved this by taking the log of price, our response variable, removing the most extreme outliers defined by the cook's distance, adding the mean price per zipcode variable and factorized versions of some variables, resolving the singularity issue and transforming sqm_above and sqm_basement. All the other methods we tried to improve our model turned out to be either ineffective or yielded a worse model than before.

In some occasions we noticed disagreements within our model selection criteria, namely BIC and AIC against the RMSE. Some methods improved the BIC slightly or even significantly but at the same the raised the mean RMSE slightly or even substantially. In our opinion the BIC proved to be an ineffective predictor compared to the RMSE. Thus it might be advisable to use a cross validation method and the subsequent calculation of the mean RMSE at every step along the way if possible and if computational power and time is sufficiently available.

Despite our attempts at improving the model through correlation analysis, step-wise BIC reduction, variable transformations and using the BIC as a selection metric, desired results were not achieved by that. This failure of producing favorable outcomes does not translate into those methods not being successfully applicable in different settings and with different types of data sets.