ANALYSIS OF HOUSE SALES PRICES

Created by: Carlos Estevez: cestevez@smu.edu and Jacob Lijo: lijoj@smu.edu

ANALISIS 1: RESTATEMENT OF PROBLEM NORTH AMES, EDWARDS, AND BROOKSIDE

Century 21 Ames is an important company of real estate that sell houses in Ames Iowa in the US. They only sell houses in the NAmes, Edwards, and BrkSide neighborhoods. One of the most critical aspects of the Real State business is the sales price estimation by different parameters. Century 21 Ames is interested in knowing if Housing Sales Price is related to the square footage of the living area of the house (GrLIvArea) and if the Sales Price (and its relationship to square footage) depends on which neighborhood the house is located in. Our job is to build a model to address these questions.

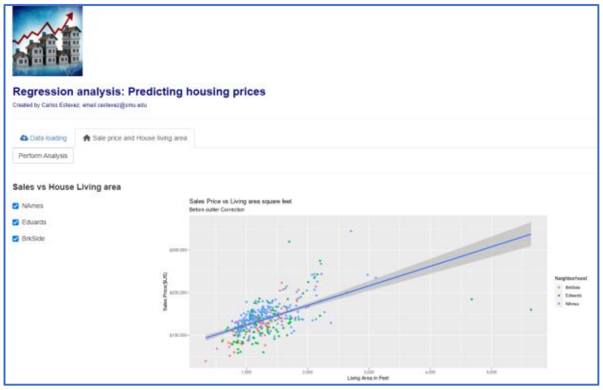
BUILDING AND FITTING THE MODEL

We will build a multiple linear regression model. We have two different predictors and one response variable:

- Sales Price: House sales price in dollars
- GrLivArea: Above grade (ground) living area square feet
- Neighborhood: We will analyze only three neighborhoods: NAmes, Edwards and BrkSide.

We have a dataset: train.csv with 1460 observations.

CHECKING ASSUMPTIONS



Regression analysis: Predicting housing prices

Created by Carlos Estevez, email cestevez@smu.edu

Sales vs House Living area

☑ NAmes
☑ Eduards
☑ BrkSide

Figure 1: Assumptions

You can access our Shiny App for more details about the assumptions. More details: https://estevez.shinyapps.io/HousePricePrediction/

The first impression is that it seems there's a linear relationship between the House Living area and the Housing Sales Price, however, it looks like there are a few outliers. One of the reasons why the Correlation Coefficient is too low could be these extreme outliers. Let's move forward and create a Tentative Model to explore the Residuals, Cooks'D, and Leverage Analysis

TENTATIVE MODEL

*We will leave Neighborhood out this time. We want to focus on the relationship between Sales Prices and Housing living Areas.

Predicted Sales = 78205.578 + 45.979*GAREA

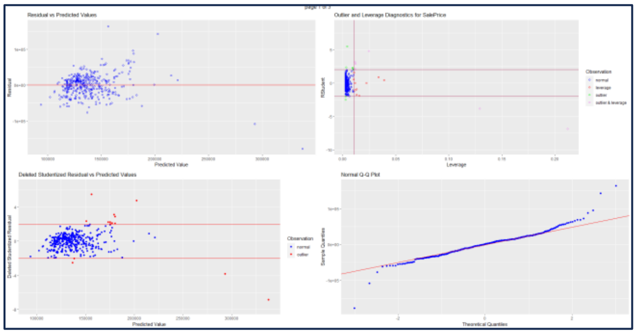


Figure 2:Checking the Assumptions

As we can see in the previous image there's evidence of influential cases (Outliers). We will remove these outliers an d address the assumptions afterward.

Initially, we will remove the following observations: -131,-339,-169,-190.

After tweaking the model and adding the neighborhood variable, the results are much better

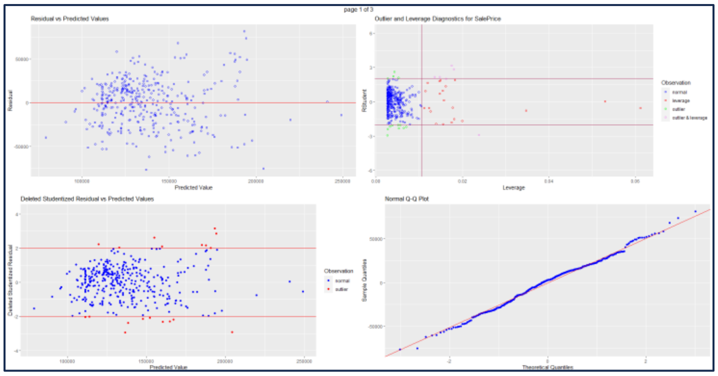


Figure 3: Checking the Assumptions

The correlation coefficient improved from 0.54 to 0.67. You can also see the details in our shinny app.

CHECKING THE ASSUMPTIONS

As you can see in the above image the results are much better.

Linearity: It seems there's evidence of linearity in the first plot. As you can see in the first plot. The residuals also reinforce this statement since they are randomly scattered.

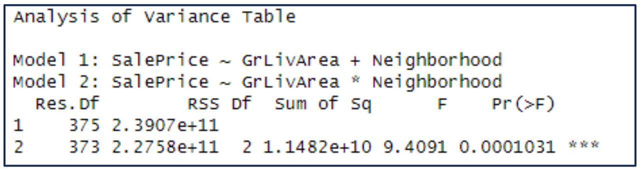
Normality: According to the visual information shown by the qq-plot, there's evidence of normality.

Constant variance: There's evidence of constant variance as the residuals vs fitted values plot shows. The observations are randomly scattered around the mean.

Independence: We will assume independence

THE FULL MODEL AND REDUCED MODEL

We decided on the Full Model because it is more robust than the reduced model. We use an ANOVA to compare both models.

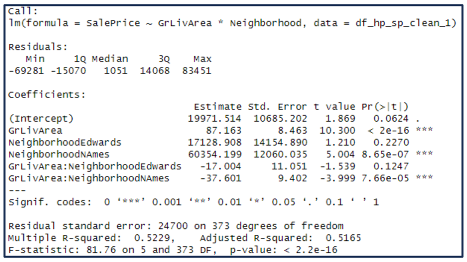


```
Analysis of Variance Table

Model 1: SalePrice ~ GrLivArea + Neighborhood
Model 2: SalePrice ~ GrLivArea * Neighborhood
  Res.Df        RSS Df  Sum of Sq      F    Pr(>F)    
1    375 2.3907e+11                                   
2    373 2.2758e+11  2 1.1482e+10 9.4091 0.0001031 ***
```

Figure 4: ANOVA

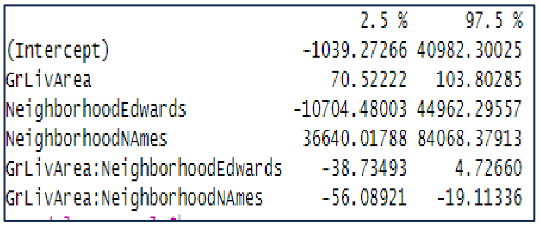
As you can see, in the results the Full Model has a lower Sum of Squared residuals and a lower R-Square.

ESTIMATES RESULTS AND EQUATION



```
Call:
lm(formula = SalePrice ~ GrLivArea * Neighborhood, data = df_hp_sp_clean_1)

Residuals:
   Min     1Q Median     3Q    Max 
-69281 -15070   1051  14068  83451 

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)    
(Intercept)                 19971.514  10685.202   1.869   0.0624 .  
GrLivArea                      87.163      8.463  10.300  < 2e-16 ***
NeighborhoodEdwards         17128.908  14154.890   1.210   0.2270    
NeighborhoodNAmes           60354.199  12060.035   5.004 8.65e-07 ***
GrLivArea:NeighborhoodEdwards  -17.004     11.051  -1.539   0.1247    
GrLivArea:NeighborhoodNAmes    -37.601      9.402  -3.999 7.66e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 24700 on 373 degrees of freedom
Multiple R-squared:  0.5229,	Adjusted R-squared:  0.5165 
F-statistic: 81.76 on 5 and 373 DF,  p-value: < 2.2e-16
```

Figure 5: Estimate results



```
                                     2.5 %      97.5 %
(Intercept)                    -1039.27266 40982.30025
GrLivArea                         70.52222   103.80285
NeighborhoodEdwards           -10704.48003 44962.29557
NeighborhoodNAmes              36640.01788 84068.37913
GrLivArea:NeighborhoodEdwards    -38.73493     4.72660
GrLivArea:NeighborhoodNAmes      -56.08921   -19.11336
```

Figure 6: Intervals

*Predicted Sales price = B0+B2*NHOO+(GAREA+B3*NHOO) *B1*

*Predicted Sales price = B0+B2*NHOOE+B4*NHOON+B1*GAREA+B3*(NHOOE*GAREA) +B5*(NHOON*GAREA)*

FINAL EQUATION:

GAREA →Living area in square feet

NHOON →Neighborhood

Predicted Sales price { SalePrice | GAREA, NHOON } = 19,971.514+17,128.908*NHOOE+60,354.199*NHOON+87.163*GAREA-17.004*(NHOOE*GAREA)-37.601*(NHOON*GAREA)

INDIVIDUAL EQUATIONS:

Brookside -->***19,971.514+87.163*GAREA***

Edwards-->19971.514+17128.908+87.163*GAREA-17.004*(GAREA) = ***37,100.42+70.159*GAREA***

North Ames -->19971.514+60354.199+87.163*GAREA-37.601*(NHOON*GAREA) = ***80,325.71+49.562*GAREA***

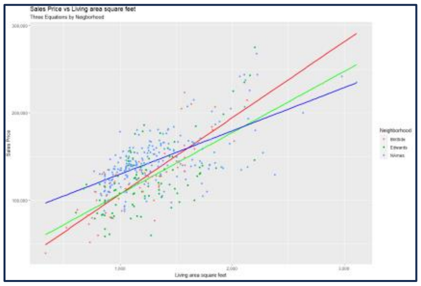


Figure 7:Equations Neighborhood

CONCLUSION

*Predicted Sales price = 19,971.514+17,128.908*NHOOE+60,354.199*NHOON+87.163*GAREA-17.004*(NHOOE*GAREA)-37.601*(NHOON*GAREA)*

B0 Intercept (Neighborhood Brookside) and B1 Slope (Neighborhood Brookside)

The average price for houses located in the Brookside neighborhood for living areas between 334 and 3112 square feet (without taking into consideration observations: 131,339,169,190) is estimated to be 19,971 dollars. These results, however, are not significant because of extrapolation.

The important inference is that for the houses located in this neighborhood in Ames is that for every 100-square-foot increment in the house area, there's an estimated increase of 8,716 dollars in the price of the house. In other words, the price of the house is significantly impacted by the House's area, as p-value < 0.0001, sig. level = 0.05. A 95% confidence interval for this increase is (7,052,10,380) dollars.

B2 Intercept (Neighborhood Edwards) and B3 Slope (Neighborhood Edwards)

The average price for houses located in the Edwards neighborhood for living areas between 334 and 3112 square feet (without taking into consideration observations: 131,339,169,190) is estimated to be 17,120 dollars. These results, however, are not significant because of extrapolation.

For every one-square-foot increment in the house area for this neighborhood, there's an estimated increase of 17 dollars in the price of the house. These results are not significant.

B4 Intercept (Neighborhood North Ames) and B5 Slope (Neighborhood North Ames)

The average price for houses located in the Brookside neighborhood for living areas between 334 and 3112 square feet (without taking into consideration observations: 131,339,169,190) is estimated to be 60,354.2 dollars, keeping fixed the other variables. These results, however, are not significant because of extrapolation.

The important inference is that for the houses located in this neighborhood in Ames is that for every 100 square-foot increment in the house area, there's an estimated decrease of 3,770 dollars in the price of the house, holding the other predictors fixed(It doesn’t mean that the price of the houses diminishes in the neighborhood with respect with the housing area, it means that the price of the housing ratio is 3,770 dollars lower with respect with the reference in this case Brookside neighborhood).In other words, the price of the house is significantly impacted by the House’s area, as p-value < 0.0001, sig. level = 0.05. A 95% confidence interval is (-5600, -1900) dollars.

ANALISIS 2: ANALYSIS OF HOUSE SALES PRICES COMPLETE MODEL: STATEMENT OF PROBLEM

Century 21 Ames is an important company of real estate that sell houses in Ames Iowa in the US. We need to build a model for predicting sales prices of Homes in Ames Iowa in the US. We need to build four models and clearly explain which one is the best and why. The idea is to use R2, CV Press, and Kaggle Score to evaluate the model’s performance

BUILDING AND FITTING THE MODEL

For setting up the model we are counting on two datasets: a training dataset: with 1460 observations and a testing dataset with 1459 observations. In addition, we have 79 explanatory variables or potential predictors. We will perform the following steps to build our models.

- 1-We will conduct an EDA analysis (Including graphical display and correlation coefficients) and determine the most influential predictors and discard obvious redundancies
- 2-Consider transformation if it is necessary
- 3-Examine the residuals after establishing a tentative model, performing further transformations and identifying outliers
- 4-Use a variable selection automation method for finding a suitable subset of explanatory variables
- 5-Build the model

VARIABLE SELECTION

We decided to begin with the analysis of the most important continuous variables (Discrete and Continuous).

Continuous variables:

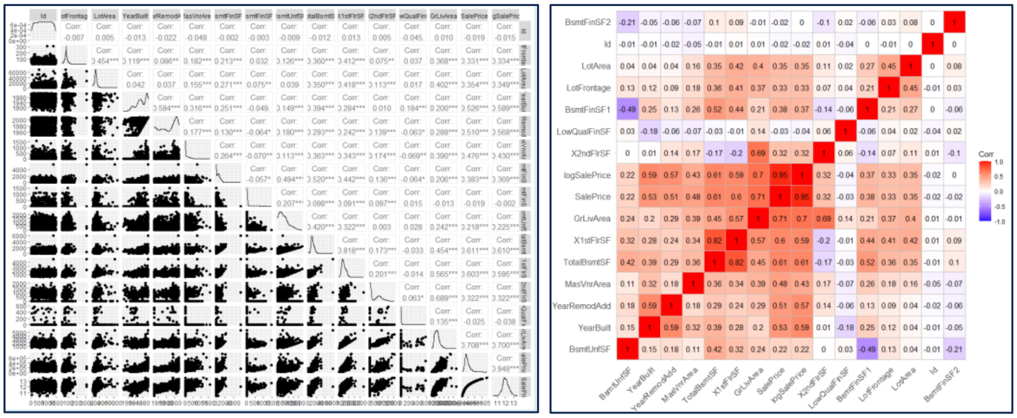
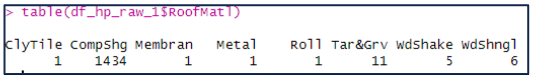


Figure 8.1 : Scatter matrix plot and correlation matrix

After dealing with the initial outliers that we visually identified in the above images, we could established the relationship between Sales Price and the most relevant continuous variables. The most important continuous variables of our model in terms of a clear linear relationship with the response are **GrLivArea, TotalBsmtSF, 1stFlrSF, GarageCars, GarageArea, WoodDeckSF, MasVnrArea, and LotArea.**

Categorical variables

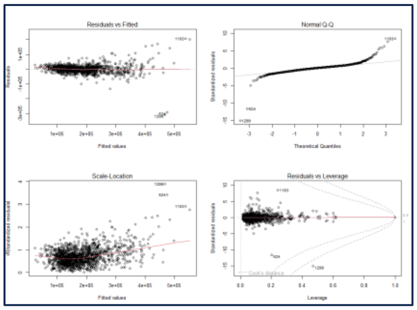
This part was more difficult. What we did was to analyze some of categorical variables, using different plots(Scatter plots and bar char). At the end some of variables were removed for the following reasons: Lack of observations or redundancy. For example, the variable Roof Material has most of the observations in one the categories, in this case it lacks of importance, so we removed it from our custom model.

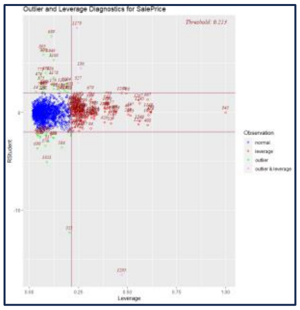


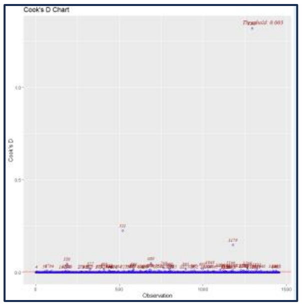
```
> table(df_hp_raw_1$RoofMatl)

ClyTile CompShg Membran   Metal    Roll Tar&Grv WdShake WdShngl
      1    1434       1       1       1      11       5       6
```

We will leave the Excel file with our comments. There’s a column called Remove. It indicates the categorical variable was removed for one of the mentioned reasons.

Explanatory
Variables.xlsx

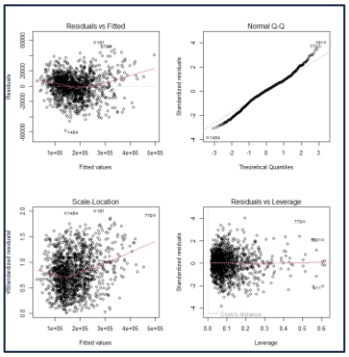


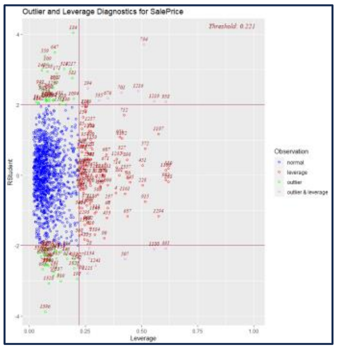


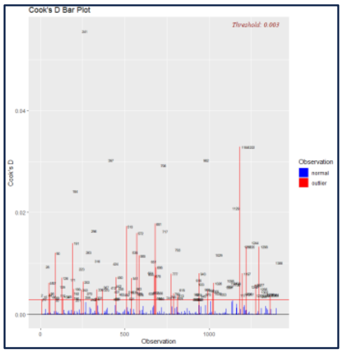


As you can see in the above image there are many assumptions violated. We addressed some of the assumptions by dealing with influential cases or outliers. The RSStudent vs Leverage provides a great tool to identify the outliers.

CHECKING THE ASSUMPTIONS







We used our tentative model to check the assumptions and deal with some influential cases using Student Redisuals, Levarage and Cook's D

As you can see in the above image the results are much better.

Linearity: It seems there's evidence of linearity in the first plot. As you can see in the first plot. The residuals also reinforce this statement since they are randomly scattered.

Normality: According to the visual information shown by the qq-plot, there's evidence of normality.

Constant variance: There's evidence of constant variance as the residuals vs fitted values plot shows. The observations are randomly scattered around the mean.

Independence: We will assume independence

COMPETING MODELS

We will use K-fold cross-validation with 20 interactions

Predictive Models	Adjusted R2	CV PRESS(RMSE)	Kaggle Score
Forward	0.7828044	38,725.61	0.16742
Backward	0.79872	37952.43	0.16921
Stepwise	0.8368288	30,654.8	0.17195
Custom	0.93539	17,250.65	0.17219

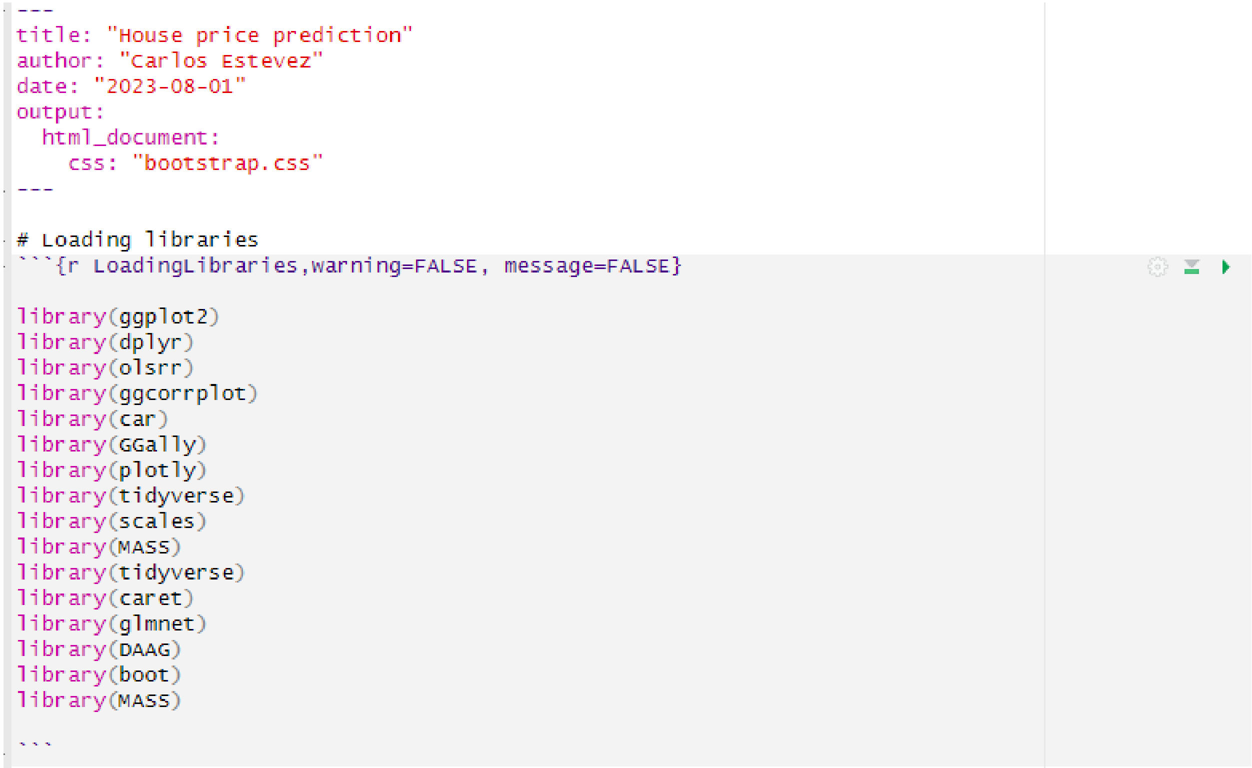
CHOOSING FINAL MODEL

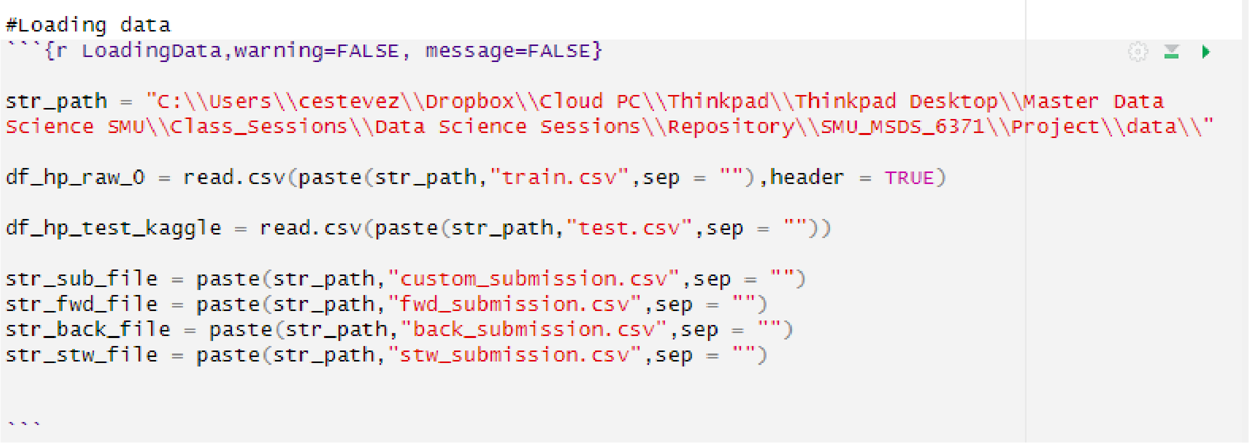
We selected the Custom Model as the best model. In the custom model, we excluded different predictors that we believed were redundant, and more importantly, we performed many data analyses for each variable that we analyzed that were not relevant in terms of predicting Sales Price. The other models were as good as the Custom Models when we tested with the training dataset, however as soon as we implemented cross-validation the R-Square and RMSE were significantly reduced. In general, the Custom Model was more consistent and accurate.

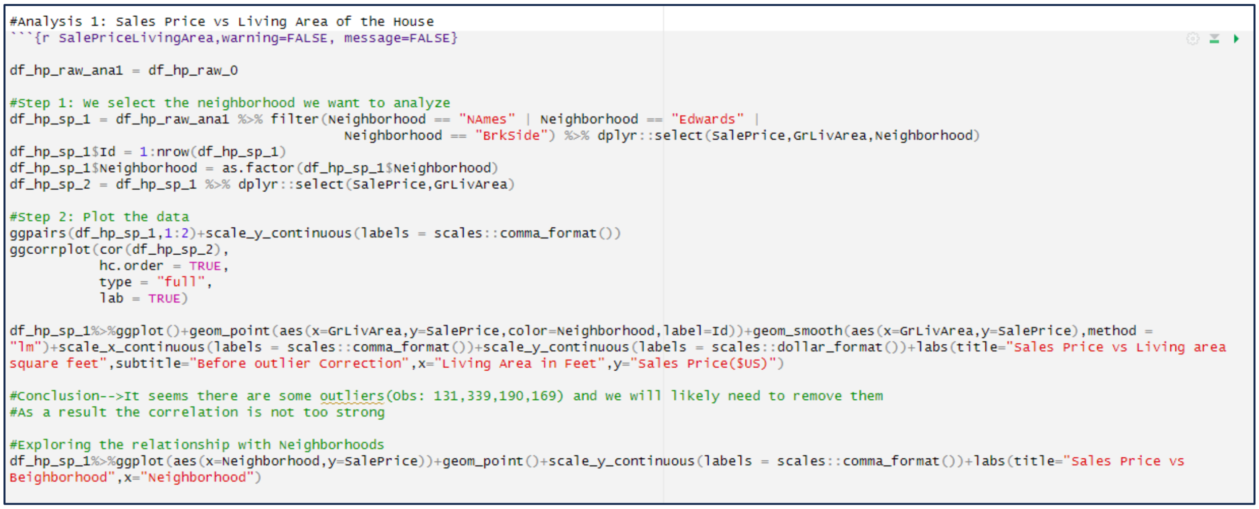
CONCLUSION

In general, we were able to build a Multiple Linear Regression model to predict housing sales prices in Ames Iowa in the US for the company Century 21. We used different statistical techniques to improve the accuracy of this model such as exploring data using different kinds of plots(Scatterplot, Histograms, and Bar chart), variable selection(Continuous and Categorical), building a tentative model, exploring the residuals, leveraging and Cook's D plots and in that sense, we identified the influential cases and made the different adjustments for fulfilling the assumptions of the model(Normality, Linearity, Constant Variance, and Independence). Finally, we built a model capable of predicting the Sales Price based in the region of Ames Iowa on a certain number of predictors provided by the company Century 21 with a high level of precision.

APPENDIX

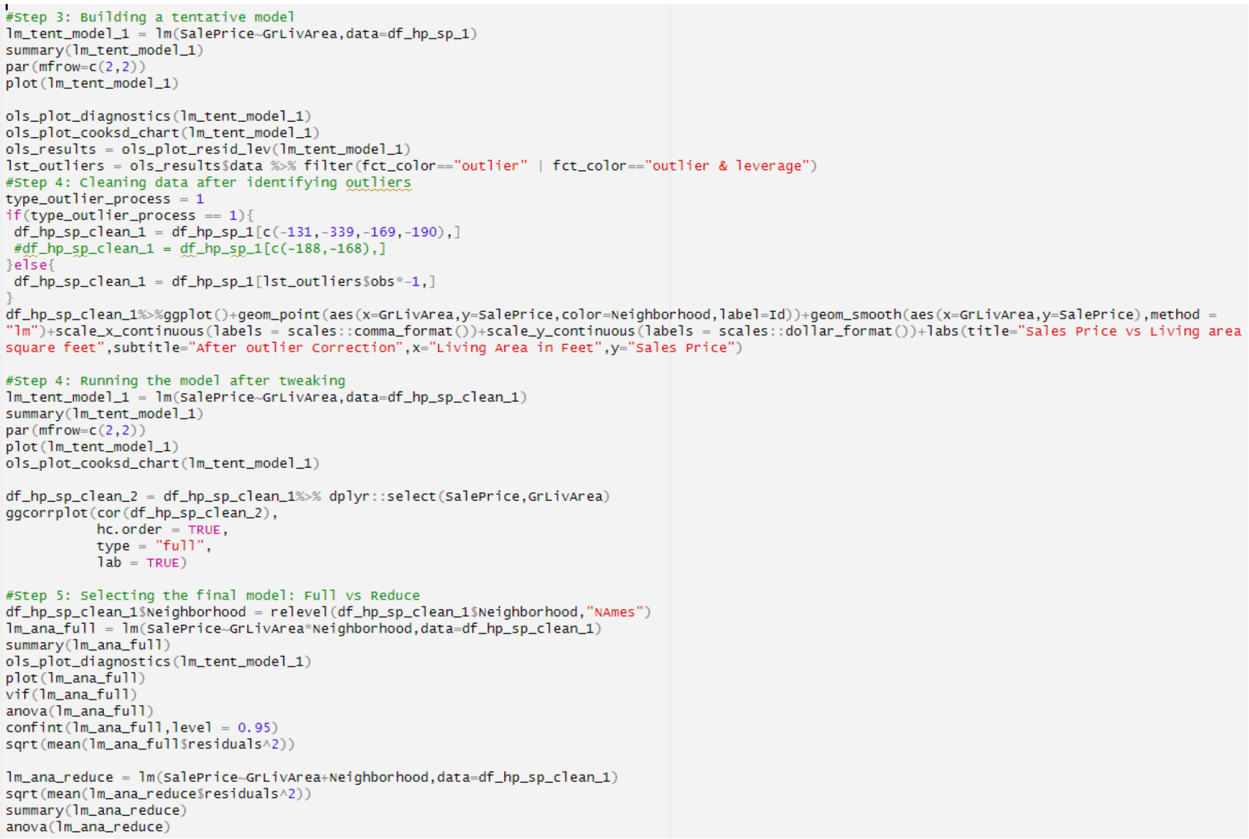


~~~
---
title: "House price prediction"
author: "Carlos Estevez"
date: "2023-08-01"
output:
  html_document:
    css: "bootstrap.css"
---

# Loading libraries
```{r LoadingLibraries,warning=FALSE, message=FALSE}

library(ggplot2)
library(dplyr)
library(olsrr)
library(ggcorrplot)
library(car)
library(GGally)
library(plotly)
library(tidyverse)
library(scales)
library(MASS)
library(tidyverse)
library(caret)
library(glmnet)
library(DAAG)
library(boot)
library(MASS)

~~~
```

Figure 9: Loading the libraries



```
#Loading data
```{r LoadingData,warning=FALSE, message=FALSE}

str_path = "C:\\Users\\cestevez\\Dropbox\\Cloud PC\\Thinkpad\\Thinkpad Desktop\\Master Data
Science SMU\\Class_Sessions\\Data Science Sessions\\Repository\\SMU_MSDS_6371\\Project\\data\\"

df_hp_raw_0 = read.csv(paste(str_path,"train.csv",sep = ""),header = TRUE)

df_hp_test_kaggle = read.csv(paste(str_path,"test.csv",sep = ""))

str_sub_file = paste(str_path,"custom_submission.csv",sep = "")
str_fwd_file = paste(str_path,"fwd_submission.csv",sep = "")
str_back_file = paste(str_path,"back_submission.csv",sep = "")
str_stw_file = paste(str_path,"stw_submission.csv",sep = "")


```
```

Figure 10: Loading the Datasets (Training and Testing)

ANALYSIS 1: THE NEXT SOURCE CODE BELONGS TO PROBLEM 1
```



~~~
#Analysis 1: Sales Price vs Living Area of the House
```{r SalePriceLivingArea,warning=FALSE, message=FALSE}

df_hp_raw_ana1 = df_hp_raw_0

#Step 1: We select the neighborhood we want to analyze
df_hp_sp_1 = df_hp_raw_ana1 %>% filter(Neighborhood == "NAmes" | Neighborhood == "Edwards" |
                                         Neighborhood == "BrkSide") %>% dplyr::select(SalePrice,GrLivArea,Neighborhood)
df_hp_sp_1$Id = 1:nrow(df_hp_sp_1)
df_hp_sp_1$Neighborhood = as.factor(df_hp_sp_1$Neighborhood)
df_hp_sp_2 = df_hp_sp_1 %>% dplyr::select(SalePrice,GrLivArea)

#Step 2: Plot the data
ggpairs(df_hp_sp_1,1:2)+scale_y_continuous(labels = scales::comma_format())
ggcorrplot(cor(df_hp_sp_2),
           hc.order = TRUE,
           type = "full",
           lab = TRUE)

df_hp_sp_1%>%ggplot()+geom_point(aes(x=GrLivArea,y=SalePrice,color=Neighborhood,label=Id))+geom_smooth(aes(x=GrLivArea,y=SalePrice),method =
"lm")+scale_x_continuous(labels = scales::comma_format())+scale_y_continuous(labels = scales::dollar_format())+labs(title="Sales Price vs Living area
square feet",subtitle="Before outlier Correction",x="Living Area in Feet",y="Sales Price($US)")

#Conclusion-->It seems there are some outliers(Obs: 131,339,190,169) and we will likely need to remove them
#As a result the correlation is not too strong

#Exploring the relationship with Neighborhoods
df_hp_sp_1%>%ggplot(aes(x=Neighborhood,y=SalePrice))+geom_point()+scale_y_continuous(labels = scales::comma_format())+labs(title="Sales Price vs
Beighborhood",x="Neighborhood")
~~~

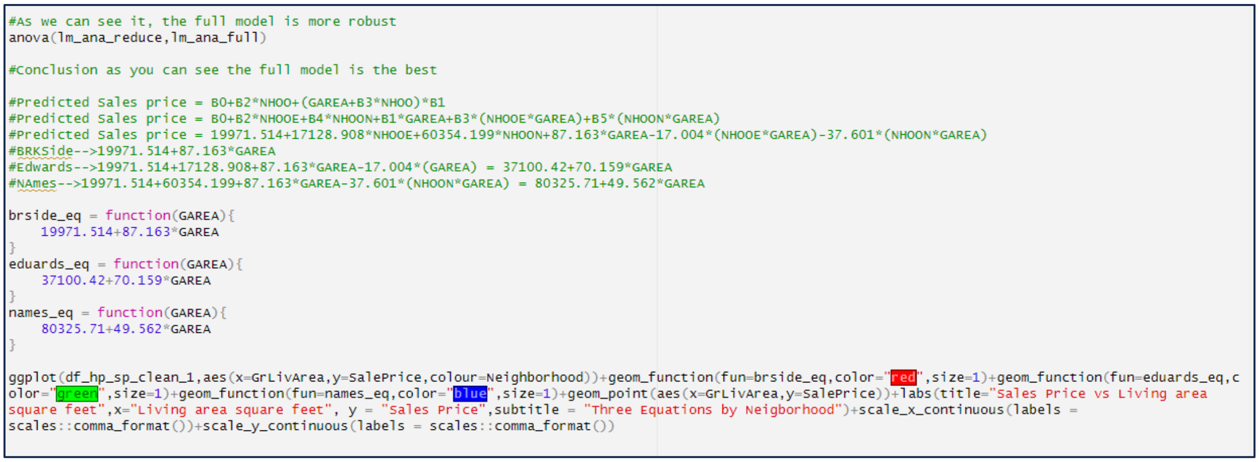
Figure 11: Filtering the data for three neighborhoods



```
#Step 3: Building a tentative model
lm_tent_model_1 = lm(SalePrice~GrLivArea,data=df_hp_sp_1)
summary(lm_tent_model_1)
par(mfrow=c(2,2))
plot(lm_tent_model_1)

ols_plot_diagnostics(lm_tent_model_1)
ols_plot_cooksd_chart(lm_tent_model_1)
ols_results = ols_plot_resid_lev(lm_tent_model_1)
lst_outliers = ols_results$data %>% filter(fct_color=="outlier" | fct_color=="outlier & leverage")
#Step 4: Cleaning data after identifying outliers
type_outlier_process = 1
if(type_outlier_process == 1){
 df_hp_sp_clean_1 = df_hp_sp_1[c(-131,-339,-169,-190),]
 #df_hp_sp_clean_1 = df_hp_sp_1[c(-188,-168),]
}else{
 df_hp_sp_clean_1 = df_hp_sp_1[lst_outliers$obs*-1,]
}
df_hp_sp_clean_1%>%ggplot()+geom_point(aes(x=GrLivArea,y=SalePrice,color=Neighborhood,label=Id))+geom_smooth(aes(x=GrLivArea,y=SalePrice),method =
"lm")+scale_x_continuous(labels = scales::comma_format())+scale_y_continuous(labels = scales::dollar_format())+labs(title="Sales Price vs Living area
square feet",subtitle="After outlier Correction",x="Living Area in Feet",y="Sales Price")

#Step 4: Running the model after tweaking
lm_tent_model_1 = lm(SalePrice~GrLivArea,data=df_hp_sp_clean_1)
summary(lm_tent_model_1)
par(mfrow=c(2,2))
plot(lm_tent_model_1)
ols_plot_cooksd_chart(lm_tent_model_1)

df_hp_sp_clean_2 = df_hp_sp_clean_1%>% dplyr::select(SalePrice,GrLivArea)
ggcorrplot(cor(df_hp_sp_clean_2),
           hc.order = TRUE,
           type = "full",
           lab = TRUE)

#Step 5: Selecting the final model: Full vs Reduce
df_hp_sp_clean_1$Neighborhood = relevel(df_hp_sp_clean_1$Neighborhood,"NAmes")
lm_ana_full = lm(SalePrice~GrLivArea*Neighborhood,data=df_hp_sp_clean_1)
summary(lm_ana_full)
ols_plot_diagnostics(lm_tent_model_1)
plot(lm_ana_full)
vif(lm_ana_full)
anova(lm_ana_full)
confint(lm_ana_full,level = 0.95)
sqrt(mean(lm_ana_full$residuals^2))

lm_ana_reduce = lm(SalePrice~GrLivArea+Neighborhood,data=df_hp_sp_clean_1)
sqrt(mean(lm_ana_reduce$residuals^2))
summary(lm_ana_reduce)
anova(lm_ana_reduce)
```

Figure 12: Building the tentative model, tweaking and assembling final model

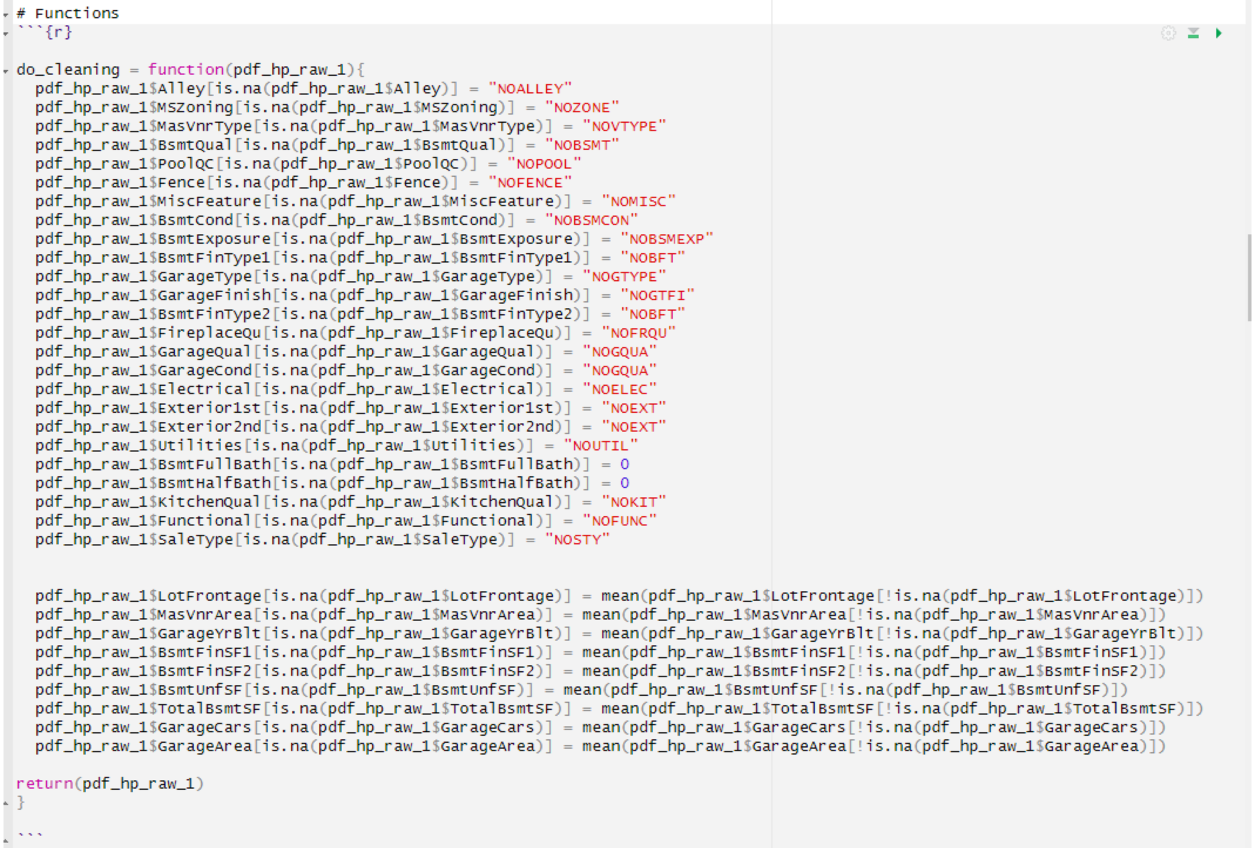
In this part we are also comparing the full model vs the reduced model. We ran an ANOVA for that.



```
#As we can see it, the full model is more robust
anova(lm_ana_reduce,lm_ana_full)

#Conclusion as you can see the full model is the best

#Predicted Sales price = B0+B2*NHOO+(GAREA+B3*NHOO)*B1
#Predicted Sales price = B0+B2*NHOOE+B4*NHOON+B1*GAREA+B3*(NHOOE*GAREA)+B5*(NHOON*GAREA)
#Predicted Sales price = 19971.514+17128.908*NHOOE+60354.199*NHOON+87.163*GAREA-17.004*(NHOOE*GAREA)-37.601*(NHOON*GAREA)
#BRKSide-->19971.514+87.163*GAREA
#Edwards-->19971.514+17128.908+87.163*GAREA-17.004*(GAREA) = 37100.42+70.159*GAREA
#NAmes-->19971.514+60354.199+87.163*GAREA-37.601*(NHOON*GAREA) = 80325.71+49.562*GAREA

brside_eq = function(GAREA){
    19971.514+87.163*GAREA
}
eduards_eq = function(GAREA){
    37100.42+70.159*GAREA
}
names_eq = function(GAREA){
    80325.71+49.562*GAREA
}

ggplot(df_hp_sp_clean_1,aes(x=GrLivArea,y=SalePrice,colour=Neighborhood))+geom_function(fun=brside_eq,color="red",size=1)+geom_function(fun=eduards_eq,c
olor="green",size=1)+geom_function(fun=names_eq,color="blue",size=1)+geom_point(aes(x=GrLivArea,y=SalePrice))+labs(title="Sales Price vs Living area
square feet",x="Living area square feet", y = "Sales Price",subtitle = "Three Equations by Neigborhood")+scale_x_continuous(labels =
scales::comma_format())+scale_y_continuous(labels = scales::comma_format())
```

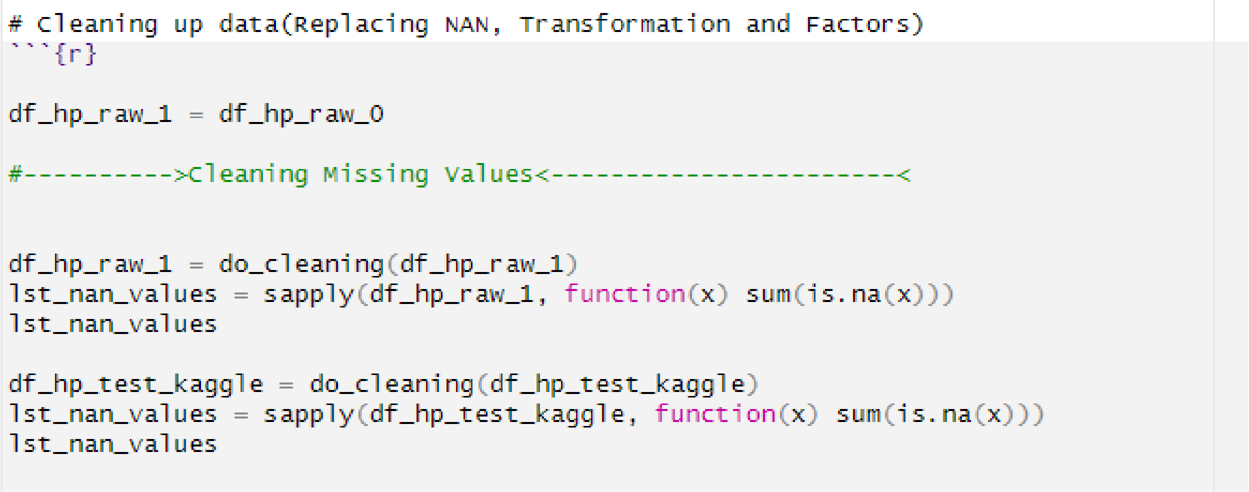
Figure 13: Formulating final equations

ANALYSIS 2: THE NEXT SOURCE CODE BELONGS TO PROBLEM 2



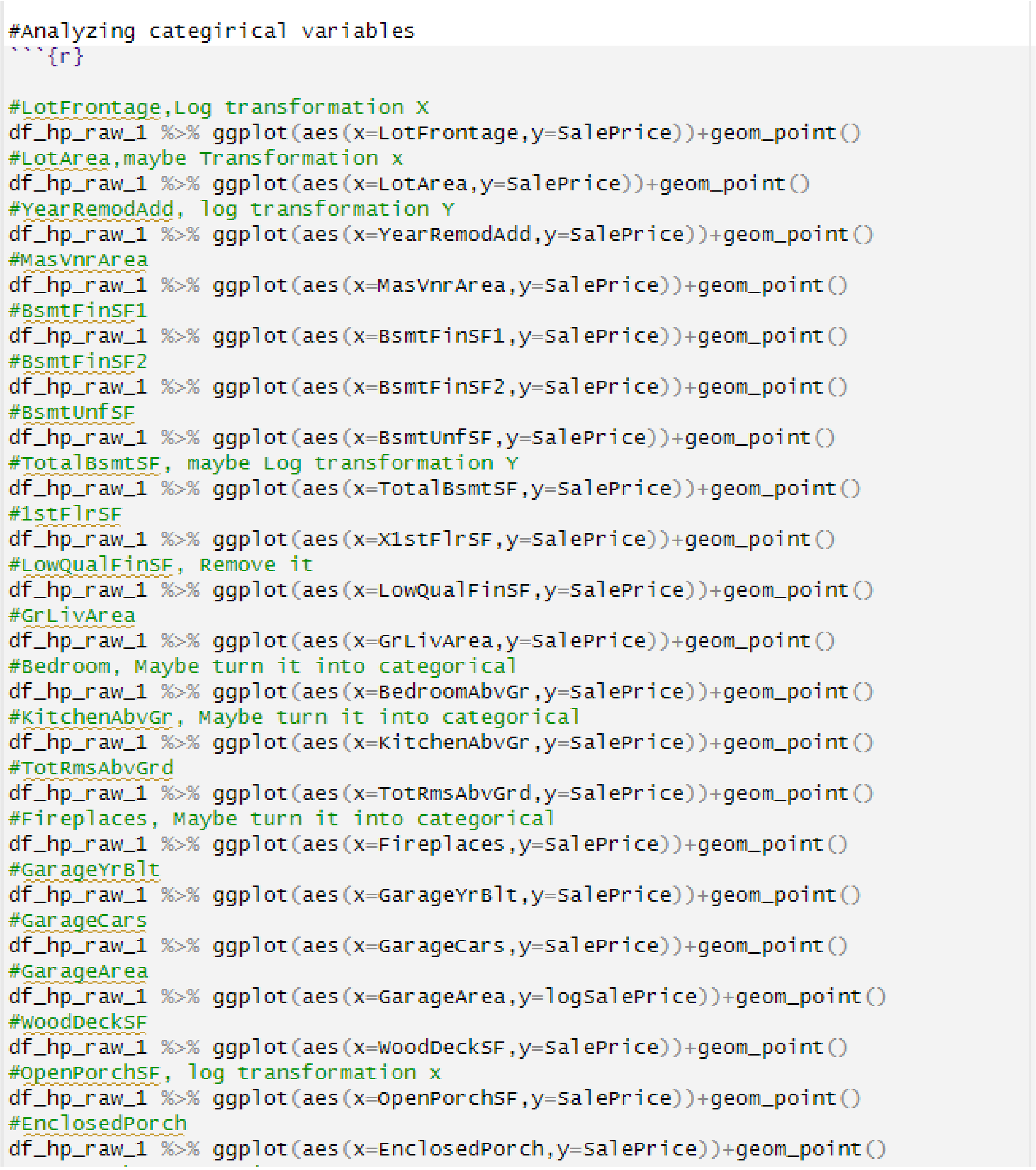
~~~
# Functions
```{r}

do_cleaning = function(pdf_hp_raw_1){
  pdf_hp_raw_1$Alley[is.na(pdf_hp_raw_1$Alley)] = "NOALLEY"
  pdf_hp_raw_1$MSZoning[is.na(pdf_hp_raw_1$MSZoning)] = "NOZONE"
  pdf_hp_raw_1$MasVnrType[is.na(pdf_hp_raw_1$MasVnrType)] = "NOVTYPE"
  pdf_hp_raw_1$BsmtQual[is.na(pdf_hp_raw_1$BsmtQual)] = "NOBSMT"
  pdf_hp_raw_1$PoolQC[is.na(pdf_hp_raw_1$PoolQC)] = "NOPOOL"
  pdf_hp_raw_1$Fence[is.na(pdf_hp_raw_1$Fence)] = "NOFENCE"
  pdf_hp_raw_1$MiscFeature[is.na(pdf_hp_raw_1$MiscFeature)] = "NOMISC"
  pdf_hp_raw_1$BsmtCond[is.na(pdf_hp_raw_1$BsmtCond)] = "NOBSMCON"
  pdf_hp_raw_1$BsmtExposure[is.na(pdf_hp_raw_1$BsmtExposure)] = "NOBSMEXP"
  pdf_hp_raw_1$BsmtFinType1[is.na(pdf_hp_raw_1$BsmtFinType1)] = "NOBFT"
  pdf_hp_raw_1$GarageType[is.na(pdf_hp_raw_1$GarageType)] = "NOGTYPE"
  pdf_hp_raw_1$GarageFinish[is.na(pdf_hp_raw_1$GarageFinish)] = "NOGTFI"
  pdf_hp_raw_1$BsmtFinType2[is.na(pdf_hp_raw_1$BsmtFinType2)] = "NOBFT"
  pdf_hp_raw_1$FireplaceQu[is.na(pdf_hp_raw_1$FireplaceQu)] = "NOFRQU"
  pdf_hp_raw_1$GarageQual[is.na(pdf_hp_raw_1$GarageQual)] = "NOGQUA"
  pdf_hp_raw_1$GarageCond[is.na(pdf_hp_raw_1$GarageCond)] = "NOGQUA"
  pdf_hp_raw_1$Electrical[is.na(pdf_hp_raw_1$Electrical)] = "NOELEC"
  pdf_hp_raw_1$Exterior1st[is.na(pdf_hp_raw_1$Exterior1st)] = "NOEXT"
  pdf_hp_raw_1$Exterior2nd[is.na(pdf_hp_raw_1$Exterior2nd)] = "NOEXT"
  pdf_hp_raw_1$Utilities[is.na(pdf_hp_raw_1$Utilities)] = "NOUTIL"
  pdf_hp_raw_1$BsmtFullBath[is.na(pdf_hp_raw_1$BsmtFullBath)] = 0
  pdf_hp_raw_1$BsmtHalfBath[is.na(pdf_hp_raw_1$BsmtHalfBath)] = 0
  pdf_hp_raw_1$KitchenQual[is.na(pdf_hp_raw_1$KitchenQual)] = "NOKIT"
  pdf_hp_raw_1$Functional[is.na(pdf_hp_raw_1$Functional)] = "NOFUNC"
  pdf_hp_raw_1$SaleType[is.na(pdf_hp_raw_1$SaleType)] = "NOSTY"


  pdf_hp_raw_1$LotFrontage[is.na(pdf_hp_raw_1$LotFrontage)] = mean(pdf_hp_raw_1$LotFrontage[!is.na(pdf_hp_raw_1$LotFrontage)])
  pdf_hp_raw_1$MasVnrArea[is.na(pdf_hp_raw_1$MasVnrArea)] = mean(pdf_hp_raw_1$MasVnrArea[!is.na(pdf_hp_raw_1$MasVnrArea)])
  pdf_hp_raw_1$GarageYrBlt[is.na(pdf_hp_raw_1$GarageYrBlt)] = mean(pdf_hp_raw_1$GarageYrBlt[!is.na(pdf_hp_raw_1$GarageYrBlt)])
  pdf_hp_raw_1$BsmtFinSF1[is.na(pdf_hp_raw_1$BsmtFinSF1)] = mean(pdf_hp_raw_1$BsmtFinSF1[!is.na(pdf_hp_raw_1$BsmtFinSF1)])
  pdf_hp_raw_1$BsmtFinSF2[is.na(pdf_hp_raw_1$BsmtFinSF2)] = mean(pdf_hp_raw_1$BsmtFinSF2[!is.na(pdf_hp_raw_1$BsmtFinSF2)])
  pdf_hp_raw_1$BsmtUnfSF[is.na(pdf_hp_raw_1$BsmtUnfSF)] = mean(pdf_hp_raw_1$BsmtUnfSF[!is.na(pdf_hp_raw_1$BsmtUnfSF)])
  pdf_hp_raw_1$TotalBsmtSF[is.na(pdf_hp_raw_1$TotalBsmtSF)] = mean(pdf_hp_raw_1$TotalBsmtSF[!is.na(pdf_hp_raw_1$TotalBsmtSF)])
  pdf_hp_raw_1$GarageCars[is.na(pdf_hp_raw_1$GarageCars)] = mean(pdf_hp_raw_1$GarageCars[!is.na(pdf_hp_raw_1$GarageCars)])
  pdf_hp_raw_1$GarageArea[is.na(pdf_hp_raw_1$GarageArea)] = mean(pdf_hp_raw_1$GarageArea[!is.na(pdf_hp_raw_1$GarageArea)])

 return(pdf_hp_raw_1)
}

```
~~~

Figure 14: Dealing with NAN Values



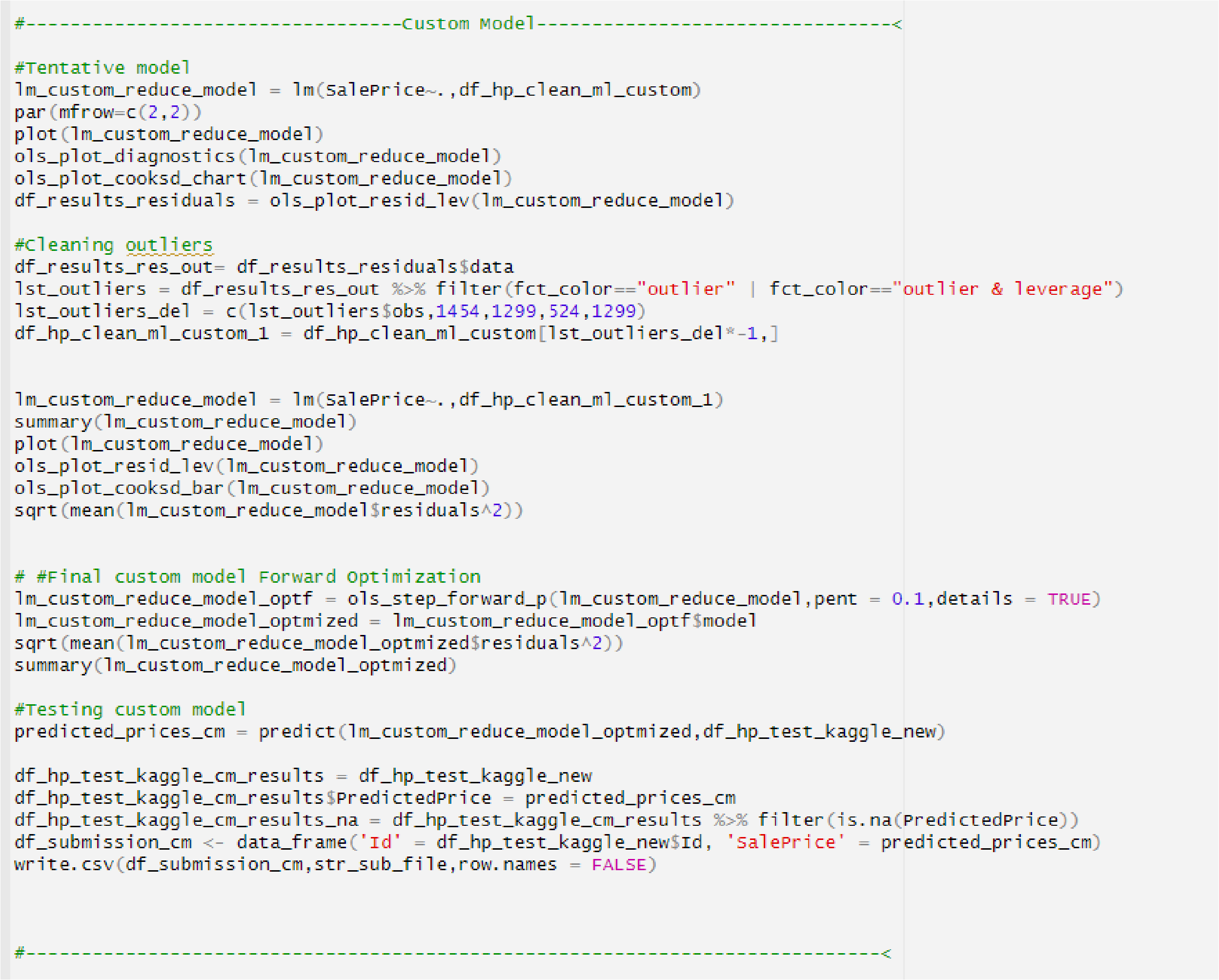
~~~
# Cleaning up data(Replacing NAN, Transformation and Factors)
```{r}

df_hp_raw_1 = df_hp_raw_0

#---------->Cleaning Missing Values<-----------------------<


df_hp_raw_1 = do_cleaning(df_hp_raw_1)
lst_nan_values = sapply(df_hp_raw_1, function(x) sum(is.na(x)))
lst_nan_values

df_hp_test_kaggle = do_cleaning(df_hp_test_kaggle)
lst_nan_values = sapply(df_hp_test_kaggle, function(x) sum(is.na(x)))
lst_nan_values
~~~

Figure 15: Dealing with NAN Values



~~~
#Analyzing categirical variables
```{r}

#LotFrontage,Log transformation X
df_hp_raw_1 %>% ggplot(aes(x=LotFrontage,y=SalePrice))+geom_point()
#LotArea,maybe Transformation x
df_hp_raw_1 %>% ggplot(aes(x=LotArea,y=SalePrice))+geom_point()
#YearRemodAdd, log transformation Y
df_hp_raw_1 %>% ggplot(aes(x=YearRemodAdd,y=SalePrice))+geom_point()
#MasVnrArea
df_hp_raw_1 %>% ggplot(aes(x=MasVnrArea,y=SalePrice))+geom_point()
#BsmtFinSF1
df_hp_raw_1 %>% ggplot(aes(x=BsmtFinSF1,y=SalePrice))+geom_point()
#BsmtFinSF2
df_hp_raw_1 %>% ggplot(aes(x=BsmtFinSF2,y=SalePrice))+geom_point()
#BsmtUnfSF
df_hp_raw_1 %>% ggplot(aes(x=BsmtUnfSF,y=SalePrice))+geom_point()
#TotalBsmtSF, maybe Log transformation Y
df_hp_raw_1 %>% ggplot(aes(x=TotalBsmtSF,y=SalePrice))+geom_point()
#1stFlrSF
df_hp_raw_1 %>% ggplot(aes(x=X1stFlrSF,y=SalePrice))+geom_point()
#LowQualFinSF, Remove it
df_hp_raw_1 %>% ggplot(aes(x=LowQualFinSF,y=SalePrice))+geom_point()
#GrLivArea
df_hp_raw_1 %>% ggplot(aes(x=GrLivArea,y=SalePrice))+geom_point()
#Bedroom, Maybe turn it into categorical
df_hp_raw_1 %>% ggplot(aes(x=BedroomAbvGr,y=SalePrice))+geom_point()
#KitchenAbvGr, Maybe turn it into categorical
df_hp_raw_1 %>% ggplot(aes(x=KitchenAbvGr,y=SalePrice))+geom_point()
#TotRmsAbvGrd
df_hp_raw_1 %>% ggplot(aes(x=TotRmsAbvGrd,y=SalePrice))+geom_point()
#Fireplaces, Maybe turn it into categorical
df_hp_raw_1 %>% ggplot(aes(x=Fireplaces,y=SalePrice))+geom_point()
#GarageYrBlt
df_hp_raw_1 %>% ggplot(aes(x=GarageYrBlt,y=SalePrice))+geom_point()
#GarageCars
df_hp_raw_1 %>% ggplot(aes(x=GarageCars,y=SalePrice))+geom_point()
#GarageArea
df_hp_raw_1 %>% ggplot(aes(x=GarageArea,y=logSalePrice))+geom_point()
#WoodDeckSF
df_hp_raw_1 %>% ggplot(aes(x=WoodDeckSF,y=SalePrice))+geom_point()
#OpenPorchSF, log transformation x
df_hp_raw_1 %>% ggplot(aes(x=OpenPorchSF,y=SalePrice))+geom_point()
#EnclosedPorch
df_hp_raw_1 %>% ggplot(aes(x=EnclosedPorch,y=SalePrice))+geom_point()
~~~

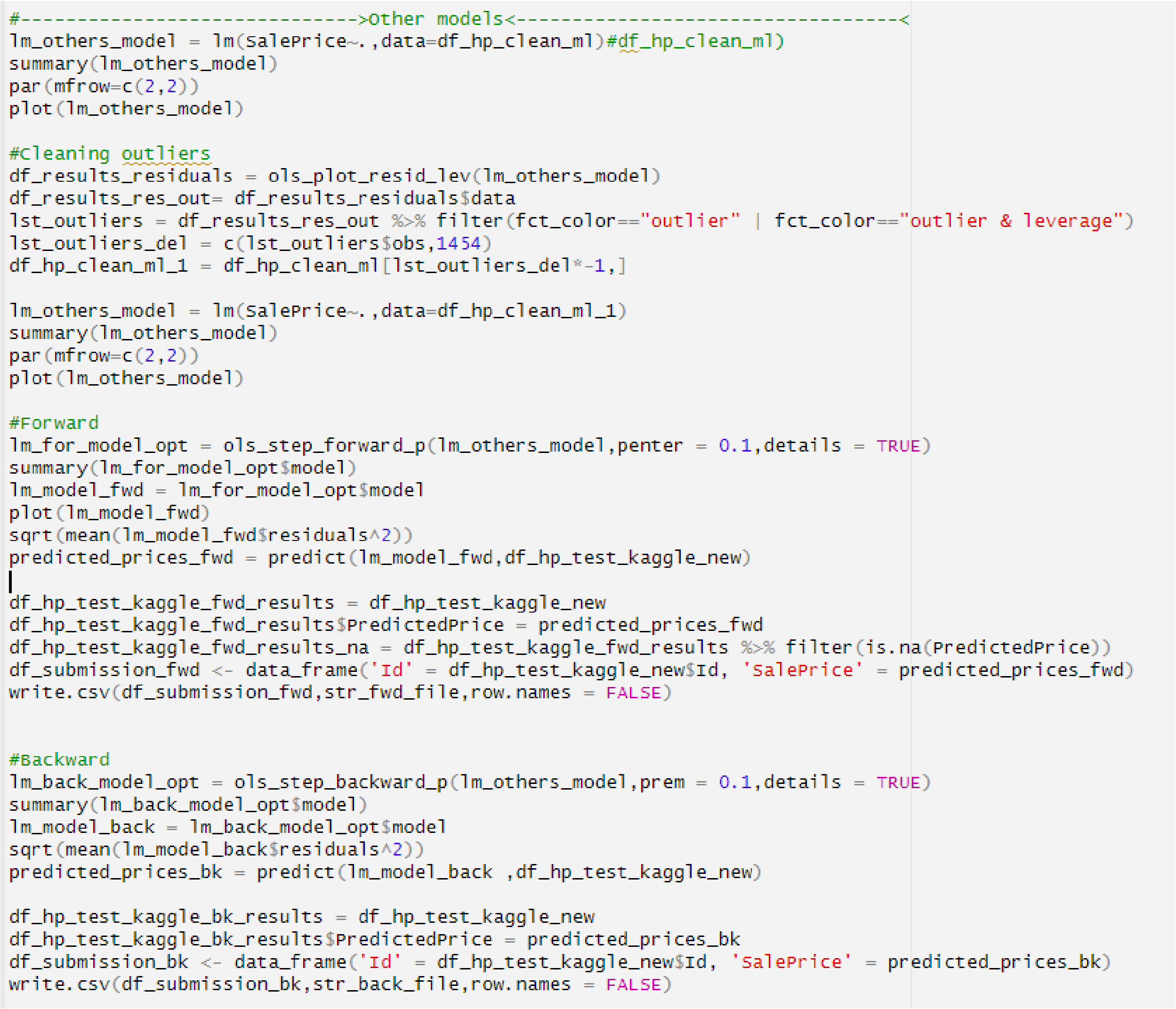
Figure 16: Analysis of categorical variables



```
#----------------------------------Custom Model---------------------------------<

#Tentative model
lm_custom_reduce_model = lm(SalePrice~.,df_hp_clean_ml_custom)
par(mfrow=c(2,2))
plot(lm_custom_reduce_model)
ols_plot_diagnostics(lm_custom_reduce_model)
ols_plot_cooksd_chart(lm_custom_reduce_model)
df_results_residuals = ols_plot_resid_lev(lm_custom_reduce_model)

#Cleaning outliers
df_results_res_out= df_results_residuals$data
lst_outliers = df_results_res_out %>% filter(fct_color=="outlier" | fct_color=="outlier & leverage")
lst_outliers_del = c(lst_outliers$obs,1454,1299,524,1299)
df_hp_clean_ml_custom_1 = df_hp_clean_ml_custom[lst_outliers_del*-1,]


lm_custom_reduce_model = lm(SalePrice~.,df_hp_clean_ml_custom_1)
summary(lm_custom_reduce_model)
plot(lm_custom_reduce_model)
ols_plot_resid_lev(lm_custom_reduce_model)
ols_plot_cooksd_bar(lm_custom_reduce_model)
sqrt(mean(lm_custom_reduce_model$residuals^2))


# #Final custom model Forward Optimization
lm_custom_reduce_model_optf = ols_step_forward_p(lm_custom_reduce_model,pent = 0.1,details = TRUE)
lm_custom_reduce_model_optmized = lm_custom_reduce_model_optf$model
sqrt(mean(lm_custom_reduce_model_optmized$residuals^2))
summary(lm_custom_reduce_model_optmized)

#Testing custom model
predicted_prices_cm = predict(lm_custom_reduce_model_optmized,df_hp_test_kaggle_new)

df_hp_test_kaggle_cm_results = df_hp_test_kaggle_new
df_hp_test_kaggle_cm_results$PredictedPrice = predicted_prices_cm
df_hp_test_kaggle_cm_results_na = df_hp_test_kaggle_cm_results %>% filter(is.na(PredictedPrice))
df_submission_cm <- data_frame('Id' = df_hp_test_kaggle_new$Id, 'SalePrice' = predicted_prices_cm)
write.csv(df_submission_cm,str_sub_file,row.names = FALSE)



#------------------------------------------------------------------------------<
```

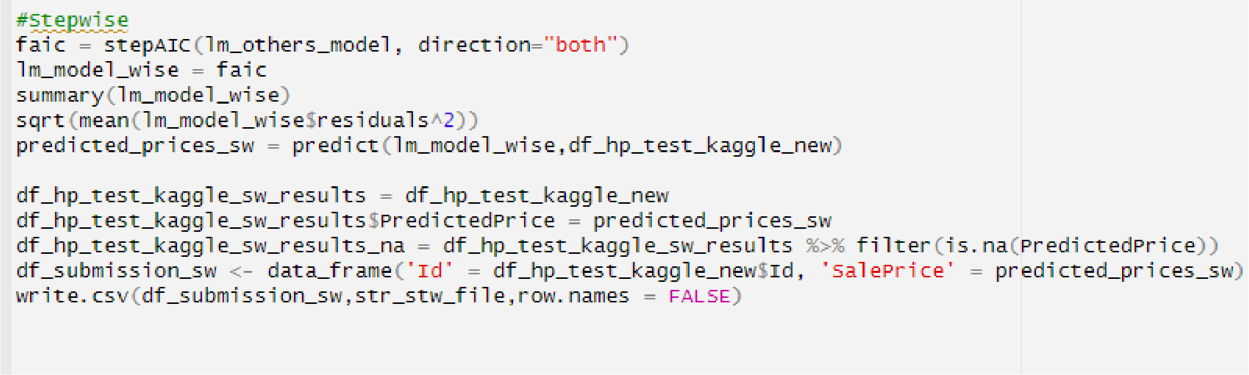
Figure 17: Custom model

*In this part we assemble the custom model, we also removed the influential cases to fulfill the assumptions of Linear Regression



```
#------------------------------->Other models<-----------------------------------<
lm_others_model = lm(SalePrice~.,data=df_hp_clean_ml)#df_hp_clean_ml)
summary(lm_others_model)
par(mfrow=c(2,2))
plot(lm_others_model)

#Cleaning outliers
df_results_residuals = ols_plot_resid_lev(lm_others_model)
df_results_res_out= df_results_residuals$data
lst_outliers = df_results_res_out %>% filter(fct_color=="outlier" | fct_color=="outlier & leverage")
lst_outliers_del = c(lst_outliers$obs,1454)
df_hp_clean_ml_1 = df_hp_clean_ml[lst_outliers_del*-1,]

lm_others_model = lm(SalePrice~.,data=df_hp_clean_ml_1)
summary(lm_others_model)
par(mfrow=c(2,2))
plot(lm_others_model)

#Forward
lm_for_model_opt = ols_step_forward_p(lm_others_model,penter = 0.1,details = TRUE)
summary(lm_for_model_opt$model)
lm_model_fwd = lm_for_model_opt$model
plot(lm_model_fwd)
sqrt(mean(lm_model_fwd$residuals^2))
predicted_prices_fwd = predict(lm_model_fwd,df_hp_test_kaggle_new)

df_hp_test_kaggle_fwd_results = df_hp_test_kaggle_new
df_hp_test_kaggle_fwd_results$PredictedPrice = predicted_prices_fwd
df_hp_test_kaggle_fwd_results_na = df_hp_test_kaggle_fwd_results %>% filter(is.na(PredictedPrice))
df_submission_fwd <- data_frame('Id' = df_hp_test_kaggle_new$Id, 'SalePrice' = predicted_prices_fwd)
write.csv(df_submission_fwd,str_fwd_file,row.names = FALSE)


#Backward
lm_back_model_opt = ols_step_backward_p(lm_others_model,prem = 0.1,details = TRUE)
summary(lm_back_model_opt$model)
lm_model_back = lm_back_model_opt$model
sqrt(mean(lm_model_back$residuals^2))
predicted_prices_bk = predict(lm_model_back ,df_hp_test_kaggle_new)

df_hp_test_kaggle_bk_results = df_hp_test_kaggle_new
df_hp_test_kaggle_bk_results$PredictedPrice = predicted_prices_bk
df_submission_bk <- data_frame('Id' = df_hp_test_kaggle_new$Id, 'SalePrice' = predicted_prices_bk)
write.csv(df_submission_bk,str_back_file,row.names = FALSE)
```

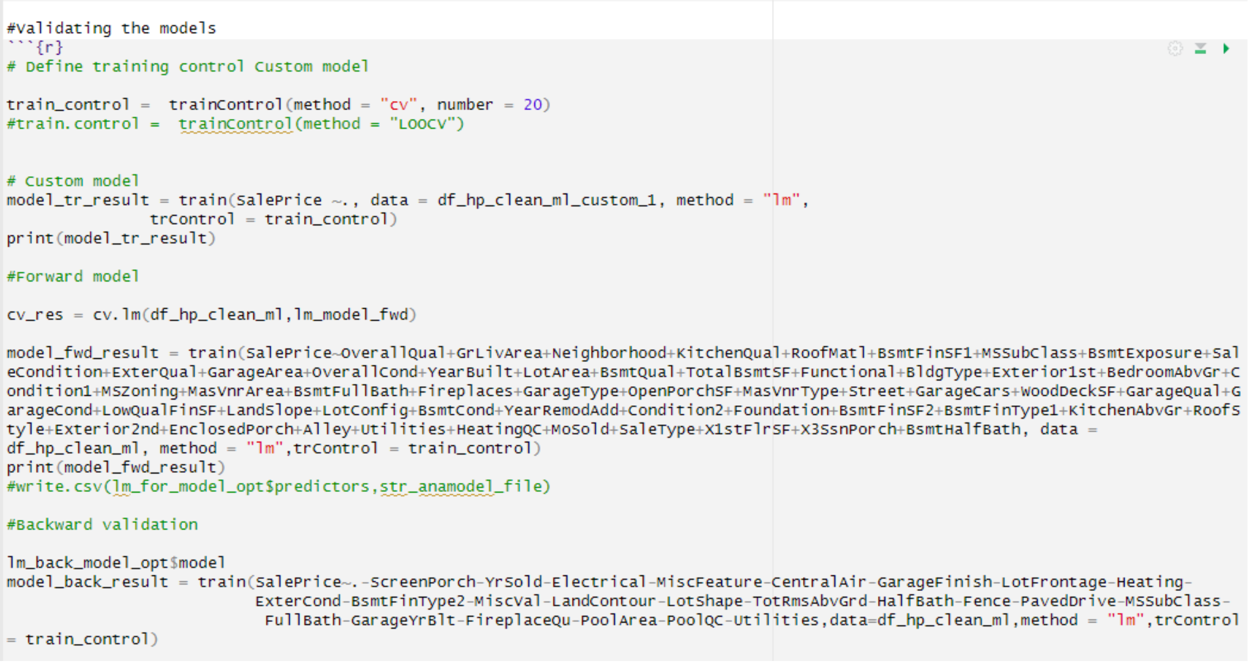
Figure 18: Other models



```
#Stepwise
faic = stepAIC(lm_others_model, direction="both")
lm_model_wise = faic
summary(lm_model_wise)
sqrt(mean(lm_model_wise$residuals^2))
predicted_prices_sw = predict(lm_model_wise,df_hp_test_kaggle_new)

df_hp_test_kaggle_sw_results = df_hp_test_kaggle_new
df_hp_test_kaggle_sw_results$PredictedPrice = predicted_prices_sw
df_hp_test_kaggle_sw_results_na = df_hp_test_kaggle_sw_results %>% filter(is.na(PredictedPrice))
df_submission_sw <- data_frame('Id' = df_hp_test_kaggle_new$Id, 'SalePrice' = predicted_prices_sw)
write.csv(df_submission_sw,str_stw_file,row.names = FALSE)
```

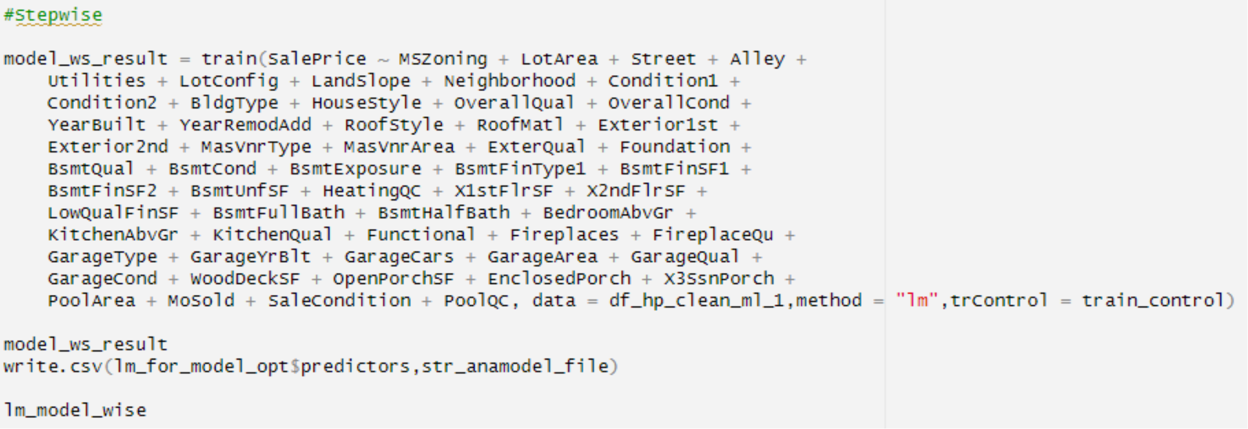
Figure 19: Other models

*In this part we assemble the other models: Forward, Backward, and Stepwise.



~~~
#Validating the models
```{r}
# Define training control Custom model

train_control =  trainControl(method = "cv", number = 20)
#train.control =  trainControl(method = "LOOCV")


# Custom model
model_tr_result = train(SalePrice ~., data = df_hp_clean_ml_custom_1, method = "lm",
               trControl = train_control)
print(model_tr_result)

#Forward model

cv_res = cv.lm(df_hp_clean_ml,lm_model_fwd)

model_fwd_result = train(SalePrice~OverallQual+GrLivArea+Neighborhood+KitchenQual+RoofMatl+BsmtFinSF1+MSSubClass+BsmtExposure+Sal
eCondition+ExterQual+GarageArea+OverallCond+YearBuilt+LotArea+BsmtQual+TotalBsmtSF+Functional+BldgType+Exterior1st+BedroomAbvGr+C
ondition1+MSZoning+MasVnrArea+BsmtFullBath+Fireplaces+GarageType+OpenPorchSF+MasVnrType+Street+GarageCars+WoodDeckSF+GarageQual+G
arageCond+LowQualFinSF+LandSlope+LotConfig+BsmtCond+YearRemodAdd+Condition2+Foundation+BsmtFinSF2+BsmtFinType1+KitchenAbvGr+RoofS
tyle+Exterior2nd+EnclosedPorch+Alley+Utilities+HeatingQC+MoSold+SaleType+X1stFlrSF+X3SsnPorch+BsmtHalfBath, data =
df_hp_clean_ml, method = "lm",trControl = train_control)
print(model_fwd_result)
#write.csv(lm_for_model_opt$predictors,str_anamodel_file)

#Backward validation

lm_back_model_opt$model
model_back_result = train(SalePrice~.-ScreenPorch-YrSold-Electrical-MiscFeature-CentralAir-GarageFinish-LotFrontage-Heating-
                          ExterCond-BsmtFinType2-MiscVal-LandContour-LotShape-TotRmsAbvGrd-HalfBath-Fence-PavedDrive-MSSubClass-
                           FullBath-GarageYrBlt-FireplaceQu-PoolArea-PoolQC-Utilities,data=df_hp_clean_ml,method = "lm",trControl
= train_control)
~~~

Figure 20: Model cross-validation

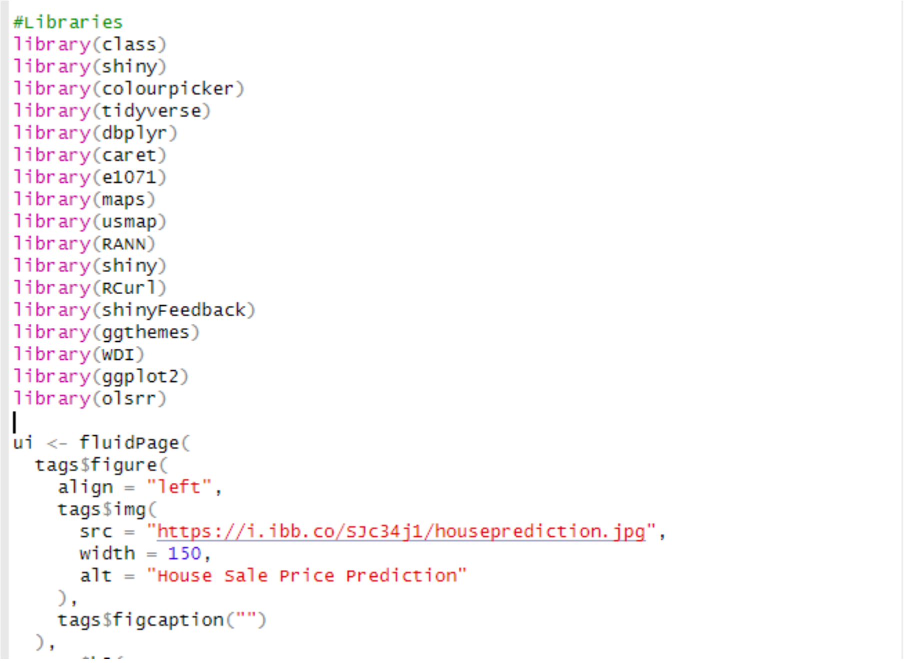


```
#Stepwise

model_ws_result = train(SalePrice ~ MSZoning + LotArea + Street + Alley +
    Utilities + LotConfig + LandSlope + Neighborhood + Condition1 +
    Condition2 + BldgType + HouseStyle + OverallQual + OverallCond +
    YearBuilt + YearRemodAdd + RoofStyle + RoofMatl + Exterior1st +
    Exterior2nd + MasVnrType + MasVnrArea + ExterQual + Foundation +
    BsmtQual + BsmtCond + BsmtExposure + BsmtFinType1 + BsmtFinSF1 +
    BsmtFinSF2 + BsmtUnfSF + HeatingQC + X1stFlrSF + X2ndFlrSF +
    LowQualFinSF + BsmtFullBath + BsmtHalfBath + BedroomAbvGr +
    KitchenAbvGr + KitchenQual + Functional + Fireplaces + FireplaceQu +
    GarageType + GarageYrBlt + GarageCars + GarageArea + GarageQual +
    GarageCond + WoodDeckSF + OpenPorchSF + EnclosedPorch + X3SsnPorch +
    PoolArea + MoSold + SaleCondition + PoolQC, data = df_hp_clean_ml_1,method = "lm",trControl = train_control)

model_ws_result
write.csv(lm_for_model_opt$predictors,str_anamodel_file)

lm_model_wise
```

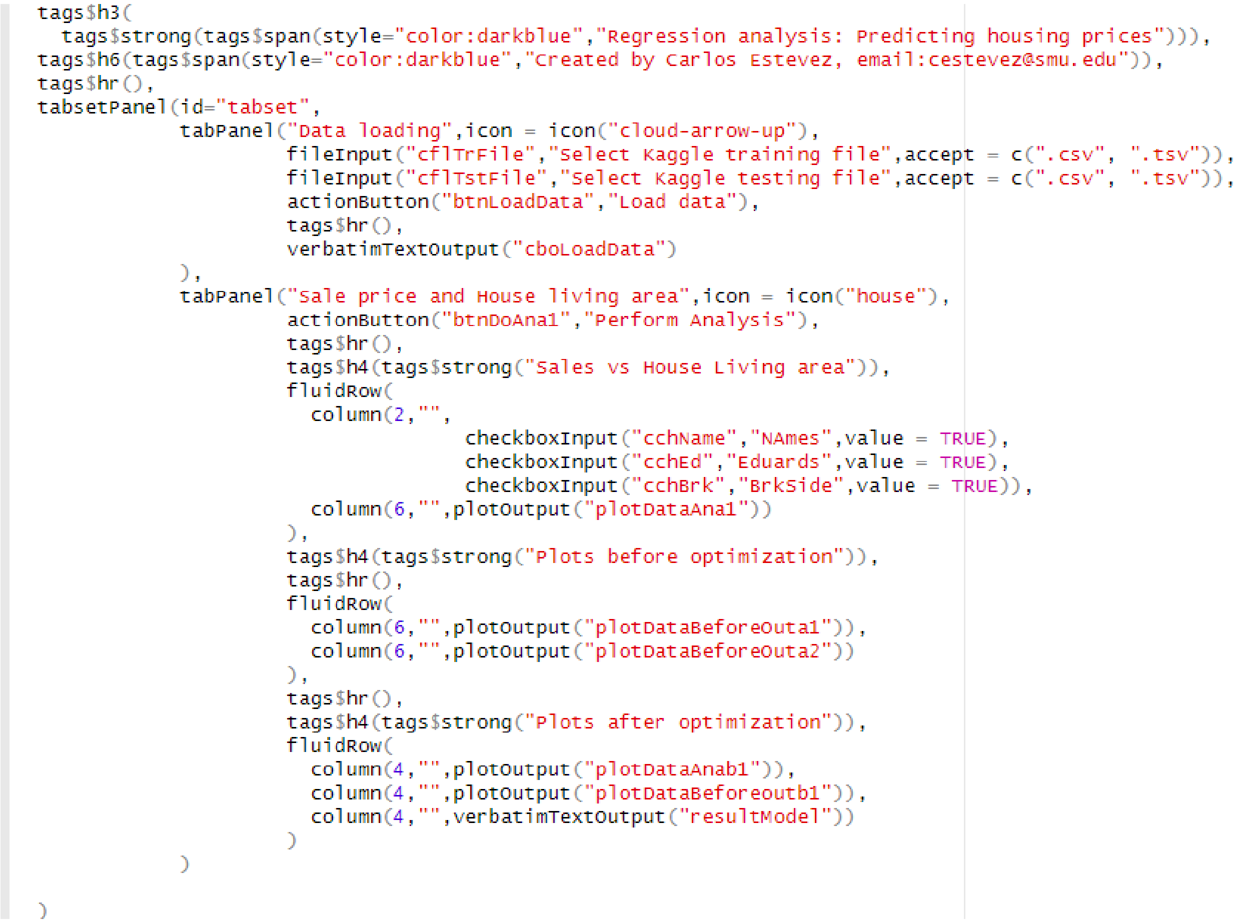
Figure 21: Model cross-validation

SHINY APP

UI

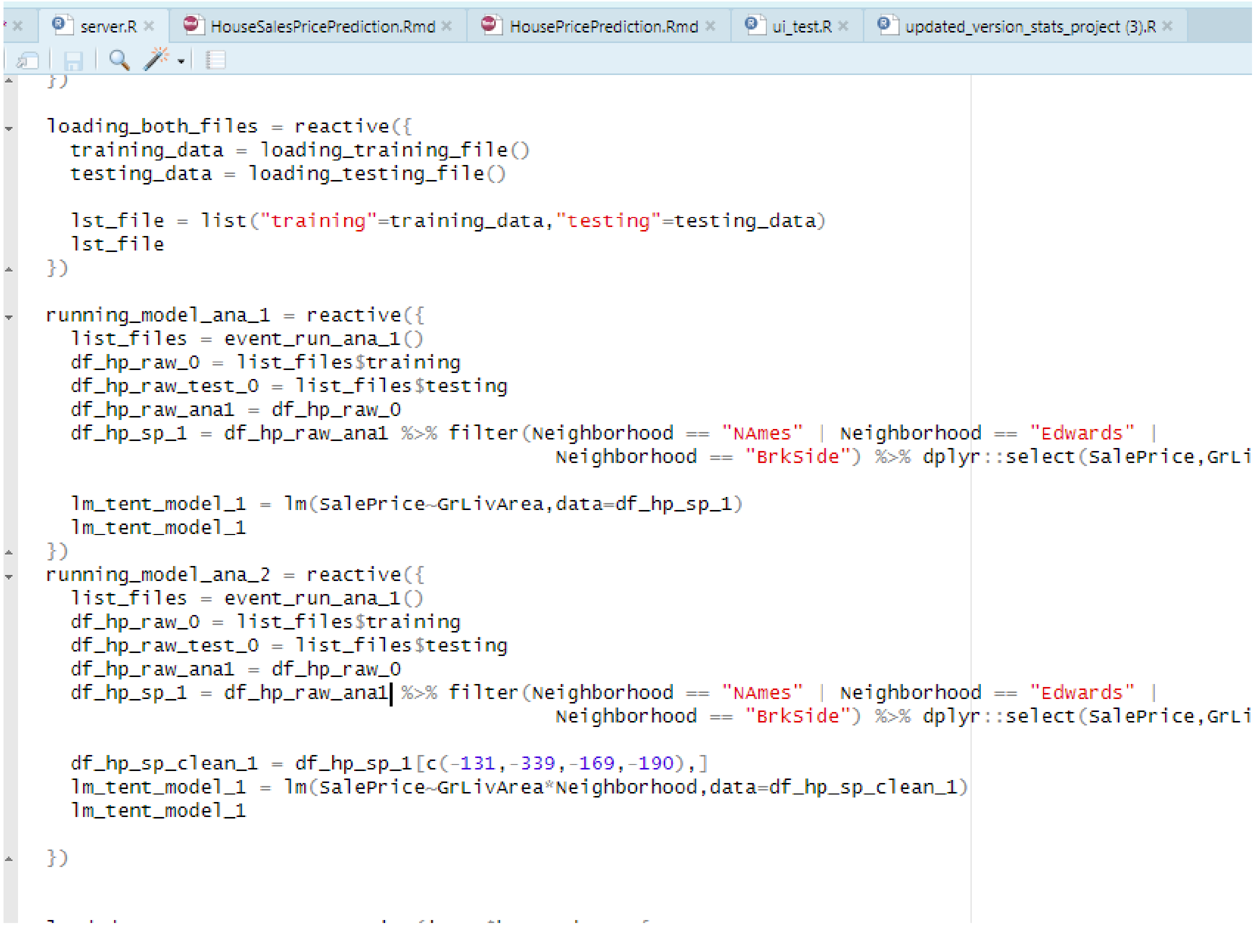


```r
#Libraries
library(class)
library(shiny)
library(colourpicker)
library(tidyverse)
library(dbplyr)
library(caret)
library(e1071)
library(maps)
library(usmap)
library(RANN)
library(shiny)
library(RCurl)
library(shinyFeedback)
library(ggthemes)
library(WDI)
library(ggplot2)
library(olsrr)

ui <- fluidPage(
  tags$figure(
    align = "left",
    tags$img(
      src = "https://i.ibb.co/SJc34j1/houseprediction.jpg",
      width = 150,
      alt = "House Sale Price Prediction"
    ),
    tags$figcaption("")
  ),
```



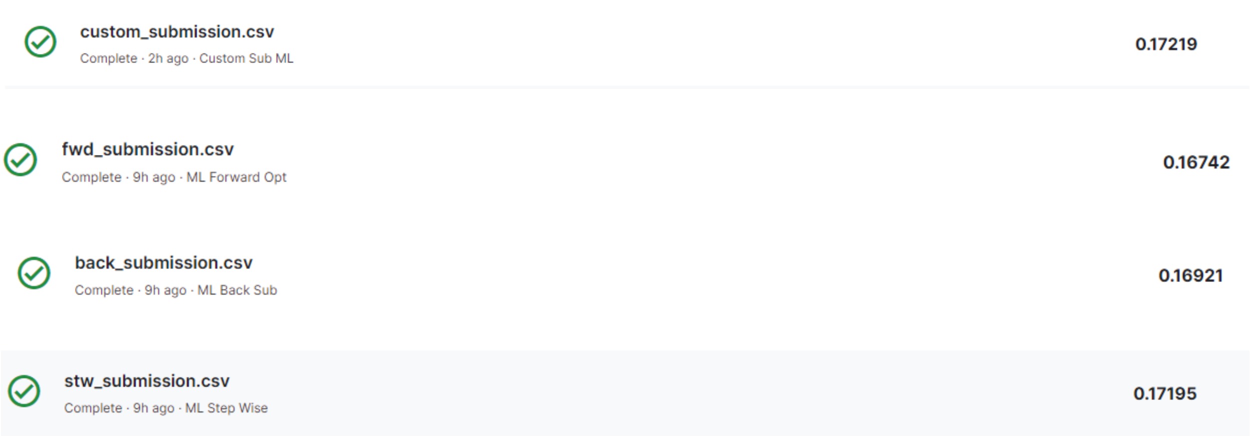
```r
  tags$h3(
    tags$strong(tags$span(style="color:darkblue","Regression analysis: Predicting housing prices"))),
  tags$h6(tags$span(style="color:darkblue","Created by Carlos Estevez, email:cestevez@smu.edu")),
  tags$hr(),
  tabsetPanel(id="tabset",
              tabPanel("Data loading",icon = icon("cloud-arrow-up"),
                       fileInput("cflTrFile","Select Kaggle training file",accept = c(".csv", ".tsv")),
                       fileInput("cflTstFile","Select Kaggle testing file",accept = c(".csv", ".tsv")),
                       actionButton("btnLoadData","Load data"),
                       tags$hr(),
                       verbatimTextOutput("cboLoadData")
              ),
              tabPanel("Sale price and House living area",icon = icon("house"),
                       actionButton("btnDoAna1","Perform Analysis"),
                       tags$hr(),
                       tags$h4(tags$strong("Sales vs House Living area")),
                       fluidRow(
                         column(2,"",
                                      checkboxInput("cchName","NAmes",value = TRUE),
                                      checkboxInput("cchEd","Eduards",value = TRUE),
                                      checkboxInput("cchBrk","BrkSide",value = TRUE)),
                         column(6,"",plotOutput("plotDataAna1"))
                       ),
                       tags$h4(tags$strong("Plots before optimization")),
                       tags$hr(),
                       fluidRow(
                         column(6,"",plotOutput("plotDataBeforeOuta1")),
                         column(6,"",plotOutput("plotDataBeforeOuta2"))
                       ),
                       tags$hr(),
                       tags$h4(tags$strong("Plots after optimization")),
                       fluidRow(
                         column(4,"",plotOutput("plotDataAnab1")),
                         column(4,"",plotOutput("plotDataBeforeoutb1")),
                         column(4,"",verbatimTextOutput("resultModel"))
                       )
              )

  )
```

SERVER


server.R
HouseSalesPricePrediction.Rmd
HousePricePrediction.Rmd
ui_test.R
updated_version_stats_project (3).R

```
})

loading_both_files = reactive({
  training_data = loading_training_file()
  testing_data = loading_testing_file()

  lst_file = list("training"=training_data,"testing"=testing_data)
  lst_file
})

running_model_ana_1 = reactive({
  list_files = event_run_ana_1()
  df_hp_raw_0 = list_files$training
  df_hp_raw_test_0 = list_files$testing
  df_hp_raw_ana1 = df_hp_raw_0
  df_hp_sp_1 = df_hp_raw_ana1 %>% filter(Neighborhood == "NAmes" | Neighborhood == "Edwards" |
                                           Neighborhood == "BrkSide") %>% dplyr::select(SalePrice,GrLi

  lm_tent_model_1 = lm(SalePrice~GrLivArea,data=df_hp_sp_1)
  lm_tent_model_1
})
running_model_ana_2 = reactive({
  list_files = event_run_ana_1()
  df_hp_raw_0 = list_files$training
  df_hp_raw_test_0 = list_files$testing
  df_hp_raw_ana1 = df_hp_raw_0
  df_hp_sp_1 = df_hp_raw_ana1 %>% filter(Neighborhood == "NAmes" | Neighborhood == "Edwards" |
                                           Neighborhood == "BrkSide") %>% dplyr::select(SalePrice,GrLi

  df_hp_sp_clean_1 = df_hp_sp_1[c(-131,-339,-169,-190),]
  lm_tent_model_1 = lm(SalePrice~GrLivArea*Neighborhood,data=df_hp_sp_clean_1)
  lm_tent_model_1

})
```

KAGGLE SCORE



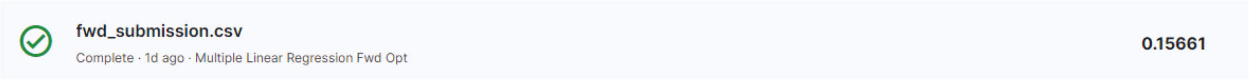
custom_submission.csv
Complete · 2h ago · Custom Sub ML
0.17219

fwd_submission.csv
Complete · 9h ago · ML Forward Opt
0.16742

back_submission.csv
Complete · 9h ago · ML Back Sub
0.16921

stw_submission.csv
Complete · 9h ago · ML Step Wise
0.17195

Additional model:



fwd_submission.csv

Complete · 1d ago · Multiple Linear Regression Fwd Opt

0.15661