UCLouvain
LSTAT2120 : Linear Models
Academic year 2019-2020
Project: House price prediction

Richard Delava
Matis Brassard-Verrier

January 3rd, 2019

Contents

1 Introduction **2**
1.1 Problem statement . . . 2
1.2 Presentation of chosen dataset . . . 2

2 Variables **2**
2.1 Variable interpretation . . . 2
2.2 General analysis . . . 3
2.3 Correlation matrix . . . 5

3 Model selection **6**
3.1 Feature engineering . . . 6
3.2 Hypothesis testing . . . 8
3.2.1 Multicollinearity . . . 8
3.2.2 Heteroskedasticity . . . 9
3.2.3 Autocorrelation . . . 9
3.2.4 Outliers . . . 9
3.3 Variable selection . . . 11
3.3.1 Stepwise regression . . . 11
3.3.2 LASSO regression . . . 11
3.4 Results . . . 12
3.5 Model interpretation . . . 12
3.5.1 Linear combination of coefficients . . . 14
3.6 Prediction interval . . . 15

4 Conclusion **15**

A APPENDIX **16**
A.1 R code . . . 16

1 Introduction

1.1 Problem statement

The general goal behind this project is to apply linear models knowledge to a real-life situation represented in a dataset. In other words, we have to adapt linear model selection strategies to a dataset in order to get a final model that is as precise as possible. The following general steps will be followed in this project:

- Proceeding to explanatory variable selection;
- Testing the underlying hypotheses (e.g. no multicollinearity) and proceeding to adjustments if they are not respected;
- Fitting a final linear model to the data;
- Testing the quality of this model;
- Interpreting the coefficients obtained with this model;
- Computing a prediction interval for one arbitrarily selected observation using this model.

It is therefore important to choose a dataset that is well suitable for linear model fitting.

1.2 Presentation of chosen dataset

The dataset that is we have chosen was found on Kaggle and is titled *House Sales in King County, USA*. It contains the house sales prices of homes in King County in the state of Washington during a one-year period, going from May 2014 to May 2015. Along with the house sale price, which is the response variable we will aim to predict, every data entry contains multiple characteristics of the house that was sold, which are the explanatory variables that we will use. These explanatory variables will be presented in the next section.

2 Variables

In this section, we will provide a quick descriptive analysis of the variables.

2.1 Variable interpretation

Let us start with a quick presentation of all the available explanatory variables and how to interpret their values:

Name	**Description**
ID	Unique ID for each home sold
Date	Date of the home sale
Bedrooms	Number of bedrooms
Bathrooms	Number of bathrooms (0.5 is for a toilet without shower)
Sqft_living	Square footage of the living space
Sqft_lot	Square footage of the lot area
Floors	Number of floors
Waterfront	Dummy variable 'seaview'. Indicates if house is along a body of water.
View	Index from 1 to 4 about the quality of the view
Condition	Index from 1 to 5 about the condition of the house
Grade	Index from 1 to 13 about quality and design of the construction
Sqft_above	Square footage of the living space above ground level
Sqft_basement	Square footage of the basement (below ground level)
Yr_build	The year of construction
Yr_renovated	Year of renovation (0 if none)
Zipcode	Zipcode of the area
Lat	Latitude of the house
Long	Longitude of the house
Sqft_living15	Average square footage of the living space of the 15 nearest neighbors
Sqft_lot15	Average square footage of the lot area of the 15 nearest neighbors
Price	Sale price of the home

2.2 General analysis

First, let us see a table containing the mean, standard deviation, skewness and kurtosis of the available explanatory variables. From this moment, we will remove the ID and the date for the rest of the analysis, as we see no use to those explanatory variables.

	Name	min	Median	Mean	Std. dev.	Max	Skewness	Kurtosis
1	price	75000.00	450000.00	540088.14	367127.20	7700000.00	4.02	37.58
2	bedrooms	0.00	3.00	3.37	0.93	33.00	1.97	52.05
3	bathrooms	0.00	2.25	2.11	0.77	8.00	0.51	4.28
4	sqft_living	290.00	1910.00	2079.90	918.44	13540.00	1.47	8.24
5	sqft_lot	520.00	7618.00	15106.97	41420.51	1651359.00	13.06	288.01
6	floors	1.00	1.50	1.49	0.54	3.50	0.62	2.52
7	waterfront	0.00	0.00	0.01	0.09	1.00	11.38	130.60
8	view	0.00	0.00	0.23	0.77	4.00	3.40	13.89
9	condition	1.00	3.00	3.41	0.65	5.00	1.03	3.53
10	grade	1.00	7.00	7.66	1.18	13.00	0.77	4.19
11	sqft_above	290.00	1560.00	1788.39	828.09	9410.00	1.45	6.40
12	sqft_basement	0.00	0.00	291.51	442.58	4820.00	1.58	5.71
13	yr_built	1900.00	1975.00	1971.01	29.37	2015.00	-0.47	2.34
14	yr_renovated	0.00	0.00	84.40	401.68	2015.00	4.55	21.70
15	zipcode	98001.00	98065.00	98077.94	53.51	98199.00	0.41	2.15
16	lat	47.16	47.57	47.56	0.14	47.78	-0.49	2.32
17	long	-122.52	-122.23	-122.21	0.14	-121.31	0.88	4.05
18	sqft_living15	399.00	1840.00	1986.55	685.39	6210.00	1.11	4.60
19	sqft_lot15	651.00	7620.00	12768.46	27304.18	871200.00	9.51	153.73

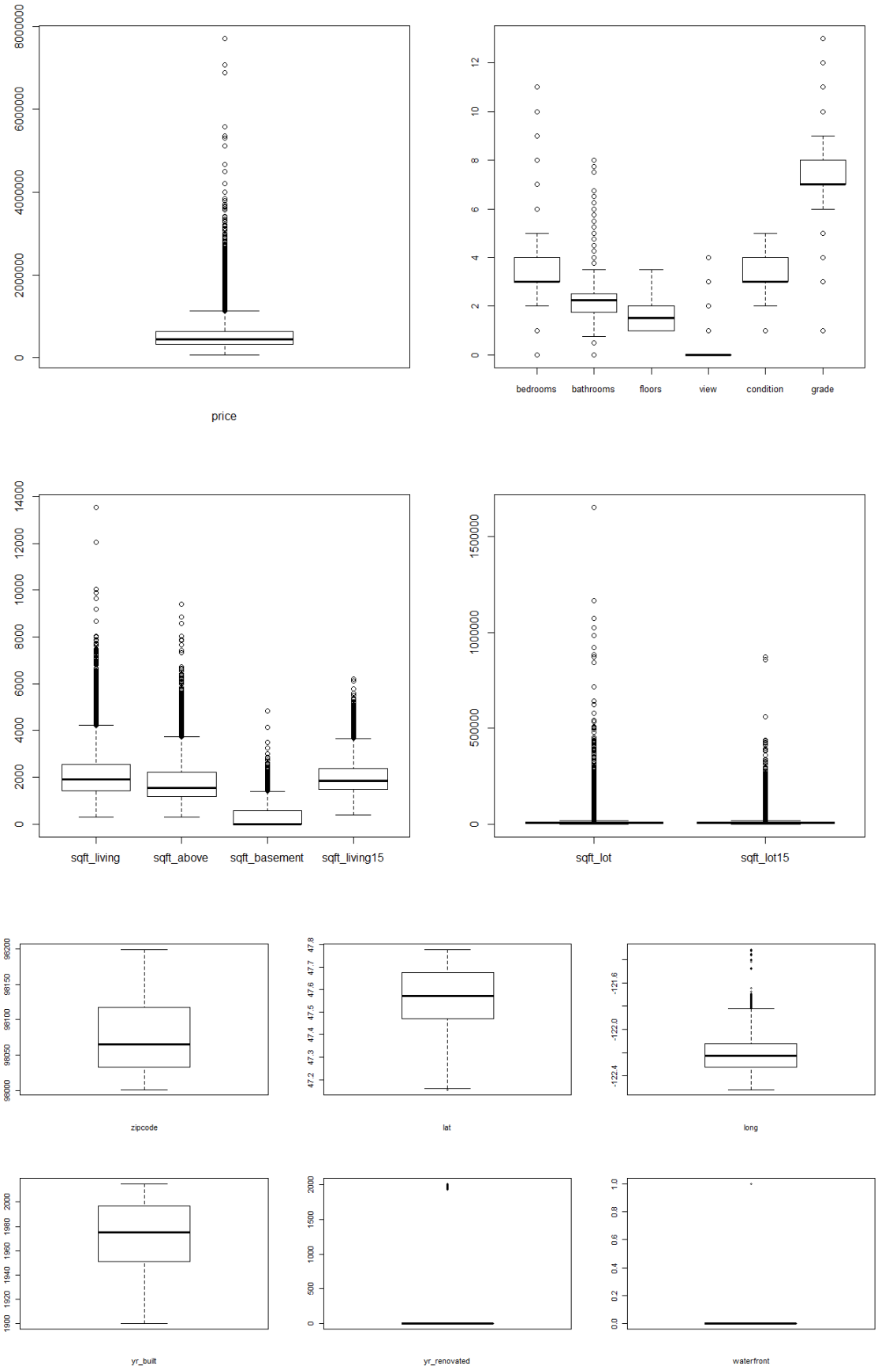

Figure 1: Boxplots of the raw (unmodified) explanatory variables

2.3 Correlation matrix

We can end the analysis of the variables with the correlation matrix of the explanatory variables.

	price	beds	baths	liv	lot	flrs	water	view	cond.	grd	above	bsmt	yr_b	yr_r	zip	lat	long	liv15	lt15
price	1.00	0.31	0.53	0.70	0.09	0.26	0.27	0.40	0.04	0.67	0.61	0.32	0.05	0.13	-0.05	0.31	0.02	0.59	0.08
beds	0.31	1.00	0.52	0.58	0.03	0.18	-0.01	0.08	0.03	0.36	0.48	0.30	0.15	0.02	-0.15	-0.01	0.13	0.39	0.03
bathss	0.53	0.52	1.00	0.75	0.09	0.50	0.06	0.19	-0.12	0.66	0.69	0.28	0.51	0.05	-0.20	0.02	0.22	0.57	0.09
sqft_living	0.70	0.58	0.75	1.00	0.17	0.35	0.10	0.28	-0.06	0.76	0.88	0.44	0.32	0.06	-0.20	0.05	0.24	0.76	0.18
sqft_lot	0.09	0.03	0.09	0.17	1.00	-0.01	0.02	0.07	-0.01	0.11	0.18	0.02	0.05	0.01	-0.13	-0.09	0.23	0.14	0.72
floors	0.26	0.18	0.50	0.35	-0.01	1.00	0.02	0.03	-0.26	0.46	0.52	-0.25	0.49	0.01	-0.06	0.05	0.13	0.28	-0.01
water	0.27	-0.01	0.06	0.10	0.02	0.02	1.00	0.40	0.02	0.08	0.07	0.08	-0.03	0.09	0.03	-0.01	-0.04	0.09	0.03
view	0.40	0.08	0.19	0.28	0.07	0.03	0.40	1.00	0.05	0.25	0.17	0.28	-0.05	0.10	0.08	0.01	-0.08	0.28	0.07
cond.	0.04	0.03	-0.12	-0.06	-0.01	-0.26	0.02	0.05	1.00	-0.14	-0.16	0.17	-0.36	-0.06	0.00	-0.01	-0.11	-0.09	-0.00
grade	0.67	0.36	0.66	0.76	0.11	0.46	0.08	0.25	-0.14	1.00	0.76	0.17	0.45	0.01	-0.18	0.11	0.20	0.71	0.12
sqft_above	0.61	0.48	0.69	0.88	0.18	0.52	0.07	0.17	-0.16	0.76	1.00	-0.05	0.42	0.02	-0.26	-0.00	0.34	0.73	0.19
sqft_bst	0.32	0.30	0.28	0.44	0.02	-0.25	0.08	0.28	0.17	0.17	-0.05	1.00	-0.13	0.07	0.07	0.11	-0.14	0.20	0.02
yr_built	0.05	0.15	0.51	0.32	0.05	0.49	-0.03	-0.05	-0.36	0.45	0.42	-0.13	1.00	-0.22	-0.35	-0.15	0.41	0.33	0.07
yr_renov	0.13	0.02	0.05	0.06	0.01	0.01	0.09	0.10	-0.06	0.01	0.02	0.07	-0.22	1.00	0.06	0.03	-0.07	-0.00	0.01
zip	-0.05	-0.15	-0.20	-0.20	-0.13	-0.06	0.03	0.08	0.00	-0.18	-0.26	0.07	-0.35	0.06	1.00	0.27	-0.56	-0.28	-0.15
lat	0.31	-0.01	0.02	0.05	-0.09	0.05	-0.01	0.01	-0.01	0.11	-0.00	0.11	-0.15	0.03	0.27	1.00	-0.14	0.05	-0.09
long	0.02	0.13	0.22	0.24	0.23	0.13	-0.04	-0.08	-0.11	0.20	0.34	-0.14	0.41	-0.07	-0.56	-0.14	1.00	0.33	0.25
sqft_living15	0.59	0.39	0.57	0.76	0.14	0.28	0.09	0.28	-0.09	0.71	0.73	0.20	0.33	-0.00	-0.28	0.05	0.33	1.00	0.18
sqft_lot15	0.08	0.03	0.09	0.18	0.72	-0.01	0.03	0.07	-0.00	0.12	0.19	0.02	0.07	0.01	-0.15	-0.09	0.25	0.18	1.00

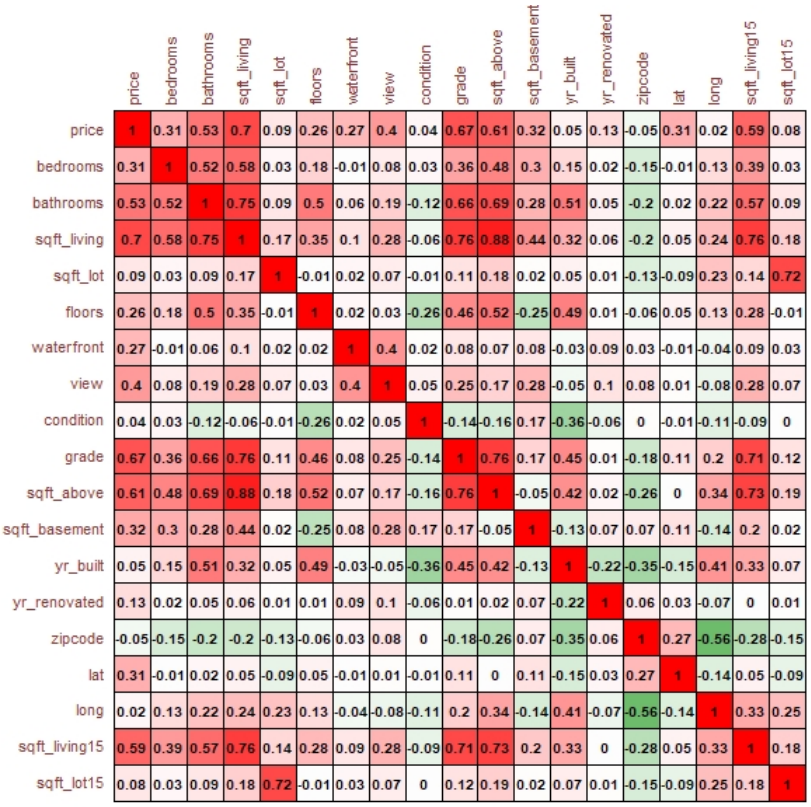

Figure 2: Correlation matrix: score between (-1,1)

This matrix will be used later for multicollinearity analysis.

3 Model selection

Select an adequate model for the response variable by considering all quantitative and qualitative variables and by using our model selection strategies. Consider also possible interactions of the qualitative variables with one or several quantitative variables. Verify the underlying hypotheses and, if necessary, take remedial actions. For example, check whether there are outliers and/or influential observations, multicollinearity, heteroskedasticty, and autocorrelation. If necessary, try to improve the model by using the methods seen in class.

3.1 Feature engineering

Before selecting the variables to be used for the model, we start by modifying some of them. By looking at the histograms and boxplots above, we can see that the distributions of some of the variables including the explanatory variable 'price' are heavily skewed. To solve this issue and have better performance with the future linear model, we simply take the log of these variables.

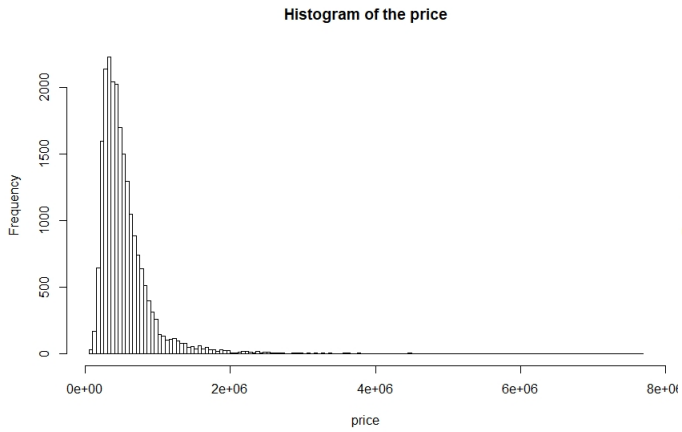

Figure 3: Histogram of the price

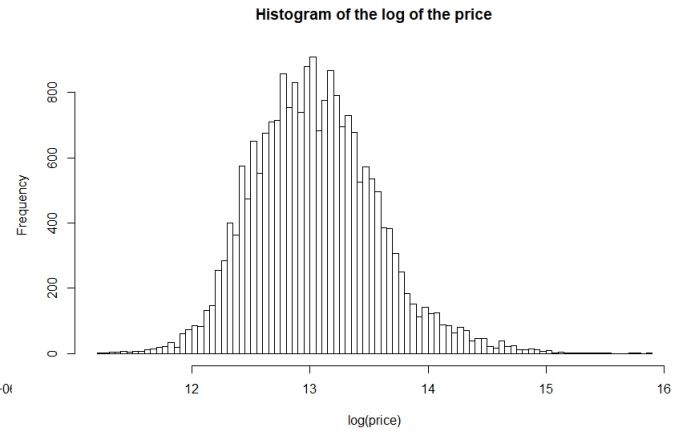

Figure 4: Histogram of the log of the price

Here is the example for the Price. By taking the log, the distribution becomes closer to a normal distribution. We did the same with the sqft_living and the sqft_lot which have the same problem/distribution.

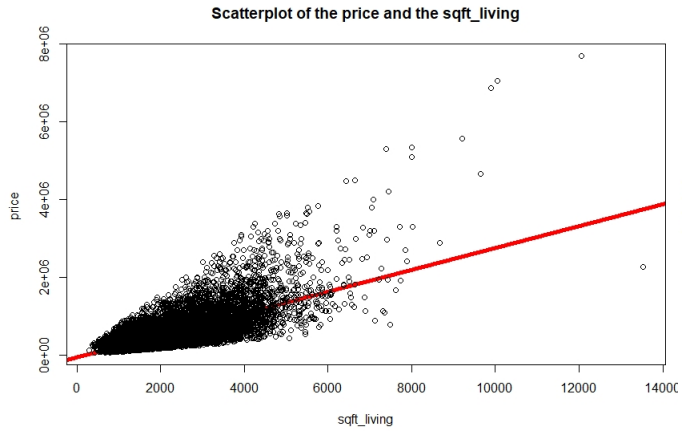

Figure 5: Price / Sqft_living

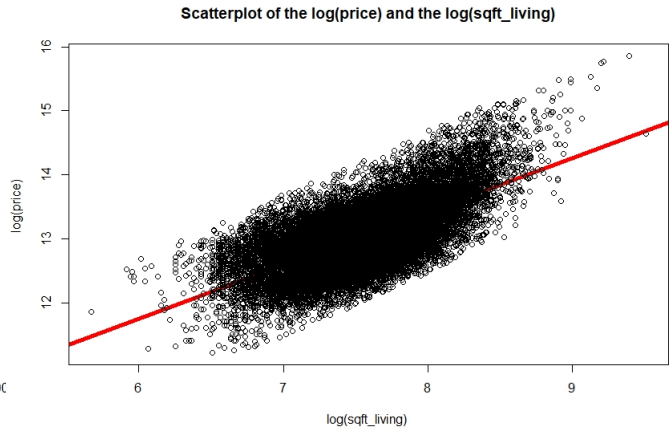

Figure 6: log(Price) / log(Sqft_living)

In addition to that, as you can see in the figure 5 and 6. It also helps to solve the problem of heteroskedasticity.

After that, as a linear model can only work with numerical variables, we need to take care of the qualitative variables.

Zipcode, latitude and longitude: We have built a heatmap of the log of the sale price according to the position using latitude and longitude. Looking at the map, we can see that there are two distinguishable areas: the expensive

area, especially in the Seattle-Bellevue area, and the cheaper surroundings. For simplicity of the model and for interpretability, instead of using the zipcodes or coordinates, we will create a dummy variable to mention if the house is in the expensive area or not. It is rather hard to determine a shape for the expensive area, so we decided to use the following approximation: all houses north of the 47.52° latitude line (the dashed line on the heatmap) are considered to be part of the expensive area, and those south of the line are considered to be part of the cheaper surroundings. This allowed us to create the dummy variable "expensive_area", which takes the value 0 or 1 to indicate if the house is in the expensive area or not.

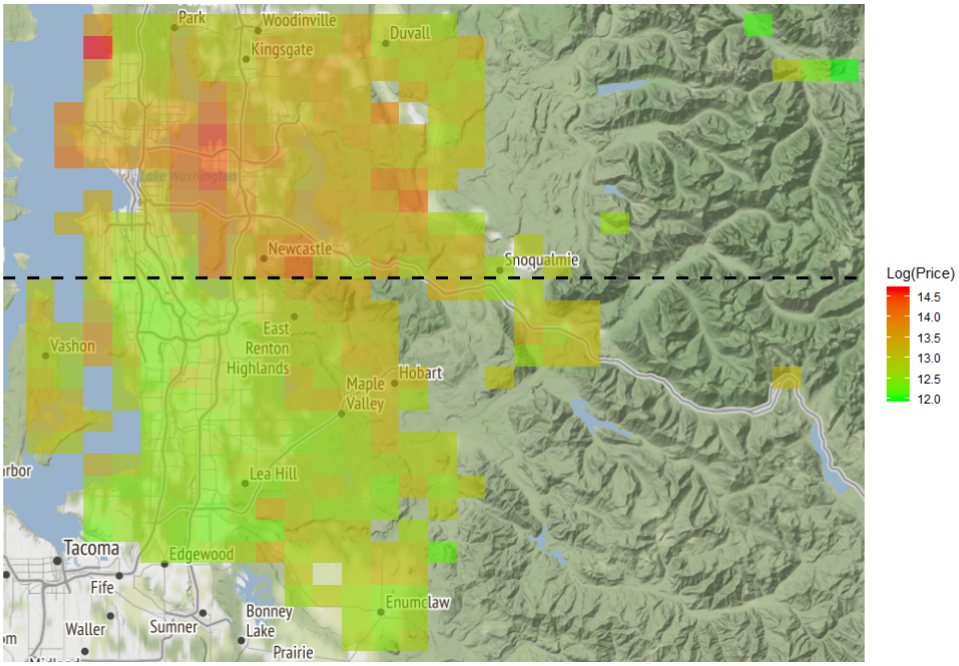

Figure 7: Heatmap of log of sale prices

Yr_built: This variable goes from 1900 to 2015, so its coefficient in the linear model would not be easily interpretable. Therefore, we have decided to transform it into the age of the house at the time of the sale, using the variables yr_built and date. The name of this new variable is house_age (= 2016 - yr_built).

Yr_renovated: This variable is either 0 or the year of renovation of the house, as we can see in the figure on the left. This structure is not useful for linear models. Instead, we decided to transform this variable into a dummy variable indicating if the house was never renovated (0) or if it was renovated at one point (1). As we can see in the figure on the right, the presence of a renovation at one point does seem to have an impact of the price.

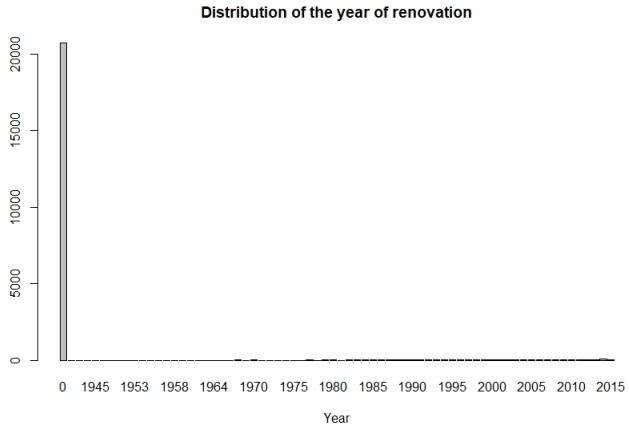

Figure 8: Distribution, yr_renovated

Figure 9: Price / Renovated

Condition: Lastly, we decided use a simple numeric scale from 1 to 5 for the condition of the house. The lower the number, the worse is the condition of the house. In the next figure, we can see that there is a positive linear relationship between the price and the condition.

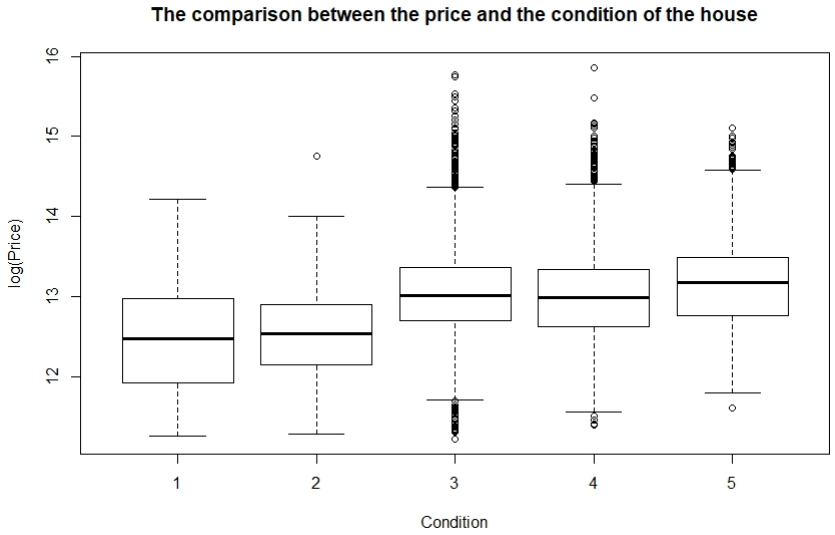

Figure 10: Price / Condition

3.2 Hypothesis testing

3.2.1 Multicollinearity

By looking at the correlation matrix, we can see that some of the variables have high correlation with each other. To make sure it does not effect interpretability of the coefficients too much, we will compute the VIF. VIFs (Variance Inflation Factor) measure the severity of multicollinearity in a linear model using OLS estimators.

The VIFs were computed using the *vif* function of the R package *car: Companion to Applied Regression*. After computing the VIF of all the combinations of the explanatory variables, here are the results:

Variable	**VIF**
bedrooms	1.852718
bathrooms	3.333624
sqft_living (log)	27.126830
sqft_lot (log)	6.910911
floors	2.399886
waterfront	1.206645
view	1.377382
condition	1.236150
grade	3.233437
sqft_above (log)	26.408217
sqft_basement (log)	6.617934
sqft_living15 (log)	2.791892
sqft_lot15 (log)	6.661876
expensive_area	1.261724
house_age	2.174754
renovated	1.149602

We can see that there are some VIF's above 10 and that the average VIF (5.98) is considerably bigger than 1, indicating a multicollinearity problem. Specifically with 'sqft_living (log)', 'sqft_above (log)' and 'sqft_basement (log)'. In fact, the sqft_living variable is the result of the the sum of the basement and the above area. Even if we applied a log transformation to those variables, there is still an important dependence between those variables, resulting in high VIF's. To solve this issue, we simply have to remove either sqft_living or sqft_above. We decided to

remove sqft_above which was less significant than sqft_living in the correlation matrix. Here are the new VIF's once the sqft_above has been removed:

Variable	VIF
bedrooms	1.851938
bathrooms	3.318522
sqft_living (log)	6.038985
sqft_lot (log)	6.893338
floors	2.263932
waterfront	1.206580
view	1.377382
condition	1.228105
grade	3.146418
sqft_basement (log)	1.668733
sqft_living15 (log)	2.786306
sqft_lot15 (log)	6.660432
expensive_area	1.260532
house_age	2.165823
renovated	1.149426

Now we can see that no VIF is higher than 10 and that the average VIF (2.867763) is not considerably bigger than 1. We can consider that the multicollinearity problem is solved.

3.2.2 Heteroskedasticity

We already mentioned the problem of heteroskedasticity with the price and the living area. To solve this issues, we used the logarithm of both. To investigate further issues related to the variance of the error terms, we will conduct a hypothesis test to determine if there is still a heteroskedasticity problem.

In this experiment, we decided to use the Goldfeld-Quandt test. This test separates the original data into two subsets of equal size and run two linear regression models to compare the distribution of the error terms. These two subsets are defined using the explanatory variable: 1 lowest values 0 highest values. This test was conducted in R using the *gqtest* function in the package *lmtest: Testing Linear Regression Models*. We do not obtain an extremely small p-value (< 0.277), so we do not reject the null hypothesis of homoskedasticity and we assume that there is no heteroskedasticity problem with our data.

3.2.3 Autocorrelation

Autocorrelation occurs when there is a correlation between successive observations of same variable, when the residuals are not independent from each other. A test to detect autocorrelation is the Breusch-Godfrey test. This test was conducted in R using the *bgtest* function in the package *lmtest: Testing Linear Regression Models*. This test resulted in a p-value of 0.1264. As we cannot reject the null hypothesis (no autocorrelation) with a 95% confidence level, we can assume that there is no autocorrelation problem with our data.

3.2.4 Outliers

Outliers can be divided in to distinct categories: the outliers with respect to the x-axis and the with the y-axis. We can easily detect them by looking at the distribution of the various features individually. However, we are more interested in the observation that really have an influence on the coefficients. These values are called influential observations.

To decide whether an observation is an outlier/influential or not, we use a multivariate approach called the Cook's distance. Cook's distance is computed with respect to a given regression model and evaluates the impact of an observation on the fitted values by running the model with and without this observation.

The Cook's distance of every observation can be seen on figure 11.

We can see that one observation is extremely influential compared to the other ones. This observation is a house with 33 bedrooms and one bathroom. This data is not plausible and probably originated from a false data entry. It this data was kept, it might false the estimators. This dataline was therefore removed, as one can notice looking at

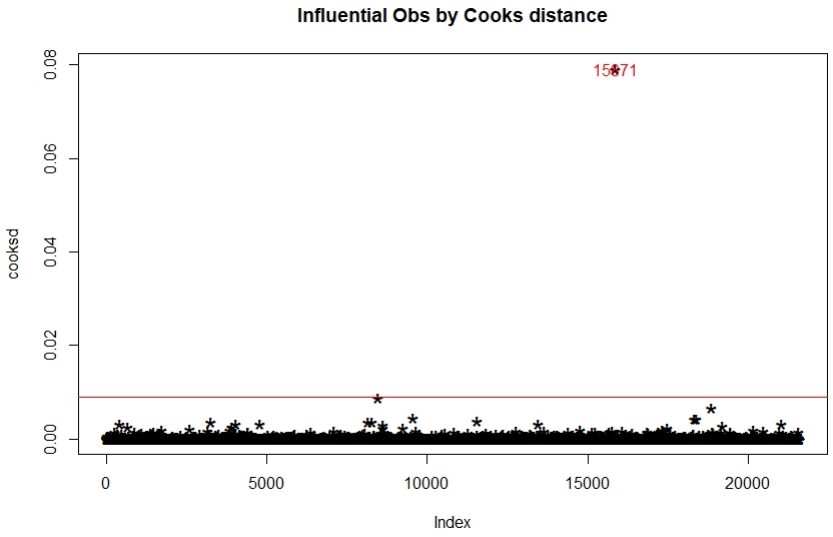

Figure 11: Cook's distances before removal

the previous boxplots of the explanatory variables.

Now, we can have a look at the new Cook's distance of every observation on figure 12.

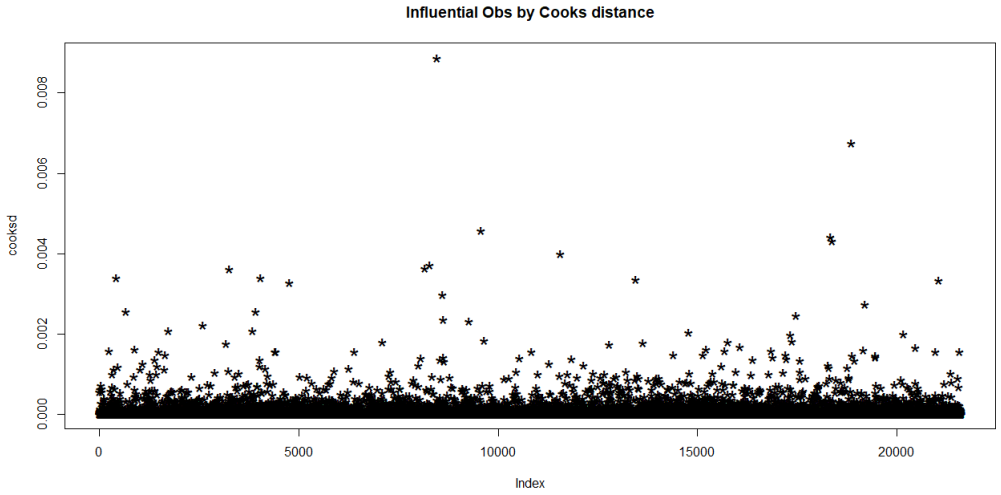

Figure 12: Cook's distances after removal

We can see that none of the Cook's distance are higher than the decision factor threshold of $F_{16,21612-16;0.95} = 0.4975$, so we can affirm that there are no outliers anymore and that we can keep all of our data.

3.3 Variable selection

Three main techniques to proceed to variable selection were seen in this class. We will go further using two of those techniques, the first one being stepwise regression (forward and backward) and the second one being LASSO regression.

We will judge the quality of those models using a training and testing dataset, the first one composed of 75% of the data and the other one of 25%. We will fit models on the training dataset and perform predictions of the testing dataset. We will then use the MSE as a selection criterion to select the model that has the best predictive power. We will also include the full model (no outliers/heteroskedasticity/multicollinearity) in the comparison. This full model can be seen in figure 13.

```
Residuals:
     Min       1Q   Median       3Q      Max
-1.27880 -0.14902 -0.00757  0.13956  1.32857

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.040e+00  5.482e-02 128.433  < 2e-16 ***
bedrooms      -2.189e-02  2.421e-03  -9.041  < 2e-16 ***
bathrooms      5.722e-02  3.821e-03  14.972  < 2e-16 ***
sqft_living    3.538e-01  9.346e-03  37.854  < 2e-16 ***
sqft_lot       3.144e-02  4.700e-03   6.688 2.31e-11 ***
floors         4.311e-02  4.502e-03   9.577  < 2e-16 ***
waterfront     4.496e-01  2.051e-02  21.920  < 2e-16 ***
view           5.971e-02  2.474e-03  24.132  < 2e-16 ***
condition      5.721e-02  2.752e-03  20.789  < 2e-16 ***
grade          1.438e-01  2.438e-03  58.969  < 2e-16 ***
sqft_basement -5.896e-04  6.584e-04  -0.896    0.371
sqft_living15  1.903e-01  8.234e-03  23.115  < 2e-16 ***
sqft_lot15    -2.627e-02  5.128e-03  -5.122 3.05e-07 ***
expensive_area 4.540e-01  3.816e-03 118.979  < 2e-16 ***
house_age      3.006e-03  8.094e-05  37.133  < 2e-16 ***
renovated      6.796e-02  8.607e-03   7.896 3.01e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2375 on 21596 degrees of freedom
Multiple R-squared:  0.7968,    Adjusted R-squared:  0.7966
F-statistic:  5645 on 15 and 21596 DF,  p-value: < 2.2e-16
```

Figure 13: Summary of the basic model

3.3.1 Stepwise regression

Stepwise regression is useful to determine if the addition or removal of variables can increase the predictive power of the linear model. To perform stepwise regression, we will use the built-in R function *step*. We will perform both forward selection regression and backward elimination regression. The way these regressions were carried out in R can be seen in the appendix.

Forward selection regression: We started the model with only one variable that seemed very significant (expensive_area) and proceeded forward with the inclusion of further variables. The stepwise regression decided to include all the available variables except of sqft_basement that was removed.

Backward elimination regression: We started with a full model and let the elimination process remove superfluous variables. The elimination process only removed the sqft_basement variable, so as with the forward selection regression, we obtain the full model with the only variable removed being sqft_basement.

3.3.2 LASSO regression

The LASSO regression fitting will be performed using R functions from the package *"glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models"*. The optimal lambda parameter to perform LASSO regression was found using the *cv.glmnet* function, and the model was then built using the *glmnet* function. The code that was used can be consulted in the appendix.

The best lambda found is 0.0004600288 and it drove four variables to 0: "bedrooms", "sqft_lot", "sqft_basement" and "sqft_lot15". Here are the coefficients that were obtained using LASSO regression:

Variable	β
Intercept	7.414397141
bedrooms	0
bathrooms	0.036717732
sqft_living (log)	0.342812155
sqft_lot (log)	0
floors	0.014070183
waterfront	0.376540334
view	0.059429855
condition	0.041676690
grade	0.144199846
sqft_basement (log)	0
sqft_living15 (log)	0.171211024
sqft_lot15 (log)	0
expensive_area	0.443083527
house_age	0.002317833
renovated	0.045815594

3.4 Results

We can now compare the SSE, MSE and R^2 that were obtained with the predictions on the testing dataset using the different models fitted on the training dataset. The data is randomly split into 75% training and 25% testing.

Model	SSE	MSE	R^2
Basic model	299.6053	0.05545165	0.7937932
Stepwise regression model	299.5972	0.05545016	0.7937987
LASSO regression model	304.1565	0.05629401	0.7906607

Based on these criteria, we will select the stepwise regression model as a final model as it has the lowest SSE and MSE and the highest R^2 on the testing dataset. We will now fit this model to the entire dataset and analyse it more in depth in the next subsection.

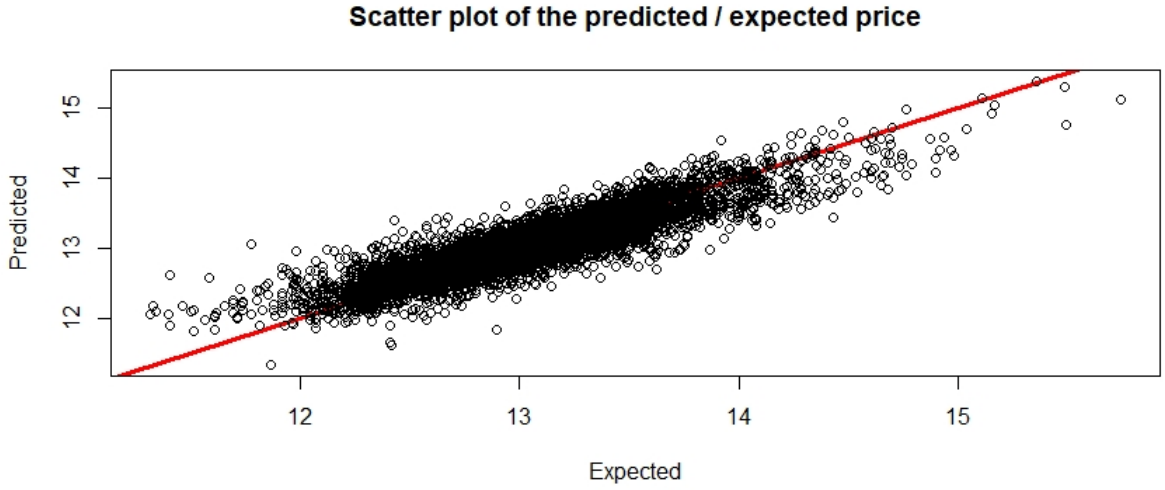

Figure 14: Predictions of the stepwise model

3.5 Model interpretation

A summary of the final model can be seen in figure 15.

We can quickly interpret the estimated coefficients for the quantitative and qualitative variables of the obtained model. We will especially give more details to the coefficients of the qualitative variables.

```
Residuals:
     Min       1Q   Median       3Q      Max
-1.20449 -0.15025 -0.00823  0.14180  1.33510

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     7.071e+00  6.313e-02 111.992  < 2e-16 ***
bedrooms       -2.163e-02  2.798e-03  -7.729 1.15e-14 ***
bathrooms       6.022e-02  4.400e-03  13.685  < 2e-16 ***
sqft_living     3.410e-01  1.021e-02  33.398  < 2e-16 ***
sqft_lot        3.193e-02  5.433e-03   5.878 4.24e-09 ***
floors          4.523e-02  4.653e-03   9.720  < 2e-16 ***
waterfront      4.293e-01  2.342e-02  18.333  < 2e-16 ***
view            6.023e-02  2.847e-03  21.156  < 2e-16 ***
condition       5.690e-02  3.182e-03  17.884  < 2e-16 ***
grade           1.474e-01  2.826e-03  52.146  < 2e-16 ***
sqft_living15   1.925e-01  9.476e-03  20.317  < 2e-16 ***
sqft_lot15     -2.592e-02  5.989e-03  -4.328 1.51e-05 ***
expensive_area  4.548e-01  4.383e-03 103.756  < 2e-16 ***
house_age       3.075e-03  9.351e-05  32.882  < 2e-16 ***
renovated       5.669e-02  9.836e-03   5.763 8.41e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2382 on 16194 degrees of freedom
Multiple R-squared:  0.7976,    Adjusted R-squared:  0.7975
F-statistic:  4559 on 14 and 16194 DF,  p-value: < 2.2e-16
```

Figure 15: Summary of the final stepwise model

bedrooms: This coefficient is negative. It should intuitively be positive, and it is hard to understand why it is negative. At least it is rather small, so its impact is not very noticeable. One possible explanation could be that farms and rural homes could have more bedrooms on average and also have a lower price, influencing the coefficient to be negative. We have not been able to find a solution or explanation to this counter-intuitive coefficient and improvements could be made in that regard.

bathrooms: The coefficient is positive. Therefore, the more bathrooms a house has, the more expensive it should be. This intuitively makes sense.

sqft_living (log): The coefficient is positive and rather big. Therefore, the bigger a house is, the more expensive it should be. This intuitively makes sense.

sqft_lot (log): The coefficient is positive and rather high. Therefore, the bigger a house's lot area is, the more expensive it should be. This intuitively makes sense.

floors: The coefficient is positive. Therefore, the more floors a house has, the more expensive it should be. This intuitively makes sense.

waterfront: This coefficient is positive and it is the second biggest coefficient of them all. This makes sense, as homes on the waterfront are logically more expensive than others. We can interpret this the following way: if two homes are the exact same but one is on the waterfront and the other is not, then we can expect the log of the price of the waterfront home to be bigger by 0.4293. This increase is very big if we remember than we are working in the log scale.

view: This coefficient is positive. This makes sense, as the better the view is, the more expensive a home should be.

condition: This coefficient is positive. This makes sense, as the better the condition of the house is, the more expensive it should be.

grade: This coefficient is positive and rather high. This makes sense, as the better the quality of the construction of the home is, the more expensive it should be.

sqft_living15 (log): This coefficient is positive and rather high. It makes sense that if a home is located in a area surrounded with big homes, it is probably in a richer area and its price should be higher.

sqft_lot15 (log): This coefficient is negative. It might seem counterintuitive, but a possible explanation can be made: if the lot area of the surrounding homes is very big, it is probable that we are in a rural area rather than an urban one. And, as seen on the heatmap, the homes in the urban area are much more expensive than in the rural area of the county.

expensive_area: This coefficient is positive and the highest coefficient of them all. Of course, if a home is located in the expensive area of the state, we can expect it to be more expensive. We see that the impact of the localisation of the home is enormous and that it was a good idea to create the dummy expensive_area variable. In fact, if two homes are the exact same but one is located in the expensive area, we expect the log of its price to be higher by 0.4548, which is a huge increase since we are working in the log scale.

house_age: This coefficient is positive and not rather small. It is hard to find an intuitive relationship between the age and the price of a house, so it is hard to explain this positive coefficient. Since it is small, its impact is not very noticeable anyway.

renovated: This coefficient is positive. It makes sense that a house that was renovated would be more expensive than another one of the same age that was not. We can interpret this the following way: if two homes have the exact same characteristics but one was never renovated and one was renovated, then the predicted log of the price of the renovated house will be bigger by 0.05669.

We obtain a model with an adjusted R-squared of 0.7975. We are able to explain 79.75% of the variation of the price with our model, which is a considerable proportion of the variation.

3.5.1 Linear combination of coefficients

In this small subsection, we wanted to test a linear combination of the coefficients of the number of bedrooms and the number of bathrooms.

We have decided to combine the variables expensive_area and waterfront into a product of them both. Therefore, this product will be equal to 1 if the home is on the waterfront in the expensive area and equal to 0 elsewhere. We have decided to combine these two because we intuitively expect those homes to be the most expensive of all, and we think that we should be able to prove that the coefficient is positive.

The summary of the new model can be seen in the next figure:

```
Residuals:
     Min       1Q   Median       3Q      Max
-1.27881 -0.14902 -0.00748  0.13976  1.32784

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                7.042e+00  5.483e-02 128.440  < 2e-16 ***
bedrooms                  -2.193e-02  2.422e-03  -9.055  < 2e-16 ***
bathrooms                  5.715e-02  3.822e-03  14.955  < 2e-16 ***
sqft_living                3.537e-01  9.346e-03  37.849  < 2e-16 ***
sqft_lot                   3.144e-02  4.700e-03   6.689 2.31e-11 ***
floors                     4.327e-02  4.503e-03   9.610  < 2e-16 ***
waterfront                 4.202e-01  2.924e-02  14.370  < 2e-16 ***
view                       5.973e-02  2.474e-03  24.139  < 2e-16 ***
condition                  5.720e-02  2.752e-03  20.787  < 2e-16 ***
grade                      1.437e-01  2.438e-03  58.947  < 2e-16 ***
sqft_basement             -5.797e-04  6.585e-04  -0.880    0.379
sqft_living15              1.902e-01  8.235e-03  23.090  < 2e-16 ***
sqft_lot15                -2.617e-02  5.129e-03  -5.102 3.38e-07 ***
expensive_area             4.536e-01  3.826e-03 118.553  < 2e-16 ***
house_age                  3.007e-03  8.095e-05  37.147  < 2e-16 ***
renovated                  6.796e-02  8.606e-03   7.897 3.00e-15 ***
waterfront:expensive_area  5.317e-02  3.765e-02   1.412    0.158
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2375 on 21595 degrees of freedom
Multiple R-squared:  0.7968,    Adjusted R-squared:  0.7967
F-statistic:  5293 on 16 and 21595 DF,  p-value: < 2.2e-16
```

Figure 16: Summary of the model with linear combination

We see that the coefficient multiplying waterfront*expensive_area is positive but that it does not hold a great level of confidence. We will now test the following hypotheses:

$$H_0 = \beta_{exp} + \beta_{exp*waterfront} * \bar{x}_{waterfront} = 0$$

$$H_1 = \beta_{exp} + \beta_{exp*waterfront} * \bar{x}_{waterfront} > 0$$

These hypotheses were tested using the *glht* function of the R package *multcomp: Simultaneous Inference in General Parametric Models*. Here are the results:

Estimate	Std. Error	t value	Pr(>t)
0.453568	0.003784	119.9	<2e-16

We can then see that when we used the explained linear combination, we are able to see an effect on the presence of a home on the expensive area, at the mean level of waterfront presence, with a high degree of confidence. However. the coefficient multiplying the product of the two variables itself does not hold a great level of confidence.

3.6 Prediction interval

To further evaluate our predictions, we will first express the output variable price in dollars and no longer in logarithmic scale for interpretability. Then, we will compute confidence intervals with 95% confidence and see how many observations were excluded from the boundaries.

In the next table, you can find the predictions and boundaries of the first observations as an example:

	Prediction	Lower bound	Upper bound	Real
1	287844.1	180429.6	459205.3	257500
2	721211.3	452031.5	1150684.8	662500
3	693196.8	434386.7	1106207.6	696000
4	613163.8	384352.5	978190.3	605000
...	...	...	...	...

As you can see the real outcomes are relatively close to the predicted values. In fact, we have tested all of our data and we have observed that every single observation was included in the 95% prediction interval for price. In general, the model predicts the price of a house with a mean absolute error of 15793.09$ which is a good estimate knowing the big prices of these houses.

4 Conclusion

We can conclude this report by stating that our objective was obtained: we were able to obtain a powerful predictive model for the sale price of houses in King County, USA. In short, we were able to:

- Analyze and modify the available data;
- Test the underlying assumptions of linear models and apply corrections when these assumptions were not respected;
- Choose a model that was considered the best according to decision criteria;
- Perform predictions on the data using this final model and verify that these predictions were accurate.

A APPENDIX

A.1 R code

```
############################## LSTAT2120 Project ################################


#### Data importing and analysis ####


referencedata <- read.csv("kc_house_data.csv")
housedata <- referencedata

# outliers
housedata <- housedata[which(housedata$bedrooms < 33), ]


# boxplots
par(mfrow=c(1,2))
boxplot(housedata$price, xlab = "price")
boxplot(housedata[, c(4, 5, 8, 10, 11, 12)], cex.axis = 0.8)
boxplot(housedata[, c(6, 13, 14, 20)])
boxplot(housedata[, c(7, 21)])
par(mfrow=c(2,3))
boxplot(housedata$zipcode, xlab = "zipcode")
boxplot(housedata$lat, xlab = "lat")
boxplot(housedata$long, xlab = "long")
boxplot(housedata$yr_built, xlab  = "yr_built")
boxplot(housedata$yr_renovated, xlab = "yr_renovated")
boxplot(housedata$waterfront, xlab = "waterfront")
par(mfrow=c(1,1))

# heatmap
library(ggmap)
register_google(key = "AIzaSyDR2ob6a6HSgsBhZkN-k0QNVeJT3uio4Wg")
map <- get_map(location = c(left = min(housedata$long),
                 bottom = min(housedata$lat),
                 right = max(housedata$long),
                 top = max(housedata$lat)))
ggmap(map, extent = "device")

heatmapdata <- data.frame(cbind(log(housedata$price), housedata$lat,
                                housedata$long))
colnames(heatmapdata) <- c("logprice", "lat", "long")

ggmap(map, extent = "device") +
    stat_summary_2d(data = heatmapdata,
                    aes(x = long, y = lat,
                        z = logprice),
                    fun = mean, alpha = 0.6, bins = 30) +
    scale_fill_gradient(name = "Log(Price)", low = "green",
                        high = "red") +
    annotate('segment', x=min(housedata$long), xend=max(housedata$long),
             y=47.52, yend = 47.52,
             colour='black', lty = 2, lwd = 1.3)


#### Feature engineering ####

# expensive_area
housedata$expensive_area <- sapply(1:nrow(housedata),
    function(i) as.numeric(housedata$lat[i] >= 47.52))

# house_age
housedata$house_age <- sapply(1:nrow(housedata),
```

```
    function(i) as.numeric(substr(housedata$date[i], 1, 4)) -
                housedata$yr_built[i])
for (i in 1:nrow(housedata))
{
    if (housedata$house_age[i] < 0)
    {
        housedata$house_age[i] <- 0
    }
}

# renovated
housedata$renovated <- sapply(1:nrow(housedata),
                              function(i) as.numeric(housedata$yr_renovated[i] > 0))

# logarithme de certaines variables
housedata$price <- log(housedata$price)
housedata$sqft_living <- log(housedata$sqft_living)
housedata$sqft_above <- ifelse(housedata['sqft_above']> 0,
                               log(housedata$sqft_above), 0)
housedata$sqft_basement <- ifelse(housedata['sqft_basement']> 0,
                                  log(housedata$sqft_basement), 0)
housedata$sqft_living15 <- ifelse(housedata['sqft_living15']> 0,
                                  log(housedata$sqft_living15), 0)
housedata$sqft_lot <- ifelse(housedata['sqft_lot']> 0,
                             log(housedata$sqft_lot), 0)
housedata$sqft_lot15 <- ifelse(housedata['sqft_lot15']> 0,
                               log(housedata$sqft_lot15), 0)

# removing useless variables
housedata$yr_built <- NULL
housedata$yr_renovated <- NULL
housedata$zipcode <- NULL
housedata$lat <- NULL
housedata$long <- NULL


#### Hypothesis testing ####

## Multicollinearity

library(car)
vifs <- vif(lm(price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+
               waterfront+view+condition+grade+sqft_above+sqft_basement+
               sqft_living15+sqft_lot15+expensive_area+house_age+
               renovated, housedata))
vifs
mean(vifs) # multicollinearity problem, we remove sqft_above

newvifs <- vif(lm(price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+
               waterfront+view+condition+grade+sqft_basement+
               sqft_living15+sqft_lot15+expensive_area+house_age+
               renovated, housedata))
newvifs
mean(newvifs) # no more multicollinearity problem

housedata$sqft_above <- NULL

## Heteroskedasticity

library(lmtest)

bptest(lm(price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+
              waterfront+view+condition+grade+sqft_basement+
              sqft_living15+sqft_lot15+expensive_area+house_age+
              renovated, housedata)) # heteroskedasticity
```

```
gqtest(lm(price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+
              waterfront+view+condition+grade+sqft_basement+
              sqft_living15+sqft_lot15+expensive_area+house_age+
              renovated, housedata), data = housedata)

plot(housedata$bedrooms, housedata$price)
plot(housedata$bathrooms, housedata$price)
plot(housedata$sqft_living, housedata$price)
plot(housedata$sqft_lot, housedata$price)
plot(housedata$floors, housedata$price)
plot(housedata$waterfront, housedata$price)
plot(housedata$view, housedata$price)
plot(housedata$condition, housedata$price)
plot(housedata$grade, housedata$price)
plot(housedata$sqft_basement, housedata$price)
plot(housedata$sqft_living15, housedata$price)
plot(housedata$sqft_lot15, housedata$price)
plot(housedata$expensive_area, housedata$price)
plot(housedata$house_age, housedata$price)
plot(housedata$renovated, housedata$price)


## Autocorrelation

library(lmtest)

bgtest(lm(price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+
              waterfront+view+condition+grade+sqft_basement+
              sqft_living15+sqft_lot15+expensive_area+house_age+
              renovated, housedata)) # no autocorrelation

## Outliers

cooksd <- cooks.distance(lm(price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+
                      waterfront+view+condition+grade+sqft_basement+
                      sqft_living15+sqft_lot15+expensive_area+house_age+
                      renovated, housedata))
plot(cooksd, pch="*", cex=1, main="Influential␣Obs␣by␣Cooks␣distance")
# plot cook's distance
abline(h = qf(0.05, 16, 21612-16), col="red")
abline(h = 4/21612, col="red")
qf(0.05, 16, 21612-16) # no influential observations


#### Cross validation ####

# Testing and training dataset

library(ISLR)
library(glmnet)
library(dplyr)
library(tidyr)

set.seed(7)
smp_size <- floor(0.75 * nrow(housedata))
train_ind <- sample(seq_len(nrow(housedata)), size = smp_size)

train <- housedata[train_ind, ]
test <- housedata[-train_ind, ]

# Basic model

basic_model <- lm(price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+
                      waterfront+view+condition+grade+sqft_basement+
```

```
                      sqft_living15+sqft_lot15+expensive_area+house_age+
                      renovated, train)
basic_pred <- predict(basic_model, test)
mean((basic_pred - test$price)^2)

# Stepwise regression

step(basic_model, direction = "backward")

step(lm(price~expensive_area, train),
     scope = price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+
         waterfront+view+condition+grade+sqft_basement+
         sqft_living15+sqft_lot15+expensive_area+house_age+
         renovated, direction = "forward")

stepwise_model <- lm(price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+
                     waterfront+view+condition+grade+
                     sqft_living15+sqft_lot15+expensive_area+house_age+
                     renovated, train)
stepwise_pred <- predict(stepwise_model, test)

mean((stepwise_pred - test$price)^2)


# Lasso regression

x_test = as.matrix(as.matrix(test[, 4:ncol(test)]))
y_test = test$price

x_train = as.matrix(as.matrix(train[, 4:ncol(train)]))
y_train = train$price

grid = 10^seq(10, -2, length = 100)
lasso_mod = glmnet(x_train, y_train, alpha = 1, lambda = grid)
# Fit lasso model on training data

plot(lasso_mod)    # Draw plot of coefficients

set.seed(1)
cv.out = cv.glmnet(x_train, y_train, alpha = 1)
# Fit lasso model on training data
plot(cv.out) # Draw plot of training MSE as a function of lambda
bestlam = cv.out$lambda.min # Select lamda that minimizes training MSE
#lasso_mod = glmnet(x_train, y_train, alpha = 1, lambda = bestlam)
# Fit lasso model on training data
lasso_pred = predict(lasso_mod, s = bestlam, newx = x_test)
# Use best lambda to predict test data
mean((lasso_pred - y_test)^2) # Calculate test MSE

x <- as.matrix(housedata[, 4:ncol(housedata)])
y <- as.matrix(housedata$price)

out = glmnet(x, y, alpha = 1, lambda = grid) # Fit lasso model on full dataset
lasso_coef = predict(out, type = "coefficients", s = bestlam)[1:16,]
# Display coefficients using lambda chosen by CV
lasso_coef

lasso_coef[lasso_coef != 0] # Display only non-zero coefficients


#intervals

stepwise_pred_interval <- predict(stepwise_model, test, interval="predict")

# check if is in interval
```

```
inter_check <- function(pred) {
    count_pos = 0
    for (i in 1:length(pred[,1])) {
        if(pred[i,1] >= pred[i,2] &  pred[i,1] <= pred[i,3]){
            count_pos = count_pos + 1
        }
    }
    return((count_pos*100) / length(pred[,1]))
}
inter_check(stepwise_pred_interval)
# inter_check(lasso_pred_interval)
# inter_check(pred_interval)

real_pred_int = list()
for (i in 1:length(stepwise_pred_interval[,1])) {
    val = c()
    val = c(val, exp(stepwise_pred_interval[i,1]))
    val = c(val, exp(stepwise_pred_interval[i,2]))
    val = c(val, exp(stepwise_pred_interval[i,3]))

    real_pred_int[[i]] = val
}

real_pred = exp(stepwise_pred)
real_SSE_step <- sum((real_pred - exp(y_test)) ^ 2)
real_MSE_step <- abs(mean((real_pred - exp(y_test))))

head(real_pred_int)

plot((y_test), log(real_pred),
main="Scatter␣plot␣of␣the␣predicted/expected␣price",
xlab="Expected",ylab="Predicted", abline(a=0, b=1, col='red', lwd=3))

plot((y_test), log(real_pred) - (y_test), abline(h=0, col='red', lwd=3),
main="Plot␣of␣the␣residuals", xlab="log(Price)", ylab="Residuals")



# Calculating MSE, SSE and R-squared

data.frame(MSE = c(mean((basic_pred - test$price)^2),
                   mean((stepwise_pred - test$price)^2),
                   mean((lasso_pred - test$price)^2)),
           SSE = c(sum((basic_pred - test$price)^2),
                   sum((stepwise_pred - test$price)^2),
                   sum((lasso_pred - test$price)^2)),
           Rsquared = c(1-sum((basic_pred - test$price)^2)/
                            sum((test$price - mean(test$price))^2),
                        1-sum((stepwise_pred - test$price)^2)/
                            sum((test$price - mean(test$price))^2),
                        1-sum((lasso_pred - test$price)^2)/
                            sum((test$price - mean(test$price))^2)),
           row.names = c("Basic", "Stepwise", "Lasso"))


#### Linear combination ####

library(multcomp)
model <- lm(price~bedrooms+bathrooms+sqft_living+sqft_lot+floors+
                  waterfront+view+condition+grade+
                  sqft_living15+sqft_lot15+expensive_area+house_age+
                  renovated + waterfront*expensive_area, housedata)
summary(model)
coefeq <- matrix(0, nrow = 1, ncol = length(model$coefficients))
colnames(coefeq) <- names(model$coefficients)
```

```r
coefeq[1, "expensive_area"] <- 1
coefeq[1, "waterfront:expensive_area"] <- mean(housedata$waterfront)
coefeq %*% model$coefficients

ametest <- glht(model = model, linfct = coefeq, rhs = 0, alternative = "greater")
summary(ametest)
```

References

[1] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[2] David Kahle and Hadley Wickham. ggmap: Spatial visualization with ggplot2. *The R Journal*, 5(1):144–161, 2013.

[3] Achim Zeileis and Torsten Hothorn. Diagnostic checking in regression relationships. *R News*, 2(3):7–10, 2002.