Final Project Writeup

Ivan Chavez Github site - iachavez97.github.io;
Oluwadamilola Owolabi Github Repository -
https://github.com/DamilolaOwolabi/DS-6371-Project

Introduction

The goal of this project is to predict the sale price of homes in Ames, Iowa using existing data. We strive to utilize various statistical and data analytical techniques in order to derive the best predictive model for the sale price.

Data Description

We received this data from a study of residential homes in Ames, Iowa. The data set contains 383 rows of values with 79 different explanatory variable columns. You can find out more regarding the dataset and how it was pulled in the kaggle for house prices - advanced regression techniques. For respect to the analysis of question 1 the three main variables that I assessed were the Neighborhood, sales price, and GrLivArea variables. The variables used in the second analysis are SalePrice, GrLivArea, and the OverallQual.

Analysis Question 1:

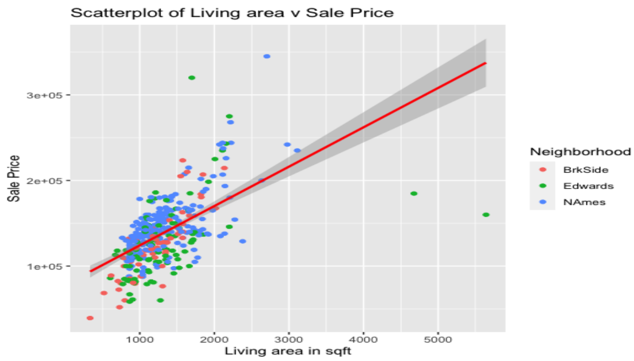
Restatement of Problem

For our first analysis our objective was to get an estimate of sale prices of houses are related to sq footage of living area in houses for three neighborhoods which are NAmes, Edwards, and BrkSide. Additionally, we were requested to provide the estimates if they differ based on the neighborhood and provide confidence intervals for each of the different neighborhoods. We will also be providing evidence that our model fits the assumptions and how we observed outliers/influential observations.

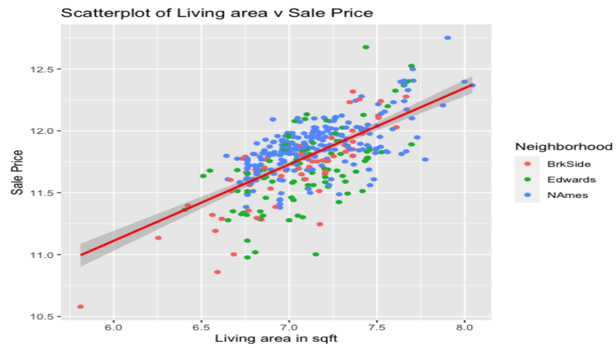
Build and Fit the Model

In building my model for the analysis I used a multiple linear regression model with Sale price being the response variable and GrLivArea and neighborhood being the explanatory variables. Below we see a scatterplot of living area for the 3 neighborhood vs the sale price of each home. I created this scatterplot to get an idea for the distribution of the points and see if there are any outliers, and we can see that there are two outliers from the same neighborhood Edwards. After analysis and looking at residual plots we decided to remove these two outlier points that are in the edwards neighborhood because they appear to be incorrectly assigned and the fact that they are in the same neighborhood helps us come to that conclusion as well.

Scatterplot before transformations



Scatterplot post log transformations and removing of outliers

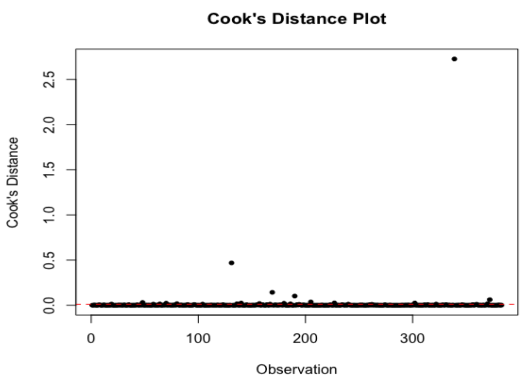


Checking Assumptions

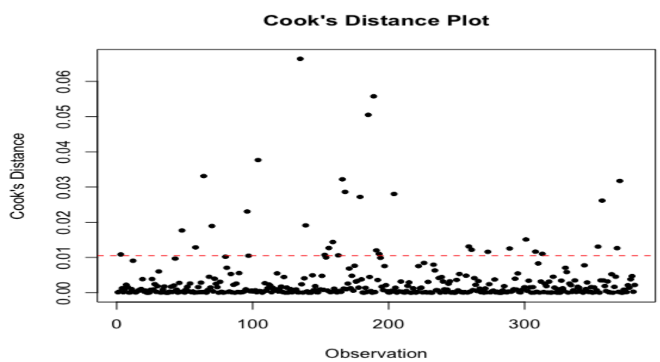
Residual Plots - The residual plots will be in the index, through the cooks d plots we were able to identify the two values that were outliers and identified them as data point 131 and 399. We removed these and they showed a much better fit for the model and a more normalized distribution of our residual plots.

Influential point analysis (Cook's D and Leverage)

Original cooks d plot below



New cook's d plot and we can see that cooks d levels have gone down significantly.



Make sure to address each assumption.

We can see that after our removing of outliers and transformations that the cooks d as well as our residuals look much better compared to the original model. So we are now able to move forward with this model and show how sale price is related to GrLivArea compared to each neighborhood.

Comparing Competing Models

Adj R^2 - the adjusted R^2 I got for my final model is .5002
Internal CV Press - the internal CV press number I got was 1437833896.

Parameters

Estimates

Here we see the summary of our model.

```
Call:
lm(formula = log(SalePrice) ~ log(GrLivArea) + Neighborhood,
    data = dfTest)

Residuals:
     Min       1Q   Median       3Q      Max
-0.73253 -0.10572  0.02277  0.12232  0.77125

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           7.48892    0.23954  31.263  < 2e-16 ***
log(GrLivArea)        0.59565    0.03386  17.594  < 2e-16 ***
NeighborhoodEdwards  -0.01405    0.03211  -0.438    0.662
NeighborhoodNAmes     0.12881    0.02867   4.492 9.39e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1934 on 377 degrees of freedom
Multiple R-squared:  0.5041,	Adjusted R-squared:  0.5002
F-statistic: 127.8 on 3 and 377 DF,  p-value: < 2.2e-16
```

Interpretation

Above we can see the table for our multiple linear regression model. Since we ran a log-log transformation we can interpret how a percent change in the GrLivArea variable will affect our final sale price. For NeighborhoodEdwards a 1% in GrLivArea the sale price will increase by approximately .011%, for NeighborhoodNAmes a 1% increase in the GrLivArea will increase the final sale price by approximately .13%, and for NeighborhoodBrkSide a 1% increase in the GrLivArea will increase the sale price by .60%.

Confidence Intervals

The confidence interval for NeighborhoodEdwards is (-.077, .049)
The confidence interval for NeighborhoodNAmes is (.072, .19)
The confidence interval for NeighborhoodBrkSide is (.53, .66)

Conclusion

We can see from our original graphs that the residuals and the plot had outliers so we assessed them and were able to get more normally distributed plots after assessing the outliers and performing our transformations. We can see from our graphs as well that there appears to be a positive relationship between the GrLivArea and the sales price of homes and we were able to quantify the percent increase of the homes sales price based on the GrLivArea. We can see as well that the NAmes neighborhood has the highest value homes with BrkSide being in second and Edwards being the least valuable of the three neighborhoods.

R Shiny: Price v. Living Area Chart link - https://iachavez97.shinyapps.io/RealEstateApp/

In our Rshiny app we are showing a scatterplot of the relation between sales price of homes and the living area. Additionally, it is able to be separated by each of the 3 different neighborhoods NAmes, BrkSide, and Edwards

Analysis Question 2

Restatement of Problem

This analysis hopes to build the best predictive model needed to predict future sale prices of homes in Ames, Iowa. We plan on doing that by looking at different types of regression models (Simple Linear Regression, Multiple Linear Regression, and Custom Multiple Regression), choosing the best model for each regression type by analyzing the various model selections and making a final decision based on the adjusted r-squared, CVpress and Kaggle score.

Model Selection

1. **Simple Linear Regression**

Stepwise

Root MSE	52232
Dependent Mean	178922
R-Square	0.5356
Adj R-Sq	0.5352
AIC	26481
AICC	26481
SBC	25324
ASE (Train)	2723500456
ASE (Test)	3657480659
CV PRESS	3.193688E12

Forward

Root MSE	52860
Dependent Mean	180708
R-Square	0.5416
Adj R-Sq	0.5412
AIC	26304
AICC	26304
SBC	25156
ASE (Train)	2789371837
ASE (Test)	3342402452
CV PRESS	3.255044E12

Backward

Root MSE	53869
Dependent Mean	179803
R-Square	0.5181
Adj R-Sq	0.5176
AIC	27214
AICC	27214
SBC	26028
ASE (Train)	2897039939
ASE (Test)	2952862743
CV PRESS	3.549373E12

Decision: Given that the adjusted R-squared for the forward model selection is higher (0.5412). We will go ahead with that instead.

2. **Multiple Linear Regression**

Stepwise

Root MSE	52570
Dependent Mean	180887
R-Square	0.5729
Adj R-Sq	0.5721
AIC	26201
AICC	26201
SBC	25062
ASE (Train)	2756372103
ASE (Test)	3025721897
CV PRESS	3.243666E12

Forward

Root MSE	53842
Dependent Mean	181178
R-Square	0.5385
Adj R-Sq	0.5378
AIC	26553
AICC	26553
SBC	25401
ASE (Train)	2891539026
ASE (Test)	2502360247
CV PRESS	3.41045E12

Backward

Root MSE	53300
Dependent Mean	180639
R-Square	0.5481
Adj R-Sq	0.5473
AIC	27166
AICC	27166
SBC	25987
ASE (Train)	2833696574
ASE (Test)	2714525649
CV PRESS	3.395127E12

Decision: Given that the adjusted R-squared for the stepwise model selection is higher (0.5712). We will go ahead with that instead.

3. **Custom Multiple Linear Regression (SalePrice ~ GrLivArea + OverallQual)**

Stepwise

Root MSE	41434
Dependent Mean	182703
R-Square	0.7432
Adj R-Sq	0.7427
AIC	26254
AICC	26254
SBC	25088
ASE (Train)	1712369832
ASE (Test)	1302181175
CV PRESS	2.036843E12

Forward

Root MSE	40608
Dependent Mean	180459
R-Square	0.7451
Adj R-Sq	0.7447
AIC	26073
AICC	26073
SBC	24913
ASE (Train)	1644778275
ASE (Test)	1578909763
CV PRESS	1.946607E12

Backward

Root MSE	40909
Dependent Mean	181664
R-Square	0.7484
Adj R-Sq	0.7480
AIC	25957
AICC	25957
SBC	24803
ASE (Train)	1669277804
ASE (Test)	1495410943
CV PRESS	1.967472E12

Decision: Given that the adjusted R-squared for the backward model selection is higher (0.7480). We will go ahead with that instead.

Checking Assumptions

(Note: Due to page limitations, the plots for the assumptions are located in the index page)

1. Simple Linear Regression

Residual Plots

Judging from the scatterplot of residuals, there is no evidence against the normality of the sales price conditional on the General living area. There is also no evidence against the linear trend between the sales price versus the General Living Area because the data points converge around the line. We were able to remove the 2 extreme points that were affecting the model using the cook's D plot. There are 2 more outliers in the residual scatterplot towards the upper right, they were left behind because they might be influential to the model and they are closer to the cluster than the previous 2 datapoints.

Cooks D-Plot

Initial: The data points 1299 and 524 are 2 high values in the model that might affect the regression fit, which should be removed.

Final: The cook's D plot hows the influence of each individual point on the fitted regression line. The previous 2 points with extremely high values have been identified and removed. The two highest values of the line (691, and 1182) have been identified on the residual plot and verified to not affect the model's p-value.

Leverage Plot

Initial: The leverage shows that there 2 really high influential points in the data (524, and 1299) that are affecting the plot to veer towards Rstudent > 0 instead of being spread apart.

Final: After the 2 influential plots were removed, the plots show to be more spread apart, hence providing a better fit.

2. **Multiple Linear Regression**

Residual Plots

Judging from the scatterplot of residuals, there is no evidence against the normality of the sales price conditional on the General living area and the number of full baths in the home. There is also no evidence against the linear trend between the sales price versus the General Living Area the number of full baths in the home because the data points converge around the line. We were able to remove the 2 extreme points that were affecting the model using the cook's D plot.

Cooks D-Plot

Initial: The data points 1299 and 524 are 2 high values in the model that might affect the regression fit, which should be removed.

Final: The cook's D plot hows the influence of each individual point on the fitted regression line. The previous 2 points with extremely high values have been identified and removed. The two highest values of the line (691, and 1182) have been identified on the residual plot and verified to not affect the model's p-value.

Leverage Plot

Initial: The leverage shows that there 2 really high influentual points in the data (524, and 1299) that are affecting the plot to veer towards Rstudent > 0 instead of being spread apart.

Final: After the 2 influential plots were removed, the plots show to be more spread apart, hence providing a better fit.

3. **Multiple Linear Regression**

Residual Plots

Judging from the scatterplot of residuals, there is no evidence against the normality of the sales price conditional on the General living area and the overall quality of the homes. There is also no evidence against the linear trend between the sales price versus the General Living Area and the overall quality of the homes, because the data points converge around the line.We were able to remove the 2 extreme points that were affecting the model using the cook's D plot.

Cooks D-Plot

Initial: The data points 1299 and 524 are 2 high values in the model that might affect the regression fit, which should be removed.

Final: The cook's D plot hows the influence of each individual point on the fitted regression line. The previous 2 points with extremely high values have been identified and removed. The two highest values of the line (691, and 1182) have been identified on the residual plot and verified to not affect the model's p-value.

Leverage Plot

Initial: The leverage shows that there 2 really high influentual points in the data (524, and 1299) that are affecting the plot to veer towards Rstudent > 0 instead of being spread apart.

Final: After the 2 influential plots were removed, the plots show to be more spread apart, hence providing a better fit.

Comparing Competing Models

Predictive Models	Adjusted R2	CV PRESS	Kaggle Score
Simple Linear Regression	.5412	3.2550e12	.4258
Multiple Linear Regression	.5721	3.2437e12	.3819
Custom MLR Model	.7480	1.9675e12	.2281

From the comparison above we can see that the custom MLR model of (SalePrice ~ GrLivArea + OverallQual) has the best and lowest Kaggle score of 0.2281. This is because the model has a high adjusted R-squared and a lower CV press compared to the rest.

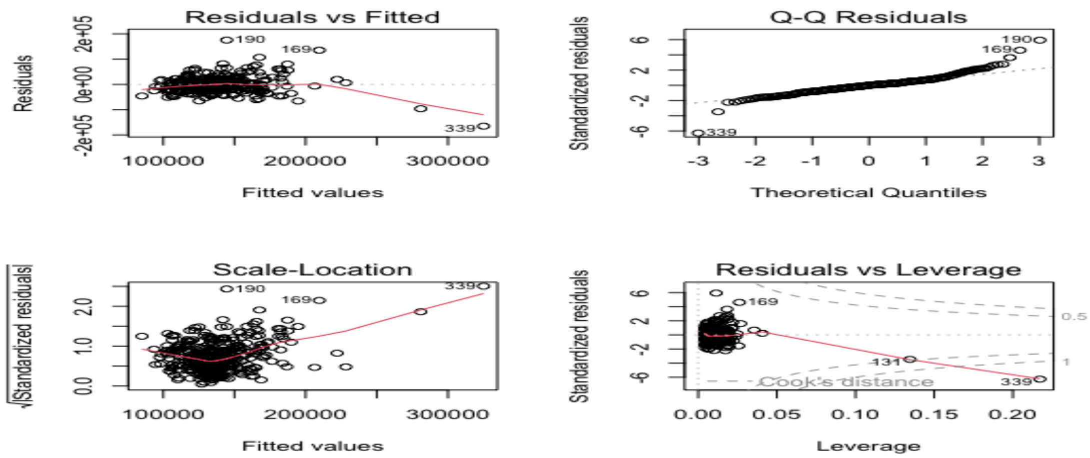
Conclusion

Based on the model selection analysis, the Kaggle scores, and the adjusted R-squared of each model shown in the table above, the custom MLR Model with a backward model selection is the best model to predict sale prices for homes in Ames, Iowa.

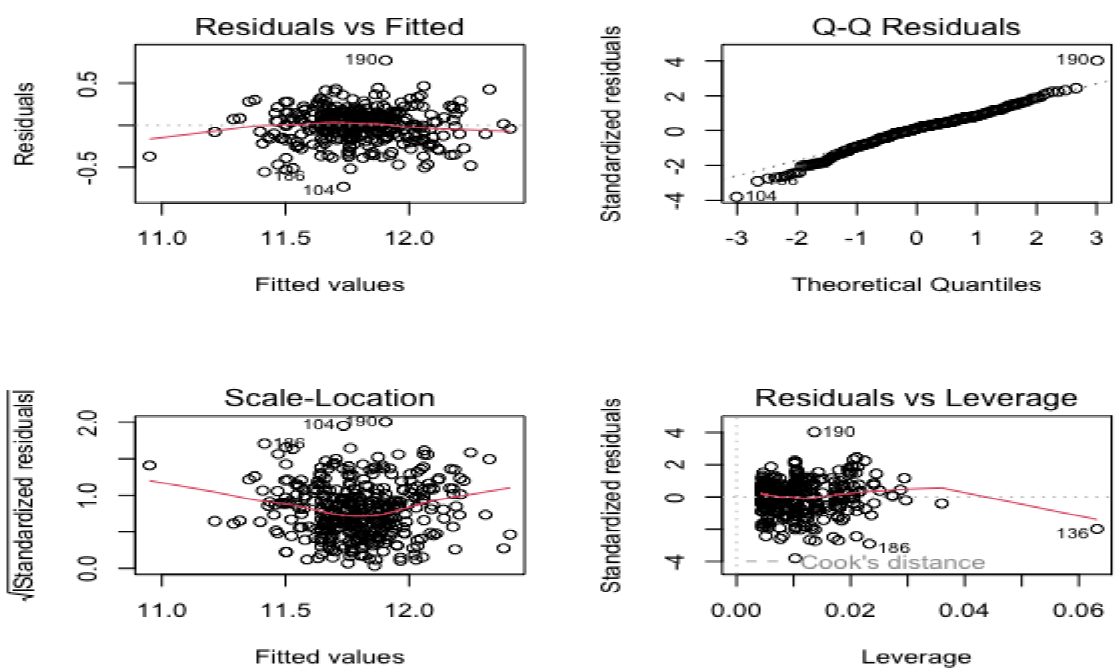
Index

Residual plots for analysis question 1

Original plot of the linear model before transformations or removing of outliers



Plot of the residuals after transformations and removing of outliers



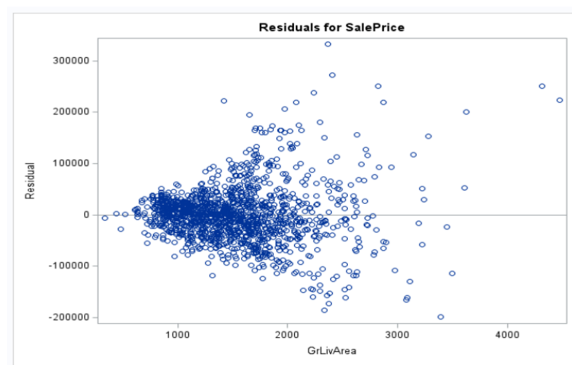
Residual Plots for analysis 2

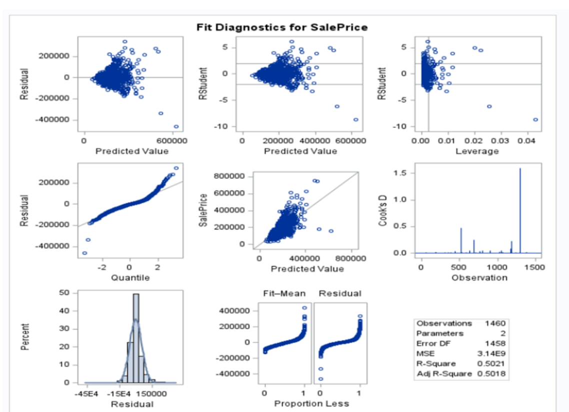
1. **Simple Linear Regressions**

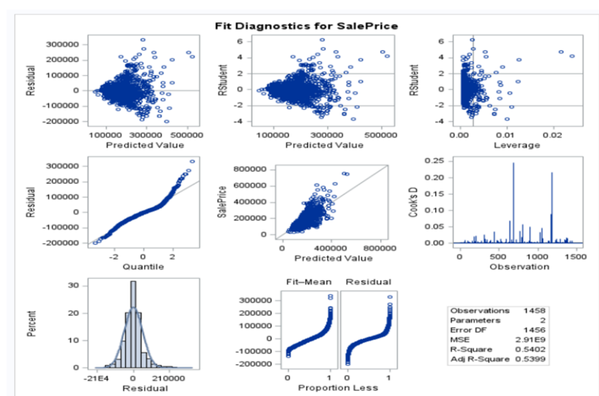
Residual Plots

Initial

Final

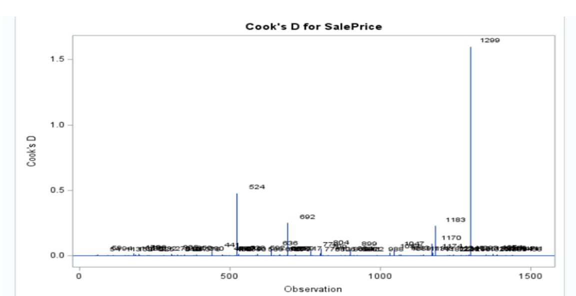




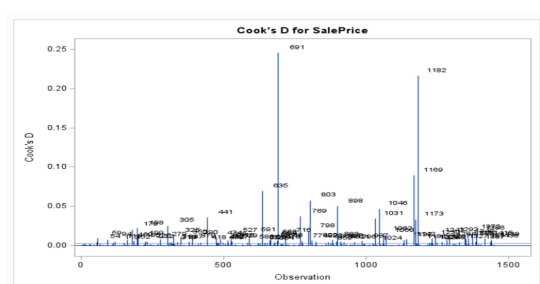


Cooks D-Plot

Initial

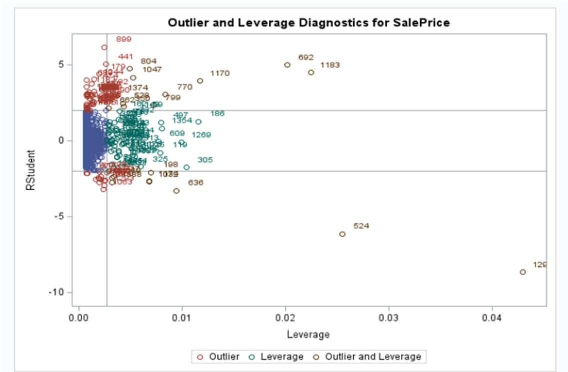


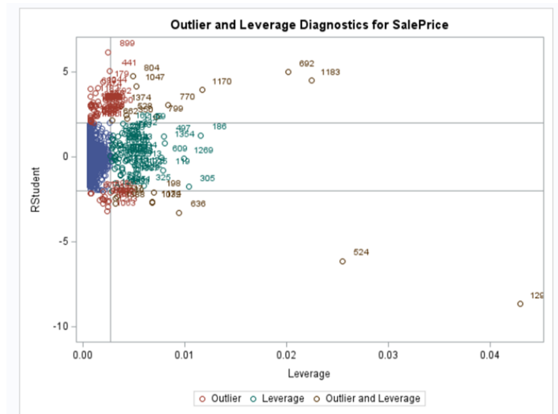
Final



Leverage Plot

Initial | Final

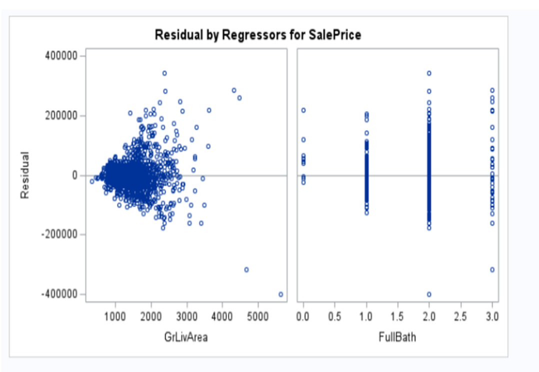


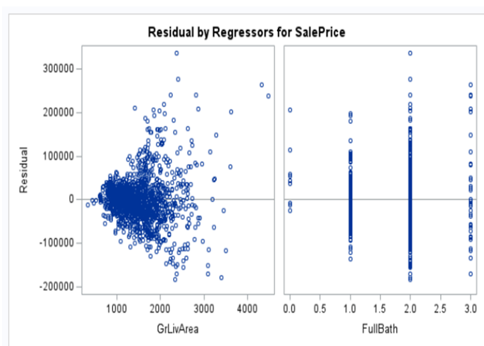


2. Multiple Linear Regressions

Residual Plots

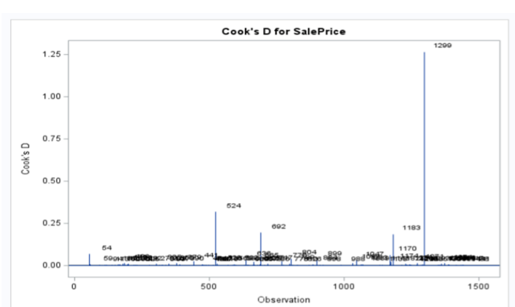
Initial | Final

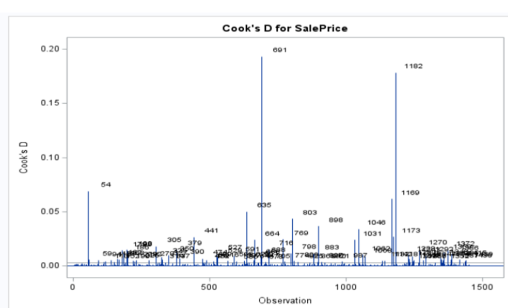




Cooks D-Plot

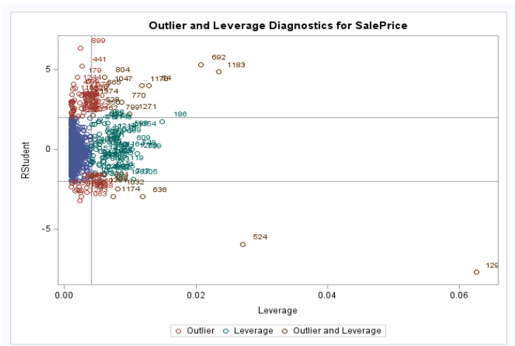
Initial | Final

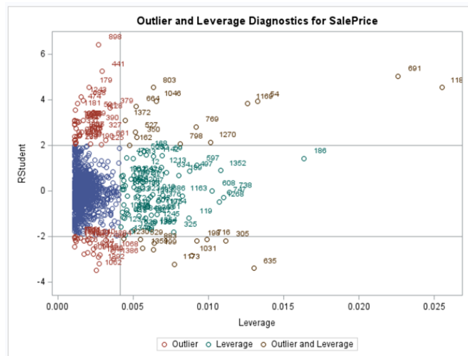




Leverage Plot

Initial | Final

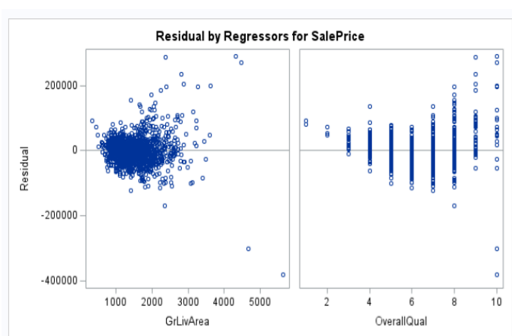


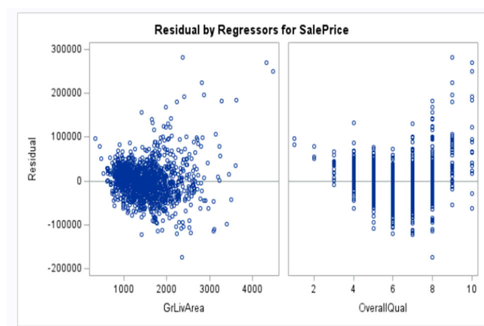


3. Custom Multiple Linear Regressions

Residual Plots

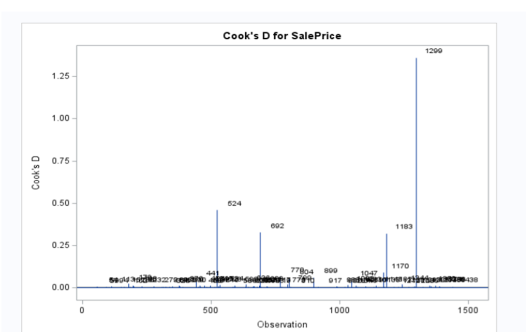
Initial | Final

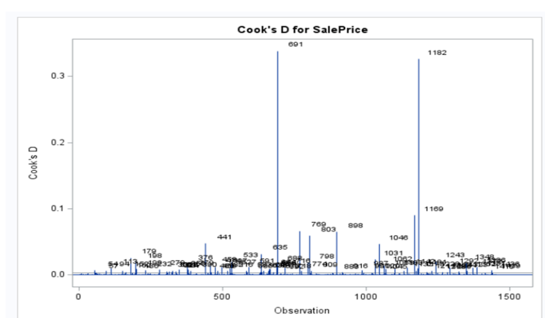




Cooks D-Plot

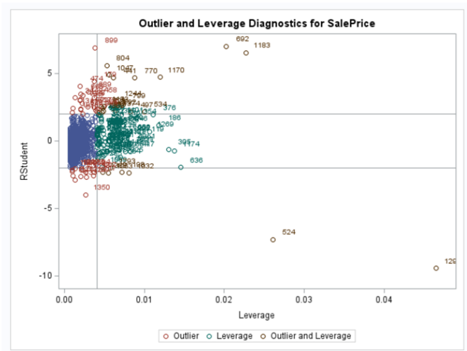
Initial | Final

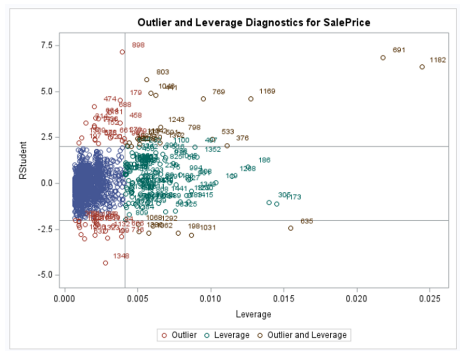




Leverage Plot

Initial Final





Codes

Analysis 1 R code:

```
#Final project real estate analysis

#final project stats
library(dplyr)
library(ggplot2)
hd <- read.csv("/Users/ivanchavez/Library/CloudStorage/OneDrive-SouthernMethodistUniversity/DS 6371 -
Stats/test.csv", header = TRUE)
head(hd)

hdT <- read.csv("/Users/ivanchavez/Library/CloudStorage/OneDrive-SouthernMethodistUniversity/DS 6371 -
Stats/train.csv", header = TRUE)
head(hdT)

summary(hd)
# filtered the neighborhood to only have the values that Century 21 Ames handles
hd2 <- hd %>% filter(Neighborhood == "NAmes" | Neighborhood == "Edwards" | Neighborhood == "BrkSide")
head(hd2)

summary(hd2$Neighborhood)
hd2$Neighborhood

hdT2 <- hdT %>% filter(Neighborhood == "NAmes" | Neighborhood == "Edwards" | Neighborhood == "BrkSide")
head(hdT2)
# filtered out just the columns i wanted from the training set & test set

Columns_to_keep <- c("SalePrice", "Id", "LotArea", "Neighborhood","YearBuilt", "GrLivArea")
```

```
Columns_to_keep2 <- c("Id", "LotArea", "Neighborhood","YearBuilt", "GrLivArea")

hdT2Updated <- select(hdT2, Columns_to_keep)
head(hdT2Updated)
hd2Updated <- select(hd2, Columns_to_keep2)
head(hd2Updated)
head(hdT2Updated)
summary(hdT2Updated)

#checking for na values in the training dataset
sum(is.na(hd2Updated$SalePrice))
sum(is.na(hd2Updated$Id))
sum(is.na(hd2Updated$LotArea))
sum(is.na(hd2Updated$Neighborhood))
sum(is.na(hd2Updated$YearBuilt))
sum(is.na(hd2Updated$GrLivArea))
# Create a scatterplot of Living area v Sale Price on new data set - afte rlog transformation and removing of the
outlying values
hdT2Updated %>% ggplot(aes(x = GrLivAreaSqrt, y = SalePrice, color = Neighborhood)) +
  geom_point()  + geom_smooth(method = "lm", color = "red") +
  labs(x = "Living area in sqft", y = "Sale Price", title = "Scatterplot of Living area v Sale Price")


# multiple linear regression comparing the sq living area to the sale price & neighborhood
#no adjustments and no removing of outliers in this model
fit <- lm(SalePrice~GrLivArea + Neighborhood, data = hdT2Updated)
summary(fit)
par(mfrow=c(2,2))
plot(fit)

#looking at the cooks distance for our model
#cooks dtest
c_dist <- cooks.distance(fit)

# Plot Cook's distance
plot(c_dist, pch = 20, main = "Cook's Distance Plot", ylab = "Cook's Distance", xlab = "Observation")
abline(h = 4/length(c_dist), col = 'red', lty = 2)  # Threshold line

# Identify the observation with the highest Cook's distance
max_cooks_index <- which.max(c_dist)
max_cooks_index

#printing the observation with the max cook distance
hdT2Updated[339,]

# Identify the indices of the top 3 observations with the highest Cook's distance
top_indices <- order(c_dist, decreasing = TRUE)[1:3]
top_indices

#printing the top 3
hdT2Updated[339,]
hdT2Updated[131,]
hdT2Updated[169,]
```

```
#removing points 339 & 131 both are in the Edwards neighborhood so shows some similarity
#in the data points also these points have an abnormally low sale price for the grlivarea
#removing them to test how the model performs w/o these points
rr <- c(131,339)
dfTest <- hdT2Updated[-rr,]
summary(dfTest)
dfTest[339,]
dfTest[131,]

# Create a scatterplot of Living area v Sale Price on new data set with points removed
dfTest %>% ggplot(aes(x = GrLivArea, y = SalePrice, color = Neighborhood)) +
  geom_point()  + geom_smooth(method = "lm", color = "red") +
  labs(x = "Living area in sqft", y = "Sale Price", title = "Scatterplot of Living area v Sale Price")

# Create a scatterplot of Living area v Sale Price on new data set with points removed
dfTest %>% ggplot(aes(x = log(GrLivArea), y = log(SalePrice), color = Neighborhood)) +
  geom_point()  + geom_smooth(method = "lm", color = "red") +
  labs(x = "Living area in sqft", y = "Sale Price", title = "Scatterplot of Living area v Sale Price")

fit2 <- lm(SalePrice ~ GrLivArea + Neighborhood, data = dfTest)
summary(fit2)
plot(fit2)
# creating our model with log of the sale price and GrLivArea
fit <- lm(log(SalePrice)~log(GrLivArea) + Neighborhood, data = dfTest)
summary(fit)
par(mfrow=c(2,2))
plot(fit)

confint(fit)

#looking at the cooks distance for our new model
#cooks dtest
c_dist <- cooks.distance(fit)

# Plot Cook's distance
plot(c_dist, pch = 20, main = "Cook's Distance Plot", ylab = "Cook's Distance", xlab = "Observation")
abline(h = 4/length(c_dist), col = 'red', lty = 2)  # Threshold line

#performing the cv press
set.seed(123)  # for reproducibility

# Number of folds for cross-validation
num_folds <- 5

# Create an index for the folds
folds <- sample(rep(1:num_folds, length.out = nrow(dfTest)))

# Initialize a vector to store PRESS values
press_values <- numeric(num_folds)

# Perform cross-validation
for (i in 1:num_folds) {
  # Split the data into training and test sets
```

```
  train_data <- dfTest[folds != i, ]
  test_data <- dfTest[folds == i, ]

  # Fit the model on the training set
  fit <- lm(log(SalePrice)~log(GrLivArea) + Neighborhood, data = dfTest)

  # Make predictions on the test set
  predicted <- predict(fit, newdata = test_data)

  # Calculate PRESS for this fold
  press_values[i] <- sum((test_data$Y - predicted)^2)
}

# Calculate overall cross-validated PRESS
cv_press <- sum(press_values)

# Print or use cv_press as needed
print(cv_press)
```

Analysis 2 SAS code:

```
*To import test data;
FILENAME REFFILE
"C:\Users\owola\Documents\MY_COURSES\FALL_2023\DS_6371_Stats_Foundations\Project\house-prices-advance
d-regression-techniques\test.csv";
PROC IMPORT DATAFILE=REFFILE
DBMS=CSV
OUT=testData;
GETNAMES=YES;
RUN;

proc print data = testData;
run;

*To import train data;
FILENAME REFFILE
"C:\Users\owola\Documents\MY_COURSES\FALL_2023\DS_6371_Stats_Foundations\Project\house-prices-advance
d-regression-techniques\train.csv";
PROC IMPORT DATAFILE=REFFILE
DBMS=CSV
OUT=trainData;
GETNAMES=YES;
RUN;
```

```
proc print data = trainData;
run;

*Getting the log data for trainData;
data trainData;
set trainData;
log_GrLivArea = log(GrLivArea);
log_SalePrice = log(SalePrice);
log_overallQual = log(overallQual);
log_FullBath = log(FullBath);
run;

*Plotting the single linear regression for the sales price and living area;
*Linear-Linear plot;
proc corr; run;
symbol c=blue v= dot;
proc sgscatter data = trainData;
matrix SalePrice GrLivArea;

*Creating the Sinmple Linear Regression;
*Finding the right variable;
/*
proc sgscatter data = trainData;
matrix SalePrice MSSubClass LotArea OverallQual OverallCond MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF
TotalBsmtSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr
TotRmsAbvGrd GarageCars GarageArea GarageQual WoodDeckSF   OpenPorchSF EnclosedPorch 3SsnPorch
ScreenPorch PoolArea MiscVal MoSold YrSold;
*/

proc reg data = trainData;
model SalePrice = GrLivArea;
run;

*Getting the labelled cooks data and leverages;
proc reg data=trainData  plots(only label) =(CooksD RStudentByLeverage);
  model SalePrice = GrLivArea; /* can also use INFLUENCE option */
run;


data NewtrainData;
set trainData;
if _n_ = 1299 then delete;
if _n_ = 524 then delete;
run;

proc print data = NewtrainData;
```

```
run;

proc reg data = NewtrainData;
model SalePrice = GrLivArea;
run;

*Getting the labelled cooks data and leverages;
proc reg data = NewtrainData  plots(only label) =(CooksD RStudentByLeverage);
  model SalePrice = GrLivArea; /* can also use INFLUENCE option */
run;

*Running the model selection with the train/test split;
*running the forward selection;
proc glmselect data = NewtrainData plots = all;
partition fraction(test= 0.2);
model SalePrice = GrLivArea /selection = Forward(stop=CV) cvmethod=random(5) stats = adjrsq CVDETAILS;
run;

*running the Backward selection;
proc glmselect data = NewtrainData plots = all;
partition fraction(test= 0.2);
model SalePrice = GrLivArea /selection = Backward(stop=CV) cvmethod=random(5) stats = adjrsq CVDETAILS;
run;

*running the Stepwise selection;
proc glmselect data = NewtrainData plots = all;
partition fraction(test= 0.2);
model SalePrice = GrLivArea /selection = Stepwise(stop=CV) cvmethod=random(5) stats = adjrsq CVDETAILS;
run;



*Predicting the sales price with the testdata for the simple linear regression;

*Creating a new dataset with testData;
data testData;
set testData;
SalePrice = .;

*Creating a new train dataset;
data NewtrainDataSLR;
set trainData testData;
run;

*Since the forward model has the highest p-value, we shall go ahead with it.;
```

```
proc glmselect data = NewtrainDataSLR plots = all;
class GrLivArea;
model SalePrice = GrLivArea /selection = Forward(stop=CV) cvmethod=random(5) stats = adjrsq CVDETAILS;
output out = resultsSLR p = predict;
run;

*Cant have -ve predictions bcos of RMSLE;
*Also must have only 2 columns with appropriate labels;

data resultsSLR2;
set resultsSLR;
if predict > 0 then SalePrice = Predict;
if predict < 0 then SalePrice = 10000;
keep id SalePrice;
where id > 1460;
;

proc means data = resultSLR2;
var SalePrice;
run;



*Multiple Linear Regression;
proc corr; run;
symbol c=blue v= dot;
proc sgscatter data = trainData;
matrix SalePrice GrLivArea FullBath;

proc reg data = trainData;
model SalePrice = GrLivArea FullBath;
run;

*Getting the labelled cooks data and leverages;
proc reg data=trainData  plots(only label) =(CooksD RStudentByLeverage);
   model SalePrice = GrLivArea FullBath; /* can also use INFLUENCE option */
run;

data NewtrainData2;
set trainData;
if _n_ = 1299 then delete;
if _n_ = 524 then delete;
run;

proc print data = NewtrainData2;
run;
```

```
proc reg data = NewtrainData2;
model SalePrice = GrLivArea FullBath;
run;

*Getting the labelled cooks data and leverages;
proc reg data = NewtrainData2  plots(only label) =(CooksD RStudentByLeverage);
  model SalePrice = GrLivArea FullBath; /* can also use INFLUENCE option */
run;

*running the forward selection;
proc glmselect data = NewtrainData2 plots = all;
partition fraction(test= 0.2);
model SalePrice = GrLivArea FullBath /selection = Forward(select=CV choose=CV stop=CV) cvmethod=random(5)
stats = adjrsq CVDETAILS;
run;

*running the Backward selection;
proc glmselect data = NewtrainData2 plots = all;
partition fraction(test= 0.2);
model SalePrice = GrLivArea FullBath /selection = Backward(stop=CV) cvmethod=random(5) stats = adjrsq
CVDETAILS;
run;

*running the Stepwise selection;
proc glmselect data = NewtrainData2 plots = all;
partition fraction(test= 0.2);
model SalePrice = GrLivArea FullBath /selection = Stepwise(stop=CV) cvmethod=random(5) stats = adjrsq
CVDETAILS;
run;

*Predicting the sales price with the testdata for the multiple linear regression;

*Creating a new dataset with testData;
data testData;
set testData;
SalePrice = .;

*Creating a new train dataset;
data NewtrainDataMLR;
set trainData testData;
run;

*Since the stepwise model has the highest p-value, we shall go ahead with it.;
proc glmselect data = NewtrainDataMLR plots = all;
class GrLivArea FullBath;
```

```
model SalePrice = GrLivArea FullBath /selection = Stepwise(stop=CV) cvmethod=random(5) stats = adjrsq
CVDETAILS;
output out = resultsMLR p = predictMLR;
run;

*Cant have -ve predictions bcos of RMSLE;
*Also must have only 2 columns with appropriate labels;

data resultsMLR2;
set resultsMLR;
if predictMLR > 0 then SalePrice = PredictMLR;
if predictMLR < 0 then SalePrice = 10000;
keep id SalePrice;
where id > 1460;
;

proc means data = resultsMLR2;
var SalePrice;
run;



*Custom Multiple Linear Regression (SalePrice ~ GrLivArea + OverallQual);
proc corr; run;
symbol c=blue v= dot;
proc sgscatter data = trainData;
matrix SalePrice GrLivArea OverallQual;

proc reg data = trainData;
model SalePrice = GrLivArea OverallQual;
run;

*Getting the labelled cooks data and leverages;
proc reg data=trainData  plots(only label) =(CooksD RStudentByLeverage);
  model SalePrice = GrLivArea OverallQual; /* can also use INFLUENCE option */
run;

data NewtrainData3;
set trainData;
if _n_ = 1299 then delete;
if _n_ = 524 then delete;
run;

proc print data = NewtrainData3;
run;
```

```
proc reg data = NewtrainData3;
model SalePrice = GrLivArea OverallQual;
run;

*Getting the labelled cooks data and leverages;
proc reg data = NewtrainData3  plots(only label) =(CooksD RStudentByLeverage);
  model SalePrice = GrLivArea OverallQual; /* can also use INFLUENCE option */
run;

*running the forward selection;
proc glmselect data = NewtrainData3 plots = all;
partition fraction(test= 0.2);
model SalePrice = GrLivArea OverallQual /selection = Forward(select=CV choose=CV stop=CV)
cvmethod=random(5) stats = adjrsq CVDETAILS;
run;

*running the Backward selection;
proc glmselect data = NewtrainData3 plots = all;
partition fraction(test= 0.2);
model SalePrice = GrLivArea OverallQual /selection = Backward(stop=CV) cvmethod=random(5) stats = adjrsq
CVDETAILS;
run;

*running the Stepwise selection;
proc glmselect data = NewtrainData3 plots = all;
partition fraction(test= 0.2);
model SalePrice = GrLivArea OverallQual /selection = Stepwise(stop=CV) cvmethod=random(5) stats = adjrsq
CVDETAILS;
run;

*Predicting the sales price with the testdata for the custom linear regression;

*Creating a new dataset with testData;
data testData;
set testData;
SalePrice = .;

*Creating a new train dataset;
data NewtrainDataCLR;
set trainData testData;
run;

*Since the backward model has the highest p-value, we shall go ahead with it.;
proc glmselect data = NewtrainDataCLR plots = all;
class GrLivArea OverallQual;
```

```
model SalePrice = GrLivArea OverallQual /selection = Backward(stop=CV) cvmethod=random(5) stats = adjrsq
CVDETAILS;
output out = resultsCLR p = predictCLR;
run;

*Cant have -ve predictions bcos of RMSLE;
*Also must have only 2 columns with appropriate labels;

data resultsCLR2;
set resultsCLR;
if predictCLR > 0 then SalePrice = PredictCLR;
if predictCLR < 0 then SalePrice = 10000;
keep id SalePrice;
where id > 1460;
;proc means data = resultsCLR2;
var SalePrice;
Run;
```

R Shiny app code:

```
library(shiny)
library(ggplot2)
library(readr)



Data <- read.csv("train.csv")

sorted_neighborhoods <- c("NAmes", "Edwards", "BrkSide")

ui <- fluidPage(
  titlePanel("Real Estate Analysis"),
  sidebarLayout(
    sidebarPanel(
      radioButtons("plot_type", "Select Plot Type", choices = c("Scatterplot")),
      br(),
      selectInput("neighborhood_filter", "Filter by Neighborhood", choices = c("All",sorted_neighborhoods)),
      br(),
      checkboxInput("add_regression", "Add Linear Regression Line"),
    ),
    mainPanel(
      plotOutput("data_plot")
    )
  )
)
```

```
server <- function(input, output) {
  output$data_plot <- renderPlot({
    plot_data <- Data

    # Filter data based on the selected neighborhood
    if (input$neighborhood_filter != "All") {
      plot_data <- plot_data[plot_data$Neighborhood == input$neighborhood_filter, ]
    }

    p <- ggplot()  # Initialize ggplot object

    if (input$plot_type == "Scatterplot") {
      p <- p + geom_point(data = plot_data, aes(x = GrLivArea, y = SalePrice), color = "blue") +
        labs(title = "Scatterplot of Sqft. Living area vs. Sale Price", x = "Sq foot iving area", y = "Sale Price")
    }

    if (input$add_regression) {
      p <- p + geom_smooth(data = plot_data,aes(x = GrLivArea, y = SalePrice), method = "lm", color = "red")
    }

    print(p)  # Print the ggplot object
  })
}

shinyApp(ui = ui, server = server)
```