Interpretation and prediction of house prices

Lucía Canales, Víctor Collado, Alejandro Marcos, Arturo Pérez and Luis J. Vidal

2023-12-15

Contents

Introduction and objectives **1**

Exploratory Data Analysis **1**
Exploration of the variables and data cleaning . . . 1
Treatment of missing values . . . 1
Transformations of variables . . . 2
Correlation . . . 4
Outliers . . . 4

Creation of the model **5**
Linear Model . . . 6
Model selection . . . 6
Interaction terms. . . . 6
Model assumptions . . . 7
Prediction and estimation of the error. . . . 8
Influence of the variables . . . 9
Ridge regression . . . 11
Generalized Additive Models . . . 12

Conclusions **15**

Introduction and objectives

The price of a house is a complex quantity that depends on many different factors, ranging from its surface area to the quality of the neighborhood it is built on. This is why its computation is an interesting exercise which will be the aim of this work. We have been given two different data sets: *train.csv* and *test.csv*. The goal is to use the former to develop a linear model that relates the different measured variables with the price of a house. Its accuracy will be tested on the latter.

The way we are going to proceed is the following: 1. We are going to carry out an exploratory data analysis to prepare the variables (omit or replace missing values, encode categorical variables,...) 2. We are going to fit a linear model and try to improve the fit through careful updates to the base variables. 3. We are going to use this model to predict the values of the test dataset. 4. We are going to see whether the data used satisfies the hypothesis of the model and whether or not we can improve it using more advanced techniques.

Our aim is that of creating a reasonable, accurate model for which we will perform an in-depth analysis of its different parameters using the tools learned in class.

Exploratory Data Analysis

Exploration of the variables and data cleaning

The first step is the proper examination of the given data. We need to study the different variables and treat them when working with them proves to be hard or outright impossible, or when they fail to satisfy an hypothesis of the model.

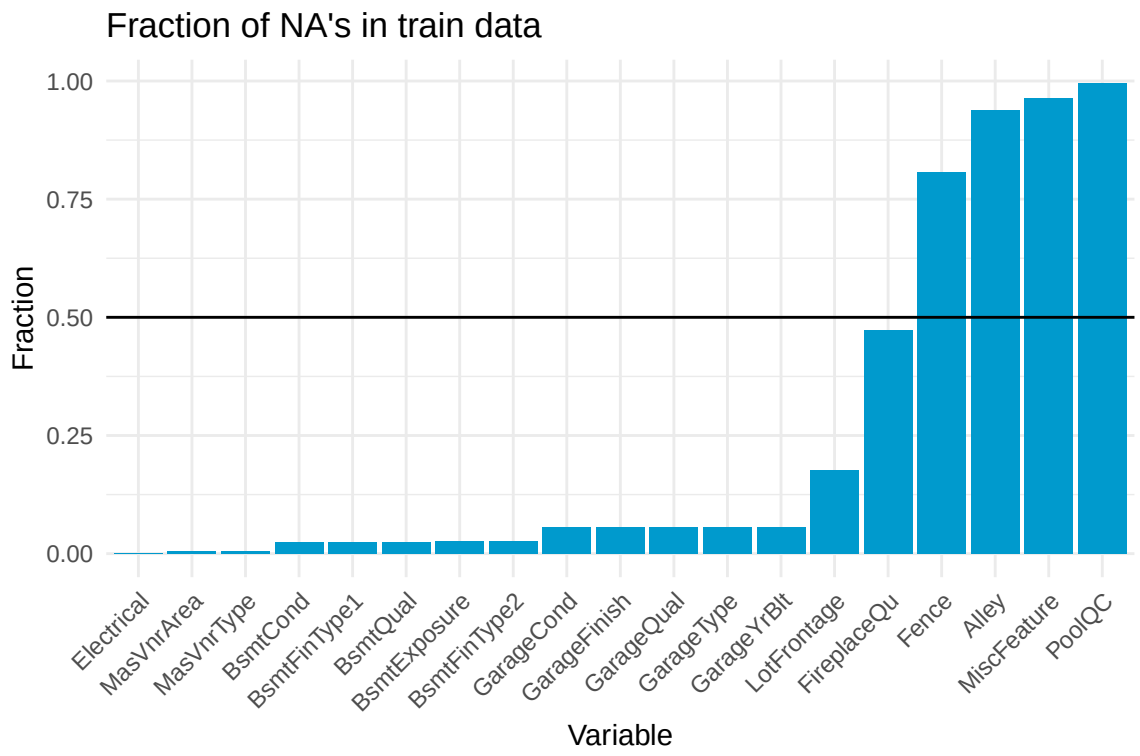
First of all, we load the data and carry out the transformations needed for each dataset:

We drop the ID variable and apply the logarithm to the `SalePrice` variable in order to correct for skewness (Besides, we are asked to do this by the assignment). Moreover we transform variables that take string values and some integer values into factor variables.

We also remove `Utilities` because it is almost constant (with the exception of one observation, the rest of them belong to the "AllPub" category).

Treatment of missing values

We need to take a deep look at each variable and analyze the missing values of each of them. In order to do this we can make a barplot of the proportion of missing values.



We can see that over 50% of the observations in the variables `Fence`, `Alley`, `MiscFeature` and `PoolQC` are NAs. Now, let us recall that the NAs in these columns indicate the absence of whatever the variable represents. Therefore, these columns are almost constant and we can ignore them without paying more thought, deleting their corresponding columns in both the train and test datasets.

We are now ready to treat the rest of the missing values. For starters, for some categorical variables (namely, those related with `Fireplace`, `Garage` and `Basement`) the NAs do not represent a missing value but rather the lack of a certain attribute or asset. However we cannot just add NA as factor level: there are some observations for which there is genuinely missing data. For these kind of observations we are going to impute the missing values by the mean (if the feature is numerical) or by the mode (if it is categorical). In case that the mode of a categorical value is the "NA" factor level, the imputation will be performed using the second most common factor level.

We will perform an imputation by either the mean or the mode for the variables which still have NAs. The only exception to this rule will be `LotFrontage` due to the high proportion of missing values it has. Instead we will perform an imputation by k-Nearest-Neighbors.

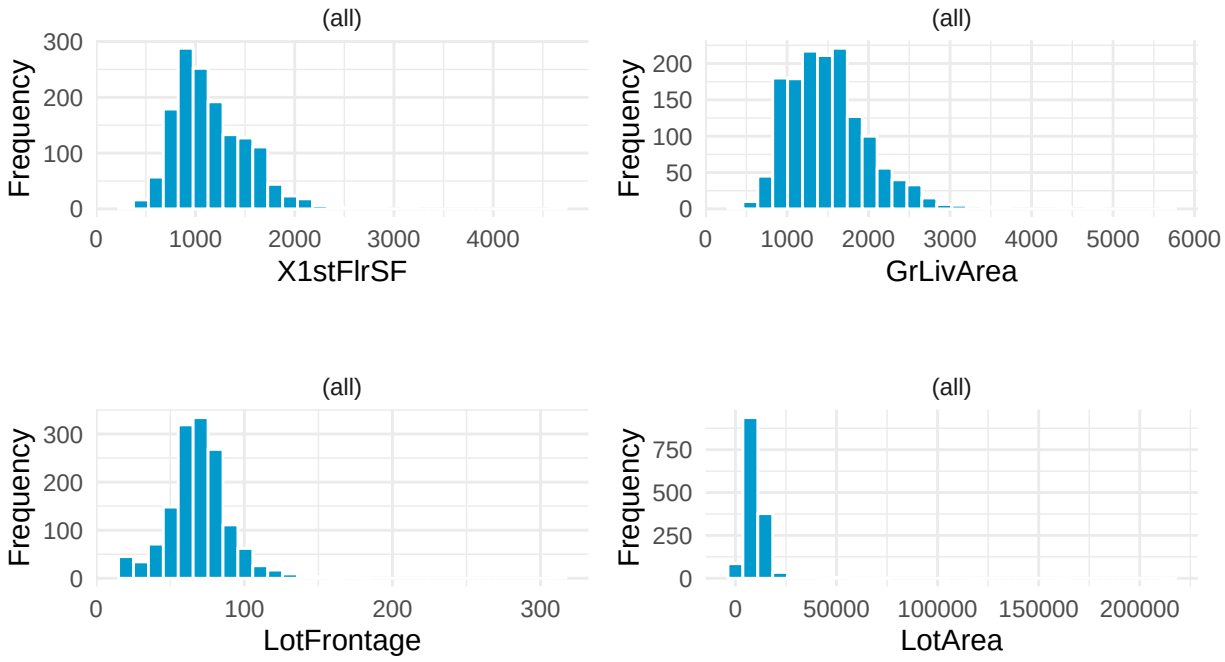
Transformations of variables

There are certain categorical variables whose factors follow an ordered structure. We have decided it is appropriate to transform these variables into integers. Although there is some subjective element to this decision (if we encode an average basement as a 1 and a good basement as a 2, does it mean that a good basement is twice as good as an average one?). However, we have decided it is more important to encode the ordered structure of said variables.

The variables we have transformed in this manner are `BsmtQual`, `BsmtExposure`, `BsmtFinType1`, `HeatingQC`, `KitchenQual` and `FireplaceQual`.

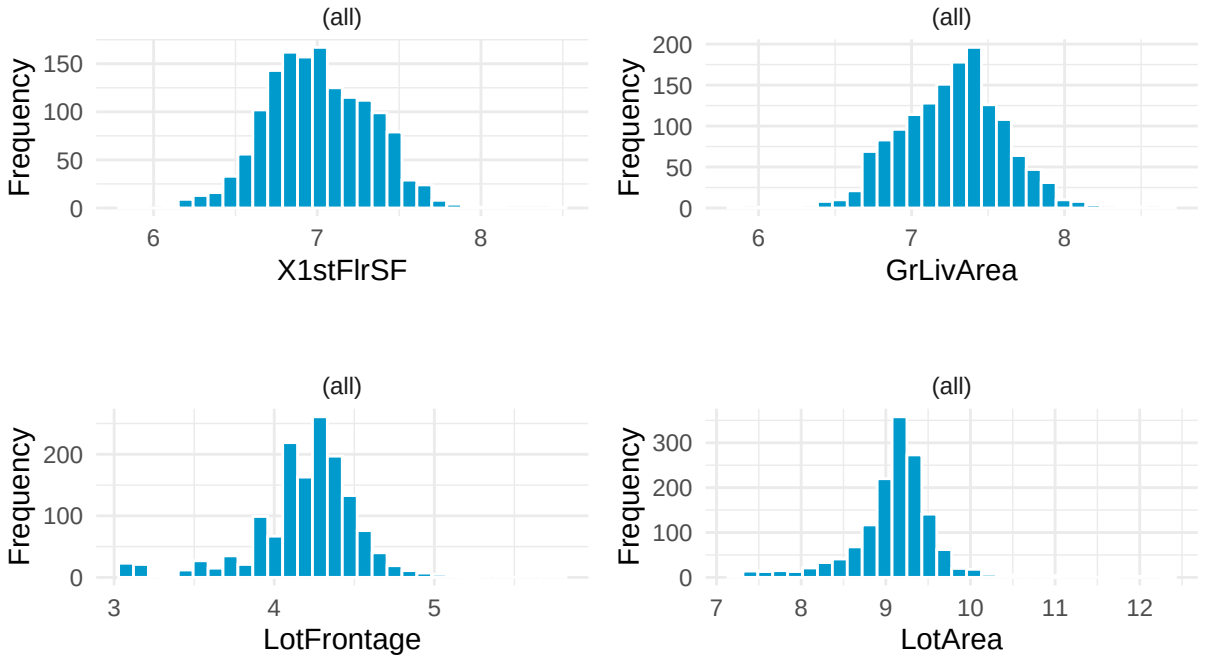
We now transform the year variables, dividing them into intervals.

We now correct the skewness of the quantitative variables. In order to better visualize the skewness of the variables we plot their histograms.



We are going to correct `X1stFlrSF`, `GrLivArea`, `LotFrontage` and `LotArea` using the logarithm.

Now that we have mitigated the asymmetry of the variables, we can plot their histograms again to check whether their distribution is closer to a normal curve or not.



Perhaps one could think that it would be wise to fit a boxcox transformation to modify the data. However, the optimal value of λ is close to one which implies that the transformation does not change our data all that much. Therefore we have decided to discard the boxcox transformation altogether in order to prevent overfitting.

Correlation

We now study the correlation of the predictors with `SalePrice`. We are not going to do any visualization due to the high number of variables (We would either show a poor visualization or a really big number of plots. Moreover, if we decided to do, say, scatter plots the computational cost would surge). Instead we are going to compute the correlation matrix and select the variables with correlation coefficient greater than 0.6 in absolute value.

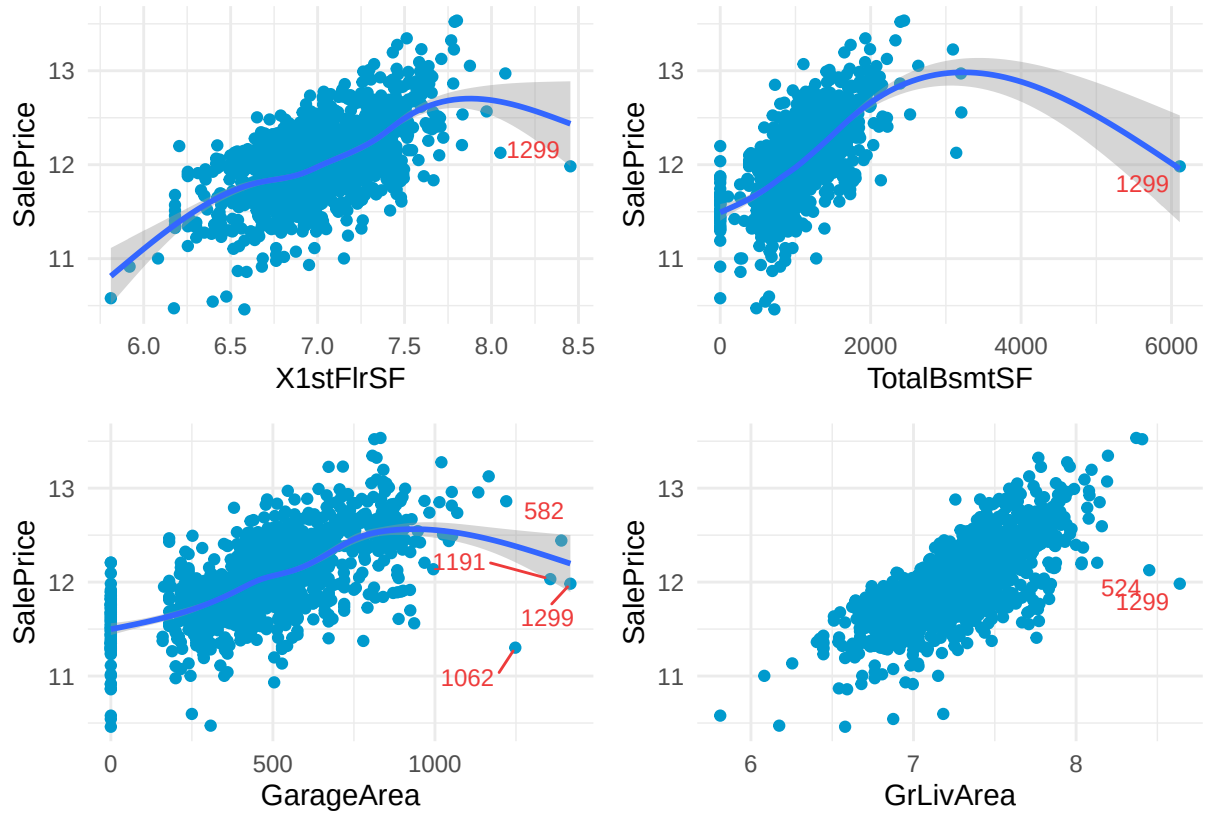
The variables that fulfill this criterion are `X1stFlrSF`, `TotalBsmtSF`, `GarageArea`, `KitchenQual`, `GarageCars`, `GrLivArea` and `OverallQual` with positive correlation and `GarageFinish` with negative correlation.

```
## GarageFinish    X1stFlrSF  TotalBsmtSF   GarageArea  KitchenQual   GarageCars
##           1           66           67           68           69           70
##   GrLivArea  OverallQual    SalePrice
##          71           72           73
```

Outliers

Now that we have checked the correlation of the predictors with the response variable we can look for outliers in said variables (because they are the ones that have the greatest influence on the response).

We start with the quantitative variables: `X1stFlrSF`, `TotalBsmtSF`, `GarageArea` and `GrLivArea`:



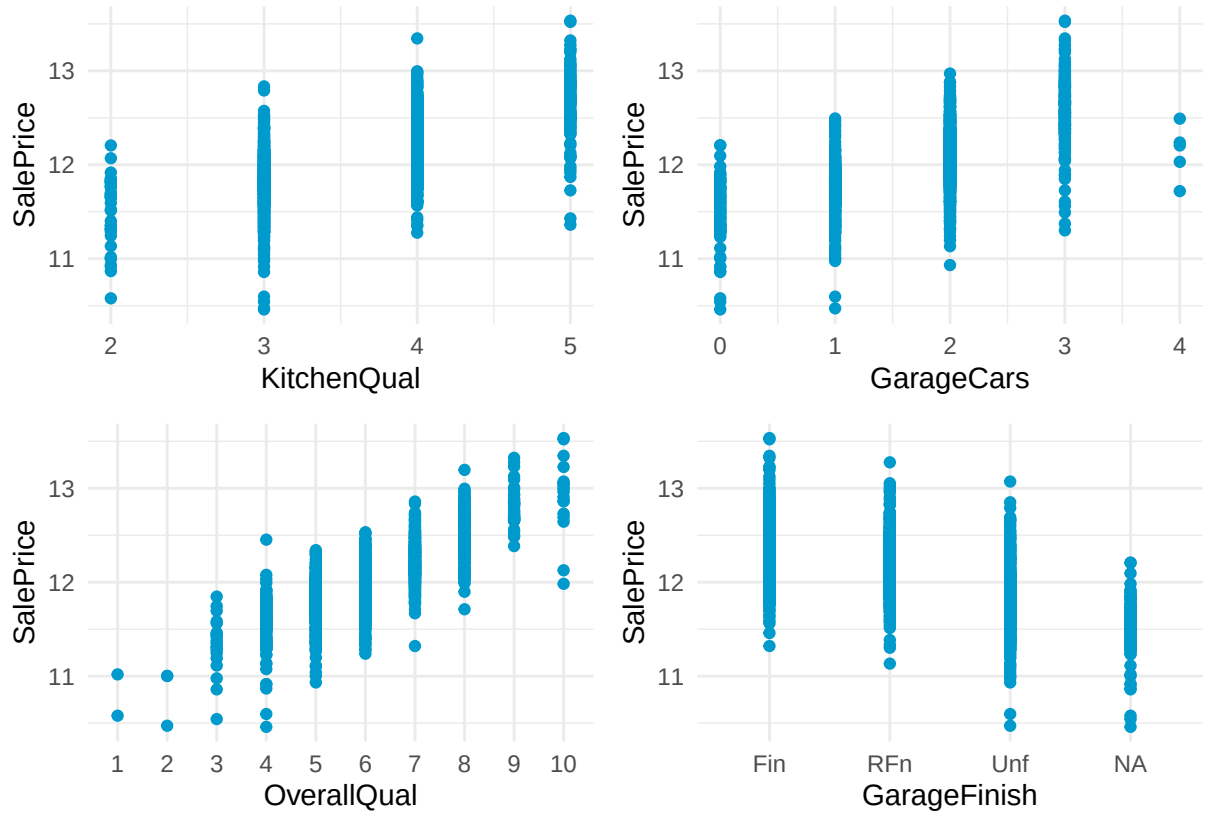
There is one blatant outlier in `X1stFlrSF`: the 1299th observation.

For the variable `TotalBsmtSF`, we can see one outlier at the right. This outlier corresponds, once more, to the 1299th observation.

Related to `GarageArea`, there are 3 or 4 possible outliers at the right of the plot. One of them is the same as before, observation number 1299. The rest of the outliers are observations 582, 1062 and 1191.

Finally, there are two points with possible outliers: they have a big value for the `GrLivArea` predictor but the price of the house is lower than what one may expect (524 and 1299).

Now, for the qualitative variables: `KitchenQual`, `GarageCars`, `OverallQual` and `GarageFinish`:



We can not easily point at the existence of outliers using the plot for `KitchenQual`.

Related to `GarageCars` it appears that garages with 4 cars are outliers, but we cannot say for certain. These observations correspond to 421, 748, 1191, 1341 and 1351.

There are two observations with an `OverallQual` of 10 but with low `SalePrice`. They are the 524th and the 1299th.

Finally, we cannot infer much from the graphic related to `GarageFinish`.

Having done this, we can now remove some of the observations we have mentioned in order to get our models to fulfill the linear regression hypotheses.

Creation of the model

Now that we have cleaned our data we are ready to start building our model. We are going to explore three different ideas, namely:

1. A linear model.
2. A ridge model.
3. A GAM model.

We are mainly interested in the linear model due to the abundance of tools we have to improve and analyze it. However, in order to make our final prediction we will use whichever minimizes the mean squared error in a cross-validation setting.

Linear Model

For starters we build a linear model that includes all of the variables in our dataset.

```
mod0 <- lm(SalePrice ~ ., data = train_data)
summary(mod0)$adj.r.squared
```

```
## [1] 0.9426631
```

This model may seem problematic due to the presence of missing values in some of its coefficients. However this only happens in the levels of a categorical variables for which there are not enough observations in said category.

Model selection

This procedure has allowed us to greatly reduce the number of variables. However we can reduce it further by studying the multicollinearity, removing those with redundant information. We are going to use the $GVIF^{1/(2\cdot df)}$ as a measure of multicollinearity, available through the `vif` function in the package `car`. This quantity is then compared with $\sqrt{10} \approx 3.2$, removing each time the variable with the greatest $GVIF^{1/(2\cdot df)}$ as long as it is greater than 3.2.

```
mod1 <- stepAIC(mod0, direction="both", trace=FALSE)
summary(mod1)$adj.r.squared
```

```
## [1] 0.9428805
```

We find NAs for the VIF of the variables that had missing values in their coefficients. The way we are going to proceed is by removing said variables.

We now remove the variable with the greatest $GVIF^{1/(2\cdot df)}$ (checking that it is greater than 3.2). In this case, it is `GrLivArea`. We update the model accordingly:

We repeat this process until all variables have $GVIF^{1/(2\cdot df)}$ smaller than 3.2. This results in the removal of `GarageQual`.

We can now check whether we have improved our fit.

```
summary(mod2)$adj.r.squared
```

```
## [1] 0.9414667
```

The value has actually decreased when compared to `mod1`. However, the decrease in R^2 is not significant enough to persuade us to keep the variables.

Interaction terms.

We now consider the inclusion of interaction terms between categorical and numerical variables. One way to proceed would be to check all possible interaction terms with a loop and keeping those that are the most significant. However, we have chosen to start by checking the most sensible combinations first:

Interaction of `LotArea` with `LotConfig`: We obtain a p-value of 0.4757. Therefore we do not have enough evidence to affirm that the interaction is significant. We then discard it.

Interaction of `LotArea` with `LandSlope`: The p-value is 0.4827 so we will not keep the interaction in the model.

Interaction of `LotArea` with `Neighborhood`: Since the p-value is 0.007618 so we will keep it.

Interaction of `X3SsnPorch` with `EnclosedPorch`: The p-value is 0.002254 we will keep it.

Therefore our final model uses the following predictors:

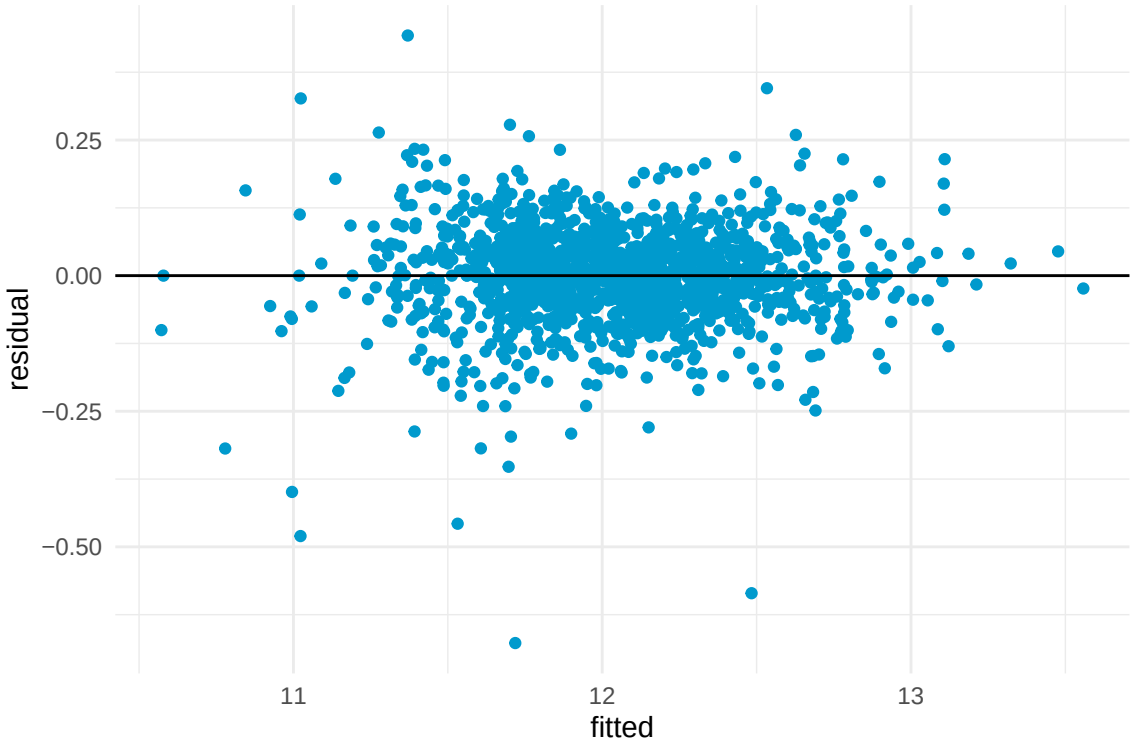
`MSSubClass`, `MSZoning`,`LotArea`, `LotConfig`, `LandSlope`, `Neighborhood`, `Condition1`, `Condition2`, `OverallQual`, `OverallCond`, `YearBuilt`, `YearRemodAdd`, `RoofStyle`, `RoofMatl`, `Exterior1st`, `MasVnrType`, `Foundation`, `BsmtExposure`, `BsmtFinSF1`, `BsmtFinSF2`, `BsmtUnfSF`, `Heating`, `HeatingQC`, `CentralAir`, `X1stFlrSF`, `X2ndFlrSF`, `LowQualFinSF`, `BsmtFullBath`, `FullBath`, `HalfBath`, `KitchenAbvGr`, `KitchenQual`, `Functional`, `Fireplaces`, `GarageType`, `GarageCars`, `GarageArea`, `WoodDeckSF`, `OpenPorchSF`, `EnclosedPorch`, `X3SsnPorch`, `ScreenPorch`, `PoolArea`, `SaleType` and `SaleCondition`.

And with the following interactions:

`LotArea:Neighborhood` and `X3SsnPorch:EnclosedPorch`.

Model assumptions

We now check whether the hypotheses of the model are satisfied, namely, linearity, normality and homocedasticity. Lets start with the linearity.

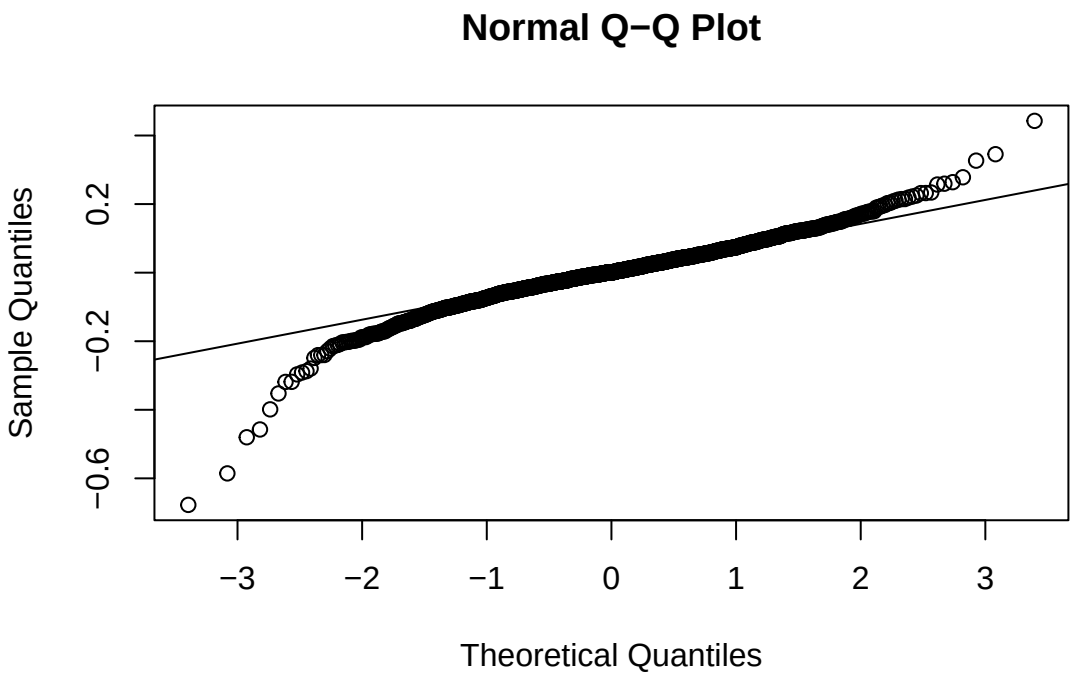


The model seems to be satisfy linearity judging from the plot.

One of the assumptions of linear models is the normality of the residuals, which we are going to check using a Shapiro test and a qq-plot.

```
##
##  Shapiro-Wilk normality test
##
## data:  mod3$residuals
```

```
## W = 0.94855, p-value < 2.2e-16
```



As we can see in the plot and in the Shapiro test, we can not say that the distribution of the residual is normal. Hence the normality assumption is not fulfilled.

The last assumption we are going to test is homocedasticity, that is, that the residuals have equal variance. We can check this in the plot we did for checking linearity (residuals vs. fitted): we can see that the residuals are more clumped together in the center, but they spread out more towards the extremes. This suggests that the data is not homocedastic. However, we can formally check this usign the Breusch-Paga test, the null hypothesis being that the errors are homocedastic.

```
##
##  studentized Breusch-Pagan test
##
## data:  mod3
## BP = 274.08, df = 202, p-value = 0.0005504
```

We can see that the p-value of the test is too small so we can conclude that the errors are not homocedastic.

Prediction and estimation of the error.

Now that we have developed and studied our linear model we can finally use it to make a prediction on the test dataset:

```
test_predicted_linear <- as.vector(predict(mod3, newdata = test_data))
```

In this competition the metric that will be used to check the accuracy of our model is the root mean squared error (RMSE). Although there is no way to know the RMSE of our prediction for certain, we can estimate it through cross-validation in the training dataset. In particular, we are going to do a 5-fold cross-validation.

```
## Linear Regression
##
## 1456 samples
```

```
##   45 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1165, 1165, 1164, 1164, 1166
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.1681212  0.8303591  0.08677521
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

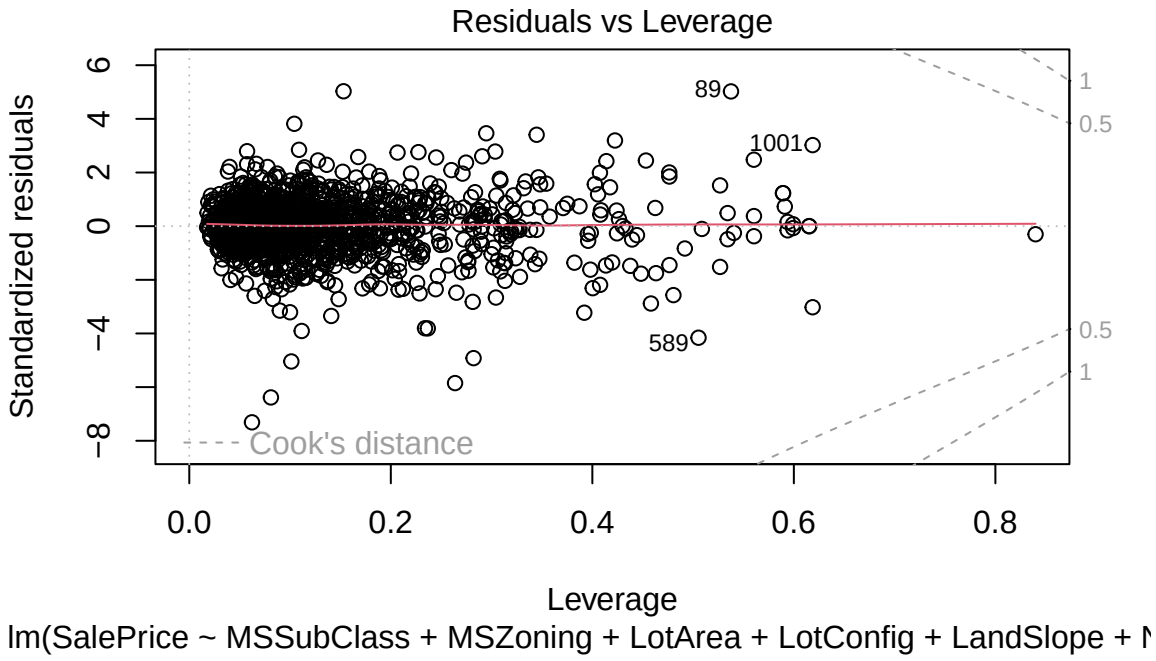
Each time the code is run we obtain an estimation of the RMSE of around $0.14 - 0.17$. This means that we can expect a relative error close to 1% which is reasonably accurate.

Now that we have finished our model we can check whether we have influential observations, obtaining the following result:

```
named integer(0)
```

As we can see, we do not have any influential observation so there is no need to remove more observations. This can also be seen in the following plot:

```
plot(mod3, 5)
```



As we can see there are some points with a relatively high leverage, but we do not need to remove them because they are not influential.

Influence of the variables

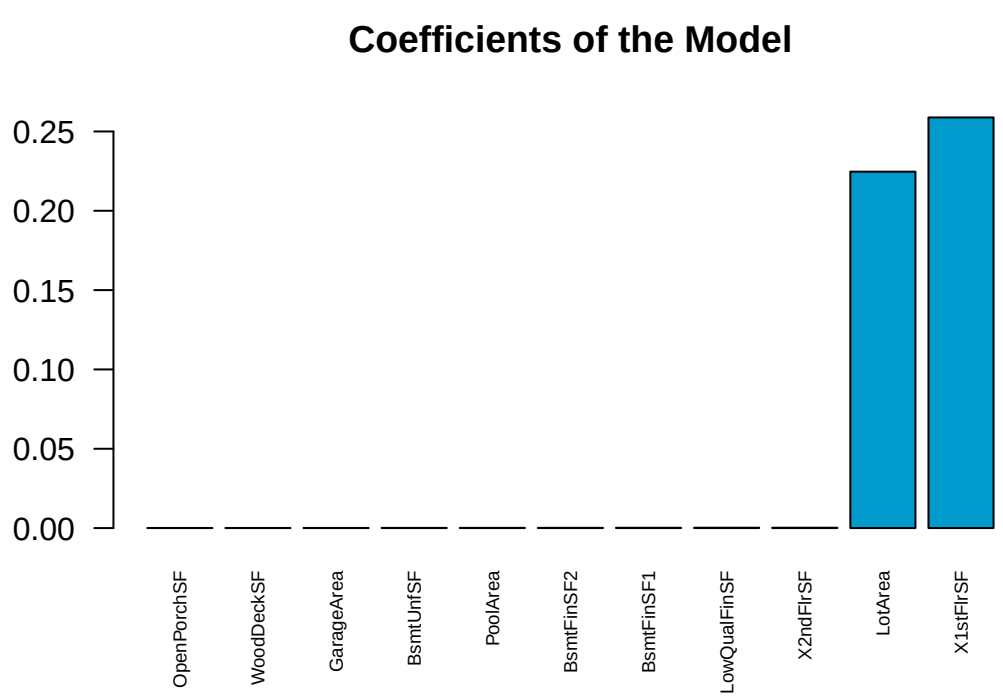
To end the section regarding the linear model, we are going to study the influence of the variables on the sale price.

```
## The variables with bigger coefficient are:
## 1. (Intercept): 7.4907
## 2. NeighborhoodIDOTRR: -4.0781
## 3. NeighborhoodBrkSide: -3.8631
## 4. NeighborhoodNridgHt: -2.9148
## 5. NeighborhoodEdwards: -2.8750
## 6. NeighborhoodMeadowV: -2.8645
## 7. NeighborhoodTimber: -2.7454
## 8. RoofMatlMembran: 2.7156
## 9. NeighborhoodOldTown: -2.6771
## 10. NeighborhoodNWAmes: -2.6526
## 11. NeighborhoodClearCr: -2.6520
## 12. NeighborhoodVeenker: -2.6426
## 13. NeighborhoodGilbert: -2.6198
## 14. NeighborhoodSomerst: -2.5731
## 15. RoofMatlMetal: 2.5656
## 16. RoofMatlWdShngl: 2.5328
## 17. RoofMatlTar&Grv: 2.4765
## 18. RoofMatlCompShg: 2.4542
## 19. RoofMatlRoll: 2.4402
## 20. NeighborhoodBrDale: -2.4139
## 21. RoofMatlWdShake: 2.3985
## 22. NeighborhoodStoneBr: -2.3822
## 23. NeighborhoodNoRidge: -2.3512
## 24. NeighborhoodSawyerW: -2.2952
## 25. NeighborhoodCrawfor: -2.1521
## 26. NeighborhoodCollgCr: -2.1182
## 27. NeighborhoodNAmes: -2.0722
## 28. NeighborhoodSawyer: -2.0641
## 29. NeighborhoodBlueste: -2.0025
## 30. NeighborhoodMitchel: -1.8900
## 31. NeighborhoodNPkVill: -1.8740
## 32. NeighborhoodSWISU: -1.7072
## 33. OverallQual10: 0.7309
## 34. OverallQual9: 0.6878
## 35. OverallQual8: 0.5815
## 36. MSZoningFV: 0.5371
## 37. MSZoningRH: 0.5161
## 38. OverallQual7: 0.5151
## 39. MSZoningRL: 0.4927
```

If we accept that the most influential variables are those with the biggest coefficients in absolute value, it is easy to see that the most influential categorical variables are `Neighborhood`, `RoofMatl`, `OverallQual` and `MSZoning`.

If we analyze Neighborhood, for instance, the coefficient of each category represents the average change in `SalePrice` when compared to the reference level (`Blmngtn`). All coefficients are negative so we can deduce that the reference level has the most expensive houses on average.

On the other hand, for quantitative variables the analysis is harder due to the different scales of each variable. However, this does not stop us from comparing the effect of predictors with the same unit. In particular, we are interested in which of the covariates related to the area of the house have the greatest effect in the model:



As we can see, the most influential variables related to area are by far `X1stFlorSF` and `LotArea`.

To finish this section we can review the most striking differences between the cheapest houses and the most expensive ones. In particular, we are going to compare de 40 most expensive houses with the 40 cheapest.

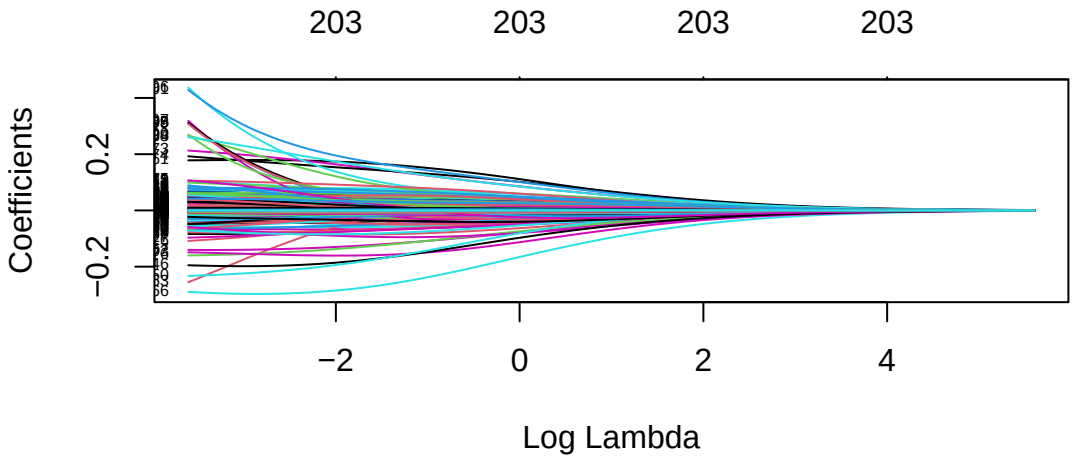
We can observe that for many quantitative variables (fireplace, second floor, half bathroom,...) the cheaper houses display a large proportion of zeroes. This suggests that the lack of certain features (garage, porch,...) drives the cost down, just like one would expect. This also indicates that the most expensive houses all have

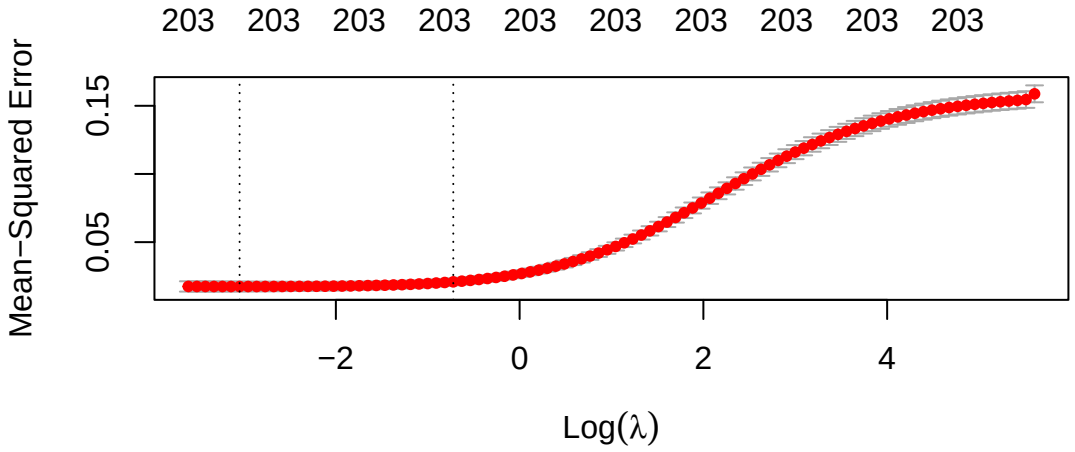
If we now compare the mean of the area variables we find the (not surprising) result that, on average, the cheaper houses are smaller than the most expensive ones: their rooms occupy less surface area on average. Moreover, the quality of the different assets (heating, kitchen, overall quality) are all smaller in the 40 cheapest homes.

```
##     Variables Mean_of_Min_40 Mean_of_Max_40
## 1     LotArea       8.723764       9.638359
## 2   BsmtUnfSF     491.750000     859.375000
## 3   HeatingQC       3.175000       4.950000
## 4   X1stFlrSF       6.588809       7.551957
## 5 KitchenQual       2.775000       4.700000
## 6    FullBath       1.000000       2.175000
## 7    HalfBath       0.075000       0.725000
## 8 OverallQual       3.925000       8.925000
## 9 OverallCond       4.475000       5.225000
```

Ridge regression

We are going to use the linear model we have just developed as a base to build a ridge model, our motivation being that the penalization terms can perhaps improve our fit. We will also estimate the RMSE through cross-validation in order to determine which model has the most predictive power. Firstly, we select the best ridge parameter for the final linear model.





We then select the optimal value of lambda, this value being in our case:

```
## [1] 0.04746507
```

We can now build our model using this lambda. We obtain a RMSE of:

```
## [1] 0.5854533
```

Which is around five times worse than what we obtained with the linear model. Therefore, it is less accurate than our linear model. This combined with the harder interpretability of the ridge model leads us to discard the ridge regression.

Generalized Additive Models

When checking the assumptions of the linear model we saw that both the normality and homocedasticity of the residuals were not fulfilled. This suggests that we either need to transform the data further or consider a different model. After exhaustive work we have discarded the first option, so that leaves us with the second one. In particular, we are going to check whether the inclusion of smooth terms in the model can lead us to more accurate predictions: we are going to use the tools of generalized additive models to see whether we can

improve our fit.

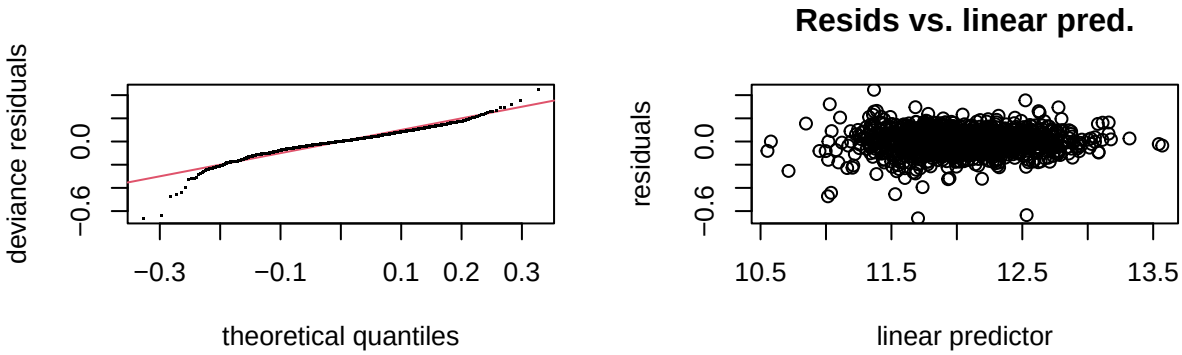
The point of this model is just raw accuracy: we are no longer interested in interpetability (we have developped a linear model for that). This gives us the freedom of adding some (highly correlated) variables to our data that we suspect can improve our fit. Namely, we are going to add the total surface (`TotalSf`), the total floor surface (`TotalFloorSf`) and the total porch surface (`TotalPorchSf`). To compensate for this we may drop some of the features that contribute to each of the new variables.

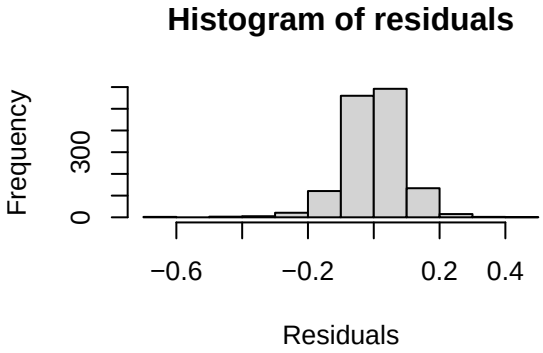
```
train_data$TotalSf <- train_data$TotalBsmtSF + train_data$GrLivArea
train_data$TotalFloorSf <- train_data$X1stFlrSF + train_data$X2ndFlrSF
train_data$TotalPorchSf <- train_data$OpenPorchSF + train_data$EnclosedPorch +
  train_data$X3SsnPorch + train_data$ScreenPorch

test_data$TotalSf <- test_data$TotalBsmtSF + test_data$GrLivArea
test_data$TotalFloorSf <- test_data$X1stFlrSF + test_data$X2ndFlrSF
test_data$TotalPorchSf <- test_data$OpenPorchSF + test_data$EnclosedPorch +
  test_data$X3SsnPorch + test_data$ScreenPorch
```

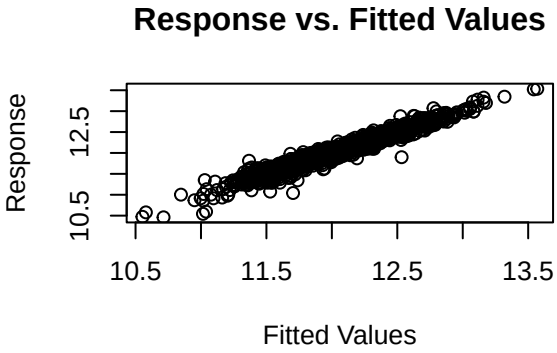
We are now ready to create our final model using the linear model we built as a base. We are going to add some smooth terms to most of the variables related to the surface area of the house. We also remove the interaction terms which will make the model worse by increasing the RMSE. Hence our final model will have the following variables:

1. Linear terms: `MSZoning`, `LotConfig`, `LandSlope`, `Neighborhood`, `Condition1`, `Condition2`, `OverallQual`, `OverallCond`, `YearBuilt`, `YearRemodAdd`, `RoofMatl`, `Exterior1st`, `MasVnrArea`, `Foundation`, `BsmtExposure`, `BsmtUnfSF`, `Heating`, `HeatingQC`, `CentralAir`, `BsmtFullBath`, `FullBath`, `HalfBath`, `KitchenAbvGr`, `KitchenQual`, `Functional`, `Fireplaces`, `GarageType`, `EnclosedPorch`, `ScreenPorch`, `PoolArea`, `SaleType`, `SaleCondition` and `X3SsnPorch`.
2. Smooth terms: `TotalSf`, `TotalFloorSf`, `TotalPorchSf`, `TotalBsmtSF`, `BsmtFinSF1`, `BsmtFinSF2`, `X2ndFlrSF`, `GarageArea` and `LotArea`.

```
##     GCV.Cp
## 0.01048118
```








```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 11 iterations.
## The RMS GCV score gradient at convergence was 1.289517e-08 .
## The Hessian was positive definite.
## Model rank =  201 / 203
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                    k'   edf k-index p-value
## s(TotalSf)      6.000 0.729    0.99    0.34
## s(TotalFloorSf) 6.000 0.784    1.02    0.83
## s(TotalPorchSf) 8.000 7.233    1.01    0.59
## s(TotalBsmtSF)  6.000 0.565    0.97    0.10
## s(BsmtFinSF1)   6.000 2.718    0.98    0.21
## s(BsmtFinSF2)   6.000 1.636    0.98    0.20
## s(X2ndFlrSF)    6.000 0.784    0.99    0.35
## s(GarageArea)   6.000 4.238    1.06    0.99
## s(LotArea)      6.000 1.000    1.02    0.81
```

We can see that the contribution of the smooth terms is significant. Moreover, we achieve a RMS of around 0.01 (relative error of approximately 0.1%). This is a great improvement on the linear model. Therefore, this is the final model we are going to use to make our predictions.

```
gam_predicted <- predict.gam(gam_model, test_data)
SalePrice <- exp(gam_predicted)
```

```r
Id <- seq(1461, length(SalePrice) + 1460)
submission <- data.frame(Id, SalePrice)
write.csv2(submission, "price_submission.csv", row.names=FALSE)
```

Conclusions

In this report we have:

1. Succesfully cleaned the data we have worked with.
2. Analyzed its different variables studying their outliers, multicollinearity, influence on the response...
3. Developed three different models to predict housing prices using the data given to us.

The main results of our work are:

1. The development of a linear model with an estimated RMSE of 0.1 on the logarithm of `SalePrice`, and the exhaustive interpretation of its parameters and the chosen variables.
2. The development of a more accurate (albeit less interpretable) GAM model with an estimated RMSE of 0.01.

The GAM model we have built is the one we are going to use to predict the prices (due to its higher accuracy), but the linear regression we performed on the data has given us greater insights into what are the features that influence the price of a house the most. Namely, we have reached the following conclusions:

1. The most influential variables are `Neighborhood`, `RoofMatl`, `X1stFlorSF` and `LotArea`.
2. Cheaper homes are all smaller on average than the most expensive ones.
3. Cheaper homes in general lack certain assets that expensive houses do not, like fireplaces, half bathrooms,...

Perhaps a line of further research could be the exploration of the effect of new variables (number of baths, boolean variables that indicate the presence or absence of certain features,...) or the use of more sophisticated models. Moreover, it may be the case that a combination of the presented models can further improve accuracy. For example, we could take the mean:

```r
# Not used to get final predictions
combinated_preds <- (test_predicted_linear + gam_predicted + pred_ridge)/3
```

However, we need to adapt a cross validation scheme in order to estimate its rmse. In any case, our work constitutes a solid first approximation to the topic of predictive models.